



# Stellaris<sup>®</sup> LM4F232H5QD 微处理器


数据手册

---

# 版权

版权 © 2007-2012 Texas Instruments Incorporated 保留所有权利。Stellaris 以及 StellarisWare® 是 Texas Instruments Incorporated 的注册商标。ARM 和 Thumb 是 ARM 公司的注册商标，Cortex 是 ARM 公司的商标。其它名称和商标均为其所有者的财产。

预告信息涉及样品阶段和产品试制阶段的新产品。特征数据和其他规格都可能在没有预先通告的情况下改变。

 请留意，此数据手册最后部分包含 Texas Instruments 半导体产品的可用性、标准保修期和关键任务应用的重要信息通告以及相关的免责声明。

Texas Instruments Incorporated  
108 Wild Basin, Suite 350  
Austin, TX 78746  
<http://www.ti.com/stellaris>  
<http://www-k.ext.ti.com/sc/technical-support/product-information-centers.htm>



**TEXAS  
INSTRUMENTS**



**Cortex**  
Intelligent Processors by ARM™



# 目录

修订历史 .....	<b>38</b>
关于本文档 .....	<b>42</b>
读者 .....	42
关于本手册 .....	42
相关文档 .....	42
文档约定 .....	43
<b>1 结构概述 .....</b>	<b>45</b>
1.1 Stellaris LM4F 系列概述 .....	45
1.2 LM4F232H5QD 微控制器概述 .....	48
1.3 LM4F232H5QD 微控制器特性 .....	51
1.3.1 ARM Cortex-M4F 处理器内核 .....	51
1.3.2 片上存储器 .....	53
1.3.3 串行通讯外设 .....	54
1.3.4 系统集成 .....	58
1.3.5 高级运动控制 .....	64
1.3.6 模拟 .....	66
1.3.7 JTAG 和 ARM 串行线调试 .....	67
1.3.8 封装和温度 .....	68
1.4 LM4F232H5QD 微控制器硬件细节 .....	68
<b>2 Cortex-M4F 处理器 .....</b>	<b>69</b>
2.1 结构图 .....	70
2.2 概述 .....	70
2.2.1 系统级接口 .....	70
2.2.2 集成的可配置调试 .....	71
2.2.3 跟踪端口接口单元 (TPIU) .....	71
2.2.4 Cortex-M4F 系统组件细节 .....	72
2.3 编程模型 .....	72
2.3.1 处理器模式和软件执行的权限级别 .....	72
2.3.2 堆栈 .....	73
2.3.3 寄存器映射 .....	73
2.3.4 寄存器描述 .....	75
2.3.5 异常和中断 .....	91
2.3.6 数据类型 .....	91
2.4 存储模型 .....	91
2.4.1 存储器区、类型和属性 .....	94
2.4.2 存储器访问存储系统顺序 .....	94
2.4.3 存储器访问行为 .....	94
2.4.4 存储器访问的软件顺序 .....	95
2.4.5 位带区 .....	96
2.4.6 数据保存 .....	98
2.4.7 同步原语 .....	98
2.5 异常模式 .....	99
2.5.1 异常状态 .....	99
2.5.2 异常类型 .....	100
2.5.3 异常处理程序 .....	104

2.5.4	向量表 .....	104
2.5.5	异常优先级 .....	105
2.5.6	中断优先级分组 .....	106
2.5.7	异常进入和返回 .....	106
2.6	故障处理 .....	108
2.6.1	故障类型 .....	109
2.6.2	故障扩大和硬件故障 .....	109
2.6.3	故障状态寄存器和故障地址寄存器 .....	110
2.6.4	死锁 .....	110
2.7	电源管理 .....	110
2.7.1	进入睡眠模式 .....	110
2.7.2	从睡眠模式唤醒 .....	111
2.7.3	唤醒中断控制器 .....	111
2.8	指令集总结 .....	111
<b>3</b>	<b>Cortex-M4 外设 .....</b>	<b>118</b>
3.1	功能描述 .....	118
3.1.1	系统定时器 ( SysTick ) .....	118
3.1.2	嵌套向量中断控制器 ( NVIC ) .....	119
3.1.3	系统控制模块 ( SCB ) .....	120
3.1.4	存储器保护单元 ( MPU ) .....	121
3.1.5	浮点单元 ( FPU ) .....	125
3.2	寄存器映射 .....	128
3.3	系统定时器 ( SysTick ) 寄存器描述 .....	131
3.4	NVIC寄存器描述 .....	135
3.5	系统控制模块 ( SCB ) 寄存器描述 .....	150
3.6	存储器保护单元 ( MPU ) 寄存器描述 .....	178
3.7	浮点单元 ( FPU ) 寄存器描述 .....	186
<b>4</b>	<b>JTAG 接口 .....</b>	<b>192</b>
4.1	结构图 .....	193
4.2	信号描述 .....	193
4.3	功能描述 .....	194
4.3.1	JTAG 接口管脚 .....	194
4.3.2	JTAG TAP 控制器 .....	195
4.3.3	移位寄存器 .....	196
4.3.4	操作注意事项 .....	196
4.4	初始化及配置 .....	199
4.5	寄存器描述 .....	199
4.5.1	指令寄存器 ( IR ) .....	199
4.5.2	数据寄存器 .....	201
<b>5</b>	<b>系统控制 .....</b>	<b>203</b>
5.1	信号描述 .....	203
5.2	功能描述 .....	203
5.2.1	器件标识 .....	203
5.2.2	复位控制 .....	203
5.2.3	不可屏蔽的中断 .....	208
5.2.4	功率控制 .....	208
5.2.5	时钟控制 .....	209
5.2.6	系统控制 .....	215

5.3	初始化及配置 .....	217
5.4	寄存器映射 .....	217
5.5	系统控制寄存器描述 .....	222
5.6	系统控制传统寄存器描述 .....	412
<b>6</b>	<b>系统异常模块 .....</b>	<b>467</b>
6.1	功能描述 .....	467
6.2	寄存器映射 .....	467
6.3	寄存器描述 .....	467
<b>7</b>	<b>休眠模块 .....</b>	<b>475</b>
7.1	结构图 .....	476
7.2	信号描述 .....	476
7.3	功能描述 .....	477
7.3.1	寄存器访问时序 .....	477
7.3.2	休眠时钟源 .....	477
7.3.3	系统实现 .....	479
7.3.4	电池管理 .....	479
7.3.5	实时时钟 .....	480
7.3.6	带备用电池的存储器 .....	481
7.3.7	使用 FIB 控制电源 .....	481
7.3.8	以 VDD3ON 模式管理电源 .....	481
7.3.9	启动休眠 .....	482
7.3.10	从休眠模式唤醒 .....	482
7.3.11	仲裁性电源移除 .....	482
7.3.12	中断和状态 .....	482
7.4	初始化及配置 .....	482
7.4.1	初始化 .....	483
7.4.2	RTC 匹配功能 ( 无休眠 ) .....	483
7.4.3	RTC 匹配/唤醒 .....	484
7.4.4	外部唤醒 .....	484
7.4.5	RTC 或外部唤醒 .....	484
7.5	寄存器映射 .....	484
7.6	寄存器描述 .....	485
<b>8</b>	<b>内部存储器 .....</b>	<b>503</b>
8.1	结构图 .....	503
8.2	功能描述 .....	503
8.2.1	SRAM .....	504
8.2.2	ROM .....	504
8.2.3	Flash 存储器 .....	506
8.3	寄存器映射 .....	510
8.4	Flash 存储器寄存器描述 ( Flash 控制偏移量 ) .....	511
8.5	存储器寄存器描述 ( 系统控制偏移量 ) .....	527
<b>9</b>	<b>微型直接存储器访问 ( <math>\mu</math>DMA ) .....</b>	<b>534</b>
9.1	结构图 .....	535
9.2	功能描述 .....	535
9.2.1	通道分配 .....	535
9.2.2	优先级 .....	537
9.2.3	仲裁数目 .....	537

9.2.4	请求类型 .....	537
9.2.5	通道配置 .....	538
9.2.6	传输模式 .....	539
9.2.7	待传输数目及增量 .....	548
9.2.8	外设接口 .....	548
9.2.9	软件请求 .....	548
9.2.10	中断及错误 .....	548
9.3	初始化及配置 .....	549
9.3.1	模块初始化 .....	549
9.3.2	存储器到存储器传输的配置 .....	549
9.3.3	外设简单发送的配置 .....	551
9.3.4	外设乒乓接收的配置 .....	552
9.3.5	通道分配的配置 .....	554
9.4	寄存器映射 .....	554
9.5	μDMA 通道控制结构体 .....	556
9.6	μDMA 寄存器描述 .....	563
<b>10</b>	<b>通用输入/输出端口 (GPIO) .....</b>	<b>597</b>
10.1	信号描述 .....	597
10.2	功能描述 .....	601
10.2.1	数据控制 .....	602
10.2.2	中断控制 .....	603
10.2.3	模式控制 .....	604
10.2.4	提交控制 .....	604
10.2.5	管脚控制 .....	604
10.2.6	标识 .....	605
10.3	初始化及配置 .....	605
10.4	寄存器映射 .....	606
10.5	寄存器描述 .....	608
<b>11</b>	<b>通用定时器 .....</b>	<b>653</b>
11.1	结构图 .....	654
11.2	信号描述 .....	655
11.3	功能描述 .....	657
11.3.1	GPTM 复位条件 .....	658
11.3.2	定时器模式 .....	658
11.3.3	等待触发模式 .....	667
11.3.4	同步通用定时器模块 .....	667
11.3.5	DMA 操作 .....	668
11.3.6	访问连接的 16/32 位 GPTM 寄存器值 .....	668
11.3.7	访问连接的 32/64 位宽 GPTM 寄存器值 .....	669
11.4	初始化和配置 .....	670
11.4.1	单次触发/周期定时器模式 .....	670
11.4.2	实时时钟 (RTC) 模式 .....	671
11.4.3	输入边沿计数模式 .....	671
11.4.4	输入边沿定时模式 .....	672
11.4.5	PWM 模式 .....	672
11.5	寄存器映射 .....	673
11.6	寄存器描述 .....	674

<b>12</b>	<b>看门狗定时器 (WDT)</b> .....	<b>717</b>
12.1	结构图 .....	718
12.2	功能说明 .....	718
12.2.1	寄存器访问时序 .....	719
12.3	初始化及配置 .....	719
12.4	寄存器映射 .....	719
12.5	寄存器描述 .....	720
<b>13</b>	<b>模数转换器 (ADC)</b> .....	<b>742</b>
13.1	结构图 .....	743
13.2	信号描述 .....	744
13.3	功能描述 .....	745
13.3.1	采样序列发生器 .....	745
13.3.2	模块控制 .....	745
13.3.3	硬件采样平均电路 .....	748
13.3.4	模数转换器 .....	749
13.3.5	差分采样 .....	752
13.3.6	内部温度传感器 .....	754
13.3.7	数字比较器 .....	755
13.4	初始化及配置 .....	759
13.4.1	模块初始化 .....	759
13.4.2	采样序列发生器的配置 .....	760
13.5	寄存器映射 .....	760
13.6	寄存器描述 .....	762
<b>14</b>	<b>通用异步收发器 (UART)</b> .....	<b>827</b>
14.1	结构图 .....	828
14.2	信号描述 .....	828
14.3	功能描述 .....	829
14.3.1	发送/接收逻辑 .....	830
14.3.2	波特率的产生 .....	830
14.3.3	数据传输 .....	831
14.3.4	串行红外 (SIR) .....	831
14.3.5	对 ISO 7816 的支持 .....	832
14.3.6	对调制解调器握手信号的支持 .....	832
14.3.7	对 LIN 的支持 .....	834
14.3.8	9 位 UART 模式 .....	835
14.3.9	FIFO 操作 .....	835
14.3.10	中断 .....	835
14.3.11	回送操作 .....	836
14.3.12	DMA 操作 .....	836
14.4	初始化及配置 .....	837
14.5	寄存器映射 .....	838
14.6	寄存器描述 .....	839
<b>15</b>	<b>同步串行接口 (SSI)</b> .....	<b>891</b>
15.1	结构图 .....	892
15.2	信号描述 .....	892
15.3	功能描述 .....	893
15.3.1	位速率的产生 .....	893
15.3.2	FIFO 操作 .....	894

15.3.3	中断 .....	894
15.3.4	帧格式 .....	895
15.3.5	DMA 操作 .....	901
15.4	初始化及配置 .....	902
15.5	寄存器映射 .....	903
15.6	寄存器描述 .....	904
<b>16</b>	<b>内部集成电路 (I<sup>2</sup>C) 接口 .....</b>	<b>933</b>
16.1	结构图 .....	934
16.2	信号描述 .....	934
16.3	功能描述 .....	935
16.3.1	I <sup>2</sup> C 总线功能概览 .....	935
16.3.2	可用的速度模式 .....	938
16.3.3	中断 .....	940
16.3.4	回送操作 .....	940
16.3.5	指令序列流程图 .....	941
16.4	初始化及配置 .....	949
16.5	寄存器映射 .....	950
16.6	寄存器描述 (I <sup>2</sup> C 主机) .....	951
16.7	寄存器描述 (I <sup>2</sup> C 从机) .....	966
16.8	寄存器描述 (I <sup>2</sup> C 状态和控制寄存器) .....	976
<b>17</b>	<b>控制器局域网 (CAN) 模块 .....</b>	<b>979</b>
17.1	结构图 .....	980
17.2	信号描述 .....	980
17.3	功能描述 .....	981
17.3.1	初始化 .....	982
17.3.2	基本操作 .....	982
17.3.3	报文对象的发送 .....	983
17.3.4	待发送报文对象的配置 .....	983
17.3.5	待发送报文对象的刷新 .....	984
17.3.6	已接收报文对象的验收 .....	985
17.3.7	接收数据帧 .....	985
17.3.8	接收远程帧 .....	985
17.3.9	接收/发送优先级 .....	986
17.3.10	接收报文对象的配置 .....	986
17.3.11	已接收报文对象的处理 .....	987
17.3.12	中断的处理 .....	988
17.3.13	测试模式 .....	989
17.3.14	位时序配置错误的注意事项 .....	990
17.3.15	位时间与位速率 .....	990
17.3.16	位时序参数的计算 .....	992
17.4	寄存器映射 .....	995
17.5	CAN 寄存器描述 .....	997
<b>18</b>	<b>通用串行总线 (USB) 控制器 .....</b>	<b>1026</b>
18.1	结构图 .....	1027
18.2	信号描述 .....	1027
18.3	功能描述 .....	1028
18.3.1	用作设备时的操作 .....	1028



18.3.2	用作主机时的操作 .....	1032
18.3.3	OTG模式 .....	1035
18.3.4	DMA操作 .....	1036
18.4	初始化及配置 .....	1037
18.4.1	管脚配置 .....	1037
18.4.2	端点配置 .....	1037
18.5	寄存器映射 .....	1038
18.6	寄存器描述 .....	1043
<b>19</b>	<b>模拟比较器 .....</b>	<b>1136</b>
19.1	结构图 .....	1136
19.2	信号描述 .....	1137
19.3	功能描述 .....	1137
19.3.1	内部参考电压编程 .....	1138
19.4	初始化及配置 .....	1140
19.5	寄存器映射 .....	1141
19.6	寄存器描述 .....	1141
<b>20</b>	<b>脉宽调制器 (PWM) .....</b>	<b>1151</b>
20.1	结构图 .....	1152
20.2	信号描述 .....	1153
20.3	功能描述 .....	1155
20.3.1	PWM 定时器 .....	1155
20.3.2	PWM 比较器 .....	1155
20.3.3	PWM 信号发生器 .....	1156
20.3.4	死区发生器 .....	1157
20.3.5	中断/ADC-触发选择器 .....	1157
20.3.6	同步方法 .....	1157
20.3.7	故障条件 .....	1158
20.3.8	输出控制块 .....	1159
20.4	初始化及配置 .....	1159
20.5	寄存器映射 .....	1160
20.6	寄存器描述 .....	1163
<b>21</b>	<b>正交编码器接口 (QEI) .....</b>	<b>1224</b>
21.1	结构图 .....	1224
21.2	信号描述 .....	1225
21.3	功能描述 .....	1226
21.4	初始化及配置 .....	1228
21.5	寄存器映射 .....	1228
21.6	寄存器描述 .....	1229
<b>22</b>	<b>管脚图 .....</b>	<b>1244</b>
<b>23</b>	<b>信号表 .....</b>	<b>1245</b>
23.1	按管脚编号分类的信号 .....	1246
23.2	按信号名称分类的信号 .....	1259
23.3	按功能分类的信号 ( GPIO 除外 ) .....	1271
23.4	GPIO 管脚和复用功能 .....	1280
23.5	复用功能的可能的管脚赋值 .....	1284
23.6	未用信号的连接 .....	1288

---

<b>24</b>	<b>工作特性 .....</b>	<b>1290</b>
<b>25</b>	<b>电气特性 .....</b>	<b>1291</b>
25.1	Maximum Ratings .....	1291
25.2	Recommended Operating Conditions .....	1292
25.3	Load Conditions .....	1293
25.4	JTAG and Boundary Scan .....	1294
25.5	Power and Brown-Out .....	1295
25.6	Reset .....	1296
25.7	On-Chip Low Drop-Out (LDO) Regulator .....	1297
25.8	Clocks .....	1298
25.8.1	PLL Specifications .....	1298
25.8.2	PIOSC Specifications .....	1299
25.8.3	Internal 30-kHz Oscillator Specifications .....	1299
25.8.4	Hibernation Clock Source Specifications .....	1299
25.8.5	Main Oscillator Specifications .....	1300
25.8.6	System Clock Specification with ADC Operation .....	1303
25.8.7	System Clock Specification with USB Operation .....	1303
25.9	Sleep Modes .....	1304
25.10	Hibernation Module .....	1304
25.11	Flash Memory .....	1305
25.12	Input/Output Characteristics .....	1306
25.13	Analog-to-Digital Converter (ADC) .....	1307
25.14	Synchronous Serial Interface (SSI) .....	1310
25.15	Inter-Integrated Circuit (I <sup>2</sup> C) Interface .....	1312
25.16	Universal Serial Bus (USB) Controller .....	1312
25.17	Analog Comparator .....	1312
25.18	Current Consumption .....	1314
25.18.1	Preliminary Current Consumption .....	1314
<b>A</b>	<b>寄存器快速参考 .....</b>	<b>1316</b>
<b>B</b>	<b>订购和联系信息 .....</b>	<b>1374</b>
B.1	订购信息 .....	1374
B.2	器件标记 .....	1374
B.3	套件 .....	1374
B.4	支持信息 .....	1375
<b>C</b>	<b>封装信息 .....</b>	<b>1376</b>
C.1	144 管脚 LQFP 封装 .....	1376
C.1.1	封装尺寸 .....	1376

## 插图清单

图 1-1.	Stellaris LM4F 结构图 .....	46
图 1-2.	Stellaris LM4F232H5QD 微控制器高级框图 .....	50
图 2-1.	CPU 结构图 .....	70
图 2-2.	TPIU 结构图 .....	71
图 2-3.	Cortex-M4F 的寄存器组。 .....	74
图 2-4.	位带映射 .....	97
图 2-5.	数据保存 .....	98
图 2-6.	向量表 .....	105
图 2-7.	异常堆栈框 .....	107
图 3-1.	SRD 使用示例 .....	123
图 3-2.	FPU 寄存器块 .....	126
图 4-1.	JTAG 模块结构图 .....	193
图 4-2.	测试访问端口状态机 .....	196
图 4-3.	IDCODE 寄存器格式 .....	201
图 4-4.	BYPASS 寄存器格式 .....	201
图 4-5.	边界扫描寄存器格式 .....	201
图 5-1.	基本 RST 配置 .....	205
图 5-2.	延长上电复位时间的外部电路 .....	206
图 5-3.	复位电路由开关控制 .....	206
图 5-4.	功率结构 .....	209
图 5-5.	主时钟树 .....	211
图 5-6.	模块时钟选择 .....	217
图 7-1.	休眠模块结构图 .....	476
图 7-2.	使用晶振作为休眠模块的时钟源 (单一电池源) .....	478
图 7-3.	使用专用振荡器作为休眠模块的时钟源 .....	478
图 7-4.	V <sub>DD</sub> 和 V <sub>BAT</sub> 使用稳压器 .....	479
图 8-1.	内部存储器结构图 .....	503
图 9-1.	μDMA 结构图 .....	535
图 9-2.	乒乓式 μDMA 数据会话的示例 .....	541
图 9-3.	存储器散聚模式: 创建及配置 .....	543
图 9-4.	存储器散聚模式: μDMA 复制序列 .....	544
图 9-5.	外设散聚模式: 创建及配置 .....	546
图 9-6.	外设散聚模式: μDMA 复制序列 .....	547
图 10-1.	数字 I/O 口 .....	601
图 10-2.	模拟/数字 I/O 口 .....	602
图 10-3.	GPIO_DATA 写入实例 .....	603
图 10-4.	GPIO_DATA 读取实例 .....	603
图 11-1.	GPTM 模块的结构图 .....	654
图 11-2.	读取 RTC 值 .....	661
图 11-3.	输入边沿计数模式实例, 递减计数 .....	663
图 11-4.	16 位输入边沿计时模式实例 .....	664
图 11-5.	16 位 PWM 模式实例 .....	665
图 11-6.	CCP 输出, GPTMTnMATCHR > GPTMTnILR .....	666
图 11-7.	CCP 输出, GPTMTnMATCHR = GPTMTnILR .....	666
图 11-8.	CCP 输出, GPTMTnILR > GPTMTnMATCHR .....	667
图 11-9.	定时器菊花链 .....	667

图 12-1.	WDT 模块的结构图 .....	718
图 13-1.	两个 ADC 模块的连接结构图 .....	743
图 13-2.	ADC 模块结构图 .....	743
图 13-3.	ADC 采样相位 .....	747
图 13-4.	ADC 采样率倍增 .....	747
图 13-5.	交错采样 .....	748
图 13-6.	采样平均的实例 .....	749
图 13-7.	ADC 输入端等效框图 .....	750
图 13-8.	ADC 参考电压 .....	751
图 13-9.	ADC 转换结果 .....	752
图 13-10.	差分电压表达式 .....	754
图 13-11.	内部温度传感器特性 .....	755
图 13-12.	低值带工作 ( CIC = 0x0 和/或 CTC = 0x0 ) .....	757
图 13-13.	中值带工作 ( CIC = 0x1 和/或 CTC = 0x1 ) .....	758
图 13-14.	高值带工作 ( CIC = 0x3 和/或 CTC = 0x3 ) .....	759
图 14-1.	UART 模块的结构图 .....	828
图 14-2.	UART 字符帧 .....	830
图 14-3.	IrDA 数据调制 .....	832
图 14-4.	LIN 报文 .....	834
图 14-5.	LIN 同步域 .....	835
图 15-1.	SSI 模块的结构图 .....	892
图 15-2.	TI 同步串行帧格式 ( 单次传输 ) .....	895
图 15-3.	TI 同步串行的帧格式 ( 连续传输 ) .....	896
图 15-4.	SPO = 0 和 SPH = 0 时 Freescale SPI 帧格式 ( 单次传输 ) .....	896
图 15-5.	SPO = 0 和 SPH = 0 时 Freescale SPI 的帧格式 ( 连续传输 ) .....	897
图 15-6.	SPO = 0 和 SPH = 1 时的 Freescale SPI 帧格式 .....	898
图 15-7.	SPO = 1 和 SPH = 0 时 Freescale SPI 的帧格式 ( 单次传输 ) .....	898
图 15-8.	SPO = 1 和 SPH = 0 时 Freescale SPI 的帧格式 ( 连续传输 ) .....	899
图 15-9.	SPO = 1 和 SPH = 1 时的 Freescale SPI 帧格式 .....	899
图 15-10.	MICROWIRE 帧格式 ( 单帧 ) .....	900
图 15-11.	MICROWIRE 帧格式 ( 连续传输 ) .....	901
图 15-12.	MICROWIRE 帧格式, SSIFss 输入建立和保持时间要求 .....	901
图 16-1.	I <sup>2</sup> C 结构图 .....	934
图 16-2.	I <sup>2</sup> C 总线配置 .....	935
图 16-3.	开始和停止条件 .....	936
图 16-4.	带 7 位地址的完整数据传输 .....	936
图 16-5.	第一字节中的 R/S 位 .....	936
图 16-6.	I <sup>2</sup> C 总线位传输过程中的数据有效性 .....	937
图 16-7.	高速数据格式 .....	938
图 16-8.	主机单次传输 .....	942
图 16-9.	主机单次接收 .....	943
图 16-10.	多次重复开始条件下的主机传输 .....	944
图 16-11.	多次重复开始条件下的主机接收 .....	945
图 16-12.	多次重复开始条件下的主机传输之后多次重复开始条件下的主机接收 .....	946
图 16-13.	多次重复开始条件下的主机接收之后多次重复开始条件下的主机传输 .....	947
图 16-14.	高速模式主机传输 .....	948
图 16-15.	从机指令序列 .....	949
图 17-1.	CAN 控制器结构图 .....	980

图 17-2.	CAN 数据帧/远程帧 .....	981
图 17-3.	FIFO 缓冲区中的报文对象 .....	988
图 17-4.	CAN 位时间 .....	991
图 18-1.	USB 模块结构图 .....	1027
图 19-1.	模拟比较器模块的结构图 .....	1136
图 19-2.	比较单元的结构 .....	1138
图 19-3.	比较器内部参考结构 .....	1138
图 20-1.	PWM 模块结构图 .....	1152
图 20-2.	PWM 发生器模块结构图 .....	1153
图 20-3.	PWM 递减计数模式 .....	1156
图 20-4.	PWM 递增/递减计数模式 .....	1156
图 20-5.	递增/递减计数模式产生 PWM 信号的例子 .....	1157
图 20-6.	PWM 死区发生器 .....	1157
图 21-1.	QEI 结构图 .....	1225
图 21-2.	正交编码器和速度预分频器的操作 .....	1227
图 22-1.	144 管脚 LQFP 封装管脚图 .....	1244
图 25-1.	ESD Protection on GPIOs and XOSCn Pins .....	1292
图 25-2.	ESD Protection on Non-Power, Non-GPIO, and Non-XOSCn Pins .....	1292
图 25-3.	Load Conditions .....	1293
图 25-4.	JTAG Test Clock Input Timing .....	1294
图 25-5.	JTAG Test Access Port (TAP) Timing .....	1295
图 25-6.	Power-On and Brown-Out Reset and Voltage Parameters .....	1296
图 25-7.	Brown-Out Reset Timing .....	1296
图 25-8.	External Reset Timing (RST) .....	1297
图 25-9.	Software Reset Timing .....	1297
图 25-10.	Watchdog Reset Timing .....	1297
图 25-11.	MOSC Failure Reset Timing .....	1297
图 25-12.	Hibernation Module Timing .....	1305
图 25-13.	ADC External Reference Filtering .....	1309
图 25-14.	ADC Input Equivalency Diagram .....	1310
图 25-15.	SSI Timing for TI Frame Format (FRF=01), Single Transfer Timing Measurement .....	1311
图 25-16.	SSI Timing for MICROWIRE Frame Format (FRF=10), Single Transfer .....	1311
图 25-17.	SSI Timing for SPI Frame Format (FRF=00), with SPH=1 .....	1311
图 25-18.	I <sup>2</sup> C Timing .....	1312
图 C-1.	Stellaris LM4F232H5QD 144 管脚 LQFP 封装尺寸 .....	1376

## 表格清单

表 1.	修订历史 .....	38
表 2.	文档约定 .....	43
表 1-1.	Stellaris LM4F 器件系列 .....	46
表 1-2.	Stellaris LM4F 系列器件 .....	46
表 1-3.	Stellaris LM4F232H5QD 微控制器特性 .....	48
表 2-1.	处理器模式、特权等级和堆栈使用摘要 .....	73
表 2-2.	处理器 寄存器映射 .....	74
表 2-3.	PSR 寄存器组合 .....	80
表 2-4.	存储器映射 .....	91
表 2-5.	存储器访问行为 .....	94
表 2-6.	SRAM 存储器位带区 .....	96
表 2-7.	外设存储器位带区 .....	96
表 2-8.	异常类型 .....	101
表 2-9.	中断 .....	101
表 2-10.	异常返回行为 .....	108
表 2-11.	故障 .....	109
表 2-12.	故障状态寄存器和故障地址寄存器 .....	110
表 2-13.	Cortex-M4F 指令摘要 .....	112
表 3-1.	内核外设寄存器区域 .....	118
表 3-2.	存储器属性摘要 .....	121
表 3-3.	TEX、S、C 和 B 位域编码 .....	124
表 3-4.	存储器属性编码对应的高速缓存策略 .....	124
表 3-5.	AP 位域编码 .....	124
表 3-6.	Stellaris 微控制器的 <sup>®</sup> 存储器区属性 .....	125
表 3-7.	QNaN 和 SNaN 处理 .....	127
表 3-8.	外设 寄存器映射 .....	128
表 3-9.	中断优先级分组 .....	158
表 3-10.	SIZE 域数值示例 .....	185
表 4-1.	JTAG_SWD_SWO 信号 (144LQFP) .....	193
表 4-2.	上电复位或 $\overline{\text{RST}}$ 置位后的 JTAG 端口管脚状态 .....	194
表 4-3.	JTAG 指令寄存器命令 .....	199
表 5-1.	系统控制及时钟 信号 (144LQFP) .....	203
表 5-2.	复位源 .....	204
表 5-3.	时钟源选项 .....	210
表 5-4.	使用SYSDIV域的可能的系统时钟频率 .....	212
表 5-5.	使用SYSDIV2域的可能的系统时钟频率 .....	212
表 5-6.	当DIV400=1时可能的系统时钟频率示例 .....	213
表 5-7.	系统控制 寄存器映射 .....	218
表 5-8.	替代RCC域的RCC2域 .....	243
表 6-1.	系统异常 寄存器映射 .....	467
表 7-1.	休眠 信号 (144LQFP) .....	476
表 7-2.	TRIM 值为 0x8003 时的计数器行为 .....	480
表 7-3.	TRIM 值为 0x7FFC 时的计数器行为 .....	481
表 7-4.	休眠模块的时钟运行 .....	483
表 7-5.	休眠模块 寄存器映射 .....	485
表 8-1.	Flash 存储器保护策略组合 .....	506

表 8-2.	用户可编程的 Flash 存储器驻留寄存器 .....	509
表 8-3.	Flash 寄存器映射 .....	510
表 9-1.	μDMA 通道分配 .....	536
表 9-2.	所支持的请求类型 .....	537
表 9-3.	控制结构体的存储器映射 .....	538
表 9-4.	通道控制结构体 .....	539
表 9-5.	μDMA 读操作实例：8 位外设 .....	548
表 9-6.	μDMA 中断分配 .....	549
表 9-7.	第 30 号通道的通道控制结构体偏移量 .....	550
表 9-8.	存储器传输示例的通道控制字配置 .....	550
表 9-9.	第 7 号通道的通道控制结构体偏移量 .....	551
表 9-10.	外设传输示例的通道控制字配置 .....	551
表 9-11.	第 8 号通道的主控制结构体及副控制结构体偏移量 .....	553
表 9-12.	外设乒乓接收示例的通道控制字配置 .....	553
表 9-13.	μDMA 寄存器映射 .....	555
表 10-1.	具有非 0 复位值的 GPIO 管脚 .....	598
表 10-2.	GPIO 管脚和复用功能 (144LQFP) .....	598
表 10-3.	GPIO 端口配置示例 .....	605
表 10-4.	GPIO 中断配置示例 .....	605
表 10-5.	具有非 0 复位值的 GPIO 管脚 .....	607
表 10-6.	GPIO 寄存器映射 .....	607
表 10-7.	具有非 0 复位值的 GPIO 管脚 .....	618
表 10-8.	具有非 0 复位值的 GPIO 管脚 .....	624
表 10-9.	具有非 0 复位值的 GPIO 管脚 .....	626
表 10-10.	具有非 0 复位值的 GPIO 管脚 .....	629
表 10-11.	具有非 0 复位值的 GPIO 管脚 .....	636
表 11-1.	可用的 CCP 管脚 .....	654
表 11-2.	通用定时器 信号 (144LQFP) .....	655
表 11-3.	通用定时器功能 .....	657
表 11-4.	单次触发或周期模式下启用定时器时的计数器值 .....	658
表 11-5.	带预分频器的 16 位定时器配置 .....	659
表 11-6.	带预分频器配置的 32 位定时器 (配置为 32/64 位模式) .....	660
表 11-7.	RTC 模式下启用定时器时的计数器值 .....	660
表 11-8.	输入边沿计数模式下启用定时器时的计数器值 .....	662
表 11-9.	输入事件计数模式下启用定时器时的计数器值 .....	663
表 11-10.	PWM 模式下启用定时器时的计数器值 .....	664
表 11-11.	GPTM 模式的超时动作 .....	668
表 11-12.	定时器 寄存器映射 .....	673
表 12-1.	看门狗定时器 (WDT) 寄存器映射 .....	719
表 13-1.	ADC 信号 (144LQFP) .....	744
表 13-2.	采样序列发生器的采样数和 FIFO 深度 .....	745
表 13-3.	差分采样对 .....	752
表 13-4.	ADC 寄存器映射 .....	760
表 14-1.	UART 信号 (144LQFP) .....	828
表 14-2.	流控模式 .....	833
表 14-3.	UART 寄存器映射 .....	838
表 15-1.	SSI 信号 (144LQFP) .....	893
表 15-2.	SSI 寄存器映射 .....	903

表 16-1.	I <sup>2</sup> C 信号 (144LQFP) .....	934
表 16-2.	I <sup>2</sup> C 主机定时器周期与速度模式示例 .....	939
表 16-3.	高速模式下 I <sup>2</sup> C 主机定时器周期的示例 .....	939
表 16-4.	内部集成电路 (I <sup>2</sup> C) 接口 寄存器映射 .....	950
表 16-5.	为 I2CMCS[3:0] 域写入域解码 .....	956
表 17-1.	控制器局域网 信号 (144LQFP) .....	980
表 17-2.	报文对象的配置 .....	985
表 17-3.	CAN 协议范围 .....	991
表 17-4.	CANBIT 寄存器值 .....	992
表 17-5.	CAN 寄存器映射 .....	995
表 18-1.	USB 信号 (144LQFP) .....	1027
表 18-2.	余数 (MAXLOAD/4) .....	1036
表 18-3.	实际读出的字节 .....	1036
表 18-4.	清除 RXRDY 的数据包大小 .....	1037
表 18-5.	通用串行总线 (USB) 控制器 寄存器映射 .....	1038
表 19-1.	模拟比较器 信号 (144LQFP) .....	1137
表 19-2.	内部参考电压和 ACREFLCTL 域值 .....	1139
表 19-3.	模拟比较器参考电压特性, $V_{DDA} = 3.3V$ , EN= 1 且 RNG = 0 .....	1139
表 19-4.	模拟比较器参考电压特性, $V_{DDA} = 3.3V$ , EN= 1 且 RNG = 1 .....	1140
表 19-5.	模拟比较器 寄存器映射 .....	1141
表 20-1.	PWM 信号 (144LQFP) .....	1153
表 20-2.	PWM 寄存器映射 .....	1160
表 21-1.	QEI 信号 (144LQFP) .....	1225
表 21-2.	QEI 寄存器映射 .....	1228
表 23-1.	默认为复用功能的 GPIO 管脚 .....	1245
表 23-2.	按管脚编号分类的信号 .....	1246
表 23-3.	按信号名称分类的信号 .....	1259
表 23-4.	按功能分类的信号 (GPIO 除外) .....	1271
表 23-5.	GPIO 管脚和复用功能 .....	1280
表 23-6.	复用功能的可能的管脚赋值 .....	1284
表 23-7.	未用信号的连接 (144 管脚 LQFP) .....	1288
表 24-1.	温度特性 .....	1290
表 24-2.	热学特性 .....	1290
表 24-3.	ESD 最大额定速率 .....	1290
表 25-1.	Maximum Ratings .....	1291
表 25-2.	Recommended DC Operating Conditions .....	1292
表 25-3.	GPIO Current Restrictions .....	1293
表 25-4.	GPIO Package Side Assignments .....	1293
表 25-5.	JTAG Characteristics .....	1294
表 25-6.	Power Characteristics .....	1295
表 25-7.	Reset Characteristics .....	1296
表 25-8.	LDO Regulator Characteristics .....	1297
表 25-9.	Phase Locked Loop (PLL) Characteristics .....	1298
表 25-10.	Actual PLL Frequency .....	1298
表 25-11.	PIOSC Clock Characteristics .....	1299
表 25-12.	30-kHz Clock Characteristics .....	1299
表 25-13.	HIB Oscillator Input Characteristics .....	1299
表 25-14.	Main Oscillator Input Characteristics .....	1300



表 25-15.	Crystal Parameters .....	1302
表 25-16.	Supported MOSC Crystal Frequencies .....	1302
表 25-17.	System Clock Characteristics with ADC Operation .....	1303
表 25-18.	System Clock Characteristics with USB Operation .....	1303
表 25-19.	Sleep Modes AC Characteristics .....	1304
表 25-20.	Hibernation Module Battery Characteristics .....	1304
表 25-21.	Hibernation Module AC Characteristics .....	1304
表 25-22.	Flash Memory Characteristics .....	1305
表 25-23.	EEPROM Characteristics .....	1305
表 25-24.	GPIO Module Characteristics .....	1306
表 25-25.	ADC Electrical Characteristics .....	1307
表 25-26.	SSI Characteristics .....	1310
表 25-27.	I <sup>2</sup> C Characteristics .....	1312
表 25-28.	Analog Comparator Characteristics .....	1312
表 25-29.	Analog Comparator Voltage Reference Characteristics .....	1313
表 25-30.	Analog Comparator Voltage Reference Characteristics, V <sub>DDA</sub> = 3.3V, EN= 1, and RNG = 0 .....	1313
表 25-31.	Analog Comparator Voltage Reference Characteristics, V <sub>DDA</sub> = 3.3V, EN= 1, and RNG = 1 .....	1313
表 25-32.	Preliminary Current Consumption .....	1314
表 B-1.	器件订购信息 .....	1374

# 寄存器列表

<b>Cortex-M4F 处理器</b> .....	<b>69</b>
寄存器 1: Cortex 通用寄存器 0 ( R0 ) .....	76
寄存器 2: Cortex 通用寄存器 1 ( R1 ) .....	76
寄存器 3: Cortex 通用寄存器 2 ( R2 ) .....	76
寄存器 4: Cortex 通用寄存器 3 ( R3 ) .....	76
寄存器 5: Cortex 通用寄存器 4 ( R4 ) .....	76
寄存器 6: Cortex 通用寄存器 5 ( R5 ) .....	76
寄存器 7: Cortex 通用寄存器 6 ( R6 ) .....	76
寄存器 8: Cortex 通用寄存器 7 ( R7 ) .....	76
寄存器 9: Cortex 通用寄存器 8 ( R8 ) .....	76
寄存器 10: Cortex 通用寄存器 9 ( R9 ) .....	76
寄存器 11: Cortex 通用寄存器 10 ( R10 ) .....	76
寄存器 12: Cortex 通用寄存器 11 ( R11 ) .....	76
寄存器 13: Cortex 通用寄存器 12 ( R12 ) .....	76
寄存器 14: 堆栈指针寄存器 ( SP ) .....	77
寄存器 15: 链接寄存器 ( LR ) .....	78
寄存器 16: 程序计数器寄存器 ( PC ) .....	79
寄存器 17: 程序状态寄存器 ( PSR ) .....	80
寄存器 18: 优先级屏蔽寄存器 ( PRIMASK ) .....	84
寄存器 19: 故障屏蔽寄存器 ( FAULTMASK ) .....	85
寄存器 20: 基本优先级屏蔽寄存器 ( BASEPRI ) .....	86
寄存器 21: 控制寄存器 ( CONTROL ) .....	87
寄存器 22: 浮点状态控制 ( FPSC ) 寄存器 .....	89
<b>Cortex-M4 外设</b> .....	<b>118</b>
寄存器 1: SysTick 控制及状态寄存器 ( STCTRL ) , 偏移量 0x010 .....	132
寄存器 2: SysTick 重载值寄存器 ( STRELOAD ) , 偏移量 0x014 .....	134
寄存器 3: SysTick 当前值寄存器 ( STCURRENT ) , 偏移量 0x018 .....	135
寄存器 4: 中断 0-31 集启用寄存器 ( EN0 ) , 偏移量 0x100 .....	136
寄存器 5: 中断 32-63 集启用寄存器 ( EN1 ) , 偏移量 0x104 .....	136
寄存器 6: 中断 64-95 集启用寄存器 ( EN2 ) , 偏移量 0x108 .....	136
寄存器 7: 中断 96-127 集启用寄存器 ( EN3 ) , 偏移量 0x10C .....	136
寄存器 8: 中断 128-138 设置启用寄存器 ( EN4 ) , 偏移量 0x110 .....	137
寄存器 9: 中断 0-31 清除启用寄存器 ( DIS0 ) , 偏移量 0x180 .....	138
寄存器 10: 中断 32-63 清除启用寄存器 ( DIS1 ) , 偏移量 0x184 .....	138
寄存器 11: 中断 64-95 清除启用寄存器 ( DIS2 ) , 偏移量 0x188 .....	138
寄存器 12: 中断 96-127 清除启用寄存器 ( DIS3 ) , 偏移量 0x18C .....	138
寄存器 13: 中断 128-138 清除启用寄存器 ( DIS4 ) , 偏移量 0x190 .....	139
寄存器 14: 中断 0-31 置位挂起寄存器 ( PEND0 ) , 偏移量 0x200 .....	140
寄存器 15: 中断 32-63 置位挂起寄存器 ( PEND1 ) , 偏移量 0x204 .....	140
寄存器 16: 中断 64-95 置位挂起寄存器 ( PEND2 ) , 偏移量 0x208 .....	140
寄存器 17: 中断 96-127 置位挂起寄存器 ( PEND3 ) , 偏移量 0x20C .....	140
寄存器 18: 中断 128-138 置位挂起寄存器 ( PEND4 ) , 偏移量 0x210 .....	141
寄存器 19: 中断 0-31 清除挂起寄存器 ( UNPEND0 ) , 偏移量 0x280 .....	142
寄存器 20: 中断 32-63 清除挂起寄存器 ( UNPEND1 ) , 偏移量 0x284 .....	142
寄存器 21: 中断 64-95 清除挂起寄存器 ( UNPEND2 ) , 偏移量 0x288 .....	142

寄存器 22:	中断 96-127 清除挂起寄存器 (UNPEND3), 偏移量 0x28C	142
寄存器 23:	中断 128-138 清除挂起寄存器 (UNPEND4), 偏移量 0x290	143
寄存器 24:	中断 0-31 活动位寄存器 (ACTIVE0), 偏移量 0x300	144
寄存器 25:	中断 32-63 活动位寄存器 (ACTIVE1), 偏移量 0x304	144
寄存器 26:	中断 64-95 活动位寄存器 (ACTIVE2), 偏移量 0x308	144
寄存器 27:	中断 96-127 活动位寄存器 (ACTIVE3), 偏移量 0x30C	144
寄存器 28:	中断 128-138 活动位寄存器 (ACTIVE4), 偏移量 0x310	145
寄存器 29:	中断 0-3 优先级寄存器 (PRI0), 偏移量 0x400	146
寄存器 30:	中断 4-7 优先级寄存器 (PRI1), 偏移量 0x404	146
寄存器 31:	中断 8-11 优先级寄存器 (PRI2), 偏移量 0x408	146
寄存器 32:	中断 12-15 优先级寄存器 (PRI3), 偏移量 0x40C	146
寄存器 33:	中断 16-19 优先级寄存器 (PRI4), 偏移量 0x410	146
寄存器 34:	中断 20-23 优先级寄存器 (PRI5), 偏移量 0x414	146
寄存器 35:	中断 24-27 优先级寄存器 (PRI6), 偏移量 0x418	146
寄存器 36:	中断 28-31 优先级寄存器 (PRI7), 偏移量 0x41C	146
寄存器 37:	中断 32-35 优先级寄存器 (PRI8), 偏移量 0x420	146
寄存器 38:	中断 36-39 优先级寄存器 (PRI9), 偏移量 0x424	146
寄存器 39:	中断 40-43 优先级寄存器 (PRI10), 偏移量 0x428	146
寄存器 40:	中断 44-47 优先级寄存器 (PRI11), 偏移量 0x42C	146
寄存器 41:	中断 48-51 优先级寄存器 (PRI12), 偏移量 0x430	146
寄存器 42:	中断 52-55 优先级寄存器 (PRI13), 偏移量 0x434	146
寄存器 43:	中断 56-59 优先级寄存器 (PRI14), 偏移量 0x438	146
寄存器 44:	中断 60-63 优先级寄存器 (PRI15), 偏移量 0x43C	146
寄存器 45:	中断 64-67 优先级寄存器 (PRI16), 偏移量 0x440	148
寄存器 46:	中断 68-71 优先级寄存器 (PRI17), 偏移量 0x444	148
寄存器 47:	中断 72-75 优先级寄存器 (PRI18), 偏移量 0x448	148
寄存器 48:	中断 76-79 优先级寄存器 (PRI19), 偏移量 0x44C	148
寄存器 49:	中断 80-83 优先级寄存器 (PRI20), 偏移量 0x450	148
寄存器 50:	中断 84-87 优先级寄存器 (PRI21), 偏移量 0x454	148
寄存器 51:	中断 88-91 优先级寄存器 (PRI22), 偏移量 0x458	148
寄存器 52:	中断 92-95 优先级寄存器 (PRI23), 偏移量 0x45C	148
寄存器 53:	中断 96-99 优先级寄存器 (PRI24), 偏移量 0x460	148
寄存器 54:	中断 100-103 优先级寄存器 (PRI25), 偏移量 0x464	148
寄存器 55:	中断 104-107 优先级寄存器 (PRI26), 偏移量 0x468	148
寄存器 56:	中断 108-111 优先级寄存器 (PRI27), 偏移量 0x46C	148
寄存器 57:	中断 112-115 优先级寄存器 (PRI28), 偏移量 0x470	148
寄存器 58:	中断 116-119 优先级寄存器 (PRI29), 偏移量 0x474	148
寄存器 59:	中断 120-123 优先级寄存器 (PRI30), 偏移量 0x478	148
寄存器 60:	中断 124-127 优先级寄存器 (PRI31), 偏移量 0x47C	148
寄存器 61:	中断 128-131 优先级寄存器 (PRI32), 偏移量 0x480	148
寄存器 62:	中断 132-135 优先级寄存器 (PRI33), 偏移量 0x484	148
寄存器 63:	中断 136-138 优先级寄存器 (PRI34), 偏移量 0x488	148
寄存器 64:	软件触发中断寄存器 (SWTRIG), 偏移量 0xF00	150
寄存器 65:	辅助控制寄存器 (ACTLR), 偏移量 0x008	151
寄存器 66:	CPU ID 基础寄存器 (CPUID), 偏移量 0xD00	153
寄存器 67:	中断控制及状态寄存器 (INTCTRL), 偏移量 0xD04	154
寄存器 68:	向量表偏移寄存器 (VTABLE), 偏移量 0xD08	157
寄存器 69:	应用程序中断及复位控制寄存器 (APINT), 偏移量 0xD0C	158

寄存器 70:	系统控制寄存器 ( SYSCTRL ) , 偏移量 0xD10 .....	160
寄存器 71:	配置及控制寄存器 ( CFGCTRL ) , 偏移量 0xD14 .....	162
寄存器 72:	系统处理程序优先级寄存器 1 ( SYSPRI1 ) , 偏移量 0xD18 .....	164
寄存器 73:	系统处理程序优先级寄存器 2 ( SYSPRI2 ) , 偏移量 0xD1C .....	165
寄存器 74:	系统处理程序优先级寄存器 3 ( SYSPRI3 ) , 偏移量 0xD20 .....	166
寄存器 75:	系统处理程序控制及状态寄存器 ( SYSHNDCTRL ) , 偏移量 0xD24 .....	167
寄存器 76:	可配置故障状态寄存器 ( FAULTSTAT ) , 偏移量 0xD28 .....	170
寄存器 77:	硬故障状态寄存器 ( HFAULTSTAT ) , 偏移量 0xD2C .....	176
寄存器 78:	存储器管理故障地址寄存器 ( MMADDR ) , 偏移量 0xD34 .....	177
寄存器 79:	总线故障地址寄存器 ( FAULTADDR ) , 偏移量 0xD38 .....	178
寄存器 80:	MPU 类型寄存器 ( MPUATYPE ) , 偏移量 0xD90 .....	179
寄存器 81:	MPU 控制寄存器 ( MPUCTRL ) , 偏移量 0xD94 .....	180
寄存器 82:	MPU 区编号寄存器 ( MPUNUMBER ) , 偏移量 0xD98 .....	182
寄存器 83:	MPU 区基地址寄存器 ( MPUBASE ) , 偏移量 0xD9C .....	183
寄存器 84:	MPU 区基地址别名寄存器 1 ( MPUBASE1 ) , 偏移量 0xDA4 .....	183
寄存器 85:	MPU 区基地址别名寄存器 2 ( MPUBASE2 ) , 偏移量 0xDAC .....	183
寄存器 86:	MPU 区基地址别名寄存器 3 ( MPUBASE3 ) , 偏移量 0xDB4 .....	183
寄存器 87:	MPU 区属性和大小寄存器 ( MPUATTR ) , 偏移量 0xDA0 .....	185
寄存器 88:	MPU 区属性和大小别名寄存器 1 ( MPUATTR1 ) , 偏移量 0xDA8 .....	185
寄存器 89:	MPU 区属性和大小别名寄存器 2 ( MPUATTR2 ) , 偏移量 0xDB0 .....	185
寄存器 90:	MPU 区属性和大小别名寄存器 3 ( MPUATTR3 ) , 偏移量 0xDB8 .....	185
寄存器 91:	协处理器访问控制 ( CPAC ) , 偏移量 0xD88 .....	187
寄存器 92:	浮点上下文控制 ( FPCC ) , 偏移量 0xF34 .....	188
寄存器 93:	浮点上下文访问 ( FPCA ) , 偏移量 0xF38 .....	190
寄存器 94:	浮点默认状态控制 ( FPDSC ) , 偏移量 0xF3C .....	191
<b>系统控制</b>	<b>.....</b>	<b>203</b>
寄存器 1:	器件标识寄存器 0 ( DID0 ) , 偏移量 0x000 .....	223
寄存器 2:	器件标识寄存器 1 ( DID1 ) , 偏移量 0x004 .....	225
寄存器 3:	掉电复位控制寄存器 ( PBORCTL ) , 偏移量 0x030 .....	227
寄存器 4:	原始中断状态寄存器 ( RIS ) , 偏移量 0x050 .....	228
寄存器 5:	中断屏蔽控制寄存器 ( IMC ) , 偏移量 0x054 .....	230
寄存器 6:	屏蔽中断状态和清除寄存器 ( MISC ) , 偏移量 0x058 .....	232
寄存器 7:	复位原因寄存器 ( RESC ) , 偏移量 0x05C .....	234
寄存器 8:	运行模式时钟配置寄存器 ( RCC ) , 偏移量 0x060 .....	236
寄存器 9:	GPIO 高性能总线控制寄存器 ( GPIOHBCTL ) , 偏移量 0x06C .....	240
寄存器 10:	运行模式时钟配置寄存器 2 ( RCC2 ) , 偏移量 0x070 .....	243
寄存器 11:	主振荡器控制寄存器 ( MOSCCTL ) , 偏移量 0x07C .....	246
寄存器 12:	深度睡眠时钟配置寄存器 ( DSLPCLKCFG ) , 偏移量 0x144 .....	247
寄存器 13:	系统属性寄存器 ( SYSPROP ) , 偏移量 0x14C .....	249
寄存器 14:	精确内部振荡器校准寄存器 ( PIOSCCAL ) , 偏移量 0x150 .....	250
寄存器 15:	精确内部振荡器统计寄存器 ( PIOSCSTAT ) , 偏移量 0x154 .....	251
寄存器 16:	PLL 频率寄存器 0 ( PLLFREQ0 ) , 偏移量 0x160 .....	252
寄存器 17:	PLL 频率寄存器 1 ( PLLFREQ1 ) , 偏移量 0x164 .....	253
寄存器 18:	PLL 状态寄存器 ( PLLSTAT ) , 偏移量 0x168 .....	254
寄存器 19:	看门狗定时器外设存在寄存器 ( PPWD ) , 偏移量 0x300 .....	255
寄存器 20:	16/32 位通用定时器外设存在寄存器 ( PPTIMER ) , 偏移量 0x304 .....	256
寄存器 21:	通用输入/输出外设存在寄存器 ( PPGPIO ) , 偏移量 0x308 .....	258
寄存器 22:	微型直接存储器访问外设存在寄存器 ( PPDMA ) , 偏移量 0x30C .....	261

寄存器 23:	休眠外设存在寄存器 ( PPHIB ) , 偏移量 0x314 .....	262
寄存器 24:	通用异步收发器外设存在寄存器 ( PPUART ) , 偏移量 0x318 .....	263
寄存器 25:	同步串行接口外设存在寄存器 ( PPSSI ) , 偏移量 0x31C .....	265
寄存器 26:	内部集成电路外设存在寄存器 ( PPI2C ) , 偏移量 0x320 .....	266
寄存器 27:	通用串行总线外设存在寄存器 ( PPUSB ) , 偏移量 0x328 .....	268
寄存器 28:	控制器局域网外设存在寄存器 ( PPCAN ) , 偏移量 0x334 .....	269
寄存器 29:	模数转换器外设存在寄存器 ( PPADC ) , 偏移量 0x338 .....	270
寄存器 30:	模拟比较器外设存在寄存器 ( PPACMP ) , 偏移量 0x33C .....	271
寄存器 31:	脉宽调制器外设存在寄存器 ( PPPWM ) , 偏移量 0x340 .....	272
寄存器 32:	正交编码器接口外设存在寄存器 ( PPQEI ) , 偏移量 0x344 .....	273
寄存器 33:	EEPROM 外设存在寄存器 ( PPEEPROM ) , 偏移量 0x358 .....	274
寄存器 34:	32/64 位宽通用定时器外设存在寄存器 ( PPWTIMER ) , 偏移量 0x35C .....	275
寄存器 35:	看门狗定时器软件复位寄存器 ( SRWD ) , 偏移量 0x500 .....	277
寄存器 36:	16/32 位通用定时器软件复位寄存器 ( SRTIMER ) , 偏移量 0x504 .....	278
寄存器 37:	通用输入/输出软件复位寄存器 ( SRGPIO ) , 偏移量 0x508 .....	280
寄存器 38:	微型直接存储器访问软件复位寄存器 ( SRDMA ) , 偏移量 0x50C .....	283
寄存器 39:	休眠软件复位寄存器 ( SRHIB ) , 偏移量 0x514 .....	284
寄存器 40:	通用异步收发器软件复位寄存器 ( SRUART ) , 偏移量 0x518 .....	285
寄存器 41:	同步串行接口软件复位寄存器 ( SRSSI ) , 偏移量 0x51C .....	287
寄存器 42:	内部集成电路软件复位寄存器 ( SRI2C ) , 偏移量 0x520 .....	289
寄存器 43:	通用串行总线软件复位寄存器 ( SRUSB ) , 偏移量 0x528 .....	291
寄存器 44:	控制器局域网软件复位寄存器 ( SRCAN ) , 偏移量 0x534 .....	292
寄存器 45:	模数转换器软件复位寄存器 ( SRADC ) , 偏移量 0x538 .....	293
寄存器 46:	模数比较器软件复位寄存器 ( SRACMP ) , 偏移量 0x53C .....	294
寄存器 47:	脉宽调制器软件复位寄存器 ( SRPWM ) , 偏移量 0x540 .....	295
寄存器 48:	正交编码器接口软件复位寄存器 ( SRQEI ) , 偏移量 0x544 .....	296
寄存器 49:	32/64 位宽通用定时器软件复位寄存器 ( SRWTIMER ) , 偏移量 0x55C .....	297
寄存器 50:	看门狗定时器运行模式时钟门控控制寄存器 ( RCGCWD ) , 偏移量 0x600 .....	299
寄存器 51:	16/32 位通用定时器运行模式时钟门控控制寄存器 ( RCGCTIMER ) , 偏移量 0x604 .....	300
寄存器 52:	通用输入/输出运行模式时钟门控控制寄存器 ( RCGCGPIO ) , 偏移量 0x608 .....	302
寄存器 53:	微型直接存储器访问运行模式时钟门控控制寄存器 ( RCGCDMA ) , 偏移量 0x60C .....	305
寄存器 54:	休眠运行模式时钟门控控制寄存器 ( RCGCHIB ) , 偏移量 0x614 .....	306
寄存器 55:	通用异步收发器运行模式时钟门控控制寄存器 ( RCGCUART ) , 偏移量 0x618 .....	307
寄存器 56:	同步串行接口运行模式时钟门控控制寄存器 ( RCGCSSI ) , 偏移量 0x61C .....	309
寄存器 57:	内部集成电路运行模式时钟门控控制寄存器 ( RCGCI2C ) , 偏移量 0x620 .....	311
寄存器 58:	通用串行总线运行模式时钟门控控制寄存器 ( RCGCUSB ) , 偏移量 0x628 .....	313
寄存器 59:	控制器局域网运行模式时钟门控控制寄存器 ( RCGCCAN ) , 偏移量 0x634 .....	314
寄存器 60:	模数转换器运行模式时钟门控控制寄存器 ( RCGCADC ) , 偏移量 0x638 .....	315
寄存器 61:	模拟比较器运行模式时钟门控控制寄存器 ( RCGCACMP ) , 偏移量 0x63C .....	316
寄存器 62:	脉宽调制器运行模式时钟门控控制寄存器 ( RCGCPWM ) , 偏移量 0x640 .....	317
寄存器 63:	正交编码器接口运行模式时钟门控控制寄存器 ( RCGCQEI ) , 偏移量 0x644 .....	318
寄存器 64:	32/64 位宽通用定时器运行模式时钟门控控制寄存器 ( RCGCWTIMER ) , 偏移量 0x65C .....	319
寄存器 65:	看门狗定时器睡眠模式时钟门控控制寄存器 ( SCGCWD ) , 偏移量 0x700 .....	321
寄存器 66:	16/32 位通用定时器睡眠模式时钟门控控制寄存器 ( SCGCTIMER ) , 偏移量 0x704 .....	322
寄存器 67:	通用输入/输出睡眠模式时钟门控控制寄存器 ( SCGCGPIO ) , 偏移量 0x708 .....	324
寄存器 68:	微型直接存储器访问睡眠模式时钟门控控制寄存器 ( SCGCDMA ) , 偏移量 0x70C .....	327
寄存器 69:	休眠睡眠模式时钟门控控制寄存器 ( SCGCHIB ) , 偏移量 0x714 .....	328

寄存器 70:	通用异步收发器睡眠模式时钟门控控制寄存器 ( SCGCUART ) , 偏移量 0x718 .....	329
寄存器 71:	同步串行接口睡眠模式时钟门控控制寄存器 ( SCGCSSI ) , 偏移量 0x71C .....	331
寄存器 72:	内部集成电路睡眠模式时钟门控控制寄存器 ( SCGCI2C ) , 偏移量 0x720 .....	333
寄存器 73:	通用串行总线睡眠模式时钟门控控制寄存器 ( SCGCUSB ) , 偏移量 0x728 .....	335
寄存器 74:	控制器局域网睡眠模式时钟门控控制寄存器 ( SCGCCAN ) , 偏移量 0x734 .....	336
寄存器 75:	模数转换器睡眠模式时钟门控控制寄存器 ( SCGCADC ) , 偏移量 0x738 .....	337
寄存器 76:	模拟比较器睡眠模式时钟门控控制寄存器 ( SCGCACMP ) , 偏移量 0x73C .....	338
寄存器 77:	脉宽调制器睡眠模式时钟门控控制寄存器 ( SCGCPWM ) , 偏移量 0x740 .....	339
寄存器 78:	正交编码器接口睡眠模式时钟门控控制寄存器 ( SCGCQEI ) , 偏移量 0x744 .....	340
寄存器 79:	32/64 位宽通用定时器睡眠模式时钟门控控制寄存器 ( SCGCWTIMER ) , 偏移量 0x75C .....	341
寄存器 80:	看门狗定时器深度睡眠模式时钟门控控制寄存器 ( DCGCWD ) , 偏移量 0x800 .....	343
寄存器 81:	16/32 位通用定时器深度睡眠模式时钟门控控制寄存器 ( DCGCTIMER ) , 偏移量 0x804 .....	344
寄存器 82:	通用输入/输出深度睡眠模式时钟门控控制寄存器 ( DCGCGPIO ) , 偏移量 0x808 .....	346
寄存器 83:	微型直接存储器访问深度睡眠模式时钟门控控制寄存器 ( DCGCDMA ) , 偏移量 0x80C .....	349
寄存器 84:	休眠深度睡眠模式时钟门控控制寄存器 ( DCGCHIB ) , 偏移量 0x814 .....	350
寄存器 85:	通用异步收发器深度睡眠模式时钟门控控制寄存器 ( DCGCUART ) , 偏移量 0x818 .....	351
寄存器 86:	同步串行接口深度睡眠模式时钟门控控制寄存器 ( DCGCSSI ) , 偏移量 0x81C .....	353
寄存器 87:	内部集成电路深度睡眠模式时钟门控控制寄存器 ( DCGCI2C ) , 偏移量 0x820 .....	355
寄存器 88:	通用串行总线深度睡眠模式时钟门控控制寄存器 ( DCGCUSB ) , 偏移量 0x828 .....	357
寄存器 89:	控制器局域网深度睡眠模式时钟门控控制寄存器 ( DGCCAN ) , 偏移量 0x834 .....	358
寄存器 90:	模数转换器深度睡眠模式时钟门控控制寄存器 ( DCGCADC ) , 偏移量 0x838 .....	359
寄存器 91:	模拟比较器深度睡眠模式时钟门控控制寄存器 ( DCGCACMP ) , 偏移量 0x83C .....	360
寄存器 92:	脉宽调制器深度睡眠模式时钟门控控制寄存器 ( DCGCPWM ) , 偏移量 0x840 .....	361
寄存器 93:	正交编码器接口深度睡眠模式时钟门控控制寄存器 ( DCGCQEI ) , 偏移量 0x844 .....	362
寄存器 94:	32/64 位宽通用定时器深度睡眠模式时钟门控控制寄存器 ( DCGCWTIMER ) , 偏移量 0x85C .....	363
寄存器 95:	看门狗定时器功率控制寄存器 ( PCWD ) , 偏移量 0x900 .....	365
寄存器 96:	16/32 位通用定时器功率控制寄存器 ( PCTIMER ) , 偏移量 0x904 .....	366
寄存器 97:	通用输入/输出功率控制寄存器 ( PCGPIO ) , 偏移量 0x908 .....	369
寄存器 98:	微型直接存储器访问功率控制寄存器 ( PCDMA ) , 偏移量 0x90C .....	373
寄存器 99:	休眠功率控制寄存器 ( PCHIB ) , 偏移量 0x914 .....	374
寄存器 100:	通用异步收发器功率控制寄存器 ( PCUART ) , 偏移量 0x918 .....	375
寄存器 101:	同步串行接口功率控制寄存器 ( PCSSI ) , 偏移量 0x91C .....	378
寄存器 102:	内部集成电路功率控制寄存器 ( PCI2C ) , 偏移量 0x920 .....	380
寄存器 103:	通用串行总线功率控制寄存器 ( PCUSB ) , 偏移量 0x928 .....	382
寄存器 104:	控制器局域网功率控制寄存器 ( PCCAN ) , 偏移量 0x934 .....	383
寄存器 105:	模数转换器功率控制寄存器 ( PCADC ) , 偏移量 0x938 .....	384
寄存器 106:	模拟比较器功率控制寄存器 ( PCACMP ) , 偏移量 0x93C .....	385
寄存器 107:	脉宽调制器功率控制寄存器 ( PCPWM ) , 偏移量 0x940 .....	386
寄存器 108:	正交编码器接口功率控制寄存器 ( PCQEI ) , 偏移量 0x944 .....	387
寄存器 109:	32/64 位宽通用定时器功率控制寄存器 ( PCWTIMER ) , 偏移量 0x95C .....	388
寄存器 110:	看门狗定时器外设就绪寄存器 ( PRWD ) , 偏移量 0xA00 .....	391
寄存器 111:	16/32 位通用定时器外设就绪寄存器 ( PRTIMER ) , 偏移量 0xA04 .....	392
寄存器 112:	通用输入/输出外设就绪寄存器 ( PRGPIO ) , 偏移量 0xA08 .....	394
寄存器 113:	微型直接存储器访问外设就绪寄存器 ( PRDMA ) , 偏移量 0xA0C .....	397
寄存器 114:	休眠外设就绪寄存器 ( PRHIB ) , 偏移量 0xA14 .....	398
寄存器 115:	通用异步收发器外设就绪寄存器 ( PRUART ) , 偏移量 0xA18 .....	399

寄存器 116:	同步串行接口外设就绪寄存器 ( PRSSI ) , 偏移量 0xA1C .....	401
寄存器 117:	内部集成电路外设就绪寄存器 ( PRI2C ) , 偏移量 0xA20 .....	403
寄存器 118:	通用串行总线外设就绪寄存器 ( PRUSB ) , 偏移量 0xA28 .....	405
寄存器 119:	控制器局域网外设就绪寄存器 ( PRCAN ) , 偏移量 0xA34 .....	406
寄存器 120:	模数转换器外设就绪寄存器 ( PRADC ) , 偏移量 0xA38 .....	407
寄存器 121:	模拟比较器外设就绪寄存器 ( PRACMP ) , 偏移量 0xA3C .....	408
寄存器 122:	脉宽调制器外设就绪寄存器 ( PRPWM ) , 偏移量 0xA40 .....	409
寄存器 123:	正交编码器接口外设就绪寄存器 ( PRQEI ) , 偏移量 0xA44 .....	410
寄存器 124:	32/64 位宽通用定时器外设就绪寄存器 ( PRWTIMER ) , 偏移量 0xA5C .....	411
寄存器 125:	器件功能寄存器 0 ( DC0 ) , 偏移量 0x008 .....	413
寄存器 126:	器件功能寄存器 1 ( DC1 ) , 偏移量 0x010 .....	415
寄存器 127:	器件功能寄存器 2 ( DC2 ) , 偏移量 0x014 .....	418
寄存器 128:	器件功能寄存器 3 ( DC3 ) , 偏移量 0x018 .....	420
寄存器 129:	器件功能寄存器 4 ( DC4 ) , 偏移量 0x01C .....	424
寄存器 130:	器件功能寄存器 5 ( DC5 ) , 偏移量 0x020 .....	426
寄存器 131:	器件功能寄存器 6 ( DC6 ) , 偏移量 0x024 .....	428
寄存器 132:	器件功能寄存器 7 ( DC7 ) , 偏移量 0x028 .....	429
寄存器 133:	器件功能寄存器 8 ( DC8 ) , 偏移量 0x02C .....	432
寄存器 134:	软件复位控制寄存器 0 ( SRCR0 ) , 偏移量 0x040 .....	435
寄存器 135:	软件复位控制寄存器 1 ( SRCR1 ) , 偏移量 0x044 .....	437
寄存器 136:	软件复位控制寄存器 2 ( SRCR2 ) , 偏移量 0x048 .....	440
寄存器 137:	运行模式时钟门控控制寄存器 0 ( RCGC0 ) , 偏移量 0x100 .....	442
寄存器 138:	运行模式时钟门控控制寄存器 1 ( RCGC1 ) , 偏移量 0x104 .....	445
寄存器 139:	运行模式时钟门控控制寄存器 2 ( RCGC2 ) , 偏移量 0x108 .....	448
寄存器 140:	睡眠模式时钟门控控制寄存器 0 ( SCGC0 ) , 偏移量 0x110 .....	450
寄存器 141:	睡眠模式时钟门控控制寄存器 1 ( SCGC1 ) , 偏移量 0x114 .....	452
寄存器 142:	睡眠模式时钟门控控制寄存器 2 ( SCGC2 ) , 偏移量 0x118 .....	455
寄存器 143:	深度睡眠模式时钟门控控制寄存器 0 ( DCGC0 ) , 偏移量 0x120 .....	457
寄存器 144:	深度睡眠模式时钟门控控制寄存器 1 ( DCGC1 ) , 偏移量 0x124 .....	459
寄存器 145:	深度睡眠模式时钟门控控制寄存器 2 ( DCGC2 ) , 偏移量 0x128 .....	462
寄存器 146:	器件功能寄存器 9 ( DC9 ) , 偏移量 0x190 .....	464
寄存器 147:	非易失性存储器信息寄存器 ( NVMSTAT ) , 偏移量 0x1A0 .....	466
<b>系统异常模块 .....</b>	<b>467</b>	
寄存器 1:	系统异常原始中断状态 ( SYSEXCRIS ) , 偏移量 0x000 .....	468
寄存器 2:	系统异常中断屏蔽 ( SYSEXCIM ) , 偏移量 0x004 .....	470
寄存器 3:	系统异常屏蔽的中断状态 ( SYSEXC MIS ) , 偏移量 0x008 .....	472
寄存器 4:	系统异常中断清零 ( SYSEXCIC ) , 偏移量 0x00C .....	474
<b>休眠模块 .....</b>	<b>475</b>	
寄存器 1:	休眠 RTC 计数器寄存器 ( HIBRTCC ) , 偏移量 0x000 .....	486
寄存器 2:	休眠 RTC 匹配寄存器 0 ( HIBRTCM0 ) , 偏移量 0x004 .....	487
寄存器 3:	休眠 RTC 加载寄存器 ( HIBRTCLD ) , 偏移量 0x00C .....	488
寄存器 4:	休眠控制寄存器 ( HIBCTL ) , 偏移量 0x010 .....	489
寄存器 5:	休眠中断屏蔽寄存器 ( HIBIM ) , 偏移量 0x014 .....	493
寄存器 6:	休眠原始中断状态寄存器 ( HIBRIS ) , 偏移量 0x018 .....	495
寄存器 7:	休眠屏蔽中断状态寄存器 ( HIBMIS ) , 偏移量 0x01C .....	497
寄存器 8:	休眠中断清除寄存器 ( HIBIC ) , 偏移量 0x020 .....	499
寄存器 9:	休眠 RTC 修正寄存器 ( HIBRTCT ) , 偏移量 0x024 .....	500
寄存器 10:	休眠 RTC 亚秒寄存器 ( HIBRTCSS ) , 偏移量 0x028 .....	501

寄存器 11:	休眠数据寄存器 ( HIBDATA ) , 偏移量 0x030-0x06F .....	502
<b>内部存储器</b>	<b>.....</b>	<b>503</b>
寄存器 1:	Flash 存储器地址寄存器 ( FMA ) , 偏移量 0x000 .....	512
寄存器 2:	Flash 存储器数据寄存器 ( FMD ) , 偏移量 0x004 .....	513
寄存器 3:	Flash 存储器控制寄存器 ( FMC ) , 偏移量 0x008 .....	514
寄存器 4:	Flash 控制器原始中断状态寄存器 ( FCRIS ) , 偏移量 0x00C .....	516
寄存器 5:	Flash 控制器中断屏蔽寄存器 ( FCIM ) , 偏移量 0x010 .....	518
寄存器 6:	Flash 控制器屏蔽中断状态和清除寄存器 ( FCMISC ) , 偏移量 0x014 .....	520
寄存器 7:	Flash 存储器控制寄存器 2 ( FMC2 ) , 偏移量 0x020 .....	522
寄存器 8:	Flash 写缓冲器有效寄存器 ( FWBVAL ) , 偏移量 0x030 .....	523
寄存器 9:	Flash 写入缓冲区寄存器 n ( FWBn ) , 偏移量 0x100 - 0x17C .....	524
寄存器 10:	Flash 容量寄存器 ( FSIZE ) , 偏移量 0xFC0 .....	525
寄存器 11:	SRAM 大小寄存器 ( SSIZE ) , 偏移量 0xFC4 .....	526
寄存器 12:	ROM 软件映射寄存器 ( ROMSWMAP ) , 偏移量 0xFCC .....	527
寄存器 13:	ROM 控制寄存器 ( RMCTL ) , 偏移量 0xF0 .....	528
寄存器 14:	Flash 存储器保护读取启用寄存器 0 ( FMPRE0 ) , 偏移量 0x130 和 0x200 .....	529
寄存器 15:	Flash 存储器保护读取启用寄存器 1 ( FMPRE1 ) , 偏移量 0x204 .....	529
寄存器 16:	Flash 存储器保护读取启用寄存器 2 ( FMPRE2 ) , 偏移量 0x208 .....	529
寄存器 17:	Flash 存储器保护读取启用寄存器 3 ( FMPRE3 ) , 偏移量 0x20C .....	529
寄存器 18:	Flash 存储器保护编程启用寄存器 0 ( FMPPE0 ) , 偏移量 0x134 和 0x400 .....	530
寄存器 19:	Flash 存储器保护编程启用寄存器 1 ( FMPPE1 ) , 偏移量 0x404 .....	530
寄存器 20:	Flash 存储器保护编程启用寄存器 2 ( FMPPE2 ) , 偏移量 0x408 .....	530
寄存器 21:	Flash 存储器保护编程启用寄存器 3 ( FMPPE3 ) , 偏移量 0x40C .....	530
寄存器 22:	启动配置寄存器 ( BOOTCFG ) , 偏移量 0x1D0 .....	531
寄存器 23:	用户寄存器 0 ( USER_REG0 ) , 偏移量 0x1E0 .....	533
寄存器 24:	用户寄存器 1 ( USER_REG1 ) , 偏移量 0x1E4 .....	533
寄存器 25:	用户寄存器 2 ( USER_REG2 ) , 偏移量 0x1E8 .....	533
寄存器 26:	用户寄存器 3 ( USER_REG3 ) , 偏移量 0x1EC .....	533
<b>微型直接存储器访问 ( <math>\mu</math>DMA )</b>	<b>.....</b>	<b>534</b>
寄存器 1:	DMA 通道源地址末指针寄存器 ( DMASRCENDP ) , 偏移量 0x000 .....	557
寄存器 2:	DMA 通道目的地址末指针寄存器 ( DMADSTENDP ) , 偏移量 0x004 .....	558
寄存器 3:	DMA 通道控制字寄存器 ( DMACHCTL ) , 偏移量 0x008 .....	559
寄存器 4:	DMA 状态寄存器 ( DMASTAT ) , 偏移量 0x000 .....	564
寄存器 5:	DMA 配置寄存器 ( DMACFG ) , 偏移量 0x004 .....	566
寄存器 6:	DMA 通道控制基指针寄存器 ( DMACTLBASE ) , 偏移量 0x008 .....	567
寄存器 7:	DMA 副通道控制基指针寄存器 ( DMAALTBASE ) , 偏移量 0x00C .....	568
寄存器 8:	DMA 通道等待请求状态寄存器 ( DMAWAITSTAT ) , 偏移量 0x010 .....	569
寄存器 9:	DMA 通道软件请求寄存器 ( DMASWREQ ) , 偏移量 0x014 .....	570
寄存器 10:	DMA 通道采用猝发置位寄存器 ( DMAUSEBURSTSET ) , 偏移量 0x018 .....	571
寄存器 11:	DMA 通道采用猝发清除寄存器 ( DMAUSEBURSTCLR ) , 偏移量 0x01C .....	572
寄存器 12:	DMA 通道请求屏蔽置位寄存器 ( DMAREQMASKSET ) , 偏移量 0x020 .....	573
寄存器 13:	DMA 通道请求屏蔽清零寄存器 ( DMAREQMASKCLR ) , 偏移量 0x024 .....	574
寄存器 14:	DMA 通道启用置位寄存器 ( DMAENASET ) , 偏移量 0x028 .....	575
寄存器 15:	DMA 通道启用清除寄存器 ( DMAENACLAR ) , 偏移量 0x02C .....	576
寄存器 16:	DMA 通道主副置位寄存器 ( DMAALTSET ) , 偏移量 0x030 .....	577
寄存器 17:	DMA 通道主副清除寄存器 ( DMAALTCLR ) , 偏移量 0x034 .....	578
寄存器 18:	DMA 通道优先置位寄存器 ( DMAPRIOSET ) , 偏移量 0x038 .....	579
寄存器 19:	DMA 通道优先清零寄存器 ( DMAPRIOCLR ) , 偏移量 0x03C .....	580



寄存器 20:	DMA 总线错误清除寄存器 (DMAERRCLR), 偏移量 0x04C	581
寄存器 21:	DMA 通道分配寄存器 (DMACHASGN), 偏移量 0x500	582
寄存器 22:	DMA 通道中断状态寄存器 (DMACHIS), 偏移量 0x504	583
寄存器 23:	DMA 通道映射选择寄存器 0 (DMACHMAP0), 偏移量 0x510	584
寄存器 24:	DMA 通道映射选择寄存器 1 (DMACHMAP1), 偏移量 0x514	585
寄存器 25:	DMA 通道映射选择寄存器 2 (DMACHMAP2), 偏移量 0x518	586
寄存器 26:	DMA 通道映射选择寄存器 3 (DMACHMAP3), 偏移量 0x51C	587
寄存器 27:	DMA 外设标识寄存器 0 (DMAPeriphID0), 偏移量 0xFE0	588
寄存器 28:	DMA 外设标识寄存器 1 (DMAPeriphID1), 偏移量 0xFE4	589
寄存器 29:	DMA 外设标识寄存器 2 (DMAPeriphID2), 偏移量 0xFE8	590
寄存器 30:	DMA 外设标识寄存器 3 (DMAPeriphID3), 偏移量 0xFEC	591
寄存器 31:	DMA 外设标识寄存器 4 (DMAPeriphID4), 偏移量 0xFD0	592
寄存器 32:	DMA PrimeCell 标识寄存器 0 (DMAPCellID0), 偏移量 0xFF0	593
寄存器 33:	DMA PrimeCell 标识寄存器 1 (DMAPCellID1), 偏移量 0xFF4	594
寄存器 34:	DMA PrimeCell 标识寄存器 2 (DMAPCellID2), 偏移量 0xFF8	595
寄存器 35:	DMA PrimeCell 标识寄存器 3 (DMAPCellID3), 偏移量 0xFFC	596
<b>通用输入/输出端口 (GPIO)</b>		<b>597</b>
寄存器 1:	GPIO 数据寄存器 (GPIODATA), 偏移量 0x000	609
寄存器 2:	GPIO 方向寄存器 (GPIODIR), 偏移量 0x400	610
寄存器 3:	GPIO 中断检测寄存器 (GPIOIS), 偏移量 0x404	611
寄存器 4:	GPIO 中断双边沿 (GPIOIBE), 偏移量 0x408	612
寄存器 5:	GPIO 中断事件寄存器 (GPIOIEV), 偏移量 0x40C	613
寄存器 6:	GPIO 中断屏蔽寄存器 (GPIOIM), 偏移量 0x410	614
寄存器 7:	GPIO 原始中断状态寄存器 (GPIORIS), 偏移量 0x414	615
寄存器 8:	GPIO 屏蔽中断状态寄存器 (GPIOMIS), 偏移量 0x418	616
寄存器 9:	GPIO 中断清除寄存器 (GPIOICR), 偏移量 0x41C	617
寄存器 10:	GPIO 备用功能选择寄存器 (GPIOAFSEL), 偏移量 0x420	618
寄存器 11:	GPIO 2-mA 驱动选择寄存器 (GPIODR2R), 偏移量 0x500	620
寄存器 12:	GPIO 4-mA 驱动选择寄存器 (GPIODR4R), 偏移量 0x504	621
寄存器 13:	GPIO 8-mA 驱动选择寄存器 (GPIODR8R), 偏移量 0x508	622
寄存器 14:	GPIO 开漏选择寄存器 (GPIOODR), 偏移量 0x50C	623
寄存器 15:	GPIO 上拉电阻选择寄存器 (GPIOPUR), 偏移量 0x510	624
寄存器 16:	GPIO 下拉电阻选择寄存器 (GPIOPDR), 偏移量 0x514	626
寄存器 17:	GPIO 斜率控制选择寄存器 (GPIOSLR), 偏移量 0x518	628
寄存器 18:	GPIO 数字使能寄存器 (GPIODEN), 偏移量 0x51C	629
寄存器 19:	GPIO 锁定寄存器 (GPIOLOCK), 偏移量 0x520	631
寄存器 20:	GPIO 确认寄存器 (GPIOCR), 偏移量 0x524	632
寄存器 21:	GPIO 模拟选择寄存器 (GPIOAMSEL), 偏移量 0x528	634
寄存器 22:	GPIO 端口控制寄存器 (GPIOPCTL), 偏移量 0x52C	636
寄存器 23:	GPIO ADC 控制寄存器 (GPIOADCCTL), 偏移量 0x530	638
寄存器 24:	GPIO DMA 控制寄存器 (GPIODMACTL), 偏移量 0x534	639
寄存器 25:	GPIO 中断选择寄存器 (GPIOSI), 偏移量 0x538	640
寄存器 26:	GPIO 外设标识寄存器 4 (GPIOPeriphID4), 偏移量 0xFD0	641
寄存器 27:	GPIO 外设标识寄存器 5 (GPIOPeriphID5), 偏移量 0xFD4	642
寄存器 28:	GPIO 外设标识寄存器 6 (GPIOPeriphID6), 偏移量 0xFD8	643
寄存器 29:	GPIO 外设标识寄存器 7 (GPIOPeriphID7), 偏移量 0xFDC	644
寄存器 30:	GPIO 外设标识寄存器 0 (GPIOPeriphID0), 偏移量 0xFE0	645
寄存器 31:	GPIO 外设标识寄存器 1 (GPIOPeriphID1), 偏移量 0xFE4	646

寄存器 32:	GPIO 外设标识寄存器 2 ( GPIOPeriphID2 ) , 偏移量 0xFE8 .....	647
寄存器 33:	GPIO 外设标识寄存器 3 ( GPIOPeriphID3 ) , 偏移量 0xFEC .....	648
寄存器 34:	GPIO PrimeCell 标识寄存器 0 ( GPIOCellID0 ) , 偏移量 0xFF0 .....	649
寄存器 35:	GPIO PrimeCell 标识寄存器 1 ( GPIOCellID1 ) , 偏移量 0xFF4 .....	650
寄存器 36:	GPIO PrimeCell 标识寄存器 2 ( GPIOCellID2 ) , 偏移量 0xFF8 .....	651
寄存器 37:	GPIO PrimeCell 标识寄存器 3 ( GPIOCellID3 ) , 偏移量 0xFFC .....	652
<b>通用定时器</b>	<b>.....</b>	<b>653</b>
寄存器 1:	GPTM 配置寄存器 ( GPTMCFG ) , 偏移量 0x000 .....	675
寄存器 2:	GPTM Timer A 模式寄存器 ( GPTMTAMR ) , 偏移量 0x004 .....	676
寄存器 3:	GPTM Timer B 模式寄存器 ( GPTMTBMR ) , 偏移量 0x008 .....	679
寄存器 4:	GPTM 控制寄存器 ( GPTMCTL ) , 偏移量 0x00C .....	682
寄存器 5:	GPTM 同步寄存器 ( GPTMSYNC ) , 偏移量 0x010 .....	685
寄存器 6:	GPTM 中断屏蔽寄存器 ( GPTMIMR ) , 偏移量 0x018 .....	688
寄存器 7:	GPTM 原始中断状态寄存器 ( GPTMRIS ) , 偏移量 0x01C .....	691
寄存器 8:	GPTM 屏蔽的中断状态寄存器 ( GPTMMIS ) , 偏移量 0x020 .....	694
寄存器 9:	GPTM 中断清除寄存器 ( GPTMICR ) , 偏移量 0x024 .....	697
寄存器 10:	GPTM Timer A 间隔加载寄存器 ( GPTMTAILR ) , 偏移量 0x028 .....	699
寄存器 11:	GPTM Timer B 间隔加载寄存器 ( GPTMTBILR ) , 偏移量 0x02C .....	700
寄存器 12:	GPTM Timer A 匹配寄存器 ( GPTMTAMATCHR ) , 偏移量 0x030 .....	701
寄存器 13:	GPTM Timer B 匹配寄存器 ( GPTMTBMATCHR ) , 偏移量 0x034 .....	702
寄存器 14:	GPTM Timer A 预分频寄存器 ( GPTMTAPR ) , 偏移量 0x038 .....	703
寄存器 15:	GPTM Timer B 预分频寄存器 ( GPTMTBPR ) , 偏移量 0x03C .....	704
寄存器 16:	GPTM TimerA 预分频匹配寄存器 ( GPTMTAPMR ) , 偏移量 0x040 .....	705
寄存器 17:	GPTM TimerB 预分频匹配寄存器 ( GPTMTBPMR ) , 偏移量 0x044 .....	706
寄存器 18:	GPTM Timer A 寄存器 ( GPTMTAPR ) , 偏移量 0x048 .....	707
寄存器 19:	GPTM Timer B 寄存器 ( GPTMTBPR ) , 偏移量 0x04C .....	708
寄存器 20:	GPTM Timer A 值寄存器 ( GPTMTAV ) , 偏移量 0x050 .....	709
寄存器 21:	GPTM Timer B 值寄存器 ( GPTMTBV ) , 偏移量 0x054 .....	710
寄存器 22:	GPTM RTC 预分频寄存器 ( GPTMRTCPD ) , 偏移量 0x058 .....	711
寄存器 23:	GPTM Timer A 预分频快照寄存器 ( GPTMTAPS ) , 偏移量 0x05C .....	712
寄存器 24:	GPTM Timer B 预分频快照寄存器 ( GPTMTBPS ) , 偏移量 0x060 .....	713
寄存器 25:	GPTM Timer A 预分频值寄存器 ( GPTMTAPV ) , 偏移量 0x064 .....	714
寄存器 26:	GPTM Timer B 预分频值寄存器 ( GPTMTBPV ) , 偏移量 0x068 .....	715
寄存器 27:	GPTM 外设属性寄存器 ( GPTMPP ) , 偏移量 0xFC0 .....	716
<b>看门狗定时器 (WDT)</b>	<b>.....</b>	<b>717</b>
寄存器 1:	看门狗加载寄存器 ( WDTLOAD ) , 偏移量 0x000 .....	721
寄存器 2:	看门狗当前值寄存器 ( WDTVALUE ) , 偏移量 0x004 .....	722
寄存器 3:	看门狗控制寄存器 ( WDTCTL ) , 偏移量 0x008 .....	723
寄存器 4:	看门狗中断清除寄存器 ( WDTICR ) , 偏移量 0x00C .....	725
寄存器 5:	看门狗原始中断状态寄存器 ( WDTRIS ) , 偏移量 0x010 .....	726
寄存器 6:	看门狗可屏蔽中断状态寄存器 ( WDTMIS ) , 偏移量 0x014 .....	727
寄存器 7:	看门狗测试寄存器 ( WDTTEST ) , 偏移量 0x418 .....	728
寄存器 8:	看门狗锁定寄存器 ( WDTLOCK ) , 偏移量 0xC00 .....	729
寄存器 9:	看门狗外设标识寄存器 4 ( WDTPeriphID4 ) , 偏移量 0xFD0 .....	730
寄存器 10:	看门狗外设标识寄存器 5 ( WDTPeriphID5 ) , 偏移量 0xFD4 .....	731
寄存器 11:	看门狗外设标识寄存器 6 ( WDTPeriphID6 ) , 偏移量 0xFD8 .....	732
寄存器 12:	看门狗外设标识寄存器 7 ( WDTPeriphID7 ) , 偏移量 0xFDC .....	733
寄存器 13:	看门狗外设标识寄存器 0 ( WDTPeriphID0 ) , 偏移量 0xFE0 .....	734

寄存器 14:	看门狗外设标识寄存器 1 ( WDTPeriphID1 ) , 偏移量 0xFE4 .....	735
寄存器 15:	看门狗外设标识寄存器 2 ( WDTPeriphID2 ) , 偏移量 0xFE8 .....	736
寄存器 16:	看门狗外设标识寄存器 3 ( WDTPeriphID3 ) , 偏移量 0xFEC .....	737
寄存器 17:	看门狗 PrimeCell 标识寄存器 0 ( WDTPCellID0 ) , 偏移量 0xFF0 .....	738
寄存器 18:	看门狗 PrimeCell 标识寄存器 1 ( WDTPCellID1 ) , 偏移量 0xFF4 .....	739
寄存器 19:	看门狗 PrimeCell 标识寄存器 2 ( WDTPCellID2 ) , 偏移量 0xFF8 .....	740
寄存器 20:	看门狗 PrimeCell 标识寄存器 3 ( WDTPCellID3 ) , 偏移量 0xFFC .....	741
<b>模数转换器 (ADC)</b> .....		<b>742</b>
寄存器 1:	ADC 有效采样序列发生器寄存器 ( ADCACTSS ) , 偏移量 0x000 .....	763
寄存器 2:	ADC原始中断状态寄存器 ( ADCRIS ) , 偏移量 0x004 .....	764
寄存器 3:	ADC 中断掩码寄存器 ( ADCIM ) , 偏移量 0x008 .....	766
寄存器 4:	ADC 中断状态及清除寄存器 ( ADCISC ) , 偏移量 0x00C .....	768
寄存器 5:	ADC 上溢状态寄存器 ( ADCOSTAT ) , 偏移量 0x010 .....	771
寄存器 6:	ADC 事件复用选择寄存器 ( ADCEMUX ) , 偏移量 0x014 .....	773
寄存器 7:	ADC 下溢状态寄存器 ( ADCUSTAT ) , 偏移量 0x018 .....	778
寄存器 8:	ADC 触发源选择寄存器 ( ADCTSSEL ) , 偏移量 0x01C .....	779
寄存器 9:	ADC 采样序列发生器优先级寄存器 ( ADCSSPRI ) , 偏移量 0x020 .....	781
寄存器 10:	ADC 采样相位控制寄存器 ( ADCSPC ) , 偏移量 0x024 .....	782
寄存器 11:	ADC 处理器采样序列启动寄存器 ( ADCPSI ) , 偏移量 0x028 .....	783
寄存器 12:	ADC 采样平均控制寄存器 ( ADCSAC ) , 偏移量 0x030 .....	785
寄存器 13:	ADC 数字比较器中断状态及清除寄存器 ( ADCDCISC ) , 偏移量 0x034 .....	786
寄存器 14:	ADC 控制寄存器 ( ADCCTL ) , 偏移量 0x038 .....	788
寄存器 15:	ADC 采样序列输入复用选择寄存器 0 ( ADCSSMUX0 ) , 偏移量 0x040 .....	789
寄存器 16:	ADC 采样序列控制寄存器 0 ( ADCSSCTL0 ) , 偏移量 0x044 .....	791
寄存器 17:	ADC 采样序列结果 FIFO 寄存器 0 ( ADCSSFIFO0 ) , 偏移量 0x048 .....	794
寄存器 18:	ADC 采样序列结果 FIFO 寄存器 1 ( ADCSSFIFO1 ) , 偏移量 0x068 .....	794
寄存器 19:	ADC 采样序列结果 FIFO 寄存器 2 ( ADCSSFIFO2 ) , 偏移量 0x088 .....	794
寄存器 20:	ADC 采样序列结果 FIFO 寄存器 3 ( ADCSSFIFO3 ) , 偏移量 0x0A8 .....	794
寄存器 21:	ADC 采样序列 FIFO 0 状态寄存器 ( ADCSSFSTAT0 ) , 偏移量 0x04C .....	795
寄存器 22:	ADC 采样序列 FIFO 1 状态寄存器 ( ADCSSFSTAT1 ) , 偏移量 0x06C .....	795
寄存器 23:	ADC 采样序列 FIFO 2 状态寄存器 ( ADCSSFSTAT2 ) , 偏移量 0x08C .....	795
寄存器 24:	ADC 采样序列 FIFO 3 状态寄存器 ( ADCSSFSTAT3 ) , 偏移量 0x0AC .....	795
寄存器 25:	ADC 采样序列寄存器 0 ( ADCSSOP0 ) , 偏移量 0x050 .....	797
寄存器 26:	ADC 采样序列数字比较器选择寄存器 0 ( ADCSSDC0 ) , 偏移量 0x054 .....	799
寄存器 27:	ADC 采样序列扩展输入复用选择寄存器 ( ADCSSEMUX0 ) , 偏移量 0x058 .....	801
寄存器 28:	ADC 采样序列输入复用选择寄存器 1 ( ADCSSMUX1 ) , 偏移量 0x060 .....	803
寄存器 29:	ADC 采样序列输入复用选择寄存器 2 ( ADCSSMUX2 ) , 偏移量 0x080 .....	803
寄存器 30:	ADC 采样序列控制寄存器 1 ( ADCSSCTL1 ) , 偏移量 0x064 .....	804
寄存器 31:	ADC 采样序列控制寄存器 2 ( ADCSSCTL2 ) , 偏移量 0x084 .....	804
寄存器 32:	ADC 采样序列 1 工作寄存器 ( ADCSSOP1 ) , 偏移量 0x070 .....	806
寄存器 33:	ADC 采样序列 2 工作寄存器 ( ADCSSOP2 ) , 偏移量 0x090 .....	806
寄存器 34:	ADC 采样序列数字比较器选择寄存器 1 ( ADCSSDC1 ) , 偏移量 0x074 .....	807
寄存器 35:	ADC 采样序列数字比较器选择寄存器 2 ( ADCSSDC2 ) , 偏移量 0x094 .....	807
寄存器 36:	ADC 采样序列扩展输入复用选择寄存器 1 ( ADCSSEMUX1 ) , 偏移量 0x078 .....	808
寄存器 37:	ADC 采样序列扩展输入复用选择寄存器 2 ( ADCSSEMUX2 ) , 偏移量 0x098 .....	808
寄存器 38:	ADC 采样序列输入复用选择寄存器 3 ( ADCSSMUX3 ) , 偏移量 0x0A0 .....	810
寄存器 39:	ADC 采样序列控制寄存器 3 ( ADCSSCTL3 ) , 偏移量 0x0A4 .....	811
寄存器 40:	ADC 采样序列器 3 工作寄存器 ( ADCSSOP3 ) , 偏移量 0x0B0 .....	812

寄存器 41:	ADC 采样序列 3 数字比较器选择寄存器 ( ADCSSDC3 ) , 偏移量 0x0B4 .....	813
寄存器 42:	ADC 采样序列扩展输入复用选择寄存器 3 ( ADCSSMUX3 ) , 偏移量 0x0B8 .....	814
寄存器 43:	ADC 数字比较器复位启动条件寄存器 ( ADCDCRIC ) , 偏移量 0xD00 .....	815
寄存器 44:	ADC 数字比较器控制寄存器 0 ( ADCDCCTL0 ) , 偏移量 0xE00 .....	819
寄存器 45:	ADC 数字比较器控制寄存器 1 ( ADCDCCTL1 ) , 偏移量 0xE04 .....	819
寄存器 46:	ADC 数字比较器控制寄存器 2 ( ADCDCCTL2 ) , 偏移量 0xE08 .....	819
寄存器 47:	ADC 数字比较器控制寄存器 3 ( ADCDCCTL3 ) , 偏移量 0xE0C .....	819
寄存器 48:	ADC 数字比较器控制寄存器 4 ( ADCDCCTL4 ) , 偏移量 0xE10 .....	819
寄存器 49:	ADC 数字比较器控制寄存器 5 ( ADCDCCTL5 ) , 偏移量 0xE14 .....	819
寄存器 50:	ADC 数字比较器控制寄存器 6 ( ADCDCCTL6 ) , 偏移量 0xE18 .....	819
寄存器 51:	ADC 数字比较器控制寄存器 7 ( ADCDCCTL7 ) , 偏移量 0xE1C .....	819
寄存器 52:	ADC 数字比较器范围寄存器 0 ( ADCDCCMP0 ) , 偏移量 0xE40 .....	822
寄存器 53:	ADC 数字比较器范围寄存器 1 ( ADCDCCMP1 ) , 偏移量 0xE44 .....	822
寄存器 54:	ADC 数字比较器范围寄存器 2 ( ADCDCCMP2 ) , 偏移量 0xE48 .....	822
寄存器 55:	ADC 数字比较器范围寄存器 3 ( ADCDCCMP3 ) , 偏移量 0xE4C .....	822
寄存器 56:	ADC 数字比较器范围寄存器 4 ( ADCDCCMP4 ) , 偏移量 0xE50 .....	822
寄存器 57:	ADC 数字比较器范围寄存器 5 ( ADCDCCMP5 ) , 偏移量 0xE54 .....	822
寄存器 58:	ADC 数字比较器范围寄存器 6 ( ADCDCCMP6 ) , 偏移量 0xE58 .....	822
寄存器 59:	ADC 数字比较器范围寄存器 7 ( ADCDCCMP7 ) , 偏移量 0xE5C .....	822
寄存器 60:	ADC 外设属性寄存器 ( ADCPP ) , 偏移量 0xFC0 .....	823
寄存器 61:	ADC 外设配置寄存器 ( ADCPC ) , 偏移量 0xFC4 .....	825
寄存器 62:	ADC 时钟配置寄存器 ( ADCCC ) , 偏移量 0xFC8 .....	826
<b>通用异步收发器 (UART) .....</b>		<b>827</b>
寄存器 1:	UART 数据寄存器 ( UARTDR ) , 偏移量 0x000 .....	840
寄存器 2:	UART 接收状态/错误清除寄存器 ( UARTRSR/UARTECR ) , 偏移量 0x004 .....	842
寄存器 3:	UART 标志寄存器 ( UARTFR ) , 偏移量 0x018 .....	845
寄存器 4:	UART IrDA 低功耗寄存器 ( UARTILPR ) , 偏移量 0x020 .....	847
寄存器 5:	UART 波特率分频值整数寄存器 ( UARTIBRD ) , 偏移量 0x024 .....	848
寄存器 6:	UART 波特率分频值小数寄存器 ( UARTFBRD ) , 偏移量 0x028 .....	849
寄存器 7:	UART 线控寄存器 ( UARTLCRH ) , 偏移量 0x02C .....	850
寄存器 8:	UART 控制寄存器 ( UARTCTL ) , 偏移量 0x030 .....	852
寄存器 9:	UART 中断 FIFO 深度选择寄存器 ( UARTIFLS ) , 偏移量 0x034 .....	856
寄存器 10:	UART 中断屏蔽寄存器 ( UARTIM ) , 偏移量 0x038 .....	858
寄存器 11:	UART 原始中断状态寄存器 ( UARTRIS ) , 偏移量 0x03C .....	861
寄存器 12:	UART 屏蔽中断状态寄存器 ( UARTMIS ) , 偏移量 0x040 .....	865
寄存器 13:	UART 中断清除寄存器 ( UARTICR ) , 偏移量 0x044 .....	869
寄存器 14:	UART DMA 控制寄存器 ( UARTDMACTL ) , 偏移量 0x048 .....	871
寄存器 15:	UART LIN 控制寄存器 ( UARTLCTL ) , 偏移量 0x090 .....	872
寄存器 16:	UART LIN 快照寄存器 ( UARTLSS ) , 偏移量 0x094 .....	873
寄存器 17:	UART LIN 定时寄存器 ( UARTLTIM ) , 偏移量 0x098 .....	874
寄存器 18:	UART 9 位模式自身地址寄存器 ( UART9BITADDR ) , 偏移量 0x0A4 .....	875
寄存器 19:	UART 9 位模式自身地址屏蔽寄存器 ( UART9BITAMASK ) , 偏移量 0x0A8 .....	876
寄存器 20:	UART 外设属性寄存器 ( UARTPP ) , 偏移量 0xFC0 .....	877
寄存器 21:	UART 时钟配置寄存器 ( UARTCC ) , 偏移量 0xFC8 .....	878
寄存器 22:	UART 外设标识寄存器 4 ( UARTPeriphID4 ) , 偏移量 0xFD0 .....	879
寄存器 23:	UART 外设标识寄存器 5 ( UARTPeriphID5 ) , 偏移量 0xFD4 .....	880
寄存器 24:	UART 外设标识寄存器 6 ( UARTPeriphID6 ) , 偏移量 0xFD8 .....	881
寄存器 25:	UART 外设标识寄存器 7 ( UARTPeriphID7 ) , 偏移量 0xFDC .....	882

寄存器 26:	UART 外设标识寄存器 0 ( UARTPeriphID0 ) , 偏移量 0xFE0 .....	883
寄存器 27:	UART 外设标识寄存器 1 ( UARTPeriphID1 ) , 偏移量 0xFE4 .....	884
寄存器 28:	UART 外设标识寄存器 2 ( UARTPeriphID2 ) , 偏移量 0xFE8 .....	885
寄存器 29:	UART 外设标识寄存器 3 ( UARTPeriphID3 ) , 偏移量 0xFEC .....	886
寄存器 30:	UART PrimeCell 标识寄存器 0 ( UARTPCelIID0 ) , 偏移量 0xFF0 .....	887
寄存器 31:	UART PrimeCell 标识寄存器 1 ( UARTPCelIID1 ) , 偏移量 0xFF4 .....	888
寄存器 32:	UART PrimeCell 标识寄存器 2 ( UARTPCelIID2 ) , 偏移量 0xFF8 .....	889
寄存器 33:	UART PrimeCell 标识寄存器 3 ( UARTPCelIID3 ) , 偏移量 0xFFC .....	890
<b>同步串行接口 (SSI) .....</b>		<b>891</b>
寄存器 1:	SSI 控制寄存器0 ( SSICR0 ) , 偏移量 0x000 .....	905
寄存器 2:	SSI 控制寄存器1 ( SSICR1 ) , 偏移量 0x004 .....	907
寄存器 3:	SSI 数据寄存器 ( SSIDR ) , 偏移量 0x008 .....	909
寄存器 4:	SSI 状态寄存器 ( SSISR ) , 偏移量 0x00C .....	910
寄存器 5:	SSI 时钟预分频寄存器 ( SSICPSR ) , 偏移量 0x010 .....	912
寄存器 6:	SSI 中断屏蔽寄存器 ( SSIIM ) , 偏移量 0x014 .....	913
寄存器 7:	SSI 原始中断状态寄存器 ( SSIRIS ) , 偏移量 0x018 .....	914
寄存器 8:	SSI 屏蔽中断状态寄存器 ( SSIMIS ) , 偏移量 0x01C .....	916
寄存器 9:	SSI 中断清除寄存器 ( SSIICR ) , 偏移量 0x020 .....	918
寄存器 10:	SSI DMA 控制寄存器 ( SSIDMACTL ) , 偏移量 0x024 .....	919
寄存器 11:	SSI 时钟配置寄存器 ( SSICC ) , 偏移量 0xFC8 .....	920
寄存器 12:	SSI 外设标识寄存器 4 ( SSIPeriphID4 ) , 偏移量 0xFD0 .....	921
寄存器 13:	SSI 外设标识寄存器 5 ( SSIPeriphID5 ) , 偏移量 0xFD4 .....	922
寄存器 14:	SSI 外设标识寄存器 6 ( SSIPeriphID6 ) , 偏移量 0xFD8 .....	923
寄存器 15:	SSI 外设标识寄存器 7 ( SSIPeriphID7 ) , 偏移量 0xFDC .....	924
寄存器 16:	SSI 外设标识寄存器 0 ( SSIPeriphID0 ) , 偏移量 0xFE0 .....	925
寄存器 17:	SSI 外设标识寄存器 1 ( SSIPeriphID1 ) , 偏移量 0xFE4 .....	926
寄存器 18:	SSI 外设标识寄存器 2 ( SSIPeriphID2 ) , 偏移量 0xFE8 .....	927
寄存器 19:	SSI 外设标识寄存器 3 ( SSIPeriphID3 ) , 偏移量 0xFEC .....	928
寄存器 20:	SSI PrimeCell 标识寄存器 0 ( SSIPCellIID0 ) , 偏移量 0xFF0 .....	929
寄存器 21:	SSI PrimeCell 标识寄存器 1 ( SSIPCellIID1 ) , 偏移量 0xFF4 .....	930
寄存器 22:	SSI PrimeCell 标识寄存器 2 ( SSIPCellIID2 ) , 偏移量 0xFF8 .....	931
寄存器 23:	SSI PrimeCell 标识寄存器 3 ( SSIPCellIID3 ) , 偏移量 0xFFC .....	932
<b>内部集成电路 (I<sup>2</sup>C) 接口 .....</b>		<b>933</b>
寄存器 1:	I <sup>2</sup> C 主机从机地址寄存器 ( I2CMSA ) , 偏移量 0x000 .....	952
寄存器 2:	I <sup>2</sup> C 主机控制/状态寄存器 ( I2CMCS ) , 偏移量 0x004 .....	953
寄存器 3:	I <sup>2</sup> C 主机数据寄存器 ( I2CMDR ) , 偏移量 0x008 .....	958
寄存器 4:	I <sup>2</sup> C 主机定时器周期寄存器 ( I2CMTPR ) , 偏移量 0x00C .....	959
寄存器 5:	I <sup>2</sup> C 主机中断屏蔽寄存器 ( I2CMIMR ) , 偏移量 0x010 .....	960
寄存器 6:	I <sup>2</sup> C 主机原始中断状态寄存器 ( I2CMRIS ) , 偏移量 0x014 .....	961
寄存器 7:	I <sup>2</sup> C 主机屏蔽中断状态寄存器 ( I2CMMIS ) , 偏移量 0x018 .....	962
寄存器 8:	I <sup>2</sup> C 主机中断清除寄存器 ( I2CMICR ) , 偏移量 0x01C .....	963
寄存器 9:	I <sup>2</sup> C 主机配置寄存器 ( I2CMCR ) , 偏移量 0x020 .....	964
寄存器 10:	I <sup>2</sup> C 主机时钟低电平超时计数寄存器 ( I2MCLKOCNT ) , 偏移量 0x024 .....	965
寄存器 11:	I <sup>2</sup> C 主机总线监视寄存器 ( I2CMBMON ) , 偏移量 0x02C .....	966
寄存器 12:	I <sup>2</sup> C 从机本身地址寄存器 ( I2CSOAR ) , 偏移量 0x800 .....	967
寄存器 13:	I <sup>2</sup> C 从机控制/状态寄存器 ( I2CSCSR ) , 偏移量 0x804 .....	968
寄存器 14:	I <sup>2</sup> C 从机数据寄存器 ( I2CSDR ) , 偏移量 0x808 .....	970

寄存器 15:	I <sup>2</sup> C 从机中断屏蔽 ( I2CSIMR ) , 偏移量 0x80C .....	971
寄存器 16:	I <sup>2</sup> C 从机原始中断状态寄存器 ( I2CSRIS ) , 偏移量 0x810 .....	972
寄存器 17:	I <sup>2</sup> C 从机屏蔽中断状态寄存器 ( I2CSMIS ) , 偏移量 0x814 .....	973
寄存器 18:	I <sup>2</sup> C 从机中断清除寄存器 ( I2CSICR ) , 偏移量 0x818 .....	974
寄存器 19:	I <sup>2</sup> C 从机自身地址寄存器 2 ( I2CSOAR2 ) , 偏移量 0x81C .....	975
寄存器 20:	I <sup>2</sup> C 从机应答控制寄存器 ( I2CSACKCTL ) , 偏移量 0x820 .....	976
寄存器 21:	I <sup>2</sup> C 外设属性寄存器 ( I2CPP ) , 偏移量 0xFC0 .....	977
寄存器 22:	I <sup>2</sup> C 外配置寄存器 ( I2CPC ) , 偏移量 0xFC4 .....	978
<b>控制器局域网 (CAN) 模块 .....</b>		<b>979</b>
寄存器 1:	CAN 控制寄存器 ( CANCTL ) , 偏移量 0x000 .....	998
寄存器 2:	CAN 状态寄存器 ( CANSTS ) , 偏移量 0x004 .....	1000
寄存器 3:	CAN 错误计数寄存器 ( CANERR ) , 偏移量 0x008 .....	1003
寄存器 4:	CAN 位时序寄存器 ( CANBIT ) , 偏移量 0x00C .....	1004
寄存器 5:	CAN 中断寄存器 ( CANINT ) , 偏移量 0x010 .....	1005
寄存器 6:	CAN 测试寄存器 ( CANTST ) , 偏移量 0x014 .....	1006
寄存器 7:	CAN 波特率预分频器扩展寄存器 ( CANBRPE ) , 偏移量 0x018 .....	1008
寄存器 8:	CAN IF1 指令请求寄存器 ( CANIF1CRQ ) , 偏移量 0x020 .....	1009
寄存器 9:	CAN IF2 指令请求寄存器 ( CANIF2CRQ ) , 偏移量 0x080 .....	1009
寄存器 10:	CAN IF1 指令屏蔽寄存器 ( CANIF1CMSK ) , 偏移量 0x024 .....	1010
寄存器 11:	CAN IF2 指令屏蔽寄存器 ( CANIF2CMSK ) , 偏移量 0x084 .....	1010
寄存器 12:	CAN IF1 屏蔽寄存器 1 ( CANIF1MSK1 ) , 偏移量 0x028 .....	1013
寄存器 13:	CAN IF2 屏蔽寄存器 1 ( CANIF2MSK1 ) , 偏移量 0x088 .....	1013
寄存器 14:	CAN IF1 屏蔽寄存器 2 ( CANIF1MSK2 ) , 偏移量 0x02C .....	1014
寄存器 15:	CAN IF2 屏蔽寄存器 2 ( CANIF2MSK2 ) , 偏移量 0x08C .....	1014
寄存器 16:	CAN IF1 仲裁寄存器 1 ( CANIF1ARB1 ) , 偏移量 0x030 .....	1015
寄存器 17:	CAN IF2 仲裁寄存器 1 ( CANIF2ARB1 ) , 偏移量 0x090 .....	1015
寄存器 18:	CAN IF1 仲裁寄存器 2 ( CANIF1ARB2 ) , 偏移量 0x034 .....	1016
寄存器 19:	CAN IF2 仲裁寄存器 2 ( CANIF2ARB2 ) , 偏移量 0x094 .....	1016
寄存器 20:	CAN IF1 报文控制寄存器 ( CANIF1MCTL ) , 偏移量 0x038 .....	1018
寄存器 21:	CAN IF2 报文控制寄存器 ( CANIF2MCTL ) , 偏移量 0x098 .....	1018
寄存器 22:	CAN IF1 数据寄存器 A1 ( CANIF1DA1 ) , 偏移量 0x03C .....	1021
寄存器 23:	CAN IF1 数据寄存器 A2 ( CANIF1DA2 ) , 偏移量 0x040 .....	1021
寄存器 24:	CAN IF1 数据寄存器 B1 ( CANIF1DB1 ) , 偏移量 0x044 .....	1021
寄存器 25:	CAN IF1 数据寄存器 B2 ( CANIF1DB2 ) , 偏移量 0x048 .....	1021
寄存器 26:	CAN IF2 数据寄存器 A1 ( CANIF2DA1 ) , 偏移量 0x09C .....	1021
寄存器 27:	CAN IF2 数据寄存器 A2 ( CANIF2DA2 ) , 偏移量 0x0A0 .....	1021
寄存器 28:	CAN IF2 数据寄存器 B1 ( CANIF2DB1 ) , 偏移量 0x0A4 .....	1021
寄存器 29:	CAN IF2 数据寄存器 B2 ( CANIF2DB2 ) , 偏移量 0x0A8 .....	1021
寄存器 30:	CAN 传输请求寄存器 1 ( CANTXRQ1 ) , 偏移量 0x100 .....	1022
寄存器 31:	CAN 传输请求寄存器 2 ( CANTXRQ2 ) , 偏移量 0x104 .....	1022
寄存器 32:	CAN 新数据寄存器 1 ( CANNWDA1 ) , 偏移量 0x120 .....	1023
寄存器 33:	CAN 新数据寄存器 2 ( CANNWDA2 ) , 偏移量 0x124 .....	1023
寄存器 34:	CAN 报文 1 中断挂起寄存器 ( CANMSG1INT ) , 偏移量 0x140 .....	1024
寄存器 35:	CAN 报文 2 中断挂起寄存器 ( CANMSG2INT ) , 偏移量 0x144 .....	1024
寄存器 36:	CAN 报文 1 有效寄存器 ( CANMSG1VAL ) , 偏移量 0x160 .....	1025
寄存器 37:	CAN 报文 2 有效寄存器 ( CANMSG2VAL ) , 偏移量 0x164 .....	1025

通用串行总线 (USB) 控制器 .....	1026
寄存器 1: USB 设备功能地址寄存器 ( USBFADDR ) , 偏移量 0x000 .....	1045
寄存器 2: USB电源寄存器 ( USBPOWER ) , 偏移量 0x001 .....	1046
寄存器 3: USB发送中断状态寄存器 ( USBTXIS ) , 偏移量 0x002 .....	1049
寄存器 4: USB接收中断状态寄存器 ( USBRXIS ) , 偏移量 0x004 .....	1051
寄存器 5: USB发送中断启用寄存器 ( USBTXIE ) , 偏移量 0x006 .....	1052
寄存器 6: USB接收中断启用寄存器 ( USBRXIE ) , 偏移量 0x008 .....	1053
寄存器 7: USB通用中断状态寄存器 ( USBIS ) , 偏移量 0x00A .....	1054
寄存器 8: USB中断启用 ( USBIE ) , 偏移量 0x00B .....	1057
寄存器 9: USB帧值 ( USBFRAME ) , 偏移量 0x00C .....	1060
寄存器 10: USB 端点索引寄存器 ( USBEPIDX ) , 偏移量 0x00E .....	1061
寄存器 11: USB 测试模式寄存器 ( USBTEST ) , 偏移量 0x00F .....	1062
寄存器 12: USB FIFO 端点 0 ( USBFIFO0 ) , 偏移量 0x020 .....	1064
寄存器 13: USB FIFO 端点 1 ( USBFIFO1 ) , 偏移量 0x024 .....	1064
寄存器 14: USB FIFO 端点 2 ( USBFIFO2 ) , 偏移量 0x028 .....	1064
寄存器 15: USB FIFO 端点 3 ( USBFIFO3 ) , 偏移量 0x02C .....	1064
寄存器 16: USB FIFO 端点 4 ( USBFIFO4 ) , 偏移量 0x030 .....	1064
寄存器 17: USB FIFO 端点 5 ( USBFIFO5 ) , 偏移量 0x034 .....	1064
寄存器 18: USB FIFO 端点 6 ( USBFIFO6 ) , 偏移量 0x038 .....	1064
寄存器 19: USB FIFO 端点 7 ( USBFIFO7 ) , 偏移量 0x03C .....	1064
寄存器 20: USB设备控制寄存器 ( USBDEVCTL ) , 偏移量 0x060 .....	1065
寄存器 21: USB 发送动态 FIFO 大小 ( USBTXFIFOSZ ) , 偏移量 0x062 .....	1067
寄存器 22: USB 接收动态 FIFO 大小 ( USBRXFIFOSZ ) , 偏移量 0x063 .....	1067
寄存器 23: USB 发送 FIFO 起始地址 ( USBTXFIFOADD ) , 偏移量 0x064 .....	1068
寄存器 24: USB 接收 FIFO 起始地址 ( USBRXFIFOADD ) , 偏移量 0x066 .....	1068
寄存器 25: USB连接时序寄存器 ( USBCONTIM ) , 偏移量 0x07A .....	1069
寄存器 26: USB OTG VBUS脉冲时序寄存器 ( USBVPLEN ) , 偏移量 0x07B .....	1070
寄存器 27: USB 全速模式下最后的传输与帧结束时序寄存器 ( USBFSEOF ) , 偏移量 0x07D .....	1071
寄存器 28: USB 低速模式下最后的传输与帧结束时序寄存器 ( USBLSEOF ) , 偏移量 0x07E .....	1072
寄存器 29: USB 发送端点 0 功能地址 ( USBTXFUNCADDR0 ) , 偏移量 0x080 .....	1073
寄存器 30: USB 发送端点 1 功能地址 ( USBTXFUNCADDR1 ) , 偏移量 0x088 .....	1073
寄存器 31: USB 发送端点 2 功能地址 ( USBTXFUNCADDR2 ) , 偏移量 0x090 .....	1073
寄存器 32: USB 发送端点 3 功能地址 ( USBTXFUNCADDR3 ) , 偏移量 0x098 .....	1073
寄存器 33: USB 发送端点 4 功能地址 ( USBTXFUNCADDR4 ) , 偏移量 0x0A0 .....	1073
寄存器 34: USB 发送端点 5 功能地址 ( USBTXFUNCADDR5 ) , 偏移量 0x0A8 .....	1073
寄存器 35: USB 发送端点 6 功能地址 ( USBTXFUNCADDR6 ) , 偏移量 0x0B0 .....	1073
寄存器 36: USB 发送端点 7 功能地址 ( USBTXFUNCADDR7 ) , 偏移量 0x0B8 .....	1073
寄存器 37: USB 发送端点 0 集线器地址 ( USBTXHUBADDR0 ) , 偏移量 0x082 .....	1074
寄存器 38: USB 发送端点 1 集线器地址 ( USBTXHUBADDR1 ) , 偏移量 0x08A .....	1074
寄存器 39: USB 发送端点 2 集线器地址 ( USBTXHUBADDR2 ) , 偏移量 0x092 .....	1074
寄存器 40: USB 发送端点 3 集线器地址 ( USBTXHUBADDR3 ) , 偏移量 0x09A .....	1074
寄存器 41: USB 发送端点 4 集线器地址 ( USBTXHUBADDR4 ) , 偏移量 0x0A2 .....	1074
寄存器 42: USB 发送端点 5 集线器地址 ( USBTXHUBADDR5 ) , 偏移量 0x0AA .....	1074
寄存器 43: USB 发送端点 6 集线器地址 ( USBTXHUBADDR6 ) , 偏移量 0x0B2 .....	1074
寄存器 44: USB 发送端点 7 集线器地址 ( USBTXHUBADDR7 ) , 偏移量 0x0BA .....	1074
寄存器 45: USB 发送端点 0 集线器端口 ( USBTXHUBPORT0 ) , 偏移量 0x083 .....	1075
寄存器 46: USB 发送端点 1 集线器端口 ( USBTXHUBPORT1 ) , 偏移量 0x08B .....	1075
寄存器 47: USB 发送端点 2 集线器端口 ( USBTXHUBPORT2 ) , 偏移量 0x093 .....	1075

寄存器 48:	USB 发送端点 3 集线器端口 ( USBTXHUBPORT3 ) , 偏移量 0x09B .....	1075
寄存器 49:	USB 发送端点 4 集线器端口 ( USBTXHUBPORT4 ) , 偏移量 0x0A3 .....	1075
寄存器 50:	USB 发送端点 5 集线器端口 ( USBTXHUBPORT5 ) , 偏移量 0x0AB .....	1075
寄存器 51:	USB 发送端点 6 集线器端口 ( USBTXHUBPORT6 ) , 偏移量 0x0B3 .....	1075
寄存器 52:	USB 发送端点 7 集线器端口 ( USBTXHUBPORT7 ) , 偏移量 0x0BB .....	1075
寄存器 53:	USB 接收端点 1 功能地址 ( USBRXFUNCADDR1 ) , 偏移量 0x08C .....	1076
寄存器 54:	USB 接收端点 2 功能地址 ( USBRXFUNCADDR2 ) , 偏移量 0x094 .....	1076
寄存器 55:	USB 接收端点 3 功能地址 ( USBRXFUNCADDR3 ) , 偏移量 0x09C .....	1076
寄存器 56:	USB 接收端点 4 功能地址 ( USBRXFUNCADDR4 ) , 偏移量 0x0A4 .....	1076
寄存器 57:	USB 接收端点 5 功能地址 ( USBRXFUNCADDR5 ) , 偏移量 0x0AC .....	1076
寄存器 58:	USB 接收端点 6 功能地址 ( USBRXFUNCADDR6 ) , 偏移量 0x0B4 .....	1076
寄存器 59:	USB 接收端点 7 功能地址 ( USBRXFUNCADDR7 ) , 偏移量 0x0BC .....	1076
寄存器 60:	USB 接收端点 1 集线器地址 ( USBRXHUBADDR1 ) , 偏移量 0x08E .....	1077
寄存器 61:	USB 接收端点 2 集线器地址 ( USBRXHUBADDR2 ) , 偏移量 0x096 .....	1077
寄存器 62:	USB 接收端点 3 集线器地址 ( USBRXHUBADDR3 ) , 偏移量 0x09E .....	1077
寄存器 63:	USB 接收端点 4 集线器地址 ( USBRXHUBADDR4 ) , 偏移量 0x0A6 .....	1077
寄存器 64:	USB 接收端点 5 集线器地址 ( USBRXHUBADDR5 ) , 偏移量 0x0AE .....	1077
寄存器 65:	USB 接收端点 6 集线器地址 ( USBRXHUBADDR6 ) , 偏移量 0x0B6 .....	1077
寄存器 66:	USB 接收端点 7 集线器地址 ( USBRXHUBADDR7 ) , 偏移量 0x0BE .....	1077
寄存器 67:	USB 接收端点 1 集线器端口 ( USBRXHUBPORT1 ) , 偏移量 0x08F .....	1078
寄存器 68:	USB 接收端点 2 集线器端口 ( USBRXHUBPORT2 ) , 偏移量 0x097 .....	1078
寄存器 69:	USB 接收端点 3 集线器端口 ( USBRXHUBPORT3 ) , 偏移量 0x09F .....	1078
寄存器 70:	USB 接收端点 4 集线器端口 ( USBRXHUBPORT4 ) , 偏移量 0x0A7 .....	1078
寄存器 71:	USB 接收端点 5 集线器端口 ( USBRXHUBPORT5 ) , 偏移量 0x0AF .....	1078
寄存器 72:	USB 接收端点 6 集线器端口 ( USBRXHUBPORT6 ) , 偏移量 0x0B7 .....	1078
寄存器 73:	USB 接收端点 7 集线器端口 ( USBRXHUBPORT7 ) , 偏移量 0x0BF .....	1078
寄存器 74:	USB 发送端点 1 最大传输数据 ( USBTXMAXP1 ) , 偏移量 0x110 .....	1079
寄存器 75:	USB 发送端点 2 最大传输数据 ( USBTXMAXP2 ) , 偏移量 0x120 .....	1079
寄存器 76:	USB 发送端点 3 最大传输数据 ( USBTXMAXP3 ) , 偏移量 0x130 .....	1079
寄存器 77:	USB 发送端点 4 最大传输数据 ( USBTXMAXP4 ) , 偏移量 0x140 .....	1079
寄存器 78:	USB 发送端点 5 最大传输数据 ( USBTXMAXP5 ) , 偏移量 0x150 .....	1079
寄存器 79:	USB 发送端点 6 最大传输数据 ( USBTXMAXP6 ) , 偏移量 0x160 .....	1079
寄存器 80:	USB 发送端点 7 最大传输数据 ( USBTXMAXP7 ) , 偏移量 0x170 .....	1079
寄存器 81:	USB 端点 0 控制和状态低字节寄存器 ( USBCSRL0 ) , 偏移量 0x102 .....	1080
寄存器 82:	USB 端点 0 控制和状态高字节寄存器 ( USBCSRH0 ) , 偏移量 0x103 .....	1084
寄存器 83:	USB 端点 0 接收字节数量寄存器 ( USBCOUNT0 ) , 偏移量 0x108 .....	1086
寄存器 84:	USB 端点 0 类型寄存器 ( USBTYPE0 ) , 偏移量 0x10A .....	1087
寄存器 85:	USB NAK 限制寄存器 ( USBNAKLMT ) , 偏移量 0x10B .....	1088
寄存器 86:	USB 端点 1 发送控制和状态低字节 ( USBTXCSRL1 ) , 偏移量 0x112 .....	1089
寄存器 87:	USB 端点 2 发送控制和状态低字节 ( USBTXCSRL2 ) , 偏移量 0x122 .....	1089
寄存器 88:	USB 端点 3 发送控制和状态低字节 ( USBTXCSRL3 ) , 偏移量 0x132 .....	1089
寄存器 89:	USB 端点 4 发送控制和状态低字节 ( USBTXCSRL4 ) , 偏移量 0x142 .....	1089
寄存器 90:	USB 端点 5 发送控制和状态低字节 ( USBTXCSRL5 ) , 偏移量 0x152 .....	1089
寄存器 91:	USB 端点 6 发送控制和状态低字节 ( USBTXCSRL6 ) , 偏移量 0x162 .....	1089
寄存器 92:	USB 端点 7 发送控制和状态低字节 ( USBTXCSRL7 ) , 偏移量 0x172 .....	1089
寄存器 93:	USB 发送端点 1 控制和状态高字节 ( USBTXCSRH1 ) , 偏移量 0x113 .....	1093
寄存器 94:	USB 发送端点 2 控制和状态高字节 ( USBTXCSRH2 ) , 偏移量 0x123 .....	1093
寄存器 95:	USB 发送端点 3 控制和状态高字节 ( USBTXCSRH3 ) , 偏移量 0x133 .....	1093



寄存器 96:	USB 发送端点 4 控制和状态高字节 ( USBTXCSRH4 ) , 偏移量 0x143 .....	1093
寄存器 97:	USB 发送端点 5 控制和状态高字节 ( USBTXCSRH5 ) , 偏移量 0x153 .....	1093
寄存器 98:	USB 发送端点 6 控制和状态高字节 ( USBTXCSRH6 ) , 偏移量 0x163 .....	1093
寄存器 99:	USB 发送端点 7 控制和状态高字节 ( USBTXCSRH7 ) , 偏移量 0x173 .....	1093
寄存器 100:	USB 接收端点 1 最大传输数据 ( USBRXMAXP1 ) , 偏移量 0x114 .....	1097
寄存器 101:	USB 接收端点 2 最大传输数据 ( USBRXMAXP2 ) , 偏移量 0x124 .....	1097
寄存器 102:	USB 接收端点 3 最大传输数据 ( USBRXMAXP3 ) , 偏移量 0x134 .....	1097
寄存器 103:	USB 接收端点 4 最大传输数据 ( USBRXMAXP4 ) , 偏移量 0x144 .....	1097
寄存器 104:	USB 接收端点 5 最大传输数据 ( USBRXMAXP5 ) , 偏移量 0x154 .....	1097
寄存器 105:	USB 接收端点 6 最大传输数据 ( USBRXMAXP6 ) , 偏移量 0x164 .....	1097
寄存器 106:	USB 接收端点 7 最大传输数据 ( USBRXMAXP7 ) , 偏移量 0x174 .....	1097
寄存器 107:	USB 接收端点 1 控制和状态低字节 ( USBRXCSRL1 ) , 偏移量 0x116 .....	1098
寄存器 108:	USB 接收端点 2 控制和状态低字节 ( USBRXCSRL2 ) , 偏移量 0x126 .....	1098
寄存器 109:	USB 接收端点 3 控制和状态低字节 ( USBRXCSRL3 ) , 偏移量 0x136 .....	1098
寄存器 110:	USB 接收端点 4 控制和状态低字节 ( USBRXCSRL4 ) , 偏移量 0x146 .....	1098
寄存器 111:	USB 接收端点 5 控制和状态低字节 ( USBRXCSRL5 ) , 偏移量 0x156 .....	1098
寄存器 112:	USB 接收端点 6 控制和状态低字节 ( USBRXCSRL6 ) , 偏移量 0x166 .....	1098
寄存器 113:	USB 接收端点 7 控制和状态低字节 ( USBRXCSRL7 ) , 偏移量 0x176 .....	1098
寄存器 114:	USB 接收端点 1 控制和状态高字节 ( USBRXCSRH1 ) , 偏移量 0x117 .....	1102
寄存器 115:	USB 接收端点 2 控制和状态高字节 ( USBRXCSRH2 ) , 偏移量 0x127 .....	1102
寄存器 116:	USB 接收端点 3 控制和状态高字节 ( USBRXCSRH3 ) , 偏移量 0x137 .....	1102
寄存器 117:	USB 接收端点 4 控制和状态高字节 ( USBRXCSRH4 ) , 偏移量 0x147 .....	1102
寄存器 118:	USB 接收端点 5 控制和状态高字节 ( USBRXCSRH5 ) , 偏移量 0x157 .....	1102
寄存器 119:	USB 接收端点 6 控制和状态高字节 ( USBRXCSRH6 ) , 偏移量 0x167 .....	1102
寄存器 120:	USB 接收端点 7 控制和状态高字节 ( USBRXCSRH7 ) , 偏移量 0x177 .....	1102
寄存器 121:	USB 接收端点 1 字节计数 ( USBRXCOUNT1 ) , 偏移量 0x118 .....	1106
寄存器 122:	USB 接收端点 2 字节计数 ( USBRXCOUNT2 ) , 偏移量 0x128 .....	1106
寄存器 123:	USB 接收端点 3 字节计数 ( USBRXCOUNT3 ) , 偏移量 0x138 .....	1106
寄存器 124:	USB 接收端点 4 字节计数 ( USBRXCOUNT4 ) , 偏移量 0x148 .....	1106
寄存器 125:	USB 接收端点 5 字节计数 ( USBRXCOUNT5 ) , 偏移量 0x158 .....	1106
寄存器 126:	USB 接收端点 6 字节计数 ( USBRXCOUNT6 ) , 偏移量 0x168 .....	1106
寄存器 127:	USB 接收端点 7 字节计数 ( USBRXCOUNT7 ) , 偏移量 0x178 .....	1106
寄存器 128:	USB 端点 1 主机发送配置类型 ( USBTXTYPE1 ) , 偏移量 0x11A .....	1107
寄存器 129:	USB 端点 2 主机发送配置类型 ( USBTXTYPE2 ) , 偏移量 0x12A .....	1107
寄存器 130:	USB 端点 3 主机发送配置类型 ( USBTXTYPE3 ) , 偏移量 0x13A .....	1107
寄存器 131:	USB 端点 4 主机发送配置类型 ( USBTXTYPE4 ) , 偏移量 0x14A .....	1107
寄存器 132:	USB 端点 5 主机发送配置类型 ( USBTXTYPE5 ) , 偏移量 0x15A .....	1107
寄存器 133:	USB 端点 6 主机发送配置类型 ( USBTXTYPE6 ) , 偏移量 0x16A .....	1107
寄存器 134:	USB 端点 7 主机发送配置类型 ( USBTXTYPE7 ) , 偏移量 0x17A .....	1107
寄存器 135:	USB 端点 1 主机发送间隔 ( USBTXINTERVAL1 ) , 偏移量 0x11B .....	1109
寄存器 136:	USB 端点 2 主机发送间隔 ( USBTXINTERVAL2 ) , 偏移量 0x12B .....	1109
寄存器 137:	USB 端点 3 主机发送间隔 ( USBTXINTERVAL3 ) , 偏移量 0x13B .....	1109
寄存器 138:	USB 端点 4 主机发送间隔 ( USBTXINTERVAL4 ) , 偏移量 0x14B .....	1109
寄存器 139:	USB 端点 5 主机发送间隔 ( USBTXINTERVAL5 ) , 偏移量 0x15B .....	1109
寄存器 140:	USB 端点 6 主机发送间隔 ( USBTXINTERVAL6 ) , 偏移量 0x16B .....	1109
寄存器 141:	USB 端点 7 主机发送间隔 ( USBTXINTERVAL7 ) , 偏移量 0x17B .....	1109
寄存器 142:	USB 端点 1 主机配置接收类型 ( USBRXTYPE1 ) , 偏移量 0x11C .....	1110
寄存器 143:	USB 端点 2 主机配置接收类型 ( USBRXTYPE2 ) , 偏移量 0x12C .....	1110

寄存器 144:	USB 端点 3 主机配置接收类型 ( USBRXTYPE3 ) , 偏移量 0x13C .....	1110
寄存器 145:	USB 端点 4 主机配置接收类型 ( USBRXTYPE4 ) , 偏移量 0x14C .....	1110
寄存器 146:	USB 端点 5 主机配置接收类型 ( USBRXTYPE5 ) , 偏移量 0x15C .....	1110
寄存器 147:	USB 端点 6 主机配置接收类型 ( USBRXTYPE6 ) , 偏移量 0x16C .....	1110
寄存器 148:	USB 端点 7 主机配置接收类型 ( USBRXTYPE7 ) , 偏移量 0x17C .....	1110
寄存器 149:	USB 端点 1 主机接收轮询间隔 ( USBRXINTERVAL1 ) , 偏移量 0x11D .....	1112
寄存器 150:	USB 端点 2 主机接收轮询间隔 ( USBRXINTERVAL2 ) , 偏移量 0x12D .....	1112
寄存器 151:	USB 端点 3 主机接收轮询间隔 ( USBRXINTERVAL3 ) , 偏移量 0x13D .....	1112
寄存器 152:	USB 端点 4 主机接收轮询间隔 ( USBRXINTERVAL4 ) , 偏移量 0x14D .....	1112
寄存器 153:	USB 端点 5 主机接收轮询间隔 ( USBRXINTERVAL5 ) , 偏移量 0x15D .....	1112
寄存器 154:	USB 端点 6 主机接收轮询间隔 ( USBRXINTERVAL6 ) , 偏移量 0x16D .....	1112
寄存器 155:	USB 端点 7 主机接收轮询间隔 ( USBRXINTERVAL7 ) , 偏移量 0x17D .....	1112
寄存器 156:	USB 端点 1 块传输中请求包数量 ( USBRQPKTCOUNT1 ) , 偏移量 0x304 .....	1113
寄存器 157:	USB 端点 2 块传输中请求包数量 ( USBRQPKTCOUNT2 ) , 偏移量 0x308 .....	1113
寄存器 158:	USB 端点 3 块传输中请求包数量 ( USBRQPKTCOUNT3 ) , 偏移量 0x30C .....	1113
寄存器 159:	USB 端点 4 块传输中请求包数量 ( USBRQPKTCOUNT4 ) , 偏移量 0x310 .....	1113
寄存器 160:	USB 端点 5 块传输中请求包数量 ( USBRQPKTCOUNT5 ) , 偏移量 0x314 .....	1113
寄存器 161:	USB 端点 6 块传输中请求包数量 ( USBRQPKTCOUNT6 ) , 偏移量 0x318 .....	1113
寄存器 162:	USB 端点 7 块传输中请求包数量 ( USBRQPKTCOUNT7 ) , 偏移量 0x31C .....	1113
寄存器 163:	USB 接收双包缓存禁用寄存器 ( USBRDPKTBUFDIS ) , 偏移量 0x340 .....	1114
寄存器 164:	USB 发送双包缓存禁用寄存器 ( USBTXDPKTBUFDIS ) , 偏移量 0x342 .....	1115
寄存器 165:	USB 外部电源控制寄存器 ( USBEPC ) , 偏移量 0x400 .....	1116
寄存器 166:	USB 外部电源控制原始中断状态寄存器 ( USBEPCRIS ) , 偏移量 0x404 .....	1119
寄存器 167:	USB 外部电源控制中断屏蔽寄存器 ( USBEPCIM ) , 偏移量 0x408 .....	1120
寄存器 168:	USB 外部电源控制中断状态和清除寄存器 ( USBEPCISC ) , 偏移量 0x40C .....	1121
寄存器 169:	USB 设备恢复原始中断状态寄存器 ( USBDRRIS ) , 偏移量 0x410 .....	1122
寄存器 170:	USB 设备恢复中断屏蔽寄存器 ( USBDRIM ) , 偏移量 0x414 .....	1123
寄存器 171:	USB 设备恢复中断状态和清除寄存器 ( USBDRISC ) , 偏移量 0x418 .....	1124
寄存器 172:	USB 通用控制和状态寄存器 ( USBGPCS ) , 偏移量 0x41C .....	1125
寄存器 173:	USB VBUS 浮动控制寄存器 ( USBVDC ) , 偏移量 0x430 .....	1126
寄存器 174:	USB VBUS 浮动控制原始中断状态寄存器 ( USBVDCRIS ) , 偏移量 0x434 .....	1127
寄存器 175:	USB VBUS 浮动控制中断屏蔽寄存器 ( USBVDCIM ) , 偏移量 0x438 .....	1128
寄存器 176:	USB VBUS 浮动控制中断状态和清除寄存器 ( USBVDCISC ) , 偏移量 0x43C .....	1129
寄存器 177:	USB ID 有效检测原始中断状态寄存器 ( USBIDVRIS ) , 偏移量 0x444 .....	1130
寄存器 178:	USB ID 有效检测中断屏蔽寄存器 ( USBIDVIM ) , 偏移量 0x448 .....	1131
寄存器 179:	USB ID 有效检测中断状态与清除寄存器 ( USBIDVISC ) , 偏移量 0x44C .....	1132
寄存器 180:	USB DMA 选择寄存器 ( USBDMASEL ) , 偏移量 0x450 .....	1133
寄存器 181:	USB 外设属性寄存器 ( USBPP ) , 偏移量 0xFC0 .....	1135
<b>模拟比较器</b>	<b>.....</b>	<b>1136</b>
寄存器 1:	模拟比较器屏蔽中断状态寄存器 ( ACMIS ) , 偏移量 0x00 .....	1142
寄存器 2:	模拟比较器原始中断状态寄存器 ( ACRIS ) , 偏移量 0x04 .....	1143
寄存器 3:	模拟比较器中断启用寄存器 ( ACINTEN ) , 偏移量 0x08 .....	1144
寄存器 4:	模拟比较器参考电压控制寄存器 ( ACREFCTL ) , 偏移量 0x010 .....	1145
寄存器 5:	模拟比较器状态寄存器 0 ( ACSTAT0 ) , 偏移量 0x020 .....	1146
寄存器 6:	模拟比较器状态寄存器 1 ( ACSTAT1 ) , 偏移量 0x040 .....	1146
寄存器 7:	模拟比较器状态寄存器 2 ( ACSTAT2 ) , 偏移量 0x060 .....	1146
寄存器 8:	模拟比较器控制寄存器 0 ( ACCTL0 ) , 偏移量 0x024 .....	1147
寄存器 9:	模拟比较器控制寄存器 1 ( ACCTL1 ) , 偏移量 0x044 .....	1147

寄存器 10:	模拟比较器控制寄存器 2 ( ACCTL2 ) , 偏移量 0x064 .....	1147
寄存器 11:	模拟比较器外设属性寄存器 ( ACMPPP ) , 偏移量 0xFC0 .....	1149
<b>脉宽调制器 (PWM)</b>	<b>.....</b>	<b>1151</b>
寄存器 1:	PWM 主机控制寄存器 ( PWMCTL ) , 偏移量 0x000 .....	1164
寄存器 2:	PWM 时间基准同步寄存器 ( PWMSYNC ) , 偏移量 0x004 .....	1166
寄存器 3:	PWM 输出使能寄存器 ( PWMENABLE ) , 偏移量 0x008 .....	1167
寄存器 4:	PWM 输出反相寄存器 ( PWMINVERT ) , 偏移量 0x00C .....	1169
寄存器 5:	PWM 输出故障寄存器 ( PWMFAULT ) , 偏移量 0x010 .....	1171
寄存器 6:	PWM 中断使能寄存器 ( PWMINTEN ) , 偏移量 0x014 .....	1173
寄存器 7:	PWM 原始中断状态寄存器 ( PWMRIS ) , 偏移量 0x018 .....	1175
寄存器 8:	PWM 中断状态和清除寄存器 ( PWMISC ) , 偏移量 0x01C .....	1177
寄存器 9:	PWM 状态寄存器 ( PWMSTATUS ) , 偏移量 0x020 .....	1179
寄存器 10:	PWM 故障条件值寄存器 ( PWMFAULTVAL ) , 偏移量 0x024 .....	1181
寄存器 11:	PWM 使能更新寄存器 ( PWMENUPD ) , 偏移量 0x028 .....	1183
寄存器 12:	PWM0 控制寄存器 ( PWM0CTL ) , 偏移量 0x040 .....	1186
寄存器 13:	PWM1 控制寄存器 ( PWM1CTL ) , 偏移量 0x080 .....	1186
寄存器 14:	PWM2 控制寄存器 ( PWM2CTL ) , 偏移量 0x0C0 .....	1186
寄存器 15:	PWM3 控制寄存器 ( PWM3CTL ) , 偏移量 0x100 .....	1186
寄存器 16:	PWM0 中断和触发使能寄存器 ( PWM0INTEN ) , 偏移量 0x044 .....	1190
寄存器 17:	PWM1 中断和触发使能寄存器 ( PWM1INTEN ) , 偏移量 0x084 .....	1190
寄存器 18:	PWM2 中断和触发使能寄存器 ( PWM2INTEN ) , 偏移量 0x0C4 .....	1190
寄存器 19:	PWM3 中断和触发使能寄存器 ( PWM3INTEN ) , 偏移量 0x104 .....	1190
寄存器 20:	PWM0 原始中断状态寄存器 ( PWM0RIS ) , 偏移量 0x048 .....	1193
寄存器 21:	PWM1 原始中断状态寄存器 ( PWM1RIS ) , 偏移量 0x088 .....	1193
寄存器 22:	PWM2 原始中断状态寄存器 ( PWM2RIS ) , 偏移量 0x0C8 .....	1193
寄存器 23:	PWM3 原始中断状态寄存器 ( PWM3RIS ) , 偏移量 0x108 .....	1193
寄存器 24:	PWM0 中断状态和清除寄存器 ( PWM0ISC ) , 偏移量 0x04C .....	1195
寄存器 25:	PWM1 中断状态和清除寄存器 ( PWM1ISC ) , 偏移量 0x08C .....	1195
寄存器 26:	PWM2 中断状态和清除寄存器 ( PWM2ISC ) , 偏移量 0x0CC .....	1195
寄存器 27:	PWM3 中断状态和清除寄存器 ( PWM3ISC ) , 偏移量 0x10C .....	1195
寄存器 28:	PWM0 装载寄存器 ( PWM0LOAD ) , 偏移量 0x050 .....	1197
寄存器 29:	PWM1 装载寄存器 ( PWM1LOAD ) , 偏移量 0x090 .....	1197
寄存器 30:	PWM2 装载寄存器 ( PWM2LOAD ) , 偏移量 0x0D0 .....	1197
寄存器 31:	PWM3 装载寄存器 ( PWM3LOAD ) , 偏移量 0x110 .....	1197
寄存器 32:	PWM0 计数寄存器 ( PWM0COUNT ) , 偏移量 0x054 .....	1198
寄存器 33:	PWM1 计数寄存器 ( PWM1COUNT ) , 偏移量 0x094 .....	1198
寄存器 34:	PWM2 计数寄存器 ( PWM2COUNT ) , 偏移量 0x0D4 .....	1198
寄存器 35:	PWM3 计数寄存器 ( PWM3COUNT ) , 偏移量 0x114 .....	1198
寄存器 36:	PWM0 比较寄存器 A ( PWM0CMPA ) , 偏移量 0x058 .....	1199
寄存器 37:	PWM1 比较寄存器 A ( PWM1CMPA ) , 偏移量 0x098 .....	1199
寄存器 38:	PWM2 比较寄存器 A ( PWM2CMPA ) , 偏移量 0x0D8 .....	1199
寄存器 39:	PWM3 比较寄存器 A ( PWM3CMPA ) , 偏移量 0x118 .....	1199
寄存器 40:	PWM0 比较寄存器 B ( PWM0CMPB ) , 偏移量 0x05C .....	1200
寄存器 41:	PWM1 比较寄存器 B ( PWM1CMPB ) , 偏移量 0x09C .....	1200
寄存器 42:	PWM2 比较寄存器 B ( PWM2CMPB ) , 偏移量 0x0DC .....	1200
寄存器 43:	PWM3 比较寄存器 B ( PWM3CMPB ) , 偏移量 0x11C .....	1200
寄存器 44:	PWM0 发生器 A 控制寄存器 ( PWM0GENA ) , 偏移量 0x060 .....	1201
寄存器 45:	PWM1 发生器 A 控制寄存器 ( PWM1GENA ) , 偏移量 0x0A0 .....	1201

寄存器 46:	PWM2 发生器 A 控制寄存器 ( PWM2GENA ) , 偏移量 0x0E0 .....	1201
寄存器 47:	PWM3 发生器 A 控制寄存器 ( PWM3GENA ) , 偏移量 0x120 .....	1201
寄存器 48:	PWM0 发生器 B 控制寄存器 ( PWM0GENB ) , 偏移量 0x064 .....	1204
寄存器 49:	PWM1 发生器 B 控制寄存器 ( PWM1GENB ) , 偏移量 0x0A4 .....	1204
寄存器 50:	PWM2 发生器 B 控制寄存器 ( PWM2GENB ) , 偏移量 0x0E4 .....	1204
寄存器 51:	PWM3 发生器 B 控制寄存器 ( PWM3GENB ) , 偏移量 0x124 .....	1204
寄存器 52:	PWM0 死区控制寄存器 ( PWM0DBCTL ) , 偏移量 0x068 .....	1206
寄存器 53:	PWM1 死区控制寄存器 ( PWM1DBCTL ) , 偏移量 0x0A8 .....	1206
寄存器 54:	PWM2 死区控制寄存器 ( PWM2DBCTL ) , 偏移量 0x0E8 .....	1206
寄存器 55:	PWM3 死区控制寄存器 ( PWM3DBCTL ) , 偏移量 0x128 .....	1206
寄存器 56:	PWM0 死区上升沿延迟寄存器 ( PWM0DBRISE ) , 偏移量 0x06C .....	1207
寄存器 57:	PWM1 死区上升沿延迟寄存器 ( PWM1DBRISE ) , 偏移量 0x0AC .....	1207
寄存器 58:	PWM2 死区上升沿延迟寄存器 ( PWM2DBRISE ) , 偏移量 0x0EC .....	1207
寄存器 59:	PWM3 死区上升沿延迟寄存器 ( PWM3DBRISE ) , 偏移量 0x12C .....	1207
寄存器 60:	PWM0 死区下降沿延迟寄存器 ( PWM0DBFALL ) , 偏移量 0x070 .....	1208
寄存器 61:	PWM1 死区下降沿延迟寄存器 ( PWM1DBFALL ) , 偏移量 0x0B0 .....	1208
寄存器 62:	PWM2 死区下降沿延迟寄存器 ( PWM2DBFALL ) , 偏移量 0x0F0 .....	1208
寄存器 63:	PWM3 死区下降沿延迟寄存器 ( PWM3DBFALL ) , 偏移量 0x130 .....	1208
寄存器 64:	PWM0 故障源寄存器 0 ( PWM0FLTSRC0 ) , 偏移量 0x074 .....	1209
寄存器 65:	PWM1 故障源寄存器 0 ( PWM1FLTSRC0 ) , 偏移量 0x0B4 .....	1209
寄存器 66:	PWM2 故障源寄存器 0 ( PWM2FLTSRC0 ) , 偏移量 0x0F4 .....	1209
寄存器 67:	PWM3 故障源寄存器 0 ( PWM3FLTSRC0 ) , 偏移量 0x134 .....	1209
寄存器 68:	PWM0 故障源寄存器 1 ( PWM0FLTSRC1 ) , 偏移量 1x078 .....	1211
寄存器 69:	PWM1 故障源寄存器 1 ( PWM1FLTSRC1 ) , 偏移量 0x0B8 .....	1211
寄存器 70:	PWM2 故障源寄存器 1 ( PWM2FLTSRC1 ) , 偏移量 0x0F8 .....	1211
寄存器 71:	PWM3 故障源寄存器 1 ( PWM3FLTSRC1 ) , 偏移量 0x138 .....	1211
寄存器 72:	PWM0 最小故障时间寄存器 ( PWM0MINFLTPER ) , 偏移量 0x07C .....	1214
寄存器 73:	PWM1 最小故障时间寄存器 ( PWM1MINFLTPER ) , 偏移量 0x0BC .....	1214
寄存器 74:	PWM2 最小故障时间寄存器 ( PWM2MINFLTPER ) , 偏移量 0x0FC .....	1214
寄存器 75:	PWM3 最小故障时间寄存器 ( PWM3MINFLTPER ) , 偏移量 0x13C .....	1214
寄存器 76:	PWM0 故障管脚逻辑感知寄存器 ( PWM0FLTSEN ) , 偏移量 0x800 .....	1215
寄存器 77:	PWM1 故障管脚逻辑感知寄存器 ( PWM1FLTSEN ) , 偏移量 0x880 .....	1215
寄存器 78:	PWM2 故障管脚逻辑感知寄存器 ( PWM2FLTSEN ) , 偏移量 0x900 .....	1215
寄存器 79:	PWM3 故障管脚逻辑感知寄存器 ( PWM3FLTSEN ) , 偏移量 0x980 .....	1215
寄存器 80:	PWM0 故障状态寄存器 0 ( PWM0FLTSTAT0 ) , 偏移量 0x804 .....	1216
寄存器 81:	PWM1 故障状态寄存器 0 ( PWM1FLTSTAT0 ) , 偏移量 0x884 .....	1216
寄存器 82:	PWM2 故障状态寄存器 0 ( PWM2FLTSTAT0 ) , 偏移量 0x904 .....	1216
寄存器 83:	PWM3 故障状态寄存器 0 ( PWM3FLTSTAT0 ) , 偏移量 0x984 .....	1216
寄存器 84:	PWM0 故障状态寄存器 1 ( PWM0FLTSTAT1 ) , 偏移量 1x808 .....	1218
寄存器 85:	PWM1 故障状态寄存器 1 ( PWM1FLTSTAT1 ) , 偏移量 1x888 .....	1218
寄存器 86:	PWM2 故障状态寄存器 1 ( PWM2FLTSTAT1 ) , 偏移量 0x908 .....	1218
寄存器 87:	PWM3 故障状态寄存器 1 ( PWM3FLTSTAT1 ) , 偏移量 0x988 .....	1218
寄存器 88:	PWM 外设属性寄存器 ( PWMPP ) , 偏移量 0xFC0 .....	1221
寄存器 89:	PWM 外设属性寄存器 ( PWMPC ) , 偏移量 0xFC4 .....	1223
<b>正交编码器接口 (QEI) .....</b>		<b>1224</b>
寄存器 1:	QEI 控制寄存器 ( QEICTL ) , 偏移量 0x000 .....	1230
寄存器 2:	QEI 状态寄存器 ( QEISTAT ) , 偏移量 0x004 .....	1233
寄存器 3:	QEI 位置寄存器 ( QEIPOS ) , 偏移量 0x008 .....	1234

---

寄存器 4:	QEI 最大位置寄存器 (QEIMAXPOS), 偏移量 0x00C .....	1235
寄存器 5:	QEI 定时器加载寄存器 (QEILOAD), 偏移量 0x010 .....	1236
寄存器 6:	QEI 定时器寄存器 (QEITIME), 偏移量 0x014 .....	1237
寄存器 7:	QEI 速度计数寄存器 (QEICOUNT), 偏移量 0x018 .....	1238
寄存器 8:	QEI 速度寄存器 (QEISPEED), 偏移量 0x01C .....	1239
寄存器 9:	QEI 中断启用寄存器 (QEIINTEN), 偏移量 0x020 .....	1240
寄存器 10:	QEI 原始中断状态寄存器 (QEIRIS), 偏移量 0x024 .....	1241
寄存器 11:	QEI 中断状态和清除寄存器 (QEIISC), 偏移量 0x028 .....	1242

# 修订历史

修订历史列表列出 LM4F232H5QD 数据手册各个版本之间的区别。

表 1. 修订历史

日期	版本号	描述
2012 年 4 月	12454.2480	<ul style="list-style-type: none"> <li>■ 文档版本号现在将自动生成，并表明源文件的修订版本。</li> <li>■ JTAG 章节中说明了“恢复锁定的微控制器”程序将擦除 EEPROM 并将其换位写入计数器恢复为出厂默认值。</li> <li>■ 在“休眠模块”章节中，修正了休眠 RTC 亚秒 (HIBRTCSS) 寄存器的复位值。</li> <li>■ “系统控制”章节的末尾部分将寄存器重新编排为组传统寄存器。</li> <li>■ 在 uDMA 章节的“<math>\mu</math>DMA 通道分配”和“所支持的请求类型”表中，将显示通用定时器单一请求的 <math>\mu</math>DMA 支持修正为显示通用定时器猝发请求的 <math>\mu</math>DMA 支持。</li> <li>■ 在 SSI 章节中，向 SSI 控制 1 (SSICR1) 寄存器添加了 SLBY6 位。这使系统时钟比 SSICLK 至少快 6 倍（而不是正常情况下的 12 倍）。</li> <li>■ 在 I<sup>2</sup>C 章节中，对快速指令的描述进行了说明。</li> <li>■ 在“工作特性”章节中，删除了“热学特性”表中的最大功耗参数。</li> <li>■ 在“电气特性”章节中： <ul style="list-style-type: none"> <li>– 删除了数据已经过审核的多个表中的挂起特性注释。</li> <li>– 在“电源特性”表中，调整了 POR 和 BOR 阈值的最小、标称和最大值。</li> <li>– 在“主振荡器输入特性”表中，调整了主振荡器启动时间的最大值。</li> <li>– 在“HIB 振荡器输入特性”表中，添加了振荡器启动时间的最大值，并更改了外部时钟基准占空比的最小和最大值。</li> <li>– 添加了晶振参数表。</li> <li>– 在“休眠模块电池特性”表中，调整了电池电压过低的检测值。</li> <li>– 在“休眠模块交流特性”表中，添加了占空比参数。</li> <li>– 在“Flash 存储器特性”表中，调整了 T<sub>PROG64</sub> 和 T<sub>ERASE</sub> 的最大值。</li> <li>– 在“GPIO 模块特性”表中，调整了 GPIO 上升和下降时间的值。</li> <li>– 在“ADC 电气特性”表中，更新了 ADC 输入共模电压参数的最大值，并调整了系统性能参数的值。</li> <li>– 在“SSI 特性”表中，调整了 SSIClk 上升和下降时间的值。</li> <li>– 在“初始电流消耗”表中，更新了深度睡眠模式参数的标称值。</li> </ul> </li> <li>■ 其它几处小的增补及改动。</li> </ul>
2012 年 3 月	12013	<ul style="list-style-type: none"> <li>■ 在“Cortex-M4 外设”章节中： <ul style="list-style-type: none"> <li>– 修正了中断 128-138 置位启用 (EN4)、中断 128-138 清除启用 (DIS4)、中断 128-138 置位挂起 (PEND4)、中断 128-138 清除挂起 (UNPEND4) 和中断 128-138 激活位 (ACTIVE4) 寄存器中的缺失位。</li> <li>– 添加了缺失的中断 132-135 优先级 (PRI33) 和中断 136-138 优先级 (PRI34) 寄存器。</li> </ul> </li> <li>■ 在“系统控制”章节中，</li> </ul>

表 1. 修订历史 (续)

日期	版本号	描述
		<ul style="list-style-type: none"> <li>- 修正了功率结构图。</li> <li>- 添加了“系统时钟的配置在 EEPROM 操作过程中不可更改”的注释。</li> <li>- 将器件标识 0 (DID0) 寄存器中第 27 位的复位修正为 1。</li> <li>- 由于此功能不存在，因此删除了那些无意义的位，从而澄清了软件复位 (SRx)、运行模式时钟门控控制 (RCGCx)、睡眠模式时钟门控控制 (SCGCx)、深度睡眠模式时钟门控控制 (DCGCx)、功率控制 (PCx) 和外设就绪 (PRx) 寄存器。相反地，外设存在 (PPx) 寄存器显示了适用于此系列中所有器件的位域，以 0 指示该功能不存在。</li> <li>■ 在“休眠”章节中： <ul style="list-style-type: none"> <li>- 修正了休眠模块结构图</li> <li>- 对 VDD3ON 模式的实现进行了说明。</li> <li>- 删除了休眠控制 (HIBCTL) 寄存器中的 LOWBATEN 位。</li> </ul> </li> <li>■ 在“定时器”章节中，添加了注释：如果启用了 PWM 输出反相，则边沿检测中断行为将反相。</li> <li>■ 在“看门狗定时器”章节中，向“初始化和配置”部分添加了关于看门狗定时器的维护信息。</li> <li>■ 在 ADC 章节中： <ul style="list-style-type: none"> <li>- 修正了“ADC 输入端等效结构图”</li> <li>- 将参考电压内部信号名称更改为 VREFP 和 VREFN。</li> <li>- 对“差分采样”部分进行了说明。</li> <li>- 修正了“内部温度传感器特性图”。</li> <li>- 修正了 ADC 触发源选择 (ADCTSSEL) 寄存器中 PSn 位域的位置。</li> <li>- 修正了 ADC 控制 (ADCCTL) 和 ADC 采样序列控制 3 (ADCSSCTL3) 寄存器的复位。</li> </ul> </li> <li>■ 在 UART 章节中： <ul style="list-style-type: none"> <li>- 对 UART LIN 定时器 (UARTLTIM) 寄存器添加了注释：如果 PIOSC 被用作 UART 波特时钟，则应对该寄存器中的值读取两次以确保数据正确。</li> <li>- 删除了 UART 9 位模式自身地址屏蔽 (UART9BITAMASK) 寄存器中的 RANGE 位。</li> <li>- 修正了 UART 时钟配置 (UARTCC) 寄存器中的 CS 位域值。</li> </ul> </li> <li>■ 在 SSI 章节中： <ul style="list-style-type: none"> <li>- 对初始化和配置部分的步骤进行了说明。</li> <li>- 修正了 SSI 时钟配置 (SSICC) 寄存器中的 CS 位域值。</li> </ul> </li> <li>■ 在 I<sup>2</sup>C 章节中： <ul style="list-style-type: none"> <li>- 对时钟低电平超时的操作进行了说明。</li> <li>- 添加了关于高速操作和超快模式的信息。</li> <li>- 修正了 I<sup>2</sup>C 主机总线监视 (I2CMBMON) 寄存器的复位。</li> </ul> </li> <li>■ 在 USB 章节中，修正了 USB 外设属性 (USBPP) 寄存器中 ECNT 位的复位。</li> <li>■ 在“模拟比较器”章节中： <ul style="list-style-type: none"> <li>- 重新编写了“内部参考编程”部分。</li> </ul> </li> </ul>

表 1. 修订历史 (续)

日期	版本号	描述
		<ul style="list-style-type: none"> <li>- 修正了模拟比较器参考电压控制 (ACREFCTL) 寄存器中 RNG 位的位描述。</li> <li>■ 在“信号表”章节中，删除了不应在此处出现的 LPC 信号。</li> <li>■ 在“电气特性”章节中：                         <ul style="list-style-type: none"> <li>- 对 GPIO 管脚操作和规格 (包括 ESD 保护、<math>V_{OH}</math>、<math>V_{OL}</math>、输入迟滞、输入漏电以及注入电流) 进行了说明。</li> <li>- 修正了 TCK 时钟低电平和高电平时间的标称值。</li> <li>- 对深度睡眠模式下的复位时序进行了说明。</li> <li>- 修正了 30 kHz 内部振荡器的频率范围。</li> <li>- 对休眠振荡器输入的 <math>V_{IH}</math>、<math>V_{IL}</math> 和输入迟滞进行了说明。</li> <li>- 将“休眠模块 AC 特性表”中 <math>T_{WAKE\_TO\_HIB}</math> 的单位值从微秒更改为休眠时钟。</li> <li>- 向“GPIO 模块特性表”添加了值。</li> <li>- 添加了使用内部参考时 ADC 的规格。</li> <li>- 添加了差分模式下 ADC 输入共模电压的规格。</li> <li>- 添加了 VDDA 的电流消耗规格。</li> </ul> </li> <li>■ 其它几处小的增补及改动。</li> </ul>
2011 年 11 月	11003	<ul style="list-style-type: none"> <li>■ 重新编排了“结构概述”章节。</li> <li>■ 在“系统控制”章节中：                         <ul style="list-style-type: none"> <li>- 修正了运行模式时钟门控控制寄存器 0 (RCGC0) 寄存器的复位值。</li> <li>- 修正了系统优先级 (SYSPROP) 寄存器的复位。</li> <li>- 由于 ROM 软件映射 (ROMSWMAP) 寄存器包含 TPSW 位的信息，因此将其从非易失性存储器信息 (NVMSTAT) 寄存器中删除。</li> </ul> </li> <li>■ 更改了系统异常原始中断状态 (SYSEXCRIS)、系统异常中断屏蔽 (SYSEXCIM)、系统异常屏蔽的中断状态 (SYSEXCIMIS) 和系统异常中断清除 (SYSEXCIC) 寄存器中的位名称，以表明它们用于浮点异常。</li> <li>■ 在“休眠”章节中，添加了“仲裁电源关断”部分，修正了“使用专用振荡器作为带 VDD3ON 模式的休眠时钟源图”。</li> <li>■ 在“内部存储器”章节中，对非易失性寄存器的编程和使用进行了说明，包括对启动配置 (BOOTCFG) 和用户寄存器 n (USER_REGn) 寄存器的修正。</li> <li>■ 在 GPIO 章节中，修正了“具有非 0 复位值的 GPIO 管脚表”。</li> <li>■ 在“通用定时器”章节中，添加了对定时器操作的说明。</li> <li>■ 在 UART 章节中，对中断行为进行了说明。</li> <li>■ 在 I<sup>2</sup>C 章节中：                         <ul style="list-style-type: none"> <li>- 添加了超快 (1 Mbps) 模式和高速 (3.33 Mbps) 模式的内容，修正了器件功能 2 (DC2)、I<sup>2</sup>C 主机控制/状态 (I2CMCS) 和 I<sup>2</sup>C 外设属性 (I2CPP) 寄存器的复位值。</li> <li>- 修正了 I<sup>2</sup>C 主机控制/状态 (I2CMCS) 寄存器的复位。</li> <li>- 将 HSTPR 位添加到 I<sup>2</sup>C 主机定时器周期 (I2CMTPR) 寄存器。</li> <li>- 添加了 I<sup>2</sup>C 外设配置 (I2CPC) 寄存器。</li> </ul> </li> <li>■ 在 USB 章节中：                         <ul style="list-style-type: none"> <li>- 修正了 USB 设备恢复中断屏蔽 (USBDRIM) 寄存器的描述。</li> </ul> </li> </ul>



表 1. 修订历史 (续)

日期	版本号	描述
		<ul style="list-style-type: none"><li>■ 在“模拟比较器”章节中：<ul style="list-style-type: none"><li>– 修正了“内部参考电压和 ACREFTL 域值表”。</li><li>– 修正了模拟比较器外设属性 (ACMPPP) 寄存器中的位域。</li></ul></li><li>■ 在“电气特性”章节中：<ul style="list-style-type: none"><li>– 对负载电容公式进行了说明。</li><li>– 修正了“模拟比较器参考电压特性表”中的值。</li></ul></li><li>■ 其它几处小的增补及改动。</li></ul>
2011 年 9 月	10502	开始记录修订历史。

## 关于本文档

该数据手册提供 LM4F232H5QD 微控制器的参考信息，描述了围绕 ARM® Cortex™-M4F 内核设计的片上系统 (SoC) 器件的功能模块。

### 读者

本手册的受众是系统软件开发人员、硬件设计人员和应用设计人员。

### 关于本手册

本文档分成若干章节，每个章节与主要功能相对应。

### 相关文档

相关文档可从以下网站获取：Stellaris® 网址 [www.ti.com/stellaris](http://www.ti.com/stellaris)：

- “Stellaris® 勘误表”
- “ARM® Cortex™-M4 勘误表”
- “Cortex™-M3/M4 Instruction Set 技术用户手册”
- “Stellaris® Boot Loader 用户指南”
- “Stellaris® Graphics Library 用户指南”
- “Stellaris® Peripheral Driver Library 用户指南”
- “Stellaris® ROM 用户指南”
- “Stellaris® USB Library 用户指南”

本数据手册也可参考下列相关文档：

- “ARM® Debug Interface V5 架构规格”
- “ARM® Embedded Trace Macrocell 架构规格”
- “IEEE Standard 1149.1-Test Access Port and Boundary-Scan Architecture”

这个文献列表的实时性根据发布的日期来判断。其他文献（包括应用说明和白皮书），请访问网站获取。

## 文档约定

本文档使用的约定如表 2 在 43 页所示。

表 2. 文档约定

表示法	含义
通用寄存器的表示法	
REGISTER	APB 寄存器用大写的粗体表示。例如，PBORCTL 是上电和掉电复位控制寄存器。如果一个寄存器名称包含一个小写的 n，它就代表了多个寄存器。例如，SRCRn 代表了下面三个软件复位控制寄存器的任何一个或全部：SRCR0、SRCR1 和 SRCR2。
位	寄存器的一个位。
位域	两个或更多连续和相关联的位。
偏移量 0xnnn	寄存器地址的一个十六进制增量，增量是相对表 2-4 在 91 页中指定的模块基址而言的。
寄存器 N	为了方便引用，在整篇文档中，寄存器被顺序编号。寄存器编号对软件没有意义。
保留	标注为“保留”的寄存器位保留下来供将来使用在大多数情况下，保留位被设置成 0；但是，用户软件不应当依赖保留位的值。为了使软件兼容未来的器件，保留位的值应当在读-修改-写过程中保留下来。
yy:xx	寄存器位的范围从 xx 到 yy (xx 和 yy 包括在内)。例如，31:15 表示相应寄存器的位 15 到 31。
寄存器位/域类型	寄存器位框图中的这个值指明了在控制器上运行的软件是否能改变位域的值。
RC	软件可以读取这个域。位/域在被读取之后由硬件清零。
RO	软件可以读取这个域。始终写入芯片复位值。
R/W	软件可以读或写此域。
R/WC	软件可以读或写此域。向该寄存器写入任意值都会将寄存器清零。
R/W1C	软件可以读或写此域。向 W1C 位写入 0 不影响寄存器中的位值。写 1 清零寄存器中位的值；剩余的位保持变。 此寄存器类型主要用来清零中断状态位，执行读操作来提供中断状态，写入读取的值只清除在读寄存器时被报告的中断。
R/W1S	软件可以读或向此域写 1。向 R/W1S 位写 0 不影响寄存器中的位值。
W1C	软件可以写这个域。向 W1C 位写入 0 不影响寄存器中的位值。写 1 清零寄存器中位的值；剩余的位保持变。读寄存器返回的数据没有意义。 此寄存器通常用来清零中断寄存器中相应的位。
WO	只有软件的写操作才有效；读寄存器返回的数据没有意义。
寄存器位/域复位值	寄存器位框图中的这个值是任何复位后的位/域值，但特别注明的除外。
0	芯片复位时位被清零。
1	芯片复位时位被置位。
-	不确定。
管脚/信号表示法	
[ ]	管脚复用功能；管脚默认用作不带方括号的信号。
管脚	参考封装上的物理连接。
信号	参考管脚的电气信号编码。
使一个信号有效	将信号的值从逻辑假状态变为逻辑真状态对于高电平有效的信号，有效的信号值为 1 (高)；对于低电平有效的信号，有效的信号值为 0 (低)。有效极性 (高或低) 由信号名称来定义 (见下面的 SIGNAL 和 $\overline{\text{SIGNAL}}$ )。
使一个信号无效	将信号的值从逻辑真状态变为逻辑假状态。
SIGNAL	信号名称采用大写字母，使用 Courier 字体。信号名称带有上横线表示该信号低有效。使 $\overline{\text{SIGNAL}}$ 有效就将其驱动为低电平；使 SIGNAL 无效就将其驱动为高电平。

表 2. 文档约定 ( 续 )

表示法	含义
SIGNAL	信号名称采用大写字母，使用 Courier 字体。高有效的信号没有上横线。使 SIGNAL 有效就将其驱动为高电平；使 SIGNAL 无效就将其驱动为低电平。
数值	
X	大写的 X 表示几个值都是可取的，此处的 X 可以是任何合法的样式。例如，二进制值 0X00 可以是 0100 或 0000，十六进制值 0xX 可以是 0x0 或 0x1，等等。
0x	十六进制值带有前缀 0x。例如，0x00FF 是十六进制值 FF。 寄存器表中的所有其它数假设为二进制。在概念信息中，二进制数用一个 b 后缀表示，例如，1011b，写十进制数无需前缀或后缀。

# 1 结构概述

Texas Instruments 作为行业领导者，为微控制器市场带来了 32 位功能的基于 ARM® Cortex™-M 的微控制器。对于当前 8 位和 16 位 MCU 的用户而言，配合使用 Stellaris® 和 Cortex-M 即可获得业界功能最强大的开发工具、软件和知识生态系统。迁移到 Stellaris 后，设计者将受益于其强大的工具、简洁的代码编写方式以及突出的性能。更重要的是，设计者完全有信心进入具有从 1 美金到 1 GHz 的兼容产品的 ARM 生态系统。Thumb-2 技术可以使 16 位和 32 位指令并存，带来了代码密度和性能的最佳平衡。Thumb-2 比纯 32 位代码占用少 26%，同时带来了 25% 的性能提升，可有效降低系统成本。该 Texas Instruments Stellaris 微控制器系列是，为注重成本的嵌入式微控制器应用提供了高性能的 32 位运算能力。

本章给出了 Stellaris LM4F 系列微控制器的概述以及 LM4F232H5QD 微控制器的细节：

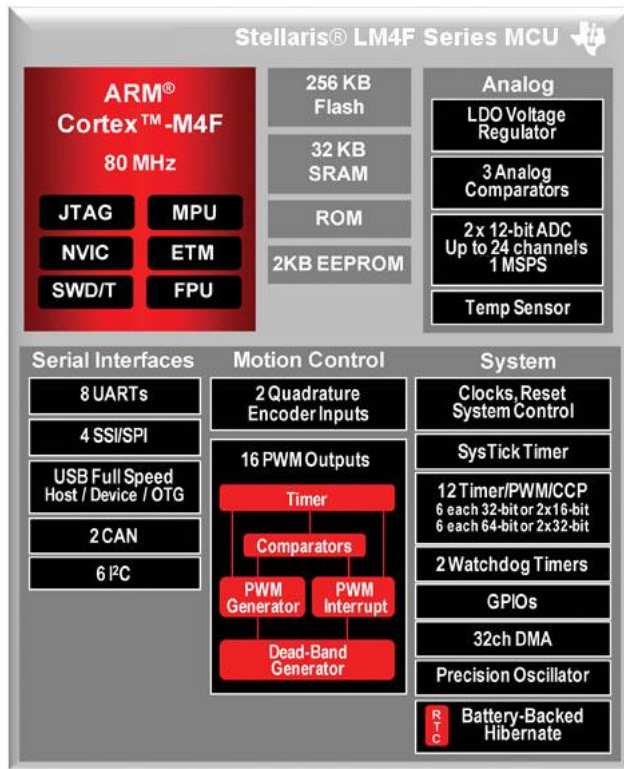
- “Stellaris LM4F 系列概述” 在 45 页
- “LM4F232H5QD 微控制器概述” 在 48 页
- “LM4F232H5QD 微控制器特性” 在 51 页
- “LM4F232H5QD 微控制器硬件细节” 在 68 页

## 1.1 Stellaris LM4F 系列概述

该 Stellaris LM4F 系列隶属 ARM Cortex-M4 微控制器，具有顶级性能和高级集成特性。本产品系列适用于需要控制成本，同时又需要大量控制和连接能力的应用，例如：

- 低功耗手持智能设备
- 游戏设备
- 家用和商用监控
- 运动控制
- 医疗器械
- 测试和测量仪器
- 工厂自动化
- 火警和安防用具
- 智能能源/智能电网解决方案
- 智能照明控制
- 运输业

图 1-1. Stellaris LM4F 结构图



Stellaris LM4F 微控制器系列包括十五个管脚兼容的器件，概述如下。每个系列嵌入的 Flash 和 SRAM 大小都各不相同。

表 1-1. Stellaris LM4F 器件系列

通用 MCU ( LM4F11x 系列 )	通用 MCU + USB 设备 ( LM4F12x 系列 )	通用 MCU + USB OTG ( LM4F13x 系列 )	运动控制 ( LM4F21x 系列 )	运动控制 + USB OTG ( LM4F23x 系列 )	封装
LM4F110	LM4F120	LM4F130	LM4F210	LM4F230	64 管脚 LQFP
LM4F111	LM4F121	LM4F131	LM4F211	LM4F231	64 管脚 LQFP
LM4F112	LM4F122	LM4F132	LM4F212	LM4F232	100 管脚 LQFP 144 管脚 LQFP 157 焊珠 BGA ( 仅限 LM4F212 和 LM4F232 )

表 1-2. Stellaris LM4F 系列器件

器件号	Flash (KB)	SRAM (KB)	带备用电池的休眠模块	PWM	PWM 故障	QEI 通道	CAN MAC	USB <sup>a</sup>	UART	UART 调制解调器信号	I <sup>2</sup> C	SSI/SPI	ADC 通道	ADC 外部参考	模拟/数字比较器	耐 5 V 电压 GPIO <sup>b</sup>	封装
LM4F110B2QR	32	12	✓	-	-	-	1	-	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F110C4QR	64	24	✓	-	-	-	1	-	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F110E5QR	128	32	✓	-	-	-	1	-	8	-	4	4	12	-	2/16	0-43	64LQFP

表 1-2. Stellaris LM4F 系列器件 (续)

器件号	Flash (KB)	SRAM (KB)	带备用电池的休眠模块	PWM	PWM 故障	QEI 通道	CAN MAC	USB <sup>a</sup>	UART	UART 调制解调器/信号	I <sup>2</sup> C	SSI/SPI	ADC 通道	ADC 外部参考	模拟/数字比较器	耐 5 V 电压 GPIO <sup>b</sup>	封装
LM4F110H5QR	256	32	✓	-	-	-	1	-	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F111B2QR	32	12	-	-	-	-	1	-	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F111C4QR	64	24	-	-	-	-	1	-	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F111E5QR	128	32	-	-	-	-	1	-	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F111H5QR	256	32	-	-	-	-	1	-	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F112C4QC	64	24	✓	-	-	-	1	-	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F112E5QC	128	32	✓	-	-	-	1	-	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F112H5QC	256	32	✓	-	-	-	1	-	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F112H5QD	256	32	✓	-	-	-	1	-	8	✓	6	4	24	✓	3/16	0-105	144LQFP
LM4F120B2QR	32	12	✓	-	-	-	1	D	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F120C4QR	64	24	✓	-	-	-	1	D	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F120E5QR	128	32	✓	-	-	-	1	D	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F120H5QR	256	32	✓	-	-	-	1	D	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F121B2QR	32	12	-	-	-	-	1	D	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F121C4QR	64	24	-	-	-	-	1	D	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F121E5QR	128	32	-	-	-	-	1	D	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F121H5QR	256	32	-	-	-	-	1	D	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F122C4QC	64	24	✓	-	-	-	1	D	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F122E5QC	128	32	✓	-	-	-	1	D	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F122H5QC	256	32	✓	-	-	-	1	D	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F122H5QD	256	32	✓	-	-	-	1	D	8	✓	6	4	24	✓	3/16	0-105	144LQFP
LM4F130C4QR	64	24	✓	-	-	-	1	O	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F130E5QR	128	32	✓	-	-	-	1	O	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F130H5QR	256	32	✓	-	-	-	1	O	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F131C4QR	64	24	-	-	-	-	1	O	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F131E5QR	128	32	-	-	-	-	1	O	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F131H5QR	256	32	-	-	-	-	1	O	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F132C4QC	64	24	✓	-	-	-	1	O	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F132E5QC	128	32	✓	-	-	-	1	O	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F132H5QC	256	32	✓	-	-	-	1	O	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F132H5QD	256	32	✓	-	-	-	1	O	8	✓	6	4	24	✓	3/16	0-105	144LQFP
LM4F210E5QR	128	32	✓	16	2	2	2	-	8	✓	4	4	12	-	2/16	0-43	64LQFP
LM4F210H5QR	256	32	✓	16	2	2	2	-	8	✓	4	4	12	-	2/16	0-43	64LQFP
LM4F211E5QR	128	32	-	16	6	2	2	-	8	✓	6	4	12	-	2/16	0-49	64LQFP
LM4F211H5QR	256	32	-	16	6	2	2	-	8	✓	6	4	12	-	2/16	0-49	64LQFP
LM4F212E5QC	128	32	✓	16	8	2	2	-	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F212H5BB	256	32	✓	16	8	2	2	-	8	✓	6	4	24	✓	3/16	0-120	157BGA
LM4F212H5QC	256	32	✓	16	8	2	2	-	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F212H5QD	256	32	✓	16	8	2	2	-	8	✓	6	4	24	✓	3/16	0-105	144LQFP

表 1-2. Stellaris LM4F 系列器件 (续)

器件号	Flash (KB)	SRAM (KB)	带备用电池的休眠模块	PWM	PWM 故障	QEI 通道	CAN MAC	USB <sup>a</sup>	UART	UART 调制解调器/信号	I <sup>2</sup> C	SSI/SPI	ADC 通道	ADC 外部参考	模拟/数字比较器	耐 5 V 电压 GPIO <sup>b</sup>	封装
LM4F230E5QR	128	32	✓	16	2	2	2	O	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F230H5QR	256	32	✓	16	2	2	2	O	8	-	4	4	12	-	2/16	0-43	64LQFP
LM4F231E5QR	128	32	-	16	6	2	2	O	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F231H5QR	256	32	-	16	6	2	2	O	8	-	6	4	12	-	2/16	0-49	64LQFP
LM4F232E5QC	128	32	✓	16	8	2	2	O	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F232H5BB	256	32	✓	16	8	2	2	O	8	✓	6	4	24	✓	3/16	0-120	157BGA
LM4F232H5QC	256	32	✓	16	8	2	2	O	8	✓	6	4	22	✓	3/16	0-69	100LQFP
LM4F232H5QD	256	32	✓	16	8	2	2	O	8	✓	6	4	24	✓	3/16	0-105	144LQFP

a. Stellaris 微控制器的 USB 选项包括仅用作设备 (D) 功能、主机/设备 (H) 功能以及 OTG/主机/设备 (O) 功能。

b. 最少的 GPIO 专用管脚数量；如果不使用特定外设，可用管脚会更多。详细信息请参考数据手册。

## 1.2 LM4F232H5QD 微控制器概述

该 Stellaris LM4F232H5QD 微控制器结合了复杂集成功能和高性能，并具备如表 1-3 中所示的各种特性。

表 1-3. Stellaris LM4F232H5QD 微控制器特性

特性	描述
内核	ARM Cortex-M4F 处理器内核
性能	80-MHz 运行速度；100 DMIPS 性能
Flash	256 KB 单周期 Flash 存储器
系统 SRAM	32 KB 单周期 SRAM
内置 ROM	装有 StellarisWare <sup>®</sup> 软件包的内置 ROM：
通信接口	
通用异步收发器 (UART)	八个 UART
同步串行接口 (SSI)	四个 SSI 模块
内部集成电路 (I <sup>2</sup> C)	六个 I <sup>2</sup> C 模块，具有四种传输速率（包括高速模式）
控制器局域网 (CAN)	两个 CAN 2.0 A/B 控制器
通用串行总线 (USB)	USB 2.0 OTG/主机/设备
系统集成	
微型直接存储器访问 (μDMA)	ARM <sup>®</sup> PrimeCell <sup>®</sup> 32 通道的可配置 μDMA 控制器
通用定时器 (GPTM)	六个 16/32 位 GPTM 模块和六个 32/64 位宽 GPTM 模块
看门狗定时器 (WDT)	2 个看门狗定时器
休眠	带备用电池的低功耗休眠模块
通用输入/输出端口 (GPIO)	14 个物理 GPIO 模块
高级运动控制	
脉宽调制器 (PWM)	2 个 PWM 模块
正交编码器接口 (QEI)	2 个 QEI 模块
模拟支持	
模数转换器 (ADC)	两个 12 位 ADC 模块

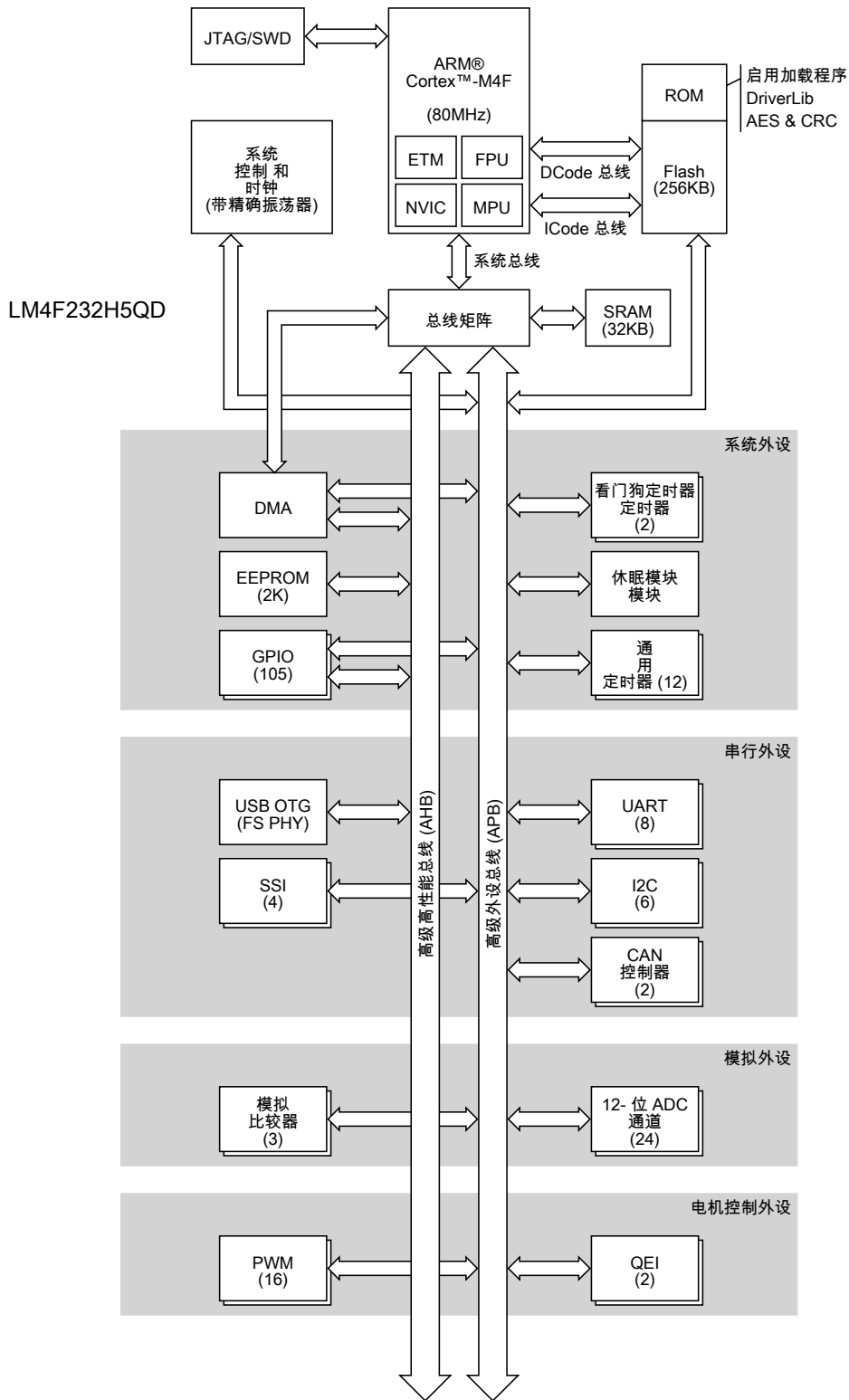


表 1-3. Stellaris LM4F232H5QD 微控制器特性 ( 续 )

特性	描述
模拟比较器控制器	三个独立集成的模拟比较器
数字比较器	16 个数字比较器
JTAG 和串行线调试 (SWD)	一个 JTAG 模块，带集成的 ARM SWD
封装	144 管脚 LQFP
工作范围	工业 ( -40°C 到 85°C ) 温度范围

图 1-2 在 50 页 显示了 Stellaris LM4F232H5QD 微控制器的特性。请注意有两条片内总线连接内核和外设。高级外设总线 (APB) 是旧系统的总线，高级高性能总线 (AHB) 提供比 APB 总线更好的背靠背的访问性能。

图 1-2. Stellaris LM4F232H5QD 微控制器高级框图



至于那些对功耗有特别要求的应用方案，LM4F232H5QD 微控制器还具有一个带备用电池的休眠模块，从而有效地使 LM4F232H5QD 在长时间未被激活时进入低功耗状态。一个上电/掉电序列发生器、连续的时间计数器 (RTC)、一对匹配寄存器、一个到系统总线的 APB 接口以及专用的带备用电池的存储器、休眠模块等功能组件使 LM4F232H5QD 微控制器极其适合于电池应用。

另外，LM4F232H5QD 微控制器的优势还在于能够方便的运用多种 ARM 的开发工具和片上系统 (SoC) 的底层 IP 应用方案，以及广大的用户群体。另外，该微控制器使用了兼容 ARM 的 Thumb® 指令集的 Thumb2 指令集来减少存储容量的需求，并以此达到降低成本的目的。最后，LM4F232H5QD 微控制器与各个 Stellaris 系列的所有成员是代码兼容的，这为适应各种精确的需求提供了灵活性。

Texas Instruments 为了能够帮助用户产品快速的上市，提供了一整套的解决方案，包括评估和开发的板卡、白皮书和应用笔记、方便使用的外设驱动程序库、以及强劲的支持、销售和分销网络。

## 1.3 LM4F232H5QD 微控制器特性

有关 LM4F232H5QD 微控制器元件特性和通用功能的更多详细信息请参考下面的章节。

### 1.3.1 ARM Cortex-M4F 处理器内核

Stellaris 产品系列的® 所有成员 (包括 LM4F232H5QD 微控制器) 都是围绕 ARM Cortex-M 处理器内核设计。该 ARM Cortex-M 处理器为高性能、低成本的平台提供一个满足小存储要求解决方案 (minimal memory implementation)、简化管脚数以及低功耗三方面要求的内核，与此同时，它还提供出色的计算性能和卓越的系统中断响应能力。

#### 1.3.1.1 处理器内核 (见第 69 页)

- 32 位的 ARM Cortex-M4F 架构优化了小封装嵌入式应用方案。
- 80-MHz 运行速度；100 DMIPS 性能
- 优越的处理性能和更快的中断处理。
- 混合 16 位/32 位的 Thumb-2 指令集提供与 32 位 ARM 内核所期望的高性能而采用了更紧凑的内存大小，而这通常在 8 位和 16 位设备相关的存储容量中，特别是在微控制器级应用的几千字节存储中。
  - 单周期乘法指令和硬件除法器
  - 精确的位操作 (bit-banding)，不仅最大限度的利用了存储器空间而且还改良了对外设的控制
  - 非对齐式数据访问，使数据能够更为有效的安置到存储器中
- 符合 IEEE754 的单精度浮点单元 (FPU)
- 16 位 SIMD 矢量处理单元
- 快速代码执行允许更低的处理器时钟和增加休眠模式时间
- Harvard 结构 - 将数据和指令所使用的总线进行了分离
- 高效的处理器内核，系统和存储器
- 硬件除法和高速数字信号处理专用乘加
- 用于信号处理的饱和算法

- 对时间苛刻的应用提供可确定的, 高性能的中断处理
- 存储器保护单元为操作系统机能提供特权操作模式
- 增强的系统调试提供全方位的断点和跟踪能力
- 串行线调试和串行线跟踪减少调试和跟踪过程中需求的管脚数
- 从 ARM7 处理器系列中移植过来, 以获得更好的性能和电源效率
- 优化的单周期 Flash 存储器使用
- 集成多种休眠模式, 更低功耗

### 1.3.1.2 系统定时器 (SysTick) (见第 118 页)

ARM Cortex-M4F 含有集成的系统定时器 SysTick。SysTick 提供了一种简单的、24 位写 1 清 0 (clear-on-write)、递减的、自加载 (wrap-on-zero) 的计数器, 同时具有灵活的控制机制。该计数器可以在几种不同的场合中使用, 例如:

- 用作 RTOS 的节拍定时器, 按照可编程的频率 (例如, 100Hz) 定时触发, 调用系统定时器服务子程序
- 用作高速报警定时器, 采用系统时钟作为时钟源
- 用作频率可变的报警或信号定时器——其周期取决于所采用的参考时钟源以及计数器的动态范围
- 用作简单计数器, 测量任务的完成时刻、总体耗时等等
- 基于未到达/到达时间的内部时钟源控制。

### 1.3.1.3 嵌套向量中断控制器 (NVIC) (见第 119 页)

LM4F232H5QD 控制器包含 ARM 嵌套向量中断控制器 (NVIC)。NVIC 和 Cortex-M4F 可以在处理模式中对所有的异常进行优先级划分并进行处理异常发生时处理器状态被自动存储到堆栈, 中断服务程序 (ISR) 结束时又自动被恢复。向量的读取与状态保存并行, 高效率进入中断。处理器支持尾链 (tail-chaining), 这样使得执行背靠背中断不需要重叠的状态保存和恢复。可通过软件设置 7 个异常 (系统处理) 和 96 个中断的 8 级优先级。

- 确定性的, 快速中断处理: 总是 12 周期或尾链时 6 周期
- 外部不可屏蔽中断信号 (NMI) 可用于安全关键应用
- 动态优先级中断
- 通过硬件实现所需的寄存器可灵活的处理特殊的中断

### 1.3.1.4 系统控制模块 (SCB) (见第 120 页)

系统控制块 (SCB) 提供系统实现信息和系统控制, 包括系统异常的配置, 控制和报告。

### 1.3.1.5 存储器保护单元 (MPU) (见第 121 页)

MPU 支持标准的 ARM7 受保护的内存系统架构 (PMSA) 模型。MPU 提供保护区支持, 重叠保护区, 访问权限和导出存储属性到系统。

### 1.3.1.6 浮点单元 (FPU) ( 见第 125 页 )

该 FPU 完全支持单精度的加法、减法、乘法、除法、乘加和平方根运算。它也提供定点数据和浮点数据之间的格式转换，以及浮点常数的指令。

- 使用 32 位指令进行单精度 ( C 浮点 ) 数据处理操作
- 结合乘加指令，提高了精度 ( 熔合 MAC )
- 硬件支持转换、加、减、可选累加的乘法、除以及平方根
- 硬件支持反常与所有 IEEE 舍入模式。
- 32 个专用的 32 位单精度寄存器，也可作为 16 位双字寄存器进行寻址
- 去耦三级流水线

### 1.3.2 片上存储器

LM4F232H5QD 微控制器整合了以下片上存储器和特性：

- 32 KB 单周期访问的 SRAM
- 256 KB 单周期 Flash 存储器，
- 装有 StellarisWare 软件包的内部 ROM：
  - Stellaris 外设驱动库
  - Stellaris 引导加载程序
  - 高级加密标准 (AES) 密码表
  - 循环冗余检验 (CRC) 错误检测功能

#### 1.3.2.1 SRAM ( 见第 504 页 )

LM4F232H5QD 微控制器提供 32 KB 单周期片上 SRAM。Stellaris 器件的内部产品系列的 位于器件存储器映射的偏移量 0x2000.0000 处。

由于读-修改-写 (RMW) 操作是很耗时的，因此 ARM 引入了位带 (*bit-banding*) 技术 ( 在 Cortex-M4F 处理器中 )。在位带启用的处理器中，存储器映射的特定区域 (SRAM 和外设空间) 能够使用地址别名，在单个原子操作中访问各个位。

使用微直接内存访问控制器 (μDMA) 可以向片上 SRAM 传输数据，也可以从片上 SRAM 接收数据。

#### 1.3.2.2 Flash 存储器 ( 见第 506 页 )

LM4F232H5QD 微控制器提供 256 KB 单周期片上 Flash 存储器。Flash 存储器由一系列 1 KB 的块组织在一起，这些块可以被单独擦除。擦除一个块将使块中的所有位变为 1。这些区块配对后便组成了一组可分别进行保护的 2 KB 区块。该保护允许块被标记为只读或只执行，以提供不同等级的代码保护。只读块不能被擦除或编程，块的内容受保护不能修改。只执行块不能被擦除或编程，只能通过控制器取指机制来读取它的内容，块的内容受保护不能被控制器或调试器读取。

#### 1.3.2.3 ROM ( 见第 504 页 )

LM4F232H5QD ROM 出厂时经过预编程，包含了下面的软件和程序：

- Stellaris 外设驱动库
- Stellaris 引导加载程序

- 高级加密标准 (AES) 密码表
- 循环冗余检验 (CRC) 错误检测功能

该 Stellaris 外设驱动库是具有引导加载能力的用于控制片上外设的免版权的软件库。该库可用于外设初始化和控制功能，可选择轮询式和中断驱动式外设支持。另外，驱动库充分利用 ARM Cortex-M4F 内核的中断性能的全部优点。不需要特殊的程序或者用户汇编代码开端/收尾功能。在需要现场编程能力的应用中，免版权的 Stellaris 引导加载程序可用于应用程序加载，支持现场固件升级。

高级加密标准 (AES) 是美国政府使用的公开定义的加密标准。AES 是一种强大的加密方法，拥有不错的性能和大小。AES 在硬件和软件方面都很快，它非常容易使用，并且只需要很少的存储空间。该 Texas Instruments 加密数据包开放了所有源代码，建立在宽通用许可证 (LGPL) 源代码基础之上。LGPL 是指可在应用中使用，但不受任何 copyleft 的影响的代码（代码并不会自动变成开源代码）。然而对于该数据包源的修改必须是开源的。

循环冗余检查 (CRC) 是一项用来确认一段数据的内容与先前检验的数据是否相同的技术。CRC 技术可用于确认信息的正确接收（在传送中没有丢失或改变），用来确认解压后的数据，用来证实 Flash 存储器的内容没有更改，以及其它数据需要被确认的情况。CRC 优于简单的校验和（例如异或所有的位），因为它更容易捕捉到变化。

### 1.3.3 串行通讯外设

LM4F232H5QD 控制器支持同步和异步串行通信：

- 两个 CAN 2.0 A/B 控制器
- USB 2.0 OTG/主机/设备
- 八个支持 IrDA、9 位以及 ISO 7816 的 UART（其中一个带调制解调器流量控制和状态的 UART）
- 六个 I<sup>2</sup>C 模块，具有四种传输速率（包括高速模式）
- 四个同步串行接口模块 (SSI)

下面的章节各个通讯功能的提供更多细节。

#### 1.3.3.1 控制器局域网（见第 979 页）

控制器局域网 (CAN) 是一种用于连接电子控制设备 (Electronic Control Unit, 简称为 ECU) 的多播共享型串行总线标准。CAN 总线针对抗电磁干扰进行了专门设计，适用于具有较强电磁干扰的环境，不但可以使用与 RS-485 类似的差分平衡传输线，也可以使用更加可靠的双绞线。CAN 总线最初是针对汽车应用而研发的，不过时至今日已经广泛应用于各种嵌入式控制领域（例如工业方面和医疗方面）。CAN 总线在总线长度小于 40 米时最高可达 1 Mbps 位速率。位速率越低则有效通讯距离越远（例如 125 kbps 时通讯距离可达 500 米）。

发送器向所有 CAN 结点发送一条报文（广播）。每个节点根据接收到的标识符来判断是否处理该报文。标识符还决定了总线访问竞争中报文享有的优先级。每个 CAN 报文都能够传输 0 到 8 个字节的用户信息。

LM4F232H5QD 微控制器包括 2 个 CAN 单元，有如下特性：

- 支持 CAN 总线协议 2.0 A/B
- 位速率最高可达 1 Mbps
- 32 个报文对象，每个报文对象都具有独立的标识符掩码

- 可屏蔽中断
- 支持禁用自动重新发送 ( Disable Automatic Retransmission, 简称为 DAR ) 模式, 因此可用于时间触发 CAN ( Time Triggered CAN, 简称为 TTCAN ) 应用
- 自测试操作具有可编程的回送模式
- 可编程的FIFO模式, 能存储多个报文对象
- 提供 CANnTX、CANnRX 管脚, 可无缝连接片外 CAN 收发器。

### 1.3.3.2 USB ( 见第 1026 页 )

通用串行总线 (USB) 是一种串行总线标准, 可以在不重启系统的情况下, 通过标准接口连接和断开外设。

LM4F232H5QD 微控制器USB 设备、USB 主机和 USB OTG ( 当连接到其它已启用 USB 的系统上时, 协商决定用作主机还是设备 )

USB 模块特性 :

- 符合 USB-IF 认证标准
- 支持 USB 2.0 全速模式 (12 Mbps) 和低速模式 (1.5 Mbps)
- 四种传输类型 : 控制传输、中断传输、批次传输和等时传输
- 16 个端点
  - 一个专用的输入控制端点和一个专用输出控制端点
  - 7 个可配置的输入端点和 7 个可配置的输出端点
- 4 KB 专用端点内存空间 : 一个端点可定义为双缓存 1023 字节的等时传输
- 支持 VBUS 电压浮动 (droop) 和有效 ID 检测, 并产生中断信号
- 用微型直接内存访问 (μDMA) 有效的传输数据
  - 用于发送和接收的独立通道多达 3 个输入端点和 3 个输出端点
  - 当发送 FIFO 中包含所需数量的数据后, 可产生通道请求

### 1.3.3.3 UART ( 见第 827 页 )

通用异步收发器 (UART) 是用于 RS-232C 串行通信的集成电路, 包含可异步工作的发送器 ( 并转串 ) 和接收器 ( 串转并 )。

LM4F232H5QD 微控制器包含八个完全可编程的 16C550 型 UART。虽然其功能与 16C550 UART 相似, 但并不是寄存器兼容的。UART 能够根据 RX、TX、调制解调器流量控制、调制解调器状态和错误条件产生独立可屏蔽的中断。如果任何中断发生并且未被屏蔽, 那么模块将生成一个组合的中断。

这八个 UART 如下特性 :

- 可编程的波特率发生器, 在常规模式 ( 16 分频 ) 下最高可达 5 Mbps, 在高速模式 ( 8 分频 ) 下最高可达 10 Mbps

- 相互独立的 16×8 发送 (TX) FIFO 和接收 (RX) FIFO，可降低中断服务对 CPU 的占用
- FIFO 长度可编程，包括提供传统双缓冲接口的 1 字节深的操作
- FIFO 触发水平可设为 1/8, 1/4, 1/2, 3/4 和 7/8
- 标准的异步通信位：起始位、停止位和奇偶校验位
- 行中止的产生和检测
- 完全可编程的串行接口特性
  - 5、6、7 或 8 个数据位
  - 偶校验、奇校验、粘着或无奇偶校验位的产生/检测
  - 产生 1 或 2 个停止位
- IrDA 串行红外 (SIR) 编解码器
  - 可选择采用 IrDA 串行红外 (SIR) 输入输出或普通 UART 输入输出
  - 支持 IrDA SAR 编解码功能，半双工时数据传输率最高 115.2 Kbps
  - 支持标准的 3/16 位时间以及低功耗位时间 (1.41~2.23  $\mu$ s)
  - 可编程的内部时钟发生器，允许对参考时钟进行 1 到 256 分频以得到低功耗模式的位持续时间。
- 支持与 ISO 7816 智能卡的通讯
- 调制解调器流量控制和状态 (在 UART1 模块上)
- 支持 LIN 协议
- 支持 EIA-485 (9 位)
- 提供标准的基于 FIFO 深度的中断以及发送结束中断
- 用微型直接内存访问 ( $\mu$ DMA) 有效的传输数据
  - 相互独立的发送通道和接收通道
  - 当接收 FIFO 中有数据时产生单次请求；当接收 FIFO 到达预设的触发深度时产生猝发请求
  - 当发送 FIFO 中有空闲单元时产生单次请求；当发送 FIFO 到达预设的触发深度时产生猝发请求

#### 1.3.3.4 I<sup>2</sup>C (见第 933 页)

内部集成电路 (I<sup>2</sup>C) 总线通过一个两线设计 (串行数据线 SDA 和串行时钟线 SCL) 来提供双向数据传输。I<sup>2</sup>C 总线可与诸如串行存储器 (RAM 和 ROM)、网络设备、LCD、音频发生器等外部 I<sup>2</sup>C 器件相连。I<sup>2</sup>C 总线也可在产品的开发和生产过程中用于系统的测试和诊断。

I<sup>2</sup>C 总线上的设备可被指定为主机或从机。I<sup>2</sup>C 模块支持作为主机或从机来发送和接收数据，也支持既用作主机又用作从机的同步操作。I<sup>2</sup>C 主机和从机都能够产生中断。



LM4F232H5QD 微控制器包含六个具有以下特性的 I<sup>2</sup>C 模块：

- I<sup>2</sup>C 总线上的设备可被指定为主机或从机
  - 在主机或从机模式下都支持发送和接受数据
  - 支持它们作为主机和从机的同步操作
- 四种 I<sup>2</sup>C 模式：
  - 主机传送
  - 主机接收
  - 从机传送
  - 从机接收
- 四种传输速度：
  - 标准 (100 Kbps)
  - 快速 (400 Kbps)
  - 超快速 (1 Kbps)
  - 高速 (3.33 Mbps)
- 时钟低电平超时中断
- 双从机地址功能
- 时钟低电平超时中断
- 双从机地址功能
- 主机和从机产生中断
  - 主机因为传送或接收数据结束(或者是因为错误而取消)产生中断
  - 从机在主机向其发送数据或发出请求时，或检测到 START 或 STOP 信号时产生中断
- 主机带有仲裁和时钟同步功能，支持多主机以及 7 位寻址模式

### 1.3.3.5 SSI (见第 891 页)

同步串行接口 (SSI) 是一个将数据并转串的 4 线双向的通信接口。此 SSI 模块对从外围器件接收到的数据执行串并转换，对发送到外围设备的数据执行并串转换。这个 SSI 模块可以配置用作主设备或从设备。作为从设备的时候，还可以通过配置将 SSI 模块的输出禁能，从而使一个主设备可以与多个从设备相连。TX 和 RX 的通路都有内部 FIFO 负责缓冲。

此 SSI 模块还包含一个可编程的位速率时钟分频器和预分频器，SSI 模块输入的时钟信号将通过它们来生成 SSI 输出的串行时钟信号。位速率根据输入时钟产生，最大位速率取决于连接的外设。

LM4F232H5QD 微控制器包含四个具有以下特性的 SSI 模块：

- 提供可编程控制的接口，可与 Freescale 的 SPI 接口、MICROWIRE 或者 Texas Instruments 同步串行接口相连
- 主机或者从机方式
- 可编程的时钟位速率以及预分频器；
- 相互独立的发送 FIFO 和接收 FIFO，二者均为 16 位宽、8 个单元深；
- 可编程决定数据帧大小，范围为 4 到 16 位
- 内部回送测试 (loopback test) 模式，可进行诊断/调试测试
- 标准 FIFO 中断以及发送结束中断；
- 用微型直接内存访问 ( $\mu$ DMA) 有效的传输数据
  - 相互独立的发送通道和接收通道
  - 当接收 FIFO 中有数据时产生单次请求；当接收 FIFO 中包含 4 个数据单元时产生猝发请求
  - 当发送 FIFO 中有空闲单元时产生单次请求；当发送 FIFO 中包含 4 个空闲单元时产生猝发请求

### 1.3.4 系统集成

LM4F232H5QD 微控制器将各种标准系统功能集成到设备中，包括：

- 直接存储器访问控制器 (DMA)
- 系统控制和时钟，包括片上 16-M 高精度振荡器
- 六个 32 位定时器 ( 最多可达 412 个 16 位定时器 )
- 六个 64 位宽定时器 ( 最多可达 12 个 32 位宽定时器 )
- 十二个 16/32 位捕获比较 PWM (CCP) 管脚
- 十二个 32/64 位捕获比较 PWM (CCP) 管脚
- 带备用电池的低功耗休眠模块
- 在休眠模块中的实时时钟
- 两个看门狗定时器
  - 一个定时器使用主时钟振荡器
  - 一个定时器使用内部时钟振荡器
- 高达 105 个 GPIO，具体取决于配置
  - 高度灵活的管脚复用，可配置为 GPIO 或任一外设功能
  - 可独立配置的 2、4 或 8 mA 端口驱动能力
  - 高达 4 个 GPIO 具有 18 mA 驱动能力

下面的章节提供这些功能的更多信息。

### 1.3.4.1 直接内存访问（见第 534 页）

LM4F232H5QD 微控制器内置一个直接存储器访问（Direct Memory Access，简称为DMA）控制器，我们称之为微型DMA（ $\mu$ DMA）控制器。 $\mu$ DMA 控制器所提供的工作方式能够分载 Cortex-M4F 处理器参与的数据传输任务，从而更加高效地使用内核以及总线带宽。 $\mu$ DMA 控制器能够自动执行存储器与外设之间的数据传输。片上每个支持  $\mu$ DMA 功能的外设都有专用的  $\mu$ DMA 通道，通过合理的编程配置，当外设需要时能够自动在外设和存储器之间传输数据。 $\mu$ DMA 控制器具有以下特性：

- ARM PrimeCell® 32 通道的可配置  $\mu$ DMA 控制器；
- 支持存储器到存储器、存储器到外设、外设到存储器的  $\mu$ DMA 传输，包括：
  - 基本模式，用于简单的传输需求
  - 乒乓模式，用于实现持续数据流
  - 散聚模式，借助一个可编程的任务列表，由单个请求触发一连串的指定传输
- 高度灵活的可配置的通道配置；
  - 各通道均可独立配置、独立操作
  - 每个支持  $\mu$ DMA 功能的片上模块都有其专用通道
  - 灵活的通道分配
  - 对于双向模块，为其接收和发送各提供一个通道
  - 专用的软件通道，可由软件启动  $\mu$ DMA 传输
  - 每通道都可分别配置优先级
  - 可选配置：任一通道均可用作软件启动传输
- 优先级分为两级；
- 通过优化设计，改进了  $\mu$ DMA 控制器与处理器内核之间的总线访问性能：
  - 当内核不访问总线时， $\mu$ DMA 控制器即可占用总线
  - RAM条带处理
  - 外设总线分段
- 支持 8 位、16 位或 32 位数据宽度
- 待传输数目可编程为 2 的整数幂，有效范围 1 到 1024
- 源地址及目的地址可自动递增，递增单位可以是字节、半字、字、不递增
- 可屏蔽的外设请求
- 传输结束中断，且每个通道有独立的中断

### 1.3.4.2 系统控制及时钟（见第 203 页）

系统控制决定了器件的所有操作它不但提供了有关器件的信息，控制节能特性、对器件和单个外设的时钟进行控制，还处理复位的检测和报告。

- 设备识别信息：版本、型号、SRAM 大小、Flash 存储器大小等
- 功率控制
  - 片上低压差线性稳压器 (LDO)
  - 3.3 V 通电/断电序列，并控制内核的数字逻辑和模拟电路
  - 微控制器的低功耗选项：带有时钟门控的休眠和深度休眠模式
  - 片上模块的低功耗选择：软件控制各独立外设和存储器的掉电
  - 3.3 V 电源掉电检测，可通过中断或复位来报告
- 在微控制器中有多个时钟源可以使用
  - 精确内部振荡器 (PIOSC)：片上时钟源提供一个室温 16 MHz  $\pm$ 1% 的时钟。
    - 整个温度范围 16 MHz  $\pm$ 3%
    - 可使用 7 位的调整分辨率进行校准
    - 低功耗模式可软件控制掉电
  - 主振荡器 (MOSC)：主振荡器可通过两种方式提供一个频率精确的时钟源：外部单端时钟源连接到 OSC0 输入管脚，或者外部晶振串接在 OSC0 输入管脚和 OSC1 输出管脚间。
    - 可通过/避开片上 PLL 使用的外部晶振：支持从 4 MHz 到 25 MHz 的频率。
    - 外部晶振：支持从直流到最大芯片速度的频率
  - 内部 30 KHz 晶振：提供一个 30 kHz  $\pm$  50% 的工作频率。它是为在深度休眠省电模式期间使用的
  - 用于休眠模式的 32.768 kHz 外部晶振：有了该外部晶振，主时钟源不需要额外的晶振
- 灵活的复位源
  - 上电复位 (POR)
  - 复位管脚有效
  - 欠压 (BOR) 警告系统电源掉电
  - 软件复位
  - 看门狗定时器复位
  - 主振荡器 (MOSC) 故障

### 1.3.4.3 可编程定时器（见第 653 页）

可编程定时器可对驱动定时器输入管脚的外部事件进行计数或定时。每个 16/32 位 GPTM 模块包含两个 16 位的定时器/计数器，用户可以将它们配置成独立运行的定时器或事件计数器，或将它们配置成一个 32 位定时器或一个 32 位实时时钟 (RTC)。每个 32/64 位宽 GPTM 模块包含两个 32 位的定时器/计数器（称作 TimerA 和 TimerB），用户可以将它们配置成独立运行的定时器或事件计数器

或将它们配置成一个 64 位定时器或一个 64 位实时时钟 (RTC)。定时器也可以用来触发模数转换 (ADC)。

通用定时器模块 (GPTM) 包含六个 16/32 位 GPTM 时钟以及六个 32/64 位宽 GPTM 时钟，有以下功能选项：

- 16/32 位运行模式：
  - 16 位或 32 位可编程的单次定时器
  - 16 位或 32 位可编程的周期定时器
  - 具有 8 位预分频的 16 位通用定时器
  - 当有 32.768 KHz 的外部时钟源时可作为 32 位的实时时钟
  - 16 位输入沿计数或定时捕获模式，并带 8 位的预分频器
  - 带 8 位预分频器的 16 位 PWM 模式以及软件编程实现的 PWM 信号反相输出
- 32/64 位运行模式：
  - 32 位或 64 位可编程的单次定时器
  - 32 位或 64 位可编程的周期定时器
  - 具有 16 位预分频的 32 位通用定时器
  - 当有 32.768 KHz 的外部时钟源时可作为 64 位的实时时钟
  - 带有 16 位预分频器的 32 位输入沿计数或定时捕获模块
  - 带有 16 位预分频器的 32 位 PWM 模式以及软件编程实现的 PWM 信号反相输出
- 可以向上或向下计数
- 十二个 16/32 位捕获比较 PWM 管脚 (CCP)
- 十二个 32/64 位捕捉比较 PWM 管脚 (CCP)
- 菊花链式的定时器模块允许一个定时器开始计时多路时钟事件
- 定时器同步功能允许所选的定时器在同一时钟周期开始计数
- ADC 事件触发器
- 当调试时，CPU 出现暂停标识时，用户可以停止定时器事件（包括 RTC 模式）
- 可以确定从产生中断到进入中断服务程序所经过的时间
- 用微型直接内存访问 ( $\mu$ DMA) 有效的传输数据
  - 每个定时器具有专用通道
  - 定时器中断响应突发请求

#### 1.3.4.4 CCP 管脚 ( 见第 661 页 )

捕获/比较/PWM 管脚 (CCP) 可以被通用定时器模块使用，通过将 CCP 管脚作为输入，可定时/计数外部事件。此外，GPTM 模块可以在 CCP 管脚上产生一个简单的 PWM 输出。

LM4F232H5QD 微控制器包括十二个 16/32 位 CCP 管脚，通过编程，这些管脚能够按以下模式运行：

- 捕捉：通用定时器根据 CCP 输入的编程事件递增/递减。当编程事件发生时，它捕获和存储当前定时器的值。
- 比较：通用定时器根据 CCP 输入的编程事件递增/递减。它将当前定时器的值和存储的值相比较，当匹配时，产生中断信号。
- PWM：通用定时器根据系统时钟递增/递减。PWM 信号的产生基于计数值和存储值的匹配，并输出到 CCP 脚。

#### 1.3.4.5 休眠模块 ( 见第 475 页 )

休眠模块提供了一种逻辑，将主处理器和外设电源暂时关闭，在外部事件或基于时间的事件发生时唤醒休眠模块包括上电顺序逻辑，有如下特性：

- 32 位实时秒计数器 (RTC)，其时钟分辨率是 1/32,768
  - 32 位的 RTC 秒匹配寄存器和 15 位的亚秒匹配寄存器 ( 其时钟分辨率是 1/32,768 )，用作定时唤醒和中断产生
  - RTC 预分频器调整，对时钟速率进行良好地调节
- 电源控制的两种机制：
  - 使用离散的外部稳压器控制系统电源
  - 使用在寄存器控制下的内部开关控制片上电源
- 使用外部信号用作唤醒的专用管脚
- 只要  $V_{BAT}$  有效，RTC 运行的存储器和休眠所占存储器就一直有效
- 电池电量低检测、发出信号和中断发生；在电量低时提供可选的唤醒操作
- GPIO 管脚的状态在休眠过程中可保持不变
- 时钟源可以是 32.768 KHz 的外部晶振或振荡器
- 16 个 32 位字的带备用电池存储器，用于在休眠过程中保存状态
- 可编程的 RTC 匹配、外部唤醒和低电池电压事件的中断

#### 1.3.4.6 看门狗定时器 ( 见第 717 页 )

当系统由于软件错误或是由于因外部设备故障而无法按预期的方式响应的时候，使用看门狗定时器可以重新获得控制权。该 Stellaris 看门狗定时器在到达超时值时，可能会产生不可屏蔽的中断或复位此外，看门狗定时器兼容 ARM FIRM，其可被配置成在第一次超时时产生到微控制器的中断，并在第二次超时时产生复位信号。一旦看门狗定时器被配置，锁定寄存器可以写入，防止定时器的配置被意外更改。

LM4F232H5QD 微控制器含有两个看门狗定时器模块：看门狗定时器 0 使用系统时钟计时；看门狗定时器 1 由 PIOSC 定时器时钟驱动。该 Stellaris 看门狗定时器模块有如下特性：

- 32 位递减计数器，带可编程的加载寄存器
- 带启用功能的独立看门狗时钟
- 带中断屏蔽功能和可选 NMI 功能的可编程中断产生逻辑
- 软件跑飞时可锁定寄存器以提供保护
- 带启用/禁能的复位产生逻辑
- 调试期间，微控制器的 CPU 暂停时，用户可启用的停滞

#### 1.3.4.7 可编程的 GPIO（见第 597 页）

通用输入/输出 (GPIO) 管脚为各种连接方式带来了灵活性。该 Stellaris GPIO 模块包含 14 个物理 GPIO 块，每个对应一个独立的 GPIO 端口。GPIO 模块遵循 FIRM 规范（遵循 ARM 实时微控制器底层 IP 规范）并支持高达 0-105 个可编程的输入/输出管脚。可用的 GPIO 数目取决于正在使用的外设（有关每个 GPIO 管脚可用的信号见“信号表”在 1245 页）

- 高达 105 个 GPIO，具体取决于配置
- 高度灵活的管脚复用，可配置为 GPIO 或任一外设功能
- 配置为输入模式可承受 5 V 电压
- 两种端口访问方式：通过高级高性能总线 (AHB)（具有更好连续访问性能）或者通过传统的高级外设总线 (APB)（使用现有代码，具有更好的向后兼容能力）访问 A-H 和 J 端口；K-N 和 P 端口通过 AHB 访问
- AHB 端口每个时钟周期均可实现快速切换；APB 端口每两个时钟周期均可实现快速切换
- 可编程控制 GPIO 中断
  - 屏蔽中断发生
  - 边沿触发（上升沿和/或下降沿）
  - （高或低）电平触发
  - 端口 P 提供按管脚中断的功能
- 在读和写操作中通过地址线进行位屏蔽
- 可用于启动一个 ADC 采样序列或  $\mu$ DMA 传输
- 在休眠模式中，可以保持管脚的状态
- 配置为数字输入的管脚均为施密特触发
- 可编程控制 GPIO 管脚配置
  - 弱上拉或下拉电阻

- 数字通信时可配置为 2 mA、4 mA 和 8 mA 驱动电流；对于需要大电流的应用，最多四个管脚承载 18 mA
- 8 mA 驱动电流的斜率控制
- 开漏启用
- 数字输入启用

### 1.3.5 高级运动控制

LM4F232H5QD 微控制器将运动控制功能集成到芯片中，包括：

- 两个 PWM 模块，共 16 路高级 PWM 输出，用于运动和能源应用
- 八路故障输入，可用于低延迟关机
- 两路正交编码器输入 (QEI)

下面章节提供运动控制功能的更多详细信息。

#### 1.3.5.1 PWM (见第 1151 页)

脉宽调制 (PWM) 是一项功能强大的技术，它是一种对模拟信号电平进行数字化编码的方法。在脉宽调制中使用高分辨率计数器来产生方波，并且可以通过调整方波的占空比来对模拟信号电平进行编码。PWM 通常用于开关电源和运动控制等典型应用。每个 LM4F232H5QD PWM 模块包含 4 个 PWM 发生器模块和一个控制模块。每个 PWM 发生器模块包括 1 个定时器 (16 位递减或递增/递减计数器)、2 个比较器、1 个 PWM 信号发生器、1 个死区发生器和中断/ADC 触发选择器。每个 PWM 发生器模块产生两个 PWM 信号，这两个信号基于同一个定时器和频率和也可以是编程产生独立的信号，如插入了死区延时互补信号。

每个 PWM 发生模块具有以下特性：

- 4 个故障条件处理输入，能快速实现低延迟关闭，避免马达在被可控之前产生破坏，共计 8 个输入。
- 1 个 16 位的计数器
  - 运行在递减或递增/递减模式
  - 输出频率由一个 16 位的加载值控制
  - 可同步更新加载值
  - 当计数器的值到达零或者加载值的时候生成输出信号
- 2 个 PWM 比较器
  - 比较器值的更新可以同步
  - 在匹配的时候产生输出信号
- PWM 信号发生器
  - 根据计数器和 PWM 比较器的输出信号来产生 PWM 输出信号
  - 可产生两个独立的 PWM 信号



- 死区发生器
  - 产生 2 个带有可编程死区延时的 PWM 信号，适合驱动半 H 桥 ( half-H bridge )
  - 可以被旁路，不修改输入的 PWM 信号
- 可启动一个 ADC 采样序列

控制模块决定了 PWM 信号的极性，以及将哪个信号传递到管脚。PWM 发生器模块的输出在传递到器件管脚之前由输出控制模块管理 PWM 控制模块具有以下特性：

- 每个 PWM 信号都具有 PWM 输出启用
- 每个 PWM 信号都可以选择将输出反相 ( 极性控制 )
- 每个 PWM 信号都可以选择进行故障处理
- PWM 发生器模块的定时器同步
- PWM 发生器模块的定时器/比较器更新同步
- PWM 发生器模块之间的定时器/比较器更新的扩展 PWM 同步
- PWM 发生器模块中断状态被汇总
- 多路故障信号，可编程极性和过滤的扩展故障处理
- PWM 发生器可独立操作或者与其它发生器同步操作

### 1.3.5.2 QEI ( 见第 1224 页 )

正交编码器 ( 又名双通道增量式编码器 )，用于将线性位移转换成脉冲信号。通过监控脉冲的数目和两个信号的相对相位，用户可以跟踪旋转的位置、方向和速度。此外还有第三个通道，称为索引信号，用来对位置计数器进行复位，以确定绝对位置。该 Stellaris 带有索引脉冲的正交编码器 (QEI) 模块对由正交编码器转轮所产生的编码进行解码，从而计算位置对时间的积分，并确定旋转的方向。另外，该接口还能捕获编码器转轮的运行速率。QEI 输入的频率高达 1/4 处理器频率(例如，80 MHz 系统可达 20 MHz)。

该 LM4F232H5QD 微控制器包含两个 QEI 模块，可同时控制两个电机，并具有以下特性：

- 使用位置积分器来跟踪编码器的位置
- 输入可编程噪音过滤
- 使用内置定时器来捕获速度
- QEI 输入的频率高达 1/4 处理器频率(例如，50 MHz 系统可达 12.5 MHz)。
- 以下情况下产生中断：
  - 检测到索引脉冲
  - 速度定时器发生计满返回事件
  - 旋转方向发生改变
  - 检测到正交错误

### 1.3.6 模拟

LM4F232H5QD 微控制器将模拟功能集成到芯片中，包括：

- 212 位模数转换器 (ADC)，带有 24 个模拟输入通道；采样速率为 1M 采样/秒
- 三个模拟比较器
- 16 个数字比较器
- 片上电压稳压器

下面的章节提供模拟功能的更多信息。

#### 1.3.6.1 ADC（见第 742 页）

模数转换器 (ADC) 外设用于将连续的模拟电压转换成离散的数字量。该 Stellaris ADC 模块具有 12 位转换精度并支持 24 个输入通道；同时还内置温度传感器。4 个带缓冲的采样序列无需使用控制器，就可以对最多 24 个模拟输入源进行快速采样。每个采样序列发生器都可灵活配置其输入源、触发事件、中断的产生、序列发生器的优先级等内容。每个 ADC 块内置数字比较器功能，采样转换结果可移交给数字比较器模块，该数字比较器模块内置 8 路数字比较器。

LM4F232H5QD 微控制器内置 2 个 ADC 模块，具有以下特性：

- 24 个共用模拟输入通道
- 12 位精度的 ADC
- 单端和差分输入配置
- 片上内置温度传感器
- 1M 次/秒的采样率
- 可选的移相器，采样点以采样周期计可延后 22.5° 到 337.5°
- 4 个可编程的采样转换序列发生器，序列长度 1 到 8 个单元不等，且各自带有相应长度的转换结果 FIFO
- 灵活的触发控制
  - 控制器（软件）
  - 定时器
  - 模拟比较器
  - PWM
  - GPIO
- 硬件可对多达 64 个采样值进行平均计算
- 数字比较器模块提供 8 路数字比较器
- 转换器使用两个外部参考信号或使用 VDDA 和 GNDA 作为电压参考

- 模拟部分的电源/地与数字部分的电源/地相互独立
- 用微型直接内存访问 ( $\mu$ DMA) 有效的传输数据
  - 每个采样序列发生器各自有专用的通道
  - ADC 模块的 DMA 操作均采用猝发请求

### 1.3.6.2 模拟比较器 (见第 1136 页)

模拟比较器是一个外设，它能比较两个模拟电压的大小，并通过自身提供的逻辑输出端将比较结果以信号的形式输出。该 LM4F232H5QD 微控制器提供三个独立集成的模拟比较器，该模拟比较器可配置为驱动输出或产生中断或 ADC 事件。

比较器可以向器件管脚提供输出，以替换板上的模拟比较器。比较器也可以通过中断或触发 ADC 通知应用让它开始捕获采样序列。中断产生逻辑和 ADC 触发是各自独立的。这就意味着，中断可以在上升沿产生，而 ADC 在下降沿触发。

LM4F232H5QD 微控制器提供三个独立集成的模拟比较器，具有如下功能：

- 可以比较外部输入管脚和外部输入管脚或内部可编程的参考电压
- 比较器可将测试电压与下面的其中一种电压相比较
  - 单个外部独立的参考电压
  - 共用的外部参考电压
  - 公用的内部参考电压

### 1.3.7 JTAG 和 ARM 串行线调试 (见第 192 页)

联合测试行动组 (JTAG) 是一个 IEEE 标准，它定义了数字集成电路的测试访问端口和边界扫描结构，并且提供了一个标准化的串行接口来控制关联的测试逻辑。TAP，指令寄存器 (IR) 和数据寄存器 (DR) 可用来测试组合印制线路板的互连并获取组件的制造信息。JTAG 端口还提供了方法来访问和控制可测性设计的特性，如 I/O 管脚的观察和控制，扫描测试以及调试。Texas Instruments 实现了通过 ARM 串行线 JTAG 调试端口 (SWJ-DP) 接口取代 ARM SW-DP 和 JTAG-DP 的功能。SWJ-DP 接口将 SWD 和 JTAG 调试端口集成到一个模块中，提供所有的正常 JTAG 调试和测试功能，可实时访问系统存储器而不停止内核，且不需要任何目标寄居代码。SWJ-DP 接口具有如下特性：

- IEEE 1149.1-1990 兼容的测试访问端口 (TAP) 控制器
- 4 位指令寄存器 (IR) 链，用于存储 JTAG 指令
- IEEE 标准指令：BYPASS、IDCODE、SAMPLE/PRELOAD、EXTEST 和 INTEST
- ARM 附加指令：APACC、DPACC 和 ABORT
- 集成的 ARM 串行线调试 (SWD)
  - 串行线 JTAG 调试端口 (SWJ-DP)
  - Flash 修补和断点 (FPB) 单元，用于实现断点操作
  - 数据观察点和触发 (DWT) 单元，用于执行观察点、触发源和系统性能分析
  - 仪表跟踪宏单元 (ITM)，用于支持 printf 型调试

- 用于指令追踪捕捉的嵌入式追踪宏单元 (ETM)
- 跟踪端口接口单元 (TPIU) 用作跟踪端口分析仪的桥接

### 1.3.8 封装和温度

- 工业温度范围 ( -40°C 到 85°C ) 内符合 RoHS 标准的 144 管脚 LQFP 封装

## 1.4 LM4F232H5QD 微控制器硬件细节

有关管脚和封装的详细信息可在下一节中找到：

- “管脚图” 在 1244页
- “信号表” 在 1245页
- “工作特性” 在 1290页
- “电气特性” 在 1291页
- “封装信息” 在 1376页

## 2 Cortex-M4F 处理器

ARM® Cortex™-M4F 处理器提供了一个高性能、低成本的平台，可满足系统对降低存储需求、简化管脚数以及降低功耗三方面的要求，与此同时，它还提供出色的计算性能和优越的系统中断响应能力。特性包括：

- 32 位的 ARM® Cortex™-M4F 架构优化了小封装嵌入式应用方案。
- 80-MHz 运行速度；100 DMIPS 性能
- 优越的处理性能和更快的中断处理。
- 混合 16 位/32 位的 Thumb-2 指令集提供与 32 位 ARM 内核所期望的高性能而采用了更紧凑的内存大小，而这通常在 8 位和 16 位设备相关的存储容量中，特别是在微控制器级应用的几千字节存储中。
  - 单周期乘法指令和硬件除法器
  - 精确的位操作 ( bit-banding ) ，不仅最大限度的利用了存储器空间而且还改良了对外设的控制
  - 非对齐式数据访问，使数据能够更为有效的安置到存储器中
- 符合 IEEE754 的单精度浮点单元 (FPU)
- 16 位 SIMD 矢量处理单元
- 快速代码执行允许更低的处理器时钟和增加休眠模式时间
- Harvard 结构 - 将数据和指令所使用的总线进行了分离
- 高效的处理器内核，系统和存储器
- 硬件除法和以快速数字信号处理为导向的乘加
- 采用饱和算法处理信号
- 对时间苛刻的应用提供可确定的, 高性能的中断处理
- 存储器保护单元为操作系统机能提供特权操作模式
- 增强的系统调试提供全方位的断点和跟踪能力
- 串行线调试和串行线跟踪减少调试和跟踪过程中需求的管脚数
- 从 ARM7 处理器系列中移植过来，以获得更好的性能和电源效率
- 优化的单周期 Flash 存储器使用
- 集成多种休眠模式，更低功耗

该微控制器的®系列微控制器基于 Cortex-M3 内核，为注重成本的嵌入式微控制器应用，如工厂自动化与控制、工业控制电源设备、楼宇自动化和步进电机提供了高性能的 32 位运算能力。

本章描述微控制器的 Cortex-M4F 处理器的执行信息，包括编程模块、存储器模块、异常模块、故障处理和电源管理。

有关指令集的技术细节，请参考“ARM® Cortex™-M4 技术参考手册”。

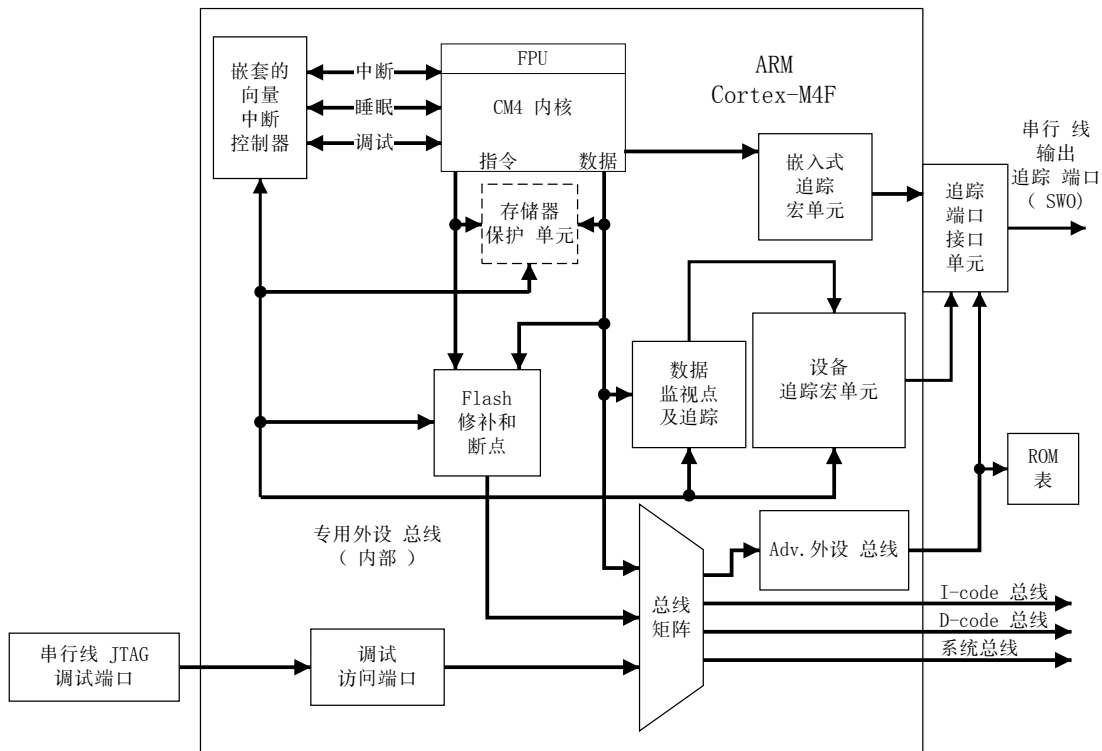
## 2.1 结构图

Cortex-M4F 处理器基于高性能的处理器内核，采用三级流水线的哈佛架构，是满足嵌入式应用的理想的处理器。该处理器带有高效的指令集和特别优化的设计，具有优异的能耗效率，并提供符合 IEEE754 的单精度浮点型计算单元、一系列单周期和 SIMD 乘法器和乘加功能，以及专用的硬件除法器的高端处理硬件。

为促进成本敏感型设备的设计，Cortex-M4F 处理器实现了紧耦合的系统部件以降低处理尺寸，同时提高了中断处理能力和系统调试能力。Cortex-M4F 处理器采用了基于 Thumb-2 技术的 Thumb® 指令集，确保高代码密度和降低程序存储需求。Cortex-M4F 采用现代 32 位架构和 8 位、16 位微处理器的高密度指令集，提供了优异的性能。

Cortex-M4F 处理器集成了嵌入中断处理器 (NVIC)，达到业界领先的中断性能。该微控制器的 NVIC 包括一个不可屏蔽中断 (NMI) 并提供 8 个中断优先级。紧密集成的处理器内核和 NVIC 提供快速的中断服务程序和显著的降低了中断延迟。硬件入栈和停止多步装载和存储操作进一步降低了中断延迟。中断处理不需要任何的汇编从而减少了 ISR 的代码开销。尾链优化同样显著地降低了 ISR 切换时的开销。为优化低功耗设计，NVIC 集成了睡眠模式，包括深度睡眠模式，该模式可使整个芯片迅速地降低功耗。

图 2-1. CPU 结构图



## 2.2 概述

### 2.2.1 系统级接口

Cortex-M4F 处理器采用 AMBA® 技术实现多接口来提供高速、低延迟的存储器访问。内核支持非对齐的数据访问和原子位操作，使得外设的控制，系统自旋锁和线程安全布尔数据处理更快。

Cortex-M4F 处理器内有一个存储器保护单元 (MPU)，可提供细粒度的存储器控制，使应用可以实现安全特权级别和隔离代码、数据和基于多任务的堆栈。

## 2.2.2 集成的可配置调试

Cortex-M4F 处理器实现了一个完整的硬件调试方案，通过一个传统的 JTAG 端口或者适合于微控制器和其他小封装的 2 脚的 SWD 来提供处理器和存储器的高度的系统可观测。该微控制器的通过兼容 ARM CoreSight™ 的串行线 JTAG 调试端口 (SWJ-DP) 接口取代了 ARM SW-DP 和 JTAG—DP。SWJ-DP 接口将 SWD 和 JTAG 调试端口组合到一个模块中。有关 SWJ-DP 的详细信息，请参考“ARM® Debug Interface V5 架构规格”。

对于系统跟踪，处理器集成了一个仪表跟踪宏单元 (ITM)，具有数据断点和分析单元。能够简单地低成本的系统跟踪事件，串行线观测器 (SWV) 通过一个单引脚能导出软件产生的信息、数据跟踪和分析信息的数据流。

嵌入式跟踪宏单元 (ETM) 提供了优异的指令追踪能力，相比传统的追踪单元，其追踪范围更加精确，同时还提供了全指令追踪功能。更多有关 ARM ETM 的详细信息，请参考“ARM® Embedded Trace Macrocell 架构规格”。

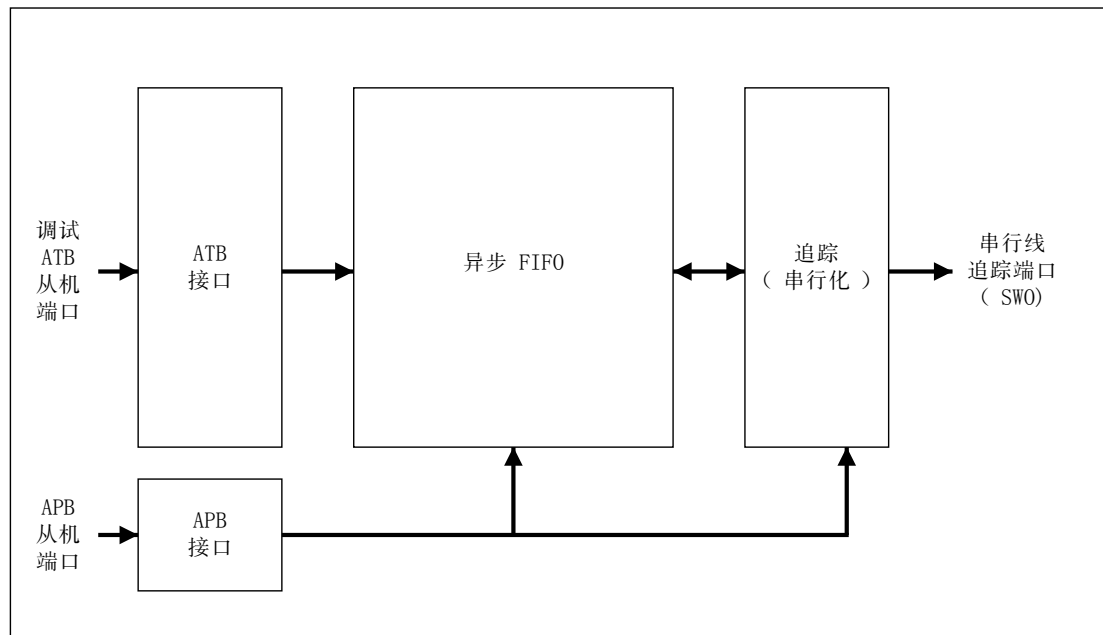
Flash 补丁和断点单元 (FPB) 提供高达 8 个硬件断点比较仪，以备调试器使用。在 FPB 中比较仪提供在程序代码中的 CODE 存储器区多达 8 个字的重映射功能。这允许存储在只读 Flash 中应用可以拼接到片上 SRAM 或 Flash 的另一个区。当需要拼接时，应用编程 FPB 来重映射一组地址。当这些地址被访问时，访问被重定位到 FPB 配置中指定的重映射表中。

更多有关 Cortex-M4F 调试功能的信息，请参考“ARM® Debug Interface V5 架构规格”。

## 2.2.3 跟踪端口接口单元 (TPIU)

TPIU 充当来自 ITM 的 Cortex-M4F 跟踪数据以及片外跟踪端口分析仪之间的桥接器，如图 2-2 在 71 页 中所示。

图 2-2. TPIU 结构图



## 2.2.4 Cortex-M4F 系统组件细节

Cortex-M4F 包含以下系统组件：

- SysTick  
24 位的递减定时器，可被用作实时操作系统 (RTOS) 的节拍定时器，或者作为一个简单的计数器，参见“系统定时器 ( SysTick )”在 118 页。
- 嵌套式向量化中断控制器 (NVIC)  
一个嵌入的中断控制器，支持低延迟中断处理，参见“嵌套向量中断控制器 ( NVIC )”在 119 页。
- 系统控制模块 (SCB)  
处理器的编程模型接口。系统控制块 (SCB) 提供系统实现信息和系统控制，包括系统异常的配置、控制和报告 ( 请参考“系统控制模块 ( SCB )”在 120 页 ) 。
- 存储器保护单元 (MPU)  
通过为不同的存储器区定义存储器属性来提高系统的稳定性。MPU 提供多达 8 个不同区和一个可选的预定义的背景区，参见“存储器保护单元 ( MPU )”在 121 页。
- 浮点单元 (FPU)  
完全支持单精度的加、减、乘、除、乘加以及平方根操作。它也提供定点和浮点数据格式之间的转换以及浮点常数指令。

## 2.3 编程模型

这部分描述了 Cortex-M4F 的编程模型。另外还有单个的内核寄存器描述，处理器模式的信息和软件执行、堆栈的权限级别。

### 2.3.1 处理器模式和软件执行的权限级别

Cortex-M4F 具有两种工作模式：

- 线程模式  
用于执行应用程序软件。处理器复位后，进入线程模式。
- 处理器模式  
用于处理异常。当处理器完成异常的处理之后返回到线程模式。

另外，Cortex-M4F 有两个权限级别：

- 无特权级  
在此模式下，软件有如下限制：
  - 限制访问 MSR 和 MRS 指令，且不使用 CPS 指令
  - 不能访问系统定时器、NVIC 或者系统控制块
  - 限制对某些存储器和外设的访问
- 特权级  
在此模式下，软件可以使用所有的指令和访问所有的资源。



在线程模式下，CONTROL 寄存器（参见 87 页）控制软件是在特权级还是非特权级。在处理器模式下，软件执行总是在特权级下。

在线程模式下，只有特权级软件可以对 CONTROL 寄存器进行写操作来改变软件的特权级。非特权级软件可使用 SVC 指令来产生一个系统调用，把控制权转移到特权级软件。

### 2.3.2 堆栈

该处理器使用向下的满栈，意味着在存储器中堆栈指针指向的是最后入栈项目。当处理器推入一个新的项目入栈时，先递减堆栈指针，再把项目写入到新的存储器位置。处理器实现了两个堆栈：主堆栈和处理堆栈，每个堆栈的指针都包含于独立的寄存器中（请参考 77 页上的 SP 寄存器）。

在线程模式，CONTROL 寄存器（见 87 页）控制处理器是使用主堆栈还是处理堆栈。在处理器模式下，处理器总是使用主堆栈。处理器操作选项如表 2-1 在 73 页所示：

表 2-1. 处理器模式、特权等级和堆栈使用摘要

处理器模式	用途	特权等级	使用的堆栈
线程模式	应用程序	特权级或非特权级 <sup>a</sup>	主堆栈或进程堆栈 <sup>a</sup>
处理器模式	异常处理程序	总是特权级	主堆栈

a. 参见 CONTROL (87 页)。

### 2.3.3 寄存器映射

图 2-3 在 74 页描述了 Cortex-M4F 的寄存器组。表 2-2 在 74 页列出了内核寄存器。核心寄存器并没有映射存储器且可以通过寄存器名称访问，所以基址是 n/a（不适用）且没有偏移。

图 2-3. Cortex-M4F 的寄存器组。

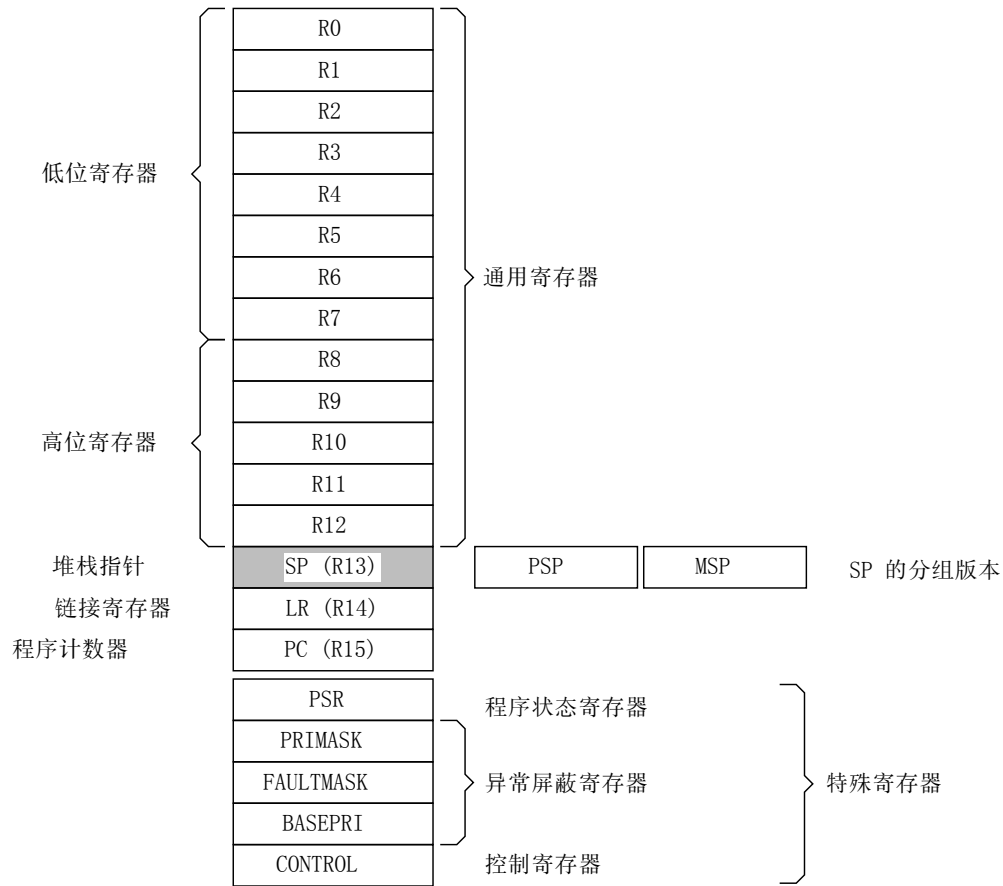


表 2-2. 处理器 寄存器映射

偏移量	名称	类型	复位	描述	见页面
-	R0	R/W	-	Cortex 通用寄存器 0	76
-	R1	R/W	-	Cortex 通用寄存器 1	76
-	R2	R/W	-	Cortex 通用寄存器 2	76
-	R3	R/W	-	Cortex 通用寄存器 3	76
-	R4	R/W	-	Cortex 通用寄存器 4	76
-	R5	R/W	-	Cortex 通用寄存器 5	76
-	R6	R/W	-	Cortex 通用寄存器 6	76
-	R7	R/W	-	Cortex 通用寄存器 7	76
-	R8	R/W	-	Cortex 通用寄存器 8	76
-	R9	R/W	-	Cortex 通用寄存器 9	76
-	R10	R/W	-	Cortex 通用寄存器 10	76
-	R11	R/W	-	Cortex 通用寄存器 11	76

表 2-2. 处理器 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
-	R12	R/W	-	Cortex 通用寄存器 12	76
-	SP	R/W	-	堆栈指针寄存器	77
-	LR	R/W	0xFFFF.FFFF	链接寄存器	78
-	PC	R/W	-	程序计数器寄存器	79
-	PSR	R/W	0x0100.0000	程序状态寄存器	80
-	PRIMASK	R/W	0x0000.0000	优先级屏蔽寄存器	84
-	FAULTMASK	R/W	0x0000.0000	故障屏蔽寄存器	85
-	BASEPRI	R/W	0x0000.0000	基本优先级屏蔽寄存器	86
-	CONTROL	R/W	0x0000.0000	控制寄存器	87
-	FPSC	R/W	-	浮点状态控制	89

### 2.3.4 寄存器描述

本章列出 ( 以图 2-3 在 74 页中所示的顺序 ) 并描述了 Cortex-M4F 寄存器。内核寄存器地址不是存储器映射的, 因此只能通过寄存器名称访问, 而不能通过使用偏移量的方式访问。

注意: 在寄存器描述栏中, 描述了程序在线程模式和处理器模式执行时的类型。调试访问可能不同。

- 寄存器 1: Cortex 通用寄存器 0 ( R0 )  
 寄存器 2: Cortex 通用寄存器 1 ( R1 )  
 寄存器 3: Cortex 通用寄存器 2 ( R2 )  
 寄存器 4: Cortex 通用寄存器 3 ( R3 )  
 寄存器 5: Cortex 通用寄存器 4 ( R4 )  
 寄存器 6: Cortex 通用寄存器 5 ( R5 )  
 寄存器 7: Cortex 通用寄存器 6 ( R6 )  
 寄存器 8: Cortex 通用寄存器 7 ( R7 )  
 寄存器 9: Cortex 通用寄存器 8 ( R8 )  
 寄存器 10: Cortex 通用寄存器 9 ( R9 )  
 寄存器 11: Cortex 通用寄存器 10 ( R10 )  
 寄存器 12: Cortex 通用寄存器 11 ( R11 )  
 寄存器 13: Cortex 通用寄存器 12 ( R12 )

Rn 寄存器是供数据操作的 32 位通用寄存器，既可在特权模式下访问，也可在非特权模式下访问。

#### Cortex 通用寄存器 0 (R0)

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:0	DATA	R/W	-	寄存器数据。

**寄存器 14: 堆栈指针寄存器 ( SP )**

堆栈指针 (SP) 即寄存器 R13。在线程模式下，该寄存器功能取决于控制寄存器 (CONTROL) 寄存器中的 ASP 位。当将 ASP 位清零时，此寄存器是主堆栈指针 (MSP)。当将 ASP 位置位时，此寄存器是处理堆栈指针 (PSP)。复位时，ASP 清零，同时处理器将地址 0x0000.0000 处的值载入 MSP。仅可在特权级模式下才可访问 MSP 寄存器；在特权级和非特权级模式都可访问 PSP 寄存器。

**堆栈指针寄存器 (SP)**

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SP															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SP															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:0	SP	R/W	-	该域是堆栈指针的地址。

**寄存器 15: 链接寄存器 ( LR )**

链接寄存器 (LR) 是寄存器 R14。它存储子程序、函数调用和异常的返回信息。LR 在特权和非特权模式下均可访问。

EXC\_RETURN 在发生异常时被载入到 LR 中。其值及相关描述参见 表 2-10 在 108页。

**链接寄存器 (LR)**

类型 R/W, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LINK															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LINK															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	LINK	R/W	0xFFFF.FFFF	此域是返回地址。

**寄存器 16: 程序计数器寄存器 ( PC )**

程序计数器 (PC) 是寄存器 R15，用于保存当前程序的地址。复位时，处理器将地址 0x0000.0004 处的复位向量载入到 PC 寄存器。复位时，该复位向量的位 0 载入到 EPSR 寄存器中的 THUMB 位，此时该位必须是 1。在特权和非特权模式下均可访问 PC 寄存器。

**程序计数器寄存器 (PC)**

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PC															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:0	PC	R/W	-	此域是当前程序的地址。

**寄存器 17: 程序状态寄存器 ( PSR )**

注意: 此寄存器也被称为 xPSR 寄存器。

程序状态寄存器 (PSR) 有三个功能, 且寄存器中的位被分配为不同的功能:

- 应用程序状态寄存器 (APSR) 位域 31:27, 位域 19:16
- 运行程序状态寄存器 (EPSR), 位域 26:24, 15:10
- 中断程序状态寄存器 (IPSR), 位域 7:0

仅可在特权级模式下才可访问 PSR、IPSR 和 EPSR 寄存器; 特权级和非特权级模式都可访问 APSR 寄存器。

APSR 保存之前指令执行的状态标记的当前状态。

EPSR 包含 Thumb 状态位和 If-Then (IT) 指令或中断的多周期载入和存储指令的“可中断-可继续指令”(ICI) 域的执行状态位。尝试通过应用软件使用 MSR 指令直接读取 EPSR 的操作, 其返回值总是为零。尝试在应用程序软件中使用 MSR 指令对 EPSR 进行写操作, 将始终被忽略。故障处理器可通过检查堆栈式的 PSR 中的 EPSR 值来确定该故障操作 (见“异常进入和返回”在 106 页)。

IPSR 包含的是当前中断服务程序 (ISR) 的异常类型号。

这些寄存器可以单独访问或者是任何两三个一起访问, 访问时使用寄存器的名称作为 MSR 或 MRS 指令的参数。例如, 可以使用 PSR + MRS 指令组合对所有寄存器进行读操作; APSR + MSR 指令组合只能对 APSR 寄存器进行写操作。80 页列出了 PSR 所有可能的寄存器组合。更多关于如何访问程序状态寄存器的信息, 请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”中的 MRS 和 MSR 指令。

**表 2-3. PSR 寄存器组合**

寄存器	类型	组合
PSR	R/W <sup>a, b</sup>	APSR、EPSR 和 IPSR
IEPSR	RO	EPSR 和 IPSR
IAPSR	R/W <sup>a</sup>	APSR 和 IPSR
EAPSR	R/W <sup>b</sup>	APSR 和 EPSR

a. 处理器忽略对 IPSR 位的写操作。

b. 读 EPSR 位返回零, 处理器忽略这些位的写操作。

**程序状态寄存器 (PSR)**

类型 R/W, 复位 0x0100.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	N	Z	C	V	Q	ICI / IT		THUMB	保留				GE			
类型	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICI / IT				保留				ISRNUM							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



位/域	名称	类型	复位	描述
31	N	R/W	0	<p>APSR 负数或小于标志</p> <p>值 描述</p> <p>1 先前操作的结果是负数或小于。</p> <p>0 先前操作的结果是整数、零、大于或相等。</p> <p>当访问 PSR 或 APSR 时该位的值才有意义。</p>
30	Z	R/W	0	<p>APSR 零标志</p> <p>值 描述</p> <p>1 先前操作结果为0。</p> <p>0 先前操作结果非0。</p> <p>当访问 PSR 或 APSR 时该位的值才有意义。</p>
29	C	R/W	0	<p>APSR 进位或借位标志</p> <p>值 描述</p> <p>1 先前加法操作导致进位或者先前的减法操作没有产生借位。</p> <p>0 先前的加法操作没有导致进位或者先前的减法操作导致了借位。</p> <p>当访问 PSR 或 APSR 时该位的值才有意义。</p>
28	V	R/W	0	<p>APSR 溢出标志</p> <p>值 描述</p> <p>1 先前的操作导致了溢出。</p> <p>0 先前的操作没导致溢出。</p> <p>当访问 PSR 或 APSR 时该位的值才有意义。</p>
27	Q	R/W	0	<p>APSR DSP 溢出和饱和标志</p> <p>值 描述</p> <p>1 产生了 DSP 溢出或饱和。</p> <p>0 自从复位或自从该位上次清零以来没有发生DSP溢出或饱和。</p> <p>当访问 PSR 或 APSR 时该位的值才有意义。</p> <p>该位可由软件使用 MRS 指令清零。</p>
26:25	ICI / IT	RO	0x0	<p>EPSR ICI / IT 状态</p> <p>这些位以及位 15:10 包含中断的多指令载入操作或多指令存储操作的“可中断-可继续指令”(ICI) 域或者 IT 指令的执行状态位。</p> <p>当 EPSR 保持 ICI 执行状态时，位 26:25 都是 0。</p> <p>If-Then 模块在 IT 指令之后最多包含四条指令。该模块中的每条指令都是带有条件的。这些指令的条件有可能都一样，其中一些也可能和其它相反。有关更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。</p> <p>当访问 PSR 或 EPSR 时该域的值才有意义。</p>

位/域	名称	类型	复位	描述
24	THUMB	RO	1	<p>EPSR Thumb 状态</p> <p>该位指示Thumb的状态且应该一直为1。</p> <p>以下情况可清除 THUMB 位：</p> <ul style="list-style-type: none"> <li>■ BLX、BX 和 POP{PC} 指令</li> <li>■ 从异常返回上的堆栈式 xPSR 值恢复</li> <li>■ 异常进入或复位时的向量值的位 0</li> </ul> <p>当该位清零时，尝试执行指令会导致出错或锁定。有关更多信息，请参考“死锁”在 110页。</p> <p>当访问 PSR 或 EPSR 时该位才有意义。</p>
23:20	保留	RO	0x00	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
19:16	GE	R/W	0x0	<p>大于或等于标志</p> <p>有关更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”中 SEL 指令的描述。</p> <p>当访问 PSR 或 APSR 时该位域才有意义。</p>
15:10	ICI / IT	RO	0x0	<p>EPSR ICI / IT 状态</p> <p>这些位以及位 26:25 都包含可用于中断的多指令载入操作或多指令存储操作的“可中断-可继续指令”(ICI) 域以及 IT 指令的执行状态位。</p> <p>在执行 LDM、STM、PUSHPOP、VLDM、VSTM、VPUSH 或 VPOP 指令过程中，如果出现中断，处理器将暂时停止多指令载入或多指令存储操作，并将多指令操作中的下一个寄存器的操作数存储到位 15:12 中。处理完中断后，处理器返回到位 15:12 指向的寄存器，然后恢复多指令载入和存储操作。当 EPSR 保持 ICI 执行状态时，位 11:10 都是 0。</p> <p>If-Then 模块在 16 位的 IT 指令之后最多包含四条指令。该模块中的每条指令都是带有条件的。这些指令的条件有可能都一样，其中一些也可能和其它相反。有关更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。</p> <p>当访问 PSR 或 EPSR 时该域的值才有意义。</p>
9:8	保留	RO	0x0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>

位/域	名称	类型	复位	描述
7:0	ISRNUM	RO	0x00	IPSR ISR 编号 该域包含的是当前中断服务程序 (ISR) 的异常类型号。
			值	描述
			0x00	线程模式
			0x01	保留
			0x02	NMI
			0x03	硬故障
			0x04	存储器管理故障
			0x05	总线故障
			0x06	用法故障
			0x07-0x0A	保留
			0x0B	SVCall
			0x0C	保留用于调试
			0x0D	保留
			0x0E	PendSV
			0x0F	SysTick
			0x10	中断向量 0
			0x11	中断向量 1
			...	...
			0x9A	中断向量 138

有关更多信息，请参考“异常类型”在 100页。  
当访问 PSR 或 IPSR 时，该域的值才有意义。

## 寄存器 18: 优先级屏蔽寄存器 ( PRIMASK )

PRIMASK 寄存器可屏蔽所有优先级可编程的异常。只有固定优先级的复位、NMI 和硬故障是例外的。当异常可能影响到关键任务执行时间时应该被禁止。该寄存器只能在特权模式下访问。MSR 和 MRS 指令用于访问 PRIMASK 寄存器，CPS 指令可用于更改 PRIMASK 寄存器的值。有关这些指令的更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。有关异常优先级的更多信息，请参考“异常类型”在 100 页。

## 优先级屏蔽寄存器 (PRIMASK)

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															PRIMASK
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	PRIMASK	R/W	0	优先级屏蔽 值 描述 1 禁止所有可配置优先级的异常。 0 无影响。

## 寄存器 19: 故障屏蔽寄存器 ( FAULTMASK )

FAULTMASK 寄存器可屏蔽除不可屏蔽中断 (NMI) 外的所有异常。当异常可能影响到关键任务执行时间时应该被禁止。该寄存器只能在特权模式下访问。MSR 和 MRS 指令用于访问 FAULTMASK 寄存器，CPS 指令可用于更改 FAULTMASK 寄存器的值。有关这些指令的更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。有关异常优先级的更多信息，请参考“异常类型”在 100页。

### 故障屏蔽寄存器 (FAULTMASK)

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	FAULTMASK	R/W	0	故障屏蔽  值 描述 1 禁止除了NMI的所有异常。 0 无影响。  处理器从除了 NMI 处理程序外的任何异常处理程序退出时将清零 FAULTMASK 位。

## 寄存器 20: 基本优先级屏蔽寄存器 (BASEPRI)

BASEPRI 寄存器定义了异常处理的最小优先级。当 BASEPRI 设为非零值时，它将会禁止激活所有优先级低于或等于 BASEPRI 值的异常。当异常可能影响到关键任务执行时间时应该被禁止。该寄存器只能在特权模式下访问。有关异常优先级的更多信息，请参考“异常类型”在 100 页。

## 基本优先级屏蔽寄存器 (BASEPRI)

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留							BASEPRI			保留					
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:5	BASEPRI	R/W	0x0	<p>基本优先级</p> <p>任何可编程优先级的异常的优先级低于或等于该域的值时将被屏蔽。PRIMASK 寄存器可用来屏蔽所有优先级可编程的异常。优先级越高，优先级越低。</p> <p>值 描述</p> <p>0x0 所有异常都不屏蔽。</p> <p>0x1 所有优先级在 1-7 的异常都将屏蔽。</p> <p>0x2 所有优先级在 2-7 的异常都将屏蔽。</p> <p>0x3 所有优先级在 3-7 的异常都将屏蔽。</p> <p>0x4 所有优先级在 4-7 的异常都将屏蔽。</p> <p>0x5 所有优先级在 5-7 的异常都将屏蔽。</p> <p>0x6 所有优先级在 6-7 的异常都将屏蔽。</p> <p>0x7 所有优先级为 7 的异常都将屏蔽。</p>
4:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 21: 控制寄存器 ( CONTROL )

当处理器处于线程模式时，CONTROL 寄存器控制使用的堆栈和软件执行的特权等级，并指示 FPU 是否处于活动状态。该寄存器只能在特权模式下访问。

处理器模式总是使用 MSP，因此在处理器模式下，处理器将忽略对 CONTROL 寄存器的 ASP 位执行的明确的写操作。异常进入和返回机制自动采用 EXC\_RETURN ( 参见表 2-10 在 108 页 ) 的值来更新 CONTROL 寄存器。在带操作系统环境下，线程在线程模式下运行应该使用进程堆栈，而内核和异常处理应该使用主堆栈。默认情况下，线程模式使用 MSP。要将线程模式中使用的堆栈指针切换到 PSP，可按照“Cortex™-M3/M4 Instruction Set 技术用户手册”中的详细介绍使用 MSR 指令将 ASP 位置位，或使用相应的 EXC\_RETURN 值在发生异常时返回线程模式，如表 2-10 在 108 页所示。

**注意：** 当改变堆栈指针时，软件必须在执行 MSR 指令后立即使用一条 ISB 指令，确保 ISB 后的指令使用新的堆栈指针。请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。

### 控制寄存器 (CONTROL)

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													FPCA	ASP	TMPL
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
2	FPCA	R/W	0	激活浮点上下文  <b>值 描述</b> 1 激活浮点上下文 0 没有激活浮点上下文  Cortex-M4F 使用该位来确定在处理异常时是否保留浮点状态。
1	ASP	R/W	0	<b>重要：</b> 两个位控制何时启用 FPCA：浮点上下文控制 (FPCC) 寄存器中的 ASPEN 位和 辅助控制 (ACTLR) 寄存器中的 DISFPCA 位。  活动堆栈指针  <b>值 描述</b> 1 PSP 是当前堆栈指针。 0 MSP 是当前堆栈指针。  在处理器模式下，该位读出零写会忽略。当异常返回时，Cortex-M4F 自动更新该位。

位/域	名称	类型	复位	描述
0	TMPL	R/W	0	线程模式特权级
				值 描述
				1 非特权级软件可以在线程模式被执行。
				0 只有特权级的软件才能在线程模式下被执行。



## 寄存器 22: 浮点状态控制 (FPSC) 寄存器

FPSC 寄存器提供浮点系统所有必要的用户级控制。

## 浮点状态控制 (FPSC)

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	N	Z	C	V	保留	AHP	DN	FZ	RMODE			保留				
类型	R/W	R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO
复位	-	-	-	-	0	-	-	-	-	-	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								IDC	保留		IXC	UFC	OFC	DZC	IOC
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	-	0	0	-	-	-	-	-

位/域	名称	类型	复位	描述
31	N	R/W	-	负极状态码标志 浮点比较操作会更新此状态码标志。
30	Z	R/W	-	零状态码标志 浮点比较操作会更新此状态码标志。
29	C	R/W	-	进位状态码标志 浮点比较操作会更新此状态码标志。
28	V	R/W	-	上溢状态码标志 浮点比较操作会更新此状态码标志。
27	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
26	AHP	R/W	-	Alternative 半精度 置位时，选择 Alternative 半精度格式。清零时，选择 IEEE 半精度格式。 FPDSC 寄存器中的 AHP 位包含该位的默认值。
25	DN	R/W	-	默认 NaN 模式 置位时，任何涉及一个或多个 NaN 的操作都会返回默认 NaN。清零时，NaN 操作数传递到浮点操作的输出。 FPDSC 寄存器中的 DN 位包含该位的默认值。
24	FZ	R/W	-	清零模式 置位时，清零模式启用。清零时，清零模式被禁用，浮点系统的行为完全符合 IEEE 754 标准。 FPDSC 寄存器中的 FZ 位包含该位的默认值。

位/域	名称	类型	复位	描述
23:22	RMODE	R/W	-	舍入模式 几乎所有的浮点指令都使用指定的舍入模式。 FPDSC 寄存器中的 RMODE 位包含该位的默认值。  值 描述 0x0 向最接近值舍入 (RN) 模式 0x1 向正无穷大舍入 (RP) 模式 0x2 向负无穷大舍入 (RM) 模式 0x3 向 0 舍入 (RZ) 模式
21:8	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7	IDC	R/W	-	输入反常累积异常 置位时，表示该异常自上一次向该位写 0 已经发生。
6:5	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
4	IXC	R/W	-	不精确的累积异常 置位时，表示该异常自上一次向该位写 0 已经发生。
3	UFC	R/W	-	下溢累积异常 置位时，表示该异常自上一次向该位写 0 已经发生。
2	OFC	R/W	-	上溢累积异常 置位时，表示该异常自上一次向该位写 0 已经发生。
1	DZC	R/W	-	除零累积异常 置位时，表示该异常自上一次向该位写 0 已经发生。
0	IOC	R/W	-	无效操作累积异常 置位时，表示该异常自上一次向该位写 0 已经发生。

### 2.3.5 异常和中断

Cortex-M4F 处理器支持中断和系统异常。处理器和嵌入向量中断控制器分级和处理所有异常。异常改变了软件控制流。处理器使用处理器模式处理除了复位之外的所有异常。有关更多信息，请参考“异常进入和返回”在 106页。

NVIC寄存器控制中断处理。有关更多信息，请参考“嵌套向量中断控制器 (NVIC)”在 119页。

### 2.3.6 数据类型

Cortex-M4F 支持 32 位字、16 位半字和 8 位字节。处理器也支持 64 位数据传输指令。所有指令和数据访问都是小端模式。有关更多信息，请参考“存储器区、类型和属性”在 94页。

## 2.4 存储模型

本节描述处理器存储映射、存储器访问和位带特征。处理器提供4GB的寻址空间和混合的存储映射。

LM4F232H5QD 控制器的存储器映射请参考表 2-4 在 91页。在本手册中，寄存器地址将采用十六进制递增的形式给出，并与存储器映射表中模块的基址一一对应

在SRAM和外设区包含位带区。位带区对位数据提供原子操作，参见“位带区”在 96页。

处理器为核心外设寄存器保留了相应范围的专用外设总线 (PPB) 地址区 (参见“Cortex-M4 外设”在 118页)。

注意：在存储器映射，当读写所有保留空间时返回一个总线故障。

表 2-4. 存储器映射

起始	结束	描述	详见页
<b>存储器</b>			
0x0000.0000	0x0003.FFFF	片上Flash	511
0x0004.0000	0x00FF.FFFF	保留	-
0x0100.0000	0x1FFF.FFFF	为 ROM 保留	504
0x2000.0000	0x2000.7FFF	片上 SRAM 位带	504
0x2000.8000	0x21FF.FFFF	保留	-
0x2200.0000	0x220F.FFFF	片上 SRAM 的位带别名区，起始地址为 0x2000.0000	504
0x2210.0000	0x3FFF.FFFF	保留	-
<b>FIRM 外设</b>			
0x4000.0000	0x4000.0FFF	看门狗定时器0	720
0x4000.1000	0x4000.1FFF	看门狗定时器1	720
0x4000.2000	0x4000.3FFF	保留	-
0x4000.4000	0x4000.4FFF	GPIO 端口 A	608
0x4000.5000	0x4000.5FFF	GPIO 端口 B	608
0x4000.6000	0x4000.6FFF	GPIO 端口 C	608
0x4000.7000	0x4000.7FFF	GPIO 端口 D	608
0x4000.8000	0x4000.8FFF	SSI0	904
0x4000.9000	0x4000.9FFF	SSI1	904
0x4000.A000	0x4000.AFFF	SSI2	904
0x4000.B000	0x4000.BFFF	SSI3	904
0x4000.C000	0x4000.CFFF	UART0	839
0x4000.D000	0x4000.DFFF	UART1	839

表 2-4. 存储器映射 (续)

起始	结束	描述	详见页
0x4000.E000	0x4000.EFFF	UART2	839
0x4000.F000	0x4000.FFFF	UART3	839
0x4001.0000	0x4001.0FFF	UART4	839
0x4001.1000	0x4001.1FFF	UART5	839
0x4001.2000	0x4001.2FFF	UART6	839
0x4001.3000	0x4001.3FFF	UART7	839
0x4001.4000	0x4001.FFFF	保留	-
外设			
0x4002.0000	0x4002.0FFF	I <sup>2</sup> C 0	951
0x4002.1000	0x4002.1FFF	I <sup>2</sup> C 1	951
0x4002.2000	0x4002.2FFF	I <sup>2</sup> C 2	951
0x4002.3000	0x4002.3FFF	I <sup>2</sup> C 3	951
0x4002.4000	0x4002.4FFF	GPIO 端口 E	608
0x4002.5000	0x4002.5FFF	GPIO 端口 F	608
0x4002.6000	0x4002.6FFF	GPIO 端口 G	608
0x4002.7000	0x4002.7FFF	GPIO 端口 H	608
0x4002.8000	0x4002.8FFF	PWM 0	1163
0x4002.9000	0x4002.9FFF	PWM 1	1163
0x4002.A000	0x4002.BFFF	保留	-
0x4002.C000	0x4002.CFFF	QE10	1229
0x4002.D000	0x4002.DFFF	QE11	1229
0x4002.E000	0x4002.FFFF	保留	-
0x4003.0000	0x4003.0FFF	16/32 位定时器 0	674
0x4003.1000	0x4003.1FFF	16/32 位定时器 1	674
0x4003.2000	0x4003.2FFF	16/32 位定时器 2	674
0x4003.3000	0x4003.3FFF	16/32 位定时器 3	674
0x4003.4000	0x4003.4FFF	16/32 位定时器 4	674
0x4003.5000	0x4003.5FFF	16/32 位定时器 5	674
0x4003.6000	0x4003.6FFF	32/64 位定时器 0	674
0x4003.7000	0x4003.7FFF	32/64 位定时器 1	674
0x4003.8000	0x4003.8FFF	ADC0	762
0x4003.9000	0x4003.9FFF	ADC1	762
0x4003.A000	0x4003.BFFF	保留	-
0x4003.C000	0x4003.CFFF	模拟比较器	1136
0x4003.D000	0x4003.DFFF	GPIO 端口 J	608
0x4003.E000	0x4003.FFFF	保留	-
0x4004.0000	0x4004.0FFF	CAN0 控制器	997
0x4004.1000	0x4004.1FFF	CAN1 控制器	997
0x4004.2000	0x4004.BFFF	保留	-
0x4004.C000	0x4004.CFFF	32/64 位定时器 2	674
0x4004.D000	0x4004.DFFF	32/64 位定时器 3	674
0x4004.E000	0x4004.EFFF	32/64 位定时器 4	674

表 2-4. 存储器映射 (续)

起始	结束	描述	详见页
0x4004.F000	0x4004.FFFF	32/64 位定时器 5	674
0x4005.0000	0x4005.0FFF	USB	1043
0x4005.1000	0x4005.7FFF	保留	-
0x4005.8000	0x4005.8FFF	GPIO 端口 A ( AHB 槽 )	608
0x4005.9000	0x4005.9FFF	GPIO 端口 B ( AHB 槽 )	608
0x4005.A000	0x4005.AFFF	GPIO 端口 C ( AHB 槽 )	608
0x4005.B000	0x4005.BFFF	GPIO 端口 D ( AHB 槽 )	608
0x4005.C000	0x4005.CFFF	GPIO 端口 E ( AHB 槽 )	608
0x4005.D000	0x4005.DFFF	GPIO 端口 F ( AHB 槽 )	608
0x4005.E000	0x4005.EFFF	GPIO 端口 G ( AHB 槽 )	608
0x4005.F000	0x4005.FFFF	GPIO 端口 H ( AHB 槽 )	608
0x4006.0000	0x4006.0FFF	GPIO 端口 J ( AHB 槽 )	608
0x4006.1000	0x4006.1FFF	GPIO 端口 K ( AHB 槽 )	608
0x4006.2000	0x4006.2FFF	GPIO 端口 L ( AHB 槽 )	608
0x4006.3000	0x4006.3FFF	GPIO 端口 M ( AHB 槽 )	608
0x4006.4000	0x4006.4FFF	GPIO 端口 N ( AHB 槽 )	608
0x4006.5000	0x4006.5FFF	GPIO 端口 P ( AHB 槽 )	608
0x4006.6000	0x400B.FFFF	保留	-
0x400C.0000	0x400C.0FFF	I <sup>2</sup> C 4	951
0x400C.1000	0x400C.1FFF	I <sup>2</sup> C 5	951
0x400C.2000	0x400F.8FFF	保留	-
0x400F.9000	0x400F.9FFF	系统异常模块	467
0x400F.A000	0x400F.BFFF	保留	-
0x400F.C000	0x400F.CFFF	休眠模块	485
0x400F.D000	0x400F.DFFF	Flash 存储器控制	511
0x400F.E000	0x400F.EFFF	系统控制	222
0x400F.F000	0x400F.FFFF	μDMA	554
0x4010.0000	0x41FF.FFFF	保留	-
0x4200.0000	0x43FF.FFFF	0x4000.0000 到 0x400F.FFFF 的位带别名	-
0x4400.0000	0xDFFF.FFFF	保留	-
专用的外设总线			
0xE000.0000	0xE000.0FFF	仪表跟踪宏单元 ( ITM )	71
0xE000.1000	0xE000.1FFF	数据观察点和跟踪 ( DWT )	71
0xE000.2000	0xE000.2FFF	Flash 修补和断点 ( FPB )	71
0xE000.3000	0xE000.DFFF	保留	-
0xE000.E000	0xE000.EFFF	Cortex-M4F 外设 ( SysTick、NVIC、MPU、FPU 和 SCB )	128
0xE000.F000	0xE003.FFFF	保留	-
0xE004.0000	0xE004.0FFF	跟踪端口接口单元 ( TPIU )	71
0xE004.1000	0xE004.1FFF	嵌入式跟踪宏单元 ( ETM )	71
0xE004.2000	0xFFFF.FFFF	保留	-

### 2.4.1 存储器区、类型和属性

存储器映射和 MPU 的编程将存储器映射分割成几个区域。每个区被定义了存储类型且有些区有附件的存储器属性。存储类型和属性决定访问该区的行为。

存储类型：

- 普通:处理器为了效率可重新排序和不确定的读操作。
- 设备：处理器保存传送顺序并严格依照顺序和其它设备或严格排序存储器交换信息。
- 严格排序处理器保存传送顺序并严格依照顺序和其它设备交换信息。

设备和严格排序存储器的顺序要求不同，这就意味着，存储系统可以将写操作缓冲到设备存储器，而不可缓冲到严格排序存储器。

附件的存储属性是永不执行区 (XN)。意味着处理器阻止指令访问。只要在 XN 区执行指令就会产生故障异常。

### 2.4.2 存储器访问存储系统顺序

大多数的存储器访问时通过具体的存储器访问指令，存储系统并不能保证访问的顺序和指令的编程顺序一致，提供的顺序不影响指令序列的行为。通常，如果程序的正常执行依赖于两次存储器访问依照编程顺序，则软件要在两次存储器访问指令之间设置存储器阻碍指令。（参见“存储器访问的软件顺序”在 95 页）。

然而，存储器系统保证设备和严格顺序存储之间的访问顺序。两条存储器访问指 A1 和 A 2，如果 A1 和 A2 都访问设备或者严格顺序存储器，并且在编程顺序上 A1 在 A2 前边，则 A1 将一直在 A2 前边被获取。

### 2.4.3 存储器访问行为

表 2-5 在 94 页 示出存储器映射中每个区的访问行为。关于存储器类型和 XN 属性的具体信息，参见“存储器区、类型和属性”在 94 页。Stellaris® 设备保留如下表所示的地址范围。更多信息，参见表 2-4 在 91 页。

表 2-5. 存储器访问行为

地址范围	存储器区域	存储器类型	从不执行 (XN)	描述
0x0000.0000 - 0x1FFF.FFFF	代码	正常	-	这个可执行区域用于存放程序代码。数据也可以保存到这里。
0x2000.0000 - 0x3FFF.FFFF	SRAM	正常	-	这个可执行区域用于存放数据。代码也可以保存在这里。这个区域包括了位带和位带别名区（参见表 2-6 在 96 页）。
0x4000.0000 - 0x5FFF.FFFF	外设	设备	XN	这个区域包括了位带和位带别名区（参见表 2-7 在 96 页）。
0x6000.0000 - 0x9FFF.FFFF	外部 RAM	正常	-	这个可执行区域用于存放数据。
0xA000.0000 - 0xDFFF.FFFF	外部设备	设备	XN	这个区域用作外部器件存储器。
0xE000.0000- 0xE00F.FFFF	专用外设总线	严格排序	XN	这个区域包括 NVIC、系统定时器和系统控制模块。
0xE010.0000- 0xFFFF.FFFF	保留	-	-	-

CODE、SRAM 和外部 RAM 可以保存程序。然而，推荐在 CODE 区保存程序，因为 Cortex-M4F 分离总线可以同时读取和访问数据。

MPU 可以不理睬默认存储器访问。更多信息，参见“存储器保护单元 (MPU)”在 121 页。  
Cortex-M4F 会在执行前预取指令，并且它会以推测的方式从分支目标地址进行预取。

#### 2.4.4 存储器访问的软件顺序

程序流中的指令顺序并不总能保证相应的存储器传送顺序，有以下几个原因：

- 为提高效率，处理器能够记录一些存储器访问，提供这些并不影响指令序列的行为。
- 处理器有多个总线接口。
- 存储器映射中的存储器或设备有不同的等待状态。
- 有些存储器访问被缓冲或者是不确定的。

“存储器访问存储系统顺序”在 94 页描述了存储系统保证存储器访问顺序的几种情况。然而，如果存储器访问顺序是关键性的，软件必须包含存储器边界指令强制顺序。Cortex-M4F 包含以下存储器边界指令：

- 数据存储器边界 (DMB) 指令确保先执行尚未完成的存储器传输，再执行后续存储器传输。
- 数据同步边界 (DSB) 指令确保先执行尚未完成的存储器传输，再执行后续指令。
- 指令同步边界 (ISB) 指令确保所有已完成的存储器传输的影响能被后续指令识别。

存储器边界指令可在以下情况中使用：

- MPU 编程
  - 如果 MPU 设置改变，且改变必须在下条指令时有效，应使用 DSB 指令以确保 MPU 在指令切换结束后立即生效。
  - 如果使用分支或者调用进入 MPU 配置代码，应使用 ISB 指令确保新的 MPU 设置在编程 MPU 区后立即生效。如果 MPU 配置代码使用异常机制进入，则不需要 ISB 指令。
- 向量表
 

如果程序改变了向量表的入口并启用了相应的异常，应在两次操作之间使用 DMB 指令。当异常在启用后立即发生时，DMB 指令可确保处理器使用新的异常向量。
- 自修改代码
 

如果程序包含自修改代码，应在程序代码修改后立即使用 ISB 指令。ISB 指令确保在执行后续指令时使用更新的程序。
- 存储器映射切换
 

如果系统包含存储器映射切换机制，应在程序中切换存储器映射之后使用 DSB 指令。DSB 指令确保在执行后续指令时使用更新的存储器映射。
- 动态异常优先级改变
 

当异常处于挂起或者活动状态时，如果其优先级需要更改，应在更改之后使用 DSB 指令。在 DSB 指令执行之后，更改将生效。

访问严格顺序存储器时（例如系统控制块），不需要使用 DMB 指令。

有关存储器边界指令的更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。

## 2.4.5 位带区

位带区映在位带别名区中的每个字到带区中的单个位。位带区占用SRAM和外设存储器区中最少1MB空间。对32-MBSRAM别名区的访问映射到SRAM中1-MB的位带区，如表2-6在96页所示。对32-MB外设别名区的访问映射到1-MB的外设位带区，如表2-7在96页所示。关于位带区的特定地址范围，参见表2-4在91页。

**注意：**对在SRAM或外设位带别名区中一个字的访问映射到SRAM或外设位带区中的一个位。

对位带区的一个字访问结果是对相应存储器的一个字访问，类似半字或字节访问。这样就与相应外设的访问要求相符合。

**表 2-6. SRAM 存储器位带区**

地址范围	存储器区域	指令和数据访问
0x2000.0000 - 0x200F.FFFF	SRAM 位带区	对这个存储器范围的直接访问行为如同对SRAM存储器的访问。但是该区域也可通过位带别名进行位寻址。
0x2200.0000 - 0x23FF.FFFF	SRAM 位带别名	对这个区域的数据访问被重新映射到带区。一个写操作被执行行为读-修改-写。指令访问没有重新映射。

**表 2-7. 外设存储器位带区**

地址范围	存储器区域	指令和数据访问
0x4000.0000 - 0x400F.FFFF	外设位带区	对这个存储器范围的直接访问行为如同对外设存储器的访问。但是该区域也可通过位带别名进行位寻址。
0x4200.0000 - 0x43FF.FFFF	外设位带别名	对这个区域的数据访问被重新映射到带区。一个写操作被执行行为读-修改-写。指令访问不被允许。

下面的公式演示了别名区和位带区的映射关系：

$$\text{bit\_word\_offset} = (\text{byte\_offset} \times 32) + (\text{bit\_number} \times 4)$$

$$\text{bit\_word\_addr} = \text{bit\_band\_base} + \text{bit\_word\_offset}$$

其中：

**bit\_word\_offset**  
在位带区中的目标位的位置。

**bit\_word\_addr**  
在别名区中由目标位映射的字的地址。

**bit\_band\_base**  
别名区的起始地址。

**byte\_offset**  
字节在包含目标位的位带区中的编号。

**bit\_number**  
目标位的位置，0-7。

图2-4在97页示出了SRAM位带别名区和SRAM位带区映射的例子：

- 在0x23FF.FFE0处的别名字映射到了位带区中的0x200F.FFFF的第0位：

$$0x23FF.FFE0 = 0x2200.0000 + (0x000F.FFFF * 32) + (0 * 4)$$



- 在 0x23FF.FFFC 处的别名字映射到了位带区中的 0x200F.FFFF 的第 7 位：

$$0x23FF.FFFC = 0x2200.0000 + (0x000F.FFFF * 32) + (7 * 4)$$

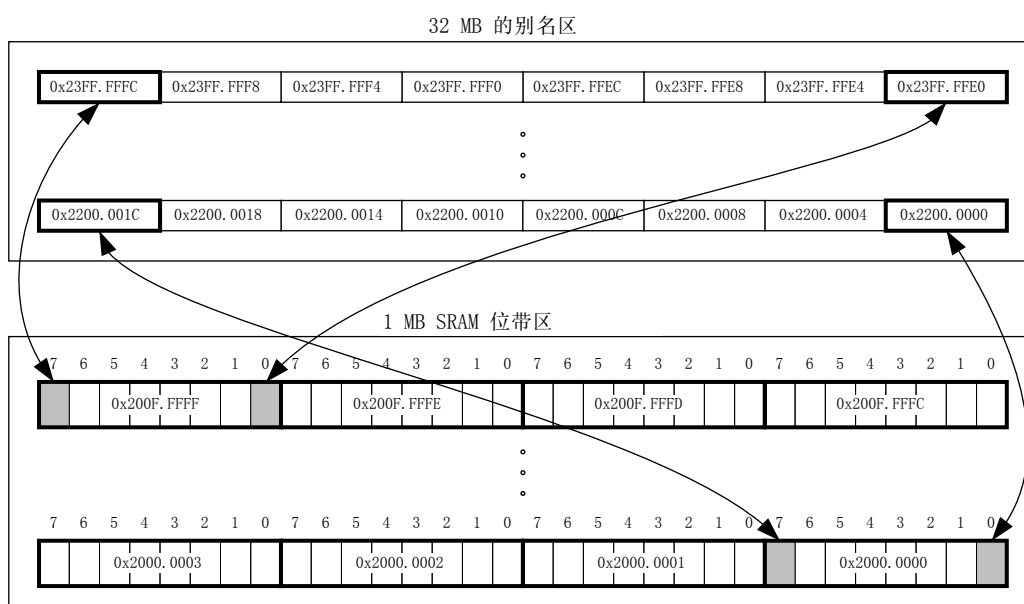
- 在 0x2200.0000 处的别名字映射到了位带区中的 0x2000.0000 的第 0 位：

$$0x2200.0000 = 0x2200.0000 + (0 * 32) + (0 * 4)$$

- 在 0x2200.001C 处的别名字映射到了位带区中的 0x2000.0000 的第 7 位：

$$0x2200.001C = 0x2200.0000 + (0 * 32) + (7 * 4)$$

图 2-4. 位带映射



### 2.4.5.1 直接访问别名区

在别名区写一个字会更新位带区的一个位。

在别名区写入的字，其值的第0位决定了位带区对应位的值。写入值的第0位置位使位带位为1，写入值的第0位清零使位带位为0。

别名区字的位 31:1 对位带区对应位没有影响。写入0x01和写入0xFF的影响是一样的。写入0x00和写入0x0E的影响是一样的。

当在别名区写入一个字时，0x0000.0000 表示位带区中的对应位清零，0x0000.0001 表示位带区中的对应位置位。

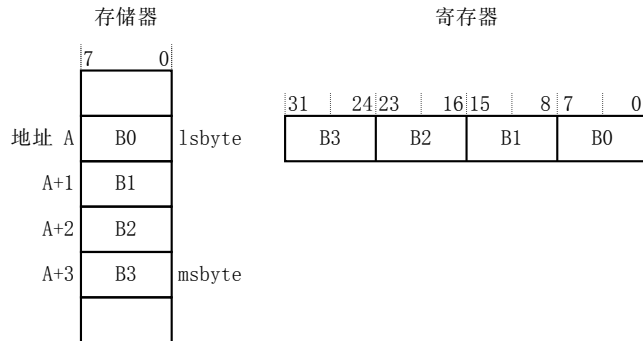
### 2.4.5.2 直接访问位带区

“存储器访问行为”在 94页 描述了对位带区的字节、半字或字进行直接访问的行为。

## 2.4.6 数据保存

处理器将存储器看做一个从零开始按增加顺序编号的线性字节集合。例如，字节0-3占据第一个存储字，字节4-7占据第二个存储字。数据以小端法 (little-endian) 格式存储，字的最不重要字节 (lsbyte) 存储在最低编号字节，最重要字节 (msbyte) 存储在最高编号字节。图 2-5 在 98 页 示出了数据是如何存储的。

图 2-5. 数据保存



## 2.4.7 同步原语

Cortex-M4F 指令集包含一对同步原语，可以提供一个没有阻碍的机制，使一个线程或进程能用于获取对存储位置的专有访问。软件可以使用这些原语执行有保证的读-修改-写存储器更新序列，或者用作信号量机制。

一对同步原语包括：

- 一个下载专用指令，用于读取一个存储器位置的值并请求对该位置的专有访问。
- 一个保存专用指令，用于尝试写入相同的存储器位置并想一个寄存器返回一个状态位。如果这个状态位为 0，表示该线程或进程获得了对存储器的专有访问，写入成功；如果这个状态位为 1，表示该线程或进程没有获得对存储器的专有访问，写入没有执行。

这对下载专用和保存专用指令是：

- 字指令 LDREX 和 STREX
- 半字指令 LDREXH 和 STREXH
- 字节指令 LDREXB 和 STREXB

软件必须使用对应的下载专用指令和保存专用指令。

要对存储器位置执行专有的读-修改-写操作，软件必须：

1. 使用一个下载专用指令读取该位置的值。
2. 按照需要修改该值。
3. 使用保存专用指令来尝试将新值写入存储器位置。
4. 测试返回的状态位。

如果状态位清零，读-修改-写操作成功完成。如果状态位为 0，不执行写操作，表示第 1 步返回的值可能已过时。软件必须重试整个读-修改-写序列。

软件可以使用同步原语来实现信号量，如下：

1. 使用一个下载专用指令读取信号量地址，检测信号量是否自由。
2. 如果信号量是自由的，使用保存专用指令向信号量地址写入获取值。
3. 如果第2步返回的状态位表示保存成功，那么软件已经获取了信号量。但是，如果保存失败，那么可能在软件执行第1步后另外的进程已经获取了信号量。

Cortex-M4F 包含一个专用的访问监视器，处理器执行一个下载专用指令后，监视器会对这个事件做一个标签。如果发生下列情况，处理器会移除这个专用访问标签。

- 执行一条 CLREX 指令。
- 执行一条 Store-Exclusive 指令，不管是否写成功。
- 发生异常时，表示处理器可以在不同线程间解决信号量冲突。

有关同步原语指令的更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。

## 2.5 异常模式

ARM Cortex-M4F 处理器和嵌套矢量中断控制器 (NVIC) 在处理器模式对所有的异常进行优先级划分和处理。异常发生时处理器状态被自动存储到堆栈，中断服务程序 (ISR) 结束时又自动被恢复。向量的读取与状态保存并行，高效率进入中断。处理器支持尾链 (tail-chaining)，这样使得执行背靠背中断不需要重叠的状态保存和恢复。

表 2-8 在 101 页 列出了所有的异常类型。软件可在七个异常（系统处理程序）以及 96 中断（在表 2-9 在 101 页中列出）上设置 8 个优先级。

系统处理程序的优先级是通过 NVIC 系统处理程序优先级 n (SYSPRIn) 寄存器来设置的。中断是通过 NVIC 中断设置启用 n (ENn) 寄存器来启用的，并且由 NVIC 中断优先级 n (PRIn) 寄存器来区分其优先等级。优先级可以被分组为先发优先级和子优先级。在“嵌套向量中断控制器 (NVIC)”在 119 页中描述了所有中断寄存器。

在内部用户可编程的最高优先级 (0) 是第 4 优先级，按顺序在复位、非屏蔽中断 (NMI) 和硬件故障之后。注意 0 是所有可编程优先级的默认优先级。

**重要：** 在一次写操作清除中断源后，对于 NVIC 来说，需要花费几个处理器周期才能看到中断源被禁止。所以如果在中断处理程序中最后清除中断，有可能中断处理程序结束了但是 NVIC 看到中断仍然有效，导致错误的重新进入中断处理程序。这种情况可以避免，或者通过在中断处理程序开始时清除中断源，或者在写操作清除中断源后执行一个读或写操作 (刷新写缓冲器)。

关于异常和中断的具体信息，参见“嵌套向量中断控制器 (NVIC)”在 119 页。

### 2.5.1 异常状态

每种异常都处于下列状态之一：

- 不活动的. 异常是不活动的，也不是挂起的。
- 挂起的. 异常正在等待处理器处理。来自外设或软件的中断请求可以将相应的中断变为挂起状态。
- 活动的. 处理器正在处理的异常，并且异常没有结束。

注意： 一个异常处理程序可以中断另一个异常处理程序的执行。就此来说，两个异常都处于活动状态。

- 活动的和挂起的. 异常正被处理器处理，并且有一个挂起的异常来自相同的源。

## 2.5.2 异常类型

异常类型有：

- **复位.** 上电或热复位会引起复位。异常模式将复位看做一种特殊形式的异常。当复位有效时，在指令的任何时刻，处理器的操作都会停止。当复位不再有效时，从向量表中复位入口的地址重新开始执行。在线程模式中该执行是特权执行。
- **NMI.** 一个非屏蔽中断 (NMI) 可以使用 NMI 信号来发出通知或者使用中断控制及状态 (INTCTRL) 寄存器由软件触发。除了复位，该异常有最高的优先级。NMI 永久启用并拥有一个固定的优先级 -2。NMI 的激活不能被其他任何异常屏蔽或阻止，也不能被除复位外的其他任何异常抢占。
- **硬故障.** 硬故障是一个异常，它的发生是由于异常处理期间有错误，或者异常不能被任何异常机制管理。硬故障拥有一个固定的优先级 -1，表明它的优先级高于任何可配置的优先级。
- **存储器管理故障.** 存储器管理故障是一个异常，它的发生是由于存在与存储器保护相关的故障（包括访问冲突和不匹配）。在处理指令存储器和数据存储器时，MPU 或固定存储器保护限制决定了该故障。该故障用来取消指令对不执行 (XN) 存储区域的访问，即使 MPU 是禁止的。
- **总线故障.** 总线故障是一个异常，它的发生是由于在处理指令或数据存储器时发生了与存储器相关的故障，如预取错误或存储器访问故障。该故障可以被启用或禁止。
- **使用故障.** 使用故障是一个异常，它的发生是由于指令执行的相关故障，如：
  - 一个未定义的指令
  - 一次非法的未对齐访问
  - 指令执行时的无效状态
  - 异常返回错误
 当内核被正确配置后，在字或半字存储器访问未对齐的地址或除以0都会引起一个使用故障。
- **SVCall.** 系统调用 (SVC) 是一个异常，它由 SVC 指令触发。在 OS 环境，应用程序可以使用 SVC 指令来访问 OS 内核函数和器件驱动。
- **调试监视器.** 这个异常是由于调试监视器（没有停止时）引起的。该异常只有在启用时才激活。如果该异常的优先级低于当前的动作，那么它不会激活。
- **PendSV.** PendSV 是一个对系统级服务发出的请求，它可以挂起，并由中断驱动。在 OS 环境，当没有其它异常活动时，可使用 PendSV 作为背景切换。PendSV 使用中断控制及状态 (INTCTRL) 触发。
- **SysTick.** SysTick 异常是在系统定时器启用中断时，系统定时器达到 0 时产生的。软件也可以通过中断控制及状态 (INTCTRL) 寄存器来产生一个 SysTick 异常。在 OS 环境，处理器可以使用该异常作为系统时标。
- **中断 (IRQ).** 中断（即 IRO）是一个异常，它由外设标记，或者通过软件请求并由 NVIC（划分优先顺序的）反馈产生。所有的中断与指令执行都是异步的。在系统中，外设使用中断与处理器通信。表 2-9 在 101 页列出了 LM4F232H5QD 控制器上的中断。

对于异步异常，除复位外，处理器可以在异常触发和处理器进入异常处理程序之间执行其他指令。

特权软件可以禁止表 2-8 在 101 页上显示的可配置优先级的异常（参见 167 页上的 SYSHNDCTRL 寄存器和 138 页上的 DIS0 寄存器）。

关于硬故障、存储器管理故障、总线故障和使用故障的更多信息，参见“故障处理”在 108 页。

表 2-8. 异常类型

异常类型	向量号	优先级 <sup>a</sup>	向量地址或偏移量 <sup>b</sup>	激活
-	0	-	0x0000.0000	栈顶在复位时通过向量表的首入口加载。
复位	1	-3 (最高)	0x0000.0004	异步
不可屏蔽的中断(NMI)	2	-2	0x0000.0008	异步
硬故障	3	-1	0x0000.000C	-
存储器管理	4	可编程 <sup>c</sup>	0x0000.0010	同步
总线故障	5	可编程 <sup>c</sup>	0x0000.0014	精确时同步，不精确时异步
使用故障	6	可编程 <sup>c</sup>	0x0000.0018	同步
-	7-10	-	-	保留
SVCcall	11	可编程 <sup>c</sup>	0x0000.002C	同步
调试监视器	12	可编程 <sup>c</sup>	0x0000.0030	同步
-	13	-	-	保留
PendSV	14	可编程 <sup>c</sup>	0x0000.0038	异步
SysTick	15	可编程 <sup>c</sup>	0x0000.003C	异步
中断	16及其以上	可编程 <sup>d</sup>	0x0000.0040 及其以上	异步

a. 0 是所有可编程优先级的默认优先级。

b. 请参考“向量表”在 104 页。

c. 参见 164 页上的 SYSPRI1 寄存器。

d. 参见 146 页上的 PRIn 寄存器。

表 2-9. 中断

向量号	中断编号（在中断寄存器中的位）	向量地址或偏移量	描述
0-15	-	0x0000.0000 - 0x0000.003C	处理器异常
16	0	0x0000.0040	GPIO 端口 A
17	1	0x0000.0044	GPIO 端口 B
18	2	0x0000.0048	GPIO 端口 C
19	3	0x0000.004C	GPIO 端口 D
20	4	0x0000.0050	GPIO 端口 E
21	5	0x0000.0054	UART0
22	6	0x0000.0058	UART1
23	7	0x0000.005C	SSI0
24	8	0x0000.0060	I <sup>2</sup> C0
25	9	0x0000.0064	PWM0 故障
26	10	0x0000.0068	PWM0 发生器 0
27	11	0x0000.006C	PWM0 发生器 1

表 2-9. 中断 (续)

向量号	中断编号 (在中断寄存器中的位)	向量地址或偏移量	描述
28	12	0x0000.0070	PWM0 发生器 2
29	13	0x0000.0074	QEI0
30	14	0x0000.0078	ADC0 序列 0
31	15	0x0000.007C	ADC0 序列 1
32	16	0x0000.0080	ADC0 序列 2
33	17	0x0000.0084	ADC0 序列 3
34	18	0x0000.0088	看门狗定时器 0 和 1
35	19	0x0000.008C	16/32 位定时器 0A
36	20	0x0000.0090	16/32 位定时器 0B
37	21	0x0000.0094	16/32 位定时器 1A
38	22	0x0000.0098	16/32 位定时器 1B
39	23	0x0000.009C	16/32 位定时器 2A
40	24	0x0000.00A0	16/32 位定时器 2B
41	25	0x0000.00A4	模拟比较器 0
42	26	0x0000.00A8	模拟比较器 1
43	27	0x0000.00AC	模拟比较器 2
44	28	0x0000.00B0	系统控制
45	29	0x0000.00B4	Flash 存储器控制
46	30	0x0000.00B8	GPIO 端口 F
47	31	0x0000.00BC	GPIO 端口 G
48	32	0x0000.00C0	GPIO 端口 H
49	33	0x0000.00C4	UART2
50	34	0x0000.00C8	SSI1
51	35	0x0000.00CC	16/32 位定时器 3A
52	36	0x0000.00D0	16/32 位定时器 3B
53	37	0x0000.00D4	I <sup>2</sup> C1
54	38	0x0000.00D8	QEI1
55	39	0x0000.00DC	CAN0
56	40	0x0000.00E0	CAN1
57-58	41-42	-	保留
59	43	0x0000.00EC	休眠模块
60	44	0x0000.00F0	USB
61	45	0x0000.00F4	PWM 发生器 3
62	46	0x0000.00F8	μDMA 软件
63	47	0x0000.00FC	μDMA 出错
64	48	0x0000.0100	ADC1 序列 0
65	49	0x0000.0104	ADC1 序列 1
66	50	0x0000.0108	ADC1 序列 2
67	51	0x0000.010C	ADC1 序列 3
68-69	52-53	-	保留
70	54	0x0000.0118	GPIO 端口 J

表 2-9. 中断 (续)

向量号	中断编号 (在中断寄存器中的位)	向量地址或偏移量	描述
71	55	0x0000.011C	GPIO 端口 K
72	56	0x0000.0120	GPIO 端口 L
73	57	0x0000.0124	SSI2
74	58	0x0000.0128	SSI3
75	59	0x0000.012C	UART3
76	60	0x0000.0130	UART4
77	61	0x0000.0134	UART5
78	62	0x0000.0138	UART6
79	63	0x0000.013C	UART7
80-83	64-67	0x0000.0140 - 0x0000.014C	保留
84	68	0x0000.0150	I <sup>2</sup> C2
85	69	0x0000.0154	I <sup>2</sup> C3
86	70	0x0000.0158	16/32 位定时器 4A
87	71	0x0000.015C	16/32 位定时器 4B
88-107	72-91	0x0000.0160 - 0x0000.01AC	保留
108	92	0x0000.01B0	16/32 位定时器 5A
109	93	0x0000.01B4	16/32 位定时器 5B
110	94	0x0000.01B8	32/64 位定时器 0A
111	95	0x0000.01BC	32/64 位定时器 0B
112	96	0x0000.01C0	32/64 位定时器 1A
113	97	0x0000.01C4	32/64 位定时器 1B
114	98	0x0000.01C8	32/64 位定时器 2A
115	99	0x0000.01CC	32/64 位定时器 2B
116	100	0x0000.01D0	32/64 位定时器 3A
117	101	0x0000.01D4	32/64 位定时器 3B
118	102	0x0000.01D8	32/64 位定时器 4A
119	103	0x0000.01DC	32/64 位定时器 4B
120	104	0x0000.01E0	32/64 位定时器 5A
121	105	0x0000.01E4	32/64 位定时器 5B
122	106	0x0000.01E8	系统异常 (不精确)
123-124	107-108	-	保留
125	109	0x0000.01F4	I <sup>2</sup> C4
126	110	0x0000.01F8	I <sup>2</sup> C5
127	111	0x0000.01FC	GPIO 端口 M
128	112	0x0000.0200	GPIO 端口 N
129-131	113-115	-	保留
132	116	0x0000.0210	GPIO 端口 P (摘要或 P0)
133	117	0x0000.0214	GPIO 端口 P1
134	118	0x0000.0218	GPIO 端口 P2

表 2-9. 中断 (续)

向量号	中断编号 (在中断寄存器中的位)	向量地址或偏移量	描述
135	119	0x0000.021C	GPIO 端口 P3
136	120	0x0000.0220	GPIO 端口 P4
137	121	0x0000.0224	GPIO 端口 P5
138	122	0x0000.0228	GPIO 端口 P6
139	123	0x0000.022C	GPIO 端口 P7
140-149	124-133	-	保留
150	134	0x0000.0258	PWM1 发生器 0
151	135	0x0000.025C	PWM1 发生器 1
152	136	0x0000.0260	PWM1 发生器 2
153	137	0x0000.0264	PWM1 发生器 3
154	138	0x0000.0268	PWM1 故障
63	47	-	保留

### 2.5.3 异常处理程序

处理器处理异常是通过使用：

- 中断服务程序 (ISR). 中断 (IRQx) 是由 ISR 处理的异常。
- 故障处理程序. 硬件障、存储器管理故障、使用故障以及总线故障都是故障异常，由故障处理程序处理。
- 系统处理程序. NMI、PendSV、SVCall、SysTick 和故障异常都是系统异常，由系统处理程序处理。

### 2.5.4 向量表

向量表包含了堆栈指针的复位值和开始地址，对于所有的异常处理程序也可称作异常向量。向量表由表 2-8 在 101 页中示出的向量地址或偏移量组成。图 2-6 在 105 页示出向量表中异常向量的次序。每个向量的最低位必须为 1，表示异常处理程序是 Thumb 码。



图 2-6. 向量表

异常号	IRQ 号	偏移量	向量
154	138	0x0268	IRQ131
°		°	°
°		°	°
°		°	°
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	SysTick
14	-2	0x003C	PendSV
13		0x0038	保留
12			保留用于调试
11	-5	0x002C	SVCa11
10			保留
9			保留
8			保留
7			保留
6	-10	0x0018	用法故障
5	-11	0x0014	总线故障
4	-12	0x0010	存储器管理故障
3	-13	0x000C	硬故障
2	-14	0x0008	NMI
1		0x0004	复位
		0x0000	初始的 SP 值

系统复位时，向量表固定在地址 0x0000.0000。特权软件可以通过向向量表偏移量 (VTABLE) 寄存器进行写操作，来将向量表的开始地址重新定位为不同的存储器地址，范围是 0x0000.0400 至 0x3FFF.FC00（请参考“向量表”在 104 页）。请注意，在配置 VTABLE 寄存器时，偏移量必须在 1024 字节的边界对齐。

### 2.5.5 异常优先级

如表 2-8 在 101 页所示，所有的异常都有其相关的优先级，优先级的值低表示其优先级高，除复位、硬件复位和 NMI 外的所有异常都可以配置优先级。如果软件没有配置任何优先级，那么可配置优先级的所有异常的优先级值为 0。关于配置异常优先级的更多信息，参见 164 页和 146 页。

**注意：** 对于 Stellaris 的可配置优先级值的范围是 0-7。这就意味着带有负的优先级值的复位、硬故障和 NMI 等异常，总是比其他异常有更高的优先级。

例如，分配一个较高的优先级值给 IRQ[0] 同时分配一个较低的优先级值给 IRQ[1]，表示 IRQ[1] 比 IRQ[0] 有更高的优先级。如果 IRQ[1] 和 IRQ[0] 都有效，那么 IRQ[1] 先于 IRQ[0] 被处理。

如果多个挂起的异常有相同的优先级，那么异常号最低的优先。例如，如果 IRQ[1] 和 IRQ[0] 都挂起并且它们有相同的优先级，那么 IRQ[0] 先于 IRQ[1]。

当处理程序正在执行一个异常处理程序时，如果有个更高优先级的异常发生，那么它将取代当前的异常处理程序。如果有一个同样优先级的异常发生，那么不管其异常号如何都不会取代当前的异常处理程序。但是，新中断的状态变为挂起。

## 2.5.6 中断优先级分组

为提高系统中对中断优先级的控制，NVIC支持优先级分组。这个分组将中断优先级寄存器入口分为两个区域。

- 高区域定义组的优先级
- 低区域定义在同一组内的子优先级

只有组的优先级可以决定中断异常的取代。当处理器正在执行一个中断异常处理程序时，同优先级组的另一个中断不能取代当前的处理程序。

如果多个挂起的中断处于相同的优先级组，那么子优先级决定处理的顺序。如果多个挂起的中断有相同的组优先级和子优先级，那么最低IRQ编号的中断先被处理。

关于将中断优先级域分为组优先级和子优先级的更多信息，参见 158页。

## 2.5.7 异常进入和返回

异常处理的描述使用以下术语：

- 抢占式. 当处理器正在执行一个异常处理程序时，如果另一个异常的优先级更高，那么它可以抢占当前正在执行的异常处理程序。关于中断抢占的更多信息，参见“中断优先级分组”在 106页。当一个异常抢占另一个异常时，它们称为嵌套异常。更多信息参见“异常进入”在 106页。
- 返回. 当异常处理程序完成，并且没有挂起的有足够优先级的异常被服务，并且完成的异常处理程序不是一个后到的异常时，发生返回。处理器弹出堆栈并恢复到中断发生前的状态。有关更多信息，请参考“异常返回”在 107页。
- 尾链. 使用尾链机制可加速异常处理。当一个异常处理程序完成时，如果有一个挂起的异常满足进入的要求，堆栈弹出将被跳过并且控制权直接转移到新的异常处理程序。
- 后到. 使用后到机制可加速抢占。如果在先前的异常正在保存状态期间有一个更高优先级的异常发生，那么处理器会切换到处理更高优先级的异常并开始为该异常取出向量。后到不会影响状态保存，因为对于前后两个异常来说，要保存的状态是一样的。所以，状态保存会没有中断的持续进行。处理器可以在直到先前异常处理程序的第一条指令进入到执行阶段时再接受后到的异常。从后到的异常处理程序返回时，正常的尾链规则有效。

### 2.5.7.1 异常进入

异常进入发生的条件是：有一个挂起的有足够优先级的异常，同时处理器处于线程模式，或者新的异常优先级高于正在处理的异常，这种情况新的异常将取代原先的异常。

当一个异常取代另一个异常时，它们是嵌套的。

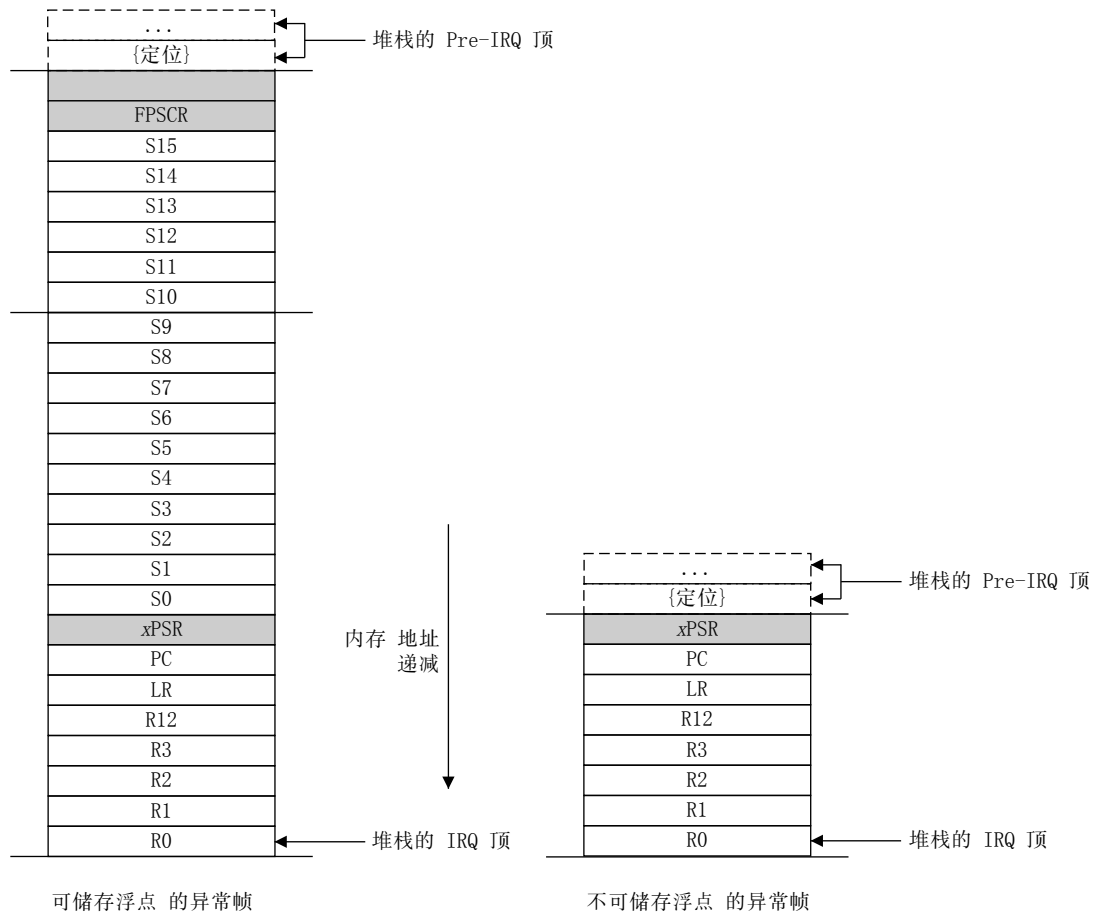
足够的优先级表示异常比屏蔽寄存器设置的任何限制都要更优先 (参见 84页 上的 PRIMASK；85页 上的 FAULTMASK；以及 86页 上的 BASEPRI)。比它的优先级低的异常会挂起而不是由处理器处理。

当处理器取得一个异常时，除非该异常是尾链的或后到的异常，否则处理器会将信息压入当前堆栈中。这个操作即 *堆栈*，8 个数据字的结构即 *堆栈框*。

当使用浮点程序时，Cortex-M4F处理器自动将该建立的浮点状态压入异常入口堆栈。在图 2-7 在 107页示出了当浮点状态保存在堆栈中并用作中断或异常的结果时，Cortex-M4F 堆栈结构的布局。

注意： 如果没有分配浮点状态的堆栈空间，该堆栈框就与不带有 FPU 的 ARMv7-M 具有相同结构。图 2-7 在 107页 也示出了堆栈框。

图 2-7. 异常堆栈框



入栈操作完成后，堆栈指针立即指向堆栈框的最低地址。

堆栈框包含返回地址，它是被中断程序的下一条指令的地址。当异常返回时这个地址重新载入 PC 以便被中断的程序重新开始。

当入栈操作进行时，处理器并行从向量表读取异常处理程序的开始地址。当入栈操作完成时，处理器开始执行异常处理程序。同时，处理器将 EXC\_RETURN 值写入 LR，表明哪个堆栈指针与堆栈框相对应，以及处理器在进入异常前处于何种操作模式。

如果在异常进入期间没有更高优先级的异常发生，处理器开始执行异常处理程序并自动将相应挂起的中断状态更改为活动的。

如果在异常进入期间有另一个更优先的异常发生，即后到，处理器开始执行后到的异常处理程序，并且不改变先前异常的挂起状态。

### 2.5.7.2 异常返回

当处理器处于处理器模式时，系统将发生异常返回，并执行下列指令之一，以将 EXC\_RETURN 值装载到 PC 中：

- 装载 PC 的 LDM 或 POP 指令
- 使用任意寄存器的 BX 指令

- 将 PC 作为目标的 LDR 指令

EXC\_RETURN 是异常进入时载入 LR 的值。异常机制依赖这个值来检测处理器什么时候完成异常处理程序。这个值的低 5 位提供了有关返回堆栈和处理模式的信息。表 2-10 在 108 页 示出了 EXC\_RETURN 值，包括其异常返回行为的描述。

EXC\_RETURN 的位 31:5 都将置位。当这个值载入 PC 时，它表示对于处理器异常已经完成，处理器开始执行适合的异常返回序列。

表 2-10. 异常返回行为

EXC_RETURN[31:0]	描述
0xFFFF.FFE0	保留
0xFFFF.FFE1	返回到处理器模式。 异常返回使用来自 MSP 的浮点状态。 返回后执行操作使用 MSP。
0xFFFF.FFE2 - 0xFFFF.FFE8	保留
0xFFFF.FFE9	返回到线程模式。 异常返回使用来自 MSP 的浮点状态。 返回后执行操作使用 MSP。
0xFFFF.FFEA - 0xFFFF.FFEC	保留
0xFFFF.FFED	返回到线程模式。 异常返回使用来自 MSP 的浮点状态。 返回后执行操作使用 PSP。
0xFFFF.FFEE - 0xFFFF.FFF0	保留
0xFFFF.FFF1	返回到处理器模式。 异常返回使用来自 MSP 的非浮点状态。 返回后执行操作使用 MSP。
0xFFFF.FFF2 - 0xFFFF.FFF8	保留
0xFFFF.FFF9	返回到线程模式。 异常返回使用来自 MSP 的非浮点状态。 返回后执行操作使用 MSP。
0xFFFF.FFFA - 0xFFFF.FFFC	保留
0xFFFF.FFFD	返回到线程模式。 异常返回使用来自 PSP 的非浮点状态。 返回后执行操作使用 PSP。
0xFFFF.FFFE - 0xFFFF.FFFF	保留

## 2.6 故障处理

故障是异常的一个子集（见“异常模式”在 99 页）。下面的条件产生一个故障：

- 在取指或载入向量表或访问数据时的总线错误。
- 内部检测出的错误，如没有定义的指令或尝试用一个 BX 指令更改状态。
- 尝试从一个标记为不可执行 (XN) 的存储区域执行指令。
- 由于权限冲突或尝试访问未管理区域而产生的 MPU 错误。

## 2.6.1 故障类型

表 2-11 在 109 页 示出了故障类型、用于处理故障的处理程序、相应的故障状态寄存器和指示故障发生的寄存器位。关于故障状态寄存器的更多信息，参见 170 页。

表 2-11. 故障

故障	处理程序	故障状态寄存器	位名字
读向量时的总线错误	硬故障	硬件故障状态 (HFAULTSTAT)	VECT
故障扩大到硬件故障	硬故障	硬件故障状态 (HFAULTSTAT)	FORCED
存取指令时，MPU 或默认的存储器不匹配	存储器管理故障	存储器管理故障状态 (MFAULTSTAT)	IERR <sup>a</sup>
存取数据时，MPU 或默认的存储器不匹配	存储器管理故障	存储器管理故障状态 (MFAULTSTAT)	DERR
异常堆栈时，MPU 或默认的存储器不匹配	存储器管理故障	存储器管理故障状态 (MFAULTSTAT)	MSTKE
异常退出堆栈时，MPU 或默认的存储器不匹配	存储器管理故障	存储器管理故障状态 (MFAULTSTAT)	MUSTKE
在浮点怠惰状态保存过程中，MPU 或默认的存储器不匹配	存储器管理故障	存储器管理故障状态 (MFAULTSTAT)	MLSPERR
异常堆栈时的总线错误	总线故障	总线故障状态 (BFAULTSTAT)	BSTKE
异常退出堆栈时的总线错误	总线故障	总线故障状态 (BFAULTSTAT)	BUSTKE
预取指令时的总线错误	总线故障	总线故障状态 (BFAULTSTAT)	IBUS
在浮点怠惰状态保存过程中，总线错误	总线故障	总线故障状态 (BFAULTSTAT)	BLSPE
精确数据总线错误	总线故障	总线故障状态 (BFAULTSTAT)	PRECISE
非精确数据总线错误	总线故障	总线故障状态 (BFAULTSTAT)	IMPRE
尝试访问协处理器	用法故障	使用故障状态 (UFAULTSTAT)	NOCP
没有定义的指令	用法故障	使用故障状态 (UFAULTSTAT)	UNDEF
尝试进入无效指令集状态 <sup>b</sup>	用法故障	使用故障状态 (UFAULTSTAT)	INVSTAT
无效的 EXC_RETURN 值	用法故障	使用故障状态 (UFAULTSTAT)	INVPC
非法的未对齐下载或存储	用法故障	使用故障状态 (UFAULTSTAT)	UNALIGN
除以 0	用法故障	使用故障状态 (UFAULTSTAT)	DIV0

a. 此故障出现在对 XN 区访问时（即使禁用了 MPU）。

b. 尝试使用一个非 Thumb 指令集的指令集，或者在 ICI 持续期间返回一个非下载存储复合指令。

## 2.6.2 故障扩大和硬件故障

除硬件故障外的所有故障异常都有可配置的异常优先级（参见 164 页 上的 SYSPRI1）。软件可以禁止执行这些故障的处理程序（参见 167 页 上的 SYSHNDCTRL）。

通常，异常优先级和异常屏蔽寄存器的值决定了处理器是否可以进入故障处理程序，一个故障处理程序是否可以取代另外一个故障处理程序，参见“异常模式”在 99 页。

在某些状况下，一个带有可配置优先级的故障可被看做一个硬件故障。该处理称为优先级扩大，该故障被描述为**扩大到硬故障**。扩大到硬件故障发生在：

- 故障处理程序引起了与它所服务的故障类型相同的故障。这种扩大到硬件故障的发生是因为故障处理程序不能取代自身，因为它的优先级与当前优先级一样。
- 故障处理程序引起了与它所服务的故障优先级相同或更低的故障。这种情况的发生是因为新的故障处理程序不能取代当前正在执行的故障程序。

- 一个异常引起了故障，该故障的的优先级等于或低于当前正在执行的异常。
- 一个故障发生了，但是该故障的处理程序没有启用。

当进入一个总线故障处理程序时，如果在堆栈入栈期间发生了一个总线故障，该总线故障不会扩大到硬件故障。因此如果一个损坏的堆栈引起了一个故障，即使该处理程序的入栈失败，故障处理程序依然会执行。故障处理程序运行，但是堆栈内容是损坏的。

注意：只有复位和 NMI 可以取代固定优先级的硬件故障。硬件故障可以取代除复位、NMI 或另外硬件故障之外的任何异常。

### 2.6.3 故障状态寄存器和故障地址寄存器

故障状态寄存器显示了故障原因。对于总线故障和存储器管理故障，故障地址寄存器显示了造成故障的操作要访问的地址，参见表 2-12 在 110 页。

表 2-12. 故障状态寄存器和故障地址寄存器

处理程序	状态寄存器名称	地址寄存器名称	寄存器描述
硬故障	硬件故障状态 (HFAULTSTAT)	-	176 页
存储器管理故障	存储器管理故障状态 (MFAULTSTAT)	存储器管理故障地址 (MMADDR)	170 页 177 页
总线故障	总线故障状态 (BFAULTSTAT)	总线故障地址 (FAULTADDR)	170 页 178 页
用法故障	使用故障状态 (UFAULTSTAT)	-	170 页

### 2.6.4 死锁

当处理器执行 NMI 或硬件故障处理程序时，如果一个硬件故障发生，那么处理器进入死锁状态。当处理器处于死锁状态时，它不执行任何指令。处理器将一直处于死锁状态，直至其置位、产生 NMI 或被调试器停止。

注意：如果死锁状态从 NMI 处理程序引发，那么随后的 NMI 不会使处理器离开死锁状态。

## 2.7 电源管理

Cortex-M4F 处理器可使用睡眠模式来减少功耗：

- 睡眠模式停止处理器时钟。
- 深度睡眠模式停止系统时钟并关闭 PLL 和 Flash 存储器。

系统控制 (SYSCTRL) 寄存器中的 SLEEPDEEP 位用于选择睡眠模式 (参见 160 页)。关于睡眠模式行为的更多信息，参见“系统控制”在 215 页。

本节描述进入睡眠模式的机制和从睡眠模式唤醒的条件，它们对与睡眠模式和深度睡眠模式都适用。

### 2.7.1 进入睡眠模式

本节描述使用软件将处理器进入一种睡眠模式的机制。

系统可以产生假造的唤醒事件，如调试操作可唤醒处理器。所以软件必须能够在该事件后将处理器返回到睡眠模式。一段程序可能有一个空循环来将处理器返回到睡眠模式。

### 2.7.1.1 等待中断

等待中断指令 WFI 会导致立即进入睡眠模式，除非唤醒条件为真（请参考“从WFI或睡眠中退出 (Sleep-on-Exit)唤醒”在 111 页）。当处理器执行一条 WFI 指令时，它将停止执行其他指令并进入睡眠模式。有关更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。

### 2.7.1.2 等待事件

等待事件指令 WFE 会导致以 1 位事件寄存器的值为条件进入睡眠模式。当处理器执行一条 WFE 指令时，它会检测事件寄存器。如果寄存器为 0，处理器停止执行指令并进入睡眠模式。如果寄存器为 1，处理器清零寄存器，接着继续执行指令而不进入睡眠模式。

如果事件寄存器为 1，处理器执行 WFE 指令时一定不会进入睡眠模式。通常，这种情况在执行 SEV 指令后发生。软件不能直接访问这个寄存器。

有关更多信息，请参考“Cortex™-M3/M4 Instruction Set 技术用户手册”。

### 2.7.1.3 睡眠中退出 (Sleep-on-Exit)

如果 SYSCTRL 寄存器中的 SLEEPEXIT 位置位，当处理器完成所有异常处理器程序的执行后，它将返回线程模式并立即进入睡眠模式。这种机制可用于异常发生时需要处理器运行的情况。

## 2.7.2 从睡眠模式唤醒

处理器唤醒的条件依赖于促使其进入睡眠模式的机制。

### 2.7.2.1 从WFI或睡眠中退出(Sleep-on-Exit)唤醒

通常，只有在 NVIC 检测到一个异常，并且该异常的优先级足够导致异常进入时，处理器才被唤醒。有些嵌入式系统可能会在处理器唤醒后执行中断处理程序前必须执行系统恢复任务。将 PRIMASK 位置位并将 FAULTMASK 位清零可推延进入中断处理程序。如果一个中断启用并且比当前异常的优先级高，该中断到来时处理器唤醒但直到处理器将 PRIMASK 清零后才执行中断处理程序。关于 PRIMASK 和 FAULTMASK 的更多信息，参见 84 页 和 85 页。

### 2.7.2.2 从WFE 唤醒

处理器如果检测到一个足够优先执行的异常，处理器会唤醒。

另外，如果 SYSCTRL 寄存器的 SEVONPEND 位置位，任何新挂起的中断，即使该中断被禁止或没有足够的优先级执行，都会触发一个事件并唤醒处理器。关于 SYSCTRL 的更多信息，参见 160 页。

## 2.7.3 唤醒中断控制器

唤醒中断控制器 (WIC) 是一种可检测到中断并将处理器从深度睡眠模式中唤醒的外设。WIC 仅在 SCR 寄存器中的 DEEPSLEEP 位置位时才启用（请参考 160 页）。

WIC 不可编程，也没有任何寄存器或用户接口。其完全根据硬件信号工作。

当启用 WIC 且处理器进入到深度睡眠模式时，系统中的电源管理单元可关闭大部分的 Cortex-M4F 处理器。其副作用就是停止了 SysTick 定时器。当 WIC 接收到中断，它需要一定的时钟周期来唤醒处理器以恢复其状态，然后才可以处理该中断。这也造成了在深度睡眠中的中断延迟。

注意： 如果处理器检测到连接了调试器，就会禁用 WIC。

## 2.8 指令集总结

该处理器执行一个 Thumb 指令集版本。表 2-13 在 112 页 列出了所支持的指令。

注意： 在表 2-13 在 112 页 中：

- 尖括号 <> 包含了操作数的复用形式
- 大括号 {} 包含了可选的操作数
- 操作数列是不全面的
- Op2 是第二操作数，它可以是一个寄存器，也可以是一个常数。
- 大部分指令可以使用一个可选的状态码后缀

有关指令和操作数的更多信息，请参考“ARM® Cortex™-M4 技术参考手册”中的指令描述。

表 2-13. Cortex-M4F 指令摘要

助记符	操作数	简要描述	标志
ADC, ADCS	{Rd,} Rn, Op2	带进位加法	N,Z,C,V
ADD, ADDS	{Rd,} Rn, Op2	加法	N,Z,C,V
ADD, ADDW	{Rd,} Rn, #imm12	加法	-
ADR	Rd, label	载入PC相对地址	-
AND, ANDS	{Rd,} Rn, Op2	逻辑与	N,Z,C
ASR, ASRS	Rd, Rm, <Rs #n>	算术右移	N,Z,C
B	label	转移	-
BFC	Rd, #lsb, #width	位域清零	-
BFI	Rd, Rn, #lsb, #width	位域插入	-
BIC, BICS	{Rd,} Rn, Op2	位清零	N,Z,C
BKPT	#imm	断点	-
BL	label	带连接转移	-
BLX	Rm	带连接的间接转移	-
BX	Rm	间接转移	-
CBNZ	Rn, label	比较非零转移	-
CBZ	Rn, label	比较为零转移	-
CLREX	-	清除互斥	-
CLZ	Rd, Rm	计算前导0的数目	-
CMN	Rn, Op2	负向比较	N,Z,C,V
CMP	Rn, Op2	比较	N,Z,C,V
CPSID	i	改变处理器状态，禁止中断	-
CPSIE	i	改变处理器状态，启用中断	-
DMB	-	数据存储隔离	-
DSB	-	数据同步隔离	-
EOR, EORS	{Rd,} Rn, Op2	近位异或	N,Z,C
ISB	-	指令同步隔离	-
IT	-	If-Then 条件块	-
LDM	Rn{!}, reglist	加载多个寄存器，加载后自增加	-
LDMDB, LDMEA	Rn{!}, reglist	加载多个寄存器，加载前自减	-
LDMFD, LDMIA	Rn{!}, reglist	加载多个寄存器，加载后自增加	-
LDR	Rt, [Rn, #offset]	从寄存器加载字	-
LDRB, LDRBT	Rt, [Rn, #offset]	从寄存器中加载字节	-
LDRD	Rt, Rt2, [Rn, #offset]	从寄存器中加载双字节	-
LDREX	Rt, [Rn, #offset]	加载寄存器，标记互斥	-



表 2-13. Cortex-M4F 指令摘要 (续)

助记符	操作数	简要描述	标志
LDREXB	Rt, [Rn]	从寄存器加载字节, 标记互斥	-
LDREXH	Rt, [Rn]	从寄存器加载半字, 标记互斥	-
LDRH, LDRHT	Rt, [Rn, #offset]	从寄存器加载半字	-
LDRSB, LDRSBT	Rt, [Rn, #offset]	从寄存器加载带符号的字节	-
LDRSH, LDRSHT	Rt, [Rn, #offset]	从寄存器加载带符号的半字	-
LDRT	Rt, [Rn, #offset]	从寄存器加载字	-
LSL, LSLS	Rd, Rm, <Rs>#n	逻辑左移	N,Z,C
LSR, LSRS	Rd, Rm, <Rs>#n	逻辑右移	N,Z,C
MLA	Rd, Rn, Rm, Ra	乘加, 32位结果	-
MLS	Rd, Rn, Rm, Ra	乘减, 32位结果	-
MOV, MOVS	Rd, Op2	加载	N,Z,C
MOV, MOVW	Rd, #imm16	加载16位常数	N,Z,C
MOVT	Rd, #imm16	加载高位	-
MRS	Rd, spec_reg	从特殊寄存器加载到通用寄存器	-
MSR	spec_reg, Rm	从通用寄存器加载到特殊寄存器	N,Z,C,V
MUL, MULS	{Rd,} Rn, Rm	乘法, 32位结果	N,Z
MVN, MVNS	Rd, Op2	取反加载	N,Z,C
NOP	-	无操作	-
ORN, ORNS	{Rd,} Rn, Op2	逻辑或取反	N,Z,C
ORR, ORRS	{Rd,} Rn, Op2	逻辑或	N,Z,C
PKHTB, PKHBT	{Rd,} Rn, Rm, Op2	包半字	-
POP	reglist	从堆栈中弹出到寄存器	-
PUSH	reglist	将寄存器值压入堆栈	-
QADD	{Rd,} Rn, Rm	饱和加法	Q
QADD16	{Rd,} Rn, Rm	饱和加法 (16)	-
QADD8	{Rd,} Rn, Rm	饱和加法 (8)	-
QASX	{Rd,} Rn, Rm	可互调的饱和加法和饱和减法	-
QDADD	{Rd,} Rn, Rm	饱和倍乘加法	Q
QDSUB	{Rd,} Rn, Rm	饱和倍乘减法	Q
QSAX	{Rd,} Rn, Rm	可互调的饱和减法和饱和加法	-
QSUB	{Rd,} Rn, Rm	饱和减法	Q
QSUB16	{Rd,} Rn, Rm	饱和减法 (16)	-
QSUB8	{Rd,} Rn, Rm	饱和减法 (8)	-
RBIT	Rd, Rn	位反转	-
REV	Rd, Rn	在一个字中反转字节顺序	-
REV16	Rd, Rn	在每个半字中反转字节顺序	-
REVSH	Rd, Rn	反转低半字的字节顺序, 带符号扩展	-
ROR, RORS	Rd, Rm, <Rs>#n	循环右移	N,Z,C
RRX, RRXS	Rd, Rm	带进位的循环右移	N,Z,C
RSB, RSBS	{Rd,} Rn, Op2	反向减法	N,Z,C,V
SADD16	{Rd,} Rn, Rm	带符号的加法 (16)	GE
SADD8	{Rd,} Rn, Rm	带符号的加法 (8)	GE

表 2-13. Cortex-M4F 指令摘要 (续)

助记符	操作数	简要描述	标志
SASX	{Rd,} Rn, Rm	可互调的带符号加法和减法	GE
SBC, SBCS	{Rd,} Rn, Op2	带借位的减法	N,Z,C,V
SBFX	Rd, Rn, #lsb, #width	带符号位域扩展	-
SDIV	{Rd,} Rn, Rm	带符号除法	-
SEL	{Rd,} Rn, Rm	选择字节	-
SEV	-	发送事件	-
SHADD16	{Rd,} Rn, Rm	带符号的半字加法 (16)	-
SHADD8	{Rd,} Rn, Rm	带符号的半字加法 (8)	-
SHASX	{Rd,} Rn, Rm	可互调的带符号半字加法和半字减法	-
SHSAX	{Rd,} Rn, Rm	可互调的带符号半字加法和半字减法	-
SHSUB16	{Rd,} Rn, Rm	带符号的半字减法 (16)	-
SHSUB8	{Rd,} Rn, Rm	带符号的半字减法 (8)	-
SMLABB, SMLABT, SMLATB, SMLATT	Rd, Rn, Rm, Ra	带符号的长整型 (半字) 乘加	Q
SMLAD, SMLADX	Rd, Rn, Rm, Ra	带符号的倍乘加	Q
SMLAL	RdLo, RdHi, Rn, Rm	带符号乘加 (32 x 32 + 64), 64 位结果	-
SMLALBB, SMLALBT, SMLALTB, SMLALTT	RdLo, RdHi, Rn, Rm	带符号的长整型 (半字) 乘加	-
SMLALD, SMLALDX	RdLo, RdHi, Rn, Rm	带符号的长整型整数 (半字) 倍乘加	-
SMLAWB, SMLAWT	Rd, Rn, Rm, Ra	带符号的乘加, 字乘半字	Q
SMLSD SMLSDX	Rd, Rn, Rm, Ra	带符号的倍乘减	Q
SMLS LD SMLS LD X	RdLo, RdHi, Rn, Rm	带符号的长整型 (半字) 倍乘减	-
SMMLA	Rd, Rn, Rm, Ra	带符号的最高有效字乘加	-
SMMLS, SMMLR	Rd, Rn, Rm, Ra	带符号的最高有效字乘减	-
SMMUL, SMMULR	{Rd,} Rn, Rm	带符号的最高有效字乘法	-
SMUAD SMUADX	{Rd,} Rn, Rm	带符号的倍乘加	Q
SMULBB, SMULBT, SMULTB, SMULTT	{Rd,} Rn, Rm	带符号的半字乘法	-
SMULL	RdLo, RdHi, Rn, Rm	带符号乘法 (32x32), 64位结果	-
SMULWB, SMULWT	{Rd,} Rn, Rm	带符号乘法 (乘半字)	-

表 2-13. Cortex-M4F 指令摘要 (续)

助记符	操作数	简要描述	标志
SMUSD, SMUSDX	{Rd,} Rn, Rm	带符号的倍乘减	-
SSAT	Rd, #n, Rm {,shift #s}	带符号的饱和运算	Q
SSAT16	Rd, #n, Rm	带符号的饱和运算 (16)	Q
SSAX	{Rd,} Rn, Rm	可互调的饱和减法和饱和加法	GE
SSUB16	{Rd,} Rn, Rm	带符号的减法 (16)	-
SSUB8	{Rd,} Rn, Rm	带符号的减法 (8)	-
STM	Rn{!}, reglist	保存多个寄存器, 保存后自增加	-
STMDB, STMEA	Rn{!}, reglist	保存多个寄存器, 保存前自减	-
STMFD, STMIA	Rn{!}, reglist	保存多个寄存器, 保存后自增加	-
STR	Rt, [Rn {, #offset}]	保存寄存器字	-
STRB, STRBT	Rt, [Rn {, #offset}]	保存寄存器字节	-
STRD	Rt, Rt2, [Rn {, #offset}]	保存寄存器双字	-
STREX	Rt, Rt, [Rn {, #offset}]	在互斥状态保存寄存器	-
STREXB	Rd, Rt, [Rn]	在互斥状态保存寄存器字节	-
STREXH	Rd, Rt, [Rn]	在互斥状态保存寄存器半字	-
STRH, STRHT	Rt, [Rn {, #offset}]	保存寄存器半字	-
STRSB, STRSBT	Rt, [Rn {, #offset}]	保存寄存器带符号字节	-
STRSH, STRSHT	Rt, [Rn {, #offset}]	保存寄存器带符号半字	-
STRT	Rt, [Rn {, #offset}]	保存寄存器字	-
SUB, SUBS	{Rd,} Rn, Op2	减法	N,Z,C,V
SUB, SUBW	{Rd,} Rn, #imm12	减12位常数	N,Z,C,V
SVC	#imm	系统服务调用	-
SXTAB	{Rd,} Rn, Rm, {,ROR #}	将 8 位扩展到 32 位后做加法运算	-
SXTAB16	{Rd,} Rn, Rm,{,ROR #}	将 8 位扩展到 16 位后做加法运算	-
SXTAH	{Rd,} Rn, Rm,{,ROR #}	将 16 位扩展到 32 位后做加法运算	-
SXTB16	{Rd,} Rm {,ROR #n}	带符号扩展字节 (16)	-
SXTB	{Rd,} Rm {,ROR #n}	带符号扩展一个字节	-
SXTH	{Rd,} Rm {,ROR #n}	带符号扩展一个半字	-
TBB	[Rn, Rm]	查表转移字节	-
TBH	[Rn, Rm, LSL #1]	查表转移半字	-
TEQ	Rn, Op2	测试是否相等	N,Z,C
TST	Rn, Op2	测试	N,Z,C
UADD16	{Rd,} Rn, Rm	无符号的加法 (16)	GE
UADD8	{Rd,} Rn, Rm	无符号的加法 (8)	GE
UASX	{Rd,} Rn, Rm	可互调的无符号加法和减法	GE
UHADD16	{Rd,} Rn, Rm	无符号的半字加法 (16)	-
UHADD8	{Rd,} Rn, Rm	无符号的半字加法 (8)	-
UHASX	{Rd,} Rn, Rm	可互调的无符号半字加法和半字减法	-
UHSAX	{Rd,} Rn, Rm	可互调的无符号半字减法和半字加法	-
UHSUB16	{Rd,} Rn, Rm	无符号的半字减法 (16)	-
UHSUB8	{Rd,} Rn, Rm	无符号的半字减法 (8)	-

表 2-13. Cortex-M4F 指令摘要 (续)

助记符	操作数	简要描述	标志
UBFX	Rd, Rn, #sb, #width	无符号位域扩展	-
UDIV	{Rd,} Rn, Rm	无符号除法	-
UMAAL	RdLo, RdHi, Rn, Rm	无符号的长整型乘加 (32x32+64), 64 位结果	-
UMLAL	RdLo, RdHi, Rn, Rm	无符号乘加 (32 x 32 + 32 + 32), 64 位结果	-
UMULL	RdLo, RdHi, Rn, Rm	无符号乘法 (32x32), 64位结果	-
UQADD16	{Rd,} Rn, Rm	无符号的饱和加法 (16)	-
UQADD8	{Rd,} Rn, Rm	无符号的饱和加法 (8)	-
UQASX	{Rd,} Rn, Rm	可互调的无符号饱和加法和饱和减法	-
UQSAX	{Rd,} Rn, Rm	可互调的无符号饱和减法和饱和加法	-
UQSUB16	{Rd,} Rn, Rm	无符号的饱和减法 (16)	-
UQSUB8	{Rd,} Rn, Rm	无符号的饱和减法 (8)	-
USAD8	{Rd,} Rn, Rm	无符号的绝对差和	-
USADA8	{Rd,} Rn, Rm, Ra	无符号的绝对差及累加和	-
USAT	Rd, #n, Rm {,shift #s}	无符号饱和操作	Q
USAT16	Rd, #n, Rm	无符号的饱和操作 (16)	Q
USAX	{Rd,} Rn, Rm	可互调的无符号减法和加法	GE
USUB16	{Rd,} Rn, Rm	无符号的减法 (16)	GE
USUB8	{Rd,} Rn, Rm	无符号的减法 (8)	GE
UXTAB	{Rd,} Rn, Rm, {,ROR #}	轮换, 将 8 位扩展到 32 位后做加法运算	-
UXTAB16	{Rd,} Rn, Rm, {,ROR #}	轮换, 将 8 位扩展到 16 位后做加法运算	-
UXTAH	{Rd,} Rn, Rm, {,ROR #}	轮换, 无符号扩展并加半字	-
UXTB	{Rd,} Rm, {,ROR #n}	使用零扩展一个字节	-
UXTB16	{Rd,} Rm, {,ROR #n}	无符号扩展字节 (16)	-
UXTH	{Rd,} Rm, {,ROR #n}	使用零扩展一个半字	-
VABS.F32	Sd, Sm	浮点绝对	-
VADD.F32	{Sd,} Sn, Sm	浮点加法	-
VCMP.F32	Sd, <Sm   #0.0>	比较两个浮点型寄存器, 或者是浮点型寄存器和零	FPSCR
VCMPE.F32	Sd, <Sm   #0.0>	比较两个浮点型寄存器, 或者是带有无效操作校验的浮点型寄存器和零	FPSCR
VCVT.S32.F32	Sd, Sm	在浮点型数据和整数之间的转换	-
VCVT.S16.F32	Sd, Sd, #fbits	在浮点型数据和定点数据之间的转换	-
VCVTR.S32.F32	Sd, Sm	在浮点型数据和舍位整数之间的转换	-
VCVT<B H>.F32.F16	Sd, Sm	将半精度的值转换为单精度值	-
VCVTT<B T>.F32.F16	Sd, Sm	将单精度的值转换为半精度值	-
VDIV.F32	{Sd,} Sn, Sm	浮点除法	-
VFMA.F32	{Sd,} Sn, Sm	浮点熔合乘加	-
VFNMA.F32	{Sd,} Sn, Sm	浮点熔合乘加取反	-
VFMS.F32	{Sd,} Sn, Sm	浮点熔合乘减	-
VFNMS.F32	{Sd,} Sn, Sm	浮点熔合乘减取反	-
VLDM.F<32 64>	Rn{!}, list	加载多个扩展寄存器	-

表 2-13. Cortex-M4F 指令摘要 (续)

助记符	操作数	简要描述	标志
VLDR.F<32 64>	<Dd Sd>, [Rn]	从存储器加载扩展寄存器	-
VLMA.F32	{Sd,} Sn, Sm	浮点乘加	-
VLMS.F32	{Sd,} Sn, Sm	浮点乘减	-
VMOV.F32	Sd, #imm	浮点立即移动	-
VMOV	Sd, Sm	浮点移动寄存器	-
VMOV	Sn, Rt	将 ARM 内核寄存器复制到单精度	-
VMOV	Sm, Sm1, Rt, Rt2	将 2 个 ARM 内核寄存器复制到 2 个单精度	-
VMOV	Dd[x], Rt	将 ARM 内核寄存器复制到标量	-
VMOV	Rt, Dn[x]	将标量复制到 ARM 内核	-
VMRS	Rt, FPSCR	将 FPSCR 移动到 ARM 内核寄存器或 APSR	N,Z,C,V
VMSR	FPSCR, Rt	从 ATM 内核寄存器移动到 FPSCR	FPSCR
VMUL.F32	{Sd,} Sn, Sm	浮点乘法	-
VNEG.F32	Sd, Sm	浮点取反	-
VNMLA.F32	{Sd,} Sn, Sm	浮点乘加	-
VNMLS.F32	{Sd,} Sn, Sm	浮点乘减	-
VNMUL	{Sd,} Sn, Sm	浮点乘法	-
VPOP	list	从扩展寄存器弹出	-
V PUSH	list	压入扩展寄存器	-
VSQRT.F32	Sd, Sm	计算浮点数据平方根	-
VSTM	Rn{!}, list	浮点寄存器多存储	-
VSTR.F3<32 64>	Sd, [Rn]	将扩展寄存器存到存储器	-
VSUB.F<32 64>	{Sd,} Sn, Sm	浮点除法	-
WFE	-	等待事件	-
WFI	-	等待中断	-

## 3 Cortex-M4 外设

本章描述实施 Stellaris® (Cortex-M4 处理器外设) 的信息, 包括:

- SysTick (见第 118 页)
 

提供一个简单易用、配置灵活的24位单调递减计数器。该计数器具有写入即清零、过零自动重载等特性。
- 嵌套式向量化中断控制器 (NVIC) (见第 119 页)
  - 实现异常及中断的快速响应处理
  - 控制电源管理
  - 实现系统控制寄存器
- 系统控制模块 (SCB) (见第 120 页)
 

可提供系统构成信息, 并进行系统控制, 包括系统异常的配置、控制以及上报。
- 存储器保护单元 (MPU) (见第 121 页)
 

支持标准的ARMv7受保护存储器系统架构 (PMSA) 模型。MPU为系统的保护区、重叠保护区、访问权限以及导出存储器属性提供全面支持。
- 浮点单元 (FPU) (见第 125 页)
 

完全支持单精度的加、减、乘、除、乘加以及平方根操作。其也提供定点和浮点数据格式之间的转换以及浮点常数指令。

表 3-1 在 118页 显示专用外设总线 (PPB) 的地址映射。某些外设寄存器空间还会进一步划分为两个区, 在表中分别列为两组地址。

表 3-1. 内核外设寄存器区域

地址	内核外设	描述 (参见第 ... 页)
0xE000.E010-0xE000.E01F	系统定时器	118
0xE000.E100-0xE000.E4EF 0xE000.EF00-0xE000.EF03	嵌套向量中断控制器	119
0xE000.E008-0xE000.E00F 0xE000.ED00-0xE000.ED3F	系统控制模块	120
0xE000.ED90-0xE000.EDB8	存储器保护单元	121
0xE000.EF30-0xE000.EF44	浮点单元	125

### 3.1 功能描述

本章提供关于 Stellaris 实施 (Cortex-M4 处理器外设) 的信息: SysTick、NVIC、SCB 和 MPU。

#### 3.1.1 系统定时器 (SysTick)

Cortex-M4 包括一个集成的系统定时器 SysTick。该定时器提供简单易用、控制灵活的 24 位递减计数器, 具有写入即清零、过零自动重载等功能。该计数器的用途广泛, 举例来说, 可以:

- 用作 RTOS 节拍定时器, 可按照编程的频率 (例如 100 Hz) 定时触发, 并可调用系统定时器服务子程序。
- 用作高速报警定时器, 采用系统时钟作为时钟源。

- 用作频率可变的报警或信号定时器 - 其持续时间取决于所采用的参考时钟源以及计数器的动态范围。
- 用作简单计数器，测量任务的完成时刻、总体耗时等等。
- 用于实现基于失配/匹配周期的内部时钟源控制。STCTRL 控制及状态寄存器的 COUNT 标志位可以用于判定某个动作是否在指定的时间内完成，以此作为动态时钟管理控制循环的一部分。

定时器包含以下 3 个寄存器：

- SysTick 控制和状态 (STCTRL)：控制和状态计数器用来配置其时钟、启用计数器、启用 SysTick 中断以及确定计数器状态。
- SysTick 重载值 (STRELOAD)：计数器的重载值，用来提供计数器的重载值。
- SysTick 当前值 (STCURRENT)：计数器的当前值。

启用定时器后，计数器将在每个时钟递减一次，从重载值逐个递减到 0，之后在下一个时钟沿重新装入 STRELOAD 寄存器中的值，然后在后续时钟上递减。清空 STRELOAD 寄存器将在下次重载时禁用计数器。计数器达到 0 时，已设置 COUNT 状态位。读取后，COUNT 位将清零。

写入 STCURRENT 寄存器将清除此寄存器以及 COUNT 状态位。这个写操作并不会触发 SysTick 异常逻辑。读取该寄存器时，当前值是访问该寄存器时寄存器的值。

SysTick 计数器根据系统时钟或精确内部振荡器 (PIOSC) 4 分频运行。如果在低功耗模式下停止提供该时钟信号，则 SysTick 计数器将停止。在 SysTick 控制和状态寄存器 (STCTRL) 中设置 CLK\_SRC 位并确保深度睡眠时钟配置 (DSLPCCLKCFG) 寄存器中的 PIOSCPD 位已清零，从而使 SysTick 在深度休眠模式中保持运行状态。确保软件使用字对齐访问来访问 SysTick 寄存器。

在复位时，SysTick 计数器的重载值和当前值并未定义；SysTick 计数器的正确初始化序列为：

1. 将 STRELOAD 寄存器中的值编程。
2. 向 STCURRENT 寄存器中写入任意值来将其清零。
3. 配置 STCTRL 寄存器以执行所需操作。

注意：在调试过程中处理器暂停时，此计数器不会递减。

### 3.1.2 嵌套向量中断控制器 (NVIC)

本节描述嵌套向量中断控制器 (NVIC) 及其使用的寄存器。NVIC 支持：

- 96 中断。
- 每个中断的优先级均可编程，取值范围 0~7。优先级数字越大则其优先级越低，也就是说 0 代表最高优先级；
- 可实现异常及中断的快速响应处理；
- 中断信号可以是电平检测或脉冲检测；
- 动态重设中断优先级；
- 优先级可分组，划分为分组优先级域以及子优先级域；
- 支持咬尾中断；

- 提供一个外部的不可屏蔽中断 ( NMI )。

处理器在异常入口处能够自动将状态入栈，在退出异常时能够自动将状态出栈。这个过程是处理器自行完成的、无需多余的指令开销，因此可快速响应异常并进行处理。

### 3.1.2.1 电平式中断及脉冲式中断

处理器支持电平式中断及脉冲式中断。脉冲式中断通常又称为边沿触发中断。

对于电平式中断而言，只要外设产生中断信号就会始终保持处于触发状态，直到外设中断信号复原后才不再触发。一般来说这需要ISR ( 中断服务子程序 ) 对外设进行操作，使得外设不再产生中断请求信号。脉冲中断是在处理器时钟的上升沿同步采样的中断信号。为了确保NVIC能够检测到中断，外设所产生的中断信号必须保持至少一个时钟周期，在此期间NVIC可检测到脉冲并锁存中断。

处理器进入ISR后，将自动清除该中断的挂起状态 ( 参见“中断的硬件控制及软件控制”在120页以了解更多信息 )。对于电平式中断，如果处理器从ISR返回后，中断信号仍未复原，则中断将再次挂起，于是处理器必须再次运行其ISR。因此，外设可以像这样保持中断信号持续享用服务，直到其不再需要服务为止。

### 3.1.2.2 中断的硬件控制及软件控制

Cortex-M4 锁存所有中断。当满足以下条件之一，外设中断即变为挂起状态：

- NVIC检测到某个中断信号为高电平，并且该中断为未激活状态；
- NVIC检测到某个中断信号的上升沿；
- 软件写入相应的中断设置暂挂寄存器位，或者写入软件触发中断(SWTRIG)寄存器使软件产生的中断变为暂挂状态。请参阅140页上PEND0寄存器中的INT位或150页上的SWTRIG。

中断挂起后将保持挂起状态，直到满足以下条件之一：

- 处理器进入该中断的ISR，将中断状态由挂起改为已激活。然后：
  - 对于电平式中断，当处理器从ISR返回后，NVIC将再次采样中断信号。假如仍然检测到中断信号，那么中断状态将再次变为挂起，使得处理器立即重新进入该ISR。否则，中断状态将变为未激活。
  - 对于脉冲式中断，处理器时刻监视着中断信号的状态。只要检测到中断脉冲信号就会将中断状态改为挂起并激活。在此情况下，当处理器从ISR返回后，中断状态将再次变为挂起，使得处理器立即重新进入该ISR。

如果当处理器在ISR内期间并未产生中断脉冲信号，那么当处理器从ISR返回后，中断状态将变为未激活。
- 软件对中断清除挂起寄存器中的相应标志位执行写操作
  - 对于电平式中断，若仍旧产生中断信号，那么中断状态将保持不变。否则，中断状态将变为未激活。
  - 对于脉冲式中断，假如中断状态是挂起并激活，则将变为未激活。

### 3.1.3 系统控制模块 ( SCB )

系统控制模块提供系统构成信息，并能实现系统控制功能，包括系统异常的配置、控制以及上报。



### 3.1.4 存储器保护单元 ( MPU )

本节介绍存储器保护单元 ( MPU )。MPU将存储器映射空间划分为若干个存储器区，并分别规定每一个存储器区的起始地址、大小、访问权限以及存储属性。MPU支持为每一个存储器区分别定义其属性设置，支持重叠区的设置，此外还能将存储属性导出到系统。

存储属性影响对该存储器区进行访问时的表现。Cortex-M4 MPU 定义八个单独存储器区 ( 0-7 ) 以及背景区。

存储器区出现重叠时，存储器访问将由编号最大存储器区的属性决定。例如，第7存储器区始终优先于其它存储器区，因此一旦与其它区发生重叠时，都将以第7区的存储属性为准。

背景区的存储器访问属性与默认的存储器映射相同，但只允许特权级软件依此进行访问。

Cortex-M4 MPU 存储器映射是统一的，表示指令访问和数据访问都是共用相同的存储器区设置。

假如程序试图访问某个地址，而该地址被MPU设置为禁止访问，那么处理器将产生一个存储管理故障，并由此触发故障异常。这个异常可能会导致操作系统环境下某个进程终止运行。在操作系统环境下，操作系统内核能够按照进程的实际需求来动态更新MPU区的设置。一般来说，嵌入式操作系统都需要通过MPU实现存储器保护。

MPU 区的配置是基于存储器类型的 ( 参见“存储器区、类型和属性”在 94页以了解更多信息 )。

表 3-2 在 121页 显示 MPU 区的可能属性。请参阅“一个 Stellaris 微处理器”一节在 125页 以了解关于将微控制器实施编程的准则。

表 3-2. 存储器属性摘要

存储器类型	描述
严格顺序	对严格顺序存储器的所有访问必须按照程序顺序进行。
设备	存储器映射外部设备
普通	普通存储器

为避免出现无法预料的执行结果，如果某些中断的处理函数可能访问某存储器区，那么在更新该存储器区的属性之前，应当先关闭这些中断。

在访问MPU寄存器时，应确保软件按照正确的宽度进行对齐访问：

- 除了 MPU 区属性及大小 (MPUATTR) 寄存器，必须以字对齐访问来访问所有 MPU 寄存器。
- MPUATTR 寄存器可按字节对齐、半字对齐或字对齐进行访问。

处理器不支持对MPU寄存器进行未对齐访问。

在配置MPU时，由于MPU可能曾经被编程过，因此所有当前不用的存储器区都应当禁用，防止存储器区的旧设置对当前更新后的MPU设置产生不良影响。

#### 3.1.4.1 更新单个MPU区

为了更新 MPU 区域的属性，必须更新 MPU 区编号 (MPUNUMBER)、MPU 区基地址 (MPUBASE) 和 MPUATTR 寄存器。用户可以分别编程各个寄存器，也可以利用多字写操作来同时编程所有寄存器。您可以使用 MPUBASEx 和 MPUATTRx 别名，通过使用 STM 指令同时对最多四个区进行编程。

采用单字写操作，更新单个MPU区

下面的例程可以完成单个MPU区的配置：

```

; R1 = region number
; R2 = size/enable
; R3 = attributes
; R4 = address
LDR R0,=MPUNUMBER      ; 0xE000ED98, MPU region number register
STR R1, [R0, #0x0]     ; Region Number
STR R4, [R0, #0x4]     ; Region Base Address
STRH R2, [R0, #0x8]   ; Region Size and Enable
STRH R3, [R0, #0xA]   ; Region Attribute

```

假如之前启用了某个存储器区，又需要更改其设置，那么在向MPU写入新的设置之前应当先将此存储器区禁用。例如：

```

; R1 = region number
; R2 = size/enable
; R3 = attributes
; R4 = address
LDR R0,=MPUNUMBER      ; 0xE000ED98, MPU region number register
STR R1, [R0, #0x0]     ; Region Number
BIC R2, R2, #1         ; Disable
STRH R2, [R0, #0x8]   ; Region Size and Enable
STR R4, [R0, #0x4]     ; Region Base Address
STRH R3, [R0, #0xA]   ; Region Attribute
ORR R2, #1             ; Enable
STRH R2, [R0, #0x8]   ; Region Size and Enable

```

在以下情况下，软件中必须采用存储器隔离指令：

- 假如在修改MPU配置时，可能存在某些仍在进行中的存储器传输过程（例如缓冲式写操作），则该操作将可能受到MPU配置改变的影响。
- 配置MPU后，如果包含存储器传输，则必须使用新的MPU设置。

假如进入异常处理程序后立即开始配置MPU，或配置完MPU后立即从异常中返回，那么可以不采用存储器隔离指令，因为异常入口和异常出口已经能够实现相当于存储器隔离指令的保护机制。

在配置MPU期间，并不需要软件再增加存储器隔离指令，这是因为访问MPU时必须通过私有外设总线（PPB），而PPB是严格顺序存储器区。

举例来说，如果要让所有的存储器访问行为在编程序列后立即生效，则应使用DSB指令和ISB指令。更改MPU配置后应当立即发出一条DSB指令，例如上下文切换的结尾。如果对MPU存储区编程的代码是通过分支语句或调用语句进入的，那么应当发出一条ISB指令。如果编程序列使用异常返回或通过进入异常来输入，那么无需发出ISB指令。

采用多字写操作，更新单个MPU区

取决于写入信息如何分割，MPU也可以直接采用多字写操作进行编程。考虑以下重新编程：

```

; R1 = region number
; R2 = address
; R3 = size, attributes in one
LDR R0, =MPUNUMBER    ; 0xE000ED98, MPU region number register
STR R1, [R0, #0x0]    ; Region Number
STR R2, [R0, #0x4]    ; Region Base Address
STR R3, [R0, #0x8]    ; Region Attribute, Size and Enable

```

STM 指令可以用于优化：

```
; R1 = region number
; R2 = address
; R3 = size, attributes in one
LDR R0, =MPUNUMBER ; 0xE000ED98, MPU region number register
STM R0, {R1-R3} ; Region number, address, attribute, size and enable
```

此操作可以按预整合信息的两个字来完成，表示 MPU 区基地址 (MPUBASE) 寄存器 (请参阅 183 页) 包含所需区域号且具有 VALID 位集。这种方式比较适合静态配置数据的应用场合，例如对于 Bootloader，就能够进一步精简其大小。

```
; R1 = address and region number in one
; R2 = size and attributes in one
LDR R0, =MPUBASE ; 0xE000ED9C, MPU Region Base register
STR R1, [R0, #0x0] ; Region base address and region number combined
; with VALID (bit 4) set
STR R2, [R0, #0x4] ; Region Attribute, Size and Enable
```

### 存储子区

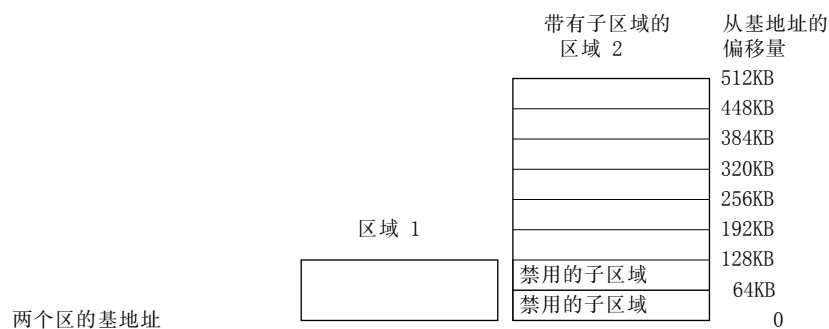
任何不小于256字节的存储器区还能进一步均分为8个存储子区。在 MPU 区属性及大小 (MPUATTR) 寄存器的 SRD 域中设置相应位 (请参阅 185页) 以禁用存储子区。SRD 域的最低位控制首个子区，最高位控制最后子区。禁用某个子区表示另一区域重叠禁用的范围匹配。如果禁用某个子区后，也没有其它已启用的存储器区重叠在该区域上，那么MPU会产生一个故障。

如果存储器区的大小为32、64或128字节，是无法再划分为子区的。对于这些大小的区域，SRD 域必须配置为 0x00，否则 MPU 行为将无法预测。

### SRD的用法示例

在下面的例子中，两个存储器区具有相同的基地址，因此它们相互重叠。第1个区的大小为128kB，第2个区的大小为512kB。为了确保来自第一个区域的属性应用于第一个 128 KB 区域，请将第二个区域的 SRD 域配置为 0x03 以禁用前两个子区域，如图 3-1 在 123页 所示。

图 3-1. SRD 使用示例



#### 3.1.4.2 MPU访问权限属性

MPUATTR 寄存器的访问权限位 TEX、S、C、B、AP 和 XN 控制对相应存储器区域的访问。假如试图访问某个区域，而该区域并未开放相关权限，那么MPU将会产生一个访问权限故障。

表 3-3 在 124页 显示 TEX、C、B、和 S 访问权限位的编码。出于完整方面的考虑，显示所有编码，不过目前的 Cortex-M4 实施不支持高速缓存或可共享性的概念。请参阅“一个 Stellaris 微处理器”一节 在 125页 以了解关于将 MPU 编程以实施 Stellaris。

表 3-3. TEX、S、C 和 B 位域编码

TEX	S	C	B	存储器类型	可共享性	其他属性
000b	x <sup>a</sup>	0	0	严格顺序	可共享	-
000	x <sup>a</sup>	0	1	设备	可共享	-
000	0	1	0	普通	不可共享	外部及内部均为完全写入式。无写分配。
000	1	1	0	普通	可共享	
000	0	1	1	普通	不可共享	
000	1	1	1	普通	可共享	外部及内部均不可高速缓冲。
001	0	0	0	普通	不可共享	
001	1	0	0	普通	可共享	外部及内部均为回写式。写及读均分配
001	x <sup>a</sup>	0	1	保留的编码	-	
001	x <sup>a</sup>	1	0	保留的编码	-	外部及内部均为回写式。写及读均分配
001	0	1	1	普通	不可共享	
001	1	1	1	普通	可共享	不可共享的设备。
010	x <sup>a</sup>	0	0	设备	不可共享	
010	x <sup>a</sup>	0	1	保留的编码	-	
010	x <sup>a</sup>	1	x <sup>a</sup>	保留的编码	-	高速缓存存储器 (BB 代表外部策略, AA 代表内部策略)。 关于 AA 和 BB 位的编码, 参见表 3-4。
1BB	0	A	A	普通	不可共享	
1BB	1	A	A	普通	可共享	

a. MPU 将忽略此位的值。

表 3-4 在 124 页 显示存储器属性编码 (TEX 值在范围 0x4-0x7 中) 的高速缓存策略。

表 3-4. 存储器属性编码对应的高速缓存策略

编码 (AA 或 BB)	相应的高速缓存策略
00	不可高速缓存
01	回写式, 写及读均分配
10	完全写入式, 无写分配
11	回写式, 无写分配

表 3-5 在 124 页 显示 MPUATTR 寄存器中的 AP 编码, 这些编码定义特权软件和非特权软件的访问权限。

表 3-5. AP 位域编码

AP 位域	特权权限	非特权权限	描述
000	无访问权	无访问权	任何访问操作都会产生权限故障。
001	R/W	无访问权	仅特权级软件才拥有访问权限。
010	R/W	RO	非特权级软件的写操作会导致产生一个访问权限故障。
011	R/W	R/W	完全访问权
100	无法预测	无法预测	保留。
101	RO	无访问权	仅特权级软件才拥有读取权限。
110	RO	RO	特权级软件或非特权级软件均拥有只读权限。
111	RO	RO	特权级软件或非特权级软件均拥有只读权限。

### 一个 Stellaris 微处理器

Stellaris 微控制器只有一个处理器，并且不带高速缓存。因此，应按表 3-6 在 125 页 中所示将 MPU 编程。

表 3-6. Stellaris 微控制器的® 存储器区属性

存储器区	TEX	S	C	B	存储器类型和属性
Flash 存储器	000b	0	1	0	普通存储器，不可共享，完全写入式
内部 SRAM	000b	1	1	0	普通存储器，可共享，完全写入式
外部 SRAM	000b	1	1	1	普通存储器，可共享，回写式，写分配
外设	000b	1	0	1	设备存储器，可共享

在当前的 Stellaris 微控制器的实施中，可共享性和高速缓冲策略属性对系统行为并无影响。不过，如果正确配置这些 MPU 区属性，有益于充分保障代码的可移植性。上表给出的都是常见配置的参考值。

#### 3.1.4.3 MPU 不匹配

若某个访问操作违反了 MPU 权限，处理器将会产生存储器管理故障（请参阅“异常和中断”在 91 页以了解更多信息）。MFAULTSTAT 寄存器指示故障原因。请参阅 170 页以了解更多信息。

#### 3.1.5 浮点单元 (FPU)

本节描述浮点单元 (FPU) 及其使用的寄存器。FPU 提供：

- 适用于单精度 (C 浮点) 数据处理操作的 32 位指令
- 适用于更高精度 (融合 MAC) 的组合乘加指令
- 适用于换算、加法、减法、可选累加的乘法、除法和平方根计算的硬件支持
- 适用于反常和所有 IEEE 舍入模式的硬件支持
- 32 个专用的 32 位单精度寄存器，也可作为 16 位双字寄存器进行寻址
- 去耦三级流水线

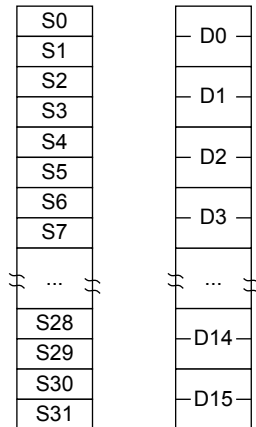
Cortex-M4F FPU 完全支持单精度的加法、减法、乘法、除法、乘加和平方根计算。它还可用于转换定点和浮点数据格式，并提供浮点常数指令。FPU 提供浮点计算功能，该功能符合 ANSI/IEEE 754-2008 标准（适用于二进制浮点运算的 IEEE 标准，也称为 IEEE 754 标准）。FPU 的单精度扩展寄存器也可以作为 16 位双字寄存器，并支持加载、存储和移动操作。

##### 3.1.5.1 寄存器块的 FPU 视图

FPU 提供包含 32 个单精度寄存器的扩展寄存器文件。这些寄存器可视为：

- 十六个 64 位双字寄存器，D0-D15
- 三十二个 32 位单字寄存器，S0-S31
- 以上寄存器的组合

图 3-2. FPU 寄存器块



寄存器之间的映射如下：

- S<2n> 映射到 D<n> 低位的那一半
- S<2n+1> 映射到 D<n> 高位的那一半

例如，您可以通过访问 S12 来访问 D6 中的低位值，通过访问 S13 来访问该元件的高位值。

### 3.1.5.2 工作模式

FPU 提供三种工作模式以适应多种应用程序的需要。

**完全合规性模式.** 在完全合规性模式下，FPU 按照硬件中的 IEEE 754 标准来处理所有工作。

**清零模式.** 将浮点状态和控制 (FPSC) 寄存器的 FZ 位置位可启用清零模式。在该模式下，FPU 将算术 CDP 操作的所有非规格化输入操作数作为 0 来处理。由 0 操作数造成的异常会发出相应的信号。VABS、VNEG 和 VMOV 不被视为算术 CDP 操作，因此不受清零模式的影响。根据 IEEE 754 标准，如果某个结果极小，其目标精度的大小小于舍入前的最小正常值，则该结果将被 0 取代。FPSC 寄存器中的 IDC 位指示清空输入的时间。FPSC 寄存器中的 UFC 位指示清空结果的时间。

**默认 NaN 模式.** 将 FPSC 寄存器中的 DN 位置位可启用默认 NaN 模式。在该模式下，包含输入 NaN 或产生 NaN 结果的任何算术数据处理操作的结果都会返回默认 NaN。小数位的传播仅由 VABS、VNEG 和 VMOV 操作维持。所有其他 CDP 操作会忽略输入 NaN 的小数位中的任何信息。

### 3.1.5.3 符合 IEEE 754 标准

当禁用默认 NaN (DN) 和清零 (FZ) 模式时，FPv4 功能在硬件上符合 IEEE 754 标准。实现此合规性不需要支持代码。

### 3.1.5.4 完全实现 IEEE 754 标准

Cortex-M4F 浮点指令集并不完全支持 IEEE 754-2008 标准中定义的所有操作。不支持的操作包括但不限于：

- 余数
- 将浮点数舍入为整数值的浮点数
- 二进制转十进制
- 十进制转二进制

### ■ 单精度和双精度值的直接比较

根据 IEEE 标准，Cortex-M4 FPU 支持融合 MAC 操作。要完全实现 IEEE 754-2008 标准，浮点功能必须随库函数增加。

### 3.1.5.5 IEEE 754 标准实现选项

#### NaN 处理

所有具有最大指数域值和非零小数域的单精度值都是有效的 NaN。最高小数位为 0 指示信号非数 (SNaN)。最高小数位为 1 指示静默非数 (QNaN)。如果两个 NaN 的任何一位都不同，则将其视为不同的 NaN 值。下表显示了默认的 NaN 值。

符号	小数	小数
0	0xFF	位 [22] = 1, 位 [21:0] 全为 0

对 ARM 浮点功能和库的输入 NaN 的处理方式定义如下：

- 在完全合规性模式下，NaN 按《ARM 架构参考手册》中的描述处理。硬件直接将 NaN 用于算术 CDP 指令。对于数据传输操作，NaN 被传输而不引起无效操作异常。对于非算术 CDP 指令、VABS、VNEG 和 VMOV，如果指令中有说明，则 NaN 被复制且符号改变，而不引起无效操作异常。
- 在默认 NaN 模式下，涉及 NaN 操作数的算术 CDP 指令将返回默认 NaN，而不考虑 NaN 操作数的小数。算术 CDP 操作中的 SNaN 将 IOC 标志 FPSCR[0] 置位。由数据传输和非算术 CDP 指令进行的 NaN 处理与在完全合规性模式下一样。

表 3-7. QNaN 和 SNaN 处理

指令类型	默认 NaN 模式	带有 QNaN 操作数	带有 SNaN 操作数
算术 CDP	关闭	如果有多个 QNaN 操作数，则将按《ARM 架构参考手册》中的规则返回一个 QNaN 或 QNaN 操作数。	IOC <sup>a</sup> 置位。SNaN 最高有效位被清除，得出的 NaN 取决于《ARM 架构参考手册》中的规则。
	开启	默认 NaN 返回。	IOCa 置位。默认 NaN 返回。
非算术 CDP	关闭/开启	NaN 传输到目的地，符号会发生相应改变。	
FCMP(Z)	-	无序比较。	IOC 置位。无序比较。
FCMPE(Z)	-	IOC 置位。无序比较。	IOC 置位。无序比较。
加载/存储	关闭/开启	传输所有 NaN。	

a. IOC 是无效操作异常标志，FPSCR[0]。

#### 比较

比较结果会修改 FPSCR 中的标志。您可以使用 MVRS APSR\_nzcv 指令（即之前的 FMSTAT）将当前标志从 FPSCR 传输到 APSR。有关将 IEEE 754-2008 标准谓词映射到 ARM 条件的详细信息，请参考《ARM 架构参考手册》。应仔细选择所使用的标志，使后续的 ARM 指令在指定执行条件下能测试 IEEE 标准中定义的谓词。

#### 下溢

Cortex-M4F FPU 使用 IEEE 754-2008 标准中描述的舍入前的极小值和丢失精度的不准确结果来产生下溢异常。

在清零模式下，根据 IEEE 标准，舍入前极小的结果将清空为 0，且 UFC 标志 FPSCR[3] 置位。有关清零模式的信息，请参考《ARM 架构参考手册》。

当 FPU 未处于清零模式时，将对非规格化操作数进行操作。如果操作没有产生极小的结果，它将返回计算结果，且 UFC 标志 FPSCR[3] 不置位。如果操作不精确，IXC 标志 FPSCR[4] 置位。如果操作产生了极小的结果，该结果为非规格化值或 0，如果结果同样不精确，UFC 标志 FPSCR[3] 置位。

### 3.1.5.6 异常

根据 FpV4 架构，FPU 将如每条指令所要求地将 FPSCR 寄存器中的累积异常状态标志置位。FPU 不支持用户模式处理。FPSCR 中的异常启用位读数为 0 并且忽略写操作。处理器也有六个输出管脚：FPIXC、FPUFC、FPOFC、FPDZC、FPIDC 和 FPIOC，每个管脚反映一个累积异常标志的状态。有关这些输出的描述，请参考“ARM Cortex-M4 集成和实现手册”（ARM DII 0239，可通过 ARM 获取）。

该处理器可通过使用怠惰堆栈来降低异常延迟。请参考第 4-5 页上的辅助控制寄存器，ACTLR。这意味着处理器在堆栈上保留了 FP 状态的空间，但并未在堆栈上保存该状态的信息。有关更多信息，请参考《ARMv7-M 架构参考手册》（可通过 ARM 获取）。

### 3.1.5.7 启用 FPU

FPU 通过复位被禁用。您必须先启用 FPU，才能使用浮点指令。处理器必须处于特权模式，才能读取/写入协处理器访问控制 (CPAC) 寄存器。以下示例代码序列可同时在特权模式和用户模式中启用 FPU。

```
; CPACR is located at address 0xE000ED88
LDR.W R0, =0xE000ED88
; Read CPACR
LDR R1, [R0]
; Set bits 20-23 to enable CP10 and CP11 coprocessors
ORR R1, R1, #(0xF << 20)
; Write back the modified value to the CPACR
STR R1, [R0]; wait for store to complete
DSB
;reset pipeline now the FPU is enabled
ISB
```

## 3.2 寄存器映射

表 3-8 在 128 页列出了 Cortex-M4 外设 SysTick、NVIC、MPU、FPU 和 SCB 寄存器。列出的偏移量是相对于 0xE000.E000 内核外设基址而言的寄存器地址的 16 进制增量。

注意：未使用的寄存器空间都是为将来或内部使用保留的。软件不应该调整任何保留的存储器地址。

表 3-8. 外设 寄存器映射

偏移量	名称	类型	复位	描述	见页面
系统定时器 ( SysTick ) 寄存器					
0x010	STCTRL	R/W	0x0000.0004	SysTick 控制及状态寄存器	132
0x014	STRELOAD	R/W	-	SysTick 重载值寄存器	134
0x018	STCURRENT	R/WC	-	SysTick 当前值寄存器	135
嵌套向量中断控制器 ( NVIC ) 寄存器					
0x100	EN0	R/W	0x0000.0000	中断 0-31 集启用寄存器	136



表 3-8. 外设 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0x104	EN1	R/W	0x0000.0000	中断 32-63 集启用寄存器	136
0x108	EN2	R/W	0x0000.0000	中断 64-95 集启用寄存器	136
0x10C	EN3	R/W	0x0000.0000	中断 96-127 集启用寄存器	136
0x110	EN4	R/W	0x0000.0000	中断 128-138 设置启用寄存器	137
0x180	DIS0	R/W	0x0000.0000	中断 0-31 清除启用寄存器	138
0x184	DIS1	R/W	0x0000.0000	中断 32-63 清除启用寄存器	138
0x188	DIS2	R/W	0x0000.0000	中断 64-95 清除启用寄存器	138
0x18C	DIS3	R/W	0x0000.0000	中断 96-127 清除启用寄存器	138
0x190	DIS4	R/W	0x0000.0000	中断 128-138 清除启用寄存器	139
0x200	PEND0	R/W	0x0000.0000	中断 0-31 置位挂起寄存器	140
0x204	PEND1	R/W	0x0000.0000	中断 32-63 置位挂起寄存器	140
0x208	PEND2	R/W	0x0000.0000	中断 64-95 置位挂起寄存器	140
0x20C	PEND3	R/W	0x0000.0000	中断 96-127 置位挂起寄存器	140
0x210	PEND4	R/W	0x0000.0000	中断 128-138 置位挂起寄存器	141
0x280	UNPEND0	R/W	0x0000.0000	中断 0-31 清除挂起寄存器	142
0x284	UNPEND1	R/W	0x0000.0000	中断 32-63 清除挂起寄存器	142
0x288	UNPEND2	R/W	0x0000.0000	中断 64-95 清除挂起寄存器	142
0x28C	UNPEND3	R/W	0x0000.0000	中断 96-127 清除挂起寄存器	142
0x290	UNPEND4	R/W	0x0000.0000	中断 128-138 清除挂起寄存器	143
0x300	ACTIVE0	RO	0x0000.0000	中断 0-31 活动位寄存器	144
0x304	ACTIVE1	RO	0x0000.0000	中断 32-63 活动位寄存器	144
0x308	ACTIVE2	RO	0x0000.0000	中断 64-95 活动位寄存器	144
0x30C	ACTIVE3	RO	0x0000.0000	中断 96-127 活动位寄存器	144
0x310	ACTIVE4	RO	0x0000.0000	中断 128-138 活动位寄存器	145
0x400	PRI0	R/W	0x0000.0000	中断 0-3 优先级寄存器	146
0x404	PRI1	R/W	0x0000.0000	中断 4-7 优先级寄存器	146
0x408	PRI2	R/W	0x0000.0000	中断 8-11 优先级寄存器	146
0x40C	PRI3	R/W	0x0000.0000	中断 12-15 优先级寄存器	146
0x410	PRI4	R/W	0x0000.0000	中断 16-19 优先级寄存器	146
0x414	PRI5	R/W	0x0000.0000	中断 20-23 优先级寄存器	146
0x418	PRI6	R/W	0x0000.0000	中断 24-27 优先级寄存器	146
0x41C	PRI7	R/W	0x0000.0000	中断 28-31 优先级寄存器	146
0x420	PRI8	R/W	0x0000.0000	中断 32-35 优先级寄存器	146

表 3-8. 外设 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0x424	PRI9	R/W	0x0000.0000	中断 36-39 优先级寄存器	146
0x428	PRI10	R/W	0x0000.0000	中断 40-43 优先级寄存器	146
0x42C	PRI11	R/W	0x0000.0000	中断 44-47 优先级寄存器	146
0x430	PRI12	R/W	0x0000.0000	中断 48-51 优先级寄存器	146
0x434	PRI13	R/W	0x0000.0000	中断 52-55 优先级寄存器	146
0x438	PRI14	R/W	0x0000.0000	中断 56-59 优先级寄存器	146
0x43C	PRI15	R/W	0x0000.0000	中断 60-63 优先级寄存器	146
0x440	PRI16	R/W	0x0000.0000	中断 64-67 优先级寄存器	148
0x444	PRI17	R/W	0x0000.0000	中断 68-71 优先级寄存器	148
0x448	PRI18	R/W	0x0000.0000	中断 72-75 优先级寄存器	148
0x44C	PRI19	R/W	0x0000.0000	中断 76-79 优先级寄存器	148
0x450	PRI20	R/W	0x0000.0000	中断 80-83 优先级寄存器	148
0x454	PRI21	R/W	0x0000.0000	中断 84-87 优先级寄存器	148
0x458	PRI22	R/W	0x0000.0000	中断 88-91 优先级寄存器	148
0x45C	PRI23	R/W	0x0000.0000	中断 92-95 优先级寄存器	148
0x460	PRI24	R/W	0x0000.0000	中断 96-99 优先级寄存器	148
0x464	PRI25	R/W	0x0000.0000	中断 100-103 优先级寄存器	148
0x468	PRI26	R/W	0x0000.0000	中断 104-107 优先级寄存器	148
0x46C	PRI27	R/W	0x0000.0000	中断 108-111 优先级寄存器	148
0x470	PRI28	R/W	0x0000.0000	中断 112-115 优先级寄存器	148
0x474	PRI29	R/W	0x0000.0000	中断 116-119 优先级寄存器	148
0x478	PRI30	R/W	0x0000.0000	中断 120-123 优先级寄存器	148
0x47C	PRI31	R/W	0x0000.0000	中断 124-127 优先级寄存器	148
0x480	PRI32	R/W	0x0000.0000	中断 128-131 优先级寄存器	148
0x484	PRI33	R/W	0x0000.0000	中断 132-135 优先级寄存器	148
0x488	PRI34	R/W	0x0000.0000	中断 136-138 优先级寄存器	148
0xF00	SWTRIG	WO	0x0000.0000	软件触发中断寄存器	150
系统控制模块 ( SCB ) 寄存器					
0x008	ACTLR	R/W	0x0000.0000	辅助控制寄存器	151
0xD00	CPUID	RO	0x410F.C241	CPU ID 基础寄存器	153
0xD04	INTCTRL	R/W	0x0000.0000	中断控制及状态寄存器	154
0xD08	VTABLE	R/W	0x0000.0000	向量表偏移寄存器	157
0xD0C	APINT	R/W	0xFA05.0000	应用程序中断及复位控制寄存器	158

表 3-8. 外设 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0xD10	SYSCTRL	R/W	0x0000.0000	系统控制寄存器	160
0xD14	CFGCTRL	R/W	0x0000.0200	配置及控制寄存器	162
0xD18	SYSPRI1	R/W	0x0000.0000	系统处理程序优先级寄存器 1	164
0xD1C	SYSPRI2	R/W	0x0000.0000	系统处理程序优先级寄存器 2	165
0xD20	SYSPRI3	R/W	0x0000.0000	系统处理程序优先级寄存器 3	166
0xD24	SYSHNDCTRL	R/W	0x0000.0000	系统处理程序控制及状态寄存器	167
0xD28	FAULTSTAT	R/W1C	0x0000.0000	可配置故障状态寄存器	170
0xD2C	HFAULTSTAT	R/W1C	0x0000.0000	硬故障状态寄存器	176
0xD34	MMADDR	R/W	-	存储器管理故障地址寄存器	177
0xD38	FAULTADDR	R/W	-	总线故障地址寄存器	178
存储器保护单元 ( MPU ) 寄存器					
0xD90	MPUTYPE	RO	0x0000.0800	MPU 类型寄存器	179
0xD94	MPUCTRL	R/W	0x0000.0000	MPU 控制寄存器	180
0xD98	MPUNUMBER	R/W	0x0000.0000	MPU 区编号寄存器	182
0xD9C	MPUBASE	R/W	0x0000.0000	MPU 区基地址寄存器	183
0xDA0	MPUATTR	R/W	0x0000.0000	MPU 区属性和大小寄存器	185
0xDA4	MPUBASE1	R/W	0x0000.0000	MPU 区基地址别名寄存器 1	183
0xDA8	MPUATTR1	R/W	0x0000.0000	MPU 区属性和大小别名寄存器 1	185
0xDAC	MPUBASE2	R/W	0x0000.0000	MPU 区基地址别名寄存器 2	183
0xDB0	MPUATTR2	R/W	0x0000.0000	MPU 区属性和大小别名寄存器 2	185
0xDB4	MPUBASE3	R/W	0x0000.0000	MPU 区基地址别名寄存器 3	183
0xDB8	MPUATTR3	R/W	0x0000.0000	MPU 区属性和大小别名寄存器 3	185
浮点单元 ( FPU ) 寄存器					
0xD88	CPAC	R/W	0x0000.0000	协处理器访问控制	187
0xF34	FPCC	R/W	0xC000.0000	浮点上下文控制	188
0xF38	FPCA	R/W	-	浮点上下文访问	190
0xF3C	FPDSC	R/W	0x0000.0000	浮点默认状态控制	191

### 3.3 系统定时器 ( SysTick ) 寄存器描述

本节按照地址偏移量由小到大的顺序依次详细介绍系统定时器的各寄存器。

## 寄存器 1: SysTick 控制及状态寄存器 ( STCTRL ) , 偏移量 0x010

注意: 本寄存器只能在特权模式下访问。

SysTick STCTRL 寄存器启用 SysTick 功能。

### SysTick 控制及状态寄存器 (STCTRL)

基址 0xE000.E000  
偏移量 0x010  
类型 R/W, 复位 0x0000.0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留															COUNT	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													CLK_SRC	INTEN	ENABLE	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

位/域	名称	类型	复位	描述						
31:17	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。						
16	COUNT	RO	0	计数标志 <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>自上一次读取此位之后，SysTick 定时器尚未计数到 0。</td> </tr> <tr> <td>1</td> <td>自上一次读取本标志位之后，系统定时器已计数到 0。</td> </tr> </table> <p>读取寄存器或者如果向 STCURRENT 寄存器写入任何值，会清空该位。 如果调试器使用 DAP 读取，仅在 AHB-AP 控制寄存器中的 MasterType 位清零时，该位才清零。否则，COUNT 位不会因为调试器的读操作而改变。关于 MasterType 的更多信息，请参考“《ARM® 调试接口 V5 架构规范》”。</p>	值	描述	0	自上一次读取此位之后，SysTick 定时器尚未计数到 0。	1	自上一次读取本标志位之后，系统定时器已计数到 0。
值	描述									
0	自上一次读取此位之后，SysTick 定时器尚未计数到 0。									
1	自上一次读取本标志位之后，系统定时器已计数到 0。									
15:3	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。						
2	CLK_SRC	R/W	1	时钟源 <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>精确内部振荡器 (PIOSC) 4 分频</td> </tr> <tr> <td>1</td> <td>系统时钟。</td> </tr> </table>	值	描述	0	精确内部振荡器 (PIOSC) 4 分频	1	系统时钟。
值	描述									
0	精确内部振荡器 (PIOSC) 4 分频									
1	系统时钟。									
1	INTEN	R/W	0	中断启用 <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>禁用中断产生软件可以使用 COUNT 位来确定计数器是否曾达到 0。</td> </tr> <tr> <td>1</td> <td>SysTick 计数到 0 时，向 NVIC 产生一个中断。</td> </tr> </table>	值	描述	0	禁用中断产生软件可以使用 COUNT 位来确定计数器是否曾达到 0。	1	SysTick 计数到 0 时，向 NVIC 产生一个中断。
值	描述									
0	禁用中断产生软件可以使用 COUNT 位来确定计数器是否曾达到 0。									
1	SysTick 计数到 0 时，向 NVIC 产生一个中断。									

位/域	名称	类型	复位	描述
0	ENABLE	R/W	0	启用
				值 描述
				0 禁用计数器。
				1 使 SysTick 以多次触发方式进行操作。即，计数器装入 RELOAD 值并开始递减计数。达到 0 时，COUNT 位将置位，如果由 INTEN 启用，将生成一个中断。然后，计数器重新装入 RELOAD 值并开始计数。

## 寄存器 2: SysTick 重载值寄存器 ( STRELOAD ) , 偏移量 0x014

注意: 本寄存器只能在特权模式下访问。

STRELOAD 寄存器指定在计数器达到 0 的时候要装入 SysTick 当前值 (STCURRENT) 寄存器的开始值。开始值可以在 0x1 和 0x00FF.FFFF 之间。因为从 1 计数到 0 时激活了 SysTick 中断和 COUNT 位, 所以开始值 0 是可能的, 但是没有效果。

系统定时器按照自动重载方式工作, 每 N+1 个时钟动作一次 ( N 的取值范围为 1~0x00FF.FFFF ) , 如是周而复始、永不停歇。例如, 如果每 100 次时钟脉冲就需要一次节拍中断, 则必须将 99 写入 RELOAD 域。

请注意, 为了正确访问此寄存器, 系统时钟速度必须超过 8 MHz。

### SysTick 重载值寄存器 (STRELOAD)

基址 0xE000.E000

偏移量 0x014

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								RELOAD							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RELOAD															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:24	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
23:0	RELOAD	R/W	0x00.0000	重载值 在计数器达到 0 时要装入 SysTick 当前值 (STCURRENT) 寄存器的值。

### 寄存器 3: SysTick 当前值寄存器 ( STCURRENT ) , 偏移量 0x018

注意: 本寄存器只能在特权模式下访问。

STCURRENT 寄存器包含 SysTick 计数器的当前值。

#### SysTick 当前值寄存器 (STCURRENT)

基址 0xE000.E000

偏移量 0x018

类型 R/WC, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								CURRENT							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CURRENT															
类型	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:24	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
23:0	CURRENT	R/WC	0x00.0000	当前值 该域包含访问寄存器时的当前值。没有提供读-修改-写保护, 所以在更改时要特别注意。 该寄存器是写清零。向该寄存器写入任意值都会将寄存器清零。清空此寄存器还会清零 STCTRL 寄存器的 COUNT 位。

## 3.4 NVIC 寄存器描述

本节按照地址偏移量由小到大的顺序依次详细介绍NVIC的各寄存器。

NVIC 寄存器只能在特权模式下才能完全访问, 在非特权模式下启用配置和控制 (CFGCTRL) 寄存器来暂挂中断。对其它NVIC寄存器的访问都会导致产生总线故障。

软件访问寄存器时应确保其对齐正确。处理器不支持对NVIC寄存器执行非对齐访问。

即使某个中断已被禁用, 该中断也能够进入挂起状态。

在对 VTABLE 寄存器编程以重定位向量表之前, 应确保已经为故障处理程序 NMI 以及所有已启用的异常 (例如中断) 设置了新向量表的向量表条目。关于更多信息, 请参阅 157页。

寄存器 4: 中断 0-31 集启用寄存器 (EN0), 偏移量 0x100

寄存器 5: 中断 32-63 集启用寄存器 (EN1), 偏移量 0x104

寄存器 6: 中断 64-95 集启用寄存器 (EN2), 偏移量 0x108

寄存器 7: 中断 96-127 集启用寄存器 (EN3), 偏移量 0x10C

注意: 本寄存器只能在特权模式下访问。

ENn 寄存器启用中断并显示启用了哪些中断。EN0 的第 0 位对应于中断 0; 第 31 位对应于中断 31。EN1 的第 0 位对应于中断 32; 第 31 位对应于中断 63。EN2 的第 0 位对应于中断 64; 第 31 位对应于中断 95。EN3 的第 0 位对应于中断 96; 第 31 位对应于中断 127。EN4 的第 0 位 (参见 137 页) 对应于中断 128; 第 10 位对应于中断 138。

请参阅表 2-9 在 101 页以了解中断分配。

假如某个已挂起的中断被启用, NVIC 将按照其优先级激活中断。假如某个中断未启用, 那么当中断源产生中断信号后, 虽然该中断状态变为挂起, 但 NVIC 永远不会按照其优先级将其激活。

#### 中断 0-31 集启用寄存器 (EN0)

基址 0xE000.E000  
偏移量 0x100  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	INT	R/W	0x0000.0000	中断启用

值	描述
0	读数为 0, 代表该中断已被禁用。 写入 0 时, 无意义。
1	读数为 1, 代表该中断已经启用。 写入 1 时, 将启用对应中断。

只能通过在 DISn 寄存器中设置相应的 INT[n] 位来清除某个位。



## 寄存器 8: 中断 128-138 设置启用寄存器 (EN4) , 偏移量 0x110

注意: 本寄存器只能在特权模式下访问。

EN4 寄存器启用中断并显示启用了哪些中断。0 位对应中断 128 ; 10 位对应中断 138。请参阅表 2-9 在 101页 以了解中断分配。

假如某个已挂起的中断被启用, NVIC将按照其优先级激活中断。假如某个中断未启用, 那么当中断源产生中断信号后, 虽然该中断状态变为挂起, 但NVIC永远不会按照其优先级将其激活。

### 中断 128-138 设置启用寄存器 (EN4)

基址 0xE000.E000  
偏移量 0x110  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留					INT										
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:11	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
10:0	INT	R/W	0x0	中断启用
	值	描述		
	0	读出为0, 代表该中断已被禁用。 写入0时, 无意义。		
	1	读出为1, 代表该中断已经启用。 写入1时, 将启用对应中断。		

只能通过在 DIS4 寄存器中设置相应的 INT[n] 位来清除某个位。

寄存器 9: 中断 0-31 清除启用寄存器 (DIS0), 偏移量 0x180

寄存器 10: 中断 32-63 清除启用寄存器 (DIS1), 偏移量 0x184

寄存器 11: 中断 64-95 清除启用寄存器 (DIS2), 偏移量 0x188

寄存器 12: 中断 96-127 清除启用寄存器 (DIS3), 偏移量 0x18C

注意: 本寄存器只能在特权模式下访问。

DISn 寄存器禁用中断。DIS0 的第 0 位对应于中断 0; 第 31 位对应于中断 31。DIS1 的第 0 位对应于中断 32; 第 31 位对应于中断 63。DIS2 的第 0 位对应于中断 64; 第 31 位对应于中断 95。DIS3 的第 0 位对应于中断 96; 第 31 位对应于中断 127。DIS4 的第 0 位 (参见 139 页) 对应于中断 128; 第 10 位对应于中断 138。

请参阅表 2-9 在 101 页 以了解中断分配。

#### 中断 0-31 清除启用寄存器 (DIS0)

基址 0xE000.E000

偏移量 0x180

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	INT	R/W	0x0000.0000	中断禁用

#### 值 描述

0 读数为 0, 代表该中断已被禁用。

写入 0 时, 无意义。

1 读数为 1, 代表该中断已经启用。

写入时, 清除 EN0 寄存器中相应的 INT[n] 位即可禁用中断 [n]。

## 寄存器 13: 中断 128-138 清除启用寄存器 (DIS4) , 偏移量 0x190

注意: 本寄存器只能在特权模式下访问。

DIS4 寄存器禁用中断。0 位对应中断 128 ; 10 位对应中断 138。请参阅表 2-9 在 101 页以了解中断分配。

### 中断 128-138 清除启用寄存器 (DIS4)

基址 0xE000.E000

偏移量 0x190

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留					INT										
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:11	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
10:0	INT	R/W	0x0	中断禁用
				值 描述
				0 读出为0，代表该中断已被禁用。 写入0时，无意义。
				1 读出为1，代表该中断已经启用。 写入时，清除 EN4 寄存器中相应的 INT[n] 位即可禁用中断 [n]。

寄存器 14: 中断 0-31 置位挂起寄存器 (PEND0), 偏移量 0x200

寄存器 15: 中断 32-63 置位挂起寄存器 (PEND1), 偏移量 0x204

寄存器 16: 中断 64-95 置位挂起寄存器 (PEND2), 偏移量 0x208

寄存器 17: 中断 96-127 置位挂起寄存器 (PEND3), 偏移量 0x20C

注意: 本寄存器只能在特权模式下访问。

PENDn 寄存器强制将中断置为挂起状态, 并显示哪些中断处于挂起状态。PEND0 的第 0 位对应于中断 0; 第 31 位对应于中断 31。PEND1 的第 0 位对应于中断 32; 第 31 位对应于中断 63。PEND2 的第 0 位对应于中断 64; 第 31 位对应于中断 95。PEND3 的第 0 位对应于中断 96; 第 31 位对应于中断 127。PEND4 的第 0 位 (参见 141 页) 对应于中断 128; 第 10 位对应于中断 138。

请参阅表 2-9 在 101 页以了解中断分配。

#### 中断 0-31 置位挂起寄存器 (PEND0)

基址 0xE000.E000  
偏移量 0x200  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	INT	R/W	0x0000.0000	中断置位暂挂

值	描述
0	读出为 0, 代表该中断未挂起。 写入 0 时, 无意义。
1	读出为 1, 代表该中断已挂起。 写入 1 时, 对应中断的状态将变为挂起, 即使该中断已被禁用也是如此。

如果对应的中断已经暂挂, 那么该位置位将没有效果。  
只能将 UNPEND0 寄存器中的相应 INT[n] 位置位来清除该位。

## 寄存器 18: 中断 128-138 置位挂起寄存器 (PEND4), 偏移量 0x210

注意: 本寄存器只能在特权模式下访问。

PEND4 寄存器强制将中断置为暂挂状态并显示哪些中断处于暂挂状态。0 位对应中断 128; 10 位对应中断 138。请参阅表 2-9 在 101 页以了解中断分配。

### 中断 128-138 置位挂起寄存器 (PEND4)

基址 0xE000.E000

偏移量 0x210

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留					INT										
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:11	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
10:0	INT	R/W	0x0	中断置位暂挂
			值	描述
			0	读数为0, 代表该中断未挂起。 写入0时, 无意义。
			1	读数为1, 代表该中断已挂起。 写入1时, 对应中断的状态将变为挂起, 即使该中断已被禁用也是如此。

如果对应的中断已经暂挂, 那么该位置位将没有效果。

只能将 UNPEND4 寄存器中的相应 INT[n] 位置位来清除该位。

寄存器 19: 中断 0-31 清除挂起寄存器 ( UNPEND0 ) , 偏移量 0x280

寄存器 20: 中断 32-63 清除挂起寄存器 ( UNPEND1 ) , 偏移量 0x284

寄存器 21: 中断 64-95 清除挂起寄存器 ( UNPEND2 ) , 偏移量 0x288

寄存器 22: 中断 96-127 清除挂起寄存器 ( UNPEND3 ) , 偏移量 0x28C

注意: 本寄存器只能在特权模式下访问。

UNPENDn 寄存器显示哪些中断处于暂挂状态并从中断除去暂挂状态。UNPEND0 的第 0 位对应于中断 0 ; 第 31 位对应于中断 31。UNPEND1 的第 0 位对应于中断 32 ; 第 31 位对应于中断 63。UNPEND2 的第 0 位对应于中断 64 ; 第 31 位对应于中断 95。UNPEND3 的第 0 位对应于中断 96 ; 第 31 位对应于中断 127。UNPEND4 的第 0 位 ( 参见 143 页 ) 对应于中断 128 ; 第 10 位对应于中断 138。

请参阅表 2-9 在 101 页 以了解中断分配。

#### 中断 0-31 清除挂起寄存器 (UNPEND0)

基址 0xE000.E000  
偏移量 0x280  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	INT	R/W	0x0000.0000	中断清除暂挂

#### 值 描述

0 读出为 0, 代表该中断未挂起。

写入 0 时, 无意义。

1 读出为 1, 代表该中断已挂起。

写入时, 清除 PEND0 寄存器中的相应 INT[n] 位, 以便使中断 [n] 不再处于暂挂状态

如果某个中断已激活, 那么写 1 将失去实际意义。

**寄存器 23: 中断 128-138 清除挂起寄存器 ( UNPEND4 ) , 偏移量 0x290**

注意: 本寄存器只能在特权模式下访问。

UNPEND4 寄存器显示哪些中断处于暂挂状态并清除中断的暂挂状态。0 位对应中断 128 ; 10 位对应中断 138。请参阅表 2-9 在 101页以了解中断分配。

**中断 128-138 清除挂起寄存器 (UNPEND4)**

基址 0xE000.E000  
偏移量 0x290  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留					INT										
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:11	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
10:0	INT	R/W	0x0	中断清除暂挂

**值 描述**

0 读出为0，代表该中断未挂起。

写入0时，无意义。

1 读出为1，代表该中断已挂起。

写入时，清除 PENDING 寄存器中的相应 INT[n] 位，以便使中断 [n] 不再处于暂挂状态

如果某个中断已激活，那么写1将失去实际意义。

寄存器 24: 中断 0-31 活动位寄存器 (ACTIVE0), 偏移量 0x300

寄存器 25: 中断 32-63 活动位寄存器 (ACTIVE1), 偏移量 0x304

寄存器 26: 中断 64-95 活动位寄存器 (ACTIVE2), 偏移量 0x308

寄存器 27: 中断 96-127 活动位寄存器 (ACTIVE3), 偏移量 0x30C

注意: 本寄存器只能在特权模式下访问。

UNPENDn 寄存器指示哪些中断处于激活状态。ACTIVE0 的第 0 位对应于中断 0; 第 31 位对应于中断 31。ACTIVE1 的第 0 位对应于中断 32; 第 31 位对应于中断 63。ACTIVE2 的第 0 位对应于中断 64; 第 31 位对应于中断 95。ACTIVE3 的第 0 位对应于中断 96; 第 31 位对应于中断 127。ACTIVE4 的第 0 位 (参见 145 页) 对应于中断 128; 第 10 位对应于中断 138。

请参阅表 2-9 在 101 页以了解中断分配。

小心 – 切勿手动将本寄存器中的位置位或清零。

#### 中断 0-31 活动位寄存器 (ACTIVE0)

基址 0xE000.E000  
偏移量 0x300  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INT															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	INT	RO	0x0000.0000	中断激活

#### 值 描述

0 对应中断未激活。

1 对应中断已激活, 或处于激活并等待状态。



**寄存器 28: 中断 128-138 活动位寄存器 ( ACTIVE4 ) , 偏移量 0x310**

注意: 本寄存器只能在特权模式下访问。

ACTIVE4 寄存器指示哪些中断处于激活状态。0 位对应中断 128 ; 10 位对应中断 131。请参阅表 2-9 在 101 页以了解中断分配。

小心 – 切勿手动将本寄存器中的位置位或清零。

**中断 128-138 活动位寄存器 (ACTIVE4)**

基址 0xE000.E000  
偏移量 0x310  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留					INT										
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:11	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
10:0	INT	RO	0x0	中断激活
				值 描述
				0 对应中断未激活。
				1 对应中断已激活，或处于激活并等待状态。

- 寄存器 29: 中断 0-3 优先级寄存器 ( PRI0 ) , 偏移量 0x400  
 寄存器 30: 中断 4-7 优先级寄存器 ( PRI1 ) , 偏移量 0x404  
 寄存器 31: 中断 8-11 优先级寄存器 ( PRI2 ) , 偏移量 0x408  
 寄存器 32: 中断 12-15 优先级寄存器 ( PRI3 ) , 偏移量 0x40C  
 寄存器 33: 中断 16-19 优先级寄存器 ( PRI4 ) , 偏移量 0x410  
 寄存器 34: 中断 20-23 优先级寄存器 ( PRI5 ) , 偏移量 0x414  
 寄存器 35: 中断 24-27 优先级寄存器 ( PRI6 ) , 偏移量 0x418  
 寄存器 36: 中断 28-31 优先级寄存器 ( PRI7 ) , 偏移量 0x41C  
 寄存器 37: 中断 32-35 优先级寄存器 ( PRI8 ) , 偏移量 0x420  
 寄存器 38: 中断 36-39 优先级寄存器 ( PRI9 ) , 偏移量 0x424  
 寄存器 39: 中断 40-43 优先级寄存器 ( PRI10 ) , 偏移量 0x428  
 寄存器 40: 中断 44-47 优先级寄存器 ( PRI11 ) , 偏移量 0x42C  
 寄存器 41: 中断 48-51 优先级寄存器 ( PRI12 ) , 偏移量 0x430  
 寄存器 42: 中断 52-55 优先级寄存器 ( PRI13 ) , 偏移量 0x434  
 寄存器 43: 中断 56-59 优先级寄存器 ( PRI14 ) , 偏移量 0x438  
 寄存器 44: 中断 60-63 优先级寄存器 ( PRI15 ) , 偏移量 0x43C

注意: 本寄存器只能在特权模式下访问。

PRIn 寄存器 ( 另请参阅 148页 ) 为每个中断提供 3 位的优先级域。所有PRIn寄存器都是可字节访问的。每个寄存器中包含4个中断的优先级位域, 如下所示:

PRIn 寄存器位域	中断
位 31:29	中断 [4n+3]
位 23:21	中断 [4n+2]
位 15:13	中断 [4n+1]
位 7:5	中断 [4n]

请参阅表 2-9 在 101页以了解中断分配。

优先级位域还可以进一步划分为组优先级位域以及子优先级位域。应用程序中断及复位控制 (APINT) 寄存器中的 PRIGROUP 域 ( 请参阅 158页 ) 指示划分优先级和亚优先级域的二进制点的位置。

这些寄存器只能在特权模式下访问。

## 中断 0-3 优先级寄存器 (PRIO)

基址 0xE000.E000

偏移量 0x400

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTD			保留				INTC			保留					
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTB			保留				INTA			保留					
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	INTD	R/W	0x0	中断[4n+3]的中断优先级 此域为编号 [4n+3] 的中断保留优先级值 0-7，其中 n 是中断优先级寄存器的编号 (n=0 表示 PRI0，依此类推)。该值越低，相应中断的优先级越高。
28:24	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
23:21	INTC	R/W	0x0	中断[4n+2]的中断优先级 此域为编号 [4n+2] 的中断保留优先级值 0-7，其中 n 是中断优先级寄存器的编号 (n=0 表示 PRI0，依此类推)。该值越低，相应中断的优先级越高。
20:16	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:13	INTB	R/W	0x0	中断[4n+1]的中断优先级 此域为编号 [4n+1] 的中断保留优先级值 0-7，其中 n 是中断优先级寄存器的编号 (n=0 表示 PRI0，依此类推)。该值越低，相应中断的优先级越高。
12:8	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:5	INTA	R/W	0x0	中断[4n] 的中断优先级 此域为编号 [4n] 的中断保留优先级值 0-7，其中 n 是中断优先级寄存器的编号 (n=0 表示 PRI0，依此类推)。该值越低，相应中断的优先级越高。
4:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

- 寄存器 45: 中断 64-67 优先级寄存器 ( PRI16 ) , 偏移量 0x440  
 寄存器 46: 中断 68-71 优先级寄存器 ( PRI17 ) , 偏移量 0x444  
 寄存器 47: 中断 72-75 优先级寄存器 ( PRI18 ) , 偏移量 0x448  
 寄存器 48: 中断 76-79 优先级寄存器 ( PRI19 ) , 偏移量 0x44C  
 寄存器 49: 中断 80-83 优先级寄存器 ( PRI20 ) , 偏移量 0x450  
 寄存器 50: 中断 84-87 优先级寄存器 ( PRI21 ) , 偏移量 0x454  
 寄存器 51: 中断 88-91 优先级寄存器 ( PRI22 ) , 偏移量 0x458  
 寄存器 52: 中断 92-95 优先级寄存器 ( PRI23 ) , 偏移量 0x45C  
 寄存器 53: 中断 96-99 优先级寄存器 ( PRI24 ) , 偏移量 0x460  
 寄存器 54: 中断 100-103 优先级寄存器 ( PRI25 ) , 偏移量 0x464  
 寄存器 55: 中断 104-107 优先级寄存器 ( PRI26 ) , 偏移量 0x468  
 寄存器 56: 中断 108-111 优先级寄存器 ( PRI27 ) , 偏移量 0x46C  
 寄存器 57: 中断 112-115 优先级寄存器 ( PRI28 ) , 偏移量 0x470  
 寄存器 58: 中断 116-119 优先级寄存器 ( PRI29 ) , 偏移量 0x474  
 寄存器 59: 中断 120-123 优先级寄存器 ( PRI30 ) , 偏移量 0x478  
 寄存器 60: 中断 124-127 优先级寄存器 ( PRI31 ) , 偏移量 0x47C  
 寄存器 61: 中断 128-131 优先级寄存器 ( PRI32 ) , 偏移量 0x480  
 寄存器 62: 中断 132-135 优先级寄存器 ( PRI33 ) , 偏移量 0x484  
 寄存器 63: 中断 136-138 优先级寄存器 ( PRI34 ) , 偏移量 0x488

注意: 本寄存器只能在特权模式下访问。

PRIn 寄存器 ( 另请参阅 146页 ) 为每个中断提供 3 位的优先级域。所有PRIn寄存器都是可字节访问的。每个寄存器中包含4个中断的优先级位域, 如下所示:

PRIn 寄存器位域	中断
位 31:29	中断 [4n+3]
位 23:21	中断 [4n+2]
位 15:13	中断 [4n+1]
位 7:5	中断 [4n]

请参阅表 2-9 在 101页以了解中断分配。

优先级位域还可以进一步划分为组优先级位域以及子优先级位域。应用程序中断及复位控制 (APINT) 寄存器中的 PRIGROUP 域 ( 请参阅 158页 ) 指示划分优先级和亚优先级域的二进制点的位置。

这些寄存器只能在特权模式下访问。

## 中断 64-67 优先级寄存器 (PRI16)

基址 0xE000.E000

偏移量 0x440

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTD			保留				INTC			保留					
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTB			保留				INTA			保留					
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	INTD	R/W	0x0	中断[4n+3]的中断优先级 此域为编号 [4n+3] 的中断保留优先级值 0-7，其中 n 是中断优先级寄存器的编号 (n=0 表示 PRI0，依此类推)。该值越低，相应中断的优先级越高。
28:24	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
23:21	INTC	R/W	0x0	中断[4n+2]的中断优先级 此域为编号 [4n+2] 的中断保留优先级值 0-7，其中 n 是中断优先级寄存器的编号 (n=0 表示 PRI0，依此类推)。该值越低，相应中断的优先级越高。
20:16	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:13	INTB	R/W	0x0	中断[4n+1]的中断优先级 此域为编号 [4n+1] 的中断保留优先级值 0-7，其中 n 是中断优先级寄存器的编号 (n=0 表示 PRI0，依此类推)。该值越低，相应中断的优先级越高。
12:8	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:5	INTA	R/W	0x0	中断[4n] 的中断优先级 此域为编号 [4n] 的中断保留优先级值 0-7，其中 n 是中断优先级寄存器的编号 (n=0 表示 PRI0，依此类推)。该值越低，相应中断的优先级越高。
4:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

**寄存器 64: 软件触发中断寄存器 ( SWTRIG ) , 偏移量 0xF00**

注意: 只有特权软件可以启用对 SWTRIG 寄存器的非特权访问。

向 SWTRIG 寄存器写入中断编号会产生“软件生成的中断”(SGI)。请参阅表 2-9 在 101 页以了解中断分配。

将配置和控制 (CFGCTRL) 中的 MAINPEND 位置位时 ( 请参阅 162 页 ) , 非特权软件可以访问 SWTRIG 寄存器。

**软件触发中断寄存器 (SWTRIG)**

基址 0xE000.E000  
偏移量 0xF00  
类型 WO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								INTID							
类型	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO	WO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	INTID	WO	0x00	中断标识 此域保留所需 SGI 的中断标识。例如, 写入值 0x3 会在 IRQ3 上生成中断。

**3.5 系统控制模块 ( SCB ) 寄存器描述**

本节按照地址偏移量由小到大的顺序依次详细介绍系统控制模块 ( SCB ) 的各寄存器。SCB 寄存器只能在特权模式下访问。

所有寄存器必须以对齐字访问进行访问, 除了 FAULTSTAT 和 SYSPRI1-SYSPRI3 寄存器, 这两个寄存器可以使用字节或对齐半字或字访问进行访问。处理器不支持对 SCB 寄存器执行非对齐访问。

## 寄存器 65: 辅助控制寄存器 ( ACTLR ) , 偏移量 0x008

注意: 本寄存器只能在特权模式下访问。

ACTLR 寄存器为禁用以下功能提供控制位: IT 堆叠; 访问默认存储器映射时的写缓冲; 对多周期指令的中断。本寄存器的默认配置已经为 Cortex-M4 处理器内核实现了优化的性能, 一般来说无需修改。

### 辅助控制寄存器 (ACTLR)

基址 0xE000.E000

偏移量 0x008

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留						DISOFP	DISFPCA	保留					DISFOLD	DISWBUF	DISMCYC
类型	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:10	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
9	DISOFP	R/W	0	禁用无序浮点 禁用相对于整数指令的无序完成的浮点指令。
8	DISFPCA	R/W	0	禁用 CONTROL.FPCA 禁用 CONTROL 寄存器中 FPCA 位的自动更新。
7:3	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2	DISFOLD	R/W	0	禁用 IT 堆叠。

**重要:** 两个位控制何时启用 FPCA: 浮点上下文控制 (FPCC) 寄存器中的 ASPEN 位和 辅助控制 (ACTLR) 寄存器中的 DISFPCA 位。

#### 值 描述

0 无影响。

1 禁用 IT 堆叠。

执行 IT 指令时, 在某些情况下, 处理器可以继续执行 IT 模块中的第一条指令。这种行为即 *IT 堆叠*, 可以提高性能, 但是, IT 堆叠会引起循环抖动。如果某项任务禁止产生抖动, 那么必须在执行任务之前将 DISFOLD 位置位以禁用 IT 堆叠。

位/域	名称	类型	复位	描述
1	DISWBUF	R/W	0	<p>禁用写入缓冲</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 在访问默认的存储空间时禁用写入缓冲。在此情况下，一旦产生总线故障，则必然是精确总线故障；但处理器性能将会有所下降，因为处理器要等待对存储器的存贮操作完全结束后才会执行下一条指令。</p> <p>注意： 该位只影响 Cortex-M4 处理器中执行的写入缓冲。</p>
0	DISMCYC	R/W	0	<p>禁用多周期指令的中断</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 禁止在多周期装载指令以及多周期存贮指令执行期间产生中断。若本标志位置位，由于处理器必须等待 LDM 或 STM 指令执行完成后才将当前状态入栈并进入中断处理程序，因此处理器的中断延时将会增加。</p>



## 寄存器 66: CPU ID 基础寄存器 ( CPUID ) , 偏移量 0xD00

注意: 本寄存器只能在特权模式下访问。

CPUID 寄存器中包含 ARM® Cortex™-M4 处理器的芯片编号、版本以及其它实施信息。

### CPU ID 基础寄存器 (CPUID)

基址 0xE000.E000  
偏移量 0xD00  
类型 RO, 复位 0x410F.C241

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMP								VAR				CON			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PARTNO												REV			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:24	IMP	RO	0x41	标准制订商代码  值 描述 0x41 ARM
23:20	VAR	RO	0x0	变化版本编号  值 描述 0x0 即 rmpn 产品版本识别号中的 rn 值, 比如 r0p0 中的 0。
19:16	CON	RO	0xF	常量  值 描述 0xF 始终读取为 0xF。
15:4	PARTNO	RO	0xC24	芯片型号  值 描述 0xC24 Cortex-M4 处理器。
3:0	REV	RO	0x1	版本号  值 描述 0x1 即 rmpn 产品版本识别号中的 pn 值, 比如 r0p1 中的 1。

## 寄存器 67: 中断控制及状态寄存器 (INTCTRL), 偏移量 0xD04

注意: 本寄存器只能在特权模式下访问。

INTCTRL 寄存器为 NMI 异常提供了设置挂起位, 为 PendSV 以及 SysTick 异常提供了设置挂起位以及清除挂起位。此外, 本寄存器还能指示出正在处理的异常序号、是否存在抢占式激活的异常、挂起异常中优先级最高的异常序号, 以及是否有中断正在挂起。

若对 INTCTRL 寄存器执行写操作时, 对 PENDSV 和 UNPENDSV 同时写 1 或对 PENDSTSET 和 PENDSTCLR 同时写 1, 可能产生不确定的结果。

## 中断控制及状态寄存器 (INTCTRL)

基址 0xE000.E000  
偏移量 0xD04  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NMISSET	保留		PENDSV	UNPENDSV	PENDSTSET	PENDSTCLR	保留	ISRPRE	ISRPEND	保留		VECPEND			
类型	R/W	RO	RO	R/W	WO	R/W	WO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VECPEND				RETBASE	保留			VECACT							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
-----	----	----	----	----

31	NMISSET	R/W	0	NMI 置位挂起
----	---------	-----	---	----------

## 值 描述

0 读取为 0, 表示 NMI 异常并未挂起。

写入 0 时, 无意义。

1 读出为 1, 代表 NMI 异常正在挂起。

写入 1 时, NMI 异常的状态将变为挂起。

由于 NMI 是最高优先级的异常, 因此一般来说, 处理器只要检测到此标志置位就会立即进入 NMI 异常处理程序, 并在进入中断处理程序后即刻清除此标志位。只有当处理器执行 NMI 异常处理程序期间又再度产生了 NMI 信号, 该位为 1。

30:29	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
-------	----	----	-----	----------------------------------------------------

28	PENDSV	R/W	0	PendSV 置位挂起
----	--------	-----	---	-------------

## 值 描述

0 读出为 0, 代表 PendSV 异常并未挂起。

写入 0 时, 无意义。

1 读出为 1, 代表 PendSV 异常挂起。

写入 1 时, PendSV 异常的状态将变为挂起。

将该位置位是把 PendSV 的异常状态设置成挂起的唯一方法。向 UNPENDSV 位写 1 可以将该位清零。

位/域	名称	类型	复位	描述
27	UNPENDSV	WO	0	<p>PendSV 清除挂起</p> <p>值 描述</p> <p>0 写入0时, 无意义。</p> <p>1 写入 1 时, 清除 PendSV 异常的挂起状态。</p> <p>该位为只写。读本寄存器时该位的值不定。</p>
26	PENDSTSET	R/W	0	<p>SysTick 置位挂起</p> <p>值 描述</p> <p>0 读出为 0, 代表 SysTick 异常并未挂起。 写入0时, 无意义。</p> <p>1 读出为 1, 代表 SysTick 异常挂起。 写入 1 时, SysTick 异常的状态将变为挂起。</p> <p>对 PENDSTCLR 位写 1 即可将该位清零。</p>
25	PENDSTCLR	WO	0	<p>PendSV 清除挂起</p> <p>值 描述</p> <p>0 写入0时, 无意义。</p> <p>1 写入 1 时, 清除 SysTick 异常的挂起状态。</p> <p>该位为只写。读本寄存器时该位的值不定。</p>
24	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
23	ISRPRE	RO	0	<p>调试中断处理</p> <p>值 描述</p> <p>0 处理器从暂停状态恢复运行并非基于中断。</p> <p>1 处理器从暂停状态恢复运行是基于中断。</p> <p>该位只有在调试模式下才有意义, 当处理器不是调试模式时, 该位读取为零。</p>
22	ISRPEND	RO	0	<p>中断挂起</p> <p>值 描述</p> <p>0 无中断挂起。</p> <p>1 有中断正在挂起。</p> <p>该位反映除 NMI 及故障之外的所有中断状态。</p>
21:20	保留	RO	0x0	<p>软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。</p>

位/域	名称	类型	复位	描述																																						
19:12	VECPEND	RO	0x00	<p>挂起中断的向量序号</p> <p>该域包含了最高优先级挂起启用异常的序号。该域的数值与 BASEPRI 和 FAULTMASK 寄存器有关，但是与 PRIMASK 寄存器无关。</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr><td>0x00</td><td>无正在挂起的异常</td></tr> <tr><td>0x01</td><td>保留</td></tr> <tr><td>0x02</td><td>NMI</td></tr> <tr><td>0x03</td><td>硬故障</td></tr> <tr><td>0x04</td><td>存储器管理故障</td></tr> <tr><td>0x05</td><td>总线故障</td></tr> <tr><td>0x06</td><td>用法故障</td></tr> <tr><td>0x07-0x0A</td><td>保留</td></tr> <tr><td>0x0B</td><td>SVCall</td></tr> <tr><td>0x0C</td><td>保留用于调试</td></tr> <tr><td>0x0D</td><td>保留</td></tr> <tr><td>0x0E</td><td>PendSV</td></tr> <tr><td>0x0F</td><td>SysTick</td></tr> <tr><td>0x10</td><td>中断向量 0</td></tr> <tr><td>0x11</td><td>中断向量 1</td></tr> <tr><td>...</td><td>...</td></tr> <tr><td>0x9A</td><td>中断向量 138</td></tr> <tr><td>0x94-0x7F</td><td>保留</td></tr> </tbody> </table>	值	描述	0x00	无正在挂起的异常	0x01	保留	0x02	NMI	0x03	硬故障	0x04	存储器管理故障	0x05	总线故障	0x06	用法故障	0x07-0x0A	保留	0x0B	SVCall	0x0C	保留用于调试	0x0D	保留	0x0E	PendSV	0x0F	SysTick	0x10	中断向量 0	0x11	中断向量 1	...	...	0x9A	中断向量 138	0x94-0x7F	保留
值	描述																																									
0x00	无正在挂起的异常																																									
0x01	保留																																									
0x02	NMI																																									
0x03	硬故障																																									
0x04	存储器管理故障																																									
0x05	总线故障																																									
0x06	用法故障																																									
0x07-0x0A	保留																																									
0x0B	SVCall																																									
0x0C	保留用于调试																																									
0x0D	保留																																									
0x0E	PendSV																																									
0x0F	SysTick																																									
0x10	中断向量 0																																									
0x11	中断向量 1																																									
...	...																																									
0x9A	中断向量 138																																									
0x94-0x7F	保留																																									
11	RETBASE	RO	0	<p>返回基地址</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr><td>0</td><td>有其它抢占式激活异常等待处理。</td></tr> <tr><td>1</td><td>当前无激活的异常，或当前正在处理的异常是唯一激活的异常。</td></tr> </tbody> </table> <p>该位反映除 NMI 及故障之外的所有中断状态。只有当处理器正在执行某个 ISR 时（即中断程序状态 (IPSR) 寄存器非 0），该位才有意义。</p>	值	描述	0	有其它抢占式激活异常等待处理。	1	当前无激活的异常，或当前正在处理的异常是唯一激活的异常。																																
值	描述																																									
0	有其它抢占式激活异常等待处理。																																									
1	当前无激活的异常，或当前正在处理的异常是唯一激活的异常。																																									
10:8	保留	RO	0x0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>																																						
7:0	VECACT	RO	0x00	<p>挂起中断的向量序号</p> <p>该域包含当前激活的异常的序号。各异常的序号请参见 VECPEND 域的介绍。假如该域为 0，则处理器将进入线程模式。该域的值与 IPSR 寄存器的 ISRNUM 域相同。</p> <p>从该值减去 16 以获取编制索引到中断置位启用 (ENn)、中断清除启用 (DISn)、中断置位挂起 (PENDn)、中断清除挂起 (UNPENDn) 和中断优先级 (PRIn) 寄存器中所需的 IRQ 编号（请参阅 80 页）。</p>																																						

## 寄存器 68: 向量表偏移寄存器 (VTABLE), 偏移量 0xD08

注意: 本寄存器只能在特权模式下访问。

VTABLE 寄存器中包含向量表基地址相对于存储器地址 0x0000.0000 的偏移量。

## 向量表偏移寄存器 (VTABLE)

基址 0xE000.E000  
偏移量 0xD08  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留		BASE	OFFSET												
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFFSET						保留									
类型	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:30	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
29	BASE	R/W	0	向量表基地址  值 描述 0 向量表位于代码存储器区。 1 向量表位于 SRAM 存储器区。
28:10	OFFSET	R/W	0x000.00	向量表偏移量 配置 OFFSET 域时, 该偏移量必须与向量表中的异常编号对齐。由于有 138 个中断, 所以偏移量必须按 1024 字节边界对齐。
9:0	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 69: 应用程序中断及复位控制寄存器 ( APINT ) , 偏移量 0xD0C

注意: 本寄存器只能在特权模式下访问。

APINT 寄存器为异常模型提供分组优先级控制, 提供数据访问的端状态, 并提供系统的复位控制。要想对本寄存器执行写操作, 必须对 VECTKEY 域写入 0x05FA, 否则写操作将被忽略。

PRIGROUP 域用于设置二进制小数点的位置, 该小数点将中断优先级 (PRIx) 寄存器中的 INTx 域划分为组优先级域以及子优先级域。表 3-9 在 158 页列出了 PRIGROUP 的值与该划分的关系。表中“组优先级域”以及“子优先级域”两列内容的位序号指的是 INTA 域中的位。对于 INTB 域来说, 相应的位是 15:13; 对于 INTC 域来说, 相应的位是 23:21; 对于 INTD 域来说, 相应的位是 31:29。

注意: 在判定异常的抢占优先级时, 只考虑组优先级域。

表 3-9. 中断优先级分组

PRIGROUP 位域	二进制小数点 <sup>a</sup>	组优先级域	子优先级域	组优先级数目	子优先级数目
0x0 - 0x4	bxxx.	[7:5]	无	8	1
0x5	bxx.y	[7:6]	[5]	4	2
0x6	bx.yy	[7]	[6:5]	2	4
0x7	b.yyy	无	[7:5]	1	8

a. INTx 域显示了二进制小数点的位置。表中 x 代表分组优先级位, y 代表子优先级位。

### 应用程序中断及复位控制寄存器 (APINT)

基址 0xE000.E000  
偏移量 0xD0C  
类型 R/W, 复位 0xFA05.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VECTKEY															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENDIANESS	保留			PRIGROUP			保留			SYSRESREQ			VECTCLRACT	VECTRESET	
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	VECTKEY	R/W	0xFA05	寄存器密钥 该域用来防止对该寄存器的意外写入操作。改变该寄存器的位之前必须将 0x05FA 写入该域。读取返回 0x05FA。
15	ENDIANESS	RO	0	数据端模式 该微控制器的系列仅支持小端模式, 因此该位被清零。
14:11	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
10:8	PRIGROUP	R/W	0x0	中断优先级分组 该域确定分组优先级与子优先级的分隔界限 (更多信息请参考表 3-9 在 158 页)。
7:3	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
2	SYSRESREQ	WO	0	<p>系统复位请求</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 复位处理器内核以及所有片上外设（调试接口除外）。</p> <p>此标志位在处理器内核复位期间自动清零。此标志位的回读值始终为0。</p>
1	VECTCLRACT	WO	0	<p>清除已激活的 NMI/故障</p> <p>该位保留用于调试功能，读取为 0。平时此标志位必须写入 0，否则可能产生无法预料的结果。</p>
0	VECTRESET	WO	0	<p>系统复位</p> <p>该位保留用于调试功能，读取为 0。平时此标志位必须写入 0，否则可能产生无法预料的结果。</p>

## 寄存器 70: 系统控制寄存器 ( SYSCTRL ) , 偏移量 0xD10

注意: 本寄存器只能在特权模式下访问。

SYSCTRL 寄存器负责控制进入/退出低功耗模式的特性。

### 系统控制寄存器 (SYSCTRL)

基址 0xE000.E000  
偏移量 0xD10  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												SEVONPEND	保留	SLEEPDEEP	SLEEPEXIT	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	R/W	R/W	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:5	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
4	SEVONPEND	R/W	0	挂起时唤醒  值 描述 0 只有已启用的中断或事件才能唤醒处理器; 如果某中断已禁用, 则不能唤醒处理器。 1 已启用的事件以及所有中断 (包括已禁用的中断) 都能唤醒处理器。  当一个事件或中断进入挂起状态时, 该事件信号会将处理器从 WFE 中唤醒。如果处理器没有等待事件, 那么该事件被登记, 并会影响下个 WFE。 处理器也可以在执行 SEV 指令时或通过外部事件唤醒。
3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2	SLEEPDEEP	R/W	0	深度睡眠启用  值 描述 0 以睡眠模式作为低功耗模式。 1 以深度睡眠模式作为低功耗模式。
1	SLEEPEXIT	R/W	0	在退出 ISR 后睡眠  值 描述 0 当从处理模式返回到线程模式后, 并不进入睡眠状态。 1 当从处理模式返回到线程模式后, 一退出ISR则立即进入睡眠或深度睡眠状态。  此功能适用于中断驱动型的应用程序。若此标志位置位, 可避免处理器返回到无内容的主函数。



---

位/域	名称	类型	复位	描述
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 71: 配置及控制寄存器 (CFGCTRL), 偏移量 0xD14

注意: 本寄存器只能在特权模式下访问。

CFGCTRL 寄存器负责控制线程模式的入口, 此外还能够启用: NMI 处理程序、硬故障处理程序以及 FAULTMASK 寄存器升级后的故障处理程序忽略总线故障; 被零除以及未对齐访问的额外处理; 非特权级软件对 SWTRIG 寄存器的访问 (见 150页)。

### 配置及控制寄存器 (CFGCTRL)

基址 0xE000.E000  
偏移量 0xD14  
类型 R/W, 复位 0x0000.0200

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留						STKALIGN	BFHFNMIGN	保留				DIV0	UNALIGNED	保留	MAINPEND	BASETHR
类型	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	R/W	R/W	RO	R/W	R/W	
复位	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:10	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
9	STKALIGN	R/W	1	异常入口的堆栈对齐  值 描述 0 堆栈按 4 字节对齐。 1 堆栈按 8 字节对齐。  在异常的入口处, 处理器利用 PSR 寄存器的第 9 位保存栈对齐状态。当从异常中返回时, 处理器根据出栈后的此标志位恢复栈对齐状态。
8	BFHFNMIGN	R/W	0	在 NMI 及故障中忽略总线故障  通过此标志位可以使得优先级为 -1 和 -2 的异常处理程序忽略由装载指令和存贮指令产生的数据总线故障。该位的设置适用于 NMI、硬故障以及经 FAULTMASK 寄存器升级后的处理程序。  值 描述 0 由装载指令及存贮指令产生的数据总线故障都会导致错误锁定。 1 优先级为 -1 和 -2 的处理程序均忽略装载指令及存贮指令产生的数据总线故障。  只有当处理程序及其数据均位于绝对安全的存储器中时, 才允许将此标志位置位。该位通常用于检测系统设备及总线桥有无控制通道隐患, 并予以修复。
7:5	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
4	DIV0	R/W	0	<p>被零除的额外处理</p> <p>若处理器执行 SDIV 或 UDIV 指令时遭遇除数为零的状况，本标志位将决定是否产生故障或暂停运行。</p> <p>值 描述</p> <p>0 当遇到被零除的状况时，不进行额外处理。被零除返回的商为 0。</p> <p>1 当遇到被零除的状况时，进行额外处理。</p>
3	UNALIGNED	R/W	0	<p>未对齐访问的额外处理</p> <p>值 描述</p> <p>0 当遇到未对齐的半字访问或字访问时，不进行额外处理。</p> <p>1 当遇到未对齐的半字访问或字访问时，进行额外处理。未对齐的访问会产生一个用法错误。</p> <p>未对齐的 LDM、STM、LDRD 和 STRD 指令始终产生故障，不论 UNALIGNED 是否置位。</p>
2	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
1	MAINPEND	R/W	0	<p>允许主中断触发</p> <p>值 描述</p> <p>0 禁止非特权级软件访问 SWTRIG 寄存器。</p> <p>1 允许非特权级软件访问 SWTRIG 寄存器 ( 请参考 150 页 ) 。</p>
0	BASETHR	R/W	0	<p>线程状态控制</p> <p>值 描述</p> <p>0 仅当异常激活后才允许处理器进入线程模式。</p> <p>1 取决于对 EXC_RETURN 值的控制，处理器随时都可进入线程模式 ( 详情请参考“异常返回”在 107 页 ) 。</p>

## 寄存器 72: 系统处理程序优先级寄存器 1 ( SYSPRI1 ) , 偏移量 0xD18

注意: 本寄存器只能在特权模式下访问。

SYSPRI1 寄存器用于配置用法故障、总线故障、存储器管理故障的异常处理程序的优先级, 取值范围为 0~7。本寄存器可以按字节访问。

### 系统处理程序优先级寄存器 1 (SYSPRI1)

基址 0xE000.E000  
偏移量 0xD18  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								USAGE			保留				
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BUS			保留					MEM			保留				
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:24	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
23:21	USAGE	R/W	0x0	用法故障优先级 该域用于配置用法故障的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
20:16	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:13	BUS	R/W	0x0	总线故障优先级 该域用于配置总线故障的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
12:8	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:5	MEM	R/W	0x0	存储器管理故障优先级 本域用于配置存储器管理故障的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
4:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 73: 系统处理程序优先级寄存器 2 (SYSPRI2), 偏移量 0xD1C

注意: 本寄存器只能在特权模式下访问。

SYSPRI2 寄存器用于配置 SVCcall 处理程序的优先级, 取值范围为 0 到 7。本寄存器可以按字节访问。

### 系统处理程序优先级寄存器 2 (SYSPRI2)

基址 0xE000.E000

偏移量 0xD1C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVC			保留												
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	SVC	R/W	0x0	SVCcall 优先级 该域用于配置 SVCcall 的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
28:0	保留	RO	0x000.0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 74: 系统处理程序优先级寄存器 3 (SYSPRI3), 偏移量 0xD20

注意: 本寄存器只能在特权模式下访问。

SYSPRI3 寄存器用于配置 SysTick 异常处理程序、PendSV 处理程序的优先级, 取值范围为 0~7。本寄存器可以按字节访问。

### 系统处理程序优先级寄存器 3 (SYSPRI3)

基址 0xE000.E000  
偏移量 0xD20  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TICK			保留					PENDSV			保留				
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DEBUG			保留				
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	TICK	R/W	0x0	SysTick异常优先级 该域用于配置 SysTick 异常的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
28:24	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
23:21	PENDSV	R/W	0x0	PendSV 优先级 本位域用于配置 PendSV 的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
20:8	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:5	DEBUG	R/W	0x0	调试优先级 该域用于配置调试的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
4:0	保留	RO	0x0.0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 75: 系统处理程序控制及状态寄存器 ( SYSHNDCTRL ), 偏移量 0xD24

注意: 本寄存器只能在特权模式下访问。

SYSHNDCTRL 寄存器用于启用系统处理程序, 并且能够指示用法故障、总线故障、存储器管理故障、SVC 异常的挂起状态, 以及系统处理程序的激活状态。

假如禁用了某个系统处理程序, 又产生了相应故障, 那么处理器会按照硬故障对其进行处理。

通过修改本寄存器可以改变系统异常的挂起状态或激活状态。操作系统内核可以通过写激活标志位来进行上下文切换, 以这种方式更改当前的异常类型。

小心 – 软件在更改本寄存器中某个激活标志位时, 如果并未对已入栈内容进行正确的修正, 则可能导致处理器产生故障异常。应确保写本寄存器的软件保存当前激活状态, 并在之后复原。

启用系统处理程序后, 假如必须修改本寄存器中的某个标志位, 则必须通过读-修改-写流程予以操作, 确保只修改所需的位。

### 系统处理程序控制及状态寄存器 (SYSHNDCTRL)

基址 0xE000.E000  
偏移量 0xD24  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留													USAGE	BUS	MEM
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SVC	BUSP	MEMP	USAGEP	TICK	PNDSV	保留	MON	SVCA	保留			USGA	保留	BUSA	MEMA
类型	R/W	R/W	R/W	R/W	R/W	R/W	RO	R/W	R/W	RO	RO	RO	R/W	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:19	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
18	USAGE	R/W	0	用法故障启用  值 描述 0 禁用用法故障异常。 1 启用用法故障异常。
17	BUS	R/W	0	总线故障启用  值 描述 0 禁用总线故障异常。 1 启用总线故障异常。
16	MEM	R/W	0	存储器管理故障启用  值 描述 0 禁用存储器管理故障异常。 1 启用存储器管理故障异常。

位/域	名称	类型	复位	描述
15	SVC	R/W	0	<p>SVC 调用挂起</p> <p>值 描述</p> <p>0 当前无挂起的 SVC 调用异常。</p> <p>1 当前有挂起的 SVC 调用异常。</p> <p>通过该位可修改 SVC 调用异常的挂起状态。</p>
14	BUSP	R/W	0	<p>总线故障挂起</p> <p>值 描述</p> <p>0 当前无正在挂起的总线故障异常。</p> <p>1 当前有正在挂起的总线故障异常。</p> <p>通过该位可修改总线故障异常的挂起状态。</p>
13	MEMP	R/W	0	<p>存储器管理故障挂起</p> <p>值 描述</p> <p>0 当前无正在挂起的存储器管理故障异常。</p> <p>1 当前有正在挂起的存储器管理故障异常。</p> <p>通过该位可修改存储器管理故障异常的挂起状态。</p>
12	USAGEP	R/W	0	<p>用法故障挂起</p> <p>值 描述</p> <p>0 当前无正在挂起的用法故障异常。</p> <p>1 当前有正在挂起的用法故障异常。</p> <p>通过该位可修改用法故障异常的挂起状态。</p>
11	TICK	R/W	0	<p>SysTick 异常激活取值</p> <p>值 描述</p> <p>0 SysTick 异常未激活。</p> <p>1 SysTick 异常已激活。</p> <p>通过本标志位可修改 SysTick 异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“Caution 重要提示”部分内容。</p>
10	PNDSP	R/W	0	<p>PendSV 异常激活</p> <p>值 描述</p> <p>0 PendSV 异常未激活。</p> <p>1 PendSV 异常已激活。</p> <p>通过本标志位可修改 PendSV 异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>
9	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>



位/域	名称	类型	复位	描述
8	MON	R/W	0	<p>调试监视器激活</p> <p>值 描述</p> <p>0 调试监视器未激活。</p> <p>1 调试监视器已激活。</p>
7	SVCA	R/W	0	<p>SVC 调用激活</p> <p>值 描述</p> <p>0 SVC 调用未激活。</p> <p>1 SVC调用已激活。</p> <p>通过本标志位可修改SVC调用异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>
6:4	保留	RO	0x0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
3	USGA	R/W	0	<p>用法故障激活</p> <p>值 描述</p> <p>0 用法故障未激活。</p> <p>1 用法故障已激活。</p> <p>通过本标志位可修改用法故障异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>
2	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
1	BUSA	R/W	0	<p>总线故障激活</p> <p>值 描述</p> <p>0 总线故障未激活。</p> <p>1 总线故障已激活。</p> <p>通过本标志位可修改总线故障异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>
0	MEMA	R/W	0	<p>存储器管理故障激活</p> <p>值 描述</p> <p>0 存储器管理故障未激活。</p> <p>1 存储器管理故障已激活。</p> <p>通过本标志位可修改存储器管理故障异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>

**寄存器 76: 可配置故障状态寄存器 ( FAULTSTAT ) , 偏移量 0xD28**

注意: 本寄存器只能在特权模式下访问。

FAULTSTAT 寄存器能够指示产生存储器管理故障、总线故障、用法故障的原因。这些功能可以进一步划分为子寄存器, 如下所示:

- 用法故障状态 (UFAULTSTAT), 即位 31:16
- 总线故障状态 (BFAULTSTAT), 即位 15:8
- 存储器管理故障状态 (MFAULTSTAT), 即位 7:0

FAULTSTAT 寄存器可按字节访问。FAULTSTAT 寄存器及其子寄存器可按如下方式访问:

- 对偏移量 0xD28 按字访问, 即可访问整个的 FAULTSTAT 寄存器;
- 对偏移量 0xD28 按字节访问, 即可访问 MFAULTSTAT 子寄存器;
- 对偏移量 0xD28 按半字访问, 即可访问 MFAULTSTAT 和 BFAULTSTAT 子寄存器;
- 对偏移量 0xD29 按字节访问, 即可访问 BFAULTSTAT 子寄存器;
- 对偏移量 0xD2A 按半字访问, 即可访问 UFAULTSTAT 子寄存器。

本寄存器中的位写 1 清零。

在故障处理程序中, 发生故障的实际地址可按如下方法确定:

1. 读取并保存存储器管理故障地址 (MMADDR) 或总线故障地址 (FAULTADDR) 的值;
2. 读取 MFAULTSTAT 子寄存器的 MMARV 位, 或 BFAULTSTAT 子寄存器的 BFARV 标志位, 以确定 MMADDR 或 FAULTADDR 中的内容是否有效。

软件必须按照此序列操作, 因为另一个优先级更高的异常有可能会更改 MMADDR 或 FAULTADDR 的值。例如, 如果某个优先级更高的处理程序抢占了当前的处理程序, 那么另一个故障可能会改变 MMADDR 或 FAULTADDR 的内容。

**可配置故障状态寄存器 (FAULTSTAT)**

基址 0xE000.E000  
偏移量 0xD28  
类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留						DIV0	UNALIGN	保留				NOCP	INVPC	INVSTAT	UNDEF
类型	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFARV	保留	BLSPERR	BSTKE	BUSTKE	IMPRE	PRECISE	IBUS	MMARV	保留	MLSPERR	MSTKE	MUSTKE	保留	DERR	IERR
类型	R/W1C	RO	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	RO	R/W1C	R/W1C	R/W1C	RO	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:26	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
25	DIV0	R/W1C	0	<p>被零除用法故障</p> <p>值 描述</p> <p>0 未发生被零除故障，或并未启用被零除的额外处理。</p> <p>1 处理器执行 SDIV 或 UDIV 指令时，除数为 0。</p> <p>若该位置位，原本用于异常返回的已入栈 PC 值将指向执行除零的那条指令。</p> <p>如果需要对被零除进行额外处理，应将配置和控制 (CFGCTRL) 寄存器 (请参考 162页) 中的 DIV0 位置位。</p> <p>该位写 1 清零。</p>
24	UNALIGN	R/W1C	0	<p>未对齐访问用法故障</p> <p>值 描述</p> <p>0 未发生未对齐访问故障，或并未启用未对齐访问的额外处理。</p> <p>1 处理器企图执行未对齐访问操作。</p> <p>未对齐的 LDM、STM、LDRD 和 STRD 指令始终产生故障，不论本标志位配置如何。</p> <p>如果需要未对齐访问进行额外处理，应将 CFGCTRL 寄存器 (请参考 162页) 中的 UNALIGNED 位置位。</p> <p>该位写 1 清零。</p>
23:20	保留	RO	0x00	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
19	NOCP	R/W1C	0	<p>无协处理器用法故障</p> <p>值 描述</p> <p>0 未发生企图访问协处理器的用法故障。</p> <p>1 处理器企图访问协处理器。</p> <p>该位写 1 清零。</p>
18	INVPC	R/W1C	0	<p>装载无效 PC 值用法故障</p> <p>值 描述</p> <p>0 未发生企图装载无效 PC 值的用法故障。</p> <p>1 处理器企图把非法的 EXC_RETURN 装载到 PC。这可能是由于环境无效，或由于 EXC_RETURN 值无效。</p> <p>若该位置位，原本用于异常返回的已入栈 PC 值将指向企图装载无效 PC 值的那条指令。</p> <p>该位写 1 清零。</p>

位/域	名称	类型	复位	描述
17	INVSTAT	R/W1C	0	<p>无效状态用法故障</p> <p>值 描述</p> <p>0 未发生无效状态的用法故障。</p> <p>1 处理器执行了某条指令，该指令企图非法使用 EPSR 寄存器。</p> <p>若该位置位，原本用于异常返回的已入栈 PC 值将指向企图非法使用执行程序状态寄存器 (EPSR) 寄存器的那条指令。</p> <p>如果某条未定义的指令使用了 EPSR 寄存器，该位不会置位。</p> <p>该位写 1 清零。</p>
16	UNDEF	R/W1C	0	<p>未定义指令用法故障</p> <p>值 描述</p> <p>0 未发生未定义指令的用法故障。</p> <p>1 处理器企图执行一条未定义指令。</p> <p>若该位置位，原本用于异常返回的已入栈 PC 值将指向那条未定义指令。</p> <p>所谓未定义指令，就是指处理器内核无法对其正确解码的指令。</p> <p>该位写 1 清零。</p>
15	BFARV	R/W1C	0	<p>总线故障地址寄存器有效</p> <p>值 描述</p> <p>0 总线故障地址 (FAULTADDR) 寄存器中未包含有效的故障地址。</p> <p>1 FAULTADDR 寄存器中包含有效的故障地址。</p> <p>当出现总线故障后，若故障地址已知，则该位将置位。其它故障有可能清零本标志位，例如之后若发生存储器管理故障，即会清零该位。</p> <p>若产生总线故障，并且由于优先级而升级为硬故障，那么硬故障处理程序必须清零本标志位。当程序返回到已入栈的总线故障处理程序时，该操作防止 FAULTADDR 寄存器的内容被覆盖。</p> <p>该位写 1 清零。</p>
14	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
13	BLSPEER	R/W1C	0	<p>浮点怠惰状态保持的总线故障</p> <p>值 描述</p> <p>0 在浮点怠惰状态保持时没有发生总线故障。</p> <p>1 在浮点怠惰状态保持时发生了总线故障。</p> <p>该位写 1 清零。</p>

位/域	名称	类型	复位	描述
12	BSTKE	R/W1C	0	<p>入栈总线故障</p> <p>值 描述</p> <p>0 在异常入口入栈操作时为发生总线故障。</p> <p>1 在异常入口处进行入栈操作时产生一个或多个总线故障。</p> <p>若该位置位，SP 寄存器的值仍然会正常变更，但是栈内上下文的内容可能包含错误内容。此类总线故障并不向 FAULTADDR 寄存器写入故障地址。</p> <p>该位写 1 清零。</p>
11	BUSTKE	R/W1C	0	<p>出栈总线故障</p> <p>值 描述</p> <p>0 异常返回出栈操作时没有发生总线故障。</p> <p>1 异常返回时的出栈操作时产生一个或多个总线故障。</p> <p>此故障将使得处理器再次返回到异常处理程序。因此，若此标志位置位，原始的返回栈并不会被破坏。前一次返回失败，因此 SP 寄存器的值不变，也不会进行入栈操作。此类总线故障并不向 FAULTADDR 寄存器写入故障地址。</p> <p>该位写 1 清零。</p>
10	IMPRE	R/W1C	0	<p>不精确数据总线错误</p> <p>值 描述</p> <p>0 未产生不精确数据总线错误。</p> <p>1 产生了数据总线错误，但栈中的返回地址未必指向导致错误产生的那条指令。</p> <p>若该位置位，不会向 FAULTADDR 寄存器写入故障地址。</p> <p>此故障是异步故障。因此，当检测到故障发生时，当前进程的优先级高于总线故障优先级，总线故障将保持挂起状态，只有等到处理器完成所有高优先级的进程后，才会转入激活状态。如果某个精确故障在处理器进入处理程序之前发生，原因是不精确的总线故障，那么处理程序可以查询到 IMPRE 位与某个精确故障状态位同时置位。</p> <p>该位写 1 清零。</p>
9	PRECISE	R/W1C	0	<p>精确数据总线错误</p> <p>值 描述</p> <p>0 未产生精确数据总线错误。</p> <p>1 产生了数据总线错误，并且原本用于异常返回的已入栈 PC 值指向导致错误发生的那条指令。</p> <p>若该位置位，将会向 FAULTADDR 寄存器写入故障地址。</p> <p>该位写 1 清零。</p>

位/域	名称	类型	复位	描述
8	IBUS	R/W1C	0	<p>指令总线错误</p> <p>值 描述</p> <p>0 未发生指令总线错误。</p> <p>1 发生了指令总线错误。</p> <p>处理器在预取指时就已经检测指令总线是否出错，不过只有在试图执行错误指令时才会置位本标志位。</p> <p>若该位置位，不会向 FAULTADDR 寄存器写入故障地址。</p> <p>该位写 1 清零。</p>
7	MMARV	R/W1C	0	<p>存储器管理故障地址寄存器有效</p> <p>值 描述</p> <p>0 存储器管理故障地址 (MMADDR) 寄存器中未包含有效的故障地址。</p> <p>1 MMADDR 寄存器中包含有效的故障地址。</p> <p>若产生存储器管理故障，并且由于优先级而升级为硬故障，那么硬故障处理程序必须清零该位。当程序返回到已入栈的活动存储器管理故障处理程序时，该操作防止 MMADDR 寄存器的内容被覆盖。</p> <p>该位写 1 清零。</p>
6	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
5	MLSPERR	R/W1C	0	<p>浮点怠惰状态保持的存储器管理故障</p> <p>值 描述</p> <p>0 浮点怠惰状态保持时未产生寄存器管理故障。</p> <p>1 浮点怠惰状态保持时未产生寄存器管理故障。</p> <p>该位写 1 清零。</p>
4	MSTKE	R/W1C	0	<p>入栈访问违规</p> <p>值 描述</p> <p>0 当在异常入口处进行入栈操作时为发生存储器管理故障。</p> <p>1 当在异常入口处进行入栈操作时产生一个或多个访问违规。</p> <p>若该位置位，SP 寄存器的值仍然会正常变更，但是栈内上下文的内容可能包含错误内容。此类故障并不向 MMADDR 寄存器写入故障地址。</p> <p>该位写 1 清零。</p>
3	MUSTKE	R/W1C	0	<p>出栈操作时的访问违规</p> <p>值 描述</p> <p>0 异常返回出栈操作时没有发生存储器管理故障。</p> <p>1 异常返回出栈操作时产生一个或多个访问违规。</p> <p>此故障将使得处理器再次返回到异常处理程序。因此，若此标志位置位，原始的返回栈并不会被破坏。前一次返回失败，因此 SP 寄存器的值不变，也不会进行入栈操作。此类故障并不向 MMADDR 寄存器写入故障地址。</p> <p>该位写 1 清零。</p>

位/域	名称	类型	复位	描述
2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
1	DERR	R/W1C	0	<p>数据访问违规</p> <p>值 描述</p> <p>0 未发生数据访问违规。</p> <p>1 处理器试图对某个地址进行装载或存贮操作，但该地址并不允许执行该操作。</p> <p>若该位置位，原本用于异常返回的已入栈PC值将指向出错的那条指令，并且企图访问的地址将写入 MMADDR 寄存器中。</p> <p>该位写 1 清零。</p>
0	IERR	R/W1C	0	<p>指令访问违规</p> <p>值 描述</p> <p>0 未发生指令访问违规。</p> <p>1 处理器试图从某个地址取指令，但该地址并不允许执行指令。</p> <p>当对任何 XN 存储器区访问时，均会产生此故障，即使 MPU 被禁用、甚至未集成 MPU 也是如此。</p> <p>若该位置位，原本用于异常返回的已入栈PC值将指向出错的那条指令，而且企图访问的地址不会写入 MMADDR 寄存器中。</p> <p>该位写 1 清零。</p>

## 寄存器 77: 硬故障状态寄存器 (HFAULTSTAT), 偏移量 0xD2C

注意: 本寄存器只能在特权模式下访问。

HFAULTSTAT 寄存器提供触发硬故障处理程序的事件的相关信息。

本寄存器中的位写 1 清零。

### 硬故障状态寄存器 (HFAULTSTAT)

基址 0xE000.E000  
偏移量 0xD2C  
类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	DBG	FORCED	保留														
类型	R/W1C	R/W1C	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留														VECT	保留	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31	DBG	R/W1C	0	调试事件 此标志位保留用于调试。平时此标志位必须写入 0, 否则可能产生无法预料的结果。
30	FORCED	R/W1C	0	强制硬故障  值 描述 0 未发生强制硬故障。 1 发生了强制硬故障。其原因可能是某个优先级可配置的故障无法得到及时处理 ( 可能是因为优先级较低或被禁用 ) 导致其升级。  若该位置位, 则硬故障处理程序必须读取其它故障状态寄存器, 找出故障原因。 该位写 1 清零。
29:2	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
1	VECT	R/W1C	0	向量表读取故障  值 描述 0 在读取向量表时未发生总线故障。 1 在读取向量表时发生总线故障。  此错误始终由硬故障处理程序处理。 若该位置位, 原本用于异常返回的已入栈 PC 值将指向被异常抢占使用的那条指令。 该位写 1 清零。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。



**寄存器 78: 存储器管理故障地址寄存器 ( MMADDR ) , 偏移量 0xD34**

注意: 本寄存器只能在特权模式下访问。

MMADDR 寄存器中包含产生存储器管理故障的地址。当发生未对齐访问故障时, MMADDR 寄存器中包含的是产生故障的实际地址。对于其它类型故障, 由于单条读写指令还可能进一步分割为多个对齐操作, 因此故障地址可能是被请求访问范围内的任一地址。存储器管理故障状态(MFAULTSTAT)中的位能够指示出故障原因以及 MMADDR 寄存器中的值是否有效(请参考 170页)。

**存储器管理故障地址寄存器 (MMADDR)**

基址 0xE000.E000

偏移量 0xD34

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:0	ADDR	R/W	-	故障地址 若MFAULTSTAT 寄存器的 MMARV 位置位, 那么该域包含产生存储器管理故障的地址。

**寄存器 79: 总线故障地址寄存器 ( FAULTADDR ) , 偏移量 0xD38**

注意: 本寄存器只能在特权模式下访问。

FAULTADDR 寄存器包含产生总线故障的地址。当发生未对齐访问故障时, FAULTADDR 寄存器中包含的是错误指令所请求的地址, 并非产生故障的实际地址。总线故障状态 (BFAULTSTAT) 寄存器中的状态位能够指示出故障原因以及 FAULTADDR 寄存器中的值是否有效 ( 请参考 170页 )。

**总线故障地址寄存器 (FAULTADDR)**

基址 0xE000.E000

偏移量 0xD38

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:0	ADDR	R/W	-	故障地址 若 BFAULTSTAT 寄存器中的 FAULTADDRV 位置位, 那么该域包含产生总线故障的地址。

**3.6 存储器保护单元 ( MPU ) 寄存器描述**

本节按照地址偏移量由小到大的顺序依次详细介绍存储器保护单元 ( MPU ) 的各寄存器。

MPU 寄存器只能在特权模式下访问。

## 寄存器 80: MPU 类型寄存器 (MPUTYPE), 偏移量 0xD90

注意: 本寄存器只能在特权模式下访问。

MPUTYPE 寄存器用于指示本芯片是否包含 MPU, 若包含 MPU 还指示其支持多少个存储器区。

### MPU 类型寄存器 (MPUTYPE)

基址 0xE000.E000  
偏移量 0xD90  
类型 RO, 复位 0x0000.0800

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								IREGION							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DREGION								保留							SEPARATE
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:24	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
23:16	IREGION	RO	0x00	I 区数目 此域可显示出芯片所支持的 MPU 指令区数目。此域位域始终包含 0x00。MPU 存储器映射是统一的, 可以用 DREGION 域设置。
15:8	DREGION	RO	0x08	D 区数目  值 描述 0x08 表示总共支持 8 个 MPU 数据区。
7:1	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	SEPARATE	RO	0	MPU 分立还是统一  值 描述 0 表示 MPU 是统一管理的。

## 寄存器 81: MPU 控制寄存器 (MPUCTRL), 偏移量 0xD94

注意: 本寄存器只能在特权模式下访问。

MPUCTRL 寄存器能够启用 MPU、启用默认存储器映射背景区, 此外还能在硬故障、不可屏蔽中断 (NMI) 以及故障屏蔽寄存器 (FAULTMASK) 升级后的处理程序中使用 MPU。

当 ENABLE 和 PRIVDEFEN 位同时置位:

- 对于特权级访问, 默认的存储器映射参见“存储模型”在 91 页。若某存储器区并未启用, 则特权级软件的任何访问特性将与默认的存储器映射相同。
- 若某存储器区并未启用, 则非特权级软件的任何访问都将产生存储器管理故障。

不论 ENABLE 位的值是多少, Execute Never (XN, 不执行) 以及 Strongly Ordered (严格排序) 规则总适用于系统控制空间。

若 ENABLE 置位, 一般来说存储器映射中必须至少启用一个存储器区, 这样才能保障系统工作正常, 除非 PRIVDEFEN 位置位。若没有启用的存储器区, 如果 PRIVDEFEN 也置位, 那么只有特权级软件才能正常操作。

若 ENABLE 清零, 系统将采用默认的存储器映射, 其存储器属性与未采用 MPU 之前的属性相同 (请参考表 2-5 在 94 页)。此时无论特权级软件还是非特权级软件均按照默认的存储器映射进行访问。

当启用 MPU 后, 将始终允许对系统控制空间以及向量表的访问。对其它区域的访问权限则取决于区属性设置以及 PRIVDEFEN 是否置位。

除非 HFNMIENA 置位, 否则当处理器执行优先级为 -1 或 -2 的异常处理程序时不能启用 MPU。当处理硬故障或者 NMI 异常或者当 FAULTMASK 启用时, 这些优先级才可用。若 HFNMIENA 位置位, 则在这两个优先级下工作时, 也将启用 MPU。

### MPU 控制寄存器 (MPUCTRL)

基址 0xE000.E000  
偏移量 0xD94  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													PRIVDEFEN	HFNMIENA	ENABLE	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
2	PRIVDEFEN	R/W	0	<p>MPU 默认区</p> <p>此标志位用于选择是否允许特权级软件按照默认的存储器映射进行访问。</p> <p>值 描述</p> <p>0 若 MPU 已经启用，此标志位为 0 代表禁用默认存储器映射。对某个地址进行访问时，若该地址不处于任一启用的区中，则会产生故障。</p> <p>1 若 MPU 已经启用，此标志位为 1 代表允许特权级软件按照默认存储器映射作为背景区进行访问。</p> <p>若此标志位置位，背景区将作为编号为 -1 的存储器区。任何已定义、已启用区的优先级都高于背景区。</p> <p>若 MPU 未启用，处理器将忽略本标志位。</p>
1	HFNMIENA	R/W	0	<p>故障期间启用 MPU</p> <p>该位控制 MPU 在硬故障、NMI 以及 FAULTMASK 处理程序期间的操作。</p> <p>值 描述</p> <p>0 不管 ENABLE 位的状态如何，在硬故障、NMI 以及 FAULTMASK 处理程序期间都将自动禁用 MPU。</p> <p>1 在硬故障、NMI 以及 FAULTMASK 处理程序期间，仍然启用 MPU。</p> <p>若此位置位，而 MPU 又没有启用，那么其结果将无法预料。</p>
0	ENABLE	R/W	0	<p>MPU 启用</p> <p>值 描述</p> <p>0 禁用 MPU。</p> <p>1 启用 MPU。</p> <p>若 HFNMIENA 位置位，而 MPU 又没有启用，那么其结果将无法预料。</p>

## 寄存器 82: MPU 区编号寄存器 (MPUNUMBER) , 偏移量 0xD98

注意: 本寄存器只能在特权模式下访问。

MPUNUMBER 寄存器用于选择 MPU 区基地址 (MPUBASE) 和 MPU 区属性及大小 (MPUATTR) 寄存器所参考的存储器区编号。一般在访问 MPUBASE 或 MPUATTR 寄存器之前, 应将需要操作的存储器区编号写入本寄存器。但是, 该区编号可以通过写入 MPUBASE 寄存器的 VALID 位设置 (请参考 183页) 来改变。该写入操作会更新 REGION 域的值。

### MPU 区编号寄存器 (MPUNUMBER)

基址 0xE000.E000  
偏移量 0xD98  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													NUMBER			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2:0	NUMBER	R/W	0x0	可访问的 MPU 区 该域用于指定 MPUBASE 和 MPUATTR 寄存器参考的 MPU 区。MPU 支持 8 个存储器区。

寄存器 83: MPU 区基地址寄存器 (MPUBASE), 偏移量 0xD9C

寄存器 84: MPU 区基地址别名寄存器 1 (MPUBASE1), 偏移量 0xDA4

寄存器 85: MPU 区基地址别名寄存器 2 (MPUBASE2), 偏移量 0xDAC

寄存器 86: MPU 区基地址别名寄存器 3 (MPUBASE3), 偏移量 0xDB4

注意: 本寄存器只能在特权模式下访问。

MPUBASE 寄存器用于定义由 MPU 区编号 (MPUNUMBER) 寄存器所选择的 MPU 区的基地址, 此外也能够更新 MPUNUMBER 寄存器的值。如果想修改当前的存储器区编号并更新 MPUNUMBER 寄存器内容, VALID 位置位 并对 MPUBASE 寄存器执行写入操作。

ADDR 域是指 MPUBASE 寄存器的 31:N 位。(N-1):5 位保留。由 MPU 区属性及大小 (MPUATTR) 寄存器中 SIZE 域指定的区大小决定了 N 的值:

$$N = \text{Log}_2(\text{区大小, 以字节为单位})$$

若在 MPUATTR 寄存器中区大小为 4 GB, 那就没有可用的 ADDR 域。此时, 该存储器区将占据处理器的全部寻址空间, 其基地址为 0x0000.0000。

基地址必须按照存储器区的大小对齐。例如, 定义某个区大小为 64 kB, 则其基地址必须按照 64 kB 的倍数对齐, 例如可以是 0x0001.0000 或 0x0002.0000。

#### MPU 区基地址寄存器 (MPUBASE)

基址 0xE000.E000  
偏移量 0xD9C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	ADDR																
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	ADDR												VALID	保留	REGION		
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	WO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:5	ADDR	R/W	0x0000.000	基地址掩码 该域中的 31:N 包含存储器区的基地址。N 的值取决于区大小, 如上所述。其余的 (N-1):5 位保留。 软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
4	VALID	WO	0	区编号有效  值 描述 0 MPUNUMBER 寄存器不变, 处理器按照 MPUNUMBER 寄存器的设置更新存储器区的基地址, 忽略 REGION 域的设置。 1 MPUNUMBER 寄存器将更新为 REGION 域的值, 并依此 REGION 域更新相应存储器区的基地址。  该位读取始终返回 0。
3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
2:0	REGION	R/W	0x0	区编号 写入本域时，其包含将要写入 MPUNUMBER 寄存器的值。读取本域时，将返回 MPUNUMBER 寄存器中的当前区编号。。



寄存器 87: MPU 区属性和大小寄存器 (MPUATTR), 偏移量 0xDA0

寄存器 88: MPU 区属性和大小别名寄存器 1 (MPUATTR1), 偏移量 0xDA8

寄存器 89: MPU 区属性和大小别名寄存器 2 (MPUATTR2), 偏移量 0xDB0

寄存器 90: MPU 区属性和大小别名寄存器 3 (MPUATTR3), 偏移量 0xDB8

注意: 本寄存器只能在特权模式下访问。

MPUATTR 寄存器用于定义由 MPU 区编号 (MPUNUMBER) 寄存器所指定的 MPU 区的大小以及存储器属性, 并且能够启用该 MPU 区及其子区。

MPUATTR 寄存器可按字或半字对齐访问, 其高半字保存区属性, 低半字保存区大小、区启用标志以及各子区的启用位。

MPU 访问权限属性位: XN、AP、TEX、S、C 以及 B, 用于控制对相应存储器区的访问。假如试图访问某个区域, 而该区域并未开放相关权限, 那么 MPU 将会产生一个访问权限故障。

SIZE 域用于定义由 MPUNUMBER 寄存器所指定的存储器区的大小, 如下所示:

$$(\text{区大小, 以字节为单位}) = 2^{(\text{SIZE}+1)}$$

存储器区最小允许为 32 字节, 即 SIZE 域最小允许为 4。表 3-10 在 185 页列出了 SIZE 域与区大小相关的一些示例值, 同时列出了 MPU 区基地址 (MPUBASE) 中对应的 N 值。

表 3-10. SIZE 域数值示例

SIZE 域编码	区大小	N 的值 <sup>a</sup>	注:
00100b (0x4)	32 B	5	允许的最小区大小
01001b (0x9)	1 KB	10	-
10011b (0x13)	1 MB	20	-
11101b (0x1D)	1 GB	30	-
11111b (0x1F)	4 GB	MPUBASE 寄存器中没有有效的 ADDR 域; 该区已经占据全部寻址空间。	最大可能的大小

a. 表示 MPUBASE 寄存器 (请参考 183 页) 中的 N 参数。

#### MPU 区属性和大小寄存器 (MPUATTR)

基址 0xE000.E000  
偏移量 0xDA0  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			XN	保留	AP			保留			TEX	S	C	B	
类型	RO	RO	RO	R/W	RO	R/W	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRD								保留			SIZE				ENABLE
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
28	XN	R/W	0	禁止指令访问  值 描述 0 启用指令存取操作。 1 禁止指令存取操作。
27	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
26:24	AP	R/W	0	访问权限 该域的更多使用信息请参考 表 3-5 在 124页。
23:22	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
21:19	TEX	R/W	0x0	类型扩展掩码 该域的更多使用信息请参考 表 3-3 在 124页。
18	S	R/W	0	可共享 该位的更多使用信息请参考 表 3-3 在 124页。
17	C	R/W	0	可高速缓存 该位的更多使用信息请参考 表 3-3 在 124页。
16	B	R/W	0	可缓冲 该位的更多使用信息请参考 表 3-3 在 124页。
15:8	SRD	R/W	0x00	子区禁用位  值 描述 0 启用相应子区。 1 禁用相应子区。  存储器区小于等于 128 字节时，将无法再分割为子区。此时写入存储器区属性时，应将 SRD 域配置为 0x00。请参阅“存储子区”一节 在 123页以了解更多信息。
7:6	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
5:1	SIZE	R/W	0x0	区大小掩码 SIZE 域用于定义由 MPUNUMBER 寄存器指定的 MPU 存储器区的大小。请参阅 表 3-10 在 185页 以了解更多信息。
0	ENABLE	R/W	0	区启用  值 描述 0 禁用此存储器区。 1 启用此存储器区。

### 3.7 浮点单元 (FPU) 寄存器描述

本节按照地址偏移量由小到大的顺序，依次罗列并描述浮点单元 (FPU) 的各寄存器。

## 寄存器 91: 协处理器访问控制 ( CPAC ) , 偏移量 0xD88

CPAC 寄存器指定协处理器的访问权限。

## 协处理器访问控制 (CPAC)

基址 0xE000.E000

偏移量 0xD88

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								CP11		CP10		保留			
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:24	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
23:22	CP11	R/W	0x00	CP11 协处理器访问权限  值 描述 0x0 拒绝访问 任何访问尝试都将产生 NOCP 用法故障。 0x1 仅特权级访问 无特权的访问将产生 NOCP 错误。 0x2 保留 任何访问的结果都无法预测。 0x3 完全访问权
21:20	CP10	R/W	0x00	CP10 协处理器访问权限  值 描述 0x0 拒绝访问 任何访问尝试都将产生 NOCP 用法故障。 0x1 仅特权级访问 无特权的访问将产生 NOCP 错误。 0x2 保留 任何访问的结果都无法预测。 0x3 完全访问权
19:0	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 92: 浮点上下文控制 ( FPCC ) , 偏移量 0xF34

FPCC 寄存器置位或返回 FPU 控制数据。

## 浮点上下文控制 (FPCC)

基址 0xE000.E000  
偏移量 0xF34  
类型 R/W, 复位 0xC000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ASPEN	LSPEN	保留													
类型	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留							MONRDY	保留	BFRDY	MMRDY	HFRDY	线程	保留	USER	LSPACT
类型	RO	RO	RO	RO	RO	RO	RO	R/W	RO	R/W	R/W	R/W	R/W	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31	ASPEN	R/W	1	<p>启用自动状态保存</p> <p>置位后, 在执行浮点指令时可使用 CONTROL 寄存器中的 FRACTV 位。在浮点上下文异常进入和退出时, 该功能可自动保存和恢复硬件状态。</p> <hr/> <p><b>重要:</b> 两个位控制何时启用 FPCA: 浮点上下文控制 (FPCC) 寄存器中的 ASPEN 位和辅助控制 (ACTLR) 寄存器中的 DISFPCA 位。</p> <hr/>
30	LSPEN	R/W	1	<p>启用怠惰状态保存</p> <p>置位后, 启用浮点上下文的怠惰状态保存。</p>
29:9	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
8	MONRDY	R/W	0	<p>监控器就绪</p> <p>置位后, 调试监控器被启用; 分配浮点堆栈框时, 优先级允许置位 MON_PEND。</p>
7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
6	BFRDY	R/W	0	<p>总线故障就绪</p> <p>置位后, 总线故障被启用; 分配浮点堆栈框时, 优先级允许将总线故障处理程序设置为挂起状态。</p>
5	MMRDY	R/W	0	<p>存储器管理故障就绪</p> <p>置位后, 存储器管理被启用; 分配浮点堆栈框时, 优先级允许将存储器管理处理程序设置为挂起状态。</p>
4	HFRDY	R/W	0	<p>硬故障就绪</p> <p>置位后, 分配浮点堆栈框时, 优先级允许将硬故障处理程序设置为挂起状态。</p>
3	线程	R/W	0	<p>线程模式</p> <p>置位后, 分配浮点堆栈框时状态为线程模式。</p>
2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

---

位/域	名称	类型	复位	描述
1	USER	R/W	0	用户特权等级 置位后，分配浮点堆栈框时特权等级为用户。
0	LSPACT	R/W	0	怠惰状态保存有效 置位后，怠惰状态保存有效。浮点堆栈框已分配，但将状态保存到堆栈框出现延迟。

## 寄存器 93: 浮点上下文访问 ( FPCA ) , 偏移量 0xF38

FPCA 寄存器包含在异常堆栈框上分配的未被占用的浮点寄存器空间的位置。

## 浮点上下文访问 (FPCA)

基址 0xE000.E000

偏移量 0xF38

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	地址															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	地址													保留		
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0

位/域	名称	类型	复位	描述
31:3	地址	R/W	-	地址 在异常堆栈框上分配的未被占用的浮点寄存器空间的位置。
2:0	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 94: 浮点默认状态控制 ( FPDSC ) , 偏移量 0xF3C

FPDSC 寄存器包含浮点状态控制 (FPSC) 寄存器的默认值。

## 浮点默认状态控制 (FPDSC)

基址 0xE000.E000

偏移量 0xF3C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留					AHP	DN	FZ	RMODE			保留				
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	-	-	-	-	-	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:27	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
26	AHP	R/W	-	AHP 位默认值 该位包含 FPSC 寄存器中 AHP 位的默认值。
25	DN	R/W	-	DN 位默认值 该位包含 FPSC 寄存器中 DN 位的默认值。
24	FZ	R/W	-	FZ 位默认值 该位包含 FPSC 寄存器中 FZ 位的默认值。
23:22	RMODE	R/W	-	RMODE 位默认值 该位包含 FPSC 寄存器中 RMODE 位域的默认值。  值 描述 0x0 向最接近值舍入 (RN) 模式 0x1 向正无穷大舍入 (RP) 模式 0x2 向负无穷大舍入 (RM) 模式 0x3 向 0 舍入 (RZ) 模式
21:0	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

## 4 JTAG 接口

联合测试行动组 (JTAG) 是一项 IEEE 标准，它定义了数字集成电路的测试访问端口和边界扫描结构，并且提供了一个标准化的串行接口来控制相关的测试逻辑。TAP、指令寄存器 (IR) 和数据寄存器 (DR) 可用于测试组装好的印刷电路板的互连性，并获取元件的制造信息。JTAG 端口还可用于访问和控制测试用设计的特性，如 I/O 管脚的观察和控制、扫描测试以及调试。

JTAG 接口由四个管脚组成：TCK、TMS、TDI 和 TDO。数据通过 TDI 串行发送至控制器，然后通过 TDO 从控制器串行输出。该数据的解析取决于 TAP 控制器的当前状态。有关 JTAG 端口和 TAP 控制器操作的详细信息，请参考“IEEE 标准 1149.1-测试访问端口和边界扫描结构”。

该 Stellaris<sup>®</sup> JTAG 控制器与植入 Cortex-M4F 内核的 ARM JTAG 控制器协同工作，可复用这两个 JTAG 控制器的 TDO 输出。ARM JTAG 指令选择 ARM TDO 输出，而 Stellaris JTAG 指令选择 Stellaris TDO 输出。多路复用器由 Stellaris JTAG 控制器控制，该控制器拥有 ARM 的全面编程信息 Stellaris 以及未执行的 JTAG 指令。

该 Stellaris JTAG 模块具有以下特性：

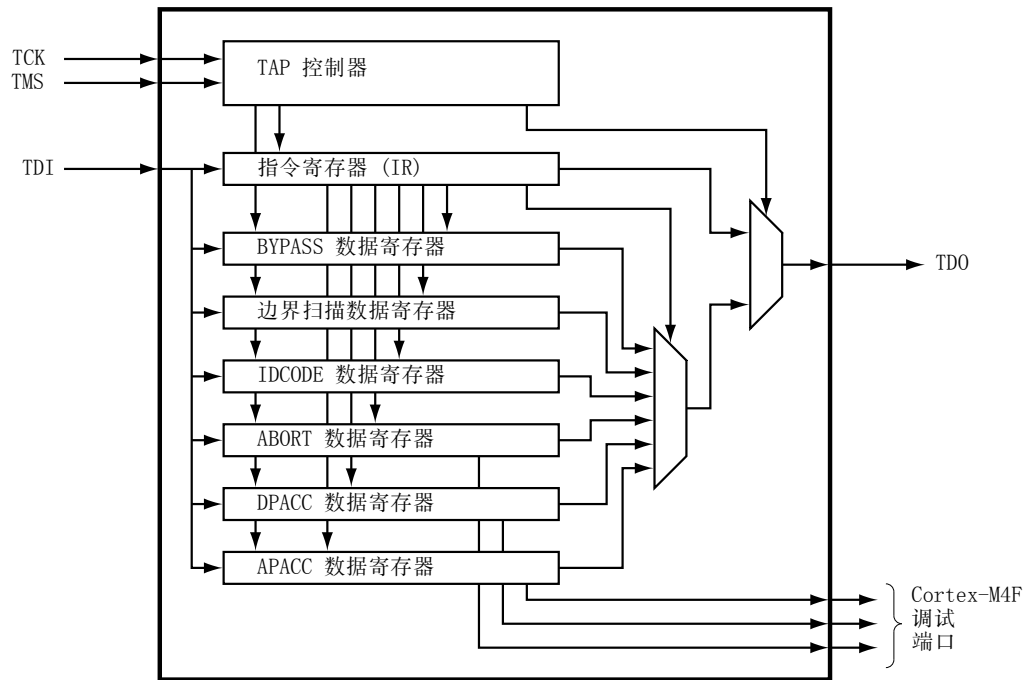
- IEEE 1149.1-1990 兼容的测试访问端口 (TAP) 控制器
- 4 位指令寄存器 (IR) 链，用于存储 JTAG 指令
- IEEE 标准指令：BYPASS、IDCODE、SAMPLE/PRELOAD、EXTEST 和 INTEST
- ARM 附加指令：APACC、DPACC 和 ABORT
- 集成的 ARM 串行线调试 (SWD)
  - 串行线 JTAG 调试端口 (SWJ-DP)
  - Flash 修补和断点 (FPB) 单元，用于实现断点操作
  - 数据观察点和触发 (DWT) 单元，用于执行观察点、触发源和系统性能分析
  - 仪表跟踪宏单元 (ITM)，用于支持 printf 型调试
  - 用于指令追踪捕捉的嵌入式追踪宏单元 (ETM)
  - 跟踪端口接口单元 (TPIU) 用作跟踪端口分析仪的桥接

有关 ARM JTAG 控制器的更多信息，请参考“ARM<sup>®</sup> Debug Interface V5 架构规格”。



## 4.1 结构图

图 4-1. JTAG 模块结构图



## 4.2 信号描述

下表列出了 JTAG/SWD 控制器的外部信号及其功能。JTAG/SWD 控制器信号是某些 GPIO 信号的复用功能，但是请注意，这些管脚的复位状态是 JTAG/SWD 功能。JTAG/SWD 控制器信号处于保护状态，要配置成 GPIO，即需要进行特殊处理，请参阅“提交控制”在 604 页。表中的“复用管脚/赋值”一栏列出了 JTAG/SWD 控制器信号的 GPIO 管脚布局。将 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618 页) 中的 AFSEL 位置位以选择 JTAG/SWD 功能。必须将括号中的数字写入 GPIO 端口控制 (GPIOCTL) 寄存器 (636 页) 中的 PMCn 域，以便将 JTAG/SWD 信号分配给指定的 GPIO 端口管脚。有关如何配置 GPIO 的更多信息，请参阅“通用输入/输出端口 (GPIO)”在 597 页。

表 4-1. JTAG\_SWO\_SWO 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
SWCLK	118	PC0 (1)	I	TTL	JTAG/SWD CLK 信号。
SWDIO	117	PC1 (1)	I/O	TTL	JTAG TMS 及 SWDIO 信号。
SWO	115	PC3 (1)	O	TTL	JTAG TDO 及 SWO 信号。
TCK	118	PC0 (1)	I	TTL	JTAG/SWD CLK 信号。
TDI	116	PC2 (1)	I	TTL	JTAG TDI 信号。
TDO	115	PC3 (1)	O	TTL	JTAG TDO 及 SWO 信号。
TMS	117	PC1 (1)	I	TTL	JTAG TMS 及 SWDIO 信号。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 4.3 功能描述

JTAG 模块的高级概念图如图 4-1 在 193 页所示。JTAG 模块由测试访问端口 (TAP) 控制器和带有并行更新寄存器的串行移位链组成。TAP 控制器是一个简单的状态机，由 TCK 和 TMS 输入控制。TAP 控制器的当前状态取决于在 TCK 上升沿时在 TMS 管脚所捕获的值的序列。TAP 控制器决定了串行移位链何时捕获新数据，何时将数据从 TDI 移位到 TDO 以及何时更新并行加载寄存器。TAP 控制器的当前状态还决定了当前是要访问指令寄存器 (IR) 链还是某个数据寄存器 (DR) 链。

带有并行加载寄存器的串行移位链由一个指令寄存器 (IR) 链和多个数据寄存器 (DR) 链组成。加载在并行加载寄存器中的当前指令决定了哪个 DR 链在 TAP 控制器排序过程中被捕捉、移位或更新。

某些指令 (如 EXTEST 和 INTEST) 会对当前位于 DR 链的数据进行操作，但不会捕捉、移动或更新任何链。为了确保 TDI 和 TDO 之间的串行通道一直连接，未被执行的指令将会译码成旁路 (BYPASS) 指令 (有关已执行指令的列表，请参阅表 4-3 在 199 页)。

有关 JTAG 时序图，请参阅“JTAG and Boundary Scan”在 1294 页。

**注意：** 在所有可能的复位源中，只有上电复位 (POR) 和确定的  $\overline{\text{RST}}$  输入对 JTAG 模块产生影响。管脚配置由  $\overline{\text{RST}}$  输入和 POR 来复位，但是内部的 JTAG 逻辑只能由 POR 来复位。更多复位信息，请参考“复位源”在 204 页。

### 4.3.1 JTAG 接口管脚

JTAG 接口由四个标准管脚组成：TCK、TMS、TDI 和 TDO。有关这些管脚的信息，以及它们在上电复位或者  $\overline{\text{RST}}$  输入导致复位之后的相关状态，请参阅表 4-2。下面接着详细介绍各个管脚的信息。

**注意：** 以下管脚配置为复位后的 JTAG 端口管脚。有关如何重新编程配置这些管脚，请参阅“通用输入/输出端口 (GPIO)”在 597 页。

表 4-2. 上电复位或  $\overline{\text{RST}}$  置位后的 JTAG 端口管脚状态

管脚名称	数据方向	内部上拉	内部下拉	驱动强度	驱动值
TCK	输入	启用	禁用	N/A	N/A
TMS	输入	启用	禁用	N/A	N/A
TDI	输入	启用	禁用	N/A	N/A
TDO	输出	启用	禁用	2-mA 驱动器	高阻

#### 4.3.1.1 测试时钟输入 (TCK)

TCK 管脚是 JTAG 模块的时钟。借助该时钟，测试逻辑可以独立于其它系统时钟而单独运行。该时钟还确保以菊链连接的多个 JTAG TAP 控制器可以在元件之间同步传送串行测试数据。正常工作期间，TCK 通过一个自由运行的时钟 (额定占空比为 50%) 来驱动。必要时，TCK 可以在 0 或 1 停止一段时间。当 TCK 保持为 0 或 1 时，TAP 控制器的状态不发生改变，且 JTAG 指令和数据寄存器中的数据不会丢失。

默认情况下，TCK 管脚的内部上拉电阻在复位后将启用，以确保管脚在没有外部源驱动的情况下不进行计时。只要 TCK 管脚连续被外部源驱动，我们就可以通过关闭内部上拉和下拉电阻来节省内部功耗 (请参阅 624 页和 626 页)。

#### 4.3.1.2 测试模式选择 (TMS)

TMS 管脚用于选择 JTAG TAP 控制器的下一个状态。TMS 在 TCK 的上升沿进行采样。根据当前的 TAP 状态和 TMS 的采样值，系统可能进入下一个状态。因为 TMS 管脚在 TCK 的上升沿被采样，所以根据“IEEE 标准 1149.1”，TMS 的值在 TCK 的下降沿改变。

保持 TMS 高电平持续 5 个连续的 TCK 周期，将驱使 TAP 控制器状态机进入测试-逻辑-复位 (Test-Logic-Reset) 状态。当 TAP 控制器进入测试-逻辑-复位 (Test-Logic-Reset) 状态时，JTAG 模块和相关寄存器都将复位为默认值。该过程可用于初始化 JTAG 控制器。图 4-2 在 196 页中列出了所有 JTAG 测试访问端口状态机。

默认情况下，TMS 管脚的内部上拉电阻在复位后将启用。GPIO 端口 C 变为上拉电阻设置时，应该确保 PC1/TMS 上的内部上拉电阻保持启用；否则可能会丢失 JTAG 通信（请参阅 624 页）。

#### 4.3.1.3 测试数据输入 (TDI)

TDI 管脚将串行信息流传送给 IR 链和 DR 链。TDI 在 TCK 的上升沿被采样，根据当前 TAP 状态和当前指令，它将这个数据传递给相应的移位寄存器链。因为 TDI 管脚在 TCK 的上升沿被采样，所以根据“IEEE 标准 1149.1”，TDI 的值在 TCK 的下降沿改变。

默认情况下，TDI 管脚的内部上拉电阻在复位后将启用。GPIO 端口 C 变为上拉电阻设置时，应该确保 PC2/TDI 上的内部上拉电阻保持启用；否则可能会丢失 JTAG 通信（请参阅 624 页）。

#### 4.3.1.4 测试数据输出 (TDO)

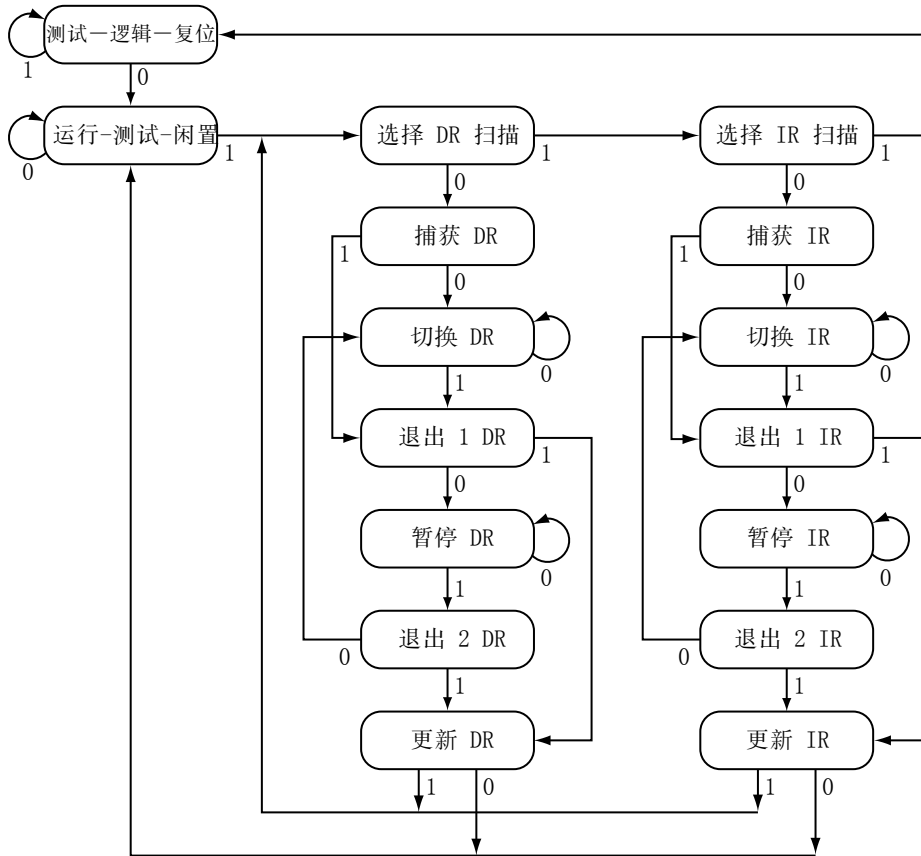
TDO 管脚将串行信息流从 IR 链或 DR 链输出。TDO 的值取决于当前的 TAP 状态、当前指令以及正在访问的数据链中的数据。要在不使用 JTAG 端口时节省功耗，若当前并没有移出数据的活动，TDO 管脚可处于非激活驱动状态。因为在菊花链配置中，TDO 可以与另一个控制器的 TDI 管脚相连，所以根据“IEEE 标准 1149.1”，TDO 的值在 TCK 的下降沿发生变化。

默认情况下，TDO 管脚的内部上拉电阻在复位后将启用，这样确保了在 JTAG 端口闲置时，该管脚依然保持恒定的逻辑电平。在特定的 TAP 控制状态中，如果高阻输出值尚可接受，那么可以通过关闭内部上拉和内部下拉电阻来节省内部功耗（请参阅 624 页和 626 页）。

### 4.3.2 JTAG TAP 控制器

JTAG TAP 控制器状态机如图 4-2 所示。TAP 控制器状态机在上电复位 (POR) 时将重置为测试-逻辑-复位 (Test-Logic-Reset) 状态。要在微控制器上电后复位 JTAG 模块，TMS 输入必须保持高电平并持续 5 个 TCK 时钟周期，以复位 TAP 控制器和所有相关的 JTAG 链。在 TMS 管脚上发出正确的序列，即可使 JTAG 模块在扩展测试序列期间移入新指令、移入数据或变为空闲。有关 TAP 控制器功能和每个状态发生的操作的详细信息，请参考“IEEE 标准 1149.1”。

图 4-2. 测试访问端口状态机



### 4.3.3 移位寄存器

移位寄存器由串行移位寄存器链和并行加载寄存器组成。当 TAP 控制器处于捕获 (CAPTURE) 状态时，串行移位寄存器链将对特定信息进行采样；当 TAP 控制器处于移位 (SHIFT) 状态时，它将允许该信息通过 TDO 管脚移出。当采样数据正从 TDO 管脚移出链时，新的数据将从 TDI 管脚移入串行移位寄存器。当 TAP 控制器处于更新 (UPDATE) 状态时，这些新数据将加载到并行加载寄存器。有关各移位寄存器的详细信息，请参阅“寄存器描述”在 199 页。

### 4.3.4 操作注意事项

在使用 JTAG 模块时必须考虑某些操作参数。由于 JTAG 管脚可被编程为 GPIO，所以必须考虑这些管脚的电路板配置和复位条件。另外，JTAG 模块已经集成了 ARM 串行线调试器。这两种操作模式的切换方法如下所述。

#### 4.3.4.1 GPIO 功能

当微控制器因 POR 或者  $\overline{RST}$  而复位时，JTAG/SWD 端口管脚将重置为默认的 JTAG/SWD 配置。默认配置包括启用数字功能（在端口 C 的 GPIO 数字使能 (GPIODEN) 寄存器中的 DEN[3:0] 置位）、启用上拉电阻（在端口 C 的 GPIO 上拉电阻选择 (GPIOPUR) 寄存器中的 PUE[3:0] 置位）、禁用下拉电阻（在端口 C 的 GPIO 下拉电阻选择 (GPIOPDR) 寄存器中的 PDE[3:0] 清零）以及启用 JTAG/SWD 管脚的复用硬件功能（在端口 C 的 GPIO 备用功能选择 (GPIOAFSEL) 寄存器中的 AFSEL[3:0] 置位）。请参阅 618 页、624 页、626 页以及 629 页。

复位后软件可以通过将 Port C GPIOAFSEL 寄存器的 AFSEL[3:0] 清零来把这些管脚配置为 GPIO。当用户不需要将 JTAG/SWD 端口用于调试或电路板级测试时，这一功能就能多提供 4 个 GPIO，以便在设计中使用。

**小心** – 用户可以建立一个软件序列来阻止调试器连接到 Stellaris 微控制器。如果将程序代码加载到 Flash 中会立即将 JTAG 管脚变成其 GPIO 功能，那么在 JTAG 管脚功能切换之前，调试器将没有足够的时间去连接和终止控制器。结果调试器可能被锁定在该部分外。为了避免这个问题，可以使用一个基于外部或软件触发器的软件程序来恢复 JTAG 功能。

GPIO 确认控制寄存器提供的保护功能可防止对重要硬件外设的意外编程。系统针对可用作四个 JTAG/SWD 管脚 (PC[3:0]) 和 NMI 管脚 (PD7 和 PF0) 的 GPIO 管脚提供了保护功能。向 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (请参阅 618 页)、GPIO 上拉电阻选择 (GPIOPUR) 寄存器 (请参阅 624 页)、GPIO 下拉电阻选择 (GPIOPDR) 寄存器 (请参阅 626 页) 以及 GPIO 数字使能 (GPIODEN) 寄存器 (请参阅 629 页) 中受保护的位写入数据将不会确认保存，除非 GPIO 锁定 (GPIOLOCK) 寄存器 (请参阅 631 页) 没有被锁定，同时 GPIO 确认 (GPIOCR) 寄存器 (请参阅 632 页) 中相应的位被置位。

#### 4.3.4.2 JTAG/SWD 通信

由于调试时钟和系统时钟可以在不同的频率运行，所以必须注意保持与 JTAG/SWD 接口的可靠通信。处于捕获-DR (Capture-DR) 状态时，将被返回先前处理的结果 (如果有)，同时伴随一个 3 位的应答响应。软件应该检测这个应答响应，以查看在初始化一个新的通信前，先前的操作是否已经完成。或者，如果系统时钟至少比调试时钟 (TCK 或 SWCLK) 快 8 倍，那么先前的操作将有足够的时间来完成，应答 (ACK) 位也无需检查。

#### 4.3.4.3 恢复“锁定的”微控制器

**注意：** 执行以下序列将会把“非易失性寄存器编程”在 508 页中列出的非易失性寄存器恢复为出厂默认值。执行以下序列将先对 Flash 存储器进行整体擦除，然后恢复非易失性寄存器。

另外，执行以下序列还将擦除 EEPROM，并将其换位写入计数器返回到出厂默认值。

如果软件将任意一个 JTAG/SWD 管脚配置为 GPIO，并且失去与调试器进行通信的能力，那么可以用调试端口解锁序列来恢复微控制器。当微控制器保持在复位时，执行总共 10 次的 JTAG 到 SWD 和 SWD 到 JTAG 的切换序列，将会整体擦除闪存。调试端口解锁序列为：

1. 发出并保持  $\overline{\text{RST}}$  信号。
2. 向器件供电。
3. 执行“JTAG 到 SWD 切换”一节在 198 页中 JTAG 到 SWD 切换序列的第 1 步和第 2 步。
4. 执行“SWD 到 JTAG 切换”一节在 198 页中 SWD 到 JTAG 切换序列的第 1 步和第 2 步。
5. 执行 JTAG 到 SWD 切换序列的第 1 步和第 2 步。
6. 执行 SWD 到 JTAG 切换序列的第 1 步和第 2 步。
7. 执行 JTAG 到 SWD 切换序列的第 1 步和第 2 步。
8. 执行 SWD 到 JTAG 切换序列的第 1 步和第 2 步。
9. 执行 JTAG 到 SWD 切换序列的第 1 步和第 2 步。
10. 执行 SWD 到 JTAG 切换序列的第 1 步和第 2 步。
11. 执行 JTAG 到 SWD 切换序列的第 1 步和第 2 步。

12. 执行 SWD 到 JTAG 切换序列的第 1 步和第 2 步。
13. 释放  $\overline{\text{RST}}$  信号。
14. 等待 400 ms。
15. 微控制器将重启电源。

#### 4.3.4.4 ARM 串行线调试 (SWD)

要无缝集成 ARM 串行线调试 (SWD) 功能，串行线调试器必须能够与 Cortex-M4F 内核相连，而无需执行或者了解 JTAG 运行周期。该集成可以通过一个在 SWD 会话开始前发出 SWD 前导码来实现。

用来启用 SWJ-DP 模块 SWD 接口的切换前导码将在 TAP 控制器处于 Test-Logic-Reset 状态时开启。此后，前导码会使 TAP 控制器依次进入以下状态：运行-测试-闲置 (Run Test Idle)、选择 DR (Select DR)、选择 IR (Select IR)、测试 - 逻辑 - 复位 (Test Logic Reset)、测试 - 逻辑 - 复位 (Test Logic Reset)、运行-测试 - 闲置 (Run Test Idle)、运行-测试 - 闲置 (Run Test Idle)、选择 DR (Select DR)、选择 IR (Select IR)、测试 - 逻辑 - 复位 (Test Logic Reset)、测试 - 逻辑 - 复位 (Test Logic Reset)、运行-测试 - 闲置 (Run Test Idle)、运行-测试 - 闲置 (Run Test Idle)、选择 DR (Select DR)、选择 IR (Select IR) 以及测试 - 逻辑 - 复位 (Test Logic Reset) 状态。

依次进入 TAP 状态机的此序列将启用 SWD 接口并禁用 JTAG 接口。有关该操作和 SWD 接口的更多信息，请参考“ARM® Debug Interface V5 架构说明”。

由于这个序列是一系列有效的可发出的 JTAG 操作，所以 ARM JTAG TAP 控制器并不完全与“IEEE 标准 1149.1”一致。这是 ARM JTAG TAP 控制器唯一不完全符合规范的地方。由于 TAP 控制器在正常操作中出现该序列的可能性很低，所以应该不会影响 JTAG 接口的正常运行。

#### JTAG 到 SWD 切换

为了将调试访问端口 (DAP) 的操作模式由 JTAG 切换到 SWD，外部调试硬件必须向微控制器发送切换前导码。用于切换到 SWD 模式的 16 位 TMS 命令为 b1110.0111.1001.1110，先发送最低有效位 (LSB)。当先发送最低有效位 (LSB) 时，该命令也可以表示为 0xE79E。完整的切换序列应该包含 TCK/SWCLK 以及 TMS/SWDIO 信号上的以下操作：

1. TMS/SWDIO 为高电平时，发送至少 50 个 TCK/SWCLK 周期，以确保 JTAG 和 SWD 都处于复位/空闲状态。
2. 在 TMS 上发送 JTAG 到 SWD 的 16 位切换命令 0xE79E。
3. TMS/SWDIO 为高电平时，发送至少 50 个 TCK/SWCLK 周期，以确保当 SWJ-DP 已经处于 SWD 模式时，SWD 将在发送切换序列之前进入线复位状态。

#### SWD 到 JTAG 切换

要将调试访问端口 (DAP) 的操作模式由 SWD 切换到 JTAG，外部调试硬件必须向微控制器发送切换前导码。切换到 JTAG 模式的 16 位 TMS 命令为 b1110.0111.0011.1100，先发送最低有效位 (LSB)。当先发送最低有效位 (LSB) 时，这个命令也可以表示为 0xE73C。完整的切换序列应该包含 TCK/SWCLK 以及 TMS/SWDIO 信号上的以下操作：

1. TMS/SWDIO 为高电平时，发送至少 50 个 TCK/SWCLK 周期，以确保 JTAG 和 SWD 都处于复位/空闲状态。
2. 在 TMS 上发送 SWD 到 JTAG 的 16 位切换命令 0xE73C。

3. TMS/SWDIO 为高电平时，发送至少 50 个 TCK/SWCLK 周期，以确保当 SWJ-DP 已经处于 JTAG 模式时，JTAG 将在发送切换序列之前进入测试 - 逻辑 - 复位 (Test Logic Reset) 状态。

## 4.4 初始化及配置

在上电复位或外部复位 ( $\overline{RST}$ ) 后，JTAG 管脚将被自动配置为 JTAG 通信功能。不需要执行用户自定义的初始化或配置。但是，如果用户应用程序将这些管脚变成 GPIO 功能，那么在恢复 JTAG 通信前，这些管脚必须恢复 JTAG 功能。要使管脚恢复 JTAG 功能，请通过 GPIOAFSEL 寄存器将 4 个 JTAG 管脚 (PC[3:0]) 设置为其复用功能。另外，要启用复用功能，应将 4 个 JTAG 管脚 (PC[3:0]) 上的 GPIO 管脚配置的任何其他更改都恢复为默认设置。

## 4.5 寄存器描述

JTAG TAP 控制器或移位寄存器链中的寄存器都没有进行存储器映射，它们不能通过芯片中的高级外设总线 (APB) 访问。但是，这些 JTAG 控制器里的寄存器都可以通过 TAP 控制器进行串行访问。这些寄存器包括指令寄存器和 6 个数据寄存器。

### 4.5.1 指令寄存器 (IR)

JTAG TAP 指令寄存器 (IR) 是一个 4 位串行扫描链。它连接在 JTAG 的 TDI 和 TDO 管脚之间，并且带有一个并行加载寄存器。当 TAP 控制器处于正确的状态时，相应的位便可移入指令寄存器 (IR)。这些位一旦移入链中并且更新后，就会被解析成当前指令。有关指令寄存器 (IR) 位的译码，请参阅表 4-3。下面对每条指令及其相关的数据寄存器进行了详细解释。

表 4-3. JTAG 指令寄存器命令

IR[3:0]	指令	描述
0x0	EXTEST	通过采样/预加载 (SAMPLE/PRELOAD) 指令将预加载到边界扫描链的值驱动到管脚上。
0x1	INTEST	通过采样/预加载 (SAMPLE/PRELOAD) 指令将预加载到边界扫描链的值驱动到控制器里。
0x2	SAMPLE/PRELOAD	捕获当前的 I/O 值，并在新的预加载数据移入时将采样的值移出边界扫描链。
0x8	ABORT	将数据移入 ARM 调试端口中止寄存器
0xA	DPACC	将数据移入和移出 ARM DP 访问寄存器。
0xB	APACC	将数据移入和移出 ARM AC 访问寄存器。
0xE	IDCODE	将“IEEE 标准 1149.1”定义的制造信息加载到 IDCODE 链并将它移出。
0xF	BYPASS	通过一个移位寄存器链将 TDI 连接到 TDO。
其它	保留	默认为 BYPASS 指令，以确保 TDI 总是连接到 TDO。

#### 4.5.1.1 EXTEST 指令

EXTEST 指令与本身的数据寄存器链没有联系。EXTEST 指令使用的是通过采样/预加载 (SAMPLE/PRELOAD) 指令预加载到边界扫描数据寄存器里的数据。当指令寄存器收到 EXTEST 指令时，边界扫描数据寄存器中预加的与输出和输出启用相关的载数据将用于驱动 GPIO 管脚，而不用于驱动来自内核的信号。通过将已知的值驱动到控制器外的测试，该指令可用于验证联通性。当指令寄存器收到 EXTEST 指令时，将访问边界扫描数据寄存器以对当前数据采样，然后将其移出，同时将新的数据加载到边界扫描数据寄存器。

#### 4.5.1.2 INTEST 指令

INTEST 指令与本身的数据寄存器链没有关联。INTEST 指令使用的是通过采样/预加载 (SAMPLE/PRELOAD) 指令预加载到边界扫描数据寄存器里的数据。当指令寄存器收到 INTEST 指

令时，边界扫描数据寄存器里预加载的与输入相关的数据将用于驱动进入内核的信号，而不用于驱动来自 GPIO 管脚的信号。通过将已知的值驱动进控制器，该指令可以用于测试。特别需要注意的是，虽然 RST 输入管脚在边界扫描数据寄存器链上，但它只供观察。当指令寄存器收到 INTEST 指令时，将访问边界扫描数据寄存器以对当前数据采样，然后将其移出，同时将新的数据加载到边界扫描数据寄存器。

#### 4.5.1.3 SAMPLE/PRELOAD 指令

采样/预加载 (SAMPLE/PRELOAD) 指令用于连接 TDI 和 TDO 之间的边界扫描数据寄存器链。该指令将对当前管脚的状态采样以供观察，并且预加载新的测试数据。每个 GPIO 管脚都有一个相关的输入、输出和输出启用信号。在该指令执行期间，当 TAP 控制器进入捕获 DR (Capture DR) 状态时，每个 GPIO 管脚的输入、输出和输出启用信号都会被捕获。当 TAP 控制器处于位移，DR (Shift DR) 状态时这些采样通过 TDO 被串行移出，它们可以在各种测试中用于观察和比较。

在这些输入、输出和输出启用信号的采样值被移出边界扫描数据寄存器的同时，新的数据也正通过 TDI 管脚移入边界扫描数据寄存器。一旦新数据被移入边界扫描数据寄存器，当 TAP 控制器进入更新 DR (Update DR) 状态时，数据将被保存在并行加载寄存器中。这种并行加载寄存器的更新可以将数据预加载到每个与输入、输出和输出启用相关的边界扫描数据寄存器中。预加载数据可以配合 EXTEST 与 INTEST 指令使用，以将数据驱动进入内核或将数据驱动到内核外。更多信息请参阅“边界扫描数据寄存器”在 201 页。

#### 4.5.1.4 ABORT 指令

终止 (ABORT) 指令用于连接 TDI 和 TDO 之间相关的中止数据寄存器链。该指令提供对 ARM 调试访问端口 (DAP) 的中止寄存器的读取和写入访问。将相应数据移入这个数据寄存器即可将各种错误位清零，或对之前的请求启动 DAP 中止。更多信息请参阅“ABORT 数据寄存器”在 202 页。

#### 4.5.1.5 DPACC 指令

DPACC 指令用于连接 TDI 和 TDO 之间相关的 DPACC 数据寄存器链。该指令提供对 ARM 调试访问端口 (DAP) 的 DPACC 寄存器的读取和写入访问。将相应数据移入该寄存器并且从该寄存器读取输出数据，便可以对 ARM 调试和状态寄存器进行读取和写入访问。更多信息请参阅“DPACC 数据寄存器”在 202 页。

#### 4.5.1.6 APACC 指令

APACC 指令用于连接 TDI 和 TDO 之间相关的 APACC 数据寄存器链。该指令提供对 ARM 调试访问端口 (DAP) 的 APACC 寄存器的读和写访问。将数据移入该寄存器并且从该寄存器读取输出数据，便可以通过调试端口对内部组件和总线进行读和写访问。更多信息请参阅“APACC 数据寄存器”在 202 页。

#### 4.5.1.7 IDCODE 指令

IDCODE 指令用于连接 TDI 和 TDO 之间相关的 IDCODE 数据寄存器链。该指令提供了有关制造商，器件编号和 ARM 内核版本等信息。测试设备和调试器可以使用这些信息来自动配置输入和输出数据流。当上电复位 (POR) 或者进入 Test-Logic-Reset 状态时，IDCODE 指令被默认加载到 JTAG 的指令寄存器。更多信息请参阅“IDCODE 数据寄存器”在 201 页。

#### 4.5.1.8 BYPASS 指令

BYPASS 指令用于连接 TDI 和 TDO 之间相关的 BYPASS 数据寄存器链。该指令用来在 TDI 和 TDO 端口之间创建一条长度最短的串行路径。BYPASS 数据寄存器是个 1 位移位寄存器。对于不需要特别测试的组件，通过加载 BYPASS 指令可以在 JTAG 扫描链中将它们旁路掉，由此提高了测试效率。更多信息请参阅“BYPASS 数据寄存器”在 201 页。



## 4.5.2 数据寄存器

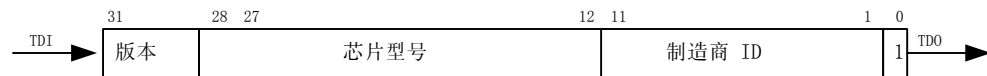
JTAG 模块包含 6 个数据寄存器。串行数据寄存器链包括：IDCODE、BYPASS、边界扫描、APACC、DPACC 和 ABORT。它们将在下面的小节中进行讨论。

### 4.5.2.1 IDCODE 数据寄存器

根据“IEEE 标准 1149.1”的定义，32 位 IDCODE 数据寄存器的格式如图 4-3 所示。该标准要求每个与 JTAG 兼容的设备都将 IDCODE 指令或者 BYPASS 指令当作默认指令来执行。IDCODE 数据寄存器的 LSB (最低位) 被定义成 1，以将它和 LSB 为 0 的 BYPASS 指令区分开来。这使得自动配置测试工具可以决定哪个指令是默认指令。

JTAG 端口主要用于制造商测试组件以及编程开发和调试。为了便于使用自动配置调试工具，IDCODE 指令将输出 0x4BA00477 值。这样，调试器就可以自动进行配置以便在调试过程中正确配合 Cortex-M4F 工作。

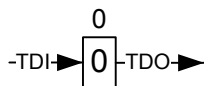
图 4-3. IDCODE 寄存器格式



### 4.5.2.2 BYPASS 数据寄存器

根据“IEEE 标准 1149.1”的定义，1 位 BYPASS 数据寄存器的格式如图 4-4 所示。该标准要求每个与 JTAG 兼容的器件将 BYPASS 指令或者 IDCODE 指令当作默认指令来执行。BYPASS 数据寄存器的 LSB 被定义成 0，以便将它与 LSB 为 1 的 IDCODE 指令区分开来。这使得自动配置测试工具可以决定哪个指令是默认指令。

图 4-4. BYPASS 寄存器格式

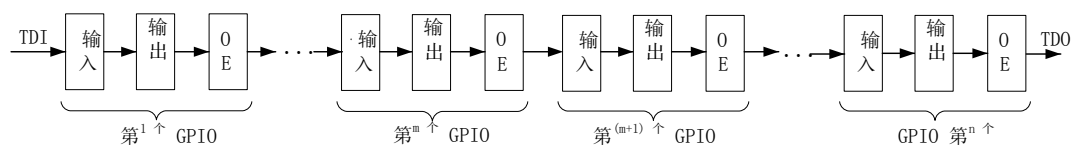


### 4.5.2.3 边界扫描数据寄存器

边界扫描数据寄存器的格式如图 4-5 所示。每个 GPIO 管脚都包含在边界扫描数据寄存器中，这些管脚的顺序为 JTAG 端口管脚排列的逆时针方向。每个 GPIO 管脚有 3 个相关的数字信号包含在链中。这些信号是输入，输出和输出启用，它们按照图示顺序排列。

当 SAMPLE/PRELOAD 指令访问边界扫描数据寄存器时，每个数字管脚的输入，输出和输出启用被采样并且移出链供校验。这些值的采样发生在 TCK 上升沿，TAP 控制器处于 Capture DR 状态。当 TAP 控制器在 Shift DR 状态下采样数据被移出边界扫描链时，新的数据可以预加载入链以便和 EXTEST 指令和 INTEST 指令一起使用。EXTEST 指令强制将数据移出控制器，INTEST 指令强制将数据移入控制器。

图 4-5. 边界扫描寄存器格式



#### 4.5.2.4 APACC 数据寄存器

根据 ARM 的定义，35 位 APACC 数据寄存器的格式请参考“ARM® Debug Interface V5 架构说明”。

#### 4.5.2.5 DPACC 数据寄存器

根据 ARM 的定义，35 位 DPACC 数据寄存器的格式请参考“ARM® Debug Interface V5 架构说明”。

#### 4.5.2.6 ABORT 数据寄存器

根据 ARM 的定义，35 位 ABORT 数据寄存器的格式请参考“ARM® Debug Interface V5 架构说明”。

## 5 系统控制

系统控制配置整个器件的操作，并提供器件信息。可配置的特性包括复位控制、NMI操作、功率控制、时钟控制和低功耗模式。

### 5.1 信号描述

以下表格列出了系统控制模块的外部信号，并描述了各自的功能。NMI 信号是GPIO 信号的复用功能，复位时用作 GPIO 功能。PD7 和 PF0 具备提交保护，并需要特定的处理过程才能被配置为任意备用功能或者随后返回 GPIO 功能，请参考“提交控制”在 604页。下表中“管脚复用/赋值”栏给出了 NMI 信号的 GPIO 管脚位置。GPIO 备选功能选择 (GPIOAFSEL) 寄存器 (618页) 里的 AFSEL 位应被置位以选择 NMI 功能。括号中的数字是必须写入 GPIO 端口控制 (GPIOCTL) 寄存器 (636页) 里的 PMCn 域的编码，以便将 NMI 信号分配给指定的 GPIO 端口管脚。更多关于 GPIO 配置的问题，请参考“通用输入/输出端口 (GPIO)”在 597页。剩下的信号 (在“管脚复用/赋值”栏中标记为“固定”) 有一个固定的管脚分配和功能。

表 5-1. 系统控制及时钟 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
NMI	62 144	PF0 (8) PD7 (8)	I	TTL	不可屏蔽的中断。
OSC0	92	固定	I	模拟	晶体主振荡器输入或外部时钟参考输入。
OSC1	93	固定	O	模拟	晶体主振荡器输出。使用单端时钟源时不连接。
RST	90	固定	I	TTL	系统复位输入。

a. TTL 表示管脚的电压水平与 TTL 一致。

### 5.2 功能描述

系统控制模块提供以下功能：

- 器件标识，见“器件标识”在 203页
- 本地控制，例如复位 (见“复位控制”在 203页)，电源 (见“功率控制”在 208页) 和时钟控制 (见“时钟控制”在 209页)
- 系统控制 (运行、睡眠和深度睡眠模式)，见“系统控制”在 215页

#### 5.2.1 器件标识

有些只读寄存器可以向软件提供关于微控制器的信息，比如版本、器件编号、存储器大小以及其器件上的外设存在。器件标识 0 (DID0) (223页) 和器件标识 1 (DID1) (225页) 寄存器提供关于器件版本、封装、温度范围等的详细信息。以系统控制偏移量 0x300 开始的外设存在寄存器，如看门狗定时器外设存在 (PPWD) 寄存器，提供关于器件中包含多少模块类型的信息。最后，外设属性寄存器的各个外设寄存器空间的偏移量 0xFC0 处提供了片上外设性能的信息 (如 GPTM 外设属性 (GPTMPP) 寄存器)。之前生成的 Stellaris® 器件使用器件功能 (DC0-DC9) 寄存器显示关于外设及其性能的信息。这些器件的寄存器用于向后兼容软件，但不提供之前的器件所不适用的外设信息。

#### 5.2.2 复位控制

本小节论述复位过程中硬件方面的功能和复位序列之后的系统软件要求。

### 5.2.2.1 复位源

LM4F232H5QD 微控制器有 6 个复位源：

1. 上电复位 (POR) (见第 205 页)。
2. 外部复位输入管脚 ( $\overline{\text{RST}}$ ) 有效 (见第 205 页)。
3. 内部掉电 (BOR) 检测器 (见第 206 页)。
4. 软件启动的复位 (利用软件复位寄存器) (见第 207 页)。
5. 违反看门狗定时器复位条件 (见第 207 页)。
6. MOSC 故障 (见第 208 页)。

表 5-2 提供了不同复位操作结果的摘要。

表 5-2. 复位源

复位源	内核复位？	JTAG 复位？	片上外设复位？
上电复位	是	是	是
$\overline{\text{RST}}$	是	仅管脚配置	是
掉电复位	是	仅管脚配置	是
使用 APINT 寄存器中的 SYSRESREQ 位进行软件系统请求复位。	是	仅管脚配置	是
使用 APINT 寄存器中的 VECTRESET 位进行软件系统请求复位。	是	仅管脚配置	否
软件外设复位	否	仅管脚配置	是 <sup>a</sup>
看门狗复位	是	仅管脚配置	是
MOSC 故障复位	是	仅管脚配置	是

a. 使用软件复位控制寄存器可在在模块基础上进行编程。

复位后，复位原因 (RESC) 寄存器根据复位原因置位。该寄存器中的位具有粘着特性 (sticky)，经过多个复位序列后仍能保持其状态，内部POR复位除外。内部POR复位后，RESC寄存器中除POR指示器对应位之外的所有位都清零。RESC寄存器的位可以通过写0来清零。

任何复位内核的复位中，通过使用启动配置 (BOOTCFG) 寄存器中配置好的GPIO信号，用户可以选择让内核直接执行 ROM 的 Boot Loader 或 Flash 存储器上的应用程序。

复位时，将执行以下序列：

1. 读取 BOOTCFG 寄存器。如果 EN 位清零，那么执行 ROM 的 Boot Loader。
2. 在ROM的Boot Loader中，指定的GPIO管脚的状态与规定的极性相比较，如果管脚状态与规定的极性匹配，那么将 ROM 映射到地址 0x0000.0000 并继续执行 ROM 的 Boot Loader。
3. 如果 EN 位置位或管脚状态与规定的极性不匹配，那么读取地址 0x0000.0004 的数据。如果该地址的数据是 0xFFFF.FFFF，那么将 ROM 映射到地址 0x0000.0000 并继续执行 ROM 的 Boot Loader。

4. 如果地址 0x0000.0004 的数据不是 0xFFFF.FFFF，堆栈指针 (SP) 装载 Flash 存储器地址 0x0000.0000 的数据，程序计数器 (PC) 装载地址 0x0000.0004 的数据。用户应用程序开始执行。

例如，如果 BOOTCFG 寄存器写入值 0x0000.3C01 并提交，那么复位时将检查 PB7 来决定是否执行 ROM 的 Boot Loader。如果 PB7 是低电平，内核无条件的开始执行 ROM 的 Boot Loader。如果 PB7 是高电平，若 0x0000.0004 位置的复位向量不是 0xFFFF.FFFF，那么执行 Flash 存储器的应用程序，若复位向量是 0xFFFF.FFFF，则执行 ROM 的 Boot Loader。

### 5.2.2.2 上电复位 (POR)

注意： JTAG 控制器仅可以通过上电复位重置。

内部上电复位 (POR) 电路监测电源电压 ( $V_{DD}$ )，并且在电源达到阈值 ( $V_{TH}$ ) 时向包括 JTAG 在内的所有内部逻辑产生复位信号。当片上上电复位脉冲结束时，微控制器必须在规定的参数范围内工作（请参考“Power and Brown-Out”在 1295 页）。对于一些应用，要求使用外部复位信号使微控制器保持在复位状态比内部 POR 更长的时间，这时可以使用  $\overline{RST}$  输入，如“外部  $\overline{RST}$  管脚”在 205 页所讨论的。

上电复位序列如下：

1. 微控制器等待内部 POR 变为无效。
2. 内部复位释放，内核从存储器加载初始堆栈指针、初始程序计数器以及由程序计数器指定的第一条指令，然后开始执行。

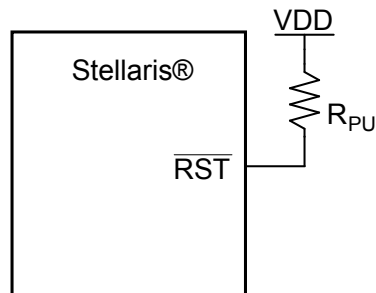
内部 POR 只在微控制器最初上电以及休眠唤醒时激活。上电复位时序如图 25-6 在 1296 页所示。

### 5.2.2.3 外部 $\overline{RST}$ 管脚

注意： 建议对  $\overline{RST}$  信号的跟踪线路越短越好。确保将连接  $\overline{RST}$  信号的任何元件放置得尽可能靠近微控制器。

如果应用中只使用内部 POR 电路，那么  $\overline{RST}$  输入必须通过一个可选的上拉电阻（0 至 100 k $\Omega$ ）连接到电源 ( $V_{DD}$ )，如图 5-1 在 205 页所示。 $\overline{RST}$  输入的滤波功能需要最小脉宽，以便复位脉冲被识别，见表 25-7 在 1296 页。

图 5-1. 基本  $\overline{RST}$  配置



$R_{PU} = 0$  至 100 k $\Omega$

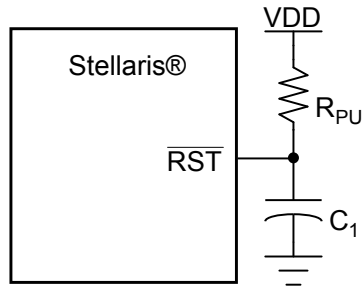
外部复位管脚 ( $\overline{RST}$ ) 复位微控制器，包括内核和所有片上外设。外部复位序列如下：

1. 外部复位管脚 ( $\overline{RST}$ ) 在  $T_{MIN}$  规定的时间持续有效，然后失效（请参考“Reset”在 1296 页）。

- 内部复位释放，内核从存储器加载初始堆栈指针、初始程序计数器以及由程序计数器指定的第一条指令，然后开始执行。

为提高噪声免疫和/或延迟上电复位， $\overline{\text{RST}}$ 输入可以连接一个RC网络，如图 5-2 在 206页所示。

图 5-2. 延长上电复位时间的外部电路

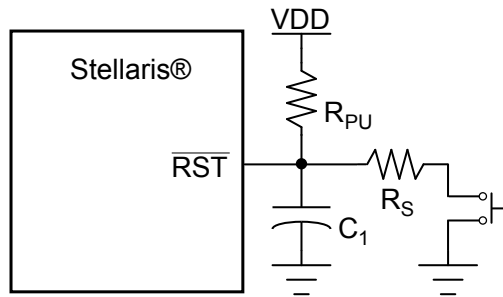


$R_{PU} = 1 \text{ k}\Omega$  至  $100 \text{ k}\Omega$

$C_1 = 1 \text{ nF}$  至  $10 \text{ }\mu\text{F}$

如果应用中需要使用外部复位开关，图 5-3 在 206页显示了适当的电路来使用。

图 5-3. 复位电路由开关控制



典型  $R_{PU} = 10 \text{ k}\Omega$

典型  $R_S = 470 \text{ }\Omega$

$C_1 = 10 \text{ nF}$

$R_{PU}$  和  $C_1$  元件确定了上电延时。

外部复位时序如图 25-8 在 1297页所示。

#### 5.2.2.4 掉电复位 (BOR)

微控制器提供的掉电检测电路在电源 ( $V_{DD}$ ) 降至低于掉电阈值电压 ( $V_{BTH}$ ) 时触发。如果检测到一个掉电条件，系统会产生一个中断或系统复位。默认条件是产生一个中断，所以BOR必须被使能。掉电复位由上电和掉电复位控制 (PBORCTL) 寄存器控制。为了使掉电条件触发一次复位，PBORCTL 寄存器的BORIOR位必须被置位；如果BORIOR位被清零，将会产生一个中断。

掉电复位序列如下：

- 当  $V_{DD}$  降至低于  $V_{BTH}$  时，内部 BOR 条件将置位。

2. 如果 BOR 条件存在，内部复位有效。
3. 内部复位释放，微控制器获取并加载初始堆栈指针、初始程序计数器以及由程序计数器指定的第一条指令，然后开始执行。

上电复位的效果等同于一次有效的外部  $\overline{\text{RST}}$  输入，并且该复位将会保持有效，直到  $V_{\text{DD}}$  恢复到正确的电压级别。在复位中断处理中可以检查 RESC 寄存器来确定掉电条件是否是复位的原因，这就允许软件决定恢复何种操作。

内部掉电复位时序如图 25-7 在 1296 页所示。

### 5.2.2.5 软件复位

软件可以复位一个特别的外设或者对整个微控制器产生一个复位

通过系统控制偏移量 0x500 处开始的外设专用复位寄存器（例如看门狗定时器软件复位 (SRWD) 寄存器），软件可以单独复位各个外设。如果外设对应的位被置位随后清零，那么该外设被复位。

软件可以通过置位应用中断和复位控制 (APINT) 寄存器的 SYSRESREQ 位来复位包括内核在内的整个微控制器。软件启动的系统复位序列如下：

1. 通过置位 SYSRESREQ 位即可产生软件微控制器复位。
2. 内部复位有效。
3. 内部复位失效，微控制器从存储器加载初始堆栈指针、初始程序计数器以及由程序计数器指定的第一条指令，然后开始执行。

软件可以通过置位 APINT 寄存器的 VECTRESET 位来复位内核。软件启动的内核复位序列如下：

1. 通过置位 VECTRESET 位来启动内核复位。
2. 内部复位有效。
3. 内部复位失效，微控制器从存储器加载初始堆栈指针、初始程序计数器以及由程序计数器指定的第一条指令，然后开始执行。

软件启动的系统复位时序如图 25-9 在 1297 页所示。

### 5.2.2.6 看门狗定时器复位

看门狗定时器模块的功能是阻止系统挂起。LM4F232H5QD 微控制器提供两个看门狗定时器模块，以防其中一个看门狗时钟源失效。一个看门狗脱离系统时钟运行，另一个脱离精确内部振荡器 (PIOSC) 运行。除了由于 PIOSC 看门狗定时器模块处于不同的时钟域，在两次访问寄存器时必须要有个时间延迟外，每个模块以相同的方式工作。看门狗定时器可被配置为第一次溢出对微控制器产生一个中断或不可屏蔽中断，第二次溢出产生一个复位。

在看门狗第一次溢出发生后，32 位看门狗计数器会重新加载看门狗定时器加载 (WDTLOAD) 寄存器的值，并从这个值继续递减计数。如果在第一次溢出中断清除之前定时器再次递减计数到零，并且复位信号已启用，那么看门狗定时器将其复位信号发送给微控制器。看门狗定时器复位序列如下：

1. 看门狗定时器第二次溢出时不需要被服务。
2. 内部复位有效。
3. 内部复位释放，微控制器从存储器加载初始堆栈指针、初始程序计数器以及由程序计数器指定的第一条指令，然后开始执行。

关于看门狗定时器模块的更多信息，可查看“看门狗定时器 (WDT)”在 717页。

看门狗复位时序如图 25-10 在 1297页所示。

### 5.2.3 不可屏蔽的中断

微控制器有四个不可屏蔽的中断源 (NMI)：

- NMI信号有效
- 主振荡器校验错误
- 中断控制和状态 (INTCTRL) 寄存器的 NMISSET 位 ( Cortex™M4F 中, [查看 154页] )。
- 看门狗控制 (WDTCTL)寄存器的 INTTYPE 位置位时看门狗模块超时中断 ( 查看 723页 ) 。

为辨认出中断源，软件必须检查中断原因。

#### 5.2.3.1 NMI 管脚

NMI 信号是 GPIO 端口管脚 PD7 或 PF0 的备用功能。如“通用输入/输出端口 (GPIO)”在 597页所描述的那样，在GPIO中该信号若想用于中断，其复用功能必须启用。注意启用NMI复用功能需要使用GPIO锁和提交功能，正如与JTAG/SWD功能相关的GPIO端口管脚那样，见 632页。高电平激活NMI信号；启用的 NMI 信号高于  $V_{IH}$  会启用 NMI 中断序列。

#### 5.2.3.2 主振荡器校验失败

LM4F232H5QD 微控制器提供了一个主振荡器校验电路。如果振荡器运行得太快或太慢，该电路会产生一个错误条件。如果主振荡器校验电路被启用并产生一个错误，此时会产生一个上电复位并将控制权交给 NMI 处理程序，或产生中断。MOSCCTL 寄存器的 MOSCIM 位决定产生的动作。在这两种情况中，系统时钟源自动切换为 PIOSC。发生 MOSC 故障复位时，NMI 处理程序将用于解决主振荡器检验故障，因为可以从通用复位处理程序移除必要代码，加速复位处理过程。通过将主振荡器控制 (MOSCCTL) 寄存器的 CVAL 位置位来启用检测电路。主振荡器校验错误在复位原因 (RESC) 寄存器的主振荡器失败状态 (MOSCFAIL) 位显示。主振荡器校验电路动作的详细描述在“主振荡器校验电路”在 215页。

### 5.2.4 功率控制

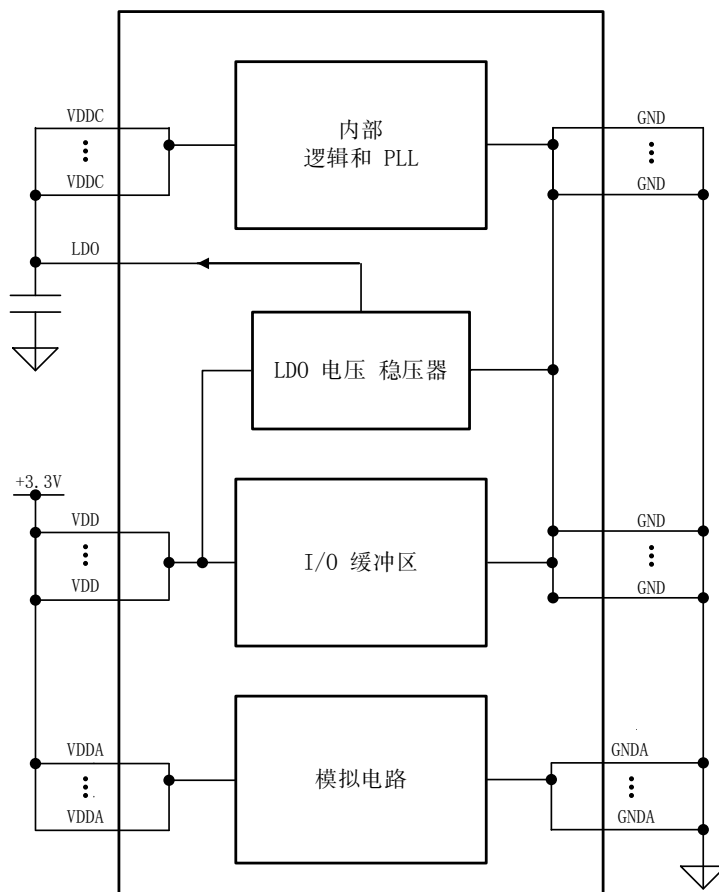
该 Stellaris 微控制器提供了一个集成的LDO调节器，它用来对大多数微控制器的内部逻辑提供电源。图 5-4显示了功率结构。

不可以使用外部LDO。

注意： VDDA 的电压必须符合表 25-2 在 1292页中的规格，否则微控制器不能正常工作。VDDA为设备的所有模拟电路供电，包括时钟电路。



图 5-4. 功率结构



## 5.2.5 时钟控制

系统控制决定了该部分的时钟控制。

### 5.2.5.1 基础时钟源

在微控制器中有多个时钟源可以使用。

- **精确内部振荡器 (PIOSC).** 精确内部振荡器是一个片上时钟源，在POR期间和之后，微控制器使用该时钟源。它不需要使用任何外部元件，可以提供室温  $16\text{ MHz} \pm 1\%$ 、整个温度范围  $\pm 3\%$  的时钟。PIOSC 是为需要精确时钟源并减少系统开销的应用而考虑的。如果需要主振荡器，软件必须在复位后启用主振荡器，并在改变时钟参考前让主振荡器达到稳定。如果休眠模块时间源为  $32.768\text{-kHz}$  振荡器，软件可以根据参考时钟调整内部精确振荡器，以获得更高精确性。不论 PIOSC 是否是系统时钟的源，PIOSC 可以被配置为 ADC 时钟以及波特率时钟的源，以用于 UART 和 SSI，见“系统控制”在 215 页。
- **主振荡器 (MOSC).** 主振荡器可通过两种方式提供一个频率精确的时钟源：外部单端时钟源连接到 OSC0 输入管脚，或者外部晶振串接在 OSC0 输入管脚和 OSC1 输出管脚间。如果 PLL 正在使用，晶振的值必须是  $5\text{ MHz}$  到  $25\text{ MHz}$ （含）之间的一个支持的频率。如果 PLL 没有被使用，晶振可以是  $4\text{ MHz}$  到  $25\text{ MHz}$  之间的任何一个支持的频率。单端时钟源的范围从 DC 到微控制器规定的速度。支持的晶振在 RCC 寄存器（见 236 页）的 XTAL 位域列出。注意 MOSC 必须为 USB PLL 提供一个时钟源，并且必须连接到晶体或振荡器。

- 内部**30KHz**振荡器. 内部 30-kHz 振荡器提供  $30\text{ kHz} \pm 50\%$  的工作频率。它是为在深度睡眠省电模式期间使用的。该省电模式受益于精简的内部配电系统，同时也允许MOSC关闭。
- 休眠模块时钟源. 休眠模块根据连接到 XOSC0 管脚的 32.768-kHz 振荡器计时。32.768-kHz 振荡器可以用于系统时钟，从而消除额外晶体或振荡器的需求。休眠模块时钟源旨在为系统提供实时时钟源，以及为深度睡眠或休眠模式省电提供精确时钟源。

内部系统时钟(SysClk) 源于上述任一时钟源，同时增加两项：主内部PLL输出和四分频的精确内部振荡器 ( $4\text{ MHz} \pm 1\%$ )。PLL时钟参考频率必须在范围5 MHz 到 25 MHz (包括)内。表 5-3 在 210页显示了不同的时钟源如何在系统中使用。

表 5-3. 时钟源选项

时钟源	驱动PLL ?		用作SysClk ?	
精确内部振荡器	是	BYPASS = 0, OSCSRC = 0x1	是	BYPASS = 1, OSCSRC = 0x1
4 分频的精确内部振荡器 ( $4\text{ MHz} \pm 1\%$ )	否	-	是	BYPASS = 1, OSCSRC = 0x2
主振荡器	是	BYPASS = 0, OSCSRC = 0x0	是	BYPASS = 1, OSCSRC = 0x0
内部30KHz振荡器	否	-	是	BYPASS = 1, OSCSRC = 0x3
休眠模块32.768-kHz振荡器	否	-	是	BYPASS = 1, OSCSRC2 = 0x7

### 5.2.5.2 时钟配置

运行模式时钟配置 (RCC) 和运行模式时钟配置 2 (RCC2) 寄存器提供对系统时钟的控制。RCC2寄存器用来扩充域，提供了RCC寄存器包含之外的其它编码。使用时，RCC2寄存器中域的值被RCC寄存器相应域的逻辑使用。特别的，RCC2提供了更多种类的时钟配置选项。这些寄存器控制下面的时钟功能：

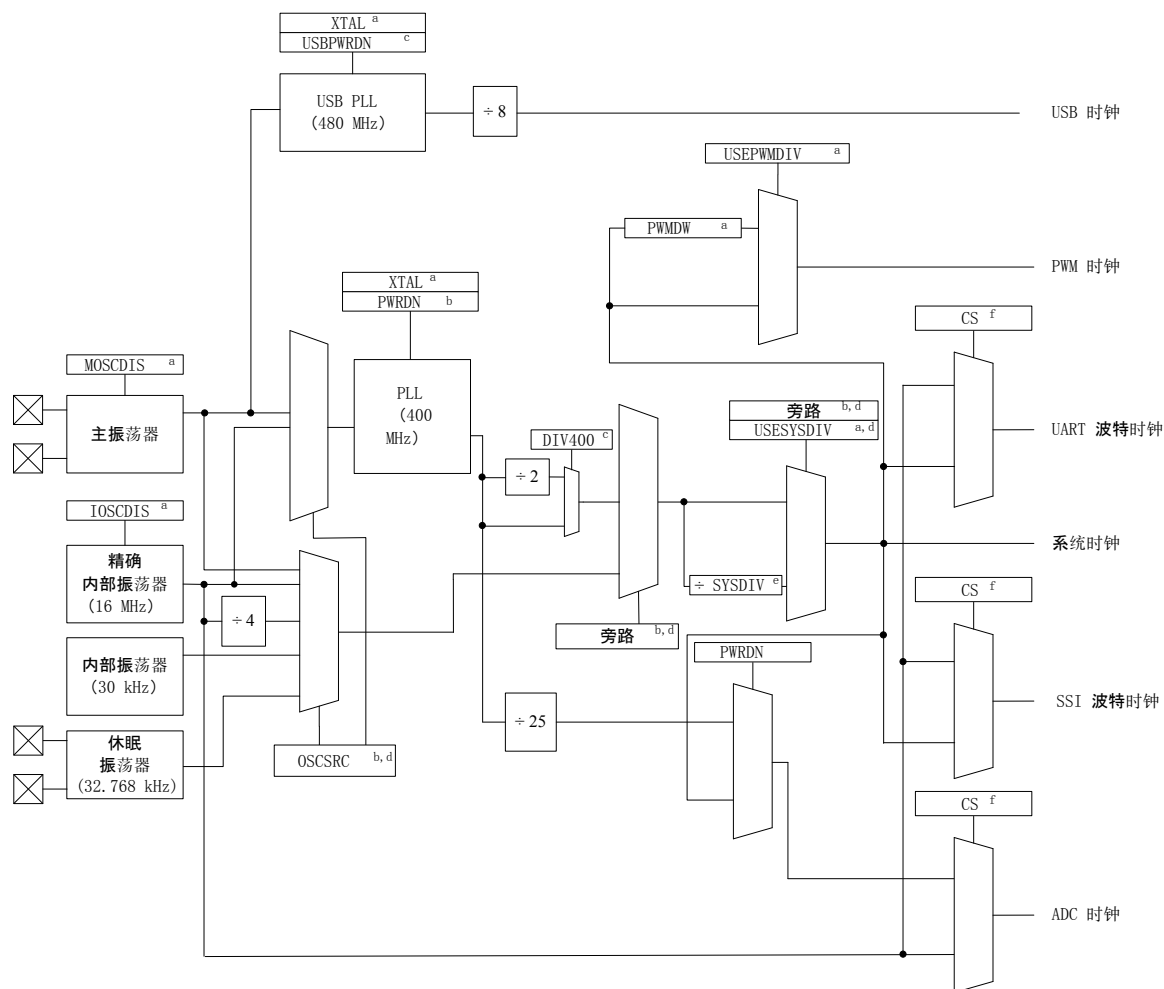
- 睡眠和深度睡眠模式中的时钟源
- 源自PLL或其它时钟源的系统时钟
- 振荡器和PLL的使能/禁止
- 时钟分频
- 晶振输入选择

**重要：** 写 RCC 寄存器后再写 RCC2 寄存器。如果需要对 RCC 寄存器进行后续写操作，则应在写 RCC 寄存器之后和写 RCC2 寄存器之前安排另一次寄存器访问。

图 5-5 所示为主时钟树逻辑。外设模块由系统时钟信号驱动并且可以独立启用/禁止。如果 PLL 被禁用，可从 PIOSC 或系统时钟中选择 ADC 时钟信号，如果 PLL 被启用，可从分频至 16 MHz 的 PLL 输出中选择 ADC 时钟信号。PWM 时钟信号是系统时钟的同步分频，这样可以向 PWM 电路提供更宽的范围（通过设置 RCC 中的 PWMDIV）。

**注意：** 如果 ADC 模块未使用 PIOSC 作为时钟源，系统时钟必须至少为 16 MHz。当 USB 模块运行时，MOSC 必须为时钟源（无论是否使用 PLL），并且系统时钟必须至少为 30 MHz。

图 5-5. 主时钟树



- 注意:
- 由 RCC 寄存器的位/域提供控制。
  - 由 RCC 寄存器的位/域提供控制，或者如果 RCC2 寄存器位 USERCC2 置位，由 RCC2 寄存器的位/域提供控制。
  - 由 RCC2 寄存器的位/域提供控制。
  - 当进入深度睡眠模式时，也可以由 DSLPCLKCFG 控制。
  - 由 RCC 寄存器的 SYSDIV 域提供控制，如果 USERCC2 位置位，由 RCC2 寄存器的 SYSDIV2 域提供控制；如果 USERCC2 和 DIV400 位都置位，则由[SYSDIV2,SYSDIV2LSB]域提供控制。
  - 由 UARTCC、SSICC 和 ADCCC 寄存器域提供控制。

### 通信时钟源

在上述的主时钟树之外，UART 和 SSI 模块在外设寄存器映射中都拥有时钟控制寄存器（偏移量 0xFC8），用来选择模块波特率时钟的时钟源。用户可以在作为波特率时钟默认源的系统时钟和 PIOSC 之间进行选择。请注意在使用 PIOSC 作为波特率时，可能有特别注意事项。更多信息，请查看描述模块运行章节的时钟控制寄存器说明。

### 使用 SYSDIV 和 SYSDIV2 域

在RCC寄存器中，SYSDIV域规定了哪个分频用于产生系统时钟，该系统时钟源自PLL输出或振荡器源（取决于该寄存器中BYPASS位的配置）。当使用PLL时，在应用分频前400MHz的VCO频率先被2

分频。表 5-4 显示了使用 PLL (BYPASS=0) 或另一个时钟源 (BYPASS=1) 时, SYSDIV 编码如何影响系统时钟频率。分频值等于 SYSDIV 编码加 1。表 5-3 在 210 页列出了可能的时钟源。

表 5-4. 使用 SYSDIV 域的可能的系统时钟频率

SYSDIV	分频系数	频率 (BYPASS=0)	频率 (BYPASS=1)	StellarisWare® 参数 <sup>a</sup>
0x0	/1	保留	时钟源频率/2	SYSCCTL_SYSDIV_1 <sup>b</sup>
0x1	/2	保留	时钟源频率/2	SYSCCTL_SYSDIV_2
0x2	/3	66.67 MHz	时钟源频率/3	SYSCCTL_SYSDIV_3
0x3	/4	50 MHz	时钟源频率/4	SYSCCTL_SYSDIV_4
0x4	/5	40 MHz	时钟源频率/5	SYSCCTL_SYSDIV_5
0x5	/6	33.33 MHz	时钟源频率/6	SYSCCTL_SYSDIV_6
0x6	/7	28.57 MHz	时钟源频率/7	SYSCCTL_SYSDIV_7
0x7	/8	25 MHz	时钟源频率/8	SYSCCTL_SYSDIV_8
0x8	/9	22.22 MHz	时钟源频率/9	SYSCCTL_SYSDIV_9
0x9	/10	20 MHz	时钟源频率/10	SYSCCTL_SYSDIV_10
0xA	/11	18.18 MHz	时钟源频率/11	SYSCCTL_SYSDIV_11
0xB	/12	16.67 MHz	时钟源频率/12	SYSCCTL_SYSDIV_12
0xC	/13	15.38 MHz	时钟源频率/13	SYSCCTL_SYSDIV_13
0xD	/14	14.29 MHz	时钟源频率/14	SYSCCTL_SYSDIV_14
0xE	/15	13.33 MHz	时钟源频率/15	SYSCCTL_SYSDIV_15
0xF	/16	12.5 MHz (默认)	时钟源频率/16	SYSCCTL_SYSDIV_16

a. 该参数在函数中使用, 例如 Stellaris 外设驱动库中的 SysCtlClockSet()。

b. SYSCCTL\_SYSDIV\_1 不会置位 USESYSYSDIV 位。所以不启用 PLL 而使用该参数的结果是系统时钟与时钟源频率相同。

RCC2 寄存器中的 SYSDIV2 域比 RCC 寄存器中的 SYSDIV 域多 2 位, 这样便可能进行更高地分频 (最高 64 分频), 同时深度睡眠功耗也因允许有更低的系统时钟频率而被改善。当使用 PLL 时, 在应用分频前 400MHz 的 VCO 频率先被 2 分频。分频值等于 SYSDIV2 编码加 1。表 5-5 显示了使用 PLL (BYPASS2=0) 或另一个时钟源 (BYPASS2=1) 时, SYSDIV2 编码如何影响系统时钟频率。表 5-3 在 210 页列出了可能的时钟源。

表 5-5. 使用 SYSDIV2 域的可能的系统时钟频率

SYSDIV2	分频系数	频率 (BYPASS2=0)	频率 (BYPASS2=1)	StellarisWare 参数 <sup>a</sup>
0x00	/1	保留	时钟源频率/2	SYSCCTL_SYSDIV_1 <sup>b</sup>
0x01	/2	保留	时钟源频率/2	SYSCCTL_SYSDIV_2
0x02	/3	66.67 MHz	时钟源频率/3	SYSCCTL_SYSDIV_3
0x03	/4	50 MHz	时钟源频率/4	SYSCCTL_SYSDIV_4
0x04	/5	40 MHz	时钟源频率/5	SYSCCTL_SYSDIV_5
...	...	...	...	...
0x09	/10	20 MHz	时钟源频率/10	SYSCCTL_SYSDIV_10
...	...	...	...	...
0x3F	/64	3.125 MHz	时钟源频率/64	SYSCCTL_SYSDIV_64

a. 该参数在函数中使用, 例如 Stellaris 外设驱动库中的 SysCtlClockSet()。

b. SYSCCTL\_SYSDIV\_1 不会置位 USESYSYSDIV 位。所以不启用 PLL 而使用该参数的结果是系统时钟与时钟源频率相同。

为了使用 PLL 时能有更多的频率选择, 提供了 DIV400 位和 SYSDIV2LSB 位。当 DIV400 位置位时, 第 22 位变成了 SYSDIV2 的 LSB。此时, 分频值等于 SYSDIV2 编码附带 SYSDIV2LSB 再加 1。表 5-6 显

示了DIV400置位后的频率选择。当DIV400被清零后，SYSDIV2LSB将被忽略，系统时钟频率仍然取决于表 5-5 在 212页所示的频率值。

表 5-6. 当DIV400=1时可能的系统时钟频率示例

SYSDIV2	SYSDIV2LSB	分频系数	频率 (BYPASS2=0) <sup>a</sup>	StellarisWare 参数 <sup>b</sup>
0x00	保留	/2	保留	-
0x01	0	/3	保留	-
	1	/4	保留	-
0x02	0	/5	80 MHz	SYSCCTL_SYSDIV_2_5
	1	/6	66.67 MHz	SYSCCTL_SYSDIV_3
0x03	0	/7	保留	-
	1	/8	50 MHz	SYSCCTL_SYSDIV_4
0x04	0	/9	44.44 MHz	SYSCCTL_SYSDIV_4_5
	1	/10	40 MHz	SYSCCTL_SYSDIV_5
...	...	...	...	...
0x3F	0	/127	3.15 MHz	SYSCCTL_SYSDIV_63_5
	1	/128	3.125 MHz	SYSCCTL_SYSDIV_64

a. 注意 DIV400 和 SYSDIV2LSB 只有当 BYPASS2 = 0 时才有效。

b. 该参数在函数中使用，例如 Stellaris 外设驱动库中的 SysCtlClockSet()。

### 5.2.5.3 精确内部振荡器操作 (PIOSC)

微控制器上电后运行 PIOSC。如果希望运行其它时钟源，PIOSC 在用于内部功能时必须保持启用。仅可以在深度睡眠模式期间禁用 PIOSC。它可以通过置位RCC寄存器中的IOSCDIS位掉电。

在室温环境下，PIOSC 可以产生一个  $\pm 1\%$  精度的 16 MHz 时钟。在整个扩展温度范围内，精度是  $\pm 3\%$ 。在工厂，PIOSC 被设置为室温 16 MHz。但是，在其它的电压或温度条件下，该频率可以通过使用软件用下面的三种方法来调整。

- 默认校准：在精确内部振荡器校准 (PIOSCCAL) 寄存器中清零UTEN位并置位UPDATE位。
- 用户自定义校准：用户可以编程UT值来调整PIOSC频率。随着UT值的增加，生成的周期也增加。为了提交一个新的UT值，首先要置位UTEN位，接着编程UT域，然后置位UPDATE位。该调整在几个时钟周期内完成，然后频率突变。
- 使用具有正在工作的 32.768-kHz 时钟源的休眠模块进行自动校准：置位PIOSCCAL寄存器中的CAL位；校准的结果显示在精确内部振荡器统计 (PIOSCSTAT) 寄存器的 RESULT 域中。校准完成后，使用CT域中返回的调整值来调整PIOSC。

### 5.2.5.4 用于主振荡器的晶振配置 (MOSC)

主振荡器支持的晶振值：从 4 到 25 MHz。

RCC 寄存器中的 XTAL 位（见 236页）描述了可用的晶体选择和默认的编程值。

软件根据晶振值来配置RCC寄存器的XTAL域。如果设计中使用PLL，那么XTAL域中的值会从内部转换到PLL设置中。

### 5.2.5.5 主 PLL 频率配置

主PLL在上电复位期间默认为禁止，如果需要可在稍后通过软件使能。软件指定输出分频来设置系统时钟频率，同时启用主 PLL 来驱动输出。PLL 以 400 MHz 的频率工作，但是在输出分频应用前将先被二分频，除非 RCC2 寄存器中的 DIV400 位置位。

要把 PIOSC 配置为主 PLL 的时钟源，可将运行模式时钟配置 2 (RCC2) 寄存器的 OSCRC2 域编程为 0x1。

如果主振荡器给主 PLL 提供了时钟基准，软件可以从 PLL 频率 n (PLLFRQn) 寄存器（请参考 252 页）中使用由硬件提供的转换（该转换通常用于对 PLL 进行编程）。内部转换可提供 ± 1% 的目标 PLL VCO 频率。表 25-10 在 1298 页显示了实际的 PLL 频率和给定晶振选择的误差。

在运行模式时钟配置 (RCC) 寄存器（请参考 236 页）中的晶振值域 (XTAL) 描述了可用的晶振选择和默认的 PLLFRQn 寄存器编程。只要 XTAL 域改变，新的设置就会被转换同时内部 PLL 设置被更新。

### 5.2.5.6 USB PLL 频率配置

USB PLL 在上电复位期间默认为禁止，可在稍后通过软件使能。为实现适当的 USB 功能，USB PLL 必须被启用运行。主振荡器是 USB PLL 的唯一时钟参考。通过清零 RCC2 寄存器的 USBPWRDN 位来启用 USB PLL。RCC 寄存器的 XTAL 位域 (晶振值) 描述了可用的晶振选择。为了正确的生成 USB 时钟，主振荡器必须连接到下面的晶振值：5、6、8、10、12、16、18、20、24 或 25 MHz。只有这些晶振能提供与 USB 时序规范一致的 USB PLL VCO 频率。

### 5.2.5.7 PLL 模式

每种 PLL 都有两种操作模式：正常模式和掉电模式

- 正常模式：PLL 倍频输入时钟参考并驱动输出。
- 掉电模式：大部分 PLL 内部电路被禁止并且 PLL 不再驱动输出。

使用 RCC/RCC2 寄存器的域来编程 PLL 模式（见 236 页和 243 页）。

### 5.2.5.8 PLL 操作

如果 PLL 配置改变，PLL 输出频率是不稳定的，直到重新集中 (重新锁定) 到一个新的设定为止。配置更改和重新锁定之间的时间是  $T_{\text{READY}}$ （请参考表 25-9 在 1298 页）。在重新锁定期间，受影响的 PLL 不可用作时钟参考。软件可以查询 PLL 状态 (PLLSTAT) 寄存器的 LOCK 位以确定 PLL 锁定的时间。

PLL 可通过以下方法之一进行更改：

- 更改为 RCC 寄存器的 XTAL 值 - 写入同样的值不会引起重新锁定。
- PLL 从掉电模式变为正常模式。

使用根据系统时钟计时的计数器测量  $T_{\text{READY}}$  的要求。如果 PLL 上电，递减计数器初值设为 0x200。如果在 PLLFRQn 寄存器中 M 或 N 值发生改变，则将计数器设为 0xC0。此时要提供硬件来保持 PLL 不被用作系统时钟，直到上述更改完成并满足  $T_{\text{READY}}$  条件为止。用户要确保在 RCC/RCC2 寄存器切换到使用 PLL 之前必须有一个稳定的时钟源 (像主振荡器那样)。

如果主 PLL 启用并且系统时钟一步切换到使用 PLL，系统控制硬件会继续使用 RCC/RCC2 寄存器选择的振荡器作为微控制器的时钟，直到主 PLL 稳定 (满足  $T_{\text{READY}}$  时间) 为止才更改到 PLL。软件可以使用很多方法来确保系统由主 PLL 提供时钟，包括周期性的检测原始中断状态 (RIS) 寄存器的 PLLLRIS 位，以及启用 PLL 锁定中断。

USB PLL 在锁定 ( $T_{\text{READY}}$ ) 期间是不受保护的，软件必须确保在使用该接口前 USB PLL 已经锁定。软件可以使用很多方法来确保  $T_{\text{READY}}$  周期已经过去，包括周期性的检测原始中断状态 (RIS) 寄存器的 USBPLLRIS 位，以及启用 USB PLL 锁定中断。

### 5.2.5.9 主振荡器校验电路

时钟控制包括能确保主振荡器工作在合适频率的电路。如果频率在相连晶振允许的频带范围外，该电路会监测主振荡器频率和信号。

检测电路通过使用主振荡器控制 (MOSCCTL) 寄存器的 CVAL 位来启用。如果该电路启用并且检测到错误，并且如果 MOSCCTL 寄存器中的 MOSCIM 位清零，那么硬件会执行下面的序列：

1. 复位原因 (RESC) 寄存器的 MOSCFAIL 位被置位。
2. 系统时钟由主振荡器切换到 PIOSC。
3. 一个内部上电复位被启动。
4. 复位不再有效，处理器在复位序列期间直接执行 NMI 处理程序。

如果 MOSCCTL 寄存器中的 MOSCIM 位置位，那么硬件会执行下面的序列：

1. 系统时钟由主振荡器切换到 PIOSC。
2. RIS 寄存器中的 MOFRIS 位置位以指示 MOSC 故障。

### 5.2.6 系统控制

为了节省功耗，外设专用的 RCGCx、SCGCx 和 DCGCx 寄存器（例如，RCGCWD）分别控制着微控制器在运行模式、睡眠模式和深度睡眠模式时该外设或系统模块的时钟门控逻辑。这些寄存器位于系统控制寄存器映射中，分别从偏移量 0x600、0x700 和 0x800 处开始。在访问任何模块寄存器前，在 RCGC 寄存器中启用外设模块时钟后必须有 3 个系统时钟的延迟。

**重要：** 要支持传统软件，RCGCn、SCGCn 和 DCGCn 寄存器在偏移量 0x100 - 0x128 时可用。对所有这些传统寄存器进行写操作时，也将对外设专用 RCGCx、SCGCx 和 DCGCx 寄存器中的相应位执行写操作。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。建议新软件使用新寄存器并且不依赖于传统操作。

如果软件使用外设专用寄存器对传统外设（如 TIMER0）执行写操作，则写操作会产生正确操作，但是该位的值不在传统寄存器中得到反映。通过对传统寄存器的写操作更改的任何位都可以通过对传统寄存器的读操作进行正确回读。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

对于微控制器有四种级别的操作，定义如下：

- 运行模式
- 睡眠模式
- 深度睡眠模式
- 休眠模式

下面章节将详细描述不同的模式。

小心 – 如果 **CortexM4F** 调试访问端口 (**DAP**) 已经启用，并且设备从低功耗睡眠或深度睡眠模式中唤醒，内核可能会在所有外设的时钟恢复到运行模式配置前开始执行代码。**DAP**通常由软件工具启用，以便在调试或**Flash**编程时访问**JTAG**或**SWD**接口。如果这种情况发生，当软件访问一个带有无效时钟的外设时会触发一次硬件错误。

软件延时循环可用在从**WFI**指令(等待中断)唤醒系统的中断程序开始处。这样可以延迟执行可能会产生错误的访问外设寄存器的指令。对于产品软件来说，该循环可以去除，因为在正常执行期间**DAP**几乎不可能被启用。

由于**DAP**默认为禁止(上电复位)，所以用户也可以让设备按上电周期运行。**DAP**只有通过 **JTAG** 或 **SWD** 接口才能启用。

### 5.2.6.1 运行模式

在运行模式，微控制器主动执行代码。运行模式提供了处理器和所有目前被外设专用 **RCGC** 寄存器启用的外设的正常操作。系统时钟可以是包括 **PLL** 在内的任何可用的时钟源。

### 5.2.6.2 睡眠模式

在睡眠模式，运行中的外设时钟频率不变，但是处理器和存储器子系统不使用时钟，所以不再执行代码。睡眠模式是通过 **Cortex-M4F** 内核执行一条 **WFI** (等待中断) 指令来进入的。系统中任何正确配置的中断事件都可以将处理器带回到运行模式。其它详细信息见“电源管理”在 110页。

当自动时钟门控启用时，外设专用 **SCGC** 寄存器启用的外设时钟被使用 (请参考 **RCC** 寄存器)；当自动时钟门控禁用时，外设专用 **RCGC** 寄存器启用的外设时钟被使用。系统时钟的源和频率与运行模式期间是一样的。

### 5.2.6.3 深度睡眠模式

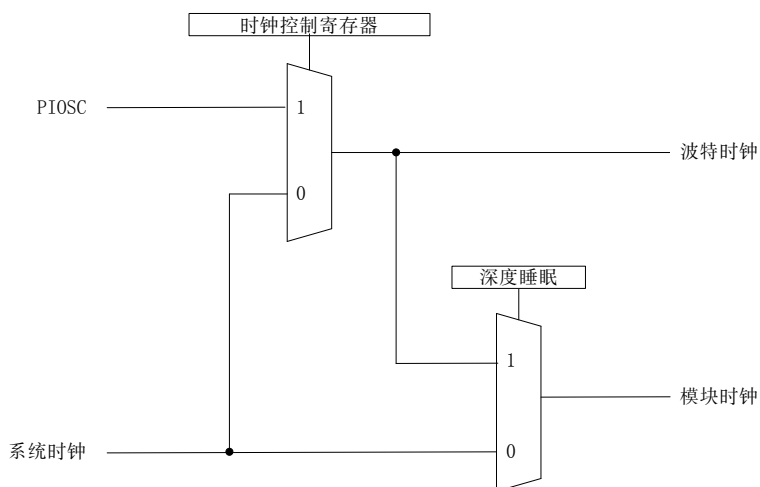
在深度睡眠模式中，除了正在停止的处理器时钟之外，有效外设的时钟频率可以改变 (由运行模式的时钟配置决定)。中断可以让微控制器从睡眠模式返回到运行模式；代码请求可以进入睡眠模式。要进入深度睡眠模式，首先置位系统控制 (**SYSCCTRL**) 寄存器的 **SLEEPDEEP** 位 (见 160 页)，然后执行一条 **WFI** 指令。系统中任何正确配置的中断事件都可以将处理器带回到运行模式。其它详细信息见“电源管理”在 110页。

该 **CortexM4F** 处理器内核和存储器子系统在深度睡眠模式中不计。当自动时钟门控启用时，外设专用 **DSCGC** 寄存器启用的外设时钟被使用 (请参考 **RCC** 寄存器)；当自动时钟门控禁用时，外设专用 **RCGC** 寄存器启用的外设时钟被使用。系统时钟源在 **DSLPCCLKCFG** 寄存器中规定。当 **DSLPCCLKCFG** 寄存器被使用时，内部振荡器源被上电，如果有必要，其它时钟会被掉电。如果执行 **WFI** 指令时 **PLL** 正在运行，硬件会让 **PLL** 掉电并改写 **RCC/RCC2** 寄存器中的 **SYSDIV** 域，更改值由 **DSLPCCLKCFG** 寄存器的 **DSDIVORIDE** 设置决定，分别为 /16 或 /64。当深度睡眠退出事件发生时，硬件会把系统时钟带回到深度睡眠模式开始时的源和频率，然后启用在深度睡眠期间停止的时钟。如果 **PIOSC** 被用作 **PLL** 的参考时钟源，在深度睡眠期间它会继续提供时钟。请参考 247页。

要实现尽可能低的深度睡眠功耗，以及无需为时钟更改而重新配置外设即可从外设唤醒处理器的能力，一些通信模块在模式寄存器空间中的偏移量 **0xFC8** 处提供了时钟控制寄存器。时钟控制寄存器中 **CS** 域允许用户选择 **PIOSC** 作为模块波特率时钟的时钟源。微控制器进入深度睡眠模式时，**PIOSC** 也成为模块时钟的源，允许发送和接收 **FIFO** 在该部分处于深度睡眠时继续操作。图 5-6 在 217页显示了如何选择时钟。



图 5-6. 模块时钟选择



#### 5.2.6.4 休眠模式

这种模式下，微控制器主要功能部件的电源关断，只有休眠模块的电路有效需要一个外部唤醒事件或 RTC 事件来使微控制器回到运行模式。该 CortexM4F 处理器和休眠模块之外的外设看见一个正常的“上电”序列，处理器启动运行代码。软件通过检查休眠模块寄存器可以确定微控制器是否已经从休眠模式重新启动。关于休眠模式操作的更多信息，请参考“休眠模块”在 475 页。

### 5.3 初始化及配置

PLL 的配置可通过直接对 RCC/RCC2 执行写操作来实现。如果 RCC2 寄存器正被使用，那么必须置位 USERCC2 位，合适的 RCC2 位/域被使用。成功改变基于 PLL 的系统时钟需要如下步骤：

1. 通过置位 RCC 寄存器的 BYPASS 位并清零 USESYS 位来旁路 PLL 和系统时钟分频器，从而配置微控制器在“原始”时钟源运行，在切换系统时钟到 PLL 前允许新的 PLL 配置有效。
2. 选择晶振值 (XTAL) 和振荡器源 (OSCSRC)，清零 RCC/RCC2 的 PWRDN 位。设置 XTAL 域可以自动为晶振提供有效的 PLL 配置数据，清零 PWRDN 位可以给 PLL 及其输出供电并将其启用。
3. 在 RCC/RCC2 中选择需要的系统分频器 (SYSDIV) 并将 RCC 中的 USESYS 位置位。SYSDIV 域决定了微控制器的系统频率。
4. 通过检测原始中断状态 (RIS) 寄存器的 PLLRIS 位来等待 PLL 被锁定。
5. 通过清零 RCC/RCC2 的 BYPASS 位来启用 PLL。

### 5.4 寄存器映射

表 5-7 在 218 页 列出了按功能分组的系统控制寄存器。列出的偏移量是相对于 0x400F.E000 系统控制基址而言的寄存器地址的 16 进制增量。

**注意：** 系统控制寄存器空间中未被使用的空间都是为将来或内部使用保留的。软件不应该调整任何保留的存储器地址。

系统控制寄存器空间中定义的其它 Flash 和 ROM 寄存器在“内部存储器”在 503 页中描述。

表 5-7. 系统控制 寄存器映射

偏移量	名称	类型	复位	描述	见页面
系统控制寄存器					
0x000	DID0	RO	-	器件标识寄存器 0	223
0x004	DID1	RO	-	器件标识寄存器 1	225
0x030	PBORCTL	R/W	0x0000.0000	掉电复位控制寄存器	227
0x050	RIS	RO	0x0000.0000	原始中断状态寄存器	228
0x054	IMC	R/W	0x0000.0000	中断屏蔽控制寄存器	230
0x058	MISC	R/W1C	0x0000.0000	屏蔽中断状态和清除寄存器	232
0x05C	RESC	R/W	-	复位原因寄存器	234
0x060	RCC	R/W	0x078E.3D51	运行模式时钟配置寄存器	236
0x06C	GPIOHBCTL	R/W	0x0000.7E00	GPIO 高性能总线控制寄存器	240
0x070	RCC2	R/W	0x07C0.6810	运行模式时钟配置寄存器 2	243
0x07C	MOSCCTL	R/W	0x0000.0000	主振荡器控制寄存器	246
0x144	DSLPLCKCFG	R/W	0x0780.0000	深度睡眠时钟配置寄存器	247
0x14C	SYSPROP	RO	0x0000.1D31	系统属性寄存器	249
0x150	PIOSCCAL	R/W	0x0000.0000	精确内部振荡器校准寄存器	250
0x154	PIOSCSTAT	RO	0x0000.0040	精确内部振荡器统计寄存器	251
0x160	PLLREQ0	RO	0x0000.0032	PLL 频率寄存器 0	252
0x164	PLLREQ1	RO	0x0000.0001	PLL 频率寄存器 1	253
0x168	PLLSTAT	RO	0x0000.0000	PLL 状态寄存器	254
0x300	PPWD	RO	0x0000.0003	看门狗定时器外设存在寄存器	255
0x304	PPTIMER	RO	0x0000.003F	16/32 位通用定时器外设存在寄存器	256
0x308	PPGPIO	RO	0x0000.3FFF	通用输入/输出外设存在寄存器	258
0x30C	PPDMA	RO	0x0000.0001	微型直接存储器访问外设存在寄存器	261
0x314	PPHIB	RO	0x0000.0001	休眠外设存在寄存器	262
0x318	PPUART	RO	0x0000.00FF	通用异步收发器外设存在寄存器	263
0x31C	PPSSI	RO	0x0000.000F	同步串行接口外设存在寄存器	265
0x320	PPI2C	RO	0x0000.003F	内部集成电路外设存在寄存器	266
0x328	PPUSB	RO	0x0000.0001	通用串行总线外设存在寄存器	268
0x334	PPCAN	RO	0x0000.0003	控制器局域网外设存在寄存器	269
0x338	PPADC	RO	0x0000.0003	模数转换器外设存在寄存器	270
0x33C	PPACMP	RO	0x0000.0001	模拟比较器外设存在寄存器	271
0x340	PPPWM	RO	0x0000.0003	脉宽调制器外设存在寄存器	272
0x344	PPQEI	RO	0x0000.0003	正交编码器接口外设存在寄存器	273

表 5-7. 系统控制 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0x358	PPEEPROM	RO	0x0000.0001	EEPROM 外设存在寄存器	274
0x35C	PPWTIMER	RO	0x0000.003F	32/64 位宽通用定时器外设存在寄存器	275
0x500	SRWD	R/W	0x0000.0000	看门狗定时器软件复位寄存器	277
0x504	SRTIMER	R/W	0x0000.0000	16/32 位通用定时器软件复位寄存器	278
0x508	SRGPIO	R/W	0x0000.0000	通用输入/输出软件复位寄存器	280
0x50C	SRDMA	R/W	0x0000.0000	微型直接存储器访问软件复位寄存器	283
0x514	SRHIB	R/W	0x0000.0000	休眠软件复位寄存器	284
0x518	SRUART	R/W	0x0000.0000	通用异步收发器软件复位寄存器	285
0x51C	SRSSI	R/W	0x0000.0000	同步串行接口软件复位寄存器	287
0x520	SRI2C	R/W	0x0000.0000	内部集成电路软件复位寄存器	289
0x528	SRUSB	R/W	0x0000.0000	通用串行总线软件复位寄存器	291
0x534	SRCAN	R/W	0x0000.0000	控制器局域网软件复位寄存器	292
0x538	SRADC	R/W	0x0000.0000	模数转换器软件复位寄存器	293
0x53C	SRACMP	R/W	0x0000.0000	模数比较器软件复位寄存器	294
0x540	SRPWM	R/W	0x0000.0000	脉宽调制器软件复位寄存器	295
0x544	SRQEI	R/W	0x0000.0000	正交编码器接口软件复位寄存器	296
0x55C	SRWTIMER	R/W	0x0000.0000	32/64 位宽通用定时器软件复位寄存器	297
0x600	RCGCWD	R/W	0x0000.0000	看门狗定时器运行模式时钟门控控制寄存器	299
0x604	RCGCTIMER	R/W	0x0000.0000	16/32 位通用定时器运行模式时钟门控控制寄存器	300
0x608	RCGCGPIO	R/W	0x0000.0000	通用输入/输出运行模式时钟门控控制寄存器	302
0x60C	RCGCDMA	R/W	0x0000.0000	微型直接存储器访问运行模式时钟门控控制寄存器	305
0x614	RCGCHIB	R/W	0x0000.0001	休眠运行模式时钟门控控制寄存器	306
0x618	RCGCUART	R/W	0x0000.0000	通用异步收发器运行模式时钟门控控制寄存器	307
0x61C	RCGCSSI	R/W	0x0000.0000	同步串行接口运行模式时钟门控控制寄存器	309
0x620	RCGCI2C	R/W	0x0000.0000	内部集成电路运行模式时钟门控控制寄存器	311
0x628	RCGCUSB	R/W	0x0000.0000	通用串行总线运行模式时钟门控控制寄存器	313
0x634	RCGCCAN	R/W	0x0000.0000	控制器局域网运行模式时钟门控控制寄存器	314
0x638	RCGCADC	R/W	0x0000.0000	模数转换器运行模式时钟门控控制寄存器	315
0x63C	RCGCACMP	R/W	0x0000.0000	模拟比较器运行模式时钟门控控制寄存器	316
0x640	RCGCPWM	R/W	0x0000.0000	脉宽调制器运行模式时钟门控控制寄存器	317
0x644	RCGCQEI	R/W	0x0000.0000	正交编码器接口运行模式时钟门控控制寄存器	318
0x65C	RCGCWTIMER	R/W	0x0000.0000	32/64 位宽通用定时器运行模式时钟门控控制寄存器	319
0x700	SCGCWD	R/W	0x0000.0000	看门狗定时器睡眠模式时钟门控控制寄存器	321

表 5-7. 系统控制 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0x704	SCGCTIMER	R/W	0x0000.0000	16/32 位通用定时器睡眠模式时钟门控控制寄存器	322
0x708	SCGCGPIO	R/W	0x0000.0000	通用输入/输出睡眠模式时钟门控控制寄存器	324
0x70C	SCGCDMA	R/W	0x0000.0000	微型直接存储器访问睡眠模式时钟门控控制寄存器	327
0x714	SCGCHIB	R/W	0x0000.0001	休眠睡眠模式时钟门控控制寄存器	328
0x718	SCGCUART	R/W	0x0000.0000	通用异步收发器睡眠模式时钟门控控制寄存器	329
0x71C	SCGCSSI	R/W	0x0000.0000	同步串行接口睡眠模式时钟门控控制寄存器	331
0x720	SCGCI2C	R/W	0x0000.0000	内部集成电路睡眠模式时钟门控控制寄存器	333
0x728	SCGUSB	R/W	0x0000.0000	通用串行总线睡眠模式时钟门控控制寄存器	335
0x734	SCGCCAN	R/W	0x0000.0000	控制器局域网睡眠模式时钟门控控制寄存器	336
0x738	SCGCADC	R/W	0x0000.0000	模数转换器睡眠模式时钟门控控制寄存器	337
0x73C	SCGCACMP	R/W	0x0000.0000	模拟比较器睡眠模式时钟门控控制寄存器	338
0x740	SCGCPWM	R/W	0x0000.0000	脉宽调制器睡眠模式时钟门控控制寄存器	339
0x744	SCGCQEI	R/W	0x0000.0000	正交编码器接口睡眠模式时钟门控控制寄存器	340
0x75C	SCGCWTIMER	R/W	0x0000.0000	32/64 位宽通用定时器睡眠模式时钟门控控制寄存器	341
0x800	DCGCWD	R/W	0x0000.0000	看门狗定时器深度睡眠模式时钟门控控制寄存器	343
0x804	DCGCTIMER	R/W	0x0000.0000	16/32 位通用定时器深度睡眠模式时钟门控控制寄存器	344
0x808	DCGCGPIO	R/W	0x0000.0000	通用输入/输出深度睡眠模式时钟门控控制寄存器	346
0x80C	DCGCDMA	R/W	0x0000.0000	微型直接存储器访问深度睡眠模式时钟门控控制寄存器	349
0x814	DCGCHIB	R/W	0x0000.0001	休眠深度睡眠模式时钟门控控制寄存器	350
0x818	DCGCUART	R/W	0x0000.0000	通用异步收发器深度睡眠模式时钟门控控制寄存器	351
0x81C	DCGCSSI	R/W	0x0000.0000	同步串行接口深度睡眠模式时钟门控控制寄存器	353
0x820	DCGCI2C	R/W	0x0000.0000	内部集成电路深度睡眠模式时钟门控控制寄存器	355
0x828	DCGUSB	R/W	0x0000.0000	通用串行总线深度睡眠模式时钟门控控制寄存器	357
0x834	DCGCCAN	R/W	0x0000.0000	控制器局域网深度睡眠模式时钟门控控制寄存器	358
0x838	DCGCADC	R/W	0x0000.0000	模数转换器深度睡眠模式时钟门控控制寄存器	359
0x83C	DCGCACMP	R/W	0x0000.0000	模拟比较器深度睡眠模式时钟门控控制寄存器	360
0x840	DCGCPWM	R/W	0x0000.0000	脉宽调制器深度睡眠模式时钟门控控制寄存器	361
0x844	DCGCQEI	R/W	0x0000.0000	正交编码器接口深度睡眠模式时钟门控控制寄存器	362
0x85C	DCGCWTIMER	R/W	0x0000.0000	32/64 位宽通用定时器深度睡眠模式时钟门控控制寄存器	363
0x900	PCWD	R/W	0x0000.0003	看门狗定时器功率控制寄存器	365
0x904	PCTIMER	R/W	0x0000.003F	16/32 位通用定时器功率控制寄存器	366
0x908	PCGPIO	R/W	0x0000.7FFF	通用输入/输出功率控制寄存器	369
0x90C	PCDMA	R/W	0x0000.0001	微型直接存储器访问功率控制寄存器	373

表 5-7. 系统控制 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0x914	PCHIB	R/W	0x0000.0001	休眠功率控制寄存器	374
0x918	PCUART	R/W	0x0000.00FF	通用异步收发器功率控制寄存器	375
0x91C	PCSSI	R/W	0x0000.000F	同步串行接口功率控制寄存器	378
0x920	PCI2C	R/W	0x0000.003F	内部集成电路功率控制寄存器	380
0x928	PCUSB	R/W	0x0000.0001	通用串行总线功率控制寄存器	382
0x934	PCCAN	R/W	0x0000.0003	控制器局域网功率控制寄存器	383
0x938	PCADC	R/W	0x0000.0003	模数转换器功率控制寄存器	384
0x93C	PCACMP	R/W	0x0000.0001	模拟比较器功率控制寄存器	385
0x940	PCPWM	R/W	0x0000.0003	脉宽调制器功率控制寄存器	386
0x944	PCQEI	R/W	0x0000.0003	正交编码器接口功率控制寄存器	387
0x95C	PCWTIMER	R/W	0x0000.0000	32/64 位宽通用定时器功率控制寄存器	388
0xA00	PRWD	R/W	0x0000.0000	看门狗定时器外设就绪寄存器	391
0xA04	PRTIMER	R/W	0x0000.0000	16/32 位通用定时器外设就绪寄存器	392
0xA08	PRGPIO	R/W	0x0000.0000	通用输入/输出外设就绪寄存器	394
0xA0C	PRDMA	R/W	0x0000.0000	微型直接存储器访问外设就绪寄存器	397
0xA14	PRHIB	R/W	0x0000.0001	休眠外设就绪寄存器	398
0xA18	PRUART	R/W	0x0000.0000	通用异步收发器外设就绪寄存器	399
0xA1C	PRSSI	R/W	0x0000.0000	同步串行接口外设就绪寄存器	401
0xA20	PRI2C	R/W	0x0000.0000	内部集成电路外设就绪寄存器	403
0xA28	PRUSB	R/W	0x0000.0000	通用串行总线外设就绪寄存器	405
0xA34	PRCAN	R/W	0x0000.0000	控制器局域网外设就绪寄存器	406
0xA38	PRADC	R/W	0x0000.0000	模数转换器外设就绪寄存器	407
0xA3C	PRACMP	R/W	0x0000.0000	模拟比较器外设就绪寄存器	408
0xA40	PRPWM	R/W	0x0000.0000	脉宽调制器外设就绪寄存器	409
0xA44	PRQEI	R/W	0x0000.0000	正交编码器接口外设就绪寄存器	410
0xA5C	PRWTIMER	R/W	0x0000.0000	32/64 位宽通用定时器外设就绪寄存器	411
系统控制传统寄存器					
0x008	DC0	RO	0x007F.007F	器件功能寄存器 0	413
0x010	DC1	RO	0x1333.2FFF	器件功能寄存器 1	415
0x014	DC2	RO	0x070F.F337	器件功能寄存器 2	418
0x018	DC3	RO	0xBFFF.FFFF	器件功能寄存器 3	420
0x01C	DC4	RO	0x0004.F1FF	器件功能寄存器 4	424
0x020	DC5	RO	0x0F30.00FF	器件功能寄存器 5	426

表 5-7. 系统控制 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0x024	DC6	RO	0x0000.0013	器件功能寄存器 6	428
0x028	DC7	RO	0xFFFF.FFFF	器件功能寄存器 7	429
0x02C	DC8	RO	0xFFFF.FFFF	器件功能寄存器 8	432
0x040	SRCR0	RO	0x0000.0000	软件复位控制寄存器 0	435
0x044	SRCR1	RO	0x0000.0000	软件复位控制寄存器 1	437
0x048	SRCR2	RO	0x0000.0000	软件复位控制寄存器 2	440
0x100	RCGC0	RO	0x0000.0040	运行模式时钟门控控制寄存器 0	442
0x104	RCGC1	RO	0x0000.0000	运行模式时钟门控控制寄存器 1	445
0x108	RCGC2	RO	0x0000.0000	运行模式时钟门控控制寄存器 2	448
0x110	SCGC0	RO	0x0000.0040	睡眠模式时钟门控控制寄存器 0	450
0x114	SCGC1	RO	0x0000.0000	睡眠模式时钟门控控制寄存器 1	452
0x118	SCGC2	RO	0x0000.0000	睡眠模式时钟门控控制寄存器 2	455
0x120	DCGC0	RO	0x0000.0040	深度睡眠模式时钟门控控制寄存器 0	457
0x124	DCGC1	RO	0x0000.0000	深度睡眠模式时钟门控控制寄存器 1	459
0x128	DCGC2	RO	0x0000.0000	深度睡眠模式时钟门控控制寄存器 2	462
0x190	DC9	RO	0x00FF.00FF	器件功能寄存器 9	464
0x1A0	NVMSTAT	RO	0x0000.0001	非易失性存储器信息寄存器	466

## 5.5 系统控制寄存器描述

所有给出的地址都是相对于 0x400F.E000 的系统控制基址而言的。“系统控制传统寄存器描述”在 412 页中列出了仅为支持传统软件而提供的寄存器。

## 寄存器 1: 器件标识寄存器 0 ( DID0 ) , 偏移量 0x000

本寄存器标识了微控制器的版本。每个微控制器由 DID0 寄存器中的 CLASS 域和 DID1 寄存器中的 PARTNO 域的组合值标识，具有唯一性。

### 器件标识寄存器 0 (DID0)

基址 0x400F.E000

偏移量 0x000

类型 RO, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留	VER			保留				CLASS							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MAJOR								MINOR							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
30:28	VER	RO	0x01	DID0版本 该域定义了 DID0 寄存器格式的版本。版本号是数字的。VER 域的值编码如下（其它所有编码保留）：  值 描述 0x1 DID0 寄存器格式的第二版。
27:24	保留	RO	0x08	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
23:16	CLASS	RO	0x05	器件分类 CLASS域的值识别内部设计，根据内部设计，所有微控制器的掩膜组在特定的生产线中产生。CLASS域的值随新产品线而改变，随fab过程的变化而改变(例如重新映射或收缩)，或者在任何MAJOR或MINOR域需要与之前微控制器有差异的地方改变。CLASS域的值如下编码（所有其它的编码保留）：  值 描述 0x05 Stellaris® Blizzard-class 微控制器
15:8	MAJOR	RO	-	主版本 这个域指定了微控制器的主版本号。主版本反映了设计的基本层的改变。主版本编号在器件型号中以字母表示（A代表第一版，B代表第二版，依此类推）。该域编码如下：  值 描述 0x0 版本 A（初值设备） 0x1 版本 B（第一个基本层版本） 0x2 版本 B（第二个基本层版本）  依此类推。

---

位/域	名称	类型	复位	描述
7:0	MINOR	RO	-	<p>次版本</p> <p>这个域指定了微控制器的次版本号。次版本反映了设计的金属层的改变。MAJOR 域改变时 MINOR 域的值重新设置。这个域的值是数字，编码如下：</p> <p>值 描述</p> <p>0x0 原始器件，或者一个主版本更新。</p> <p>0x1 第一个金属层改变。</p> <p>0x2 第二个金属层改变。</p> <p>依此类推。</p>



## 寄存器 2: 器件标识寄存器 1 ( DID1 ) , 偏移量 0x004

该寄存器标识了器件系列、器件号、温度范围、管脚数和封装类型。每个微控制器由 DID0 寄存器中的 CLASS 域和 DID1 寄存器中的 PARTNO 域的组合值标识, 具有唯一性。

### 器件标识寄存器 1 (DID1)

基址 0x400F.E000

偏移量 0x004

类型 RO, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VER				FAM				PARTNO							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PINCOUNT			保留				TEMP			PKG		ROHS	QUAL		
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	0	0	0	0	0	0	0	0	0	1	0	1	1	-	-

位/域	名称	类型	复位	描述
31:28	VER	RO	0x1	<p>DID1 版本</p> <p>该域定义了 DID1 寄存器格式的版本。版本号是数字的。VER 域的值编码如下 ( 其它所有编码保留 ) :</p> <p>值 描述</p> <p>0x0 最初的 DID1 寄存器格式定义, 表示 Stellaris LM3Snnn 器件。</p> <p>0x1 DID1 寄存器格式的第二版。</p>
27:24	FAM	RO	0x0	<p>系列</p> <p>该域提供了该器件在产品组中的 Stellaris 系列标识。该值编码如下 ( 所有其它编码保留 ) :</p> <p>值 描述</p> <p>0x0 Stellaris 系列微控制器, 即所有产品以 LM3S、LM4S 和 LM4F 作为外部器件号的开头。</p>
23:16	PARTNO	RO	0xC5	<p>器件号</p> <p>该域提供了该产品在系列中的器件号。显示的复位值指示 LM4F232H5QD 微控制器。</p>
15:13	PINCOUNT	RO	0x4	<p>封装管脚数</p> <p>该域规定了该器件封装的管脚数。该值编码如下 ( 所有其它编码保留 ) :</p> <p>值 描述</p> <p>0x0 28 管脚封装</p> <p>0x1 48 管脚封装</p> <p>0x2 100 管脚封装</p> <p>0x3 64 管脚封装</p> <p>0x4 144 管脚封装</p> <p>0x5 157 管脚封装</p>

位/域	名称	类型	复位	描述
12:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:5	TEMP	RO	0x1	<p>温度范围</p> <p>该域规定了器件的温度等级。该值编码如下 ( 所有其它编码保留 ) :</p> <p>值 描述</p> <p>0x0 商业温度范围 (0°C ~ 70°C)</p> <p>0x1 工业温度范围 (-40°C ~ 85°C)</p> <p>0x2 扩展温度范围 (-40°C ~ 105°C)</p>
4:3	PKG	RO	0x1	<p>封装类型</p> <p>该域规定封装类型。该值编码如下 ( 所有其它编码保留 ) :</p> <p>值 描述</p> <p>0x0 SOIC 封装</p> <p>0x1 LQFP 封装</p> <p>0x2 BGA 封装</p>
2	ROHS	RO	0x1	<p>RoHS 合规性</p> <p>该位说明了器件是否符合 RoHS 标准。值为 1 表示符合。</p>
1:0	QUAL	RO	-	<p>合格状态</p> <p>该域规定了器件的合格状态。该值编码如下 ( 所有其它编码保留 ) :</p> <p>值 描述</p> <p>0x0 工程样片(未经认证)</p> <p>0x1 试制产品 ( 未经认证 )</p> <p>0x2 完全合格</p>

### 寄存器 3: 掉电复位控制寄存器 ( PBORCTL ) , 偏移量 0x030

该寄存器负责控制初始上电复位后的复位条件。

#### 掉电复位控制寄存器 (PBORCTL)

基址 0x400FE000

偏移量 0x030

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															BORIOR	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	BORIOR	R/W	0	BOR中断或复位  值 描述 0 一次掉电事件引起了中断的产生并发送给中断控制器。 1 一次掉电事件引起了微控制器的复位。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

### 寄存器 4: 原始中断状态寄存器 ( RIS ) , 偏移量 0x050

该寄存器显示系统控制原始中断的状态。如果中断屏蔽控制 (IMC) 寄存器的相应位被置位, 那么中断会发送到中断控制器。向屏蔽中断状态和清除 (MISC) 寄存器的位写 1 可以清除相应的中断状态位。

#### 原始中断状态寄存器 (RIS)

基址 0x400F.E000  
偏移量 0x050  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留							MOSCPUPRIS	USBPLLRIS	PLLLRIS	保留		MOFRIS	保留	BORRIS	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:9	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
8	MOSCPUPRIS	RO	0	MOSC上电原始中断状态  值 描述 1 MOSC 经过充足的时间到达预期的频率。这个上电时间值显示为 $T_{MOSC\_START}$ 。 0 MOSC 没有经过充足的时间到达预期的频率。  通过在MISC寄存器的MOSCPUPMIS位写1可以清除这个位。
7	USBPLLRIS	RO	0	USB PLL 锁定原始中断状态  值 描述 1 USB PLL 定时器已经达到了 $T_{READY}$ , 表明 USB PLL 有足够的时间实现锁定。 0 USB PLL 定时器没有达到 $T_{READY}$ 。  通过在MISC寄存器的USBPLLLMIS位写1可以清除这个位。
6	PLLLRIS	RO	0	PLL 锁定原始中断状态  值 描述 1 PLL 定时器已经达到了 $T_{READY}$ , 表明 PLL 经过了充足的时间锁定。 0 PLL 定时器没有达到 $T_{READY}$ 。  通过在MISC寄存器的PLLLMIS位写1可以清除这个位。
5:4	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
3	MOFRIS	RO	0	<p>主振荡器故障原始中断状态</p> <p>值 描述</p> <p>1 MOSCCTL 寄存器中的 MOSCIM 位置位并且主振荡器发生故障。</p> <p>0 主振荡器未发生故障。</p> <p>通过在MISC寄存器的MOFMIS位写1可以清除这个位。</p>
2	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
1	BORRIS	RO	0	<p>掉电复位原始中断状态</p> <p>值 描述</p> <p>1 掉电条件当前起作用。</p> <p>0 掉电条件当前不起作用。</p> <p>注意为引发一次掉电事件的中断，必须清零PBORCTL寄存器的BORIOR位。</p> <p>通过在MISC寄存器的BORMIS位写1可以清除这个位。</p>
0	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>

## 寄存器 5: 中断屏蔽控制寄存器 (IMC), 偏移量 0x054

该寄存器包含了系统控制原始中断的屏蔽位。如果原始中断状态 (RIS) 寄存器中的相应位被置位, 那么原始中断会被发送到中断控制器。

### 中断屏蔽控制寄存器 (IMC)

基址 0x400F.E000  
偏移量 0x054  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留							MOSCPUPIM	USBPLLLIM	PLLLIM	保留		MOFIM	保留	BORIM	保留
类型	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	R/W	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:9	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
8	MOSCPUPIM	R/W	0	MOSC上电中断屏蔽  值 描述 1 当RIS寄存器中的MOSCPUPRIS位被置位时, 中断被发送到中断控制器。 0 MOSCPUPRIS 中断被抑制, 不会发送到中断控制器。
7	USBPLLLIM	R/W	0	USB PLL 锁定中断屏蔽  值 描述 1 当 RIS 寄存器中的 USBPLLLRIS 位被置位时, 中断被发送到中断控制器。 0 USBPLLLRIS 中断被抑制, 不会发送到中断控制器。
6	PLLLIM	R/W	0	PLL 锁定中断屏蔽  值 描述 1 当 RIS 寄存器中的 PLLLRIS 位被置位时, 中断被发送到中断控制器。 0 PLLLRIS 中断被抑制, 不会发送到中断控制器。
5:4	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3	MOFIM	RO	0	主振荡器故障中断屏蔽  值 描述 1 当RIS寄存器中的MOFRIS位被置位时, 中断被发送到中断控制器。 0 MOFRIS 中断被抑制, 中断不会被发送到中断控制器。

位/域	名称	类型	复位	描述
2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
1	BORIM	R/W	0	掉电复位中断屏蔽  值 描述 1 当 RIS 寄存器中的 BORRIS 位被置位时，中断被发送到中断控制器。 0 BORRIS 中断被抑制，不会发送到中断控制器。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 6: 屏蔽中断状态和清除寄存器 ( MISC ) , 偏移量 0x058

读该寄存器将返回原始中断状态 (RIS) 寄存器中相应中断的当前屏蔽的状态值。所有的位都是 R/W1C , 所以对一个位写入1可以清零RIS寄存器 ( 见 228页 ) 中相应的原始中断位。

### 屏蔽中断状态和清除寄存器 (MISC)

基址 0x400F.E000  
偏移量 0x058  
类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留							MOSCPUPMIS	USBPLLLMIS	PLLLMIS	保留		MOFMIS	保留	BORMIS	保留
类型	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	RO	RO	RO	RO	R/W1C	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:9	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
8	MOSCPUPMIS	R/W1C	0	MOSC上电屏蔽的中断状态  值 描述 1 当读取该位时, 值为1表示发出了非屏蔽中断信号, 这是由于MOSC PLL 经过了充足的时间锁定而引起的。 对这个位写1清零该位和RIS寄存器的MOSCPUPRIS位。 0 当读取该位时, 值为0表示MOSC PLL 没有经过了充足的时间锁定。 写0对该位状态没有影响。
7	USBPLLLMIS	R/W1C	0	USB PLL 锁定屏蔽的中断状态  值 描述 1 当读取该位时, 值为1表示发出了非屏蔽中断信号, 这是由于 USB PLL 经过了充足的时间锁定而引起的。 对这个位写1清零该位和RIS寄存器的USBPLLLRIS位。 0 当读取该位时, 值为0表示USB PLL 没有经过了充足的时间锁定。 写0对该位状态没有影响。
6	PLLLMIS	R/W1C	0	PLL 锁定屏蔽中断状态  值 描述 1 当读取该位时, 值为1表示发出了非屏蔽中断信号, 这是由于 PLL 经过了充足的时间锁定而引起的。 对这个位写1清零该位和RIS寄存器的PLLLRIS位。 0 当读取该位时, 值为0表示 PLL 没有经过了充足的时间锁定。 写0对该位状态没有影响。
5:4	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。



位/域	名称	类型	复位	描述
3	MOFMIS	RO	0	<p>主振荡器故障屏蔽中断状态</p> <p>值 描述</p> <p>1 当读取该位为1时，表示一个非屏蔽中断信号被发出，原因是主振荡器发生故障。</p> <p>对这个位写1清零该位和RIS寄存器的MOFRIS位。</p> <p>0 当读取该位时，值为0表示主振荡器没有发生故障。</p> <p>写 0 对该位状态没有影响。</p>
2	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
1	BORMIS	R/W1C	0	<p>BOR屏蔽的中断状态</p> <p>值 描述</p> <p>1 当读取该位时，值为 1 表示发出了非屏蔽中断信号，这是由于一次掉电条件而引起的。</p> <p>对这个位写1清零该位和RIS寄存器的BORRIS位。</p> <p>0 当读取该位时，值为0表示掉电条件没有发生。</p> <p>写 0 对该位状态没有影响。</p>
0	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>

## 寄存器 7: 复位原因寄存器 ( RESC ) , 偏移量 0x05C

复位后该寄存器随复位原因而置位。该寄存器中的位具有粘着性, 经过多个复位序列后仍保持它们的状态, 上电复位除外, 在上电复位时, RESC寄存器中除POR外的所有其它位都被清零。

### 复位原因寄存器 (RESC)

基址 0x400F.E000

偏移量 0x05C

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															MOSCFAIL
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										WDT1	SW	WDT0	BOR	POR	EXT
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:17	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
16	MOSCFAIL	R/W	-	MOSC 故障复位  值 描述 1 当读取该位时, 该位表示MOSC电路被启用作时钟验证, 但是MOSCCTL 寄存器中的 MOSCIM 位清零时失败, 于是产生了一次复位事件。 0 当读取该位时, 该位表示从之前的上电复位开始MOSC失败没有产生过复位。 对这个位写0可将它清零。
15:6	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5	WDT1	R/W	-	看门狗定时器1复位  值 描述 1 当读取该位时, 该位表示看门狗定时器1超时并产生了一次复位。 0 当读取该位时, 该位表示从之前的上电复位开始看门狗定时器1没有产生过复位。 对这个位写0可将它清零。
4	SW	R/W	-	软件复位  值 描述 1 当读取该位时, 该位表示软件复位引起了一次复位事件。 0 当读取该位时, 该位表示从之前的上电复位开始软件复位没有产生过复位。 对这个位写0可将它清零。

位/域	名称	类型	复位	描述
3	WDT0	R/W	-	看门狗定时器0复位  值 描述 1 当读取该位时，该位表示看门狗定时器0超时并产生了一次复位。 0 当读取该位时，该位表示从之前的上电复位开始看门狗定时器0没有产生过复位。 对这个位写0可将它清零。
2	BOR	R/W	-	掉电复位  值 描述 1 当读取该位时，该位表示掉电复位引起了一次复位事件。 0 当读取该位时，该位表示从之前的上电复位开始掉电复位没有产生过复位。 对这个位写0可将它清零。
1	POR	R/W	-	上电复位  值 描述 1 当读取该位时，该位表示上电复位引起了一次复位事件。 0 当读取该位时，该位表示上电复位没有产生过复位。 对这个位写0可将它清零。
0	EXT	R/W	-	外部复位  值 描述 1 当读取该位时，该位表示外部复位 ( $\overline{RST}$ 有效) 引起了一次复位事件。 0 当读取该位时，该位表示自之前的上电复位后，外部复位 ( $\overline{RST}$ 有效) 没有引起过复位事件。 对这个位写0可将它清零。

## 寄存器 8: 运行模式时钟配置寄存器 (RCC) , 偏移量 0x060

该寄存器中的位配置系统时钟和振荡器。

**重要:** 写 RCC 寄存器后再写 RCC2 寄存器。如果需要对 RCC 寄存器进行后续写操作, 则应在写 RCC 寄存器之后和写 RCC2 寄存器之前安排另一次寄存器访问。

应将 PWM 外设配置 (PWMP) 寄存器中的 USEPWMDIV 和 PWMDIV 域用于控制 PWM 时钟。要支持传统软件, 可使用该寄存器中的 USEPWMDIV 和 PWMDIV 域。写这些传统域的同时, 也将写 PWMP 寄存器中的相应域。如果通过对该寄存器进行写操作更改这些域, 则可以通过对该寄存器进行读操作对它们进行正确回读。如果软件使用 PWMP 寄存器配置 PWM 时钟, 则 PWM 时钟根据规定操作, 但是不会修改该寄存器中 USEPWMDIV 和 PWMDIV 域的值。

### 运行模式时钟配置寄存器 (RCC)

基址 0x400F.E000  
偏移量 0x060  
类型 R/W, 复位 0x078E.3D51

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留				ACG	SYSDIV				USESYSDIV	保留	USEPWMDIV	PWMDIV			保留
类型	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W	RO
复位	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		PWRDN	保留	BYPASS	XTAL				OSCSRC		保留		IOSCDIS	MOSCDIS	
类型	RO	RO	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	R/W	R/W
复位	0	0	1	1	1	1	0	1	0	1	0	1	0	0	0	1

位/域	名称	类型	复位	描述
31:28	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
27	ACG	R/W	0	<p>自动时钟门控</p> <p>该位规定了如果微控制器进入了睡眠模式或深度睡眠模式, 系统是否分别使用睡眠模式时钟门控控制 (SCGCn) 寄存器和深度睡眠模式时钟门控控制 (DCGCn) 寄存器。</p> <p><b>值 描述</b></p> <p>1 当微控制器处于一种睡眠模式时, SCGCn或DCGCn寄存器用来控制分配给外设的时钟。当微控制器处于一种睡眠模式时, SCGCn 和 DCGCn 寄存器使没有使用的外设耗费更少的功耗。</p> <p>0 当微控制器进入一种睡眠模式时, 运行模式时钟门控控制 (RCGCn) 寄存器被使用。</p>
26:23	SYSDIV	R/W	0xF	<p>系统时钟分频值</p> <p>规定了使用哪个分频值来产生系统时钟, 该系统时钟来自PLL输出或振荡器源 (取决于该寄存器中BYPASS位如何配置)。关于位编码的信息参见表 5-4 在 212页。</p> <p>如果 SYSDIV 的值小于 MINSYSDIV (请参考 415页), 并且 PLL 正在被使用, 那么 MINSYSDIV 值被作为分频器使用。</p> <p>如果PLL没有被使用, SYSDIV值可以小于MINSYSDIV。</p>

在运行模式中, 总是使用 RCGCn 寄存器控制时钟。

位/域	名称	类型	复位	描述
22	USESYSDIV	R/W	0	<p>启用系统时钟分频器</p> <p>值 描述</p> <p>1 系统时钟分频器用作系统时钟源。当PLL被选作源时，将强制使用系统时钟分频器。</p> <p>如果RCC2寄存器中的USERCC2位被置位，那么会使用RCC2寄存器中的SYSDIV2域来作为系统时钟分频器，而不是这个寄存器中的SYSDIV域。</p> <p>0 系统时钟不分频使用。</p>
21	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
20	USEPWMDIV	R/W	0	<p>启用 PWM 时钟分频</p> <p>值 描述</p> <p>1 PWM 时钟分频器是 PWM 的时钟源。</p> <p>0 系统时钟是 PWM 的时钟源。</p> <p>请注意，在使用 PWM 分频值时，它适用于两个 PWM 模块的时钟。</p>
19:17	PWMDIV	R/W	0x7	<p>PWM 单元时钟分频值</p> <p>该域规定的二进制分频值用于预分频系统时钟以便作为PWM模块的时序参考。该时钟的上升沿与系统时钟同步。</p> <p>值 分频系数</p> <p>0x0 /2</p> <p>0x1 /4</p> <p>0x2 /8</p> <p>0x3 /16</p> <p>0x4 /32</p> <p>0x5 /64</p> <p>0x6 /64</p> <p>0x7 /64 (默认)</p>
16:14	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
13	PWRDN	R/W	1	<p>PLL掉电</p> <p>值 描述</p> <p>1 PLL 掉电。在该位置位前，必须注意要确保有另外的时钟起作用，并且BYPASS位被置位。</p> <p>0 PLL 正常工作。</p>
12	保留	RO	1	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述																																																																					
11	BYPASS	R/W	1	<p>PLL旁路</p> <p>值 描述</p> <p>1 系统时钟来自OSC源并且被SYSDIV规定的分频值分频。</p> <p>0 系统时钟是被SYSDIV规定的分频值分频的PLL输出时钟。</p> <p>编程指南参见表 5-4 在 212页。</p> <p>注意: ADC必须使用PLL作为时钟或直接使用16MHz时钟源以便正常工作。</p>																																																																					
10:6	XTAL	R/W	0x15	<p>晶振值</p> <p>该域规定了与主振荡器连接的晶振值。该域的编码在下面提供。</p> <p>USB 接口使用的频率在表中给出。为了能在 USB 规定的计时要求内有效，必须使用 5、6、8、10、12 或 16 MHz 的晶振。</p> <table border="1"> <thead> <tr> <th>值</th> <th>不使用 PLL 时的晶振频率 (MHz)</th> <th>使用 PLL 时的晶振频率 (MHz)</th> </tr> </thead> <tbody> <tr> <td>0x00-0x5</td> <td></td> <td>保留</td> </tr> <tr> <td>0x06</td> <td>4 MHz</td> <td>保留</td> </tr> <tr> <td>0x07</td> <td>4.096 MHz</td> <td>保留</td> </tr> <tr> <td>0x08</td> <td>4.9152 MHz</td> <td>保留</td> </tr> <tr> <td>0x09</td> <td></td> <td>5 MHz (USB)</td> </tr> <tr> <td>0x0A</td> <td></td> <td>5.12 MHz</td> </tr> <tr> <td>0x0B</td> <td></td> <td>6 MHz (USB)</td> </tr> <tr> <td>0x0C</td> <td></td> <td>6.144 MHz</td> </tr> <tr> <td>0x0D</td> <td></td> <td>7.3728 MHz</td> </tr> <tr> <td>0x0E</td> <td></td> <td>8 MHz (USB)</td> </tr> <tr> <td>0x0F</td> <td></td> <td>8.192 MHz</td> </tr> <tr> <td>0x10</td> <td></td> <td>10.0 MHz (USB)</td> </tr> <tr> <td>0x11</td> <td></td> <td>12.0 MHz (USB)</td> </tr> <tr> <td>0x12</td> <td></td> <td>12.288 MHz</td> </tr> <tr> <td>0x13</td> <td></td> <td>13.56 MHz</td> </tr> <tr> <td>0x14</td> <td></td> <td>14.31818 MHz</td> </tr> <tr> <td>0x15</td> <td></td> <td>16.0 MHz ( 复位值 ) (USB)</td> </tr> <tr> <td>0x16</td> <td></td> <td>16.384 MHz</td> </tr> <tr> <td>0x17</td> <td></td> <td>18.0 MHz (USB)</td> </tr> <tr> <td>0x18</td> <td></td> <td>20.0 MHz (USB)</td> </tr> <tr> <td>0x19</td> <td></td> <td>24.0 MHz (USB)</td> </tr> <tr> <td>0x1A</td> <td></td> <td>25.0 MHz (USB)</td> </tr> </tbody> </table>	值	不使用 PLL 时的晶振频率 (MHz)	使用 PLL 时的晶振频率 (MHz)	0x00-0x5		保留	0x06	4 MHz	保留	0x07	4.096 MHz	保留	0x08	4.9152 MHz	保留	0x09		5 MHz (USB)	0x0A		5.12 MHz	0x0B		6 MHz (USB)	0x0C		6.144 MHz	0x0D		7.3728 MHz	0x0E		8 MHz (USB)	0x0F		8.192 MHz	0x10		10.0 MHz (USB)	0x11		12.0 MHz (USB)	0x12		12.288 MHz	0x13		13.56 MHz	0x14		14.31818 MHz	0x15		16.0 MHz ( 复位值 ) (USB)	0x16		16.384 MHz	0x17		18.0 MHz (USB)	0x18		20.0 MHz (USB)	0x19		24.0 MHz (USB)	0x1A		25.0 MHz (USB)
值	不使用 PLL 时的晶振频率 (MHz)	使用 PLL 时的晶振频率 (MHz)																																																																							
0x00-0x5		保留																																																																							
0x06	4 MHz	保留																																																																							
0x07	4.096 MHz	保留																																																																							
0x08	4.9152 MHz	保留																																																																							
0x09		5 MHz (USB)																																																																							
0x0A		5.12 MHz																																																																							
0x0B		6 MHz (USB)																																																																							
0x0C		6.144 MHz																																																																							
0x0D		7.3728 MHz																																																																							
0x0E		8 MHz (USB)																																																																							
0x0F		8.192 MHz																																																																							
0x10		10.0 MHz (USB)																																																																							
0x11		12.0 MHz (USB)																																																																							
0x12		12.288 MHz																																																																							
0x13		13.56 MHz																																																																							
0x14		14.31818 MHz																																																																							
0x15		16.0 MHz ( 复位值 ) (USB)																																																																							
0x16		16.384 MHz																																																																							
0x17		18.0 MHz (USB)																																																																							
0x18		20.0 MHz (USB)																																																																							
0x19		24.0 MHz (USB)																																																																							
0x1A		25.0 MHz (USB)																																																																							

位/域	名称	类型	复位	描述
5:4	OSCSRC	R/W	0x1	<p>振荡器源</p> <p>选择 OSC 的输入源。值有：</p> <p>值 输入源</p> <p>0x0 主振荡器 主振荡器</p> <p>0x1 PIOSC 精确内部振荡器 (默认)</p> <p>0x2 PIOSC/4 精确内部振荡器/4</p> <p>0x3 30 kHz 30kHz内部振荡器</p> <p>其它振荡器源参见RCC2寄存器。</p>
3:2	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
1	IOSCDIS	R/W	0	<p>精确内部振荡器禁用</p> <p>值 描述</p> <p>1 精确内部振荡器 (PIOSC) 禁用。</p> <p>0 精确内部振荡器启用。</p>
0	MOSCDIS	R/W	1	<p>主振荡器禁用</p> <p>值 描述</p> <p>1 主振荡器禁用 (默认)。</p> <p>0 主振荡器启用。</p>

### 寄存器 9: GPIO 高性能总线控制寄存器 ( GPIOHBCTL ) , 偏移量 0x06C

该寄存器控制着哪个内部总线可用来访问每个GPIO端口。当某位清零时，相应的GPIO端口穿过老的高级外设总线 (APB) 并通过APB存储器槽 (aperture) 被访问。当某位置位时，相应的GPIO端口穿过高级高性能总线 (AHB) 并通过AHB存储器槽被访问。每个GPIO端口都可以独立的被配置使用AHB或APB，但是只可以通过一个槽被访问。AHB总线能比APB总线提供更好的连续访问性能。对于启用用于AHB访问的端口 (见表 10-6 在 607页)，存储器映射中的地址槽会改变。

**重要:** 端口 K-N 和 P-Q 仅适用于 AHB 总线，因此相应位复位为 1。如果其中一个位清零，则禁用相应的端口。如果这些端口中的任何一个正在使用，应用使用读-修改-写的操作来更改该寄存器的值，以便这些端口保持启用。

#### GPIO 高性能总线控制寄存器 (GPIOHBCTL)

基址 0x400F.E000  
偏移量 0x06C  
类型 R/W, 复位 0x0000.7E00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		PORTP	PORTN	PORTM	PORTL	PORTK	PORTJ	PORTH	PORTG	PORTF	PORTE	PORTD	PORTC	PORTB	PORTA
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
13	PORTP	R/W	1	端口P高级高性能总线 该位为端口P定义了存储器槽。  值 描述 1 高级高性能总线 (AHB)。 0 端口P已禁用。
12	PORTN	R/W	1	端口N高级高性能总线 该位为端口N定义了存储器槽  值 描述 1 高级高性能总线 (AHB)。 0 端口N已禁用。
11	PORTM	R/W	1	端口M高级高性能总线 该位为端口M定义了存储器槽。  值 描述 1 高级高性能总线 (AHB)。 0 端口M已禁用。



位/域	名称	类型	复位	描述
10	PORTL	R/W	1	<p>端口L高级高性能总线 该位为端口L定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 端口L已禁用。</p>
9	PORTK	R/W	1	<p>端口K高级高性能总线 该位为端口K定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 端口K已禁用。</p>
8	PORTJ	R/W	0	<p>端口J高级高性能总线 该位为端口J定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>
7	PORTH	R/W	0	<p>端口H高级高性能总线 该位为端口H定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>
6	PORTG	R/W	0	<p>端口G高级高性能总线 该位为端口G定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>
5	PORTF	R/W	0	<p>端口F高级高性能总线 该位为端口F定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>

位/域	名称	类型	复位	描述
4	PORTE	R/W	0	<p>端口E高级高性能总线 该位为端口E定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>
3	PORTD	R/W	0	<p>端口D高级高性能总线 该位为端口D定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>
2	PORTC	R/W	0	<p>端口C高级高性能总线 该位为端口C定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>
1	PORTB	R/W	0	<p>端口B高级高性能总线 该位为端口B定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>
0	PORTA	R/W	0	<p>端口A高级高性能总线 该位为端口A定义了存储器槽。</p> <p>值 描述 1 高级高性能总线 (AHB)。 0 高级外设总线 (APB)。这个总线是传统总线。</p>

## 寄存器 10: 运行模式时钟配置寄存器 2 (RCC2) , 偏移量 0x070

该寄存器替代了类似RCC的寄存器域，如表 5-8 所示。当USERCC2位置位时，允许使用RCC2寄存器的扩展功能，同时也提供方法来向下兼容以前的器件。每个取代RCC域的RCC2域都位于相同的LSB位位置；但是，某些RCC2域比相应的RCC域更大。

表 5-8. 替代RCC域的RCC2域

RCC2域...	替代 RCC 域
SYSDIV2, 位[28:23]	SYSDIV, 位[26:23]
PWRDN2, 位[13]	PWRDN, 位[13]
BYPASS2, 位[11]	BYPASS, 位[11]
OSCSRC2, 位[6:4]	OSCSRC, 位[5:4]

**重要:** 写 RCC 寄存器后再写 RCC2 寄存器。如果需要对 RCC 寄存器进行后续写操作，则应在写 RCC 寄存器之后和写 RCC2 寄存器之前安排另一次寄存器访问。

### 运行模式时钟配置寄存器 2 (RCC2)

基址 0x400F.E000

偏移量 0x070

类型 R/W, 复位 0x07C0.6810

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	USERCC2	DIV400	保留	SYSDIV2						SYSDIV2LSB	保留					
类型	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	USBPWRDN	PWRDN2	保留	BYPASS2	保留				OSCSRC2			保留			
类型	RO	R/W	R/W	RO	R/W	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO
复位	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0

位/域	名称	类型	复位	描述
31	USERCC2	R/W	0	使用RCC2  值 描述 1 RCC2寄存器域代替RCC寄存器域。 0 使用 RCC 寄存器域而忽略 RCC2 域。
30	DIV400	R/W	0	将PLL分为 400MHz vs 200MHz 该位和SYSDIV2LSB位一起使用允许附加的频率选择。  值 描述 1 附加在SYSDIV2域的SYSDIV2LSB位之后可创建一个7位的分频器，以使用400 MHz的PLL输出，参见表 5-6 在 213页。 0 使用 SYSDIV2 本来的值并应用到被预分频的 200 MHz PLL 输出。编程指南参见表 5-5 在 212页。
29	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
28:23	SYSDIV2	R/W	0x0F	<p>系统时钟分频值2</p> <p>规定了使用哪个分频值来产生系统时钟，该系统时钟来自PLL输出或振荡器源(取决于该寄存器中BYPASS2位如何配置)。当RCC寄存器中的USESYSYSDIV位和本寄存器中的USERCC2位都置位时SYSDIV2用作分频值。编程指南参见表 5-5 在 212页。</p>
22	SYSDIV2LSB	R/W	1	<p>SYSDIV2的附加LSB</p> <p>当DIV400置位时，该位变成SYSDIV2的LSB。如果DIV400清零，该位没有用。编程指南参见表 5-5 在 212页。</p> <p>该位只有在DIV400置位时才能置位和清零。</p>
21:15	保留	RO	0x0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
14	USBPWRDN	R/W	1	<p>掉电 USB PLL</p> <p>值 描述</p> <p>1 USB PLL 掉电。</p> <p>0 USB PLL 工作正常。</p>
13	PWRDN2	R/W	1	<p>掉电 PLL 2</p> <p>值 描述</p> <p>1 PLL 掉电。</p> <p>0 PLL 工作正常。</p>
12	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
11	BYPASS2	R/W	1	<p>PLL 旁路 2</p> <p>值 描述</p> <p>1 系统时钟来自OSC源并且被SYSDIV2规定的分频值分频。</p> <p>0 系统时钟是被SYSDIV2规定的分频值分频的PLL输出时钟。</p> <p>编程指南参见表 5-5 在 212页。</p> <p>注意: ADC必须使用PLL作为时钟或直接使用16MHz时钟源以便正常工作。</p>
10:7	保留	RO	0x0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>

位/域	名称	类型	复位	描述														
6:4	OSCSRC2	R/W	0x1	振荡器源 2 选择 OSC 的输入源。值有 : <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>主振荡器 主振荡器</td> </tr> <tr> <td>0x1</td> <td>PIOSC 精确内部振荡器</td> </tr> <tr> <td>0x2</td> <td>PIOSC/4 精确内部振荡器/4</td> </tr> <tr> <td>0x3</td> <td>30 kHz 30kHz内部振荡器</td> </tr> <tr> <td>0x4-0x6</td> <td>保留</td> </tr> <tr> <td>0x7</td> <td>32.768 kHz 32.768-kHz 外部振荡器</td> </tr> </tbody> </table>	值	描述	0x0	主振荡器 主振荡器	0x1	PIOSC 精确内部振荡器	0x2	PIOSC/4 精确内部振荡器/4	0x3	30 kHz 30kHz内部振荡器	0x4-0x6	保留	0x7	32.768 kHz 32.768-kHz 外部振荡器
值	描述																	
0x0	主振荡器 主振荡器																	
0x1	PIOSC 精确内部振荡器																	
0x2	PIOSC/4 精确内部振荡器/4																	
0x3	30 kHz 30kHz内部振荡器																	
0x4-0x6	保留																	
0x7	32.768 kHz 32.768-kHz 外部振荡器																	
3:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。														

## 寄存器 11: 主振荡器控制寄存器 (MOSCCTL), 偏移量 0x07C

该寄存器提供主振荡器功能的控制, 包括启用 MOSC 时钟验证电路的能力、MOSC 发生故障时需采取的行动以及是否连接晶体。当启用时, 该电路监测MOSC的频率以确认振荡器工作在规定的范围内。如果时钟启用后变得无效, 微控制器会发出一个上电复位并重新启动到NMI处理程序或产生一个中断。

### 主振荡器控制寄存器 (MOSCCTL)

基址 0x400F.E000  
偏移量 0x07C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													NOXTAL	MOSCIM	CVAL
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2	NOXTAL	R/W	0	未连接晶体  值 描述 1 应在晶体或外部振荡器未连接到 OSC0 和 OSC1 输入时置位该位, 以减少功耗。 0 应在晶体或外部振荡器连接到 OSC0 和 OSC1 输入时清零该位, 不论是否使用或关闭 MOSC。
1	MOSCIM	R/W	0	MOSC 故障动作  值 描述 1 如果 MOSC 发生故障, 则根据 RIS 寄存器中的 MOFRIS 位的指示会产生一个中断。 0 如果 MOSC 发生故障, 则会产生 MOSC 故障复位, 并且重新启动到 NMI 处理程序。  不论采取什么动作, 如果 MOSC 发生故障, 振荡器源会自动切换至 PIOSC。
0	CVAL	R/W	0	MOSC时钟确认  值 描述 1 MOSC监测电路启用。 0 MOSC监测电路禁用。

## 寄存器 12: 深度睡眠时钟配置寄存器 (DSLPCCLKCFG), 偏移量 0x144

该寄存器为深度睡眠模式的硬件控制提供配置信息。

## 深度睡眠时钟配置寄存器 (DSLPCCLKCFG)

基址 0x400F.E000

偏移量 0x144

类型 R/W, 复位 0x0780.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			DSDIVORIDE						保留						
类型	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留						DSOSCSRC					保留				
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
28:23	DSDIVORIDE	R/W	0x0F	分频器域代替 如果当PLL正运行时启用深度睡眠模式, PLL将被禁用。在深度睡眠期间该6位域包含了一个系统分频器域来代替RCC寄存器中的SYSDIV域或RCC2寄存器中的SYSDIV2域。该分频器用于DSOSCSRC域选择的源。  值 描述 0x0 /1 0x1 /2 0x2 /3 0x3 /4 ... ... 0x3F /64
22:7	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述														
6:4	DSOSCSRC	R/W	0x0	<p>时钟源</p> <p>规定了在深度睡眠模式期间时钟源。</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td> <p>主振荡器</p> <p>使用主振荡器用作源。</p> <p>注意: 如果PIOSC正用作PLL的时钟参考, 那么在深度睡眠模式PIOSC代替MOSC作为时钟源。</p> </td> </tr> <tr> <td>0x1</td> <td> <p>PIOSC</p> <p>使用精确内部16MHz振荡器作为源。</p> </td> </tr> <tr> <td>0x2</td> <td>保留</td> </tr> <tr> <td>0x3</td> <td> <p>30 kHz</p> <p>使用30kHz内部振荡器作为源。</p> </td> </tr> <tr> <td>0x4-0x6</td> <td>保留</td> </tr> <tr> <td>0x7</td> <td> <p>32.768 kHz</p> <p>使用休眠模块 32.768-kHz 外部振荡器作为源。</p> </td> </tr> </tbody> </table>	值	描述	0x0	<p>主振荡器</p> <p>使用主振荡器用作源。</p> <p>注意: 如果PIOSC正用作PLL的时钟参考, 那么在深度睡眠模式PIOSC代替MOSC作为时钟源。</p>	0x1	<p>PIOSC</p> <p>使用精确内部16MHz振荡器作为源。</p>	0x2	保留	0x3	<p>30 kHz</p> <p>使用30kHz内部振荡器作为源。</p>	0x4-0x6	保留	0x7	<p>32.768 kHz</p> <p>使用休眠模块 32.768-kHz 外部振荡器作为源。</p>
值	描述																	
0x0	<p>主振荡器</p> <p>使用主振荡器用作源。</p> <p>注意: 如果PIOSC正用作PLL的时钟参考, 那么在深度睡眠模式PIOSC代替MOSC作为时钟源。</p>																	
0x1	<p>PIOSC</p> <p>使用精确内部16MHz振荡器作为源。</p>																	
0x2	保留																	
0x3	<p>30 kHz</p> <p>使用30kHz内部振荡器作为源。</p>																	
0x4-0x6	保留																	
0x7	<p>32.768 kHz</p> <p>使用休眠模块 32.768-kHz 外部振荡器作为源。</p>																	
3:0	保留	RO	0x0	<p>软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。</p>														



## 寄存器 13: 系统属性寄存器 (SYSPROP), 偏移量 0x14C

该寄存器提供关于某些系统控制属性在微控制器上是否存在的信息。

### 系统属性寄存器 (SYSPROP)

基址 0x400FE000

偏移量 0x14C

类型 RO, 复位 0x0000.1D31

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															FPU
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	1	1	1	0	1	0	0	1	1	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0xE98	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	FPU	RO	0x1	FPU 存在 该位指示 FPU 在 CortexM4 内核中是否存在。  值 描述 0 FPU 不存在。 1 FPU 存在。

### 寄存器 14: 精确内部振荡器校准寄存器 ( PIOSCCAL ) , 偏移量 0x150

该寄存器用来更新或重新校准精确内部振荡器。请注意, 必须将 32.768-kHz 振荡器用作休眠模块时钟源, 以使用户能校准 PIOSC。

#### 精确内部振荡器校准寄存器 (PIOSCCAL)

基址 0x400F.E000  
偏移量 0x150  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UTEN	保留														
类型	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留						CAL	UPDATE	保留	UT						
类型	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31	UTEN	R/W	0	使用用户校准值  值 描述 1 该寄存器的[6:0]位校准值用于任何更新校准的操作。 0 出厂校准值用于更新校准的操作。
30:10	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
9	CAL	R/W	0	开始校准  值 描述 1 开始对 PIOSC 进行新的校准。结果显示在 PIOSCSTAT 寄存器中。校准完成后, 操作中产生的校准值在 PIOSC 中有效。不论校准通过或是失败, 结果都将替代之前所有的更新校准操作。 0 没有动作。  该位在置位后自动清零。
8	UPDATE	R/W	0	更新校准  值 描述 1 通过 PIOSCSTAT 寄存器中的 UT 位或 DT 位更新 PIOSC 校准值。与UTEN一起使用。 0 没有动作。  该位在更新后自动清零。
7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
6:0	UT	R/W	0x0	用户校准值 用户校准值, 可被加载到PIOSC。 关于校准 PIOSC 的更多信息, 请参阅“主 PLL 频率配置”在 213页。

## 寄存器 15: 精确内部振荡器统计寄存器 (PIOSCSTAT), 偏移量 0x154

该寄存器提供关于 PIOSC 校准的用户信息。请注意, 必须将 32.768-kHz 振荡器用作休眠模块时钟源, 以使用户能校准 PIOSC。

### 精确内部振荡器统计寄存器 (PIOSCSTAT)

基址 0x400F.E000

偏移量 0x154

类型 RO, 复位 0x0000.0040

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留									DT						
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留						RESULT		保留	CT						
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:23	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
22:16	DT	RO	-	默认校准值 该域包含默认校准值。该值在每次完全上电后装入到 PIOSC 中。
15:10	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
9:8	RESULT	RO	0	校准结果  值 描述 0x0 尚未尝试校准。 0x1 完成的上次校准操作符合 1% 的精度。 0x2 失败的上次校准操作符合 1% 的精度。 0x3 保留
7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
6:0	CT	RO	0x40	校准值 该域包含上次校准操作的校准值。出厂校准后, CT 和 DT 相同。

### 寄存器 16: PLL 频率寄存器 0 ( PLLFREQ0 ) , 偏移量 0x160

该寄存器始终包含提交到系统 PLL 的当前 M 值。

可以使用以下等式计算 PLL 频率：

$$\text{PLL frequency} = (\text{XTAL frequency} * \text{MDIV}) / ((Q + 1) * (N + 1))$$

其中

$$\text{MDIV} = \text{MINT} + (\text{MFRAC} / 1024)$$

Q 和 N 值在 PLLFREQ1 寄存器中显示。表 25-10 在 1298 页显示 M、Q 和 N 值，以及由此产生的不同 XTAL 配置的 PLL 频率。

#### PLL 频率寄存器 0 (PLLFREQ0)

基址 0x400F.E000  
偏移量 0x160  
类型 RO, 复位 0x0000.0032

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留												MFRAC			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MFRAC						MINT									
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:20	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
19:10	MFRAC	RO	0x32	PLL M 分数值 该域包含 PLL M 值的整数部分。
9:0	MINT	RO	0x00	PLL M 整数部分 该域包含 PLL M 值的整数部分。

## 寄存器 17: PLL 频率寄存器 1 ( PLLFREQ1 ) , 偏移量 0x164

该寄存器始终包含提交到系统 PLL 的当前 Q 和 N 值。

M 值在 PLLFREQ0 寄存器中显示。表 25-10 在 1298 页 显示 M、Q 和 N 值，以及由此产生的不同 XTAL 配置的 PLL 频率。

### PLL 频率寄存器 1 (PLLFREQ1)

基址 0x400F.E000  
偏移量 0x164  
类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留			Q				保留				N				
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:13	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
12:8	Q	RO	0x0	PLL Q 值 该域包含 PLL Q 值。
7:5	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
4:0	N	RO	0x1	PLL N 值 该域包含 PLL N 值。

## 寄存器 18: PLL 状态寄存器 ( PLLSTAT ) , 偏移量 0x168

该寄存器显示 PLL 锁定的直接状态。

### PLL 状态寄存器 (PLLSTAT)

基址 0x400FE000  
 偏移量 0x168  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															LOCK
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	LOCK	RO	0x0	PLL 锁定
	值 描述			
	1 PLL 上电并锁定。			
	0 PLL 掉电或尚未锁定。			

## 寄存器 19: 看门狗定时器外设存在寄存器 ( PPWD ) , 偏移量 0x300

PPWD 寄存器提供关于看门狗模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的看门狗定时器。但是, 要支持传统软件, 可使用 DC1 寄存器。DC1 寄存器的读操作正确识别传统模块是否存在。

### 看门狗定时器外设存在寄存器 (PPWD)

基址 0x400FE000

偏移量 0x300

类型 RO, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	RO	0x1	看门狗定时器 1 存在  值 描述 1 看门狗模块 1 存在。 0 看门狗模块 1 不存在。
0	P0	RO	0x1	看门狗定时器 0 存在  值 描述 1 看门狗模块 0 存在。 0 看门狗模块 0 不存在。

## 寄存器 20: 16/32 位通用定时器外设存在寄存器 ( PPTIMER ) , 偏移量 0x304

PPTIMER 寄存器提供关于 16/32 位通用定时器模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的定时器。但是, 要支持传统软件, 可使用 DC2 寄存器。DC2 寄存器的读操作正确识别传统模块是否存在。软件必须使用该寄存器来确定不受 DC2 寄存器支持的模块是否存在。

### 16/32 位通用定时器外设存在寄存器 (PPTIMER)

基址 0x400FE000  
偏移量 0x304  
类型 RO, 复位 0x0000.003F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	P5	RO	0x1	16/32 位通用定时器 5 存在  值 描述 1 16/32 位通用定时器模块 5 存在。 0 16/32 位通用定时器模块 6 不存在。
4	P4	RO	0x1	16/32 位通用定时器 4 存在  值 描述 1 16/32 位通用定时器模块 4 存在。 0 16/32 位通用定时器模块 4 不存在。
3	P3	RO	0x1	16/32 位通用定时器 3 存在  值 描述 1 16/32 位通用定时器模块 3 存在。 0 16/32 位通用定时器模块 3 不存在。
2	P2	RO	0x1	16/32 位通用定时器 2 存在  值 描述 1 16/32 位通用定时器模块 2 存在。 0 16/32 位通用定时器模块 2 不存在。



位/域	名称	类型	复位	描述
1	P1	RO	0x1	16/32 位通用定时器 1 存在  值 描述 1 16/32 位通用定时器模块 1 存在。 0 16/32 位通用定时器模块 1 不存在。
0	P0	RO	0x1	16/32 位通用定时器 0 存在  值 描述 1 16/32 位通用定时器模块 0 存在。 0 16/32 位通用定时器模块 0 不存在。

## 寄存器 21: 通用输入/输出外设存在寄存器 ( PPGPIO ) , 偏移量 0x308

PPGPIO 寄存器提供关于通用输入/输出模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的 GPIO 端口。但是, 要支持传统软件, 可使用 DC4 寄存器。DC4 寄存器的读操作正确识别传统模块是否存在。软件必须使用该寄存器来确定不受 DC4 寄存器支持的模块是否存在。

### 通用输入/输出外设存在寄存器 (PPGPIO)

基址 0x400FE000  
偏移量 0x308  
类型 RO, 复位 0x0000.3FFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:15	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
14	P14	RO	0x0	GPIO 端口 Q 存在  值 描述 1 GPIO 端口 Q 存在。 0 GPIO 端口 Q 不存在。
13	P13	RO	0x1	GPIO 端口 P 存在  值 描述 1 GPIO 端口 P 存在。 0 GPIO 端口 P 不存在。
12	P12	RO	0x1	GPIO 端口 N 存在  值 描述 1 GPIO 端口 N 存在。 0 GPIO 端口 N 不存在。
11	P11	RO	0x1	GPIO 端口 M 存在  值 描述 1 GPIO 端口 M 存在。 0 GPIO 端口 M 不存在。

位/域	名称	类型	复位	描述
10	P10	RO	0x1	GPIO 端口 L 存在  值 描述 1 GPIO 端口 L 存在。 0 GPIO 端口 L 不存在。
9	P9	RO	0x1	GPIO 端口 K 存在  值 描述 1 GPIO 端口 K 存在。 0 GPIO 端口 K 不存在。
8	P8	RO	0x1	GPIO 端口 J 存在  值 描述 1 GPIO 端口 J 存在。 0 GPIO 端口 J 不存在。
7	P7	RO	0x1	GPIO 端口 H 存在  值 描述 1 GPIO 端口 H 存在。 0 GPIO 端口 H 不存在。
6	P6	RO	0x1	GPIO 端口 G 存在  值 描述 1 GPIO 端口 G 存在。 0 GPIO 端口 G 不存在。
5	P5	RO	0x1	GPIO 端口 F 存在  值 描述 1 GPIO 端口 F 存在。 0 GPIO 端口 F 不存在。
4	P4	RO	0x1	GPIO 端口 E 存在  值 描述 1 GPIO 端口 E 存在。 0 GPIO 端口 E 不存在。

---

位/域	名称	类型	复位	描述
3	P3	RO	0x1	GPIO 端口 D 存在  值 描述 1 GPIO 端口 D 存在。 0 GPIO 端口 D 不存在。
2	P2	RO	0x1	GPIO 端口 C 存在  值 描述 1 GPIO 端口 C 存在。 0 GPIO 端口 C 不存在。
1	P1	RO	0x1	GPIO 端口 B 存在  值 描述 1 GPIO 端口 B 存在。 0 GPIO 端口 B 不存在。
0	P0	RO	0x1	GPIO 端口 A 存在  值 描述 1 GPIO 端口 A 存在。 0 GPIO 端口 A 不存在。

## 寄存器 22: 微型直接存储器访问外设存在寄存器 ( PPDMA ) , 偏移量 0x30C

PPDMA 寄存器提供关于  $\mu$ DMA 模块的软件信息。

**重要:** 应使用该寄存器确定是否在该微控制器上执行  $\mu$ DMA 模块。但是, 要支持传统软件, 可使用 DC7 寄存器。读取 DC7 寄存器即可正确识别  $\mu$ DMA 模块是否存在。

### 微型直接存储器访问外设存在寄存器 (PPDMA)

基址 0x400FE000  
偏移量 0x30C  
类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	RO	0x1	$\mu$ DMA 模块存在
				值 描述
				1 $\mu$ DMA 模块存在。
				0 $\mu$ DMA 模块不存在。

## 寄存器 23: 休眠外设存在寄存器 ( PPHIB ) , 偏移量 0x314

PPHIB 寄存器提供关于休眠模块的软件信息。

**重要:** 应使用该寄存器确定是否在该微控制器上执行休眠模块。但是, 要支持传统软件, 可使用 DC1 寄存器。读取 DC1 寄存器即可正确识别休眠模块是否存在。

### 休眠外设存在寄存器 (PPHIB)

基址 0x400FE000  
偏移量 0x314  
类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	RO	0x1	休眠模块存在
				值 描述
				1 休眠模块存在。
				0 休眠模块不存在。

## 寄存器 24: 通用异步收发器外设存在寄存器 ( PPUART ) , 偏移量 0x318

PPUART 寄存器提供关于 UART 模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的 UART 模块。但是, 要支持传统软件, 可使用 DC2 寄存器。读取 DC2 寄存器即可正确识别传统 UART 模块是否存在。软件必须使用该寄存器来确定不受 DC2 寄存器支持的模块是否存在。

### 通用异步收发器外设存在寄存器 (PPUART)

基址 0x400F.E000

偏移量 0x318

类型 RO, 复位 0x0000.00FF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								P7	P6	P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	P7	RO	0x1	UART 模块 7 存在  值 描述 1 UART 模块 7 存在。 0 UART 模块 7 不存在。
6	P6	RO	0x1	UART 模块 6 存在  值 描述 1 UART 模块 6 存在。 0 UART 模块 6 不存在。
5	P5	RO	0x1	UART 模块 5 存在  值 描述 1 UART 模块 5 存在。 0 UART 模块 5 不存在。
4	P4	RO	0x1	UART 模块 4 存在  值 描述 1 UART 模块 4 存在。 0 UART 模块 4 不存在。

---

位/域	名称	类型	复位	描述
3	P3	RO	0x1	UART 模块 3 存在  值 描述 1 UART 模块 3 存在。 0 UART 模块 3 不存在。
2	P2	RO	0x1	UART 模块 2 存在  值 描述 1 UART 模块 2 存在。 0 UART 模块 2 不存在。
1	P1	RO	0x1	UART 模块 1 存在  值 描述 1 UART 模块 1 存在。 0 UART 模块 1 不存在。
0	P0	RO	0x1	UART 模块 0 存在  值 描述 1 UART 模块 0 存在。 0 UART 模块 0 不存在。



## 寄存器 25: 同步串行接口外设存在寄存器 (PPSSI), 偏移量 0x31C

PPSSI 寄存器提供关于 SSI 模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的 SSI 模块。但是, 要支持传统软件, 可使用 DC2 寄存器。读取 DC2 寄存器即可正确识别传统 SSI 模块是否存在。软件必须使用该寄存器来确定不受 DC2 寄存器支持的模块是否存在。

### 同步串行接口外设存在寄存器 (PPSSI)

基址 0x400F.E000

偏移量 0x31C

类型 RO, 复位 0x0000.000F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	P3	RO	0x1	SSI 模块 3 存在  值 描述 1 SSI 模块 3 存在。 0 SSI 模块 3 不存在。
2	P2	RO	0x1	SSI 模块 2 存在  值 描述 1 SSI 模块 2 存在。 0 SSI 模块 2 不存在。
1	P1	RO	0x1	SSI 模块 1 存在  值 描述 1 SSI 模块 1 存在。 0 SSI 模块 1 不存在。
0	P0	RO	0x1	SSI 模块 0 存在  值 描述 1 SSI 模块 0 存在。 0 SSI 模块 0 不存在。

## 寄存器 26: 内部集成电路外设存在寄存器 ( PPI2C ) , 偏移量 0x320

PPI2C 寄存器提供关于 I<sup>2</sup>C 模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的 I<sup>2</sup>C 模块。但是, 要支持传统软件, 可使用 DC2 寄存器。读取 DC2 寄存器即可正确识别传统 I<sup>2</sup>C 模块是否存在。软件必须使用该寄存器来确定不受 DC2 寄存器支持的模块是否存在。

### 内部集成电路外设存在寄存器 (PPI2C)

基址 0x400F.E000  
偏移量 0x320  
类型 RO, 复位 0x0000.003F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	P5	RO	0x1	I <sup>2</sup> C 模块 5 存在  值 描述 1 I <sup>2</sup> C 模块 5 存在。 0 I <sup>2</sup> C 模块 5 不存在。
4	P4	RO	0x1	I <sup>2</sup> C 模块 4 存在  值 描述 1 I <sup>2</sup> C 模块 4 存在。 0 I <sup>2</sup> C 模块 4 不存在。
3	P3	RO	0x1	I <sup>2</sup> C 模块 3 存在  值 描述 1 I <sup>2</sup> C 模块 3 存在。 0 I <sup>2</sup> C 模块 3 不存在。
2	P2	RO	0x1	I <sup>2</sup> C 模块 2 存在  值 描述 1 I <sup>2</sup> C 模块 2 存在。 0 I <sup>2</sup> C 模块 2 不存在。

位/域	名称	类型	复位	描述
1	P1	RO	0x1	I <sup>2</sup> C 模块 1 存在  值 描述 1 I <sup>2</sup> C 模块 1 存在。 0 I <sup>2</sup> C 模块 1 不存在。
0	P0	RO	0x1	I <sup>2</sup> C 模块 0 存在  值 描述 1 I <sup>2</sup> C 模块 0 存在。 0 I <sup>2</sup> C 模块 0 不存在。

## 寄存器 27: 通用串行总线外设存在寄存器 (PPUSB) , 偏移量 0x328

PPUSB 寄存器提供关于 USB 模块的软件信息。

**重要:** 应使用该寄存器确定是否在该微控制器上执行 USB 模块。但是, 要支持传统软件, 可使用 DC6 寄存器。读取 DC6 寄存器即可正确识别 USB 模块是否存在。

### 通用串行总线外设存在寄存器 (PPUSB)

基址 0x400FE000  
 偏移量 0x328  
 类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	RO	0x1	USB 模块存在
				值 描述
				1 USB 模块存在。
				0 USB 模块不存在。

## 寄存器 28: 控制器局域网外设存在寄存器 (PPCAN), 偏移量 0x334

PPCAN 寄存器提供关于 CAN 模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的 CAN 模块。但是, 要支持传统软件, 可使用 DC1 寄存器。读取 DC1 寄存器即可正确识别传统 CAN 模块是否存在。

## 控制器局域网外设存在寄存器 (PPCAN)

基址 0x400FE000

偏移量 0x334

类型 RO, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	RO	0x1	CAN 模块 1 存在  值 描述 1 CAN 模块 1 存在。 0 CAN 模块 1 不存在。
0	P0	RO	0x1	CAN 模块 0 存在  值 描述 1 CAN 模块 0 存在。 0 CAN 模块 0 不存在。

## 寄存器 29: 模数转换器外设存在寄存器 (PPADC), 偏移量 0x338

PPADC 寄存器提供关于 ADC 模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的 ADC 模块。但是, 要支持传统软件, 可使用 DC1 寄存器。读取 DC1 寄存器即可正确识别传统 ADC 模块是否存在。

### 模数转换器外设存在寄存器 (PPADC)

基址 0x400FE000  
偏移量 0x338  
类型 RO, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	RO	0x1	ADC 模块 1 存在  值 描述 1 ADC 模块 1 存在。 0 ADC 模块 1 不存在。
0	P0	RO	0x1	ADC 模块 0 存在  值 描述 1 ADC 模块 0 存在。 0 ADC 模块 0 不存在。

## 寄存器 30: 模拟比较器外设存在寄存器 (PPACMP), 偏移量 0x33C

PPACMP 寄存器提供关于模拟比较器模块的软件信息。

**重要:** 应使用该寄存器确定是否在该微控制器上执行模拟比较器模块。但是, 要支持传统软件, 可使用 DC2 寄存器。DC2 寄存器的读操作正确识别模拟比较器模块是否存在。请注意, 模拟比较器外设属性 (ACMP) 寄存器表示在模块中包含的模拟比较器块数量。

### 模拟比较器外设存在寄存器 (PPACMP)

基址 0x400FE000  
偏移量 0x33C  
类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	RO	0x1	模拟比较器模块存在
				值 描述
				1 模拟比较器模块存在。
				0 模拟比较器模块不存在。

### 寄存器 31: 脉宽调制器外设存在寄存器 ( PPPWM ) , 偏移量 0x340

PPPWM 寄存器提供关于 PWM 模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的 PWM 模块。但是, 要支持传统软件, 可使用 DC1 寄存器。读取 DC1 寄存器即可正确识别传统 PWM 模块是否存在。软件必须使用该寄存器来确定不受 DC1 寄存器支持的模块是否存在。

#### 脉宽调制器外设存在寄存器 (PPPWM)

基址 0x400FE000  
偏移量 0x340  
类型 RO, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	RO	0x1	PWM 模块 1 存在  值 描述 1 PWM 模块 1 存在。 0 PWM 模块 1 不存在。
0	P0	RO	0x1	PWM 模块 0 存在  值 描述 1 PWM 模块 0 存在。 0 PWM 模块 0 不存在。



## 寄存器 32: 正交编码器接口外设存在寄存器 (PPQEI) , 偏移量 0x344

PPQEI 寄存器提供关于 QEI 模块的软件信息。

**重要:** 应使用该寄存器确定在该微控制器上执行的 QEI 模块。但是, 要支持传统软件, 可使用 DC2 寄存器。DC2 寄存器的读操作正确识别传统 QEI 模块是否存在。

### 正交编码器接口外设存在寄存器 (PPQEI)

基址 0x400FE000  
偏移量 0x344  
类型 RO, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	RO	0x1	QEI 模块 1 存在  值 描述 1 QEI 模块 1 存在。 0 QEI 模块 1 不存在。
0	P0	RO	0x1	QEI 模块 0 存在  值 描述 1 QEI 模块 0 存在。 0 QEI 模块 0 不存在。

### 寄存器 33: EEPROM 外设存在寄存器 ( PEEPROM ) , 偏移量 0x358

PEEPROM 寄存器提供关于 EEPROM 模块的软件信息。

#### EEPROM 外设存在寄存器 (PEEPROM)

基址 0x400F.E000  
偏移量 0x358  
类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	RO	0x1	EEPROM 模块存在
				值 描述
				1 EEPROM 模块存在。
				0 EEPROM 模块不存在。

## 寄存器 34: 32/64 位宽通用定时器外设存在寄存器 (PPWTIMER), 偏移量 0x35C

PPWTIMER 寄存器提供关于 32/64 位宽通用定时器模块的软件信息。

## 32/64 位宽通用定时器外设存在寄存器 (PPWTIMER)

基址 0x400F.E000

偏移量 0x35C

类型 RO, 复位 0x0000.003F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	P5	RO	0x1	32/64 位宽通用定时器 5 存在  值 描述 1 32/64 位宽通用定时器模块 5 存在。 0 32/64 位宽通用定时器模块 5 不存在。
4	P4	RO	0x1	32/64 位宽通用定时器 4 存在  值 描述 1 32/64 位宽通用定时器模块 4 存在。 0 32/64 位宽通用定时器模块 4 不存在。
3	P3	RO	0x1	32/64 位宽通用定时器 3 存在  值 描述 1 32/64 位宽通用定时器模块 3 存在。 0 32/64 位宽通用定时器模块 3 不存在。
2	P2	RO	0x1	32/64 位宽通用定时器 2 存在  值 描述 1 32/64 位宽通用定时器模块 2 存在。 0 32/64 位宽通用定时器模块 2 不存在。
1	P1	RO	0x1	32/64 位宽通用定时器 1 存在  值 描述 1 32/64 位宽通用定时器模块 1 存在。 0 32/64 位宽通用定时器模块 1 不存在。

---

位/域	名称	类型	复位	描述
0	P0	RO	0x1	32/64 位宽通用定时器 0 存在  值 描述 1 32/64 位宽通用定时器模块 0 存在。 0 32/64 位宽通用定时器模块 0 不存在。

## 寄存器 35: 看门狗定时器软件复位寄存器 (SRWD), 偏移量 0x500

SRWD 寄存器为软件提供复位可用看门狗模块的功能。该寄存器特别为看门狗模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRWD 寄存器中的位。SRWD 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRWD 位完成复位过程。

从清零 SRWD 位到外设就绪时可能会有延迟。软件可以检查相应的 PRWD 位以确保正确。

**重要:** 应使用该寄存器复位看门狗模块。要支持传统软件, 可使用 SRCR0 寄存器。置位 SRCR0 寄存器中的位也会复位相应模块。通过对该 SRCR0 寄存器的写操作更改的任何位都可以在对 SRCR0 寄存器进行读操作时进行正确回读。如果软件使用该寄存器复位传统外设 (例如 Watchdog 1), 则写操作会产生正确操作, 但是该位的值不在 SRCR0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 看门狗定时器软件复位寄存器 (SRWD)

基址 0x400F.E000  
偏移量 0x500  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	看门狗定时器 1 软件复位  值 描述 1 看门狗模块 1 复位。 0 看门狗模块 1 未复位。
0	R0	R/W	0	看门狗定时器 0 软件复位  值 描述 1 看门狗模块 0 复位。 0 看门狗模块 0 未复位。

### 寄存器 36: 16/32 位通用定时器软件复位寄存器 ( SRTIMER ) , 偏移量 0x504

SRTIMER 寄存器为软件提供复位可用 16/32 位定时器模块的功能。该寄存器特别为定时器模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRTIMER 寄存器中的位。SRTIMER 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRTIMER 位完成复位过程。

从清零 SRTIMER 位到外设就绪可能会有延迟。软件可以检查相应的 PRTIMER 位以确保操作正确。

**重要:** 应使用该寄存器复位定时器模块。要支持传统软件, 可使用 SRCR1 寄存器。置位 SRCR1 寄存器中的位也会复位相应模块。通过对该 SRCR1 寄存器的写操作更改的任何位都可以在对 SRCR1 寄存器进行读操作时进行正确回读。软件必须使用该寄存器复位不处于传统寄存器中的模块。如果软件使用该寄存器复位传统外设 ( 如 Timer 1 ) , 则写操作会产生正确操作, 但是该位的值不在 SRCR1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 16/32 位通用定时器软件复位寄存器 (SRTIMER)

基址 0x400F.E000  
偏移量 0x504  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	16/32 位通用定时器 5 软件复位  值 描述 1 16/32 位通用定时器模块 5 复位。 0 16/32 位通用定时器模块 5 未复位。
4	R4	R/W	0	16/32 位通用定时器 4 软件复位  值 描述 1 16/32 位通用定时器模块 4 复位。 0 16/32 位通用定时器模块 4 未复位。

位/域	名称	类型	复位	描述
3	R3	R/W	0	16/32 位通用定时器 3 软件复位  值 描述 1 16/32 位通用定时器模块 3 复位。 0 16/32 位通用定时器模块 3 未复位。
2	R2	R/W	0	16/32 位通用定时器 2 软件复位  值 描述 1 16/32 位通用定时器模块 2 复位。 0 16/32 位通用定时器模块 2 未复位。
1	R1	R/W	0	16/32 位通用定时器 1 软件复位  值 描述 1 16/32 位通用定时器模块 1 复位。 0 16/32 位通用定时器模块 1 未复位。
0	R0	R/W	0	16/32 位通用定时器 0 软件复位  值 描述 1 16/32 位通用定时器模块 0 复位。 0 16/32 位通用定时器模块 0 未复位。

### 寄存器 37: 通用输入/输出软件复位寄存器 ( SRGPIO ) , 偏移量 0x508

SRGPIO 寄存器为软件提供复位可用 GPIO 模块的功能。该寄存器特别为 GPIO 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRGPIO 寄存器中的位。SRGPIO 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRGPIO 位完成复位过程。

从清零 SRGPIO 位到外设就绪时可能会有延迟。软件可以检查相应的 PRGPIO 位以确保正确。

**重要:** 应使用该寄存器复位 GPIO 模块。要支持传统软件, 可使用 SRCR2 寄存器。置位 SRCR2 寄存器中的位也会复位相应模块。通过对该 SRCR2 寄存器的写操作更改的任何位都可以在对 SRCR2 寄存器进行读操作时进行正确回读。软件必须使用该寄存器复位不处于传统寄存器中的模块。如果软件使用该寄存器复位传统外设 ( 如 GPIO A ), 则写操作会产生正确操作, 但是该位的值不在 SRCR2 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 通用输入/输出软件复位寄存器 (SRGPIO)

基址 0x400F.E000  
偏移量 0x508  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
13	R13	R/W	0	GPIO 端口 P 软件复位  值 描述 1 GPIO 端口 P 复位。 0 GPIO 端口 P 未复位。
12	R12	R/W	0	GPIO 端口 N 软件复位  值 描述 1 GPIO 端口 N 复位。 0 GPIO 端口 N 未复位。



位/域	名称	类型	复位	描述
11	R11	R/W	0	GPIO 端口 M 软件复位  值 描述 1 GPIO 端口 M 复位。 0 GPIO 端口 M 未复位。
10	R10	R/W	0	GPIO 端口 L 软件复位  值 描述 1 GPIO 端口 L 复位。 0 GPIO 端口 L 未复位。
9	R9	R/W	0	GPIO 端口 K 软件复位  值 描述 1 GPIO 端口 K 复位。 0 GPIO 端口 K 未复位。
8	R8	R/W	0	GPIO 端口 J 软件复位  值 描述 1 GPIO 端口 J 复位。 0 GPIO 端口 J 未复位。
7	R7	R/W	0	GPIO 端口 H 软件复位  值 描述 1 GPIO 端口 H 复位。 0 GPIO 端口 H 未复位。
6	R6	R/W	0	GPIO 端口 G 软件复位  值 描述 1 GPIO 端口 G 复位。 0 GPIO 端口 G 未复位。
5	R5	R/W	0	GPIO 端口 F 软件复位  值 描述 1 GPIO 端口 F 复位。 0 GPIO 端口 F 未复位。

位/域	名称	类型	复位	描述
4	R4	R/W	0	GPIO 端口 E 软件复位  值 描述 1 GPIO 端口 E 复位。 0 GPIO 端口 E 未复位。
3	R3	R/W	0	GPIO 端口 D 软件复位  值 描述 1 GPIO 端口 D 复位。 0 GPIO 端口 D 未复位。
2	R2	R/W	0	GPIO 端口 C 软件复位  值 描述 1 GPIO 端口 C 复位。 0 GPIO 端口 C 未复位。
1	R1	R/W	0	GPIO 端口 B 软件复位  值 描述 1 GPIO 端口 B 复位。 0 GPIO 端口 B 未复位。
0	R0	R/W	0	GPIO 端口 A 软件复位  值 描述 1 GPIO 端口 A 复位。 0 GPIO 端口 A 未复位。

**寄存器 38: 微型直接存储器访问软件复位寄存器 (SRDMA), 偏移量 0x50C**

SRDMA 寄存器为软件提供复位可用  $\mu$ DMA 模块的功能。该寄存器特别为  $\mu$ DMA 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRDMA 寄存器中的位。SRDMA 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRDMA 位完成复位过程。

从清零 SRDMA 位到外设就绪时可能会有延迟。软件可以检查相应的 PRDMA 位以确保正确。

**重要:** 应使用该寄存器复位  $\mu$ DMA 模块。要支持传统软件, 可使用 SRCR2 寄存器。置位 SRCR2 寄存器中的 UDMA 位也会复位  $\mu$ DMA 模块。如果通过对 SRCR2 寄存器的写操作置位 UDMA 位, 它可以在对 SRCR2 寄存器进行读操作时进行正确回读。如果软件使用该寄存器复位  $\mu$ DMA 模块, 则写操作会产生正确操作, 但是 UDMA 位的值不在 SRCR2 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

**微型直接存储器访问软件复位寄存器 (SRDMA)**

基址 0x400F.E000  
偏移量 0x50C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	$\mu$ DMA 模块软件复位
				值 描述
				1 $\mu$ DMA 模块复位。
				0 $\mu$ DMA 模块未复位。

### 寄存器 39: 休眠软件复位寄存器 (SRHIB) , 偏移量 0x514

SRHIB 寄存器为软件提供复位可用休眠模块的功能。该寄存器特别为休眠模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能，并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位：

1. 软件置位 SRHIB 寄存器中的位。SRHIB 位为 1 时，外设保持在复位状态。
2. 软件通过清零 SRHIB 位完成复位过程。

从清零 SRHIB 位到外设就绪时可能会有延迟。软件可以检查相应的 PRHIB 位以确保正确。

**重要:** 应使用该寄存器复位休眠模块。要支持传统软件，可使用 SRCR0 寄存器。置位 SRCR0 寄存器中的 HIB 位也会复位休眠模块。如果通过对 SRCR0 寄存器的写操作置位 HIB 位，它可以在对 SRCR0 寄存器进行读操作时进行正确回读。如果软件使用该寄存器复位休眠模块，则写操作会产生正确操作，但是 HIB 位的值不在 SRCR0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

#### 休眠软件复位寄存器 (SRHIB)

基址 0x400F.E000  
偏移量 0x514  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	休眠模块软件复位
				值 描述
				1 休眠模块复位。
				0 休眠模块未复位。

**寄存器 40: 通用异步收发器软件复位寄存器 (SRUART) , 偏移量 0x518**

SRUART 寄存器为软件提供复位可用 UART 模块的功能。该寄存器特别为 UART 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能，并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位：

1. 软件置位 SRUART 寄存器中的位。SRUART 位为 1 时，外设保持在复位状态。
2. 软件通过清零 SRUART 位完成复位过程。

从清零 SRUART 位到外设就绪时可能会有延迟。软件可以检查相应的 PRUART 位以确保正确。

**重要:** 应使用该寄存器复位 UART 模块。要支持传统软件，可使用 SRCR1 寄存器。置位 SRCR1 寄存器中的位也会复位相应模块。通过对该 SRCR1 寄存器的写操作更改的任何位都可以在对 SRCR1 寄存器进行读操作时进行正确回读。软件必须使用该寄存器复位不处于传统寄存器中的模块。如果软件使用该寄存器复位传统外设（如 UART0），则写操作会产生正确操作，但是该位的值不在 SRCR1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

**通用异步收发器软件复位寄存器 (SRUART)**

基址 0x400F.E000  
偏移量 0x518  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								R7	R6	R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	R7	R/W	0	UART 模块 7 软件复位  值 描述 1 UART 模块 7 复位。 0 UART 模块 7 未复位。
6	R6	R/W	0	UART 模块 6 软件复位  值 描述 1 UART 模块 6 复位。 0 UART 模块 6 未复位。

位/域	名称	类型	复位	描述
5	R5	R/W	0	UART 模块 5 软件复位 值 描述 1 UART 模块 5 复位。 0 UART 模块 5 未复位。
4	R4	R/W	0	UART 模块 4 软件复位 值 描述 1 UART 模块 4 复位。 0 UART 模块 4 未复位。
3	R3	R/W	0	UART 模块 3 软件复位 值 描述 1 UART 模块 3 复位。 0 UART 模块 3 未复位。
2	R2	R/W	0	UART 模块 2 软件复位 值 描述 1 UART 模块 2 复位。 0 UART 模块 2 未复位。
1	R1	R/W	0	UART 模块 1 软件复位 值 描述 1 UART 模块 1 复位。 0 UART 模块 1 未复位。
0	R0	R/W	0	UART 模块 0 软件复位 值 描述 1 UART 模块 0 复位。 0 UART 模块 0 未复位。

## 寄存器 41: 同步串行接口软件复位寄存器 (SRSSI), 偏移量 0x51C

SRSSI 寄存器为软件提供复位可用 SSI 模块的功能。该寄存器特别为 SSI 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRSSI 寄存器中的位。SRSSI 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRSSI 位完成复位过程。

从清零 SRSSI 位到外设就绪可能会有延迟。软件可以检查相应的 PRSSI 位以确保正确。

**重要:** 应使用该寄存器复位 SSI 模块。要支持传统软件, 可使用 SRCR1 寄存器。置位 SRCR1 寄存器中的位也会复位相应模块。通过对该 SRCR1 寄存器的写操作更改的任何位都可以在对 SRCR1 寄存器进行读操作时进行正确回读。软件必须使用该寄存器复位不处于传统寄存器中的模块。如果软件使用该寄存器复位传统外设 (如 SSI0), 则写操作会产生正确操作, 但是该位的值不在 SRCR1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 同步串行接口软件复位寄存器 (SRSSI)

基址 0x400F.E000  
偏移量 0x51C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	R3	R/W	0	SSI 模块 3 软件复位  值 描述 1 SSI 模块 3 复位。 0 SSI 模块 3 未复位。
2	R2	R/W	0	SSI 模块 2 软件复位  值 描述 1 SSI 模块 2 复位。 0 SSI 模块 2 未复位。

---

位/域	名称	类型	复位	描述
1	R1	R/W	0	SSI 模块 1 软件复位 值 描述 1 SSI 模块 1 复位。 0 SSI 模块 1 未复位。
0	R0	R/W	0	SSI 模块 0 软件复位 值 描述 1 SSI 模块 0 复位。 0 SSI 模块 0 未复位。



## 寄存器 42: 内部集成电路软件复位寄存器 (SRI2C), 偏移量 0x520

SRI2C 寄存器为软件提供复位可用 I<sup>2</sup>C 模块的功能。该寄存器特别为 I<sup>2</sup>C 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRI2C 寄存器中的位。SRI2C 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRI2C 位完成复位过程。

从清零 SRI2C 位到外设就绪时可能会有延迟。软件可以检查相应的 PRI2C 位以确保正确。

**重要:** 应使用该寄存器复位 I<sup>2</sup>C 模块。要支持传统软件, 可使用 SRCR1 寄存器。置位 SRCR1 寄存器中的位也会复位相应模块。通过对该 SRCR1 寄存器的写操作更改的任何位都可以在对 SRCR1 寄存器进行读操作时进行正确回读。软件必须使用该寄存器复位不处于传统寄存器中的模块。如果软件使用该寄存器复位传统外设 (如 I2C0), 则写操作会产生正确操作, 但是该位的值不在 SRCR1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 内部集成电路软件复位寄存器 (SRI2C)

基址 0x400F.E000

偏移量 0x520

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	I <sup>2</sup> C 模块 5 软件复位  值 描述 1 I <sup>2</sup> C 模块 5 复位。 0 I <sup>2</sup> C 模块 5 未复位。
4	R4	R/W	0	I <sup>2</sup> C 模块 4 软件复位  值 描述 1 I <sup>2</sup> C 模块 4 复位。 0 I <sup>2</sup> C 模块 4 未复位。

---

位/域	名称	类型	复位	描述
3	R3	R/W	0	I <sup>2</sup> C 模块 3 软件复位  值 描述 1 I <sup>2</sup> C 模块 3 复位。 0 I <sup>2</sup> C 模块 3 未复位。
2	R2	R/W	0	I <sup>2</sup> C 模块 2 软件复位  值 描述 1 I <sup>2</sup> C 模块 2 复位。 0 I <sup>2</sup> C 模块 2 未复位。
1	R1	R/W	0	I <sup>2</sup> C 模块 1 软件复位  值 描述 1 I <sup>2</sup> C 模块 1 复位。 0 I <sup>2</sup> C 模块 1 未复位。
0	R0	R/W	0	I <sup>2</sup> C 模块 0 软件复位  值 描述 1 I <sup>2</sup> C 模块 0 复位。 0 I <sup>2</sup> C 模块 0 未复位。

**寄存器 43: 通用串行总线软件复位寄存器 (SRUSB), 偏移量 0x528**

SRUSB 寄存器为软件提供复位可用 USB 模块的功能。该寄存器特别为 USB 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRUSB 寄存器中的位。SRUSB 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRUSB 位完成复位过程。

从清零 SRUSB 位到外设就绪时可能会有延迟。软件可以检查相应的 PRUSB 位以确保正确。

**重要:** 应使用该寄存器复位 USB 模块。要支持传统软件, 可使用 SRCR2 寄存器。置位 SRCR2 寄存器中的 USB0 位也会复位 USB 模块。如果通过对 SRCR2 寄存器的写操作置位 USB0 位, 它可以在对 SRCR2 寄存器进行读操作时进行正确回读。如果软件使用该寄存器复位 USB 模块, 则写操作会产生正确操作, 但是 USB0 位的值不在 SRCR2 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

**通用串行总线软件复位寄存器 (SRUSB)**

基址 0x400F.E000  
偏移量 0x528  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	USB 模块软件复位
				值 描述
				1 USB 模块复位。
				0 USB 模块未复位。

### 寄存器 44: 控制器局域网软件复位寄存器 ( SRCAN ) , 偏移量 0x534

SRCAN 寄存器为软件提供复位可用 CAN 模块的功能。该寄存器特别为 CAN 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRCAN 寄存器中的位。SRCAN 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRCAN 位完成复位过程。

从清零 SRCAN 位到外设就绪时可能会有延迟。软件可以检查相应的 PRCAN 位以确保正确。

**重要:** 应使用该寄存器复位 CAN 模块。要支持传统软件, 可使用 SRCR0 寄存器。置位 SRCR0 寄存器中的位也会复位相应模块。通过对该 SRCR0 寄存器的写操作更改的任何位都可以在对 SRCR0 寄存器进行读操作时进行正确回读。如果软件使用该寄存器复位传统外设 ( 如 CAN0 ) , 则写操作会产生正确操作, 但是该位的值不在 SRCR0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 控制器局域网软件复位寄存器 (SRCAN)

基址 0x400F.E000  
偏移量 0x534  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	CAN 模块 1 软件复位  值 描述 1 CAN 模块 1 复位。 0 CAN 模块 1 未复位。
0	R0	R/W	0	CAN 模块 0 软件复位  值 描述 1 CAN 模块 0 复位。 0 CAN 模块 0 未复位。

## 寄存器 45: 模数转换器软件复位寄存器 (SRADC), 偏移量 0x538

SRADC 寄存器为软件提供复位可用 ADC 模块的功能。该寄存器特别为 ADC 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRADC 寄存器中的位。SRADC 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRADC 位完成复位过程。

从清零 SRADC 位到外设就绪时可能会有延迟。软件可以检查相应的 PRADC 位以确保正确。

**重要:** 应使用该寄存器复位 ADC 模块。要支持传统软件, 可使用 SRCR0 寄存器。置位 SRCR0 寄存器中的位也会复位相应模块。通过对该 SRCR0 寄存器的写操作更改的任何位都可以在对 SRCR0 寄存器进行读操作时进行正确回读。如果软件使用该寄存器复位传统外设 (如 ADC0), 则写操作会产生正确操作, 但是该位的值不在 SRCR0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 模数转换器软件复位寄存器 (SRADC)

基址 0x400F.E000  
偏移量 0x538  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	ADC 模块 1 软件复位  值 描述 1 ADC 模块 1 复位。 0 ADC 模块 1 未复位。
0	R0	R/W	0	ADC 模块 0 软件复位  值 描述 1 ADC 模块 0 复位。 0 ADC 模块 0 未复位。

### 寄存器 46: 模数比较器软件复位寄存器 ( SRACMP ) , 偏移量 0x53C

SRACMP 寄存器为软件提供复位可用模拟比较器模块的功能。该寄存器特别为模拟比较器模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对模块进行复位:

1. 软件置位 SRACMP 寄存器中的位。SRACMP 位为 1 时, 模块保持在复位状态。
2. 软件通过清零 SRACMP 位完成复位过程。

从清零 SRACMP 位到模块就绪时可能会有延迟。软件可以检查相应的 PRACMP 位以确保正确。

**重要:** 应使用该寄存器复位模拟比较器模块。要支持传统软件, 可使用 SRCR1 寄存器。置位 SRCR0 寄存器中的任何 COMPn 位也会复位模拟比较器模块。如果通过对 SRCR1 寄存器的写操作置位任何 COMPn 位, 它可以在对 SRCR0 寄存器进行读操作时进行正确回读。如果软件使用该寄存器复位模拟比较器模块, 则写操作会产生正确操作, 但是 SRCR1 寄存器的 COMPn 位不反映 R0 的值。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 模数比较器软件复位寄存器 (SRACMP)

基址 0x400F.E000  
偏移量 0x53C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	模拟比较器模块 0 软件复位
				值 描述
				1 模拟比较器模块复位。
				0 模拟比较器模块未复位。

**寄存器 47: 脉宽调制器软件复位寄存器 (SRPWM), 偏移量 0x540**

SRPWM 寄存器为软件提供复位可用 PWM 模块的功能。该寄存器特别为 PWM 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRPWM 寄存器中的位。SRPWM 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRPWM 位完成复位过程。

从清零 SRPWM 位到外设就绪时可能会有延迟。软件可以检查相应的 PRPWM 位以确保正确。

**重要:** 应使用该寄存器复位 PWM 模块。要支持传统软件, 可使用 SRCR0 寄存器。置位 SRCR0 寄存器中的 PWM 位也会复位 PWM0 模块。如果通过对 SRCR0 寄存器的写操作置位 PWM 位, 它可以在对 SRCR0 寄存器进行读操作时进行正确回读。软件必须使用该寄存器复位不处于传统寄存器中的 PWM1。如果软件使用该寄存器复位 PWM0, 则写操作会产生正确操作, 但是该位的值不在 SRCR0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

**脉宽调制器软件复位寄存器 (SRPWM)**

基址 0x400F.E000  
偏移量 0x540  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	PWM 模块 1 软件复位  值 描述 1 PWM 模块 1 复位。 0 PWM 模块 1 未复位。
0	R0	R/W	0	PWM 模块 0 软件复位  值 描述 1 PWM 模块 0 复位。 0 PWM 模块 0 未复位。

### 寄存器 48: 正交编码器接口软件复位寄存器 (SRQEI) , 偏移量 0x544

SRQEI 寄存器为软件提供复位可用 QEI 模块的功能。该寄存器特别为 QEI 模块提供与传统软件复位控制 n SRCRn 寄存器相同的功能, 并且具有与相应 SRCRn 位相同的位极性。

通过软件使用简单的两步过程对外设进行复位:

1. 软件置位 SRQEI 寄存器中的位。SRQEI 位为 1 时, 外设保持在复位状态。
2. 软件通过清零 SRQEI 位完成复位过程。

从清零 SRQEI 位到外设就绪时可能会有延迟。软件可以检查相应的 PRQEI 位以确保正确。

**重要:** 应使用该寄存器复位 QEI 模块。要支持传统软件, 可使用 SRCR1 寄存器。置位 SRCR1 寄存器中的位也会复位相应模块。通过对该 SRCR1 寄存器的写操作更改的任何位都可以在对 SRCR1 寄存器进行读操作时进行正确回读。如果软件使用该寄存器复位传统外设 (如 QEI0), 则写操作会产生正确操作, 但是该位的值不在 SRCR1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 正交编码器接口软件复位寄存器 (SRQEI)

基址 0x400F.E000  
偏移量 0x544  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	QEI 模块 1 软件复位  值 描述 1 QEI 模块 1 复位。 0 QEI 模块 1 未复位。
0	R0	R/W	0	QEI 模块 0 软件复位  值 描述 1 QEI 模块 0 复位。 0 QEI 模块 0 未复位。



**寄存器 49: 32/64 位宽通用定时器软件复位寄存器 (SRWTIMER)，偏移量 0x55C**

SRWTIMER 寄存器为软件提供复位可用 32/64 位宽定时器模块的功能。

通过软件使用简单的两步过程对外设进行复位：

1. 软件置位 SRWTIMER 寄存器中的位。SRWTIMER 位为 1 时，外设保持在复位状态。
2. 软件通过清零 SRWTIMER 位完成复位过程。

从清零 SRWTIMER 位到外设就绪可能会有延迟。软件可以检查相应的 SRWTIMER 位以确保操作正确。

**32/64 位宽通用定时器软件复位寄存器 (SRWTIMER)**

基址 0x400F.E000  
偏移量 0x55C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	32/64 位宽通用定时器 5 软件复位  值 描述 1 32/64 位宽通用定时器模块 5 复位。 0 32/64 位宽通用定时器模块 5 未复位。
4	R4	R/W	0	32/64 位宽通用定时器 4 软件复位  值 描述 1 32/64 位宽通用定时器模块 4 复位。 0 32/64 位宽通用定时器模块 4 未复位。
3	R3	R/W	0	32/64 位宽通用定时器 3 软件复位  值 描述 1 32/64 位宽通用定时器模块 3 复位。 0 32/64 位宽通用定时器模块 3 未复位。

---

位/域	名称	类型	复位	描述
2	R2	R/W	0	32/64 位宽通用定时器 2 软件复位  值 描述 1 32/64 位宽通用定时器模块 2 复位。 0 32/64 位宽通用定时器模块 2 未复位。
1	R1	R/W	0	32/64 位宽通用定时器 1 软件复位  值 描述 1 32/64 位宽通用定时器模块 1 复位。 0 32/64 位宽通用定时器模块 1 未复位。
0	R0	R/W	0	32/64 位宽通用定时器 0 软件复位  值 描述 1 32/64 位宽通用定时器模块 0 复位。 0 32/64 位宽通用定时器模块 0 未复位。

## 寄存器 50: 看门狗定时器运行模式时钟门控制寄存器 (RCGCWD), 偏移量 0x600

RCGCWD 寄存器为软件提供启用和禁用运行模式中的看门狗模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制看门狗模块的计时。要支持传统软件, 可使用 RCGC0 寄存器。对 RCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC0 寄存器的写操作更改的任何位都可以通过对 RCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 (如看门狗 0) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 RCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 看门狗定时器运行模式时钟门控制寄存器 (RCGCWD)

基址 0x400F.E000  
偏移量 0x600  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	看门狗定时器 1 运行模式时钟门控制  值 描述 1 启用运行模式中的看门狗模块 1 并提供时钟。 0 看门狗模块 1 禁用。
0	R0	R/W	0	看门狗定时器 0 运行模式时钟门控制  值 描述 1 启用运行模式中的看门狗模块 0 并提供时钟。 0 看门狗模块 0 禁用。

### 寄存器 51: 16/32 位通用定时器运行模式时钟门控制寄存器 ( RCGCTIMER ) , 偏移量 0x604

RCGCTIMER 寄存器为软件提供启用和禁用运行模式中的 16/32 位定时器模块的功能。在启用时，为模块提供一个时钟并允许对模块寄存器的访问。在禁用时，时钟禁用以节能，并且对模块寄存器的访问将产生总线错误。该寄存器特别为定时器模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能，并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制定时器模块的计时。要支持传统软件，可使用 RCGC1 寄存器。对 RCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC1 寄存器的写操作更改的任何位都可以通过对 RCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 Timer 0 ) 进行写操作，则写操作会产生正确操作，但是该位的值不在 RCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

#### 16/32 位通用定时器运行模式时钟门控制寄存器 (RCGCTIMER)

基址 0x400F.E000  
偏移量 0x604  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	16/32 位通用定时器 5 运行模式时钟门控制  值 描述 1 启用运行模式中的 16/32 位通用定时器模块 5 并提供时钟。 0 16/32 位通用定时器模块 5 禁用。
4	R4	R/W	0	16/32 位通用定时器 4 运行模式时钟门控制  值 描述 1 启用运行模式中的 16/32 位通用定时器模块 4 并提供时钟。 0 16/32 位通用定时器模块 4 禁用。
3	R3	R/W	0	16/32 位通用定时器 3 运行模式时钟门控制  值 描述 1 启用运行模式中的 16/32 位通用定时器模块 3 并提供时钟。 0 16/32 位通用定时器模块 3 禁用。

位/域	名称	类型	复位	描述
2	R2	R/W	0	16/32 位通用定时器 2 运行模式时钟门控控制  值 描述 1 启用运行模式中的 16/32 位通用定时器模块 2 并提供时钟。 0 16/32 位通用定时器模块 2 禁用。
1	R1	R/W	0	16/32 位通用定时器 1 运行模式时钟门控控制  值 描述 1 启用运行模式中的 16/32 位通用定时器模块 1 并提供时钟。 0 16/32 位通用定时器模块 1 禁用。
0	R0	R/W	0	16/32 位通用定时器 0 运行模式时钟门控控制  值 描述 1 启用运行模式中的 16/32 位通用定时器模块 0 并提供时钟。 0 16/32 位通用定时器模块 0 禁用。

## 寄存器 52: 通用输入/输出运行模式时钟门控制寄存器 ( RCGCGPIO ) , 偏移量 0x608

RCGCGPIO 寄存器为软件提供启用和禁用运行模式中的 GPIO 模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 GPIO 模块的计时。要支持传统软件, 可使用 RCGC2 寄存器。对 RCGC2 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC2 寄存器的写操作更改的任何位都可以通过对 RCGC2 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器写入传统外设 ( 如 GPIO A ) , 则写操作会产生正确操作, 但是该位的值不在 RCGC2 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 通用输入/输出运行模式时钟门控制寄存器 (RCGCGPIO)

基址 0x400F.E000  
偏移量 0x608  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
13	R13	R/W	0	GPIO 端口 P 运行模式时钟门控制  值 描述 1 启用运行模式中的 GPIO 端口 P 并提供时钟。 0 GPIO 端口 P 禁用。
12	R12	R/W	0	GPIO 端口 N 运行模式时钟门控制  值 描述 1 启用运行模式中的 GPIO 端口 N 并提供时钟。 0 GPIO 端口 N 禁用。
11	R11	R/W	0	GPIO 端口 M 运行模式时钟门控制  值 描述 1 启用运行模式中的 GPIO 端口 M 并提供时钟。 0 GPIO 端口 M 禁用。

位/域	名称	类型	复位	描述
10	R10	R/W	0	GPIO 端口 L 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 L 并提供时钟。 0 GPIO 端口 L 禁用。
9	R9	R/W	0	GPIO 端口 K 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 K 并提供时钟。 0 GPIO 端口 K 禁用。
8	R8	R/W	0	GPIO 端口 J 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 J 并提供时钟。 0 GPIO 端口 J 禁用。
7	R7	R/W	0	GPIO 端口 H 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 H 并提供时钟。 0 GPIO 端口 H 禁用。
6	R6	R/W	0	GPIO 端口 G 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 G 并提供时钟。 0 GPIO 端口 G 禁用。
5	R5	R/W	0	GPIO 端口 F 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 F 并提供时钟。 0 GPIO 端口 F 禁用。
4	R4	R/W	0	GPIO 端口 E 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 E 并提供时钟。 0 GPIO 端口 E 禁用。

---

位/域	名称	类型	复位	描述
3	R3	R/W	0	GPIO 端口 D 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 D 并提供时钟。 0 GPIO 端口 D 禁用。
2	R2	R/W	0	GPIO 端口 C 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 C 并提供时钟。 0 GPIO 端口 C 禁用。
1	R1	R/W	0	GPIO 端口 B 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 B 并提供时钟。 0 GPIO 端口 B 禁用。
0	R0	R/W	0	GPIO 端口 A 运行模式时钟门控控制  值 描述 1 启用运行模式中的 GPIO 端口 A 并提供时钟。 0 GPIO 端口 A 禁用。



## 寄存器 53: 微型直接存储器访问运行模式时钟门控制寄存器 (RCGCDMA), 偏移量 0x60C

RCGCDMA 寄存器为软件提供启用和禁用运行模式中的  $\mu$ DMA 模块的功能。在启用时, 为模块提供时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制  $\mu$ DMA 模块的计时。要支持传统软件, 可使用 RCGC2 寄存器。对 RCGC2 寄存器中 UDMA 位的写操作还对该寄存器中的 R0 位执行写操作。如果通过对 RCGC2 寄存器的写操作更改 UDMA 位, 它可以通过读 RCGC2 寄存器进行正确回读。如果软件使用该寄存器控制  $\mu$ DMA 模块, 则该写操作会产生正确操作, 但是 RCGC2 寄存器中的 UDMA 位不会反映 R0 位的值。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 微型直接存储器访问运行模式时钟门控制寄存器 (RCGCDMA)

基址 0x400F.E000  
偏移量 0x60C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	$\mu$ DMA 模块运行模式时钟门控制
	值 描述			
	1 启用运行模式中的 $\mu$ DMA 模块 并提供时钟。			
	0 $\mu$ DMA 模块禁用。			

### 寄存器 54: 休眠运行模式时钟门控控制寄存器 ( RCGCHIB ) , 偏移量 0x614

RCGCHIB 寄存器为软件提供启用和禁用运行模式中休眠模块的功能。在启用时, 为模块提供时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制休眠模块的计时。要支持传统软件, 可使用 RCGC0 寄存器。对 RCGC0 寄存器中 HIB 位的写操作还对该寄存器中的 R0 位执行写操作。如果通过对 RCGC0 寄存器的写操作更改 HIB 位, 它可以通过读 RCGC0 寄存器进行正确回读。如果软件使用该寄存器控制休眠模块, 则该写操作会产生正确操作, 但是 RCGC0 寄存器中的 HIB 位不会反映 R0 位的值。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 休眠运行模式时钟门控控制寄存器 (RCGCHIB)

基址 0x400F.E000  
偏移量 0x614  
类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	1	休眠模块运行模式时钟门控控制
				<b>值 描述</b>
				1 启用运行模式中的休眠模块并提供时钟。
				0 休眠模块禁用。

## 寄存器 55: 通用异步收发器运行模式时钟门控制寄存器 ( RCGCUART ) , 偏移量 0x618

RCGCUART 寄存器为软件提供启用和禁用运行模式中的 UART 模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 UART 模块的计时。要支持传统软件, 可使用 RCGC1 寄存器。对 RCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC1 寄存器的写操作更改的任何位都可以通过对 RCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 UART0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 RCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 通用异步收发器运行模式时钟门控制寄存器 (RCGCUART)

基址 0x400F.E000  
偏移量 0x618  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								R7	R6	R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	R7	R/W	0	UART 模块 7 运行模式时钟门控制  值 描述 1 启用运行模式中的 UART 模块 7 并提供时钟。 0 UART 模块 7 禁用。
6	R6	R/W	0	UART 模块 6 运行模式时钟门控制  值 描述 1 启用运行模式中的 UART 模块 6 并提供时钟。 0 UART 模块 6 禁用。
5	R5	R/W	0	UART 模块 5 运行模式时钟门控制  值 描述 1 启用运行模式中的 UART 模块 5 并提供时钟。 0 UART 模块 5 禁用。

位/域	名称	类型	复位	描述
4	R4	R/W	0	UART 模块 4 运行模式时钟门控控制  值 描述 1 启用运行模式中的 UART 模块 4 并提供时钟。 0 UART 模块 4 禁用。
3	R3	R/W	0	UART 模块 3 运行模式时钟门控控制  值 描述 1 启用运行模式中的 UART 模块 3 并提供时钟。 0 UART 模块 3 禁用。
2	R2	R/W	0	UART 模块 2 运行模式时钟门控控制  值 描述 1 启用运行模式中的 UART 模块 2 并提供时钟。 0 UART 模块 2 禁用。
1	R1	R/W	0	UART 模块 1 运行模式时钟门控控制  值 描述 1 启用运行模式中的 UART 模块 1 并提供时钟。 0 UART 模块 1 禁用。
0	R0	R/W	0	UART 模块 0 运行模式时钟门控控制  值 描述 1 启用运行模式中的 UART 模块 0 并提供时钟。 0 UART 模块 0 禁用。

## 寄存器 56: 同步串行接口运行模式时钟门控制寄存器 ( RCGCSSI ) , 偏移量 0x61C

RCGCSSI 寄存器为软件提供启用和禁用运行模式中的 SSI 模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 SSI 模块的计时。要支持传统软件, 可使用 RCGC1 寄存器。对 RCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC1 寄存器的写操作更改的任何位都可以通过对 RCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 SSI0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 RCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 同步串行接口运行模式时钟门控制寄存器 (RCGCSSI)

基址 0x400F.E000

偏移量 0x61C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	R3	R/W	0	SSI 模块 3 运行模式时钟门控制  值 描述 1 启用运行模式中的 SSI 模块 3 并提供时钟。 0 SSI 模块 3 禁用。
2	R2	R/W	0	SSI 模块 2 运行模式时钟门控制  值 描述 1 启用运行模式中的 SSI 模块 2 并提供时钟。 0 SSI 模块 2 禁用。
1	R1	R/W	0	SSI 模块 1 运行模式时钟门控制  值 描述 1 启用运行模式中的 SSI 模块 1 并提供时钟。 0 SSI 模块 1 禁用。

---

位/域	名称	类型	复位	描述
0	R0	R/W	0	SSI 模块 0 运行模式时钟门控控制  值 描述 1 启用运行模式中的 SSI 模块 0 并提供时钟。 0 SSI 模块 0 禁用。

## 寄存器 57: 内部集成电路运行模式时钟门控制寄存器 (RCGCI2C), 偏移量 0x620

RCGCI2C 寄存器为软件提供启用和禁用运行模式中的 I<sup>2</sup>C 模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 I<sup>2</sup>C 模块的计时。要支持传统软件, 可使用 RCGC1 寄存器。对 RCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC1 寄存器的写操作更改的任何位都可以通过对 RCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 (如 I2C0) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 RCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 内部集成电路运行模式时钟门控制寄存器 (RCGCI2C)

基址 0x400F.E000  
偏移量 0x620  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	I <sup>2</sup> C 模块 5 运行模式时钟门控制  值 描述 1 启用运行模式中的 I <sup>2</sup> C 模块 5 并提供时钟。 0 I <sup>2</sup> C 模块 5 禁用。
4	R4	R/W	0	I <sup>2</sup> C 模块 4 运行模式时钟门控制  值 描述 1 启用运行模式中的 I <sup>2</sup> C 模块 4 并提供时钟。 0 I <sup>2</sup> C 模块 4 禁用。
3	R3	R/W	0	I <sup>2</sup> C 模块 3 运行模式时钟门控制  值 描述 1 启用运行模式中的 I <sup>2</sup> C 模块 3 并提供时钟。 0 I <sup>2</sup> C 模块 3 禁用。

---

位/域	名称	类型	复位	描述
2	R2	R/W	0	I <sup>2</sup> C 模块 2 运行模式时钟门控控制  值 描述 1 启用运行模式中的 I <sup>2</sup> C 模块 2 并提供时钟。 0 I <sup>2</sup> C 模块 2 禁用。
1	R1	R/W	0	I <sup>2</sup> C 模块 1 运行模式时钟门控控制  值 描述 1 启用运行模式中的 I <sup>2</sup> C 模块 1 并提供时钟。 0 I <sup>2</sup> C 模块 1 禁用。
0	R0	R/W	0	I <sup>2</sup> C 模块 0 运行模式时钟门控控制  值 描述 1 启用运行模式中的 I <sup>2</sup> C 模块 0 并提供时钟。 0 I <sup>2</sup> C 模块 0 禁用。



## 寄存器 58: 通用串行总线运行模式时钟门控制寄存器 (RCGCUSB), 偏移量 0x628

RCGCUSB 寄存器为软件提供启用和禁用运行模式中的 USB 模块的功能。在启用时, 为模块提供时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 USB 模块的计时。要支持传统软件, 可使用 RCGC2 寄存器。对 RCGC2 寄存器中 USB0 位的写操作还对该寄存器中的 R0 位执行写操作。如果通过对 RCGC2 寄存器的写操作更改 USB0 位, 它可以通过读 RCGC2 寄存器进行正确回读。如果软件使用该寄存器控制 USB 模块, 则该写操作会产生正确操作, 但是 RCGC2 寄存器中的 USB0 位不会反映 R0 位的值。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 通用串行总线运行模式时钟门控制寄存器 (RCGCUSB)

基址 0x400F.E000  
偏移量 0x628  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	USB 模块运行模式时钟门控制
				<b>值 描述</b>
				1 启用运行模式中的 USB 模块 并提供时钟。
				0 USB 模块禁用。

### 寄存器 59: 控制器局域网运行模式时钟门控制寄存器 ( RCGCCAN ) , 偏移量 0x634

RCGCCAN 寄存器为软件提供启用和禁用运行模式中的 CAN 模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 CAN 模块的计时。要支持传统软件, 可使用 RCGC0 寄存器。对 RCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC0 寄存器的写操作更改的任何位都可以通过对 RCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如 CAN0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 RCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 控制器局域网运行模式时钟门控制寄存器 (RCGCCAN)

基址 0x400F.E000  
偏移量 0x634  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	CAN 模块 1 运行模式时钟门控制  值 描述 1 启用运行模式中的 CAN 模块 1 并提供时钟。 0 CAN 模块 1 禁用。
0	R0	R/W	0	CAN 模块 0 运行模式时钟门控制  值 描述 1 启用运行模式中的 CAN 模块 0 并提供时钟。 0 CAN 模块 0 禁用。

**寄存器 60: 模数转换器运行模式时钟门控制寄存器 ( RCGCADC ) , 偏移量 0x638**

RCGCADC 寄存器为软件提供启用和禁用运行模式中的 ADC 模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 ADC 模块的计时。要支持传统软件, 可使用 RCGC0 寄存器。对 RCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC0 寄存器的写操作更改的任何位都可以通过对 RCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如 ADC0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 RCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

**模数转换器运行模式时钟门控制寄存器 (RCGCADC)**

基址 0x400F.E000

偏移量 0x638

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	ADC 模块 1 运行模式时钟门控制  值 描述 1 启用运行模式中的 ADC 模块 1 并提供时钟。 0 ADC 模块 1 禁用。
0	R0	R/W	0	ADC 模块 0 运行模式时钟门控制  值 描述 1 启用运行模式中的 ADC 模块 0 并提供时钟。 0 ADC 模块 0 禁用。

### 寄存器 61: 模拟比较器运行模式时钟门控控制寄存器 ( RCGCACMP ) , 偏移量 0x63C

RCGCACMP 寄存器为软件提供启用和禁用运行模式中模拟比较器模块的功能。在启用时, 为模块提供时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制模拟比较器模块的计时。要支持传统软件, 可使用 RCGC1 寄存器。对 RCGC1 寄存器中 COMPn 位的写操作还对该寄存器中的 R0 位执行写操作。如果通过对 RCGC1 寄存器的写操作置位任何 COMPn 位, 它可以在对 RCGC1 寄存器进行读操作时进行正确回读。如果软件使用该寄存器更改模拟比较器模块的计时, 则写操作会产生正确操作, 但是 RCGC1 寄存器的 COMPn 位不反映值 R0。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 模拟比较器运行模式时钟门控控制寄存器 (RCGCACMP)

基址 0x400F.E000  
偏移量 0x63C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	模拟比较器模块 0 运行模式时钟门控控制
	值 描述			
	1			启用运行模式中的模拟比较器模块并提供时钟。
	0			模拟比较器模块禁用。

## 寄存器 62: 脉宽调制器运行模式时钟门控制寄存器 ( RCGCPWM ) , 偏移量 0x640

RCGCPWM 寄存器为软件提供启用和禁用运行模式中的 PWM 模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 PWM 模块的计时。要支持传统软件, 可使用 RCGC0 寄存器。对 RCGC0 寄存器中 PWM 位的写操作还对该寄存器中的 R0 位执行写操作。如果通过对 RCGC0 寄存器的写操作更改 PWM 位, 它可以通过读 RCGC0 寄存器进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器写 R0, 则写操作会产生正确操作, 但是该位的值不在 RCGC0 寄存器的 PWM 位中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 脉宽调制器运行模式时钟门控制寄存器 (RCGCPWM)

基址 0x400F.E000  
偏移量 0x640  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	PWM 模块 1 运行模式时钟门控制  值 描述 1 启用运行模式中的 PWM 模块 1 并提供时钟。 0 PWM 模块 1 禁用。
0	R0	R/W	0	PWM 模块 0 运行模式时钟门控制  值 描述 1 启用运行模式中的 PWM 模块 0 并提供时钟。 0 PWM 模块 0 禁用。

### 寄存器 63: 正交编码器接口运行模式时钟门控制寄存器 ( RCGCQEI ) , 偏移量 0x644

RCGCQEI 寄存器为软件提供启用和禁用运行模式中的 QEI 模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为看门狗模块提供与传统运行模式时钟门控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 QEI 模块的计时。要支持传统软件, 可使用 RCGC1 寄存器。对 RCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 RCGC1 寄存器的写操作更改的任何位都可以通过对 RCGC1 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如 QEI0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 RCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 正交编码器接口运行模式时钟门控制寄存器 (RCGCQEI)

基址 0x400F.E000  
偏移量 0x644  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	QEI 模块 1 运行模式时钟门控制  值 描述 1 启用运行模式中的 QEI 模块 1 并提供时钟。 0 QEI 模块 1 禁用。
0	R0	R/W	0	QEI 模块 0 运行模式时钟门控制  值 描述 1 启用运行模式中的 QEI 模块 0 并提供时钟。 0 QEI 模块 0 禁用。

## 寄存器 64: 32/64 位宽通用定时器运行模式时钟门控控制寄存器 (RCGCWTIMER), 偏移量 0x65C

RCGCWTIMER 寄存器为软件提供启用和禁用运行模式中的 32/64 位定时器模块的功能。在启用时, 为模块提供一个时钟并允许对模块寄存器的访问。在禁用时, 时钟禁用以节能, 并且对模块寄存器的访问将产生总线错误。该寄存器特别为定时器模块提供与传统运行模式时钟门控控制寄存器 n RCGCn 寄存器相同的功能, 并且具有与相应 RCGCn 位相同的位极性。

### 32/64 位宽通用定时器运行模式时钟门控控制寄存器 (RCGCWTIMER)

基址 0x400F.E000

偏移量 0x65C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	32/64 位宽通用定时器 5 运行模式时钟门控控制  值 描述 1 启用运行模式中的 32/64 位宽通用定时器模块 5 并提供时钟。 0 32/64 位宽通用定时器模块 5 禁用。
4	R4	R/W	0	32/64 位宽通用定时器 4 运行模式时钟门控控制  值 描述 1 启用运行模式中的 32/64 位宽通用定时器模块 4 并提供时钟。 0 32/64 位宽通用定时器模块 4 禁用。
3	R3	R/W	0	32/64 位宽通用定时器 3 运行模式时钟门控控制  值 描述 1 启用运行模式中的 32/64 位宽通用定时器模块 3 并提供时钟。 0 32/64 位宽通用定时器模块 3 禁用。
2	R2	R/W	0	32/64 位宽通用定时器 2 运行模式时钟门控控制  值 描述 1 启用运行模式中的 32/64 位宽通用定时器模块 2 并提供时钟。 0 32/64 位宽通用定时器模块 2 禁用。

位/域	名称	类型	复位	描述
1	R1	R/W	0	<p>32/64 位宽通用定时器 1 运行模式时钟门控控制</p> <p>值 描述</p> <p>1 启用运行模式中的 32/64 位宽通用定时器模块 1 并提供时钟。</p> <p>0 32/64 位宽通用定时器模块 1 禁用。</p>
0	R0	R/W	0	<p>32/64 位宽通用定时器 0 运行模式时钟门控控制</p> <p>值 描述</p> <p>1 启用运行模式中的 32/64 位宽通用定时器模块 0 并提供时钟。</p> <p>0 32/64 位宽通用定时器模块 0 禁用。</p>



## 寄存器 65: 看门狗定时器睡眠模式时钟门控制寄存器 ( SCGCWD ) , 偏移量 0x700

SCGCWD 寄存器为软件提供启用和禁用睡眠模式中的看门狗模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能, 并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制看门狗模块的计时。要支持传统软件, 可使用 SCGC0 寄存器。对 SCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC0 寄存器的写操作更改的任何位都可以通过对 SCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如看门狗 0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 SCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 看门狗定时器睡眠模式时钟门控制寄存器 (SCGCWD)

基址 0x400F.E000  
偏移量 0x700  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	S1	R/W	0	看门狗定时器 1 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的看门狗模块 1 并提供时钟。 0 看门狗模块 1 禁用。
0	S0	R/W	0	看门狗定时器 0 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的看门狗模块 0 并提供时钟。 0 看门狗模块 0 禁用。

### 寄存器 66: 16/32 位通用定时器睡眠模式时钟门控制寄存器 ( SCGCTIMER ) , 偏移量 0x704

SCGCTIMER 寄存器为软件提供启用和禁用睡眠模式中的 16/32 位定时器模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为定时器模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能，并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制定时器模块的计时。要支持传统软件，可使用 SCGC1 寄存器。对 SCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC1 寄存器的写操作更改的任何位都可以通过对 SCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 Timer 0 ) 进行写操作，则写操作会产生正确操作，但是该位的值不在 SCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

#### 16/32 位通用定时器睡眠模式时钟门控制寄存器 (SCGCTIMER)

基址 0x400F.E000  
偏移量 0x704  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											S5	S4	S3	S2	S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
5	S5	R/W	0	16/32 位通用定时器 5 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 16/32 位通用定时器模块 5 并提供时钟。 0 16/32 位通用定时器模块 5 禁用。
4	S4	R/W	0	16/32 位通用定时器 4 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 16/32 位通用定时器模块 4 并提供时钟。 0 16/32 位通用定时器模块 4 禁用。
3	S3	R/W	0	16/32 位通用定时器 3 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 16/32 位通用定时器模块 3 并提供时钟。 0 16/32 位通用定时器模块 3 禁用。

位/域	名称	类型	复位	描述
2	S2	R/W	0	16/32 位通用定时器 2 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 16/32 位通用定时器模块 2 并提供时钟。 0 16/32 位通用定时器模块 2 禁用。
1	S1	R/W	0	16/32 位通用定时器 1 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 16/32 位通用定时器模块 1 并提供时钟。 0 16/32 位通用定时器模块 1 禁用。
0	S0	R/W	0	16/32 位通用定时器 0 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 16/32 位通用定时器模块 0 并提供时钟。 0 16/32 位通用定时器模块 0 禁用。

## 寄存器 67: 通用输入/输出睡眠模式时钟门控制寄存器 ( SCGCGPIO ) , 偏移量 0x708

SCGCGPIO 寄存器为软件提供启用和禁用睡眠模式中的 GPIO 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能，并且具有与相应 SCGCn 位相同的极性。

**重要:** 应使用该寄存器控制 GPIO 模块的计时。要支持传统软件，可使用 SCGC2 寄存器。对 SCGC2 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC2 寄存器的写操作更改的任何位都可以通过对 SCGC2 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器写入传统外设（如 GPIO A），则写操作会产生正确操作，但是该位的值不在 SCGC2 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

### 通用输入/输出睡眠模式时钟门控制寄存器 (SCGCGPIO)

基址 0x400F.E000  
偏移量 0x708  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
13	S13	R/W	0	GPIO 端口 P 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 GPIO 端口 P 并提供时钟。 0 GPIO 端口 P 禁用。
12	S12	R/W	0	GPIO 端口 N 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 GPIO 端口 N 并提供时钟。 0 GPIO 端口 N 禁用。
11	S11	R/W	0	GPIO 端口 M 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 GPIO 端口 M 并提供时钟。 0 GPIO 端口 M 禁用。

位/域	名称	类型	复位	描述
10	S10	R/W	0	GPIO 端口 L 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 L 并提供时钟。 0 GPIO 端口 L 禁用。
9	S9	R/W	0	GPIO 端口 K 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 K 并提供时钟。 0 GPIO 端口 K 禁用。
8	S8	R/W	0	GPIO 端口 J 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 J 并提供时钟。 0 GPIO 端口 J 禁用。
7	S7	R/W	0	GPIO 端口 H 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 H 并提供时钟。 0 GPIO 端口 H 禁用。
6	S6	R/W	0	GPIO 端口 G 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 G 并提供时钟。 0 GPIO 端口 G 禁用。
5	S5	R/W	0	GPIO 端口 F 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 F 并提供时钟。 0 GPIO 端口 F 禁用。
4	S4	R/W	0	GPIO 端口 E 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 E 并提供时钟。 0 GPIO 端口 E 禁用。

位/域	名称	类型	复位	描述
3	S3	R/W	0	GPIO 端口 D 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 D 并提供时钟。 0 GPIO 端口 D 禁用。
2	S2	R/W	0	GPIO 端口 C 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 C 并提供时钟。 0 GPIO 端口 C 禁用。
1	S1	R/W	0	GPIO 端口 B 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 B 并提供时钟。 0 GPIO 端口 B 禁用。
0	S0	R/W	0	GPIO 端口 A 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 GPIO 端口 A 并提供时钟。 0 GPIO 端口 A 禁用。

## 寄存器 68: 微型直接存储器访问睡眠模式时钟门控制寄存器 ( SCGC DMA ) , 偏移量 0x70C

SCGC DMA 寄存器为软件提供启用和禁用睡眠模式中的  $\mu$ DMA 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能，并且具有与相应 SCGCn 位相同的极性。

**重要:** 应使用该寄存器控制  $\mu$ DMA 模块的计时。要支持传统软件，可使用 SCGC2 寄存器。对 SCGC2 寄存器中 UDMA 位的写操作还对该寄存器中的 S0 位执行写操作。如果通过对 SCGC2 寄存器的写操作更改 UDMA 位，它可以通过读 SCGC2 寄存器进行正确回读。如果软件使用该寄存器控制  $\mu$ DMA 模块，则该写操作会产生正确操作，但是 SCGC2 寄存器中的 UDMA 位不会反映 S0 位的值。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

### 微型直接存储器访问睡眠模式时钟门控制寄存器 (SCGC DMA)

基址 0x400F.E000  
偏移量 0x70C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	S0	R/W	0	$\mu$ DMA 模块睡眠模式时钟门控制
				<b>值 描述</b>
				1 启用睡眠模式中的 $\mu$ DMA 模块并提供时钟。
				0 $\mu$ DMA 模块禁用。

### 寄存器 69: 休眠睡眠模式时钟门控制寄存器 ( SCGCHIB ) , 偏移量 0x714

SCGCHIB 寄存器为软件提供启用和禁用睡眠模式中休眠模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能，并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制休眠模块的计时。要支持传统软件，可使用 SCGC0 寄存器。对 SCGC0 寄存器中 HIB 位的写操作还对该寄存器中的 S0 位执行写操作。如果通过对 SCGC0 寄存器的写操作更改 HIB 位，它可以通过读 SCGC0 寄存器进行正确回读。如果软件使用该寄存器控制休眠模块，则该写操作会产生正确操作，但是 SCGC0 寄存器中的 HIB 位不会反映 S0 位的值。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

#### 休眠睡眠模式时钟门控制寄存器 (SCGCHIB)

基址 0x400F.E000  
偏移量 0x714  
类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	S0	R/W	1	休眠模块睡眠模式时钟门控制
值 描述				
1 启用睡眠模式中的休眠模块并提供时钟。				
0 休眠模块禁用。				



## 寄存器 70: 通用异步收发器睡眠模式时钟门控制寄存器 ( SCGCUART ) , 偏移量 0x718

SCGCUART 寄存器为软件提供启用和禁用睡眠模式中的 UART 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能, 并且具有与相应 SCGCn 位相同的极性。

**重要:** 应使用该寄存器控制 UART 模块的计时。要支持传统软件, 可使用 SCGC1 寄存器。对 SCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC1 寄存器的写操作更改的任何位都可以通过对 SCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 UART0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 SCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 通用异步收发器睡眠模式时钟门控制寄存器 (SCGCUART)

基址 0x400F.E000  
偏移量 0x718  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								S7	S6	S5	S4	S3	S2	S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	S7	R/W	0	UART 模块 7 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 UART 模块 7 并提供时钟。 0 UART 模块 7 禁用。
6	S6	R/W	0	UART 模块 6 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 UART 模块 6 并提供时钟。 0 UART 模块 6 禁用。
5	S5	R/W	0	UART 模块 5 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 UART 模块 5 并提供时钟。 0 UART 模块 5 禁用。

位/域	名称	类型	复位	描述
4	S4	R/W	0	UART 模块 4 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 UART 模块 4 并提供时钟。 0 UART 模块 4 禁用。
3	S3	R/W	0	UART 模块 3 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 UART 模块 3 并提供时钟。 0 UART 模块 3 禁用。
2	S2	R/W	0	UART 模块 2 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 UART 模块 2 并提供时钟。 0 UART 模块 2 禁用。
1	S1	R/W	0	UART 模块 1 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 UART 模块 1 并提供时钟。 0 UART 模块 1 禁用。
0	S0	R/W	0	UART 模块 0 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 UART 模块 0 并提供时钟。 0 UART 模块 0 禁用。

## 寄存器 71: 同步串行接口睡眠模式时钟门控制寄存器 ( SCGCSSI ), 偏移量 0x71C

SCGCSSI 寄存器为软件提供启用和禁用睡眠模式中的 SSI 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能, 并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 SSI 模块的计时。要支持传统软件, 可使用 SCGC1 寄存器。对 SCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC1 寄存器的写操作更改的任何位都可以通过对 SCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 SSI0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 SCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 同步串行接口睡眠模式时钟门控制寄存器 (SCGCSSI)

基址 0x400F.E000  
偏移量 0x71C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													S3	S2	S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	S3	R/W	0	SSI 模块 3 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 SSI 模块 3 并提供时钟。 0 SSI 模块 3 禁用。
2	S2	R/W	0	SSI 模块 2 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 SSI 模块 2 并提供时钟。 0 SSI 模块 2 禁用。
1	S1	R/W	0	SSI 模块 1 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 SSI 模块 1 并提供时钟。 0 SSI 模块 1 禁用。

---

位/域	名称	类型	复位	描述
0	S0	R/W	0	SSI 模块 0 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 SSI 模块 0 并提供时钟。 0 SSI 模块 0 禁用。

## 寄存器 72: 内部集成电路睡眠模式时钟门控制寄存器 ( SCGCI2C ), 偏移量 0x720

SCGCI2C 寄存器为软件提供启用和禁用睡眠模式中的 I<sup>2</sup>C 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能，并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 I<sup>2</sup>C 模块的计时。要支持传统软件，可使用 SCGC1 寄存器。对 SCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC1 寄存器的写操作更改的任何位都可以通过对 SCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 I<sup>2</sup>C0 ) 进行写操作，则写操作会产生正确操作，但是该位的值不在 SCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

### 内部集成电路睡眠模式时钟门控制寄存器 (SCGCI2C)

基址 0x400F.E000

偏移量 0x720

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											S5	S4	S3	S2	S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
5	S5	R/W	0	I <sup>2</sup> C 模块 5 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 I <sup>2</sup> C 模块 5 并提供时钟。 0 I <sup>2</sup> C 模块 5 禁用。
4	S4	R/W	0	I <sup>2</sup> C 模块 4 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 I <sup>2</sup> C 模块 4 并提供时钟。 0 I <sup>2</sup> C 模块 4 禁用。
3	S3	R/W	0	I <sup>2</sup> C 模块 3 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 I <sup>2</sup> C 模块 3 并提供时钟。 0 I <sup>2</sup> C 模块 3 禁用。

---

位/域	名称	类型	复位	描述
2	S2	R/W	0	I <sup>2</sup> C 模块 2 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 I <sup>2</sup> C 模块 2 并提供时钟。 0 I <sup>2</sup> C 模块 2 禁用。
1	S1	R/W	0	I <sup>2</sup> C 模块 1 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 I <sup>2</sup> C 模块 1 并提供时钟。 0 I <sup>2</sup> C 模块 1 禁用。
0	S0	R/W	0	I <sup>2</sup> C 模块 0 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 I <sup>2</sup> C 模块 0 并提供时钟。 0 I <sup>2</sup> C 模块 0 禁用。

## 寄存器 73: 通用串行总线睡眠模式时钟门控制寄存器 ( SCGCUSB ) , 偏移量 0x728

SCGCUSB 寄存器为软件提供启用和禁用睡眠模式中的 USB 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能, 并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 USB 模块的计时。要支持传统软件, 可使用 SCGC2 寄存器。对 SCGC2 寄存器中 USB0 位的写操作还对该寄存器中的 S0 位执行写操作。如果通过对 SCGC2 寄存器的写操作更改 USB0 位, 它可以通过读 SCGC2 寄存器进行正确回读。如果软件使用该寄存器控制 USB 模块, 则该写操作会产生正确操作, 但是 SCGC2 寄存器中的 USB0 位不会反映 S0 位的值。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 通用串行总线睡眠模式时钟门控制寄存器 (SCGCUSB)

基址 0x400F.E000  
偏移量 0x728  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	S0	R/W	0	USB 模块睡眠模式时钟门控制
	值 描述			
	1			启用睡眠模式中的 USB 模块并提供时钟。
	0			USB 模块禁用。

### 寄存器 74: 控制器局域网睡眠模式时钟门控制寄存器 ( SCGCCAN ) , 偏移量 0x734

SCGCCAN 寄存器为软件提供启用和禁用睡眠模式中的 CAN 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能, 并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 CAN 模块的计时。要支持传统软件, 可使用 SCGC0 寄存器。对 SCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC0 寄存器的写操作更改的任何位都可以通过对 SCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如 CAN0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 SCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 控制器局域网睡眠模式时钟门控制寄存器 (SCGCCAN)

基址 0x400F.E000  
偏移量 0x734  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	S1	R/W	0	CAN 模块 1 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 CAN 模块 1 并提供时钟。 0 CAN 模块 1 禁用。
0	S0	R/W	0	CAN 模块 0 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 CAN 模块 0 并提供时钟。 0 CAN 模块 0 禁用。



**寄存器 75: 模数转换器睡眠模式时钟门控控制寄存器 ( SCGCADC ) , 偏移量 0x738**

SCGCADC 寄存器为软件提供启用和禁用睡眠模式中的 ADC 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控控制寄存器 n SCGCn 寄存器相同的功能, 并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 ADC 模块的计时。要支持传统软件, 可使用 SCGC0 寄存器。对 SCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC0 寄存器的写操作更改的任何位都可以通过对 SCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如 ADC0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 SCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

**模数转换器睡眠模式时钟门控控制寄存器 (SCGCADC)**

基址 0x400F.E000

偏移量 0x738

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	S1	R/W	0	ADC 模块 1 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 ADC 模块 1 并提供时钟。 0 ADC 模块 1 禁用。
0	S0	R/W	0	ADC 模块 0 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 ADC 模块 0 并提供时钟。 0 ADC 模块 0 禁用。

### 寄存器 76: 模拟比较器睡眠模式时钟门控制寄存器 ( SCGCACMP ), 偏移量 0x73C

SCGCACMP 寄存器为软件提供启用和禁用睡眠模式中模拟比较器模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能，并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制模拟比较器模块的计时。要支持传统软件，可使用 SCGC1 寄存器。对 SCGC1 寄存器中 COMPn 位的写操作还对该寄存器中的 S0 位执行写操作。如果通过对 SCGC1 寄存器的写操作置位任何 COMPn 位，它可以在对 SCGC1 寄存器进行读操作时进行正确回读。如果软件使用该寄存器更改模拟比较器模块的计时，则写操作会产生正确操作，但是 SCGC1 寄存器的 COMPn 位不反映值 S0。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

#### 模拟比较器睡眠模式时钟门控制寄存器 (SCGCACMP)

基址 0x400F.E000  
偏移量 0x73C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	S0	R/W	0	模拟比较器模块 0 睡眠模式时钟门控制
				值 描述
				1 启用睡眠模式中的模拟比较器模块并提供时钟。
				0 模拟比较器模块禁用。

**寄存器 77: 脉宽调制器睡眠模式时钟门控制寄存器 ( SCGCPWM ) ， 偏移量 0x740**

SCGCPWM 寄存器为软件提供启用和禁用睡眠模式中的 PWM 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能，并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 PWM 模块的计时。要支持传统软件，可使用 SCGC0 寄存器。对 SCGC0 寄存器中 PWM 位的写操作还对该寄存器中的 S0 位执行写操作。如果通过对 SCGC0 寄存器的写操作更改 PWM 位，它可以通过读 SCGC0 寄存器进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器写 S0，则写操作会产生正确操作，但是该位的值不在 SCGC0 寄存器的 PWM 位中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

**脉宽调制器睡眠模式时钟门控制寄存器 (SCGCPWM)**

基址 0x400F.E000  
偏移量 0x740  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
1	S1	R/W	0	PWM 模块 1 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 PWM 模块 1 并提供时钟。 0 PWM 模块 1 禁用。
0	S0	R/W	0	PWM 模块 0 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 PWM 模块 0 并提供时钟。 0 PWM 模块 0 禁用。

### 寄存器 78: 正交编码器接口睡眠模式时钟门控制寄存器 ( SCGCQEI ) , 偏移量 0x744

SCGCQEI 寄存器为软件提供启用和禁用睡眠模式中的 QEI 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能, 并且具有与相应 SCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 QEI 模块的计时。要支持传统软件, 可使用 SCGC1 寄存器。对 SCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 SCGC1 寄存器的写操作更改的任何位都可以通过对 SCGC1 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如 QEI0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 SCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 正交编码器接口睡眠模式时钟门控制寄存器 (SCGCQEI)

基址 0x400F.E000  
偏移量 0x744  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	S1	R/W	0	QEI 模块 1 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 QEI 模块 1 并提供时钟。 0 QEI 模块 1 禁用。
0	S0	R/W	0	QEI 模块 0 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 QEI 模块 0 并提供时钟。 0 QEI 模块 0 禁用。

## 寄存器 79: 32/64 位宽通用定时器睡眠模式时钟门控制寄存器 ( SCGCWTIMER ) , 偏移量 0x75C

SCGCWTIMER 寄存器为软件提供启用和禁用睡眠模式中的 32/64 位定时器模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为定时器模块提供与传统睡眠模式时钟门控制寄存器 n SCGCn 寄存器相同的功能，并且具有与相应 SCGCn 位相同的位极性。

### 32/64 位宽通用定时器睡眠模式时钟门控制寄存器 (SCGCWTIMER)

基址 0x400F.E000

偏移量 0x75C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											S5	S4	S3	S2	S1	S0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
5	S5	R/W	0	32/64 位宽通用定时器 5 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 32/64 位宽通用定时器模块 5 并提供时钟。 0 32/64 位宽通用定时器模块 5 禁用。
4	S4	R/W	0	32/64 位宽通用定时器 4 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 32/64 位宽通用定时器模块 4 并提供时钟。 0 32/64 位宽通用定时器模块 4 禁用。
3	S3	R/W	0	32/64 位宽通用定时器 3 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 32/64 位宽通用定时器模块 3 并提供时钟。 0 32/64 位宽通用定时器模块 3 禁用。
2	S2	R/W	0	32/64 位宽通用定时器 2 睡眠模式时钟门控制  值 描述 1 启用睡眠模式中的 32/64 位宽通用定时器模块 2 并提供时钟。 0 32/64 位宽通用定时器模块 2 禁用。

---

位/域	名称	类型	复位	描述
1	S1	R/W	0	32/64 位宽通用定时器 1 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 32/64 位宽通用定时器模块 1 并提供时钟。 0 32/64 位宽通用定时器模块 1 禁用。
0	S0	R/W	0	32/64 位宽通用定时器 0 睡眠模式时钟门控控制  值 描述 1 启用睡眠模式中的 32/64 位宽通用定时器模块 0 并提供时钟。 0 32/64 位宽通用定时器模块 0 禁用。

## 寄存器 80: 看门狗定时器深度睡眠模式时钟门控制寄存器 (DCGCWD), 偏移量 0x800

DCGCWD 寄存器为软件提供启用和禁用深度睡眠模式中看门狗模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制看门狗模块的计时。要支持传统软件, 可使用 DCGC0 寄存器。对 DCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC0 寄存器的写操作更改的任何位都可以通过对 DCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 (如看门狗 0) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 DCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 看门狗定时器深度睡眠模式时钟门控制寄存器 (DCGCWD)

基址 0x400F.E000

偏移量 0x800

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	D1	R/W	0	看门狗定时器 1 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的看门狗模块 1 并提供时钟。 0 看门狗模块 1 禁用。
0	D0	R/W	0	看门狗定时器 0 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的看门狗模块 0 并提供时钟。 0 看门狗模块 0 禁用。

### 寄存器 81: 16/32 位通用定时器深度睡眠模式时钟门控制控制寄存器 ( DCGCTIMER ) , 偏移量 0x804

DCGCTIMER 寄存器为软件提供启用和禁用深度睡眠模式中的 16/32 位定时器模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为定时器模块提供与传统深度睡眠模式时钟门控制控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制定时器模块的计时。要支持传统软件, 可使用 DCGC1 寄存器。对 DCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC1 寄存器的写操作更改的任何位都可以通过对 DCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 Timer 0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 DCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 16/32 位通用定时器深度睡眠模式时钟门控制控制寄存器 (DCGCTIMER)

基址 0x400F.E000  
偏移量 0x804  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											D5	D4	D3	D2	D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	D5	R/W	0	16/32 位通用定时器 5 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 16/32 位通用定时器模块 5 并提供时钟。 0 16/32 位通用定时器模块 5 禁用。
4	D4	R/W	0	16/32 位通用定时器 4 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 16/32 位通用定时器模块 4 并提供时钟。 0 16/32 位通用定时器模块 4 禁用。
3	D3	R/W	0	16/32 位通用定时器 3 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 16/32 位通用定时器模块 3 并提供时钟。 0 16/32 位通用定时器模块 3 禁用。



位/域	名称	类型	复位	描述
2	D2	R/W	0	16/32 位通用定时器 2 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 16/32 位通用定时器模块 2 并提供时钟。 0 16/32 位通用定时器模块 2 禁用。
1	D1	R/W	0	16/32 位通用定时器 1 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 16/32 位通用定时器模块 1 并提供时钟。 0 16/32 位通用定时器模块 1 禁用。
0	D0	R/W	0	16/32 位通用定时器 0 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 16/32 位通用定时器模块 0 并提供时钟。 0 16/32 位通用定时器模块 0 禁用。

## 寄存器 82: 通用输入/输出深度睡眠模式时钟门控制寄存器 ( DCGCGPIO ) , 偏移量 0x808

DCGCGPIO 寄存器为软件提供启用和禁用深度睡眠模式中的 GPIO 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 GPIO 模块的计时。要支持传统软件, 可使用 DCGC2 寄存器。对 DCGC2 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC2 寄存器的写操作更改的任何位都可以通过对 DCGC2 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器写入传统外设 ( 如 GPIO A ) , 则写操作会产生正确操作, 但是该位的值不在 DCGC2 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 通用输入/输出深度睡眠模式时钟门控制寄存器 (DCGCGPIO)

基址 0x400F.E000  
偏移量 0x808  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
13	D13	R/W	0	GPIO 端口 P 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 P 并提供时钟。 0 GPIO 端口 P 禁用。
12	D12	R/W	0	GPIO 端口 N 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 N 并提供时钟。 0 GPIO 端口 N 禁用。
11	D11	R/W	0	GPIO 端口 M 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 M 并提供时钟。 0 GPIO 端口 M 禁用。

位/域	名称	类型	复位	描述
10	D10	R/W	0	GPIO 端口 L 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 L 并提供时钟。 0 GPIO 端口 L 禁用。
9	D9	R/W	0	GPIO 端口 K 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 K 并提供时钟。 0 GPIO 端口 K 禁用。
8	D8	R/W	0	GPIO 端口 J 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 J 并提供时钟。 0 GPIO 端口 J 禁用。
7	D7	R/W	0	GPIO 端口 H 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 H 并提供时钟。 0 GPIO 端口 H 禁用。
6	D6	R/W	0	GPIO 端口 G 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 G 并提供时钟。 0 GPIO 端口 G 禁用。
5	D5	R/W	0	GPIO 端口 F 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 F 并提供时钟。 0 GPIO 端口 F 禁用。
4	D4	R/W	0	GPIO 端口 E 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 E 并提供时钟。 0 GPIO 端口 E 禁用。

---

位/域	名称	类型	复位	描述
3	D3	R/W	0	GPIO 端口 D 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 D 并提供时钟。 0 GPIO 端口 D 禁用。
2	D2	R/W	0	GPIO 端口 C 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 C 并提供时钟。 0 GPIO 端口 C 禁用。
1	D1	R/W	0	GPIO 端口 B 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 B 并提供时钟。 0 GPIO 端口 B 禁用。
0	D0	R/W	0	GPIO 端口 A 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 GPIO 端口 A 并提供时钟。 0 GPIO 端口 A 禁用。

### 寄存器 83: 微型直接存储器访问深度睡眠模式时钟门控制寄存器 ( DCGCDMA ) , 偏移量 0x80C

DCGCDMA 寄存器为软件提供启用和禁用深度睡眠模式中的  $\mu$ DMA 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能，并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制  $\mu$ DMA 模块的计时。要支持传统软件，可使用 DCGC2 寄存器。对 DCGC2 寄存器中 UDMA 位的写操作还对该寄存器中的 D0 位执行写操作。如果通过对 DCGC2 寄存器的写操作更改 UDMA 位，它可以通过读 DCGC2 寄存器进行正确回读。如果软件使用该寄存器控制  $\mu$ DMA 模块，则该写操作会产生正确操作，但是 DCGC2 寄存器中的 UDMA 位不会反映 D0 位的值。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

#### 微型直接存储器访问深度睡眠模式时钟门控制寄存器 (DCGCDMA)

基址 0x400F.E000  
偏移量 0x80C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	D0	R/W	0	$\mu$ DMA 模块深度睡眠模式时钟门控制
				<b>值 描述</b>
				1 启用深度睡眠模式中的 $\mu$ DMA 模块并提供时钟。
				0 $\mu$ DMA 模块禁用。

**寄存器 84: 休眠深度睡眠模式时钟门控控制寄存器 ( DCGCHIB ) , 偏移量 0x814**

DCGCHIB 寄存器为软件提供启用和禁用深度睡眠模式中休眠模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制休眠模块的计时。要支持传统软件, 可使用 DCGC0 寄存器。对 DCGC0 寄存器中 HIB 位的写操作还对该寄存器中的 D0 位执行写操作。如果通过对 DCGC0 寄存器的写操作更改 HIB 位, 它可以通过读 DCGC0 寄存器进行正确回读。如果软件使用该寄存器控制休眠模块, 则该写操作会产生正确操作, 但是 DCGC0 寄存器中的 HIB 位不会反映 D0 位的值。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

休眠深度睡眠模式时钟门控控制寄存器 (DCGCHIB)

基址 0x400F.E000  
偏移量 0x814  
类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	D0	R/W	1	休眠模块深度睡眠模式时钟门控控制
值 描述				
1 启用深度睡眠模式中的休眠模块并提供时钟。				
0 休眠模块禁用。				

## 寄存器 85: 通用异步收发器深度睡眠模式时钟门控制寄存器 ( DCGCUART ) , 偏移量 0x818

DCGCUART 寄存器为软件提供启用和禁用深度睡眠模式中的 UART 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能，并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 UART 模块的计时。要支持传统软件，可使用 DCGC1 寄存器。对 DCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC1 寄存器的写操作更改的任何位都可以通过对 DCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设（如 UART0）进行写操作，则写操作会产生正确操作，但是该位的值不在 DCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

### 通用异步收发器深度睡眠模式时钟门控制寄存器 (DCGCUART)

基址 0x400F.E000  
偏移量 0x818  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								D7	D6	D5	D4	D3	D2	D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	D7	R/W	0	UART 模块 7 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 UART 模块 7 并提供时钟。 0 UART 模块 7 禁用。
6	D6	R/W	0	UART 模块 6 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 UART 模块 6 并提供时钟。 0 UART 模块 6 禁用。
5	D5	R/W	0	UART 模块 5 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 UART 模块 5 并提供时钟。 0 UART 模块 5 禁用。

位/域	名称	类型	复位	描述
4	D4	R/W	0	UART 模块 4 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 UART 模块 4 并提供时钟。 0 UART 模块 4 禁用。
3	D3	R/W	0	UART 模块 3 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 UART 模块 3 并提供时钟。 0 UART 模块 3 禁用。
2	D2	R/W	0	UART 模块 2 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 UART 模块 2 并提供时钟。 0 UART 模块 2 禁用。
1	D1	R/W	0	UART 模块 1 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 UART 模块 1 并提供时钟。 0 UART 模块 1 禁用。
0	D0	R/W	0	UART 模块 0 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 UART 模块 0 并提供时钟。 0 UART 模块 0 禁用。



## 寄存器 86: 同步串行接口深度睡眠模式时钟门控制寄存器 ( DCGCSSI ) , 偏移量 0x81C

DCGCSSI 寄存器为软件提供启用和禁用深度睡眠模式中的 SSI 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 SSI 模块的计时。要支持传统软件, 可使用 DCGC1 寄存器。对 DCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC1 寄存器的写操作更改的任何位都可以通过对 DCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 ( 如 SSI0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 DCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 同步串行接口深度睡眠模式时钟门控制寄存器 (DCGCSSI)

基址 0x400F.E000  
偏移量 0x81C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													D3	D2	D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	D3	R/W	0	SSI 模块 3 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 SSI 模块 3 并提供时钟。 0 SSI 模块 3 禁用。
2	D2	R/W	0	SSI 模块 2 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 SSI 模块 2 并提供时钟。 0 SSI 模块 2 禁用。
1	D1	R/W	0	SSI 模块 1 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 SSI 模块 1 并提供时钟。 0 SSI 模块 1 禁用。

---

位/域	名称	类型	复位	描述
0	D0	R/W	0	SSI 模块 0 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 SSI 模块 0 并提供时钟。 0 SSI 模块 0 禁用。

## 寄存器 87: 内部集成电路深度睡眠模式时钟门控制控制寄存器 (DCGC12C), 偏移量 0x820

DCGC12C 寄存器为软件提供启用和禁用深度睡眠模式中的 I<sup>2</sup>C 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 I<sup>2</sup>C 模块的计时。要支持传统软件, 可使用 DCGC1 寄存器。对 DCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC1 寄存器的写操作更改的任何位都可以通过对 DCGC1 寄存器的读操作进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器对传统外设 (如 I<sup>2</sup>C0) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 DCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 内部集成电路深度睡眠模式时钟门控制控制寄存器 (DCGC12C)

基址 0x400F.E000  
偏移量 0x820  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											D5	D4	D3	D2	D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	D5	R/W	0	I <sup>2</sup> C 模块 5 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 I <sup>2</sup> C 模块 5 并提供时钟。 0 I <sup>2</sup> C 模块 5 禁用。
4	D4	R/W	0	I <sup>2</sup> C 模块 4 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 I <sup>2</sup> C 模块 4 并提供时钟。 0 I <sup>2</sup> C 模块 4 禁用。
3	D3	R/W	0	I <sup>2</sup> C 模块 3 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 I <sup>2</sup> C 模块 3 并提供时钟。 0 I <sup>2</sup> C 模块 3 禁用。

---

位/域	名称	类型	复位	描述
2	D2	R/W	0	I <sup>2</sup> C 模块 2 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 I <sup>2</sup> C 模块 2 并提供时钟。 0 I <sup>2</sup> C 模块 2 禁用。
1	D1	R/W	0	I <sup>2</sup> C 模块 1 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 I <sup>2</sup> C 模块 1 并提供时钟。 0 I <sup>2</sup> C 模块 1 禁用。
0	D0	R/W	0	I <sup>2</sup> C 模块 0 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 I <sup>2</sup> C 模块 0 并提供时钟。 0 I <sup>2</sup> C 模块 0 禁用。

## 寄存器 88: 通用串行总线深度睡眠模式时钟门控制寄存器 ( DCGCUSB ) , 偏移量 0x828

DCGCUSB 寄存器为软件提供启用和禁用深度睡眠模式中的 USB 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能，并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 USB 模块的计时。要支持传统软件，可使用 DCGC2 寄存器。对 DCGC2 寄存器中 USB0 位的写操作还对该寄存器中的 D0 位执行写操作。如果通过对 DCGC2 寄存器的写操作更改 USB0 位，它可以通过读 DCGC2 寄存器进行正确回读。如果软件使用该寄存器控制 USB 模块，则该写操作会产生正确操作，但是 DCGC2 寄存器中的 USB0 位不会反映 D0 位的值。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

### 通用串行总线深度睡眠模式时钟门控制寄存器 (DCGCUSB)

基址 0x400F.E000  
偏移量 0x828  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	D0	R/W	0	USB 模块深度睡眠模式时钟门控制
				值 描述
				1 启用 USB 模式中的休眠模块并提供时钟。
				0 USB 模块禁用。

**寄存器 89: 控制器局域网深度睡眠模式时钟门控制寄存器 ( DCGCCAN ) , 偏移量 0x834**

DCGCCAN 寄存器为软件提供启用和禁用深度睡眠模式中的 CAN 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能，并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 CAN 模块的计时。要支持传统软件，可使用 DCGC0 寄存器。对 DCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC0 寄存器的写操作更改的任何位都可以通过对 DCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如 CAN0 ) 进行写操作，则写操作会产生正确操作，但是该位的值不在 DCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

控制器局域网深度睡眠模式时钟门控制寄存器 (DCGCCAN)

基址 0x400F.E000  
偏移量 0x834  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
1	D1	R/W	0	CAN 模块 1 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 CAN 模块 1 并提供时钟。 0 CAN 模块 1 禁用。
0	D0	R/W	0	CAN 模块 0 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 CAN 模块 0 并提供时钟。 0 CAN 模块 0 禁用。

## 寄存器 90: 模数转换器深度睡眠模式时钟门控制寄存器 (DCGCADC), 偏移量 0x838

DCGCADC 寄存器为软件提供启用和禁用深度睡眠模式中的 ADC 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 ADC 模块的计时。要支持传统软件, 可使用 DCGC0 寄存器。对 DCGC0 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC0 寄存器的写操作更改的任何位都可以通过对 DCGC0 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 (如 ADC0) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 DCGC0 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 模数转换器深度睡眠模式时钟门控制寄存器 (DCGCADC)

基址 0x400F.E000  
偏移量 0x838  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	D1	R/W	0	ADC 模块 1 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 ADC 模块 1 并提供时钟。 0 ADC 模块 1 禁用。
0	D0	R/W	0	ADC 模块 0 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 ADC 模块 0 并提供时钟。 0 ADC 模块 0 禁用。

### 寄存器 91: 模拟比较器深度睡眠模式时钟门控制寄存器 ( DCGCACMP ) , 偏移量 0x83C

DCGCACMP 寄存器为软件提供启用和禁用深度睡眠模式中模拟比较器模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 nDCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制模拟比较器模块的计时。要支持传统软件, 可使用 DCGC1 寄存器。对 DCGC1 寄存器中 COMPn 位的写操作还对该寄存器中的 D0 位执行写操作。如果通过对 DCGC1 寄存器的写操作置位任何 COMPn 位, 它可以在对 DCGC1 寄存器进行读操作时进行正确回读。如果软件使用该寄存器更改模拟比较器模块的计时, 则写操作会产生正确操作, 但是 DCGC1 寄存器的 COMPn 位不反映值 D0。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 模拟比较器深度睡眠模式时钟门控制寄存器 (DCGCACMP)

基址 0x400F.E000  
偏移量 0x83C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
																D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	D0	R/W	0	模拟比较器模块 0 深度睡眠模式时钟门控制

**值 描述**

- 1 启用深度睡眠模式中的模拟比较器模块并提供时钟。
- 0 模拟比较器模块禁用。



## 寄存器 92: 脉宽调制器深度睡眠模式时钟门控制寄存器 ( DCGCPWM ) , 偏移量 0x840

DCGCPWM 寄存器为软件提供启用和禁用深度睡眠模式中的 PWM 模块的功能。在启用时，为模块提供一个时钟。在禁用时，禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能，并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 PWM 模块的计时。要支持传统软件，可使用 DCGC0 寄存器。对 DCGC0 寄存器中 PWM 位的写操作还对该寄存器中的 D0 位执行写操作。如果通过对 DCGC0 寄存器的写操作更改 PWM 位，它可以通过读 DCGC0 寄存器进行正确回读。软件必须使用该寄存器支持不处于传统寄存器中的模块。如果软件使用该寄存器写 D0，则写操作会产生正确操作，但是该位的值不在 DCGC0 寄存器的 PWM 位中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

### 脉宽调制器深度睡眠模式时钟门控制寄存器 (DCGCPWM)

基址 0x400F.E000  
偏移量 0x840  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
1	D1	R/W	0	PWM 模块 1 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 PWM 模块 1 并提供时钟。 0 PWM 模块 1 禁用。
0	D0	R/W	0	PWM 模块 0 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 PWM 模块 0 并提供时钟。 0 PWM 模块 0 禁用。

### 寄存器 93: 正交编码器接口深度睡眠模式时钟门控制寄存器 ( DCGCQEI ) , 偏移量 0x844

DCGCQEI 寄存器为软件提供启用和禁用深度睡眠模式中的 QEI 模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为看门狗模块提供与传统深度睡眠模式时钟门控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

**重要:** 应使用该寄存器控制 QEI 模块的计时。要支持传统软件, 可使用 DCGC1 寄存器。对 DCGC1 寄存器的写操作还对该寄存器中的相应位执行写操作。通过对 DCGC1 寄存器的写操作更改的任何位都可以通过对 DCGC1 寄存器的读操作进行正确回读。如果软件使用该寄存器对传统外设 ( 如 QEI0 ) 进行写操作, 则写操作会产生正确操作, 但是该位的值不在 DCGC1 寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 正交编码器接口深度睡眠模式时钟门控制寄存器 (DCGCQEI)

基址 0x400F.E000  
偏移量 0x844  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	D1	R/W	0	QEI 模块 1 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 QEI 模块 1 并提供时钟。 0 QEI 模块 1 禁用。
0	D0	R/W	0	QEI 模块 0 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 QEI 模块 0 并提供时钟。 0 QEI 模块 0 禁用。

## 寄存器 94: 32/64 位宽通用定时器深度睡眠模式时钟门控制控制寄存器 (DCGCWTIMER), 偏移量 0x85C

DCGCWTIMER 寄存器为软件提供启用和禁用深度睡眠模式中的 32/64 位宽定时器模块的功能。在启用时, 为模块提供一个时钟。在禁用时, 禁用时钟以节能。该寄存器特别为定时器模块提供与传统深度睡眠模式时钟门控制控制寄存器 n DCGCn 寄存器相同的功能, 并且具有与相应 DCGCn 位相同的位极性。

### 32/64 位宽通用定时器深度睡眠模式时钟门控制控制寄存器 (DCGCWTIMER)

基址 0x400F.E000

偏移量 0x85C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											D5	D4	D3	D2	D1	D0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	D5	R/W	0	32/64 位宽通用定时器 5 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 32/64 位宽通用定时器模块 5 并提供时钟。 0 32/64 位宽通用定时器模块 5 禁用。
4	D4	R/W	0	32/64 位宽通用定时器 4 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 32/64 位宽通用定时器模块 4 并提供时钟。 0 32/64 位宽通用定时器模块 4 禁用。
3	D3	R/W	0	32/64 位宽通用定时器 3 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 32/64 位宽通用定时器模块 3 并提供时钟。 0 32/64 位宽通用定时器模块 3 禁用。
2	D2	R/W	0	32/64 位宽通用定时器 2 深度睡眠模式时钟门控制  值 描述 1 启用深度睡眠模式中的 32/64 位宽通用定时器模块 2 并提供时钟。 0 32/64 位宽通用定时器模块 2 禁用。

---

位/域	名称	类型	复位	描述
1	D1	R/W	0	32/64 位宽通用定时器 1 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 32/64 位宽通用定时器模块 1 并提供时钟。 0 32/64 位宽通用定时器模块 1 禁用。
0	D0	R/W	0	32/64 位宽通用定时器 0 深度睡眠模式时钟门控控制  值 描述 1 启用深度睡眠模式中的 32/64 位宽通用定时器模块 0 并提供时钟。 0 32/64 位宽通用定时器模块 0 禁用。

## 寄存器 95: 看门狗定时器功率控制寄存器 (PCWD), 偏移量 0x900

PCWD 寄存器控制提供到看门狗模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生变更, 并且 RCGCWD、SCGCWD 或 DCGCWD 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRWD 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** 看门狗模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

### 看门狗定时器功率控制寄存器 (PCWD)

基址 0x400F.E000

偏移量 0x900

类型 R/W, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	R/W	1	看门狗定时器 1 功率控制 除非相应主动模式 RCGCWD、SCGCWD 或 DCGCWD 位清零, 否则该位的值不具有意义。  值 描述 1 看门狗模块 1 通电, 但并未接收时钟。在这种情况下, 该模块无效。 该配置提供了第二低的模块功耗, 因为它仅消耗漏流。 0 看门狗模块 1 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。 该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。
0	P0	R/W	1	看门狗定时器 0 功率控制 除非相应主动模式 RCGCWD、SCGCWD 或 DCGCWD 位清零, 否则该位的值不具有意义。  值 描述 1 看门狗模块 0 通电, 但并未接收时钟。在这种情况下, 该模块无效。 该配置提供了第二低的模块功耗, 因为它仅消耗漏流。 0 看门狗模块 0 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。 该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。

### 寄存器 96: 16/32 位通用定时器功率控制寄存器 ( PCTIMER ) , 偏移量 0x904

PCTIMER 寄存器控制提供给 16/32 位定时器模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生改变, 并且 RCGCTIMER、SCGCTIMER 或 DCGCTIMER 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRTIMER 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** 定时器模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

#### 16/32 位通用定时器功率控制寄存器 (PCTIMER)

基址 0x400F.E000  
偏移量 0x904  
类型 R/W, 复位 0x0000.003F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	P5	R/W	1	<p>16/32 位通用定时器 5 功率控制</p> <p>除非相应主动模式 RCGCTIMER、SCGCTIMER 或 DCGCTIMER 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 定时器模块 5 通电, 但并未接收时钟。在这种情况下, 该模块无效。</p> <p>该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 定时器模块 5 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>
4	P4	R/W	1	<p>16/32 位通用定时器 4 功率控制</p> <p>除非相应主动模式 RCGCTIMER、SCGCTIMER 或 DCGCTIMER 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 定时器模块 4 通电, 但并未接收时钟。在这种情况下, 该模块无效。</p> <p>该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 定时器模块 4 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>

位/域	名称	类型	复位	描述
3	P3	R/W	1	<p>16/32 位通用定时器 3 功率控制</p> <p>除非相应主动模式 RCGCTIMER、SCGCTIMER 或 DCGCTIMER 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 定时器模块 3 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 定时器模块 3 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
2	P2	R/W	1	<p>16/32 位通用定时器 2 功率控制</p> <p>除非相应主动模式 RCGCTIMER、SCGCTIMER 或 DCGCTIMER 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 定时器模块 2 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 定时器模块 2 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
1	P1	R/W	1	<p>16/32 位通用定时器 1 功率控制</p> <p>除非相应主动模式 RCGCTIMER、SCGCTIMER 或 DCGCTIMER 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 定时器模块 1 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 定时器模块 1 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

---

位/域	名称	类型	复位	描述
0	P0	R/W	1	<p>16/32 位通用定时器 0 功率控制</p> <p>除非相应主动模式 RCGCTIMER、SCGCTIMER 或 DCGCTIMER 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 定时器模块 0 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 定时器模块 0 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>



## 寄存器 97: 通用输入/输出功率控制寄存器 (PCGPIO) , 偏移量 0x908

PCGPIO 寄存器控制提供到 GPIO 模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生更改, 并且 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRGPIO 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** GPIO 模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

### 通用输入/输出功率控制寄存器 (PCGPIO)

基址 0x400FE000

偏移量 0x908

类型 R/W, 复位 0x0000.7FFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
13	P13	R/W	1	GPIO 端口 P 功率控制 除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零, 否则该位的值不具有意义。  值 描述 1 GPIO 端口 P 通电, 但并未接收时钟。在这种情况下, 该模块无效。 该配置提供了第二低的模块功耗, 因为它仅消耗漏流。 0 GPIO 端口 P 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。 该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。
12	P12	R/W	1	GPIO 端口 N 功率控制 除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零, 否则该位的值不具有意义。  值 描述 1 GPIO 端口 N 通电, 但并未接收时钟。在这种情况下, 该模块无效。 该配置提供了第二低的模块功耗, 因为它仅消耗漏流。 0 GPIO 端口 N 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。 该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。

位/域	名称	类型	复位	描述
11	P11	R/W	1	<p>GPIO 端口 M 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 M 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 M 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
10	P10	R/W	1	<p>GPIO 端口 L 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 L 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 L 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
9	P9	R/W	1	<p>GPIO 端口 K 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 K 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 K 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
8	P8	R/W	1	<p>GPIO 端口 J 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 J 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 J 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

位/域	名称	类型	复位	描述
7	P7	R/W	1	<p>GPIO 端口 H 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 H 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 H 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
6	P6	R/W	1	<p>GPIO 端口 G 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 G 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 G 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
5	P5	R/W	1	<p>GPIO 端口 F 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 F 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 F 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
4	P4	R/W	1	<p>GPIO 端口 E 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 E 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 E 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

位/域	名称	类型	复位	描述
3	P3	R/W	1	<p>GPIO 端口 D 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 D 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 D 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
2	P2	R/W	1	<p>GPIO 端口 C 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 C 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 C 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
1	P1	R/W	1	<p>GPIO 端口 B 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 B 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 B 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
0	P0	R/W	1	<p>GPIO 端口 A 功率控制</p> <p>除非相应主动模式 RCGCGPIO、SCGCGPIO 或 DCGCGPIO 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 GPIO 端口 A 通电，但并未接收时钟。在这种情况下，该模块无效。 该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 GPIO 端口 A 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。 该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

## 寄存器 98: 微型直接存储器访问功率控制寄存器 (PCDMA), 偏移量 0x90C

PCDMA 寄存器控制提供到  $\mu$ DMA 模块的功率。清零模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 模块的状态未保留。如果主动模式发生更改, 并且 RCGCDMA、SCGCDMA 或 DCGCDMA 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRDMA 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:**  $\mu$ DMA 模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

### 微型直接存储器访问功率控制寄存器 (PCDMA)

基址 0x400F.E000

偏移量 0x90C

类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	R/W	1	$\mu$ DMA 模块功率控制 除非相应主动模式 RCGCDMA、SCGCDMA 或 DCGCDMA 位清零, 否则该位的值不具有意义。

#### 值 描述

- 1  $\mu$ DMA 模块通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。
- 0  $\mu$ DMA 模块未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。  
该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。

### 寄存器 99: 休眠功率控制寄存器 ( PCHIB ) , 偏移量 0x914

PCHIB 寄存器控制提供到休眠模块的功率。清零模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时，模块的状态未保留。如果主动模式发生更改，并且 RCGCHIB、SCGCHIB 或 DCGCHIB 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1，则软件应使用 SRHIB 寄存器执行外设复位。由于状态丢失而重新启用时，软件必须重新初始化模块。

**注意：** 休眠模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

#### 休眠功率控制寄存器 (PCHIB)

基址 0x400F.E000  
偏移量 0x914  
类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
	P0															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	R/W	1	<p>休眠模块功率控制</p> <p>除非相应主动模式 RCGCHIB、SCGCHIB 或 DCGCHIB 位清零，否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 休眠模块通电，但并未接收时钟。在这种情况下，该模块无效。该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 休眠模块未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

## 寄存器 100: 通用异步收发器功率控制寄存器 (PCUART), 偏移量 0x918

PCUART 寄存器控制提供到 UART 模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生更改, 并且 RCGCUART、SCGCUART 或 DCGCUART 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRUART 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** UART 模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

### 通用异步收发器功率控制寄存器 (PCUART)

基址 0x400FE000

偏移量 0x918

类型 R/W, 复位 0x0000.00FF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								P7	P6	P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	P7	R/W	1	UART 模块 7 功率控制 除非相应主动模式 RCGCUART、SCGCUART 或 DCGCUART 位清零, 否则该位的值不具有意义。  <b>值 描述</b> 1 UART 模块 7 通电, 但并未接收时钟。在这种情况下, 该模块无效。 该配置提供了第二低的模块功耗, 因为它仅消耗漏流。 0 UART 模块 7 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。 该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。
6	P6	R/W	1	UART 模块 6 功率控制 除非相应主动模式 RCGCUART、SCGCUART 或 DCGCUART 位清零, 否则该位的值不具有意义。  <b>值 描述</b> 1 UART 模块 6 通电, 但并未接收时钟。在这种情况下, 该模块无效。 该配置提供了第二低的模块功耗, 因为它仅消耗漏流。 0 UART 模块 6 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。 该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。

位/域	名称	类型	复位	描述
5	P5	R/W	1	<p>UART 模块 5 功率控制</p> <p>除非相应主动模式 RCGCUART、SCGCUART 或 DCGCUART 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 UART 模块 5 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 UART 模块 5 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
4	P4	R/W	1	<p>UART 模块 4 功率控制</p> <p>除非相应主动模式 RCGCUART、SCGCUART 或 DCGCUART 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 UART 模块 4 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 UART 模块 4 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
3	P3	R/W	1	<p>UART 模块 3 功率控制</p> <p>除非相应主动模式 RCGCUART、SCGCUART 或 DCGCUART 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 UART 模块 3 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 UART 模块 3 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>



位/域	名称	类型	复位	描述
2	P2	R/W	1	<p>UART 模块 2 功率控制</p> <p>除非相应主动模式 RCGCUART、SCGCUART 或 DCGCUART 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 UART 模块 2 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 UART 模块 2 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
1	P1	R/W	1	<p>UART 模块 1 功率控制</p> <p>除非相应主动模式 RCGCUART、SCGCUART 或 DCGCUART 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 UART 模块 1 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 UART 模块 1 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
0	P0	R/W	1	<p>UART 模块 70 功率控制</p> <p>除非相应主动模式 RCGCUART、SCGCUART 或 DCGCUART 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 UART 模块 0 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 UART 模块 0 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

### 寄存器 101: 同步串行接口功率控制寄存器 (PCSSI), 偏移量 0x91C

PCSSI 寄存器控制提供到 SSI 模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生更改, 并且 RCGCSSI、SCGCSSI 或 DCGCSSI 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRSSI 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** SSI 模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

#### 同步串行接口功率控制寄存器 (PCSSI)

基址 0x400F.E000  
偏移量 0x91C  
类型 R/W, 复位 0x0000.000F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	P3	R/W	1	<p>SSI 模块 3 功率控制</p> <p>除非相应主动模式 RCGCSSI、SCGCSSI 或 DCGCSSI 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 SSI 模块 3 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 SSI 模块 3 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>
2	P2	R/W	1	<p>SSI 模块 2 功率控制</p> <p>除非相应主动模式 RCGCSSI、SCGCSSI 或 DCGCSSI 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 SSI 模块 2 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 SSI 模块 2 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>

位/域	名称	类型	复位	描述
1	P1	R/W	1	<p>SSI 模块 1 功率控制</p> <p>除非相应主动模式 RCGCSSI、SCGCSSI 或 DCGCSSI 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 SSI 模块 1 通电，但并未接收时钟。在这种情况下，该模块无效。该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 SSI 模块 1 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
0	P0	R/W	1	<p>SSI 模块 0 功率控制</p> <p>除非相应主动模式 RCGCSSI、SCGCSSI 或 DCGCSSI 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 SSI 模块 0 通电，但并未接收时钟。在这种情况下，该模块无效。该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 SSI 模块 0 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

## 寄存器 102: 内部集成电路功率控制寄存器 (PCI2C), 偏移量 0x920

PCI2C 寄存器控制提供到 I<sup>2</sup>C 模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生更改, 并且 RCGCI2C、SCGCI2C 或 DCGCI2C 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRI2C 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

注意: I<sup>2</sup>C 模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

### 内部集成电路功率控制寄存器 (PCI2C)

基址 0x400F.E000  
偏移量 0x920  
类型 R/W, 复位 0x0000.003F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	P5	R/W	1	<p>I<sup>2</sup>C 模块 5 功率控制</p> <p>除非相应主动模式 RCGCI2C、SCGCI2C 或 DCGCI2C 位清零, 否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 I<sup>2</sup>C 模块 5 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 I<sup>2</sup>C 模块 5 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>
4	P4	R/W	1	<p>I<sup>2</sup>C 模块 4 功率控制</p> <p>除非相应主动模式 RCGCI2C、SCGCI2C 或 DCGCI2C 位清零, 否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 I<sup>2</sup>C 模块 4 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 I<sup>2</sup>C 模块 4 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>

位/域	名称	类型	复位	描述
3	P3	R/W	1	<p>I<sup>2</sup>C 模块 3 功率控制</p> <p>除非相应主动模式 RCGCI2C、SCGCI2C 或 DCGCI2C 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 I<sup>2</sup>C 模块 3 通电，但并未接收时钟。在这种情况下，该模块无效。该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 I<sup>2</sup>C 模块 3 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
2	P2	R/W	1	<p>I<sup>2</sup>C 模块 2 功率控制</p> <p>除非相应主动模式 RCGCI2C、SCGCI2C 或 DCGCI2C 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 I<sup>2</sup>C 模块 2 通电，但并未接收时钟。在这种情况下，该模块无效。该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 I<sup>2</sup>C 模块 2 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
1	P1	R/W	1	<p>I<sup>2</sup>C 模块 1 功率控制</p> <p>除非相应主动模式 RCGCI2C、SCGCI2C 或 DCGCI2C 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 I<sup>2</sup>C 模块 1 通电，但并未接收时钟。在这种情况下，该模块无效。该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 I<sup>2</sup>C 模块 1 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
0	P0	R/W	1	<p>I<sup>2</sup>C 模块 0 功率控制</p> <p>除非相应主动模式 RCGCI2C、SCGCI2C 或 DCGCI2C 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 I<sup>2</sup>C 模块 0 通电，但并未接收时钟。在这种情况下，该模块无效。该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 I<sup>2</sup>C 模块 0 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

### 寄存器 103: 通用串行总线功率控制寄存器 ( PCUSB ) , 偏移量 0x928

PCUSB 寄存器控制提供到 USB 模块的功率。清零模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时，模块的状态未保留。如果主动模式发生更改，并且 RCGCUSB、SCGCUSB 或 DCGCUSB 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1，则软件应使用 SRUSB 寄存器执行外设复位。由于状态丢失而重新启用时，软件必须重新初始化模块。

#### 通用串行总线功率控制寄存器 (PCUSB)

基址 0x400F.E000  
偏移量 0x928  
类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	R/W	1	USB 模块功率控制 除非相应主动模式 RCGCUSB、SCGCUSB 或 DCGCUSB 位清零，否则该位的值不具有意义。

**值 描述**

- 1 USB 模块通电，但并未接收时钟。在这种情况下，该模块无效。  
该配置提供了第二低的模块功耗，因为它仅消耗漏流。
- 0 USB 模块未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。  
该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。

## 寄存器 104: 控制器局域网功率控制寄存器 ( PCCAN ) , 偏移量 0x934

PCCAN 寄存器控制提供到 CAN 模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生变更, 并且 RCGCCAN、SCGCCAN 或 DCGCCAN 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRCAN 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

### 控制器局域网功率控制寄存器 (PCCAN)

基址 0x400F.E000  
偏移量 0x934  
类型 R/W, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	R/W	1	<p>CAN 模块 1 功率控制</p> <p>除非相应主动模式 RCGCCAN、SCGCCAN 或 DCGCCAN 位清零, 否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 CAN 模块 1 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 CAN 模块 1 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>
0	P0	R/W	1	<p>CAN 模块 0 功率控制</p> <p>除非相应主动模式 RCGCCAN、SCGCCAN 或 DCGCCAN 位清零, 否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 CAN 模块 0 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 CAN 模块 0 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>

### 寄存器 105: 模数转换器功率控制寄存器 (PCADC) , 偏移量 0x938

PCADC 寄存器控制提供到 ADC 模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生更改, 并且 RCGADC、SCGCADC 或 DCGADC 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRADC 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** ADC 模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

#### 模数转换器功率控制寄存器 (PCADC)

基址 0x400FE000  
偏移量 0x938  
类型 R/W, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	R/W	1	<p>ADC 模块 1 功率控制</p> <p>除非相应主动模式 RCGADC、SCGCADC 或 DCGADC 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 ADC 模块 1 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 ADC 模块 1 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>
0	P0	R/W	1	<p>ADC 模块 0 功率控制</p> <p>除非相应主动模式 RCGADC、SCGCADC 或 DCGADC 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 ADC 模块 0 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 ADC 模块 0 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>



## 寄存器 106: 模拟比较器功率控制寄存器 (PCACMP), 偏移量 0x93C

PCACMP 寄存器控制提供到模拟比较器模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生更改, 并且 RCGCACMP、SCGCACMP 或 DCGCACMP 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRACMP 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** 模拟比较器模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

### 模拟比较器功率控制寄存器 (PCACMP)

基址 0x400F.E000  
偏移量 0x93C  
类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	P0	R/W	1	模拟比较器模块 0 功率控制 除非相应主动模式 RCGCACMP、SCGCACMP 或 DCGCACMP 位清零, 否则该位的值不具有意义。

#### 值 描述

- 1 模拟比较器模块通电, 但并未接收时钟。在这种情况下, 该模块无效。  
该配置提供了第二低的模块功耗, 因为它仅消耗漏流。
- 0 模拟比较器模块未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。  
该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。

### 寄存器 107: 脉宽调制器功率控制寄存器 (PCPWM), 偏移量 0x940

PCPWM 寄存器控制提供到 PWM 模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生更改, 并且 RCGCPWM、SCGCPWM 或 DCGCPWM 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRPWM 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** PWM 模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

#### 脉宽调制器功率控制寄存器 (PCPWM)

基址 0x400F.E000  
偏移量 0x940  
类型 R/W, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	R/W	1	<p>PWM 模块 1 功率控制</p> <p>除非相应主动模式 RCGCPWM、SCGCPWM 或 DCGCPWM 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 PWM 模块 1 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 PWM 模块 1 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>
0	P0	R/W	1	<p>PWM 模块 0 功率控制</p> <p>除非相应主动模式 RCGCPWM、SCGCPWM 或 DCGCPWM 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 PWM 模块 0 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 PWM 模块 0 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>

## 寄存器 108: 正交编码器接口功率控制寄存器 (PCQEI), 偏移量 0x944

PCQEI 寄存器控制提供到 QEI 模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生变更, 并且 RCGCQEI、SCGCQEI 或 DCGCQEI 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRQEI 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** QEI 模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

### 正交编码器接口功率控制寄存器 (PCQEI)

基址 0x400FE000

偏移量 0x944

类型 R/W, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	P1	R/W	1	<p>QEI 模块 1 功率控制</p> <p>除非相应主动模式 RCGCQEI、SCGCQEI 或 DCGCQEI 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 QEI 模块 1 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 QEI 模块 1 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>
0	P0	R/W	1	<p>QEI 模块 0 功率控制</p> <p>除非相应主动模式 RCGCQEI、SCGCQEI 或 DCGCQEI 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 QEI 模块 0 通电, 但并未接收时钟。在这种情况下, 该模块无效。该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 QEI 模块 0 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>

### 寄存器 109: 32/64 位宽通用定时器功率控制寄存器 (PCWTIMER), 偏移量 0x95C

PCWTIMER 寄存器控制提供给 32/64 位宽定时器模块的功率。清零与其中一个模块对应的位即向硬件指示固件请求关闭外设。当该寄存器中的位置位时, 相应模块的状态未保留。如果主动模式发生更改, 并且 RCGCWTIMER、SCGCWTIMER 或 DCGCWTIMER 寄存器中的相应位为 1 或 Pn 位从 0 更改为 1, 则软件应使用 SRWTIMER 寄存器执行外设复位。由于状态丢失而重新启用时, 软件必须重新初始化模块。

**注意:** 定时器模块当前不具有对关闭请求做出反应的能力。置位该寄存器中的位对功耗没有影响。该寄存器用于提供未来的软件兼容性。

#### 32/64 位宽通用定时器功率控制寄存器 (PCWTIMER)

基址 0x400F.E000  
偏移量 0x95C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											P5	P4	P3	P2	P1	P0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	P5	R/W	1	<p>32/64 位宽通用定时器 5 功率控制</p> <p>除非相应主动模式 RCGCWTIMER、SCGCWTIMER 或 DCGCTIMER 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 定时器模块 5 通电, 但并未接收时钟。在这种情况下, 该模块无效。 该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 定时器模块 5 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。 该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>
4	P4	R/W	1	<p>32/64 位宽通用定时器 4 功率控制</p> <p>除非相应主动模式 RCGCWTIMER、SCGCWTIMER 或 DCGCTIMER 位清零, 否则该位的值不具有意义。</p> <p><b>值 描述</b></p> <p>1 定时器模块 4 通电, 但并未接收时钟。在这种情况下, 该模块无效。 该配置提供了第二低的模块功耗, 因为它仅消耗漏流。</p> <p>0 定时器模块 4 未通电, 并且未接收时钟。在这种情况下, 该模块的状态未保留。 该配置提供了最低的模块功耗状态, 因为它即没有动态功耗也不消耗漏流。</p>

位/域	名称	类型	复位	描述
3	P3	R/W	1	<p>32/64 位宽通用定时器 3 功率控制</p> <p>除非相应主动模式 RCGCWTIMER、SCGCWTIMER 或 DCGCTIMER 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 定时器模块 3 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 定时器模块 3 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
2	P2	R/W	1	<p>32/64 位宽通用定时器 2 功率控制</p> <p>除非相应主动模式 RCGCWTIMER、SCGCWTIMER 或 DCGCTIMER 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 定时器模块 2 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 定时器模块 2 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>
1	P1	R/W	1	<p>32/64 位宽通用定时器 1 功率控制</p> <p>除非相应主动模式 RCGCWTIMER、SCGCWTIMER 或 DCGCTIMER 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 定时器模块 1 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 定时器模块 1 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

---

位/域	名称	类型	复位	描述
0	P0	R/W	1	<p>32/64 位宽通用定时器 0 功率控制</p> <p>除非相应主动模式 RCGCWTIMER、SCGCWTIMER 或 DCGCTIMER 位清零，否则该位的值不具有意义。</p> <p>值 描述</p> <p>1 定时器模块 0 通电，但并未接收时钟。在这种情况下，该模块无效。</p> <p>该配置提供了第二低的模块功耗，因为它仅消耗漏流。</p> <p>0 定时器模块 0 未通电，并且未接收时钟。在这种情况下，该模块的状态未保留。</p> <p>该配置提供了最低的模块功耗状态，因为它即没有动态功耗也不消耗漏流。</p>

## 寄存器 110: 看门狗定时器外设就绪寄存器 ( PRWD ) , 偏移量 0xA00

PRWD 寄存器表示看门狗模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCWD 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCWD 位发生更改, 则运行模式计时更改开始。如果相应 SRWD 位从 0 更改为 1, 则复位更改开始。

PRWD 位由于任何上述事件清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

### 看门狗定时器外设就绪寄存器 (PRWD)

基址 0x400F.E000

偏移量 0xA00

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	看门狗定时器 1 外设就绪  值 描述 1 看门狗模块 1 已可以访问。 0 看门狗模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	看门狗定时器 0 外设就绪  值 描述 1 看门狗模块 0 已可以访问。 0 看门狗模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

### 寄存器 111: 16/32 位通用定时器外设就绪寄存器 ( PRTIMER ) , 偏移量 0xA04

PRTIMER 寄存器表示定时器模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCTIMER 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCTIMER 位发生更改, 则运行模式计时更改开始。如果相应 SRTIMER 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRTIMER 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

#### 16/32 位通用定时器外设就绪寄存器 (PRTIMER)

基址 0x400F.E000  
偏移量 0xA04  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	16/32 位通用定时器 5 外设就绪  值 描述 1 16/32 位定时器模块 5 已可以访问。 0 16/32 位定时器模块 5 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
4	R4	R/W	0	16/32 位通用定时器 4 外设就绪  值 描述 1 16/32 位定时器模块 4 已可以访问。 0 16/32 位定时器模块 4 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
3	R3	R/W	0	16/32 位通用定时器 3 外设就绪  值 描述 1 16/32 位定时器模块 3 已可以访问。 0 16/32 位定时器模块 3 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。



位/域	名称	类型	复位	描述
2	R2	R/W	0	16/32 位通用定时器 2 外设就绪  值 描述 1 16/32 位定时器模块 2 已可以访问。 0 16/32 位定时器模块 2 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
1	R1	R/W	0	16/32 位通用定时器 1 外设就绪  值 描述 1 16/32 位定时器模块 1 已可以访问。 0 16/32 位定时器模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	16/32 位通用定时器 0 外设就绪  值 描述 1 16/32 位定时器模块 0 已可以访问。 0 16/32 位定时器模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

### 寄存器 112: 通用输入/输出外设就绪寄存器 ( PRGPIO ) , 偏移量 0xA08

PRGPIO 寄存器表示 GPIO 模块是否就绪，可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCGPIO 位从 0 更改为 1，则功率更改开始。如果相应 RCGCGPIO 位发生更改，则运行模式计时更改开始。如果相应 SRGPIO 位从 0 更改为 1，则复位更改开始。

发生任何上述事件时，PRGPIO 位将清零，并且不会再次置位，直到模块完全通电、启用并内部复位。

#### 通用输入/输出外设就绪寄存器 (PRGPIO)

基址 0x400F.E000  
偏移量 0xA08  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
13	R13	R/W	0	GPIO 端口 P 外设就绪  值 描述 1 GPIO 端口 P 已可以访问。 0 GPIO 端口 P 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
12	R12	R/W	0	GPIO 端口 N 外设就绪  值 描述 1 GPIO 端口 N 已可以访问。 0 GPIO 端口 N 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
11	R11	R/W	0	GPIO 端口 M 外设就绪  值 描述 1 GPIO 端口 M 已可以访问。 0 GPIO 端口 M 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

位/域	名称	类型	复位	描述
10	R10	R/W	0	GPIO 端口 L 外设就绪  值 描述 1 GPIO 端口 L 已可以访问。 0 GPIO 端口 L 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
9	R9	R/W	0	GPIO 端口 K 外设就绪  值 描述 1 GPIO 端口 K 已可以访问。 0 GPIO 端口 K 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
8	R8	R/W	0	GPIO 端口 J 外设就绪  值 描述 1 GPIO 端口 J 已可以访问。 0 GPIO 端口 J 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
7	R7	R/W	0	GPIO 端口 H 外设就绪  值 描述 1 GPIO 端口 H 已可以访问。 0 GPIO 端口 H 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
6	R6	R/W	0	GPIO 端口 G 外设就绪  值 描述 1 GPIO 端口 G 已可以访问。 0 GPIO 端口 G 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
5	R5	R/W	0	GPIO 端口 F 外设就绪  值 描述 1 GPIO 端口 F 已可以访问。 0 GPIO 端口 F 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
4	R4	R/W	0	GPIO 端口 E 外设就绪  值 描述 1 GPIO 端口 E 已可以访问。 0 GPIO 端口 E 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

位/域	名称	类型	复位	描述
3	R3	R/W	0	GPIO 端口 D 外设就绪  值 描述 1 GPIO 端口 D 已可以访问。 0 GPIO 端口 D 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
2	R2	R/W	0	GPIO 端口 C 外设就绪  值 描述 1 GPIO 端口 C 已可以访问。 0 GPIO 端口 C 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
1	R1	R/W	0	GPIO 端口 B 外设就绪  值 描述 1 GPIO 端口 B 已可以访问。 0 GPIO 端口 B 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	GPIO 端口 A 外设就绪  值 描述 1 GPIO 端口 A 已可以访问。 0 GPIO 端口 A 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

**寄存器 113: 微型直接存储器访问外设就绪寄存器 ( PRDMA ) , 偏移量 0xA0C**

PRDMA 寄存器表示  $\mu$ DMA 模块是否就绪，可由软件在功率状态发生改变、运行模式计时或复位之后访问。如果相应 PCDMA 位从 0 更改为 1，则功率更改开始。如果相应 RCGCDMA 位发生改变，则运行模式计时更改开始。如果相应 SRDMA 位从 0 更改为 1，则复位更改开始。

发生任何上述事件时，PRDMA 位将清零，并且不会再次置位，直到模块完全通电、启用并内部复位。

**微型直接存储器访问外设就绪寄存器 (PRDMA)**

基址 0x400F.E000  
偏移量 0xA0C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
	R0															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	$\mu$ DMA 模块外设就绪
				<b>值 描述</b>
				1 $\mu$ DMA 模块已可以访问。
				0 $\mu$ DMA 模块尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

### 寄存器 114: 休眠外设就绪寄存器 ( PRHIB ) , 偏移量 0xA14

PRHIB 寄存器表示休眠模块是否就绪，可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCHIB 位从 0 更改为 1，则功率更改开始。如果相应 RCGCHIB 位发生更改，则运行模式计时更改开始。如果相应 SRHIB 位从 0 更改为 1，则复位更改开始。

发生任何上述事件时，PRHIB 位将清零，并且不会再次置位，直到模块完全通电、启用并内部复位。

#### 休眠外设就绪寄存器 (PRHIB)

基址 0x400F.E000  
偏移量 0xA14  
类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	1	休眠模块外设就绪
				值 描述
				1 休眠模块已可以访问。
				0 休眠模块尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

## 寄存器 115: 通用异步收发器外设就绪寄存器 ( PRUART ) , 偏移量 0xA18

PRUART 寄存器表示 UART 模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCUART 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCUART 位发生更改, 则运行模式计时更改开始。如果相应 SRUART 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRUART 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

### 通用异步收发器外设就绪寄存器 (PRUART)

基址 0x400F.E000

偏移量 0xA18

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								R7	R6	R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	R7	R/W	0	UART 模块 7 外设就绪  值 描述 1 UART 模块 7 已可以访问。 0 UART 模块 7 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
6	R6	R/W	0	UART 模块 6 外设就绪  值 描述 1 UART 模块 6 已可以访问。 0 UART 模块 6 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
5	R5	R/W	0	UART 模块 5 外设就绪  值 描述 1 UART 模块 5 已可以访问。 0 UART 模块 5 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

位/域	名称	类型	复位	描述
4	R4	R/W	0	UART 模块 4 外设就绪  值 描述 1 UART 模块 4 已可以访问。 0 UART 模块 4 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
3	R3	R/W	0	UART 模块 3 外设就绪  值 描述 1 UART 模块 3 已可以访问。 0 UART 模块 3 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
2	R2	R/W	0	UART 模块 2 外设就绪  值 描述 1 UART 模块 2 已可以访问。 0 UART 模块 2 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
1	R1	R/W	0	UART 模块 1 外设就绪  值 描述 1 UART 模块 1 已可以访问。 0 UART 模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	UART 模块 0 外设就绪  值 描述 1 UART 模块 0 已可以访问。 0 UART 模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。



## 寄存器 116: 同步串行接口外设就绪寄存器 ( PRSSI ) , 偏移量 0xA1C

PRSSI 寄存器表示 SSI 模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCSSI 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCSSI 位发生更改, 则运行模式计时更改开始。如果相应 SRSSI 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRSSI 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

### 同步串行接口外设就绪寄存器 (PRSSI)

基址 0x400F.E000

偏移量 0xA1C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												R3	R2	R1	R0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	R3	R/W	0	SSI 模块 3 外设就绪  值 描述 1 SSI 模块 3 已可以访问。 0 SSI 模块 3 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
2	R2	R/W	0	SSI 模块 2 外设就绪  值 描述 1 SSI 模块 2 已可以访问。 0 SSI 模块 2 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
1	R1	R/W	0	SSI 模块 1 外设就绪  值 描述 1 SSI 模块 1 已可以访问。 0 SSI 模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

位/域	名称	类型	复位	描述
0	R0	R/W	0	SSI 模块 0 外设就绪
				值 描述
				1 SSI 模块 0 已可以访问。
				0 SSI 模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

## 寄存器 117: 内部集成电路外设就绪寄存器 ( PRI2C ) , 偏移量 0xA20

PRI2C 寄存器表示 I<sup>2</sup>C 模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCI2C 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCI2C 位发生更改, 则运行模式计时更改开始。如果相应 SRI2C 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRI2C 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

### 内部集成电路外设就绪寄存器 (PRI2C)

基址 0x400F.E000

偏移量 0xA20

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	I <sup>2</sup> C 模块 5 外设就绪  值 描述 1 I <sup>2</sup> C 模块 5 已可以访问。 0 I <sup>2</sup> C 模块 5 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
4	R4	R/W	0	I <sup>2</sup> C 模块 4 外设就绪  值 描述 1 I <sup>2</sup> C 模块 4 已可以访问。 0 I <sup>2</sup> C 模块 4 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
3	R3	R/W	0	I <sup>2</sup> C 模块 3 外设就绪  值 描述 1 I <sup>2</sup> C 模块 3 已可以访问。 0 I <sup>2</sup> C 模块 3 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

---

位/域	名称	类型	复位	描述
2	R2	R/W	0	I <sup>2</sup> C 模块 2 外设就绪  值 描述 1 I <sup>2</sup> C 模块 2 已可以访问。 0 I <sup>2</sup> C 模块 2 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
1	R1	R/W	0	I <sup>2</sup> C 模块 1 外设就绪  值 描述 1 I <sup>2</sup> C 模块 1 已可以访问。 0 I <sup>2</sup> C 模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	I <sup>2</sup> C 模块 0 外设就绪  值 描述 1 I <sup>2</sup> C 模块 0 已可以访问。 0 I <sup>2</sup> C 模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

## 寄存器 118: 通用串行总线外设就绪寄存器 ( PRUSB ) , 偏移量 0xA28

PRUSB 寄存器表示 USB 模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCUSB 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCUSB 位发生更改, 则运行模式计时更改开始。如果相应 SRUSB 位从 0 更改为 1, 则复位更改开始。

生任何上述事件时, PRUSB 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

### 通用串行总线外设就绪寄存器 (PRUSB)

基址 0x400F.E000  
偏移量 0xA28  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	USB 模块外设就绪

#### 值 描述

- 1 USB 模块已可以访问。
- 0 USB 模块尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

### 寄存器 119: 控制器局域网外设就绪寄存器 ( PRCAN ) , 偏移量 0xA34

PRCAN 寄存器表示 CAN 模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCCAN 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCCAN 位发生更改, 则运行模式计时更改开始。如果相应 SRCAN 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRCAN 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

#### 控制器局域网外设就绪寄存器 (PRCAN)

基址 0x400F.E000  
偏移量 0xA34  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留														R1	R0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	CAN 模块 1 外设就绪  值 描述 1 CAN 模块 1 已可以访问。 0 CAN 模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	CAN 模块 0 外设就绪  值 描述 1 CAN 模块 0 已可以访问。 0 CAN 模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

## 寄存器 120: 模数转换器外设就绪寄存器 ( PRADC ) , 偏移量 0xA38

PRADC 寄存器表示 ADC 模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCADC 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCADC 位发生更改, 则运行模式计时更改开始。如果相应 SRADC 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRADC 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

### 模数转换器外设就绪寄存器 (PRADC)

基址 0x400F.E000

偏移量 0xA38

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	ADC 模块 1 外设就绪  值 描述 1 ADC 模块 1 已可以访问。 0 ADC 模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	ADC 模块 0 外设就绪  值 描述 1 ADC 模块 0 已可以访问。 0 ADC 模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

### 寄存器 121: 模拟比较器外设就绪寄存器 ( PRACMP ) , 偏移量 0xA3C

PRACMP 寄存器表示模拟比较器模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCACMP 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCACMP 位发生更改, 则运行模式计时更改开始。如果相应 SRACMP 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRACMP 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

#### 模拟比较器外设就绪寄存器 (PRACMP)

基址 0x400F.E000  
偏移量 0xA3C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
	R0															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	R0	R/W	0	模拟比较器模块 0 外设就绪

值 描述

- 1 模拟比较器模块已可以访问。
- 0 模拟比较器模块尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。



## 寄存器 122: 脉宽调制器外设就绪寄存器 ( PRPWM ) , 偏移量 0xA40

PRPWM 寄存器表示 PWM 模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCPWM 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCPWM 位发生更改, 则运行模式计时更改开始。如果相应 SRPWM 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRPWM 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

### 脉宽调制器外设就绪寄存器 (PRPWM)

基址 0x400F.E000  
偏移量 0xA40  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留														R1	R0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	PWM 模块 1 外设就绪  <b>值 描述</b> 1 PWM 模块 1 已可以访问。 0 PWM 模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	PWM 模块 0 外设就绪  <b>值 描述</b> 1 PWM 模块 0 已可以访问。 0 PWM 模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

### 寄存器 123: 正交编码器接口外设就绪寄存器 ( PRQEI ) , 偏移量 0xA44

PRQEI 寄存器表示 QEI 模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCQEI 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCQEI 位发生更改, 则运行模式计时更改开始。如果相应 SRQEI 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRQEI 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

#### 正交编码器接口外设就绪寄存器 (PRQEI)

基址 0x400F.E000  
偏移量 0xA44  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留														R1	R0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	R1	R/W	0	QEI 模块 1 外设就绪  值 描述 1 QEI 模块 1 已可以访问。 0 QEI 模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	QEI 模块 0 外设就绪  值 描述 1 QEI 模块 0 已可以访问。 0 QEI 模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

## 寄存器 124: 32/64 位宽通用定时器外设就绪寄存器 ( PRWTIMER ) , 偏移量 0xA5C

PRWTIMER 寄存器表示定时器模块是否就绪, 可由软件在功率状态发生更改、运行模式计时或复位之后访问。如果相应 PCWTIMER 位从 0 更改为 1, 则功率更改开始。如果相应 RCGCWTIMER 位发生更改, 则运行模式计时更改开始。如果相应 SRWTIMER 位从 0 更改为 1, 则复位更改开始。

发生任何上述事件时, PRWTIMER 位将清零, 并且不会再次置位, 直到模块完全通电、启用并内部复位。

### 32/64 位宽通用定时器外设就绪寄存器 (PRWTIMER)

基址 0x400F.E000

偏移量 0xA5C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											R5	R4	R3	R2	R1	R0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	R5	R/W	0	32/64 位宽通用定时器 5 外设就绪  值 描述 1 32/64 位宽定时器模块 5 已可以访问。 0 32/64 位宽定时器模块 5 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
4	R4	R/W	0	32/64 位宽通用定时器 4 外设就绪  值 描述 1 32/64 位宽定时器模块 4 已可以访问。 0 32/64 位宽定时器模块 4 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
3	R3	R/W	0	32/64 位宽通用定时器 3 外设就绪  值 描述 1 32/64 位宽定时器模块 3 已可以访问。 0 32/64 位宽定时器模块 3 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

位/域	名称	类型	复位	描述
2	R2	R/W	0	32/64 位宽通用定时器 2 外设就绪  值 描述 1 32/64 位宽定时器模块 2 已可以访问。 0 32/64 位宽定时器模块 2 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
1	R1	R/W	0	32/64 位宽通用定时器 1 外设就绪  值 描述 1 32/64 位宽定时器模块 1 已可以访问。 0 32/64 位宽定时器模块 1 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。
0	R0	R/W	0	32/64 位宽通用定时器 0 外设就绪  值 描述 1 32/64 位宽定时器模块 0 已可以访问。 0 32/64 位宽定时器模块 0 尚不可以访问。未使用时钟、未通电或处于完成复位序列的过程中。

## 5.6 系统控制传统寄存器描述

所有给出的地址都是相对于 0x400F.E000 的系统控制基址而言的。

**重要:** 本节中的寄存器仅用于实现传统软件支持；应使用“系统控制寄存器描述”在 222 页中的寄存器。

**寄存器 125: 器件功能寄存器 0 ( DC0 ) , 偏移量 0x008**

该传统寄存器由器件预定义且可用于校验特性。

**重要:** 该寄存器用于实现传统软件支持。

应使用 Flash 大小 (FSIZE) 和 SRAM 大小 (SSIZE) 寄存器确定该微控制器的存储器大小。读 DC0 即可正确识别传统存储器大小，但是软件必须针对以下未列出的存储器大小使用 FSIZE 和 SSIZE。

**器件功能寄存器 0 (DC0)**

基址 0x400FE000  
偏移量 0x008  
类型 RO, 复位 0x007F.007F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRAMSZ															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLASHSZ															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:16	SRAMSZ	RO	0x7F	SRAM 大小 显示了片上 SRAM 的大小。
				值 描述
				0x7 2KB SRAM
				0xF 4 KB SRAM
				0x17 6 KB SRAM
				0x1F 8 KB SRAM
				0x2F 12 KB SRAM
				0x3F 16 KB SRAM
				0x4F 20 KB SRAM
				0x5F 24 KB SRAM
				0x7F 32 KB SRAM

位/域	名称	类型	复位	描述
15:0	FLASHSZ	RO	0x7F	Flash 大小 表示片内 Flash 存储器的大小。
				值 描述
				0x3 8KB Flash
				0x7 16 KB Flash
				0xF 32 KB Flash
				0x1F 64 KB Flash
				0x2F 96 KB Flash
				0x3F 128 KB Flash
				0x5F 192 KB Flash
				0x7F 256 KB Flash

## 寄存器 126: 器件功能寄存器 1 ( DC1 ) , 偏移量 0x010

该寄存器由器件预定义且可用于校验特性。如果该寄存器中的某位是零, 那么对应模块不存在。在 RCGC0、SCGC0、DCGC0 和外设专用 RCGC、SCGC 和 DCGC 寄存器中的对应位不能置位。

**重要:** 该寄存器用于实现传统软件支持。

应使用外设存在寄存器确定在该微控制器上实施的模块。读 DC1 即可正确识别传统模块是否存在, 但是软件必须使用外设存在寄存器来确定不受 DCn 寄存器支持的模块是否存在。

同样, 应使用 ADC 外设属性 (ADCPP) 寄存器确定最大 ADC 采样率, 以及温度传感器是否存在。但是, 要支持传统软件, 可使用 MAXADCnSPD 域和 TEMPSNS 位。读 DC1 即可正确识别传统比率的最大 ADC 采样率, 以及温度传感器是否存在。

### 器件功能寄存器 1 (DC1)

基址 0x400F.E000

偏移量 0x010

类型 RO, 复位 0x1333.2FFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			WDT1	保留		CAN1	CAN0	保留		PWM1	PWM0	保留		ADC1	ADC0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MINSYSDIV			MAXADC1SPD		MAXADC0SPD		MPU	HIB	TEMPSNS	PLL	WDT0	SWO	SWD	JTAG	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:29	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
28	WDT1	RO	0x1	看门狗定时器 1 存在 置位表示看门狗定时器 1 存在。
27:26	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
25	CAN1	RO	0x1	CAN 模块 1 存在 置位表示 CAN 单元 1 存在。
24	CAN0	RO	0x1	CAN 模块 0 存在 置位表示 CAN 单元 0 存在。
23:22	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
21	PWM1	RO	0x1	PWM 模块 1 存在 置位表示 PWM 模块存在。
20	PWM0	RO	0x1	PWM 模块 0 存在 置位表示 PWM 模块存在。
19:18	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
17	ADC1	RO	0x1	ADC 模块 1 存在 置位表示 ADC 模块 1 存在。
16	ADC0	RO	0x1	ADC 模块 0 存在 置位表示 ADC 模块 0 存在。
15:12	MINSYSDIV	RO	0x2	系统时钟分频器 用于系统时钟的最小 4 位分频器。复位值取决于硬件。查看 RCC 寄存器可获知如何使用 SYSDIV 位来改变系统时钟。  值 描述 0x1 指定一个 PLL 2.5 分频的 80-MHz CPU 时钟。 0x2 指定一个 PLL 3 分频的 66-MHz CPU 时钟。 0x3 指定一个 PLL 4 分频的 50-MHz CPU 时钟。 0x4 指定一个 PLL 5 分频的 40-MHz CPU 时钟。 0x7 指定一个 PLL 8 分频的 25-MHz 时钟。 0x9 指定一个 PLL 10 分频的 20-MHz 时钟。
11:10	MAXADC1SPD	RO	0x3	ADC1 最大速率 该域规定了 ADC 采样数据的最大速率。  值 描述 0x3 1M 采样/秒 0x2 500K 采样/秒 0x1 250K 采样/秒 0x0 125K 采样/秒
9:8	MAXADC0SPD	RO	0x3	ADC0 最快速度 该域规定了 ADC 采样数据的最大速率。  值 描述 0x3 1M 采样/秒 0x2 500K 采样/秒 0x1 250K 采样/秒 0x0 125K 采样/秒
7	MPU	RO	0x1	MPU 存在 置位表示 CortexM4F 存储器保护单元 (MPU) 模块存在。关于 MPU 的详细信息可查看“CortexM4F”一章。
6	HIB	RO	0x1	休眠模块存在 置位表示休眠模块存在。
5	TEMPSNS	RO	0x1	温度传感器存在 置位表示片上温度传感器存在。
4	PLL	RO	0x1	PLL 存在 置位表示片上锁相环 (PLL) 存在。



---

位/域	名称	类型	复位	描述
3	WDT0	RO	0x1	看门狗定时器 0 存在 置位表示看门狗定时器 0 存在。
2	SWO	RO	0x1	SWO 跟踪端口存在 置位表示串行线输出 (SWO) 跟踪端口存在。
1	SWD	RO	0x1	SWD 存在 置位表示串行线调试器 (SWD) 存在。
0	JTAG	RO	0x1	JTAG 存在 置位表示 JTAG 调试器接口存在。

## 寄存器 127: 器件功能寄存器 2 ( DC2 ) , 偏移量 0x014

该寄存器由器件预定义且可用于校验特性。如果该寄存器中的某位是零，那么对应模块不存在。在 RCGC1、SCGC1、DCGC1 和外设专用 RCGC、SCGC 和 DCGC 寄存器中的对应位不能置位。

**重要:** 该寄存器用于实现传统软件支持。

应使用外设存在寄存器确定在该微控制器上实施的模块。读 DC2 即可正确识别传统模块是否存在，但是软件必须使用外设存在寄存器来确定不受 DCn 寄存器支持的模块是否存在。

请注意模拟比较器外设存在 (PPACMP) 寄存器识别模拟比较器模块是否存在。模拟比较器外设属性 (ACMPPP) 寄存器表示在模块中存在的模拟比较器块数量。

### 器件功能寄存器 2 (DC2)

基址 0x400F.E000

偏移量 0x014

类型 RO, 复位 0x070F.F337

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留	EPI0	保留	I2S0	保留	COMP2	COMP1	COMP0	保留			TIMER3	TIMER2	TIMER1	TIMER0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2C1HS	I2C1	I2C0HS	I2C0	保留		QE1	QE0	保留		SSI1	SSI0	保留	UART2	UART1	UART0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	0	0	1	1	0	0	1	1	0	1	1	1

位/域	名称	类型	复位	描述
31	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
30	EPI0	RO	0x0	EPI 模块 0 存在 置位表示 EPI 模块 0 存在。
29	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
28	I2S0	RO	0x0	I2S 模块 0 存在 置位表示 I2S 模块 0 存在。
27	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
26	COMP2	RO	0x1	模拟比较器 2 存在 置位表示模拟比较器 2 存在。
25	COMP1	RO	0x1	模拟比较器 1 存在 置位表示模拟比较器 1 存在。
24	COMP0	RO	0x1	模拟比较器 0 存在 置位表示模拟比较器 0 存在。
23:20	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
19	TIMER3	RO	0x1	定时器模块 3 存在 置位表示通用定时器模块 3 存在。
18	TIMER2	RO	0x1	定时器模块 2 存在 置位表示通用定时器模块 2 存在。
17	TIMER1	RO	0x1	定时器模块 1 存在 置位表示通用定时器模块 1 存在。
16	TIMER0	RO	0x1	定时器模块 0 存在 置位表示通用定时器模块 0 存在。
15	I2C1HS	RO	0x1	I2C 模块 1 速度 置位表示 I2C 模块 1 可以在高速模式中操作。
14	I2C1	RO	0x1	I2C 模块 1 存在 置位表示 I2C 模块 1 存在。
13	I2C0HS	RO	0x1	I2C 模块 0 速度 置位表示 I2C 模块 0 可以在高速模式中操作。
12	I2C0	RO	0x1	I2C 模块 0 存在 置位表示 I2C 模块 0 存在。
11:10	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
9	QEI1	RO	0x1	QEI 模块 1 存在 置位表示 QEI 模块 1 存在。
8	QEI0	RO	0x1	QEI 模块 0 存在 置位表示 QEI 模块 0 存在。
7:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
5	SSI1	RO	0x1	SSI 模块 1 存在 置位表示 SSI 模块 1 存在。
4	SSI0	RO	0x1	SSI 模块 0 存在 置位表示 SSI 模块 0 存在。
3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
2	UART2	RO	0x1	UART 模块 2 存在 置位表示 UART 模块 2 存在。
1	UART1	RO	0x1	UART 模块 1 存在 置位表示 UART 模块 1 存在。
0	UART0	RO	0x1	UART 模块 0 存在 置位表示 UART 模块 0 存在。

## 寄存器 128: 器件功能寄存器 3 ( DC3 ) , 偏移量 0x018

该寄存器由器件预定义且可用于校验特性。如果该寄存器中的某位是零，那么对应功能不存在。

**重要:** 该寄存器用于实现传统软件支持。

对于一些模块来说，应使用外设驻留外设属性寄存器确定在该微控制器上可用的管脚。读 DC3 即可正确识别传统管脚是否存在，但是软件必须使用外设属性寄存器来确定不受 DCn 寄存器支持的管脚是否存在。

### 器件功能寄存器 3 (DC3)

基址 0x400F.E000  
偏移量 0x018  
类型 RO, 复位 0xBFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	32KHZ	保留	CCP5	CCP4	CCP3	CCP2	CCP1	CCP0	ADC0AIN7	ADC0AIN6	ADC0AIN5	ADC0AIN4	ADC0AIN3	ADC0AIN2	ADC0AIN1	ADC0AIN0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWMFAULT	C2O	C2PLUS	C2MINUS	C1O	C1PLUS	C1MINUS	C0O	C0PLUS	C0MINUS	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31	32KHZ	RO	0x1	32KHz 输入时钟可用 置位表示偶数 CCP 管脚存在并且可用作 32 KHz 输入时钟。 注意: GPTMPP 寄存器不提供该信息。
30	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
29	CCP5	RO	0x1	CCP5 管脚存在 置位表示捕捉/比较/PWM 管脚 5 存在。 注意: GPTMPP 寄存器不提供该信息。
28	CCP4	RO	0x1	CCP4 管脚存在 置位表示捕捉/比较/PWM 管脚 4 存在。 注意: GPTMPP 寄存器不提供该信息。
27	CCP3	RO	0x1	CCP3 管脚存在 置位表示捕捉/比较/PWM 管脚 3 存在。 注意: GPTMPP 寄存器不提供该信息。
26	CCP2	RO	0x1	CCP2 管脚存在 置位表示捕捉/比较/PWM 管脚 2 存在。 注意: GPTMPP 寄存器不提供该信息。
25	CCP1	RO	0x1	CCP1 管脚存在 置位表示捕捉/比较/PWM 管脚 1 存在。 注意: GPTMPP 寄存器不提供该信息。

位/域	名称	类型	复位	描述
24	CCP0	RO	0x1	CCP0 管脚存在 置位表示捕捉/比较/PWM 管脚 0 存在。 注意: GPTMPP 寄存器不提供该信息。
23	ADC0AIN7	RO	0x1	ADC 模块 0 AIN7 管脚存在 置位表示 ADC 模块 0 的输入管脚 7 存在。 注意: ADCPP 寄存器中的 CH 域提供该信息。
22	ADC0AIN6	RO	0x1	ADC 模块 0 AIN6 管脚存在 置位表示 ADC 模块 0 的输入管脚 6 存在。 注意: ADCPP 寄存器中的 CH 域提供该信息。
21	ADC0AIN5	RO	0x1	ADC 模块 0 AIN5 管脚存在 置位表示 ADC 模块 0 的输入管脚 5 存在。 注意: ADCPP 寄存器中的 CH 域提供该信息。
20	ADC0AIN4	RO	0x1	ADC 模块 0 AIN4 管脚存在 置位表示 ADC 模块 0 的输入管脚 4 存在。 注意: ADCPP 寄存器中的 CH 域提供该信息。
19	ADC0AIN3	RO	0x1	ADC 模块 0 AIN3 管脚存在 置位表示 ADC 模块 0 的输入管脚 3 存在。 注意: ADCPP 寄存器中的 CH 域提供该信息。
18	ADC0AIN2	RO	0x1	ADC 模块 0 AIN2 管脚存在 置位表示 ADC 模块 0 的输入管脚 2 存在。 注意: ADCPP 寄存器中的 CH 域提供该信息。
17	ADC0AIN1	RO	0x1	ADC 模块 0 AIN1 管脚存在 置位表示 ADC 模块 0 的输入管脚 1 存在。 注意: ADCPP 寄存器中的 CH 域提供该信息。
16	ADC0AIN0	RO	0x1	ADC 模块 0 AIN0 管脚存在 置位表示 ADC 模块 0 的输入管脚 0 存在。 注意: ADCPP 寄存器中的 CH 域提供该信息。
15	PWMFAULT	RO	0x1	PWM 故障管脚存在 置位表示 PWM 故障管脚存在。查看 DC5 可得到该器件的特定错误管脚。 注意: PWMPP 寄存器中的 FCNT 域提供该信息。
14	C2O	RO	0x1	C2o 管脚存在 置位表示模拟比较器 2 输出管脚存在。 注意: ACMPPP 寄存器中的 C2O 位提供该信息。
13	C2PLUS	RO	0x1	C2+ 管脚存在 置位表示模拟比较器 2(+) 输入管脚存在。 注意: 模拟比较器 2 存在时, 此管脚存在。

位/域	名称	类型	复位	描述
12	C2MINUS	RO	0x1	C2- 管脚存在 置位表示模拟比较器 2(-) 输入管脚存在。 注意: 模拟比较器 2 存在时, 此管脚存在。
11	C1O	RO	0x1	C1o 管脚存在 置位表示模拟比较器 1 输出管脚存在。 注意: ACMPPP 寄存器中的 C1O 位提供该信息。
10	C1PLUS	RO	0x1	C1+ 管脚存在 置位表示模拟比较器 1(+) 输入管脚存在。 注意: 模拟比较器 1 存在时, 此管脚存在。
9	C1MINUS	RO	0x1	C1- 管脚存在 置位表示模拟比较器 1(-) 输入管脚存在。 注意: 模拟比较器 1 存在时, 此管脚存在。
8	C0O	RO	0x1	C0o 管脚存在 置位表示模拟比较器 0 输出管脚存在。 注意: ACMPPP 寄存器中的 C0O 位提供该信息。
7	C0PLUS	RO	0x1	C0+ 管脚存在 置位表示模拟比较器 0(+) 输入管脚存在。 注意: 模拟比较器 0 存在时, 此管脚存在。
6	C0MINUS	RO	0x1	C0- 管脚存在 置位表示模拟比较器 0(-) 输入管脚存在。 注意: 模拟比较器 0 存在时, 此管脚存在。
5	PWM5	RO	0x1	PWM5 管脚存在 置位表示 PWM 管脚 5 存在。 注意: PWMPP 寄存器中的 GCNT 域提供该信息。
4	PWM4	RO	0x1	PWM4 管脚存在 置位表示 PWM 管脚 4 存在。 注意: PWMPP 寄存器中的 GCNT 域提供该信息。
3	PWM3	RO	0x1	PWM3 管脚存在 置位表示 PWM 管脚 3 存在。 注意: PWMPP 寄存器中的 GCNT 域提供该信息。
2	PWM2	RO	0x1	PWM2 管脚存在 置位表示 PWM 管脚 2 存在。 注意: PWMPP 寄存器中的 GCNT 域提供该信息。
1	PWM1	RO	0x1	PWM1 管脚存在 置位表示 PWM 管脚 1 存在。 注意: PWMPP 寄存器中的 GCNT 域提供该信息。

---

位/域	名称	类型	复位	描述
0	PWM0	RO	0x1	PWM0 管脚存在 置位表示 PWM 管脚 0 存在。 注意: PWMPP 寄存器中的 GCNT 域提供该信息。

## 寄存器 129: 器件功能寄存器 4 ( DC4 ) , 偏移量 0x01C

该寄存器由器件预定义且可用于校验特性。如果该寄存器中的某位是零, 那么对应模块不存在。在 RCGC2、SCGC2、DCGC2 和外设专用 RCGC、SCGC 和 DCGC 寄存器中的对应位不能置位。

**重要:** 该寄存器用于实现传统软件支持。

应使用外设存在寄存器确定在该微控制器上实施的模块。读 DC4 即可正确识别传统模块是否存在, 但是软件必须使用外设存在寄存器来确定不受 DCn 寄存器支持的模块是否存在。

应使用外设驻留外设属性寄存器确定在该微控制器上可用的管脚和功能。读取 DC4 即可正确识别传统管脚或功能是否存在。软件必须使用外设属性寄存器来确定不受 DCn 寄存器支持的管脚或功能是否存在。

### 器件功能寄存器 4 (DC4)

基址 0x400F.E000  
偏移量 0x01C  
类型 RO, 复位 0x0004.F1FF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留	EPHY0	保留	EMAC0	保留			E1588	保留				PICAL	保留		
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCP7	CCP6	UDMA	ROM	保留			GPIOJ	GPIOH	GPIOG	GPIOF	GPIOE	GIPOD	GPIOC	GPIOB	GPIOA
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
30	EPHY0	RO	0x0	以太网 PHY 层 0 存在 置位表示以太网 PHY 层 0 存在。
29	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
28	EMAC0	RO	0x0	以太网 MAC 层 0 存在 置位表示以太网 MAC 层 0 存在。
27:25	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
24	E1588	RO	0x0	1588 可用 置位表示以太网 MAC 层 0 是 1588 可用。
23:19	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
18	PICAL	RO	0x1	PIOSC 校准 置位表示可以通过软件校准 PIOSC。
17:16	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。



位/域	名称	类型	复位	描述
15	CCP7	RO	0x1	CCP7 管脚存在 置位表示捕捉/比较/PWM 管脚 7 存在。 注意: GPTMPP 寄存器不提供该信息。
14	CCP6	RO	0x1	CCP6 管脚存在 置位表示捕捉/比较/PWM 管脚 6 存在。 注意: GPTMPP 寄存器不提供该信息。
13	UDMA	RO	0x1	微 DMA 模块存在 置位表示微 DMA 模块存在。
12	ROM	RO	0x1	内部 ROM 代码存在 置位表示内部代码 ROM 存在。
11:9	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件,保留位的值在读-修改-写操作过程中应该保持不变。
8	GPIOJ	RO	0x1	GPIO 端口 J 存在 置位表示 GPIO 端口 J 存在。
7	GPIOH	RO	0x1	GPIO 端口 H 存在 置位表示 GPIO 端口 H 存在。
6	GPIOG	RO	0x1	GPIO 端口 G 存在 置位表示 GPIO 端口 G 存在。
5	GPIOF	RO	0x1	GPIO 端口 F 存在 置位表示 GPIO 端口 F 存在。
4	GPIOE	RO	0x1	GPIO 端口 E 存在 置位表示 GPIO 端口 E 存在。
3	GIOD	RO	0x1	GPIO 端口 D 存在 置位表示 GPIO 端口 D 存在。
2	GPIOC	RO	0x1	GPIO 端口 C 存在 置位表示 GPIO 端口 C 存在。
1	GPIOB	RO	0x1	GPIO 端口 B 存在 置位表示 GPIO 端口 B 存在。
0	GPIOA	RO	0x1	GPIO 端口 A 存在 置位表示 GPIO 端口 A 存在。

### 寄存器 130: 器件功能寄存器 5 ( DC5 ) , 偏移量 0x020

该寄存器由器件预定义且可用于校验 PWM 特性。如果该寄存器中的某位是零，那么对应模块不存在。

**重要:** 该寄存器用于实现传统软件支持。

应使用 PWM 外设属性 (PWMPP) 寄存器确定在 PWM 模块上可用的管脚和功能。读该寄存器即可正确识别传统管脚或功能是否存在。软件必须使用 PWMPP 寄存器来确定不受 DCn 寄存器支持的管脚或功能是否存在。

#### 器件功能寄存器 5 (DC5)

基址 0x400F.E000  
偏移量 0x020  
类型 RO, 复位 0x0F30.00FF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留				PWMFAULT3	PWMFAULT2	PWMFAULT1	PWMFAULT0	保留			PWMEFLT	PWMESYNC	保留			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留								PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	

位/域	名称	类型	复位	描述
31:28	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
27	PWMFAULT3	RO	0x1	PWM 错误 3 管脚存在 置位表示 PWM 错误 3 管脚存在。
26	PWMFAULT2	RO	0x1	PWM 错误 2 管脚存在 置位表示 PWM 错误 2 管脚存在。
25	PWMFAULT1	RO	0x1	PWM 错误 1 管脚存在 置位表示 PWM 错误 1 管脚存在。
24	PWMFAULT0	RO	0x1	PWM 错误 0 管脚存在 置位表示 PWM 错误 0 管脚存在。
23:22	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
21	PWMEFLT	RO	0x1	PWM 扩展错误激活 置位表示 PWM 扩展错误特性是激活的。
20	PWMESYNC	RO	0x1	PWM 扩展 SYNC 激活 置位表示 PWM 扩展 SYNC 特性是激活的。
19:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	PWM7	RO	0x1	PWM7 管脚存在 置位表示 PWM 管脚 7 存在。

位/域	名称	类型	复位	描述
6	PWM6	RO	0x1	PWM6 管脚存在 置位表示 PWM 管脚 6 存在。
5	PWM5	RO	0x1	PWM5 管脚存在 置位表示 PWM 管脚 5 存在。
4	PWM4	RO	0x1	PWM4 管脚存在 置位表示 PWM 管脚 4 存在。
3	PWM3	RO	0x1	PWM3 管脚存在 置位表示 PWM 管脚 3 存在。
2	PWM2	RO	0x1	PWM2 管脚存在 置位表示 PWM 管脚 2 存在。
1	PWM1	RO	0x1	PWM1 管脚存在 置位表示 PWM 管脚 1 存在。
0	PWM0	RO	0x1	PWM0 管脚存在 置位表示 PWM 管脚 0 存在。

### 寄存器 131: 器件功能寄存器 6 ( DC6 ) , 偏移量 0x024

该寄存器由器件预定义且可用于校验特性。如果该寄存器中的某位是零，那么对应模块不存在。在 RCGC0、SCGC0 和 DCGC0 寄存器中的对应位不能置位。

**重要:** 该寄存器用于实现传统软件支持。

应使用 USB 外设属性 (USBPP) 寄存器确定在 USB 模块上可用的功能。读该寄存器即可正确识别传统功能是否存在。软件必须使用 USBPP 寄存器来确定不受 DCn 寄存器支持的管脚或功能是否存在。

#### 器件功能寄存器 6 (DC6)

基址 0x400FE000  
偏移量 0x024  
类型 RO, 复位 0x0000.0013

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												USB0PHY	保留		USB0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

位/域	名称	类型	复位	描述
31:5	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
4	USB0PHY	RO	0x1	USB 模块 0 PHY 存在 置位表示 USB 模块 0 PHY 存在。
3:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
1:0	USB0	RO	0x3	USB 模块 0 存在 该域表示 USB 模块 0 存在并规定了它的功能。
				sysValue 描述
			0x0	NA USB0 不存在。
			0x1	DEVICE USB0 仅是器件。
			0x2	主机 USB0 是器件或主机。
			0x3	OTG USB0 是 OTG。

## 寄存器 132: 器件功能寄存器 7 ( DC7 ) , 偏移量 0x028

该寄存器由器件预定义且可用于校验  $\mu$ DMA 通道特性。值为 1 表示该通道在这个器件上可用，值为 0 表示该通道只在这个家族的其它器件上可用。通道可以拥有多个赋值，请参考“通道分配”在 535 页了解更多信息。

**重要:** 该寄存器用于实现传统软件支持。DMA 状态 (DMASTAT) 寄存器的 DMACHANS 位域表示 DMA 通道的数量。

### 器件功能寄存器 7 (DC7)

基址 0x400F.E000

偏移量 0x028

类型 RO, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留	DMACH30	DMACH29	DMACH28	DMACH27	DMACH26	DMACH25	DMACH24	DMACH23	DMACH22	DMACH21	DMACH20	DMACH19	DMACH18	DMACH17	DMACH16
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8	DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31	保留	RO	0x1	DMA 通道 31 置位表示 $\mu$ DMA 通道 31 可用。
30	DMACH30	RO	0x1	DMA 通道 30 置位表示 $\mu$ DMA 通道 30 可用。
29	DMACH29	RO	0x1	DMA 通道 29 置位表示 $\mu$ DMA 通道 29 可用。
28	DMACH28	RO	0x1	DMA 通道 28 置位表示 $\mu$ DMA 通道 28 可用。
27	DMACH27	RO	0x1	DMA 通道 27 置位表示 $\mu$ DMA 通道 27 可用。
26	DMACH26	RO	0x1	DMA 通道 26 置位表示 $\mu$ DMA 通道 26 可用。
25	DMACH25	RO	0x1	DMA 通道 25 置位表示 $\mu$ DMA 通道 25 可用。
24	DMACH24	RO	0x1	DMA 通道 24 置位表示 $\mu$ DMA 通道 24 可用。
23	DMACH23	RO	0x1	DMA 通道 23 置位表示 $\mu$ DMA 通道 23 可用。
22	DMACH22	RO	0x1	DMA 通道 22 置位表示 $\mu$ DMA 通道 22 可用。

位/域	名称	类型	复位	描述
21	DMACH21	RO	0x1	DMA 通道 21 置位表示 $\mu$ DMA 通道 21 可用。
20	DMACH20	RO	0x1	DMA 通道 20 置位表示 $\mu$ DMA 通道 20 可用。
19	DMACH19	RO	0x1	DMA 通道 19 置位表示 $\mu$ DMA 通道 19 可用。
18	DMACH18	RO	0x1	DMA 通道 18 置位表示 $\mu$ DMA 通道 18 可用。
17	DMACH17	RO	0x1	DMA 通道 17 置位表示 $\mu$ DMA 通道 17 可用。
16	DMACH16	RO	0x1	DMA 通道 16 置位表示 $\mu$ DMA 通道 16 可用。
15	DMACH15	RO	0x1	DMA 通道 15 置位表示 $\mu$ DMA 通道 15 可用。
14	DMACH14	RO	0x1	DMA 通道 14 置位表示 $\mu$ DMA 通道 14 可用。
13	DMACH13	RO	0x1	DMA 通道 13 置位表示 $\mu$ DMA 通道 13 可用。
12	DMACH12	RO	0x1	DMA 通道 12 置位表示 $\mu$ DMA 通道 12 可用。
11	DMACH11	RO	0x1	DMA 通道 11 置位表示 $\mu$ DMA 通道 11 可用。
10	DMACH10	RO	0x1	DMA 通道 10 置位表示 $\mu$ DMA 通道 10 可用。
9	DMACH9	RO	0x1	DMA 通道 9 置位表示 $\mu$ DMA 通道 9 可用。
8	DMACH8	RO	0x1	DMA 通道 8 置位表示 $\mu$ DMA 通道 8 可用。
7	DMACH7	RO	0x1	DMA 通道 7 置位表示 $\mu$ DMA 通道 7 可用。
6	DMACH6	RO	0x1	DMA 通道 6 置位表示 $\mu$ DMA 通道 6 可用。
5	DMACH5	RO	0x1	DMA 通道 5 置位表示 $\mu$ DMA 通道 5 可用。
4	DMACH4	RO	0x1	DMA 通道 4 置位表示 $\mu$ DMA 通道 4 可用。

---

位/域	名称	类型	复位	描述
3	DMACH3	RO	0x1	DMA 通道 3 置位表示 $\mu$ DMA 通道 3 可用。
2	DMACH2	RO	0x1	DMA 通道 2 置位表示 $\mu$ DMA 通道 2 可用。
1	DMACH1	RO	0x1	DMA 通道 1 置位表示 $\mu$ DMA 通道 1 可用。
0	DMACH0	RO	0x1	DMA 通道 0 置位表示 $\mu$ DMA 通道 0 可用。

## 寄存器 133: 器件功能寄存器 8 ( DC8 ) , 偏移量 0x02C

该寄存器由器件预定义且可用于校验特性。

**重要:** 该寄存器用于实现传统软件支持。

应使用 ADC 外设属性 (ADCPP) 寄存器确定在 ADC 模块上可用的输入通道数量。读该寄存器即可正确识别传统通道是否存在，但是软件必须使用 ADCPP 寄存器来确定不受 DCn 寄存器支持的通道是否存在。

### 器件功能寄存器 8 (DC8)

基址 0x400F.E000  
偏移量 0x02C  
类型 RO, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADC1AIN15	ADC1AIN14	ADC1AIN13	ADC1AIN12	ADC1AIN11	ADC1AIN10	ADC1AIN9	ADC1AIN8	ADC1AIN7	ADC1AIN6	ADC1AIN5	ADC1AIN4	ADC1AIN3	ADC1AIN2	ADC1AIN1	ADC1AIN0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADC0AIN15	ADC0AIN14	ADC0AIN13	ADC0AIN12	ADC0AIN11	ADC0AIN10	ADC0AIN9	ADC0AIN8	ADC0AIN7	ADC0AIN6	ADC0AIN5	ADC0AIN4	ADC0AIN3	ADC0AIN2	ADC0AIN1	ADC0AIN0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31	ADC1AIN15	RO	0x1	ADC 模块 1 AIN15 管脚存在 置位表示 ADC 模块 1 的输入管脚 15 存在。
30	ADC1AIN14	RO	0x1	ADC 模块 1 AIN14 管脚存在 置位表示 ADC 模块 1 的输入管脚 14 存在。
29	ADC1AIN13	RO	0x1	ADC 模块 1 AIN13 管脚存在 置位表示 ADC 模块 1 的输入管脚 13 存在。
28	ADC1AIN12	RO	0x1	ADC 模块 1 AIN12 管脚存在 置位表示 ADC 模块 1 的输入管脚 12 存在。
27	ADC1AIN11	RO	0x1	ADC 模块 1 AIN11 管脚存在 置位表示 ADC 模块 1 的输入管脚 11 存在。
26	ADC1AIN10	RO	0x1	ADC 模块 1 AIN10 管脚存在 置位表示 ADC 模块 1 的输入管脚 10 存在。
25	ADC1AIN9	RO	0x1	ADC 模块 1 AIN9 管脚存在 置位表示 ADC 模块 1 的输入管脚 9 存在。
24	ADC1AIN8	RO	0x1	ADC 模块 1 AIN8 管脚存在 置位表示 ADC 模块 1 的输入管脚 8 存在。
23	ADC1AIN7	RO	0x1	ADC 模块 1 AIN7 管脚存在 置位表示 ADC 模块 1 的输入管脚 7 存在。
22	ADC1AIN6	RO	0x1	ADC 模块 1 AIN6 管脚存在 置位表示 ADC 模块 1 的输入管脚 6 存在。



位/域	名称	类型	复位	描述
21	ADC1AIN5	RO	0x1	ADC 模块 1 AIN5 管脚存在 置位表示 ADC 模块 1 的输入管脚 5 存在。
20	ADC1AIN4	RO	0x1	ADC 模块 1 AIN4 管脚存在 置位表示 ADC 模块 1 的输入管脚 4 存在。
19	ADC1AIN3	RO	0x1	ADC 模块 1 AIN3 管脚存在 置位表示 ADC 模块 1 的输入管脚 3 存在。
18	ADC1AIN2	RO	0x1	ADC 模块 1 AIN2 管脚存在 置位表示 ADC 模块 1 的输入管脚 2 存在。
17	ADC1AIN1	RO	0x1	ADC 模块 1 AIN1 管脚存在 置位表示 ADC 模块 1 的输入管脚 1 存在。
16	ADC1AIN0	RO	0x1	ADC 模块 1 AIN0 管脚存在 置位表示 ADC 模块 1 的输入管脚 0 存在。
15	ADC0AIN15	RO	0x1	ADC 模块 0 AIN15 管脚存在 置位表示 ADC 模块 0 的输入管脚 15 存在。
14	ADC0AIN14	RO	0x1	ADC 模块 0 AIN14 管脚存在 置位表示 ADC 模块 0 的输入管脚 14 存在。
13	ADC0AIN13	RO	0x1	ADC 模块 0 AIN13 管脚存在 置位表示 ADC 模块 0 的输入管脚 13 存在。
12	ADC0AIN12	RO	0x1	ADC 模块 0 AIN12 管脚存在 置位表示 ADC 模块 0 的输入管脚 12 存在。
11	ADC0AIN11	RO	0x1	ADC 模块 0 AIN11 管脚存在 置位表示 ADC 模块 0 的输入管脚 11 存在。
10	ADC0AIN10	RO	0x1	ADC 模块 0 AIN10 管脚存在 置位表示 ADC 模块 0 的输入管脚 10 存在。
9	ADC0AIN9	RO	0x1	ADC 模块 0 AIN9 管脚存在 置位表示 ADC 模块 0 的输入管脚 9 存在。
8	ADC0AIN8	RO	0x1	ADC 模块 0 AIN8 管脚存在 置位表示 ADC 模块 0 的输入管脚 8 存在。
7	ADC0AIN7	RO	0x1	ADC 模块 0 AIN7 管脚存在 置位表示 ADC 模块 0 的输入管脚 7 存在。
6	ADC0AIN6	RO	0x1	ADC 模块 0 AIN6 管脚存在 置位表示 ADC 模块 0 的输入管脚 6 存在。
5	ADC0AIN5	RO	0x1	ADC 模块 0 AIN5 管脚存在 置位表示 ADC 模块 0 的输入管脚 5 存在。
4	ADC0AIN4	RO	0x1	ADC 模块 0 AIN4 管脚存在 置位表示 ADC 模块 0 的输入管脚 4 存在。

---

位/域	名称	类型	复位	描述
3	ADC0AIN3	RO	0x1	ADC 模块 0 AIN3 管脚存在 置位表示 ADC 模块 0 的输入管脚 3 存在。
2	ADC0AIN2	RO	0x1	ADC 模块 0 AIN2 管脚存在 置位表示 ADC 模块 0 的输入管脚 2 存在。
1	ADC0AIN1	RO	0x1	ADC 模块 0 AIN1 管脚存在 置位表示 ADC 模块 0 的输入管脚 1 存在。
0	ADC0AIN0	RO	0x1	ADC 模块 0 AIN0 管脚存在 置位表示 ADC 模块 0 的输入管脚 0 存在。

## 寄存器 134: 软件复位控制寄存器 0 (SRCR0), 偏移量 0x040

该寄存器允许模块被单独复位。写入该寄存器的值被器件功能 1 (DC1) 寄存器的位屏蔽。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用软件复位寄存器 (如 SRWD) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该传统寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设 (如看门狗 1) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 软件复位控制寄存器 0 (SRCR0)

基址 0x400F.E000  
偏移量 0x040  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			WDT1	保留		CAN1	CAN0	保留			PWM0	保留		ADC1	ADC0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				保留			保留			HIB	保留		WDT0	保留	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
28	WDT1	RO	0x0	WDT1 复位控制 当该位置位时, 看门狗定时器模块 1 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
27:26	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
25	CAN1	RO	0x0	CAN1 复位控制 当该位置位时, CAN 模块 1 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
24	CAN0	RO	0x0	CAN0 复位控制 当该位置位时, CAN 模块 0 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
23:21	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
20	PWM0	RO	0x0	PWM 复位控制 当该位置位时, PWM 模块 0 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
19:18	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
17	ADC1	RO	0x0	ADC1复位控制 当该位置位时，ADC模块1复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
16	ADC0	RO	0x0	ADC0复位控制 当该位置位时，ADC模块0复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
15:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
6	HIB	RO	0x0	HIB 复位控制 当该位置位时，休眠模块复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
5:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
3	WDT0	RO	0x0	WDT0 复位控制 当该位置位时，看门狗定时器模块 0 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
2:0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

## 寄存器 135: 软件复位控制寄存器 1 (SRCR1), 偏移量 0x044

该寄存器允许模块被单独复位。写入该寄存器的值被器件功能 2 (DC2) 寄存器的位屏蔽。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用软件复位寄存器 (如 SRTIMER) 用于复位特定的外设。写该寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设 (如 TIMER0) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

请注意软件复位模拟比较器 (SRACMP) 寄存器仅有一个位来设置模拟比较器模块。复位该模块将会复位所有块。如果任何 COMPn 位置位, 则整个模拟比较器复位。不能单独复位各模块。

### 软件复位控制寄存器 1 (SRCR1)

基址 0x400FE000

偏移量 0x044

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留					COMP2	COMP1	COMP0	保留				TIMER3	TIMER2	TIMER1	TIMER0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	I2C1	保留	I2C0	保留		QE1	QE10	保留		SSI1	SSI0	保留	UART2	UART1	UART0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:27	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
26	COMP2	RO	0x0	模拟比较器 2 复位控制 当该位置位时, 模拟比较器模块 2 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
25	COMP1	RO	0x0	模拟比较器 1 复位控制 当该位置位时, 模拟比较器模块 1 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
24	COMP0	RO	0x0	模拟比较器 0 复位控制 当该位置位时, 模拟比较器模块 0 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
23:20	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
19	TIMER3	RO	0x0	定时器 3 复位控制 定时器 3 复位控制。当该位置位时, 通用定时器模块 3 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。

位/域	名称	类型	复位	描述
18	TIMER2	RO	0x0	定时器 2 复位控制 当该位置位时，通用定时器模块 2 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
17	TIMER1	RO	0x0	定时器 1 复位控制 当该位置位时，通用定时器模块 1 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
16	TIMER0	RO	0x0	定时器 0 复位控制 当该位置位时，通用定时器模块 0 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
15	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
14	I2C1	RO	0x0	I2C1 复位控制 当该位置位时，I2C 模块 1 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
13	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
12	I2C0	RO	0x0	I2C0 复位控制 当该位置位时，I2C 模块 0 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
11:10	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
9	QE11	RO	0x0	QE11 复位控制 当该位置位时，QE1 模块 1 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
8	QE10	RO	0x0	QE10 复位控制 当该位置位时，QE1 模块 0 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
7:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
5	SSI1	RO	0x0	SSI1 复位控制 当该位置位时，SSI 模块 1 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
4	SSI0	RO	0x0	SSI0 复位控制 当该位置位时，SSI 模块 0 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
2	UART2	RO	0x0	UART2 复位控制 当该位置位时，UART 模块 2 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。

位/域	名称	类型	复位	描述
1	UART1	RO	0x0	UART1 复位控制 当该位置位时，UART 模块 1 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
0	UART0	RO	0x0	UART0 复位控制 当该位置位时，UART 模块 0 复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。

## 寄存器 136: 软件复位控制寄存器 2 ( SRCR2 ) , 偏移量 0x048

该寄存器允许模块被单独复位。写入该寄存器的值被器件功能 4 (DC4) 寄存器的位屏蔽。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用软件复位寄存器 ( 如 SRDMA ) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设 ( 如  $\mu$ DMA ) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 软件复位控制寄存器 2 (SRCR2)

基址 0x400F.E000  
偏移量 0x048  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															USB0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		UDMA	保留			GPIOJ		GPIOH	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:17	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
16	USB0	RO	0x0	USB0 复位控制 当该位置位时, USB 模块 0 复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
15:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
13	UDMA	RO	0x0	微 DMA 复位控制 当该位置位时, $\mu$ DMA 模块复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
12:9	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
8	GPIOJ	RO	0x0	端口 J 复位控制 当该位置位时, 端口 J 模块复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。
7	GPIOH	RO	0x0	端口 H 复位控制 当该位置位时, 端口 H 模块复位。所有内部数据丢失, 同时寄存器回到复位状态。该位在置位后必须手动清零。



位/域	名称	类型	复位	描述
6	GPIOG	RO	0x0	端口 G 复位控制 当该位置位时，端口 G 模块复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
5	GPIOF	RO	0x0	端口 F 复位控制 当该位置位时，端口 F 模块复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
4	GPIOE	RO	0x0	端口 E 复位控制 当该位置位时，端口 E 模块复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
3	GPIOD	RO	0x0	端口 D 复位控制 当该位置位时，端口 D 模块复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
2	GPIOC	RO	0x0	端口 C 复位控制 当该位置位时，端口 C 模块复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
1	GPIOB	RO	0x0	端口 B 复位控制 当该位置位时，端口 B 模块复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。
0	GPIOA	RO	0x0	端口 A 复位控制 当该位置位时，端口 A 模块复位。所有内部数据丢失，同时寄存器回到复位状态。该位在置位后必须手动清零。

### 寄存器 137: 运行模式时钟门控控制寄存器 0 ( RCGC0 ) , 偏移量 0x100

该寄存器控制正常运行模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则, 该模块不使用时钟并且被禁用( 节能)。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。除非特别说明, 否则这些位的复位状态为 0 ( 不使用时钟), 以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意, 这些寄存器除了含有对接口、功能或模块进行控制的位以外, 还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC0 是运行操作的时钟配置寄存器, SCGC0 是睡眠操作的时钟配置寄存器, DCGC0 是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置(RCC)寄存器的 ACG 位说明系统使用睡眠模式。请注意, 在访问该模块中任何寄存器前, 启用模块时钟后必须有 3 个系统时钟的延迟。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用运行模式时钟门控控制寄存器( 如 RCGCWD ) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设( 如看门狗 1 ) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

同样, 应将 ADC 外设属性(ADCPC) 寄存器用于配置 ADC 采样率。但是, 要支持传统软件, 可使用 MAXADCnSPD 域。写这些传统域的同时, 也将写外设专用寄存器中的相应域。如果通过对该寄存器进行写操作更改一个域, 则可以通过对该寄存器进行读操作对它进行正确回读。软件必须使用外设专用寄存器以支持该寄存器中不可用的比率。如果软件使用外设专用寄存器设置 ADC 比率, 则写操作会产生正确操作, 但是该域的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 运行模式时钟门控控制寄存器 0 (RCGC0)

基址 0x400F.E000  
偏移量 0x100  
类型 RO, 复位 0x0000.0040

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			WDT1	保留		CAN1	CAN0	保留			PWM0	保留		ADC1	ADC0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				MAXADC1SPD		MAXADC0SPD		保留	HIB	保留		WDT0	保留		
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
28	WDT1	RO	0x0	WDT1 时钟门控控制 该位控制看门狗定时器模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。

位/域	名称	类型	复位	描述										
27:26	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。										
25	CAN1	RO	0x0	CAN1 时钟门控控制 这个位控制 CAN 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。										
24	CAN0	RO	0x0	CAN0 时钟门控控制 这个位控制 CAN 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。										
23:21	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。										
20	PWM0	RO	0x0	PWM 时钟门控控制 该位控制 PWM 模块的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。										
19:18	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。										
17	ADC1	RO	0x0	ADC1 时钟门控控制 这个位控制 SAR ADC 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。										
16	ADC0	RO	0x0	ADC0 时钟门控控制 这个位控制 ADC 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。										
15:12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。										
11:10	MAXADC1SPD	RO	0x0	ADC1 采样速率 该域设置 ADC 模块 1 的数据采样速率。不能把速率设置得超过最大速率。采样速率可通过如下对 MAXADC1SPD 位的设置而设定（其它所有编码保留）：  <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x3</td> <td>1M 采样/秒</td> </tr> <tr> <td>0x2</td> <td>500K 采样/秒</td> </tr> <tr> <td>0x1</td> <td>250K 采样/秒</td> </tr> <tr> <td>0x0</td> <td>125K 采样/秒</td> </tr> </tbody> </table>	值	描述	0x3	1M 采样/秒	0x2	500K 采样/秒	0x1	250K 采样/秒	0x0	125K 采样/秒
值	描述													
0x3	1M 采样/秒													
0x2	500K 采样/秒													
0x1	250K 采样/秒													
0x0	125K 采样/秒													

位/域	名称	类型	复位	描述
9:8	MAXADC0SPD	RO	0x0	<p>ADC0 采样速率</p> <p>该域设置 ADC0 模块的数据采样速率。不能把速率设置得超过最大速率。采样速率可通过如下对 MAXADC0SPD 位的设置而设定（其它所有编码保留）：</p> <p>值 描述</p> <p>0x3 1M 采样/秒</p> <p>0x2 500K 采样/秒</p> <p>0x1 250K 采样/秒</p> <p>0x0 125K 采样/秒</p>
7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
6	HIB	RO	0x1	<p>HIB 时钟门控控制</p> <p>该位控制休眠模块的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
5:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
3	WDT0	RO	0x0	<p>WDT0 时钟门控控制</p> <p>该位控制看门狗定时器模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
2:0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

## 寄存器 138: 运行模式时钟门控控制寄存器 1 ( RCGC1 ) , 偏移量 0x104

该寄存器控制正常运行模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则, 该模块不使用时钟并且被禁用( 节能)。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。除非特别说明, 否则这些位的复位状态为 0 ( 不使用时钟), 以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意, 这些寄存器除了含有对接口、功能或模块进行控制的位以外, 还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC1 是运行操作的时钟配置寄存器, SCGC1 是睡眠操作的时钟配置寄存器, DCGC1 是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置(RCC)寄存器的 ACG 位说明系统使用睡眠模式。请注意, 在访问该模块中任何寄存器前, 启用模块时钟后必须有 3 个系统时钟的延迟。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用运行模式时钟门控控制寄存器( 如 RCGCTIMER ) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设( 如 Timer 0 ) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 运行模式时钟门控控制寄存器 1 (RCGC1)

基址 0x400F.E000  
偏移量 0x104  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留					COMP2	COMP1	COMP0	保留			TIMER3	TIMER2	TIMER1	TIMER0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	I2C1	保留	I2C0	保留		QE11	QE10	保留		SSI1	SSI0	保留	UART2	UART1	UART0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:27	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
26	COMP2	RO	0x0	模拟比较器 2 时钟门控 这个位控制模拟比较器 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
25	COMP1	RO	0x0	模拟比较器 1 时钟门控 这个位控制模拟比较器 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
24	COMP0	RO	0x0	模拟比较器 0 时钟门控 这个位控制模拟比较器 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。

位/域	名称	类型	复位	描述
23:20	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
19	TIMER3	RO	0x0	定时器 3 时钟门控控制 这个位控制通用定时器模块 3 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
18	TIMER2	RO	0x0	定时器 2 时钟门控控制 这个位控制通用定时器模块 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
17	TIMER1	RO	0x0	定时器 1 时钟门控控制 这个位控制通用定时器模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
16	TIMER0	RO	0x0	定时器 0 时钟门控控制 这个位控制通用定时器模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
15	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
14	I2C1	RO	0x0	I2C1 时钟门控控制 这个位控制 I2C 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
13	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
12	I2C0	RO	0x0	I2C0 时钟门控控制 这个位控制 I2C 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
11:10	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
9	QEI1	RO	0x0	QEI1 时钟门控控制 这个位控制 QEI 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
8	QEI0	RO	0x0	QEI0 时钟门控控制 这个位控制 QEI 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
7:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
5	SSI1	RO	0x0	SSI1 时钟门控控制 这个位控制 SSI 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
4	SSI0	RO	0x0	SSI0 时钟门控控制 这个位控制 SSI 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
2	UART2	RO	0x0	UART2 时钟门控控制 这个位控制 UART 模块 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
1	UART1	RO	0x0	UART1 时钟门控控制 这个位控制 UART 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
0	UART0	RO	0x0	UART0 时钟门控控制 这个位控制 UART 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。

### 寄存器 139: 运行模式时钟门控控制寄存器 2 ( RCGC2 ) , 偏移量 0x108

该寄存器控制正常运行模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则, 该模块不使用时钟并且被禁用( 节能)。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。除非特别说明, 否则这些位的复位状态为 0 ( 不使用时钟), 以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意, 这些寄存器除了含有对接口、功能或模块进行控制的位以外, 还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC2 是运行操作的时钟配置寄存器, SCGC2 是睡眠操作的时钟配置寄存器, DCGC2 是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置(RCC)寄存器的 ACG 位说明系统使用睡眠模式。请注意, 在访问该模块中任何寄存器前, 启用模块时钟后必须有 3 个系统时钟的延迟。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用运行模式时钟门控控制寄存器( 如 RCGCDMA ) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设( 如  $\mu$ DMA ) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 运行模式时钟门控控制寄存器 2 (RCGC2)

基址 0x400F.E000  
偏移量 0x108  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															USB0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		UDMA	保留			GPIOJ		GPIOH	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:17	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
16	USB0	RO	0x0	USB0 时钟门控控制 这个位控制 USB 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
15:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
13	UDMA	RO	0x0	微 DMA 时钟门控控制 该位控制微 DMA 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
12:9	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。



位/域	名称	类型	复位	描述
8	GPIOJ	RO	0x0	<p>端口 J 时钟门控控制</p> <p>该位控制端口 J 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
7	GPIOH	RO	0x0	<p>端口 H 时钟门控控制</p> <p>该位控制端口 H 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
6	GPIOG	RO	0x0	<p>端口 G 时钟门控控制</p> <p>该位控制端口 G 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
5	GPIOF	RO	0x0	<p>端口 F 时钟门控控制</p> <p>该位控制端口 F 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
4	GPIOE	RO	0x0	<p>端口 E 时钟门控控制</p> <p>端口 E 时钟门控控制。该位控制端口 E 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
3	GIPOD	RO	0x0	<p>端口 D 时钟门控控制</p> <p>端口 D 时钟门控控制。该位控制端口 D 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
2	GPIOC	RO	0x0	<p>端口 C 时钟门控控制</p> <p>该位控制端口 C 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
1	GPIOB	RO	0x0	<p>端口 B 时钟门控控制</p> <p>该位控制端口 B 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
0	GPIOA	RO	0x0	<p>端口 A 时钟门控控制</p> <p>该位控制端口 A 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>

### 寄存器 140: 睡眠模式时钟门控控制寄存器 0 ( SCGC0 ) , 偏移量 0x110

该寄存器控制睡眠模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则, 该模块不使用时钟并且被禁用 ( 节能 )。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。除非特别说明, 否则这些位的复位状态为 0 ( 不使用时钟 ), 以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意, 这些寄存器除了含有对接口、功能或模块进行控制的位以外, 还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC0是运行操作的时钟配置寄存器, SCGC0是睡眠操作的时钟配置寄存器, DCGC0是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置 (RCC) 寄存器的 ACG 位说明系统使用睡眠模式。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用睡眠模式时钟门控控制寄存器 ( 如 SCGCWD ) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设 ( 如看门狗 1 ) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 睡眠模式时钟门控控制寄存器 0 (SCGC0)

基址 0x400F.E000  
偏移量 0x110  
类型 RO, 复位 0x0000.0040

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			WDT1	保留		CAN1	CAN0	保留			PWM0	保留		ADC1	ADC0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										HIB	保留		WDT0	保留	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
28	WDT1	RO	0x0	WDT1 时钟门控控制 该位控制看门狗定时器模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
27:26	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
25	CAN1	RO	0x0	CAN1 时钟门控控制 这个位控制 CAN 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。

位/域	名称	类型	复位	描述
24	CAN0	RO	0x0	CAN0 时钟门控控制 这个位控制 CAN 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
23:21	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
20	PWM0	RO	0x0	PWM 时钟门控控制 该位控制 PWM 模块的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
19:18	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
17	ADC1	RO	0x0	ADC1 时钟门控控制 这个位控制 ADC 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
16	ADC0	RO	0x0	ADC0 时钟门控控制 这个位控制 ADC 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
15:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
6	HIB	RO	0x1	HIB 时钟门控控制 该位控制休眠模块的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
5:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
3	WDT0	RO	0x0	WDT0 时钟门控控制 该位控制看门狗定时器模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
2:0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

### 寄存器 141: 睡眠模式时钟门控控制寄存器 1 ( SCGC1 ) , 偏移量 0x114

该寄存器控制睡眠模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则, 该模块不使用时钟并且被禁用 ( 节能 )。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。除非特别说明, 否则这些位的复位状态为 0 ( 不使用时钟 ), 以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意, 这些寄存器除了含有对接口、功能或模块进行控制的位以外, 还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC1是运行操作的时钟配置寄存器, SCGC1是睡眠操作的时钟配置寄存器, DCGC1 是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置 (RCC) 寄存器的 ACG 位说明系统使用睡眠模式。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用睡眠模式时钟门控控制寄存器 ( 如 SCGCTIMER ) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设 ( 如 Timer 0 ) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 睡眠模式时钟门控控制寄存器 1 (SCGC1)

基址 0x400F.E000  
偏移量 0x114  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留				COMP2	COMP1	COMP0	保留				TIMER3	TIMER2	TIMER1	TIMER0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	I2C1	保留	I2C0	保留		QE11	QE10	保留		SS11	SS10	保留	UART2	UART1	UART0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:27	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
26	COMP2	RO	0x0	模拟比较器 2 时钟门控 这个位控制模拟比较器 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
25	COMP1	RO	0x0	模拟比较器 1 时钟门控 这个位控制模拟比较器 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
24	COMP0	RO	0x0	模拟比较器 0 时钟门控 这个位控制模拟比较器 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。

位/域	名称	类型	复位	描述
23:20	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
19	TIMER3	RO	0x0	定时器 3 时钟门控控制 这个位控制通用定时器模块 3 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
18	TIMER2	RO	0x0	定时器 2 时钟门控控制 这个位控制通用定时器模块 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
17	TIMER1	RO	0x0	定时器 1 时钟门控控制 这个位控制通用定时器模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
16	TIMER0	RO	0x0	定时器 0 时钟门控控制 这个位控制通用定时器模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
15	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
14	I2C1	RO	0x0	I2C1 时钟门控控制 这个位控制 I2C 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
13	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
12	I2C0	RO	0x0	I2C0 时钟门控控制 这个位控制 I2C 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
11:10	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
9	QEI1	RO	0x0	QEI1 时钟门控控制 这个位控制 QEI 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
8	QEI0	RO	0x0	QEI0 时钟门控控制 这个位控制 QEI 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
7:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
5	SSI1	RO	0x0	SSI1 时钟门控控制 这个位控制 SSI 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
4	SSI0	RO	0x0	SSI0 时钟门控控制 这个位控制 SSI 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
2	UART2	RO	0x0	UART2 时钟门控控制 这个位控制 UART 模块 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
1	UART1	RO	0x0	UART1 时钟门控控制 这个位控制 UART 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
0	UART0	RO	0x0	UART0 时钟门控控制 这个位控制 UART 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。

## 寄存器 142: 睡眠模式时钟门控控制寄存器 2 ( SCGC2 ) , 偏移量 0x118

该寄存器控制睡眠模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则，该模块不使用时钟并且被禁用（节能）。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。除非特别说明，否则这些位的复位状态为 0（不使用时钟），以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意，这些寄存器除了含有对接口、功能或模块进行控制的位以外，还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC2 是运行操作的时钟配置寄存器，SCGC2 是睡眠操作的时钟配置寄存器，DCGC2 是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置 (RCC) 寄存器的 ACG 位说明系统使用睡眠模式。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用睡眠模式时钟门控控制寄存器（如 SCGCDMA）用于复位特定的外设。写该传统寄存器的同时，也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设（如  $\mu$ DMA）执行写操作，则写操作会产生正确操作，但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问，则必须通过读-修改-写的操作来访问外设专用寄存器，因为该操作仅影响不在传统寄存器中的外设。通过这种方法，外设专用和传统寄存器都具有一致的信息。

### 睡眠模式时钟门控控制寄存器 2 (SCGC2)

基址 0x400F.E000  
偏移量 0x118  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															USB0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		UDMA	保留			GPIOJ		GPIOH	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:17	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
16	USB0	RO	0x0	USB0 时钟门控控制 这个位控制 USB 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
15:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
13	UDMA	RO	0x0	微 DMA 时钟门控控制 该位控制微 DMA 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
12:9	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
8	GPIOJ	RO	0x0	<p>端口 J 时钟门控控制</p> <p>该位控制端口 J 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
7	GPIOH	RO	0x0	<p>端口 H 时钟门控控制</p> <p>该位控制端口 H 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
6	GPIOG	RO	0x0	<p>端口 G 时钟门控控制</p> <p>该位控制端口 G 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
5	GPIOF	RO	0x0	<p>端口 F 时钟门控控制</p> <p>该位控制端口 F 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
4	GPIOE	RO	0x0	<p>端口 E 时钟门控控制</p> <p>端口 E 时钟门控控制。该位控制端口 E 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
3	GIPOD	RO	0x0	<p>端口 D 时钟门控控制</p> <p>端口 D 时钟门控控制。该位控制端口 D 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
2	GPIOC	RO	0x0	<p>端口 C 时钟门控控制</p> <p>该位控制端口 C 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
1	GPIOB	RO	0x0	<p>端口 B 时钟门控控制</p> <p>该位控制端口 B 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
0	GPIOA	RO	0x0	<p>端口 A 时钟门控控制</p> <p>该位控制端口 A 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>



## 寄存器 143: 深度睡眠模式时钟门控控制寄存器 0 (DCGC0), 偏移量 0x120

该寄存器控制深度睡眠模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则, 该模块不使用时钟并且被禁用(节能)。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。除非特别说明, 否则这些位的复位状态为 0 (不使用时钟), 以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意, 这些寄存器除了含有对接口、功能或模块进行控制的位以外, 还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC0 是运行操作的时钟配置寄存器, SCGC0 是睡眠操作的时钟配置寄存器, DCGC0 是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置(RCC)寄存器的 ACG 位说明系统使用睡眠模式。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用深度睡眠模式时钟门控控制寄存器(如 DCGCWD)用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设(如看门狗 1)执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 深度睡眠模式时钟门控控制寄存器 0 (DCGC0)

基址 0x400F.E000  
偏移量 0x120  
类型 RO, 复位 0x0000.0040

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			WDT1	保留		CAN1	CAN0	保留			PWM0	保留		ADC1	ADC0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										HIB	保留		WDT0	保留	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
28	WDT1	RO	0x0	WDT1 时钟门控控制 该位控制看门狗定时器模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
27:26	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
25	CAN1	RO	0x0	CAN1 时钟门控控制 这个位控制 CAN 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。

位/域	名称	类型	复位	描述
24	CAN0	RO	0x0	<p>CAN0 时钟门控控制</p> <p>这个位控制 CAN 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
23:21	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
20	PWM0	RO	0x0	<p>PWM 时钟门控控制</p> <p>该位控制 PWM 模块的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
19:18	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
17	ADC1	RO	0x0	<p>ADC1 时钟门控控制</p> <p>这个位控制 ADC 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
16	ADC0	RO	0x0	<p>ADC0 时钟门控控制</p> <p>这个位控制 ADC 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
15:7	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
6	HIB	RO	0x1	<p>HIB 时钟门控控制</p> <p>该位控制休眠模块的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
5:4	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
3	WDT0	RO	0x0	<p>WDT0 时钟门控控制</p> <p>该位控制看门狗定时器模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
2:0	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>

## 寄存器 144: 深度睡眠模式时钟门控控制寄存器 1 ( DCGC1 ), 偏移量 0x124

该寄存器控制深度睡眠模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则, 该模块不使用时钟并且被禁用( 节能)。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。除非特别说明, 否则这些位的复位状态为 0 ( 不使用时钟), 以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意, 这些寄存器除了含有对接口、功能或模块进行控制的位以外, 还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC1 是运行操作的时钟配置寄存器, SCGC1 是睡眠操作的时钟配置寄存器, DCGC1 是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置 (RCC) 寄存器的 ACG 位说明系统使用睡眠模式。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用深度睡眠模式时钟门控控制寄存器 ( 如 DCGCTIMER ) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设 ( 如 Timer 0 ) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

### 深度睡眠模式时钟门控控制寄存器 1 (DCGC1)

基址 0x400F.E000  
偏移量 0x124  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留				COMP2	COMP1	COMP0	保留				TIMER3	TIMER2	TIMER1	TIMER0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	I2C1	保留	I2C0	保留		QE11	QE10	保留		SSI1	SSI0	保留	UART2	UART1	UART0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:27	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
26	COMP2	RO	0x0	模拟比较器 2 时钟门控 这个位控制模拟比较器 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
25	COMP1	RO	0x0	模拟比较器 1 时钟门控 这个位控制模拟比较器 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
24	COMP0	RO	0x0	模拟比较器 0 时钟门控 这个位控制模拟比较器 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。

位/域	名称	类型	复位	描述
23:20	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
19	TIMER3	RO	0x0	定时器 3 时钟门控控制 这个位控制通用定时器模块 3 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
18	TIMER2	RO	0x0	定时器 2 时钟门控控制 这个位控制通用定时器模块 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
17	TIMER1	RO	0x0	定时器 1 时钟门控控制 这个位控制通用定时器模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
16	TIMER0	RO	0x0	定时器 0 时钟门控控制 这个位控制通用定时器模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
15	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
14	I2C1	RO	0x0	I2C1 时钟门控控制 这个位控制 I2C 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
13	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
12	I2C0	RO	0x0	I2C0 时钟门控控制 这个位控制 I2C 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
11:10	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
9	QEI1	RO	0x0	QEI1 时钟门控控制 这个位控制 QEI 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
8	QEI0	RO	0x0	QEI0 时钟门控控制 这个位控制 QEI 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
7:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
5	SSI1	RO	0x0	SSI1 时钟门控控制 这个位控制 SSI 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
4	SSI0	RO	0x0	SSI0 时钟门控控制 这个位控制 SSI 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
2	UART2	RO	0x0	UART2 时钟门控控制 这个位控制 UART 模块 2 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
1	UART1	RO	0x0	UART1 时钟门控控制 这个位控制 UART 模块 1 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。
0	UART0	RO	0x0	UART0 时钟门控控制 这个位控制 UART 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。

### 寄存器 145: 深度睡眠模式时钟门控控制寄存器 2 ( DCGC2 ) , 偏移量 0x128

该寄存器控制深度睡眠模式的时钟门控逻辑。每一位都控制一个给定的接口、功能或模块的时钟启用。如果置位表示该模块接收到时钟并运行。否则, 该模块不使用时钟并且被禁用( 节能)。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。除非特别说明, 否则这些位的复位状态为 0 ( 不使用时钟), 以便禁用所有功能模块。应用所需的端口需要通过软件来启用。请注意, 这些寄存器除了含有对接口、功能或模块进行控制的位以外, 还可能含有其它位。这样可保证与其它系列以及将来的器件实现合理的代码兼容。RCGC2 是运行操作的时钟配置寄存器, SCGC2 是睡眠操作的时钟配置寄存器, DCGC2 是深度睡眠操作的时钟配置寄存器。置位运行模式时钟配置 (RCC) 寄存器的 ACG 位说明系统使用睡眠模式。

**重要:** 该寄存器用于实现传统软件支持。

应将外设专用深度睡眠模式时钟门控控制寄存器 ( 如 DCGCDMA ) 用于复位特定的外设。写该传统寄存器的同时, 也将写外设专用寄存器中的相应位。通过对该寄存器的写操作更改的任何位都可以通过对该寄存器的读操作进行正确回读。软件必须使用外设专用寄存器以支持不处于传统寄存器中的模块。如果软件使用外设专用寄存器对传统外设 ( 如  $\mu$ DMA ) 执行写操作, 则写操作会产生正确操作, 但是该位的值不在该寄存器中得到反映。如果软件使用传统和外设专用寄存器访问, 则必须通过读-修改-写的操作来访问外设专用寄存器, 因为该操作仅影响不在传统寄存器中的外设。通过这种方法, 外设专用和传统寄存器都具有一致的信息。

#### 深度睡眠模式时钟门控控制寄存器 2 (DCGC2)

基址 0x400F.E000  
偏移量 0x128  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															USB0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		UDMA	保留			GPIOJ		GPIOH	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:17	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
16	USB0	RO	0x0	USB0 时钟门控控制 这个位控制 USB 模块 0 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
15:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
13	UDMA	RO	0x0	微 DMA 时钟门控控制 该位控制微 DMA 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟, 对它的读或写都会产生一个总线故障。
12:9	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
8	GPIOJ	RO	0x0	<p>端口 J 时钟门控控制</p> <p>该位控制端口 J 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
7	GPIOH	RO	0x0	<p>端口 H 时钟门控控制</p> <p>该位控制端口 H 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
6	GPIOG	RO	0x0	<p>端口 G 时钟门控控制</p> <p>该位控制端口 G 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
5	GPIOF	RO	0x0	<p>端口 F 时钟门控控制</p> <p>该位控制端口 F 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
4	GPIOE	RO	0x0	<p>端口 E 时钟门控控制</p> <p>端口 E 时钟门控控制。该位控制端口 E 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
3	GIPOD	RO	0x0	<p>端口 D 时钟门控控制</p> <p>端口 D 时钟门控控制。该位控制端口 D 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
2	GPIOC	RO	0x0	<p>端口 C 时钟门控控制</p> <p>该位控制端口 C 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
1	GPIOB	RO	0x0	<p>端口 B 时钟门控控制</p> <p>该位控制端口 B 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>
0	GPIOA	RO	0x0	<p>端口 A 时钟门控控制</p> <p>该位控制端口 A 的时钟门控。如果置位表示该模块接收到时钟并运行。否则该模块不使用时钟并被禁用。如果该模块不使用时钟，对它的读或写都会产生一个总线故障。</p>

## 寄存器 146: 器件功能寄存器 9 ( DC9 ) , 偏移量 0x190

该寄存器由器件预定义且可用于校验 ADC 数字比较器特性。

**重要:** 该寄存器用于实现传统软件支持。

应使用 ADC 外设属性 (ADCPP) 寄存器确定在 ADC 模块上可用的数字比较器数量。读取该寄存器即可正确识别传统比较器是否存在。软件必须使用 ADCPP 寄存器来确定不受 DCn 寄存器支持的比较器是否存在。

### 器件功能寄存器 9 (DC9)

基址 0x400F.E000  
偏移量 0x190  
类型 RO, 复位 0x00FF.00FF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								ADC1DC7	ADC1DC6	ADC1DC5	ADC1DC4	ADC1DC3	ADC1DC2	ADC1DC1	ADC1DC0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								ADC0DC7	ADC0DC6	ADC0DC5	ADC0DC4	ADC0DC3	ADC0DC2	ADC0DC1	ADC0DC0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:24	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
23	ADC1DC7	RO	0x1	ADC1 DC7 存在 置位表示 ADC 模块 1 的数字比较器 7 存在。
22	ADC1DC6	RO	0x1	ADC1 DC6 存在 置位表示 ADC 模块 1 的数字比较器 6 存在。
21	ADC1DC5	RO	0x1	ADC1 DC5 存在 置位表示 ADC 模块 1 的数字比较器 5 存在。
20	ADC1DC4	RO	0x1	ADC1 DC4 存在 置位表示 ADC 模块 1 的数字比较器 4 存在。
19	ADC1DC3	RO	0x1	ADC1 DC3 存在 置位表示 ADC 模块 1 的数字比较器 3 存在。
18	ADC1DC2	RO	0x1	ADC1 DC2 存在 置位表示 ADC 模块 1 的数字比较器 2 存在。
17	ADC1DC1	RO	0x1	ADC1 DC1 存在 置位表示 ADC 模块 1 的数字比较器 1 存在。
16	ADC1DC0	RO	0x1	ADC1 DC0 存在 置位表示 ADC 模块 1 的数字比较器 0 存在。
15:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。



位/域	名称	类型	复位	描述
7	ADC0DC7	RO	0x1	ADC0 DC7 存在 置位表示 ADC 模块 0 的数字比较器 7 存在。
6	ADC0DC6	RO	0x1	ADC0 DC6 存在 置位表示 ADC 模块 0 的数字比较器 6 存在。
5	ADC0DC5	RO	0x1	ADC0 DC5 存在 置位表示 ADC 模块 0 的数字比较器 5 存在。
4	ADC0DC4	RO	0x1	ADC0 DC4 存在 置位表示 ADC 模块 0 的数字比较器 4 存在。
3	ADC0DC3	RO	0x1	ADC0 DC3 存在 置位表示 ADC 模块 0 的数字比较器 3 存在。
2	ADC0DC2	RO	0x1	ADC0 DC2 存在 置位表示 ADC 模块 0 的数字比较器 2 存在。
1	ADC0DC1	RO	0x1	ADC0 DC1 存在 置位表示 ADC 模块 0 的数字比较器 1 存在。
0	ADC0DC0	RO	0x1	ADC0 DC0 存在 置位表示 ADC 模块 0 的数字比较器 0 存在。

## 寄存器 147: 非易失性存储器信息寄存器 ( NVMSTAT ) , 偏移量 0x1A0

该寄存器由器件预定义且可用于校验特性。

**重要:** 该寄存器用于实现传统软件支持。

应使用 ROM 第三方软件 (ROMSWMAP) 寄存器确定该微控制器的片上 ROM 中是否存在第三方软件。读取寄存器中的 TPSW 位可以正确地识别是否存在传统第三方软件。软件应使用 ROMSWMAP 寄存器来识别不在传统器件上的软件。

### 非易失性存储器信息寄存器 (NVMSTAT)

基址 0x400F.E000  
偏移量 0x1A0  
类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
	FWB															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	FWB	RO	0x1	32 个字的 Flash 写缓存器可用 置位表示 32 个字的 Flash 存储器写缓存器特性可用。

## 6 系统异常模块

该模块是处理系统级 Cortex-M4 FPU 异常情况的 AHB 外设。对于寄存器映射到此存储器槽的功能，如果器件不提供该功能，则写入相关寄存器的所有值都将被忽略，读取值返回零。

### 6.1 功能描述

系统异常模块提供系统级中断的控制和状态。由于在发送到中断控制器前，所有中断事件会进行一次逻辑或操作，因此在任何给定的时间，系统异常模块只能向控制器发送一个中断请求。通过读取系统异常屏蔽的中断状态 (SYSEXCIMIS) 寄存器，软件可以在一个中断服务例程中处理多个中断事件。对系统异常中断屏蔽 (SYSEXCIM) 寄存器中相应的中断屏蔽位进行置位，可以定义能够触发控制器级中断的中断事件。如果不使用中断，通过系统异常原始中断状态 (SYSEXCRIS) 寄存器可随时查看原始中断状态。向系统异常中断清零 (SYSEXCIC) 寄存器的相应位写 1，即可 (为 SYSEXCIMIS 寄存器和 SYSEXCRIS 寄存器) 清除中断状态。

### 6.2 寄存器映射

表 6-1 在 467 页 列出了系统异常模块寄存器。表中列出的偏移量是寄存器地址相对于 0x400F.9000。

注意：未使用的系统异常寄存器空间保留用于未来使用或内部使用。软件不得修改任何保留的存储器地址。

表 6-1. 系统异常 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	SYSEXCRIS	RO	0x0000.0000	系统异常原始中断状态	468
0x004	SYSEXCIM	R/W	0x0000.0000	系统异常中断屏蔽	470
0x008	SYSEXCIMIS	RO	0x0000.0000	系统异常屏蔽的中断状态	472
0x00C	SYSEXCIC	W1C	0x0000.0000	系统异常中断清零	474

### 6.3 寄存器描述

所有给出的地址都是相对于 0x400F.9000 系统异常基址而言的。

## 寄存器 1: 系统异常原始中断状态 ( SYSEXCRIS ) , 偏移量 0x000

SYSEXCRIS 寄存器是原始中断状态寄存器。读取时, 该寄存器给出相应中断的当前原始状态值。写操作对本寄存器无效。

### 系统异常原始中断状态 (SYSEXCRIS)

基址 0x400F.9000

偏移量 0x000

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											FPIXCRIS	FPOFCRIS	FPUFCRIS	FPIOCRIS	FPDZCRIS	FPIDCRIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5	FPIXCRIS	RO	0	浮点不精确异常原始中断状态  值 描述 0 无中断 1 发生了浮点不精确异常。  通过在 SYSEXCIC 寄存器的 IXCIC 位写 1 可以清除这个位。
4	FPOFCRIS	RO	0	浮点上溢异常原始中断状态  值 描述 0 无中断 1 发生了浮点上溢异常。  通过在 SYSEXCIC 寄存器的 OFCIC 位写 1 可以清除这个位。
3	FPUFCRIS	RO	0	浮点下溢异常原始中断状态  值 描述 0 无中断 1 发生了浮点下溢异常。  通过在 SYSEXCIC 寄存器的 UFCIC 位写 1 可以清除这个位。
2	FPIOCRIS	RO	0	浮点无效操作原始中断状态  值 描述 0 无中断 1 发生了浮点无效操作异常。  通过在 SYSEXCIC 寄存器的 IOCIC 位写 1 可以清除这个位。

位/域	名称	类型	复位	描述
1	FPDZCRIS	RO	0	浮点除 0 异常原始中断状态  值 描述 0 无中断 1 发生了浮点除 0 异常。  通过在 SYSEXCIC 寄存器的 DZCIC 位写 1 可以清除这个位。
0	FPIDCRIS	RO	0	浮点输入反常异常原始中断状态  值 描述 0 无中断 1 发生了浮点输入反常异常。  通过在 SYSEXCIC 寄存器的 IDCIC 位写 1 可以清除这个位。

## 寄存器 2: 系统异常中断屏蔽 ( SYSEXCIM ) , 偏移量 0x004

SYSEXCIM 寄存器是中断屏蔽位/清零寄存器。

读取时, 该寄存器会给出相关中断屏蔽位的当前值。对某个位进行置位时, 可将相应的原始中断信号传递到中断控制器。将某个位清零时, 可阻止该原始中断信号传递到中断控制器。

### 系统异常中断屏蔽 (SYSEXCIM)

基址 0x400F.9000  
偏移量 0x004  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											FPIXCIM	FPOFCIM	FPUFCIM	FPIOCIM	FPDZCIM	FPIDCIM
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	R/W	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5	FPIXCIM	R/W	0	浮点不精确异常中断屏蔽  值 描述 0 FPIXCRIS 中断被抑制, 不发送到中断控制器。 1 当 SYSEXCIM 寄存器中的 FPIXCRIS 位置位时, 向中断控制器发送中断。
4	FPOFCIM	R/W	0	浮点上溢异常中断屏蔽  值 描述 0 FPOFCIS 中断被抑制, 不发送到中断控制器。 1 当 SYSEXCIM 寄存器中的 FPOFCRIS 位置位时, 向中断控制器发送中断。
3	FPUFCIM	R/W	0	浮点下溢异常中断屏蔽  值 描述 0 FPUFCRIS 中断被抑制, 不发送到中断控制器。 1 当 SYSEXCIM 寄存器中的 FPUFCRIS 位置位时, 向中断控制器发送中断。
2	FPIOCIM	R/W	0	浮点无效操作中断屏蔽  值 描述 0 FPIOCRIS 中断被抑制, 不发送到中断控制器。 1 当 SYSEXCIM 寄存器中的 FPIOCRIS 位置位时, 向中断控制器发送中断。

位/域	名称	类型	复位	描述
1	FPDZCIM	R/W	0	浮点除 0 异常中断屏蔽  值 描述 0 FPDZCRIS 中断被抑制，不发送到中断控制器。 1 当 SYSEXCRIS 寄存器中的 FPDZCRIS 位置位时，向中断控制器发送中断。
0	FPIDCIM	R/W	0	浮点输入反常异常中断屏蔽  值 描述 0 FPIDCRIS 中断被抑制，不发送到中断控制器。 1 当 SYSEXCRIS 寄存器中的 FPIDCRIS 位置位时，向中断控制器发送中断。

### 寄存器 3: 系统异常屏蔽的中断状态 ( SYSEXC MIS ) , 偏移量 0x008

SYSEXC MIS 寄存器是屏蔽的中断状态寄存器。读取时, 该寄存器给出相应中断的当前屏蔽状态值。写操作对本寄存器无效。

#### 系统异常屏蔽的中断状态 (SYSEXC MIS)

基址 0x400F.9000  
偏移量 0x008  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											FPIXCMIS	FPOFCMIS	FPUFCMIS	FPIOCMIS	FPDZCMIS	FPIDCMIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5	FPIXCMIS	RO	0	浮点不精确异常屏蔽的中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 由于不精确异常, 因此发出未屏蔽的中断信号。  通过在 SYSEXC IC 寄存器的 FPIXIC 位写 1 可以清除这个位。
4	FPOFCMIS	RO	0	浮点上溢异常屏蔽的中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 由于上溢异常, 因此发出未屏蔽的中断信号。  通过在 SYSEXC IC 寄存器的 FPOFCIC 位写 1 可以清除这个位。
3	FPUFCMIS	RO	0	浮点下溢异常屏蔽的中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 由于下溢异常, 因此发出未屏蔽的中断信号。  通过在 SYSEXC IC 寄存器的 FPUFCIC 位写 1 可以清除这个位。
2	FPIOCMIS	RO	0	浮点无效操作屏蔽的中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 由于无效操作, 因此发出未屏蔽的中断信号。  通过在 SYSEXC IC 寄存器的 FPIOIC 位写 1 可以清除这个位。



位/域	名称	类型	复位	描述
1	FPDZCMIS	RO	0	浮点除 0 异常屏蔽的中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 由于除 0 异常，因此发出未屏蔽的中断信号。  通过在 SYSEXCIC 寄存器的 FPDZCIC 位写 1 可以清除这个位。
0	FPIDCMIS	RO	0	浮点输入反常异常屏蔽的中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 由于输入反常异常，因此发出未屏蔽的中断信号。  通过在 SYSEXCIC 寄存器的 FPIDCIC 位写 1 可以清除这个位。

### 寄存器 4: 系统异常中断清零 ( SYSEXCIC ) , 偏移量 0x00C

SYSEXCIC 寄存器是中断清零寄存器。写入 1 时, 相应的中断 ( 包括原始中断, 如果启用了屏蔽中断, 则还包括屏蔽中断 ) 被清除。对本寄存器写 0 无效。

#### 系统异常中断清零 (SYSEXCIC)

基址 0x400F.9000  
偏移量 0x00C  
类型 W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										FPIXCIC	FPOFCIC	FPUFCIC	FPIOCIC	FPDZCIC	FPIDCIC
类型	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:6	保留	W1C	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5	FPIXCIC	W1C	0	浮点不精确异常中断清零 向该位写 1 可将 SYSEXCIC 寄存器的 FPIXCRIS 位和 SYSEXCICMIS 寄存器的 FPIXCMIS 位清零。
4	FPOFCIC	W1C	0	浮点上溢异常中断清零 向该位写 1 可将 SYSEXCIC 寄存器的 FPOFCRIS 位和 SYSEXCICMIS 寄存器的 FPOFCMIS 位清零。
3	FPUFCIC	W1C	0	浮点下溢异常中断清零 向该位写 1 可将 SYSEXCIC 寄存器的 FPUFCRIS 位和 SYSEXCICMIS 寄存器的 FPUFCMIS 位清零。
2	FPIOCIC	W1C	0	浮点无效操作中断清零 向该位写 1 可将 SYSEXCIC 寄存器的 FPIOCRIS 位和 SYSEXCICMIS 寄存器的 FPIOCMIS 位清零。
1	FPDZCIC	W1C	0	浮点除 0 异常中断清零 向该位写 1 可将 SYSEXCIC 寄存器的 FPDZCRIS 位和 SYSEXCICMIS 寄存器的 FPDZCMIS 位清零。
0	FPIDCIC	W1C	0	浮点输入反常值异常中断清零 向该位写 1 可将 SYSEXCIC 寄存器的 FPIDCRIS 位和 SYSEXCICMIS 寄存器的 FPIDCMIS 位清零。

## 7 休眠模块

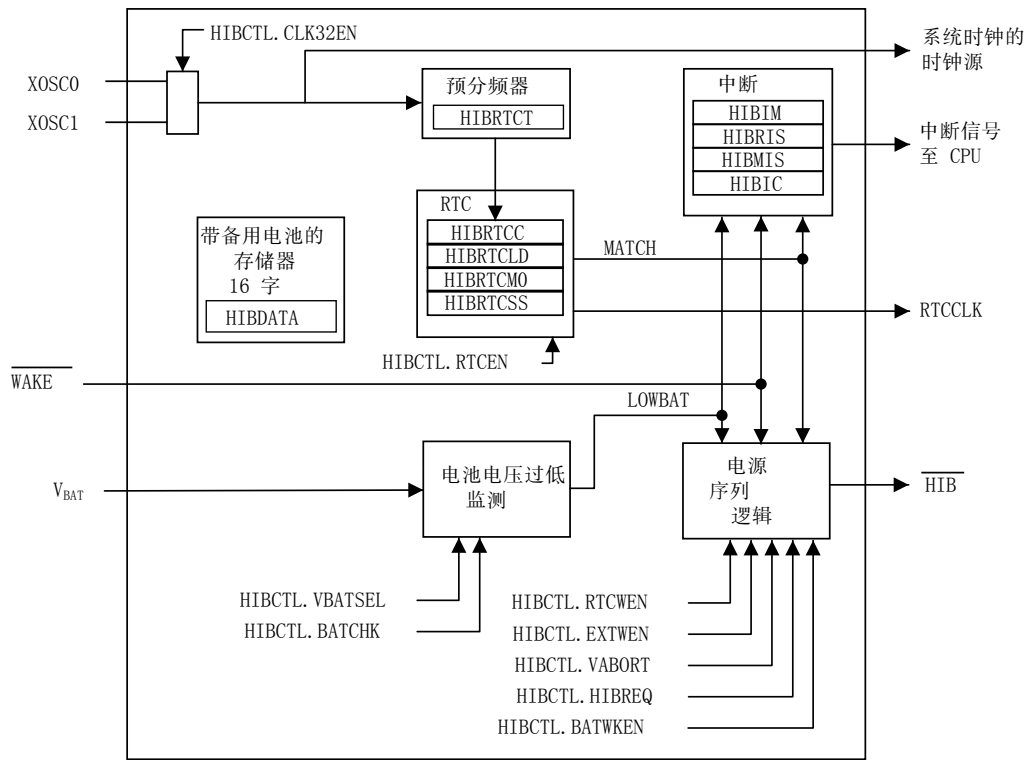
休眠模块管理电源的关断和恢复，以便降低功耗。当处理器和外设空闲时，电源可以完全关断，只维持休眠模块的供电。电源供电可以利用外部信号触发恢复，也可以利用内置实时时钟 (RTC)，在经过一段特定的时间之后恢复。休眠模块可以由电池或者辅助电源单独供电。

休眠模块具有以下特点：

- 32 位实时秒计数器 (RTC)，其时钟分辨率是 1/32,768
  - 32 位的 RTC 秒匹配寄存器和 15 位的亚秒匹配寄存器（其时钟分辨率是 1/32,768），用作定时唤醒和中断产生
  - RTC 预分频器调整，对时钟速率进行良好地调节
- 电源控制的两种机制：
  - 使用离散的外部稳压器控制系统电源
  - 使用在寄存器控制下的内部开关控制片上电源
- 使用外部信号用作唤醒的专用管脚
- 只要  $V_{BAT}$  有效，RTC 运行的内存和休眠所占内存就一直有效
- 电池电量低检测、发出信号和中断发生；在电量低时提供可选的唤醒操作
- GPIO 管脚的状态在休眠过程中可保持不变
- 时钟源可以是 32.768 KHz 的外部晶振或振荡器
- 16 个 32 位字的带备用电池存储器，用于在休眠过程中保存状态
- 可编程的 RTC 匹配、外部唤醒和低电池电压事件的中断

## 7.1 结构图

图 7-1. 休眠模块结构图



## 7.2 信号描述

以下表格列出了休眠模块的外部信号，并且分别描述了它们的功能。

RTCCLK 信号是 GPIO 信号的复用功能，它在复位时将重置为默认的 GPIO 信号。以下表格中“复用管脚/赋值”一栏列出了 RTCCLK 信号的 GPIO 管脚布局。必须将括号中的数字写入 GPIO 端口控制 (GPIOPTL) 寄存器 (636页) 的 PMCn 域中，以便把 RTCCLK 信号分配给指定 GPIO 端口管脚。此外，应将 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618页) 的 AFSEL 置位，以便选择 RTCCLK 功能。有关如何配置 GPIO，请参阅“通用输入/输出端口 (GPIO)”在 597页。

其余信号也具有固定的管脚分配和功能。

表 7-1. 休眠 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
GNDX	75	固定	-	电源	休眠振荡器的接地。使用晶振时钟源时，该管脚应仅连接到晶振负载电容，以提高振荡器对系统噪声的免疫能力。使用外部振荡器时，该管脚应连接到 GND。
HIB	73	固定	O	TTL	该输出指示处理器是否处于休眠模式。
RTCCLK	112	PK4 (7)	O	TTL	休眠模块缓冲版的 32.768-kHz 时钟。此部分处于休眠模式时，该信号并非输出信号。
VBAT	77	固定	-	电源	休眠模块的电源供应源它通常连接到电池的正极端并用作备用电池/休眠模块电源供应器的电源。

表 7-1. 休眠 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
$\overline{\text{WAKE}}$	72	固定	I	TTL	当有效时外部输入将处理器从休眠模式中唤醒。
XOSC0	74	固定	I	模拟	休眠模块晶体振荡器输入或外部时钟参考输入。请注意休眠模块 RTC 采用 32.768-kHz 晶体或 32.768-kHz 振荡器。
XOSC1	76	固定	O	模拟	休眠模块晶体振荡器输出。使用单端时钟源时不连接。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 7.3 功能描述

休眠模块具有两种电源控制机制：

- 第一种机制是利用控制信号 ( $\overline{\text{HIB}}$ ) 来控制微处理器的电源。该信号可以控制外部电压稳压器的开启或者关断。
- 第二种机制是在保持 I/O 管脚的电源供电 (VDD3ON 模式) 时，利用内部开关来控制 Cortex-M4F 以及大多数模拟和数字功能的电源通断。

休眠模块的电源采用动态方式确定。休眠模块的电源取以下较大值：主电压源 ( $V_{DD}$ ) 以及电池/辅助电压源 ( $V_{BAT}$ )。休眠模块还具有单独的时钟源，当系统时钟断电之后，该时钟源可以继续维持实时时钟 (RTC) 功能。休眠模式可以通过以下两种方法之一进入：

- 用户将休眠控制 (HIBCTL) 寄存器的 HIBREQ 位置位，从而启动休眠
- 应用有效的  $V_{BAT}$  时，电源接受仲裁，不再使用  $V_{DD}$ 。

休眠状态期间，当外部管脚 ( $\overline{\text{WAKE}}$ ) 被置位或者内部 RTC 达到特定时间段时，休眠模块就会给外部电源稳压器发出信号，指示恢复系统供电。休眠模块还可以监测电池电压是否过低。如果监测到这种情况，休眠模块就不会进入休眠状态或从休眠状态中唤醒。

从休眠状态中唤醒时， $\overline{\text{HIB}}$  信号将被清零。恢复  $V_{DD}$  电压将引起上电复位 (POR)。从  $\overline{\text{WAKE}}$  信号被置位到程序开始执行的时间就等于唤醒时间 ( $t_{\text{WAKE\_TO\_HIB}}$ ) + 上电复位时间 ( $T_{\text{POR}}$ )。

### 7.3.1 寄存器访问时序

因为休眠模块具有单独的时钟域，所以对休眠寄存器进行写入操作时，各次访问之间必须存在时间差。该延迟时间即  $t_{\text{HIB\_REG\_ACCESS}}$ ，因此软件必须把这个延迟时间插入休眠寄存器的连续写入活动之间，或者插入连续写入和读取活动之间。HIBMIS 寄存器中的 WC 中断用于告知应用程序休眠模块寄存器什么时候可以访问。或者，软件可以利用休眠控制 (HIBCTL) 寄存器中的 WRC 位来确保所需时间差是否已经过去。该位在写入过程中被清零，写入完成后置位，以告诉软件下一个写入或者读取活动可以安全地开始。软件应轮询 HIBCTL 以了解 WRC 是否置位，然后访问任何休眠寄存器。

连续读取休眠模块寄存器没有时序限制。读取活动将以外设时钟的最高速率进行。

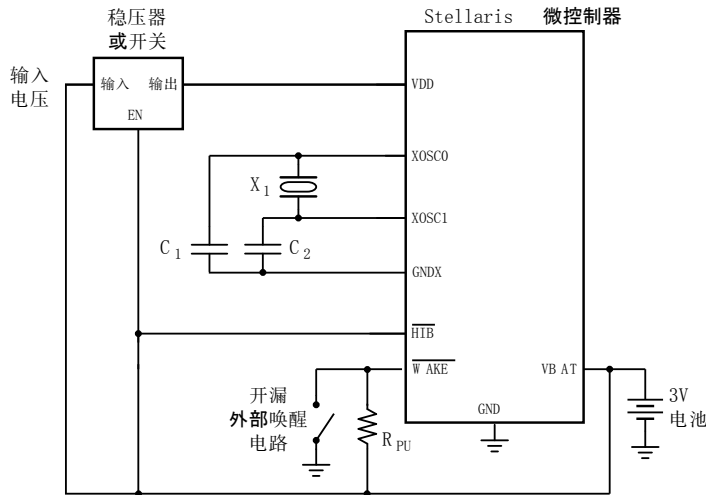
### 7.3.2 休眠时钟源

在使用休眠模块让微控制器进入休眠状态的系统中，休眠模块必须根据独立于主系统时钟之外的外部时钟源计时，即便没有使用 RTC 功能。外部振荡器或者晶振可以作为休眠模块的时钟源。要使用晶振，应将 32.768-kHz 晶振连接到 XOSC0 和 XOSC1 管脚上。或者，可以直接在 XOSC0 管脚上连接一个 32.768-kHz 振荡器，而保持 XOSC1 悬空。请注意，32.768-kHz 振荡器的电压振幅一定

要小于  $V_{BAT}$ ，否则，在休眠状态下，休眠模块就会从振荡器（而不是  $V_{BAT}$ ）中取电。参考图 7-2 在 478 页 和 图 7-3 在 478 页。

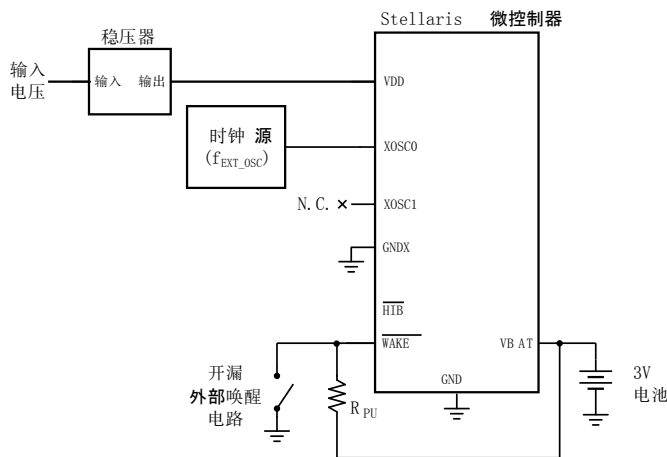
将 HIBCTL 寄存器中的 CLK32EN 位置位即可启用休眠时钟源。访问任何休眠模块寄存器之前，必须将 CLK32EN 位置位。如果采用晶振作为时钟源，那么在写入 CLK32EN 位之后，软件必须设置  $t_{HIBOSC\_START}$  延迟时间，然后方可访问休眠模块寄存器。晶振将在该延迟时间内加电并稳定下来。如果采用振荡器做时钟源，则无需设置延迟时间。使用外部时钟源时，应将 HIBCTL 寄存器的 OSCBYP 位置位。使用晶振时钟源时，GNDX 管脚只能依 图 7-2 在 478 页 所示连接到晶振负载电容，以改进振荡器对系统噪音的免疫能力。使用外部时钟源时，GNDX 管脚应连接到 GND。

图 7-2. 使用晶振作为休眠模块的时钟源（单一电池源）



注意：  
 $X_1$  = 晶振频率为  $f_{XOSC\_XTAL}$ 。  
 $C_{1,2}$  = 晶振厂家提供的负载电容说明中的电容值。  
 $R_{PU}$  = 上拉电阻为 200 k $\Omega$   
 具体参数值请参考“Hibernation Clock Source Specifications”在 1299 页。

图 7-3. 使用专用振荡器作为休眠模块的时钟源



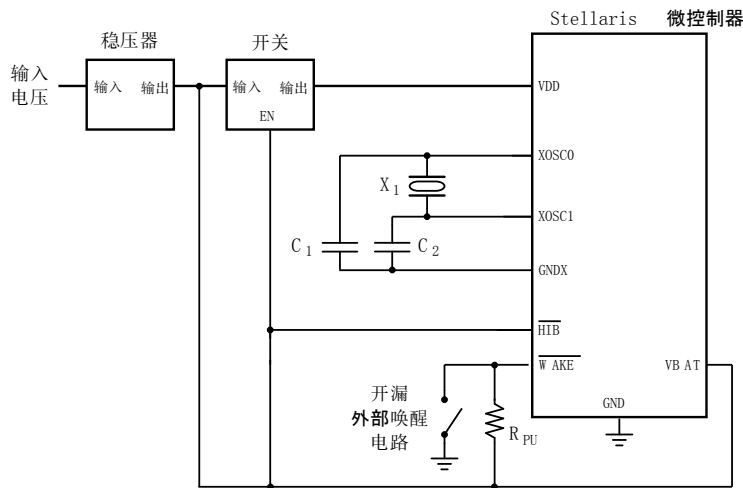
注意：  
 $R_{PU}$  = 上拉电阻为 1 M $\Omega$

### 7.3.3 系统实现

使用休眠模块时可以实现多种不同的系统配置：

- 使用单一电池源，电池提供  $V_{DD}$  和  $V_{BAT}$ ，如图 7-2 在 478 页 所示。
- 使用 VDD3ON 模式， $V_{DD}$  在休眠时仍然带电，从而允许 GPIO 管脚保留其状态，如图 7-3 在 478 页 所示。在此模式中， $V_{DDC}$  从内部断电。重新上电时，GPIO 保留值将释放，GPIO 将初始化为默认值。
- $V_{DD}$  和  $V_{BAT}$  使用独立源。在此模式中，没有电池或电池耗尽时，系统需要通过其他电路来启动，更多信息请参考 EK-LM4F232 主板示意图。
- 使用稳压器提供  $V_{DD}$  和  $V_{BAT}$ ，并通过  $\overline{HIB}$  启用开关，从而在休眠时移除  $V_{DD}$ ，如图 7-4 在 479 页 所示。

图 7-4.  $V_{DD}$  和  $V_{BAT}$  使用稳压器



添加外部电容到  $V_{BAT}$  源降低了低电池电压测量的准确性，应尽量避免。本节提及的原理图只显示了休眠模块管脚（而非整个系统）的连接。

如果应用程序不要求使用休眠模块，请参考“未用信号的连接”在 1288 页。在这种情况下，必须将运行模式时钟门控控制寄存器 0 (RCGC0) 和休眠运行模式时钟门控控制 (RCGCHIB) 寄存器中的 HIB 位清零，以禁用休眠模块的系统时钟信号，同时让休眠模块的寄存器拒绝访问。

### 7.3.4 电池管理

**重要：** 系统级问题可能会影响电池电压低监测电路的准确性。设计人员应该考虑电池的类型、放电特性，以及电池电压测量过程中的测试负载。

休眠模块可以由电池或者使用 VBAT 管脚的辅助电源单独供电。该模块可以监视电池的电压水平，并可检测电池电压何时低于  $V_{LOWBAT}$ 。通过 HIBCTL 寄存器的 VBATSEL 域，可将该电压阈值设在 1.9 V 和 2.5 V 之间。用户还可将该模块设置为当电池电压低于此阈值时，系统不进入休眠状态。此外，处于休眠状态时，该模块将检测电池电压。通过 HIBCTL 寄存器的 BATWKEN 位，微控制器可配置为当电池电压低于此阈值时即从休眠状态中唤醒。

休眠模块用于检测低电池电压状态，并在发生此状态时将休眠原始中断状态 (HIBRIS) 寄存器的 LOWBAT 位置位。如果 HIBCTL 寄存器中的 VABORT 位也被置位，那么当检测到电池电压过低

时，模块就不会进入休眠状态。用户还可将该模块设置为当电池电压低于此阈值时，模块即产生中断（请参阅“中断和状态”在 482 页）。

注意：休眠模块由电压较高的电源（ $V_{BAT}$  或  $V_{DD}$ ）供电。因此，务必将电路设计成确保  $V_{DD}$  的电压更高、 $V_{BAT}$  处于额定条件下，否则，即使  $V_{DD}$  可用，休眠模块也只由电池供电。

### 7.3.5 实时时钟

休眠模块带有一个实时时钟，其中包含一个 32 位秒计数器和一个 15 位亚秒计数器。32.768-kHz 休眠振荡器时钟信号即 RTC 的时钟源。RTCCLK 输出会提供缓冲过的 32.768-kHz 信号。休眠模块复位的同时，RTC 计数器也会复位。用户可以通过休眠 RTC 加载 (HIBRTCLD) 寄存器将 RTC 32 位秒计数器置位。当 32 位秒计数器被置位后，15 位亚秒计数器就会被清零。要得到 RTC 值，可以首先读取 HIBRTCC 寄存器，再读取休眠 RTC 亚秒 (HIBRTCSS) 寄存器的 RTCSSC 域，然后再次读取 HIBRTCC 寄存器。如果 HIBRTCC 寄存器两次读取的数值一样，则读取有效。

休眠模块还包含一个 32 位匹配寄存器，且其 HIBRTCSS 寄存器中还有一个 15 位域 (RTCSSM)。它们用于和 RTC 32 位计数器以及 15 位亚秒计数器的数值进行对比。当存在匹配时，处理器可以通过编程设置从休眠状态中唤醒，和/或产生中断信号以通知处理器。匹配产生中断的优先级高于中断清零。因此，如果 RTC 与匹配值相同，那么对休眠中断清除 (HIBIC) 寄存器的 RTCALT0 位的写入操作不会将该位清零。通过 HIBCTL 寄存器中的 RTCEN 位禁用 RTC 以后，匹配中断将被清零。

预分频器修正寄存器 HIBRTCT 可以用来补偿时钟源的误差。该寄存器的默认值为 0x7FFF。当 HIBRTCC 寄存器中的 [5:0] 位由 0x00 变成 0x01 时，该默认值每隔 64 秒就会调整一次，以便将输入时间分频。通过此配置，软件可以调整预分频器修正寄存器的数值（使其大于或者小于 0x7FFF），以便精确修正时钟速率。要降低 RTC 时钟速率，就要让预分频器修正寄存器的值大于 0x7FFF；要增加 RTC 时钟速率，就要让预分频器修正寄存器的值小于 0x7FFF。

使用接近 HIBRTCSS 寄存器亚秒匹配值的修正值时，必须小心谨慎。使用大于 0x7FFF 的修正值时，可能导致同一计数器值发生两个匹配中断。此外，使用低于 0x7FFF 的修正值时，可能丢失匹配中断。

修正值大于 0x7FFF 时，如果 HIBRTCSS 寄存器的 RTCSSC 值达到 0x7FFF，则 RTCC 值从 0x0 增加到 0x1，而 RTCSSC 值要减去修正量。RTCSSC 值在重新回到 0x0 并开始下次递增计数前，先会再次递增计数至 0x7FFF。如果匹配值在该范围以内，匹配中断将触发两次。例如，如表 7-2 在 480 页所示，如果 RTCM0=0x1 且 RTCSSM=0x7FFD 时配置匹配中断，将触发两个中断。

表 7-2. TRIM 值为 0x8003 时的计数器行为

RTCC [6:0]	RTCSSC
0x00	0x7FFD
0x00	0x7FFE
0x00	0x7FFF
0x01	0x7FFC
0x01	0x7FFD
0x01	0x7FFE
0x01	0x7FFF
0x01	0x0
0x01	0x1
-	-
0x01	0x7FFB
0x01	0x7FFC
0x01	0x7FFD
0x01	0x7FFE



表 7-2. TRIM 值为 0x8003 时的计数器行为 (续)

RTCC [6:0]	RTCSSC
0x01	0x7FFF
0x02	0x0

修正值小于 0x7FFF 时，RTCSSC 值从 0x7FFF 增加到修正值，而 RTCC 值从 0x0 增加到 0x1。如果匹配值在该范围以内，匹配中断将不会触发。例如，如表 7-3 在 481 页所示，如果 RTCM0=0x1 且 RTCSSM=0x2 时配置匹配中断，绝不会触发中断。

表 7-3. TRIM 值为 0x7FFC 时的计数器行为

RTCC [6:0]	RTCSSC
0x00	0x7FFD
0x00	0x7FFE
0x00	0x7FFF
0x01	0x3
0x01	0x4
0x01	0x5

RTC 必须通过 HIBCTL 寄存器的 RTCEN 位启用。用户可利用中断寄存器（请参阅“中断和状态”在 482 页）设置 RTC，让其产生中断信号。只要启用了 RTC 并存在有效的  $V_{BAT}$ ，RTC 就会继续计数，而不论  $V_{DD}$  是否存在或该部件是否处于休眠模式。

### 7.3.6 带备用电池的存储器

休眠模块包含 1632 位字的存储器，并采用电池或者外部辅助电源供电，因此它们在休眠期间也会保持通电。处理器软件可以在休眠之前将状态信息保存在该存储器中，并在唤醒时恢复。用户可以通过 HIBDATA 寄存器访问带备用电池的存储器。如果  $V_{DD}$  和  $V_{BAT}$  都被移除，HIBDATA 寄存器中的内容将丢失。

### 7.3.7 使用 HIB 控制电源

**重要：** 使用 HIB 控制电源时，休眠模块需要特殊的系统实施考量，因为它会关闭微控制器所有其他部分的电源。所有连接到芯片的系统信号和电源都必须驱动输出至  $0 V_{DC}$ ，或者通过由 HIB 控制的相同稳压器关闭电源。

休眠模块利用 HIB 管脚控制微控制器的电源。该管脚会连接到为微控制器和其他电路提供 3.3 V 电源的外部稳压器的启用信号端。当 HIB 信号被休眠模块置位以后，外部稳压器将关闭，而不再为微处理器和其他系统部分供电。休眠模块一直会从  $V_{BAT}$  电源（可以是电池，也可以是外部辅助电源）取电，直到唤醒事件发生。将 HIB 清零即可恢复微处理器的供电，进而使外部稳压器恢复为芯片供电。

### 7.3.8 以 VDD3ON 模式管理电源

休眠模块还可以用来切断芯片内部模块的电源。在这种情况下，所有的管脚都会保持进入休眠之前的状态。举个例子，输入管脚还是输入功能；输出驱动高电平仍保持驱动高电平，等等。在 VDD3ON 模式中，稳压器应将休眠模式下的微控制器电源保持在 3.3 V。这种电源控制模式可通过 HIBCTL 寄存器中的 VDD3ON 位启用。

### 7.3.9 启动休眠

当 HIBCTL 寄存器中的 HIBREQ 位被置位，休眠功能就会启动。如果尚未使用 HIBCTL 寄存器的 PINWEN 或 RTCWEN 位配置唤醒条件，休眠请求将被忽略。如果 HIBREQ 位置位时正在进行 Flash 存储器写入操作，那么互锁功能就会延迟系统进入休眠模式，直到写入操作完成。另外，如果电池电压低于 HIBCTL 寄存器的 VBATSEL 域所定义的阈值，那么休眠请求无效。

### 7.3.10 从休眠模式唤醒

通过将 HIBCTL 寄存器中的 PINWEN 位置位，休眠模块就会被外部 WAKE 管脚唤醒。将 RTCWEN 位置位，休眠模块就会被 RTC 匹配唤醒。请注意， $\overline{\text{WAKE}}$  管脚使用休眠模块的内部电压作为逻辑 1 的参阅。

不管使用外部唤醒还是 RTC 匹配唤醒，休眠模块的唤醒将延迟，直到  $V_{DD}$  高于最小电压值（请参阅表 25-2 在 1292 页）。

当休眠模块被唤醒时，微控制器会进行正常的上电复位。注：该复位不会重置休眠模块，但会重置微控制器的其他部分。通过观察中断状态寄存器（请参阅“中断和状态”在 482 页）以及查询带备用电池的存储器中的状态数据（请参阅“带备用电池的存储器”在 481 页），软件可以监测到休眠唤醒上电。

### 7.3.11 仲裁性电源移除

如果 CLK32EN 位已置位且 PINWEN 位或 RTCEN 位已置位，微控制器将在仲裁移除  $V_{DD}$  时进入休眠模式。重新上电时，微控制器从休眠状态唤醒。如果 CLK32EN 位已置位但 PINWEN 位或 RTCEN 位都未置位，微控制器仍然会在移除电源时进入休眠模式，但当重新接通电压  $V_{DD}$  时，MCU 将执行冷上电复位，且休眠模块将复位。如果 CLK32EN 位没有置位且  $V_{DD}$  被仲裁移除，该部件只是关机并在重新上电时执行冷上电复位。

如果仲裁移除  $V_{DD}$  时正在进行 Flash 存储器写入操作，必须在重新应用  $V_{DD}$  时重试写入操作。

### 7.3.12 中断和状态

当以下条件出现时，休眠模块可以产生中断信号：

- $\overline{\text{WAKE}}$  管脚置位
- RTC 匹配
- 电池电压过低
- 写入完成/可以写入

所有的中断相或后发送到中断控制器，因此在给定的时间，休眠模块只能向控制器产生一个中断请求。软件中断处理器可以通过读取 休眠屏蔽中断状态 (HIBMIS) 寄存器来处理多个中断事件。软件还可以通过访问 HIBRIS 寄存器，随时读取休眠模块的状态。该寄存器会显示所有挂起的事件。从休眠模式中唤醒后，该寄存器可以用来判断唤醒条件是上述事件之一还是电源断开。

通过设置休眠中断屏蔽 (HIBIM) 寄存器中相应的位，即可配置将触发中断信号的事件。通过修改休眠中断清除 (HIBIC) 寄存器中相应的位，即可清除挂起的中断。

## 7.4 初始化及配置

休眠模块有几种不同的配置。下面各小节说明了不同情况下推荐的编程步骤。因为休眠模块以 32.768 kHz 的频率运行，与根据系统时钟运行的微控制器的其他部分同步，因此软件在写入寄存器（请参考“寄存器访问时序”在 477 页）之后，必须存在一个延迟时间  $t_{\text{HIB\_REG\_ACCESS}}$ 。HIBMIS 寄存器中的 WC 中断用于告知应用程序休眠模块寄存器什么时候可以访问。

## 7.4.1 初始化

休眠模块启用系统时钟即可完成复位过程，但是如果模块的系统时钟被禁用，那么必须重新启用系统时钟，即便不使用 RTC 功能。请参考 306页。

如果使用 32.768-kHz 晶振作为休眠模块的时钟源，请遵循以下步骤：

1. 向 HIBIM 寄存器写入 0x0000.0010，以便启用 WC 中断。
2. 向偏移量为 0x10 的 HIBCTL 寄存器写入 0x40 以启用振荡器输入。
3. 等待 HIBMIS 寄存器中的 WC 中断被触发以后，然后再对休眠模块进行其他操作。

如果使用单端 32.768-kHz 振荡器作为休眠模块时钟源，请遵循以下步骤：

1. 向 HIBIM 寄存器写入 0x0000.0010，以便启用 WC 中断。
2. 在 HIBCTL 寄存器中写入 0x0001.0040（偏移量 0x10）以启用振荡器输入，并规避芯片内部的振荡器。
3. 等待 HIBMIS 寄存器中的 WC 中断被触发以后，然后再对休眠模块进行其他操作。

以上步骤只需要在整个系统第一次初始化的时候执行。如果微控制器已处于休眠状态，那么休眠模块已经上电，无需执行以上步骤。软件可以通过观察 HIBCTL 寄存器中的 CLK32EN 位监测休眠模块和时钟是否已经上电。

表 7-4 在 483页 说明了在正常运行以及休眠状态下，位的各种设置与时钟功能的关系。

表 7-4. 休眠模块的时钟运行

CLK32EN	PINWEN	RTCWEN	RTCEN	正常运行	休眠
0	X	X	X	休眠模块禁用	休眠模块禁用
1	0	0	1	RTC 匹配功能启用。	无休眠
1	0	1	1	模块计时	将 RTC 匹配作为唤醒事件
1	1	0	0	模块计时	休眠期间时钟模块断电；外部唤醒事件发生时，时钟模块上电。
1	1	0	1	模块计时	休眠期间时钟模块依旧供电，以实现 RTC 功能。根据外部事件唤醒。
1	1	1	1	模块计时	RTC 匹配或者外部唤醒事件，以先发生者为准。

## 7.4.2 RTC 匹配功能（无休眠）

请按照以下步骤配置休眠模块的 RTC 匹配功能：

1. 在 HIBCTL 寄存器中写入 0x0000.0040（偏移量 0x010）以启用 32.768-kHz 休眠振荡器。
2. 在 HIBRTCM0 寄存器中写入所需的 RTC 匹配值（偏移量 0x004）；并在 HIBRTCSS 寄存器的 RTCSSM 域中写入该 RTC 匹配值（偏移量 0x028）。
3. 在 HIBRTCLD 寄存器中写入所需的 RTC 负载值（偏移量 0x00C）。
4. 在 HIBIM 寄存器中的 RTCALTO 写入所需的 RTC 匹配中断掩码（偏移量 0x014）。
5. 在 HIBCTL 寄存器中写入 0x0000.0041（偏移量 0x010）让 RTC 开始计数。

### 7.4.3 RTC 匹配/唤醒

请使用以下步骤实现 RTC 匹配和睡眠模块的唤醒功能：

1. 在 HIBCTL 寄存器中写入 0x0000.0040 ( 偏移量 0x010 ) 以启用 32.768-kHz 睡眠振荡器。
2. 在 HIBRTCM0 寄存器中写入所需的 RTC 匹配值 ( 偏移量 0x004 ) ；并在 HIBRTCSS 寄存器的 RTCSSM 域中写入该 RTC 匹配值 ( 偏移量 0x028 ) 。
3. 在 HIBRTCLD 寄存器中写入所需的 RTC 负载值 ( 偏移量 0x00C ) 。该写入会将 15 位亚秒计数器清零。
4. 在 HIBDATA 寄存器中写入需要在断电的时候保留的任意数据 ( 偏移量 0x030-0x06F ) 。
5. 在 HIBCTL 寄存器中写入 0x0000.004B ，偏移量 0x010 ，以设置外部唤醒功能，并开始休眠序列。

### 7.4.4 外部唤醒

如果要使用 `WAKE` 管脚作为微控制器的唤醒源，请遵循以下步骤：

1. 在 HIBCTL 寄存器中写入 0x0000.0040 ( 偏移量 0x010 ) 以启用 32.768-kHz 睡眠振荡器。
2. 在 HIBDATA 寄存器中写入需要在断电的时候保留的任意数据 ( 偏移量 0x030-0x06F ) 。
3. 在 HIBCTL 寄存器中写入 0x0000.0052 ( 偏移量 0x010 ) ，以启用外部唤醒功能，并开始休眠序列。

请注意，在这种模式下，如果禁用了 RTC，那么睡眠时钟源在睡眠模式下就会断电，并在外部唤醒事件发生时恢复供电。如果在休眠之前启用 RTC，那么在休眠期间，它会继续工作。

### 7.4.5 RTC 或外部唤醒

1. 在 HIBCTL 寄存器中写入 0x0000.0040 ( 偏移量 0x010 ) 以启用 32.768-kHz 睡眠振荡器。
2. 在 HIBRTCM0 寄存器中写入所需的 RTC 匹配值 ( 偏移量 0x004 ) ；并在 HIBRTCSS 寄存器的 RTCSSM 域中写入该 RTC 匹配值 ( 偏移量 0x028 ) 。
3. 在 HIBRTCLD 寄存器中写入所需的 RTC 负载值 ( 偏移量 0x00C ) 。该写入会将 15 位亚秒计数器清零。
4. 在 HIBDATA 寄存器中写入需要在断电的时候保留的任意数据 ( 偏移量 0x030-0x06F ) 。
5. 在 HIBCTL 寄存器中写入 0x0000.005B ，偏移量 0x010 ，以设置 RTC 匹配/外部唤醒功能，并开始休眠序列。

## 7.5 寄存器映射

表 7-5 在 485 页列出了睡眠寄存器的信息。所有给出的地址都是相对于 0x400F.C000 的结束地址的睡眠模块基址而言的。请注意，对寄存器编程之前，必须启用睡眠模块的系统时钟。( 请参考 306 页 ) 。启用睡眠模块时钟之后，必须经过 3 个系统时钟的延迟才能访问睡眠模块的寄存器。另外，在访问其他任何睡眠模块寄存器之前，必须将 HIBCTL 寄存器的 CLK32EN 位置位。

**注意：** 睡眠模块寄存器处于睡眠模块的时钟域内，具有特殊的时序要求。软件应该利用 HIBCTL 寄存器中的 WRC 位来确保所需的时间差是否已经过去。如果 WRC 位被清零，那么任何写入访问都不起作用。请参考“寄存器访问时序”在 477 页。

**重要:** 休眠模块寄存器在两种情况下会复位：

1. HIBCTL 寄存器中的 RTCEN和 PINWEN 位都被清零时，系统将复位。
2.  $V_{DD}$  和  $V_{BAT}$  电源都被切断时将发生冷上电复位 (POR)。

任何其他复位条件都会被休眠模块忽略。

表 7-5. 休眠模块 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	HIBRTCC	RO	0x0000.0000	休眠 RTC 计数器寄存器	486
0x004	HIBRTCM0	R/W	0xFFFF.FFFF	休眠 RTC 匹配寄存器 0	487
0x00C	HIBRTCLD	R/W	0x0000.0000	休眠 RTC 加载寄存器	488
0x010	HIBCTL	R/W	0x8000.2000	休眠控制寄存器	489
0x014	HIBIM	R/W	0x0000.0000	休眠中断屏蔽寄存器	493
0x018	HIBRIS	RO	0x0000.0000	休眠原始中断状态寄存器	495
0x01C	HIBMIS	RO	0x0000.0000	休眠屏蔽中断状态寄存器	497
0x020	HIBIC	R/W1C	0x0000.0000	休眠中断清除寄存器	499
0x024	HIBRTCT	R/W	0x0000.7FFF	休眠 RTC 修正寄存器	500
0x028	HIBRTCSS	R/W	0x0000.0000	休眠 RTC 亚秒寄存器	501
0x030-0x06F	HIBDATA	R/W	-	休眠数据寄存器	502

## 7.6 寄存器描述

本节的其余部分按照地址偏移量的数字顺序列出和描述了休眠模块寄存器。

### 寄存器 1: 休眠 RTC 计数器寄存器 ( HIBRTCC ) , 偏移量 0x000

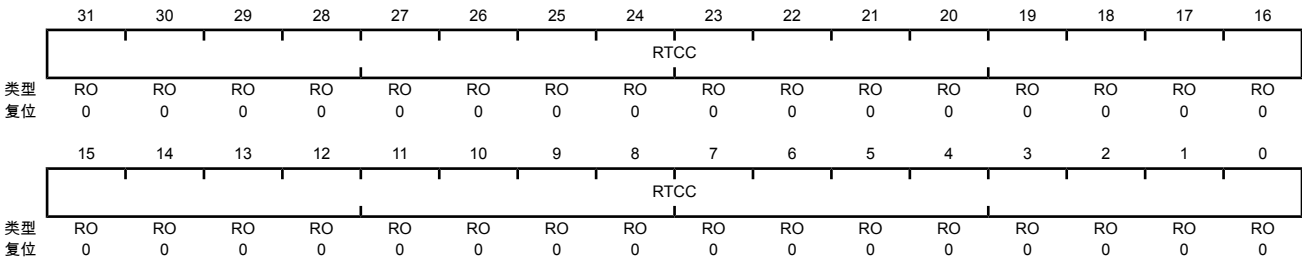
该寄存器的值是 RTC 计数器的当前 32 位值。

RTC 计数器由一个 32 位秒计数器和一个 15 位亚秒计数器组成。休眠模块复位的同时，RTC 计数器也会复位。用户可以通过 HIBRTCLD 寄存器将 RTC 32 位秒计数器置位。当 32 位秒计数器被置位后，15 位亚秒计数器就会被清零。

要得到 RTC 值，可以首先读取 HIBRTCC 寄存器，再读取 RTCSSC 寄存器的 HIBRTCSS 域，然后再次读取 HIBRTCC 寄存器。如果 HIBRTCC 寄存器两次读取的数值一样，则读取有效。

#### 休眠 RTC 计数器寄存器 (HIBRTCC)

基址 0x400F.C000  
偏移量 0x000  
类型 RO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:0	RTCC	RO	0x0000.0000	RTC 计数器

读取返回 32 位计数器数值。因为 RTC 已经启用，所以该数值代表延迟的秒数。该寄存器为只读状态。使用 HIBRTCLD 寄存器可改变该值。

**寄存器 2: 休眠 RTC 匹配寄存器 0 ( HIBRTCM0 ) , 偏移量 0x004**

该寄存器是 RTC 计数器的一个 32 位秒匹配寄存器。15 位亚秒匹配数值存储在 HIBRTCSS 寄存器的 RTCSSC 域中，它可以与该寄存器配合使用，以得到更精确的时间匹配。

**注意：** 休眠模块寄存器处于休眠模块的时钟域内，具有特殊的时序要求。软件应该利用 HIBCTL 寄存器中的 WRC 位来确保所需的时间差是否已经过去。如果 WRC 位被清零，那么任何写入访问都不起作用。请参考“寄存器访问时序”在 477 页。

**休眠 RTC 匹配寄存器 0 (HIBRTCM0)**

基址 0x400F.C000  
偏移量 0x004  
类型 R/W, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCM0															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCM0															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	RTCM0	R/W	0xFFFF.FFFF	RTC 匹配 0 将数值写入 RTC 匹配寄存器。 读取返回当前匹配值。

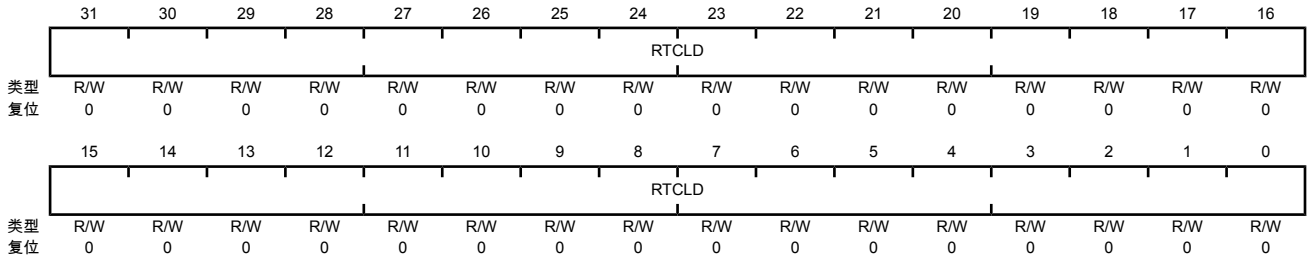
### 寄存器 3: 休眠 RTC 加载寄存器 ( HIBRTCLD ) , 偏移量 0x00C

该寄存器用来向 RTC 计数器加载一个 32 位数值。当该寄存器被写入时，加载立即发生。当该寄存器写入数值时，15 位亚秒计数器也会被清零。

**注意:** 休眠模块寄存器处于休眠模块的时钟域内，具有特殊的时序要求。软件应该利用 HIBCTL 寄存器中的 WRC 位来确保所需的时间差是否已经过去。如果 WRC 位被清零，那么任何写入访问都不起作用。请参考“寄存器访问时序”在 477 页。

#### 休眠 RTC 加载寄存器 (HIBRTCLD)

基址 0x400F.C000  
偏移量 0x00C  
类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:0	RTCLD	R/W	0x0000.0000	RTC 加载 写入活动会将当前值加载到 RTC 计数器 (RTCC) 中。 读取时会返回 32 位数值。



## 寄存器 4: 休眠控制寄存器 (HIBCTL), 偏移量 0x010

该寄存器是休眠模块的控制寄存器。在发生休眠事件之前, 该寄存器必须最后写入。如果在将HIBREQ位置位之后再写入其他寄存器, 那么就不能确保在休眠发生之前完成数据的写入。

## 休眠控制寄存器 (HIBCTL)

基址 0x400F.C000

偏移量 0x010

类型 R/W, 复位 0x8000.2000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WRC	保留												OSCHYS	OSCDRV	OSCBYP
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	VBATSEL	保留		BATCHK	BATWKEN	VDD3ON	VABORT	CLK32EN	保留	PINWEN	RTCWEN	保留	HIBREQ	RTCEN	
类型	RO	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W	R/W	RO	R/W	R/W	RO	R/W	R/W
复位	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31	WRC	RO	1	<p>写入完成/可以写入</p> <p><b>值 描述</b></p> <p>0 该接口正在处理优先级高的写入, 处于繁忙状态。当 WRC 是 0 时, 任何写入操作都会导致不确定的行为。</p> <p>1 该接口可以接受写入。</p> <p>软件必须在写入请求之间查询该位, 并延迟写入活动直到 WRC=1, 以确保操作正确。可配置一个中断来表明 WRC 已完成。</p> <p>该位的名称 WRC 表示“写入完成”, 是该位的默认用法(写入访问之间)。然而, 因为该位无需复位即可置位, 所以这个名称还表示“可以写入”, 说明该接口可以通过软件写入数值。软件可以利用这个区别在复位时监测哪种编程比较合适: 0 = 需要软件延迟循环; 1 = WRC 节奏可用。</p>
30:19	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
18	OSCHYS	R/W	0	<p>32.768-kHz 振荡器迟滞控制</p> <p>当休眠振荡器断电或者没有外接晶振时, 该位可以用来防止自激震荡。为了获得最低电平电源状态, 该位应该被置位。</p> <p><b>值 描述</b></p> <p>1 迟滞启用</p> <p>0 迟滞禁用</p>
17	OSCDRV	R/W	0	<p>振荡器驱动能力</p> <p>该位用来补偿过大或者过小的滤波电容。</p> <p><b>注意:</b> 休眠振荡器开始之后, 该位不应该被改变。振荡器运行时, 如果用户改变了这个值, 振荡器可能会不稳定。</p> <p><b>值 描述</b></p> <p>1 高电平驱动强度已启用, 24 pF。</p> <p>0 低电平驱动强度已启用, 12 pF。</p>

位/域	名称	类型	复位	描述
16	OSCBYP	R/W	0	<p>振荡器旁路</p> <p>值 描述</p> <p>1 内部 32.768-kHz 休眠振荡器已禁用，并且已断电。当使用连接到 XOSC0 的单端振荡器时，该位应该被置位。</p> <p>0 内部 32.768-kHz 休眠振荡器已启用。当使用外部 32.768-kHz 晶振时，该位应该被清零。</p>
15	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
14:13	VBATSEL	R/W	0x1	<p>选择低电池电压比较器</p> <p>该域确定检查电池状态时所用的电池电压值。如果电池电压低于指定值，HIBRIS 寄存器中的 LOWBAT 中断位就会被置位。</p> <p>值 描述</p> <p>0x0 1.9 V</p> <p>0x1 2.1 V (默认)</p> <p>0x2 2.3 V</p> <p>0x3 2.5 V</p>
12:11	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
10	BATCHK	R/W	0	<p>检查电池状态</p> <p>值 描述</p> <p>1 读取到该值则表示低电池电压比较器周期尚未完成。 将此位置位即可启动低电池电压比较器周期。如果电池电压低于 VBATSEL 域指定的电压值，HIBRIS 寄存器中的 LOWBAT 中断位就会被置位。电池检测过程中，休眠请求会被延迟。</p> <p>0 读取到该值表示低电池电压比较器周期没有激活。 写入 0 未生效。</p>
9	BATWKEN	R/W	0	<p>电池电压过低唤醒</p> <p>值 描述</p> <p>1 如果该位被置位，在休眠期间，系统每 512 秒就会检查一次电池电压。如果电压低于 VBATSEL 域指定的电压值，微控制器就会从休眠状态唤醒，HIBRIS 寄存器中的 LOWBAT 中断位会被置位。</p> <p>0 模块不会自动检查电池电压。电池电压过低不会将微处理器从休眠状态中唤醒。</p>

位/域	名称	类型	复位	描述
8	VDD3ON	R/W	0	<p>VDD 上电</p> <p>值 描述</p> <p>1 内部开关控制芯片模块的电源 ( VDD3ON 模式 )。</p> <p>0 内部开关没有使用。HIB 信号应该用来控制外部开关或者稳压器。</p> <p>请注意，在休眠模式下，不管 VDD3ON 位的状态如何，HIB 信号都会被置位。因此，当 VDD3ON 被置位时，HIB 信号不应该连接 3.3V 稳压器，并且 3.3V 电压源应该保持连接。此位在休眠状态被置位时，所有管脚都会保持休眠之前的状态。举个例子，输入管脚还是输入功能；输出驱动高电平仍保持驱动高电平，等等。</p>
7	VABORT	R/W	0	<p>启用电源切断终止</p> <p>值 描述</p> <p>1 该位被置位之后，模块会在进入休眠之前检查电池电压。如果 <math>V_{BAT}</math> 低于 VBATSEL 指定的电压值，那么微控制器不会进入休眠状态。</p> <p>0 不管电池电压处于何种水平，微控制器都会进入休眠状态。</p>
6	CLK32EN	R/W	0	<p>启用时钟</p> <p>要使用休眠模块，必须启用该位。</p> <p>值 描述</p> <p>1 休眠模块时钟源已启用。</p> <p>0 休眠模块时钟源已禁用。</p>
5	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
4	PINWEN	R/W	0	<p>外部 <math>\overline{WAKE}</math> 管脚启用</p> <p>值 描述</p> <p>1 <math>\overline{WAKE}</math> 管脚高电平会把微控制器从休眠状态唤醒。</p> <p>0 <math>\overline{WAKE}</math> 管脚状态对休眠没有影响。</p>
3	RTCWEN	R/W	0	<p>RTC 唤醒启用</p> <p>值 描述</p> <p>1 RTC 匹配事件 ( HIBRTCC 寄存器的值与 HIBRTCM0 寄存器的值匹配，而且 RTCSSC 域的值与 HIBRTCSS 寄存器中 RTCSSM 域的值匹配 ) 会将微控制器从休眠状态中唤醒。</p> <p>0 RTC 匹配事件对休眠没有影响。</p>
2	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>

位/域	名称	类型	复位	描述						
1	HIBREQ	R/W	0	<p>休眠请求</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>将该位置位，启动休眠。</td> </tr> <tr> <td>0</td> <td>无休眠请求。</td> </tr> </tbody> </table> <p>唤醒事件之后，该位会被硬件自动清零。                      如果 PINWEN 和 RTCWEN 位都已清零，将会忽略休眠请求。                      如果 BATCHK 位被置位，那么休眠请求就会被推迟。</p>	值	描述	1	将该位置位，启动休眠。	0	无休眠请求。
值	描述									
1	将该位置位，启动休眠。									
0	无休眠请求。									
0	RTCEN	R/W	0	<p>RTC 计时器启用</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>休眠模块 RTC 功能已启用。</td> </tr> <tr> <td>0</td> <td>休眠模块 RTC 功能已禁用。</td> </tr> </tbody> </table>	值	描述	1	休眠模块 RTC 功能已启用。	0	休眠模块 RTC 功能已禁用。
值	描述									
1	休眠模块 RTC 功能已启用。									
0	休眠模块 RTC 功能已禁用。									

## 寄存器 5: 休眠中断屏蔽寄存器 (HIBIM), 偏移量 0x014

这个寄存器是休眠模块中断源的中断屏蔽寄存器。该寄存器中的每一位都可以屏蔽休眠原始中断状态(HIBRIS)寄存器中相应的位。如果该位没有被屏蔽,那么中断信号就会发送到中断控制器。如果该位被屏蔽,那么中断信号就不会发送到中断控制器。在CLK32EN位置位之前,如果WC位被置位,那么该掩码不会在休眠周期中保留,除非再将该位写入一次。

**注意:** 该寄存器的WRC位位于系统时钟域内,从而对该位的写入操作能及时生效,并能在HIBCTL寄存器的CLKEN位置位之前完成。

### 休眠中断屏蔽寄存器 (HIBIM)

基址 0x400F.C000  
偏移量 0x014  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留											WC	EXTW	LOWBAT	保留	RTCALTO
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:5	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。
4	WC	R/W	0	外部写入完成/可以中断屏蔽  值 描述 1 当 HIBRIS 寄存器中的 WC 位被置位时,中断信号就会发送到中断控制器。 0 WC 中断被抑制,不会发送到中断控制器。
3	EXTW	R/W	0	外部唤醒中断屏蔽  值 描述 1 当 HIBRIS 寄存器中的 EXTW 位被置位时,中断信号就会发送到中断控制器。 0 EXTW 中断被抑制,不会发送到中断控制器。
2	LOWBAT	R/W	0	电池电压过低中断屏蔽  值 描述 1 当 HIBRIS 寄存器中的 LOWBAT 位被置位时,中断信号就会发送到中断控制器。 0 LOWBAT 中断被抑制,不会发送到中断控制器。
1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
0	RTCALTO	R/W	0	<p>RTC 警报 0 中断屏蔽</p> <p>值 描述</p> <p>1 当 HIBRIS 寄存器中的 RTCALTO 位被置位时，中断信号就会发送到中断控制器。</p> <p>0 RTCALTO 中断被抑制，不会发送到中断控制器。</p>

## 寄存器 6: 休眠原始中断状态寄存器 (HIBRIS), 偏移量 0x018

该寄存器是休眠模块中断源的原始中断状态。通过将 HIBIM 寄存器中相应的位清零, 即可屏蔽所有位。当某一位被屏蔽时, 中断信号不会发送到中断控制器。通过将 1 写入休眠中断清除 (HIBIC) 寄存器中相应的位或进入休眠, 即可将该寄存器中的所有位清零。

### 休眠原始中断状态寄存器 (HIBRIS)

基址 0x400F.C000

偏移量 0x018

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												WC	EXTW	LOWBAT	保留	RTCALTO
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:5	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
4	WC	RO	0	<p>写入完成/可以读取原始中断状态</p> <p><b>值 描述</b></p> <p>1 HIBCTL 寄存器中的 WRC 位已经被置位。</p> <p>0 HIBCTL 寄存器中的 WRC 位尚未置位。</p> <p>在 HIBIC 寄存器中的 WC 位写入 1 即可将该位清零。</p>
3	EXTW	RO	0	<p>外部唤醒原始中断状态</p> <p><b>值 描述</b></p> <p>1 <math>\overline{\text{WAKE}}</math> 管脚被置高电平。</p> <p>0 <math>\overline{\text{WAKE}}</math> 管脚没有被置高电平。</p> <p>在 HIBIC 寄存器中的 EXTW 位写入 1 即可将该位清零。</p>
2	LOWBAT	RO	0	<p>电池电压过低原始中断状态</p> <p><b>值 描述</b></p> <p>1 电池电压低于 <math>V_{\text{LOWBAT}}</math>。</p> <p>0 电池电压还没有低于 <math>V_{\text{LOWBAT}}</math>。</p> <p>在 HIBIC 寄存器中的 LOWBAT 位写入 1 即可将该位清零。</p>
1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
0	RTCALTO	RO	0	<p>RTC 警报 0 原始中断状态</p> <p>值 描述</p> <p>1 HIBRTCC 寄存器的值与 HIBRTCM0 寄存器的值匹配，而且 RTCSSC 域的值与 HIBRTCSS 寄存器中 RTCSSM 域的值匹配。</p> <p>0 不匹配</p> <p>在 HIBIC 寄存器中的 RTCALTO 位写入 1 即可将该位清零。</p>



## 寄存器 7: 休眠屏蔽中断状态寄存器 (HIBMIS), 偏移量 0x01C

这个寄存器是休眠模块中断源的可屏蔽中断的状态。该寄存器中的位是 HIBRIS 寄存器和 HIBIM 寄存器中相应的位相与得到的。如果两个相应的位均被置位, 且寄存器中位也被置位, 那么中断信号将会发送到中断控制器。

### 休眠屏蔽中断状态寄存器 (HIBMIS)

基址 0x400F.C000

偏移量 0x01C

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												WC	EXTW	LOWBAT	保留	RTCALTO
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:5	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
4	WC	RO	0	<p>写入完成/可以读取屏蔽中断状态</p> <p>值 描述</p> <p>1 由于 WRC 位被置位, 系统产生了一个没有屏蔽的中断信号。</p> <p>0 WRC 位没有被置位 或者中断被屏蔽。</p> <p>在 HIBIC 寄存器中的 WC 位写入 1 即可将该位清零。</p>
3	EXTW	RO	0	<p>外部唤醒可屏蔽中断状态</p> <p>值 描述</p> <p>1 由于 WAKE 管脚被置高电平, 系统产生了一个没有屏蔽的中断信号。</p> <p>0 外部唤醒中断还没有发生, 或者被屏蔽。</p> <p>在 HIBIC 寄存器中的 EXTW 位写入 1 即可将该位清零。</p>
2	LOWBAT	RO	0	<p>电池电压过低屏蔽中断状态</p> <p>值 描述</p> <p>1 由于电池电压过低, 系统产生了一个没有屏蔽的中断信号。</p> <p>0 电池电压过低中断没有发生, 或者被屏蔽。</p> <p>在 HIBIC 寄存器中的 LOWBAT 位写入 1 即可将该位清零。</p>
1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
0	RTCALTO	RO	0	<p>RTC 警报 0 屏蔽中断状态</p> <p>值 描述</p> <p>1 由于发生 RTC 匹配，系统产生了一个没有屏蔽的中断信号。</p> <p>0 RTC 匹配中断没有发生或者被屏蔽。</p> <p>在 HIBIC 寄存器中的 RTCALTO 位写入 1 即可将该位清零。</p>

## 寄存器 8: 休眠中断清除寄存器 ( HIBIC ) , 偏移量 0x020

该寄存器是休眠模块中断源的中断写入 1 清除寄存器。向 HIBRIS 寄存器中相应的位写入 1 , 相应的中断就会被清除。

### 休眠中断清除寄存器 (HIBIC)

基址 0x400F.C000

偏移量 0x020

类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												WC	EXTW	LOWBAT	保留	RTCALTO
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	RO	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:5	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
4	WC	R/W1C	0	写入完成/可以进行屏蔽中断清除 向该位写入 1 会将 HIBRIS 和 HIBMIS 寄存器中的 WC 位清零。 读取将返回无法确定的值。
3	EXTW	R/W1C	0	外部唤醒屏蔽中断清除 向该位写入 1 会将 HIBRIS 和 HIBMIS 寄存器中的 EXTW 位清零。 读取将返回无法确定的值。
2	LOWBAT	R/W1C	0	电池电压过低屏蔽中断清除 向该位写入 1 会将 HIBRIS 和 HIBMIS 寄存器中的 LOWBAT 位清零。 读取将返回无法确定的值。
1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	RTCALTO	R/W1C	0	RTC 警报 0 屏蔽中断清除 向该位写入 1 会将 HIBRIS 和 HIBMIS 寄存器中的 RTCALTO 位清零。 读取将返回无法确定的值。

注意: 如果 RTC 值和 HIBRTCM0 寄存器/RTCMSS 域值相等, 那么计时器中断源不能被清零。匹配中断的优先级比中断清零的优先级高。

### 寄存器 9: 休眠 RTC 修正寄存器 ( HIBRTCT ) , 偏移量 0x024

该寄存器中包含的数值用来修正 RTC 时钟预分频器。它表示计算得到的下溢数值，该数值用于修正周期。它可以表示为  $0x7FFF \pm N$  时钟周期，其中 N 是每隔 63 秒需要增加或者减少的时钟周期的数量。

**注意：** 休眠模块寄存器处于休眠模块的时钟域内，具有特殊的时序要求。软件应该利用 HIBCTL 寄存器中的 WRC 位来确保所需的时间差是否已经过去。如果 WRC 位被清零，那么任何写入访问都不起作用。请参考“寄存器访问时序”在 477页。

#### 休眠 RTC 修正寄存器 (HIBRTCT)

基址 0x400F.C000  
偏移量 0x024  
类型 R/W, 复位 0x0000.7FFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRIM															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	TRIM	R/W	0x7FFF	RTC 修正值 该数值每隔 64 秒就会加载到 RTC 预分频器中。它用于调整 RTC 的时钟速率，补偿时钟源的漂移和误差。可以利用软件来调整补偿值，即增加或者减小默认值 0x7FFF。增大该数值会降低 RTC 的速率，而减小该数值会增加 RTC 的速率。

## 寄存器 10: 休眠 RTC 亚秒寄存器 ( HIBRTCSS ) , 偏移量 0x028

该寄存器包含 RTC 亚秒计数器和匹配值。要得到 RTC 值，可以首先读取 HIBRTCC 寄存器，再读取 RTCSSC 寄存器的 HIBRTCSS 域，然后再次读取 HIBRTCC 寄存器。如果 HIBRTCC 寄存器两次读取的数值一样，则读取有效。

**注意：** 休眠模块寄存器处于休眠模块的时钟域内，具有特殊的时序要求。软件应该利用 HIBCTL 寄存器中的 WRC 位来确保所需的时间差是否已经过去。如果 WRC 位被清零，那么任何写入访问都不起作用。请参考“寄存器访问时序”在 477 页。

### 休眠 RTC 亚秒寄存器 (HIBRTCSS)

基址 0x400FC000  
偏移量 0x028  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留		RTCSSM													
类型	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		RTCSSC													
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
30:16	RTCSSM	R/W	0x0000	RTC 亚秒匹配 写入活动将该值加载到 RTC 亚秒匹配寄存器中，单位 1/32,768 秒。 读取返回当前匹配值，单位 1/32,768 秒。
15	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
14:0	RTCSSC	RO	0x0000	RTC 亚秒计数 读取返回亚秒计数值，单位 1/32,768 秒。

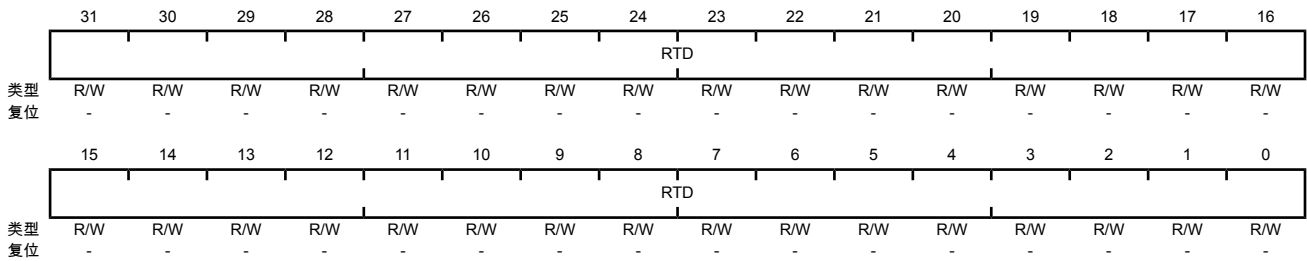
### 寄存器 11: 休眠数据寄存器 ( HIBDATA ) , 偏移量 0x030-0x06F

该地址空间将作为一个 16x32 位的存储空间 ( 64 字节 )。为了存储状态信息，并且只要有电池则在电源关断期间不会掉电，它可以由系统处理器加载。必须将 HIBCTL 寄存器中的 MEMPD 位清零，否则无法访问 HIBDATA 寄存器。

**注意：** 休眠模块寄存器处于休眠模块的时钟域内，具有特殊的时序要求。软件应该利用 HIBCTL 寄存器中的 WRC 位来确保所需的时间差是否已经过去。如果 WRC 位被清零，那么任何写入访问都不起作用。请参考“寄存器访问时序”在 477 页。

#### 休眠数据寄存器 (HIBDATA)

基址 0x400F.C000  
偏移量 0x030-0x06F  
类型 R/W, 复位 -



位/域	名称	类型	复位	描述
31:0	RTD	R/W	-	休眠模块 NV 数据

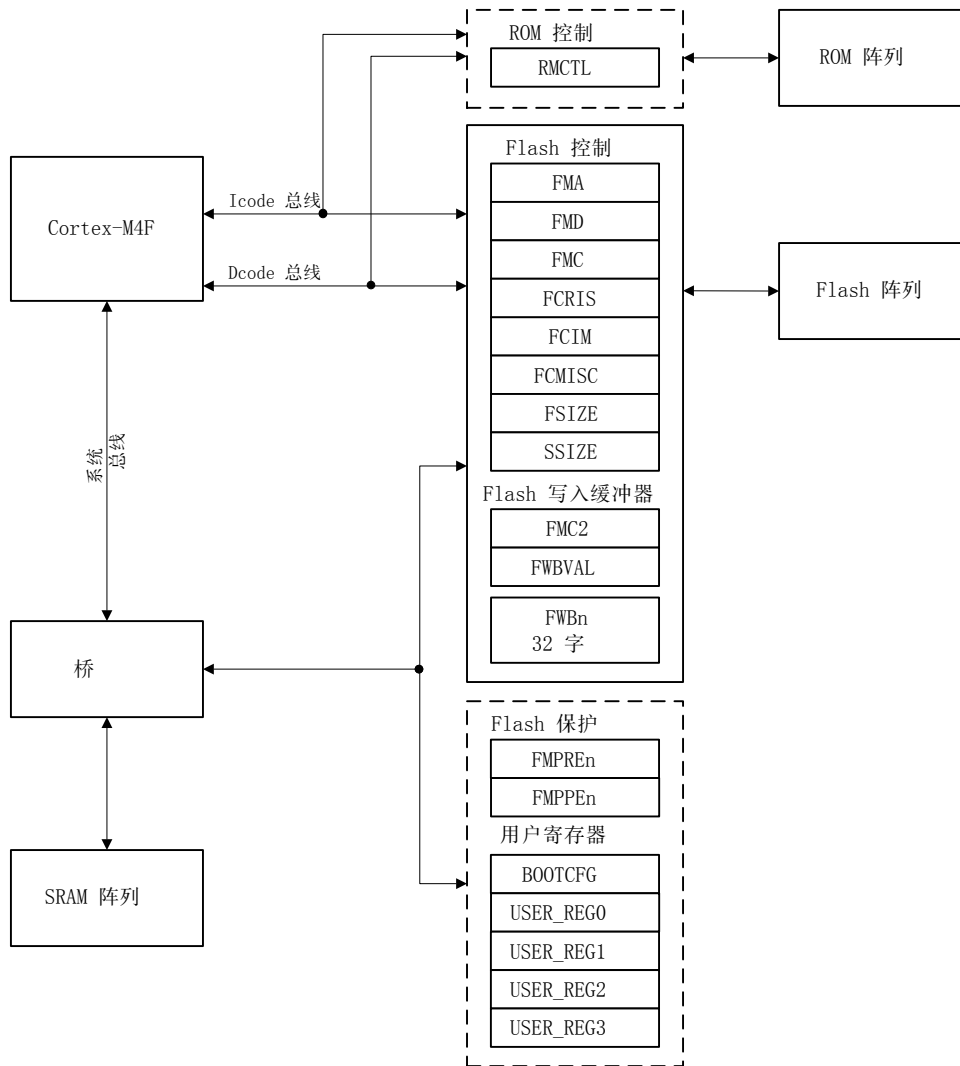
## 8 内部存储器

该 LM4F232H5QD 微控制器带有 32 KB 位带 SRAM、内部 ROM 256 KB Flash 存储器。该 Flash 存储控制器提供一个用户友好的接口，简化了 Flash 编程工作。Flash 存储器由 1 KB 可单独擦除的块构成，且能够以 2 KB 大小的块为单位对其应用存储器保护。

### 8.1 结构图

图 8-1 在 503 页说明了内部 SRAM、ROM 和 Flash 存储器的结构图和控制逻辑。图中的虚线框表示位于系统控制模块中的寄存器。

图 8-1. 内部存储器结构图



### 8.2 功能描述

本节描述了 SRAM、ROM Flash 存储器的功能。

注意：μDMA 控制器可以将数据转移到片上 SRAM，也可以从片上 SRAM 将数据转移出。但是，由于 Flash 存储器和 ROM 位于不同的内部总线，所以 μDMA 不能从 Flash 存储器或 ROM 转移数据。

## 8.2.1 SRAM

该 Stellaris® 器件的内部 SRAM 位于器件存储器映射的地址 0x2000.0000 处。为了减少读 - 修改 - 写 (RMW) 操作的时间，ARM 在处理器中引入了位带 (bit-banding) 技术。在位带启用的处理器中，存储器映射的特定区域 (SRAM 和外设空间) 能够使用地址别名，在单个原子操作中访问各个位。位带基址位于地址 0x2200.0000 处。

使用下面的公式来计算位带别名：

位带别名 = 位带基址 + ( 字节偏移量 \* 32 ) + ( 位编号 \* 4 )

例如，如果要修改地址 0x2000.1000 的第 3 位，位带别名的计算如下：

$0x2200.0000 + (0x1000 * 32) + (3 * 4) = 0x2202.000C$

通过计算得出的位带别名，对地址 0x2202.000C 执行读/写操作的指令可以直接访问地址 0x2000.1000 处字节的第 3 位。

关于位带的详细信息，请参考“位带区”在 96 页。

注意：SRAM 带有两个 32 位宽的 SRAM 存储区 (SRAM 阵列分离)。存储区被这样分开，以便一个包含所有偶数字 (偶存储区)，另一个包含所有奇数字 (奇存储区)。对同一个存储区执行读访问后立即执行写访问，中间会引起一个单时钟周期的停顿。但是对一个存储区执行读访问后对另一个存储区执行写访问，则可以在连续时钟周期内进行而不引起延迟。

## 8.2.2 ROM

该 Stellaris 器件的内部 ROM 位于器件存储器映射的地址 0x0100.0000 处。ROM 详细内容请参考“Stellaris® ROM 用户指南”。

ROM 包含下面几部分：

- Stellaris 引导装载程序和向量表
- Stellaris 为产品特定的外设和接口而发行的外设驱动库 (DriverLib)
- 高级加密标准 (AES) 密码表
- 循环冗余检验 (CRC) 错误检测功能

引导装载程序用作初始化程序的装载器 (当 Flash 存储器为空时)，也可以作为一种应用——初始的固件升级机制 (通过回调引导装载程序)。应用程序可以调用 ROM 中外设驱动库的 API，以减少对 Flash 存储器的需求，释放 Flash 存储器空间用于其它目的 (如应用程序增加的特性)。高级加密标准 (AES) 是美国政府使用的公开定义的加密标准。循环冗余检验 (CRC) 技术可用于确认一个数据块的内容是否与先前检验的相同。

### 8.2.2.1 引导装载程序概述

该 Stellaris 引导加载程序用来将代码下载到设备的 Flash 存储器中，而不需要使用调试接口。内核复位时，通过使用启动配置 (BOOTCFG) 寄存器中配置好的端口 A-H 的 GPIO 信号，用户可以选择让内核直接执行 ROM 的引导加载程序或 Flash 存储器上的应用程序。

在复位时，执行以下操作序列：

1. 读取 BOOTCFG 寄存器。如果 EN 位清零，那么执行 ROM 的引导加载程序。



2. 在 ROM 引导加载程序中，将指定的 GPIO 管脚状态与规定的极性相比较。如果管脚状态与规定的极性匹配，那么 ROM 会映射到地址 0x0000.0000，继续执行 ROM 引导加载程序。
3. 如果 EN 位被置位或者管脚状态与指定的极性不符，那么地址 0x0000.0004 的数据就会被读取，如果该数据是 0xFFFF.FFFF，那么 ROM 被映射到地址 0x0000.0000，继续执行 ROM 引导加载程序。
4. 如果地址 0x0000.0004 的数据不是 0xFFFF.FFFF，堆栈指针 (SP) 装载 Flash 存储器地址 0x0000.0000 的数据，程序计数器 (PC) 装载地址 0x0000.0004 的数据。用户应用程序开始执行。

引导装载程序使用一个简单的包接口与设备进行同步通信。由于引导装载程序不能启用 PLL，所以它的速度由内部振荡器 (PIOSC) 频率决定。下面的串行接口可以使用：

- UART0
- SSI0
- I<sup>2</sup>C0
- USB

UART0、SSI0 和 I2C0 接口的数据格式和通信协议都相同。

注意：引导加载程序的 Flash 存储器驻留版本也支持 CAN。

有关引导加载程序的软件信息可参考“Stellaris® Boot Loader 用户指南”。USB 引导加载程序使用标准的器件固件升级 USB 设备分类。

在 ROM 中使用 UART 引导加载程序的注意事项

在完成自动波特率处理前，U0Tx 不会由 ROM 引导加载程序驱动。如果 U0Tx 在此期间浮动，则所连接的接收器可能发现信号跳变，这会被其 UART 解读为有效字符。为应对这种情况，对 U0Tx 添加上拉或下拉，并为信号提供指定状态，直到 ROM 引导加载程序开始驱动 U0Tx。上拉为更优选择，因为它表示 UART 空闲，而下拉表示中止状态。

### 8.2.2.2 Stellaris 外设驱动库

该 Stellaris 外设驱动库包含一个叫做 `driverlib/rom.h` 的文件，它帮助调用 ROM 中的外设驱动库函数。有关每个函数的详细描述，请参考“Stellaris® ROM 用户指南”。更多关于调用 ROM 函数和使用 `driverlib/rom.h` 的详细信息，请参考“Stellaris® Peripheral Driver Library 用户指南”中的“使用 ROM”一章。同时提供了 `driverlib/rom_map.h` 报头文件，以提高使用不同 Stellaris 器件时的移植性。不同的器件在 ROM 中可能具有不同的 DriverLib 函数子集。`driverlib/rom_map.h` 报头文件使用创建时间标签将函数调用路由到 ROM（如果这些函数在指定器件上可用），否则，它将路由到函数的 Flash 驻留版本。

ROM 起始处的表格指向 ROM 中提供的 API 的入口指针。通过这些表格访问 API 提供了扩展性，虽然 API 的位置可能会在将来的 ROM 版本中改变，可是 API 表格不会变。该表格被分为两级，主表格包含的每个指针对应一个外设，该外设指向二级表格。二级表格包含的每个指针对应一个与外设相关的 API。主表格的位置在 0x0100.0010，恰好在 ROM 中的 Cortex-M4F 向量表后面。

DriverLib 函数在“Stellaris® Peripheral Driver Library 用户指南”中有详细描述。

其他 API 功能可用于图像和 USB 功能，但不会预加载到 ROM。该 Stellaris 图像库提供了一系列基本的图像设置，并提供一个小组件，以在装有 Stellaris 微处理器的有图形显示的板子上建立图形用户接口（更多信息请参考“Stellaris® Graphics Library 用户指南”）。该 Stellaris USB 库是一系

列数据类型和函数，可以在装有 Stellaris 微处理器的板子上建立 USB 设备、主机或 On-The-Go (OTG) 应用（更多信息请参考“Stellaris® USB Library 用户指南”）。

### 8.2.2.3 高级加密标准 (AES) 密码表

AES 是一种强大的加密方法，拥有不错的性能和大小。AES 在硬件和软件方面都很快，它非常容易使用，并且只需要很少的存储空间。AES 对于那些可以预先安排密钥的应用程序来说是理想的，比如加工或配置过程中的设置等。ROM 中提供了执行 XySSL AES 所需要的四个数据表。第一个是正向的 S-box 替换表，第二个是反向的 S-box 替换表，第三个是正向的多项式表，最后一个是反向的多项式表。关于 AES 的更多信息可参考“Stellaris® ROM 用户指南”。

### 8.2.2.4 循环冗余检验 (CRC) 错误检测

CRC 技术可用于确认信息的正确接收（在传送中没有丢失或改变），用来确认解压后的数据，用来证实 Flash 存储器的内容没有更改，以及其它数据需要被确认的情况。CRC 优于简单的校验和（例如异或所有的位），因为它更容易捕捉到变化。关于 CRC 的更多信息可参考“Stellaris® ROM 用户指南”。

## 8.2.3 Flash 存储器

在系统时钟速度为 40 MHz 或以下时，Flash 存储器是单周期读取的。Flash 存储器由一系列 1 KB 的块组织在一起，这些块可以被单独擦除。可以单独对一个 32 位的字进行编程，把当前为 1 的位变为 0。另外，可通过一个写入缓冲区对 Flash 存储器中的 32 个连续字进行编程，所需时间仅是单独对每个字编程的一半。擦除一个块将使块中的所有位变为 1。1-KB 的块可以配对，形成一系列 2-KB 的块，这些 2KB 的块可以被单独保护。该保护允许块被标记为只读或只执行，以提供不同等级的代码保护。只读块不能被擦除或编程，块的内容受保护不能修改。只执行块不能被擦除或编程，只能通过控制器取指机制来读取它的内容，块的内容受保护而不能被控制器或调试器读取。

### 8.2.3.1 预取指缓冲器

Flash 存储器控制器有一个预取指缓冲器，当 CPU 频率大于 40 MHz 时它将自动启用。在此模式下，Flash 存储器在系统时钟一半频率下工作。每个时钟周期，预取指缓冲器获取两个 32 位字，这样在代码线性执行时取指不处于等待状态。取指缓冲器包含一个分支推断机制，它可以辨认出分支从而避免因读取下一对字而增加的额外等待状态。并且，短循环分支经常保持在缓冲器中。因此，一些分支可以不用等待就能执行。其它分支会引起一个单独的等待状态。

### 8.2.3.2 Flash 存储器保护

用户可以在四对 32 位宽存储器中使用两种 Flash 存储器保护形式，以 2-KB Flash 存储器块为单位。由 FMPPEn 和 FMPREn 寄存器的各个位来控制每种形式的保护策略（每个块一种策略）。

- Flash 存储器保护编程启用寄存器 (FMPPEn)：如果某个位被置位，那么就可以对相应的块进行编程（写入）或擦除。如果某个位被清零，那么就不能修改相应的块。
- Flash 存储器保护读取启用寄存器 (FMPREn)：如果某个位被置位，那么软件或者调试器就可以对相应的块执行或读取。如果某个位被清零，那么相应的模块只能被执行，模块的内容禁止被作为数据读取。

这些策略可以进行组合，如表 8-1 在 506 页所示。

表 8-1. Flash 存储器保护策略组合

FMPPEn	FMPREn	保护
0	0	只执行保护模块只能被执行，不能被写入或擦除。这种模式用来保护代码。
1	0	模块可以被写入、擦除或执行，不能被读取。这种组合很少使用。

表 8-1. Flash 存储器保护策略组合 (续)

FMPPEn	FMPREn	保护
0	1	只读保护。模块可以被读取或执行，但不能写入或擦除。这种模式用来锁定模块防止对其进行进一步的修改，但允许对其执行任意的读或执行访问。
1	1	无保护。模块可以被写入、擦除、执行或读取。

对 Flash 存储器的读取保护块 (FMPREn 位被置位) 进行读取访问是被禁止的，否则会产生一次总线故障。对 Flash 存储器的编程保护块 (FMPPEn 位被置位) 进行编程或擦除访问是被禁止的。可以选择产生一个中断 (将 Flash 控制器中断屏蔽 (FCIM) 寄存器中的 AMASK 位置位) 提醒软件开发者在开发和调试阶段可能出现的错误软件操作。注：如果 FMPREn 位已清零，指向 Flash 存储器模块的所有读访问都被禁止，包括任何数据访问。必须注意的是，不得将所要求的数据存储在相关 FMPREn 位已清零的 Flash 存储器模块中。

对所有执行存储块来说，FMPREn 和 FMPPEn 寄存器在出厂时都被设置为 1。这些设置实现了一种开放式的访问策略和可编程策略。寄存器的位可通过清零特定寄存器的位来改变。这种改变立即生效，但不是永久的，等到寄存器被提交 (保存) 以后，位的改变就是永久性的。如果一个位从 1 变为 0 且没有提交，那么它可以通过执行一段上电复位序列来恢复。这些更改需要用 Flash 存储器控制 (FMC) 寄存器来提交。有关这些位的编程详情请参考“非易失性寄存器编程”在 508 页。

### 8.2.3.3 中断

Flash 控制器在检测到下列状态时会产生中断：

- 编程中断 - 当编程或擦除动作完成时发出的信号。
- 访问中断 - 当对受相应 FMPPEn 位保护的 2 KB 块存储器尝试编程或擦除操作时发出的信号。

能够触发控制器级中断的事件在 Flash 控制器可屏蔽中断状态 (FCMIS) 寄存器 (请参考 518 页) 中定义，将相应的 MASK 位置位可以启用该中断。如果不使用中断，原始的中断状态总是在 Flash 控制器原始中断状态 (FCRIS) 寄存器 (请参考 516 页) 中可见。

对 Flash 控制器可屏蔽中断的状态和清除 (FCMISC) 寄存器 (请参考 520 页) 中相应的位写 1 可以清除相应的中断 (适用于 FCMIS 和 FCRIS 寄存器)。

### 8.2.3.4 Flash 存储器编程

该 Stellaris 设备为 Flash 存储器编程提供了一个友好的用户接口。所有的擦除/编程操作都通过 3 个寄存器来处理：Flash 存储器地址寄存器 (FMA)、Flash 存储器数据寄存器 (FMD) 和 Flash 存储器控制寄存器 (FMC)。注意，如果微控制器的调试功能没有激活而处在“锁死”状态，必须执行一段恢复序列来再激活调试模块。请参考“恢复“锁定的”微控制器”在 197 页。

在 Flash 存储器操作 (写、页擦除或整体擦除) 过程中，对它进行访问是禁止的。所以指令和按字取指都将延迟到 Flash 存储器操作完成。如果在 Flash 存储器操作过程中需要执行指令，那么代码必须放置在 SRAM 上并在 SRAM 上执行。

注意：对 Flash 存储器进行编程时，必须考虑存储器的以下特性：

- 只有擦除才能将位从 0 变成 1。
- 写入操作只能将位从 1 变成 0。如果写入操作试图将 0 变成 1，那么该写入操作失败，不会改变任何位的状态。
- Flash 操作可在进入睡眠或深度睡眠模式之前开始 (使用等待中断指令 WFI)，但不能在这两种模式下完成。相反，该操作在事件唤醒系统后完成。这表示您不能依赖 Flash 控制器原始中断状态 (FCRIS) 寄存器的 PRIS 位来真正唤醒睡眠或深度睡眠模式下的器件。

对一个 **32** 位字进行编程

1. 将源数据写入 FMD 寄存器。
2. 将目标地址写入 FMA 寄存器。
3. 将 Flash 写密钥和 WRITE 位的值写入 FMC 寄存器 ( 写入 0xA442.0001 ) 。
4. 轮询 FMC 寄存器，直至 WRITE 位被清零。

执行一个 **1 KB** 页的擦除

1. 将页地址写入 FMA 寄存器。
2. 将 Flash 写密钥和 ERASE 位的值写入 FMC 寄存器 ( 写入 0xA442.0002 ) 。
3. 轮询 FMC 寄存器，直到 ERASE 位被清零，或者用 FCIM 寄存器中的 PMASK 位启用编程中断。

执行一次 **Flash** 存储器的整体擦除

1. 将 Flash 写密钥和 MERASE 位的值写入 FMC 寄存器 ( 写入 0xA442.0004 ) 。
2. 轮询 FMC 寄存器，直到 MERASE 位被清零，或者用 FCIM 寄存器中的 PMASK 位启用编程中断。

#### 8.2.3.5 32 字的 Flash 存储器写入缓冲器

32 字的写入缓冲器可以对两个 32 位字同时编程，从而加快 Flash 存储器的写入访问速度，从而可以在处理 16 字的同时对 32 字进行编程。被缓存的数据写入 Flash 写缓冲器 (FWBn) 寄存器。

这些寄存器在 Flash 存储器中是 32 字对齐的，所以 FWB0 寄存器对应 FMA 中的地址 ( FMA 的 [6:0] 位都是 0 )。FWB1 寄存器对应 FMA + 0x4 中的地址，后面以此类推。只有上次缓存 Flash 存储器写入操作之后更过新的 FWBn 寄存器才能被写入。Flash 写缓冲器有效 (FWBVAL) 寄存器显示了从上次缓存 Flash 存储器写操作之后，哪些寄存器已经被更新。该寄存器包含的位对应 32 个 FWBn 寄存器，其中 FWBVAL 的第 [n] 位对应 FWBn。如果 FWBVAL 寄存器的某位被置位，那么相应的 FWBn 寄存器已经被更新了。

用单次缓冲 **Flash** 存储器写入操作来编程 **32** 个字

1. 将源数据写入 FWBn 寄存器。
2. 将目标地址写入 FMA 寄存器。该地址必须是一个 32 字对齐的地址 ( 即 FMA 的 [6:0] 位必须是 0 ) 。
3. 将 Flash 写入密钥和 WRBUF 位写入 FMC2 寄存器 ( 写入 0xA442.0001 ) 。
4. 轮询 FMC2 寄存器，直到 WRBUF 位被清零，或者等待 PMIS 中断信号发出。

#### 8.2.3.6 非易失性寄存器编程

本节讨论如何更新表 8-2 在 509 页显示的 Flash 存储器自身中的寄存器。这些寄存器驻留在与主 Flash 存储器阵列分离的空间，并且不受擦除或整体擦除的影响。除启动配置 (BOOTCFG) 寄存器外，这些寄存器中的设置可进行写操作，还可校验他们的功能并在提交前读回他们的值，此时他们是非易失性的。如果某个寄存器的一个值尚未提交，那么一个上电复位即可恢复上次提交的值或 ( 如果寄存器从未提交过值 ) 默认值。其他类型的复位不起任何作用。一旦寄存器的内容被提交，唯一能恢复出厂默认值的办法就是执行“恢复“锁定的”微控制器”在 197 页中描述的操作。

要对非易失性寄存器进行写操作：

- 这些寄存器中的位只能由 1 变为 0。
- 对于除 BOOTCFG 寄存器之外的所有寄存器，将数据写入寄存器描述中提供的寄存器地址。对于 BOOTCFG 寄存器，将数据写入 FMD 寄存器。
- 寄存器可进行读操作以校验其内容。要校验存储在 BOOTCFG 寄存器中的内容，需读取 FMD 寄存器。读取 BOOTCFG 寄存器将返回之前提交的值或（如果寄存器从未提交过值）默认值。
- 新值在所有寄存器中立即生效，但 BOOTCFG 寄存器例外，该寄存器的新值在提交前并不会存储在寄存器中。
- 在提交寄存器值之前，一个上电复位将恢复上次提交的值或（如果寄存器从未提交过值）默认值。

要提交新值到非易失性寄存器：

- 如上所述写入数据。
- 将表 8-2 在 509页 显示的值写入 FMA 寄存器。
- 写入 Flash 存储器写入密钥，并将 FMC 寄存器的 COMT 位置位。这些值必须同时写入 FMC 寄存器中。
- 提交非易失性寄存器的时序与常规 Flash 存储器的写操作相同，由表 25-22 在 1305页 中所示的 T<sub>PROG64</sub> 定义。软件可轮询 FMC 寄存器的 COMT 位以确定操作何时完成，或者可将 FCIM 寄存器的 PMASK 位置位以启用中断。
- 提交 BOOTCFG 寄存器时，如果尝试将已提交为 0 的位提交为 1，FCRIS 寄存器的 INVDRIS 位会被置位。
- 一旦提交该值，上电复位将不会对寄存器内容产生任何影响。
- 对 BOOTCFG 寄存器的更改在下次上电复位后生效。
- 一旦将 NW 位更改为 0 并提交，就无法对 BOOTCFG 寄存器进行更多更改。

**重要：** 提交过后，这些寄存器只能通过“恢复“锁定的”微控制器”在 197页 中描述的操作恢复到他们的出厂设置。由该操作引起的主 Flash 存储器阵列的整体擦除操作发生在这些寄存器的恢复操作之前。

表 8-2. 用户可编程的 Flash 存储器驻留寄存器

被提交的寄存器	FMA 值	数据源
FMPRE0	0x0000.0000	FMPRE0
FMPRE1	0x0000.0002	FMPRE1
FMPRE2	0x0000.0004	FMPRE2
FMPRE3	0x0000.0006	FMPRE3
FMPPE0	0x0000.0001	FMPPE0
FMPPE1	0x0000.0003	FMPPE1
FMPPE2	0x0000.0005	FMPPE2
FMPPE3	0x0000.0007	FMPPE3

表 8-2. 用户可编程的 Flash 存储器驻留寄存器 (续)

被提交的寄存器	FMA 值	数据源
USER_REG0	0x8000.0000	USER_REG0
USER_REG1	0x8000.0001	USER_REG1
USER_REG2	0x8000.0002	USER_REG2
USER_REG3	0x8000.0003	USER_REG3
BOOTCFG	0x7510.0000	FMD

### 8.3 寄存器映射

表 8-3 在 510 页列出了 ROM 控制器寄存器和 Flash 存储器以及控制寄存器。表中列出的偏移量是相对于特定存储器控制器基址的 16 进制增量。Flash 存储器寄存器偏移量是相对于 0x400F.D000 的结束地址的 Flash 存储器控制基址而言的。的 EEPROM 基址的。ROM 和 Flash 存储器保护寄存器偏移量是相对于 0x400F.E000 的系统控制基址的。

表 8-3. Flash 寄存器映射

偏移量	名称	类型	复位	描述	见页面
<b>Flash 存储器寄存器 (Flash 控制偏移量)</b>					
0x000	FMA	R/W	0x0000.0000	Flash 存储器地址寄存器	512
0x004	FMD	R/W	0x0000.0000	Flash 存储器数据寄存器	513
0x008	FMC	R/W	0x0000.0000	Flash 存储器控制寄存器	514
0x00C	FCRIS	RO	0x0000.0000	Flash 控制器原始中断状态寄存器	516
0x010	FCIM	R/W	0x0000.0000	Flash 控制器中断屏蔽寄存器	518
0x014	FCMISC	R/W1C	0x0000.0000	Flash 控制器屏蔽中断状态和清除寄存器	520
0x020	FMC2	R/W	0x0000.0000	Flash 存储器控制寄存器 2	522
0x030	FWBVAL	R/W	0x0000.0000	Flash 写缓冲器有效寄存器	523
0x100 - 0x17C	FWBn	R/W	0x0000.0000	Flash 写入缓冲区寄存器 n	524
0xFC0	FSIZE	RO	0x0000.007F	Flash 容量寄存器	525
0xFC4	SSIZE	RO	0x0000.007F	SRAM 大小寄存器	526
0xFCC	ROMSWMAP	RO	0x0000.0000	ROM 软件映射寄存器	527
<b>存储器寄存器 (系统控制偏移量)</b>					
0x0F0	RMCTL	R/W1C	-	ROM 控制寄存器	528
0x130	FMPRE0	R/W	0xFFFF.FFFF	Flash 存储器保护读取启用寄存器 0	529
0x200	FMPRE0	R/W	0xFFFF.FFFF	Flash 存储器保护读取启用寄存器 0	529
0x134	FMPPE0	R/W	0xFFFF.FFFF	Flash 存储器保护编程启用寄存器 0	530
0x400	FMPPE0	R/W	0xFFFF.FFFF	Flash 存储器保护编程启用寄存器 0	530
0x1D0	BOOTCFG	RO	0xFFFF.FFFE	启动配置寄存器	531
0x1E0	USER_REG0	R/W	0xFFFF.FFFF	用户寄存器 0	533

表 8-3. Flash 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x1E4	USER_REG1	R/W	0xFFFF.FFFF	用户寄存器 1	533
0x1E8	USER_REG2	R/W	0xFFFF.FFFF	用户寄存器 2	533
0x1EC	USER_REG3	R/W	0xFFFF.FFFF	用户寄存器 3	533
0x204	FMPRE1	R/W	0xFFFF.FFFF	Flash 存储器保护读取启用寄存器 1	529
0x208	FMPRE2	R/W	0xFFFF.FFFF	Flash 存储器保护读取启用寄存器 2	529
0x20C	FMPRE3	R/W	0xFFFF.FFFF	Flash 存储器保护读取启用寄存器 3	529
0x404	FMPPE1	R/W	0xFFFF.FFFF	Flash 存储器保护编程启用寄存器 1	530
0x408	FMPPE2	R/W	0xFFFF.FFFF	Flash 存储器保护编程启用寄存器 2	530
0x40C	FMPPE3	R/W	0xFFFF.FFFF	Flash 存储器保护编程启用寄存器 3	530

## 8.4 Flash 存储器寄存器描述 (Flash 控制偏移量)

本节按照地址偏移的数字顺序排列和描述 Flash 寄存器。本节的寄存器地址偏移量都是相对于 Flash 控制基址 0x400F.D000 而言的。

## 寄存器 1: Flash 存储器地址寄存器 ( FMA ) , 偏移量 0x000

在单字写操作过程中, 该寄存器含有一个 4 字节对齐的地址并指定在哪里写入数据。在使用写缓冲区进行写入操作时, 此寄存器含有一个 128 字节 ( 32 字 ) 对齐的地址, 用以指定开始写入 32 字节。在擦除操作过程中, 该寄存器含有一个 1 KB 对齐的 CPU 字节地址, 并负责指定哪个块将被擦除。注意必须要符合对齐的要求, 否则操作的结果将不可预知。

### Flash 存储器地址寄存器 (FMA)

基址 0x400F.D000  
偏移量 0x000  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留														OFFSET	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFFSET															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:18	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
17:0	OFFSET	R/W	0x0	地址偏移量。 执行操作的 Flash 存储器地址偏移量, 非易失性寄存器除外 ( 有关该域值的详细内容请参考“非易失性寄存器编程”在 508 页 )。



## 寄存器 2: Flash 存储器数据寄存器 ( FMD ) , 偏移量 0x004

该寄存器中包含的是编程周期中被写入的数据。该寄存器在擦除周期中不使用。

### Flash 存储器数据寄存器 (FMD)

基址 0x400F.D000

偏移量 0x004

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	DATA	R/W	0x0000.0000	数据值 写操作的数据值。

### 寄存器 3: Flash 存储器控制寄存器 ( FMC ) , 偏移量 0x008

当该寄存器被写入时, Flash 存储器控制器将对 Flash 存储器地址 (FMA) 寄存器 ( 请参考 512页 ) 指定的位置启动适当周期数目的访问操作。如果访问是写入操作, 那么 Flash 存储器数据 (FMD) 寄存器 ( 请参考 513页 ) 中的数据将会写入指定地址。

该寄存器是启动存储器操作过程中最后被写入的寄存器。该寄存器低字节的4个控制位用来启动存储器操作。

注意不要设置多个控制位使得操作的结果不可预知。

#### Flash 存储器控制寄存器 (FMC)

基址 0x400F.D000  
偏移量 0x008  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WRKEY															
类型	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												COMT	MERASE	ERASE	WRITE
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	WRKEY	WO	0x0000	Flash存储器写密钥 该域包含一个写密钥, 用于最大限度的减少对Flash的意外写入。如果要对寄存器进行写入操作, 那么必须将 0xA442 写入该域。如果没有这个 WRKEY 值, 那么向 FMC 寄存器中的写入操作将被忽略。读取该域将返回 0。
15:4	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3	COMT	R/W	0	提交寄存器值 该位用于提交对Flash存储器驻留寄存器的写入, 并监视提交过程。

#### 值 描述

- 1 置位该位可以向Flash存储器驻留寄存器提交 ( 写入 ) 寄存器的值。读取该位为1表示之前的提交访问没有完成。
- 0 写 0 对该位状态没有影响。读取该位为0表示之前的提交访问完成。

更多 Flash 存储器驻留寄存器的信息, 请参考“非易失性寄存器编程”在 508页。

位/域	名称	类型	复位	描述
2	MERASE	R/W	0	<p>整体擦除Flash存储器 该位用于整体擦除Flash主存储器，并监视擦除过程。</p> <p>值 描述</p> <p>1 置位该位可以擦除Flash主存储器。 读取该位为 1 表示之前的整体擦除操作没有完成。</p> <p>0 写 0 对该位状态没有影响。 读取该位为 0 表示之前的整体擦除操作完成。</p> <p>要了解擦除时间，请参考“Flash Memory”在 1305页。</p>
1	ERASE	R/W	0	<p>擦除Flash存储器的页 该位用于擦除Flash存储器的页，并监视擦除过程。</p> <p>值 描述</p> <p>1 置位该位可以擦除Flash存储器中由FMA寄存器内容指定的页。 读取该位为 1 表示之前的页擦除操作没有完成。</p> <p>0 写 0 对该位状态没有影响。 读取该位为 0 表示之前的页擦除操作完成。</p> <p>要了解擦除时间，请参考“Flash Memory”在 1305页。</p>
0	WRITE	R/W	0	<p>写一个字到Flash存储器 该位用于写一个字到Flash存储器，并监视写入过程。</p> <p>值 描述</p> <p>1 置位该位可以将FMD寄存器的值存储到Flash存储器中FMA寄存器内容指定的位置。 读取该位为 1 表示写入更新操作没有完成。</p> <p>0 写 0 对该位状态没有影响。 读取该位为 0 表示之前的写入更新操作完成。</p> <p>要了解编程信息，请参考“Flash Memory”在 1305页。</p>

## 寄存器 4: Flash 控制器原始中断状态寄存器 (FCRIS), 偏移量 0x00C

该寄存器指示 Flash 控制器有中断状态。一个中断信号只有其对应的 FCIM 寄存器位被置位才能发送到中断控制器。

### Flash 控制器原始中断状态寄存器 (FCRIS)

基址 0x400F.D000  
偏移量 0x00C  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留		PROGRIS	保留		ERRIS	INVDRIS	VOLTRIS	保留						ERIS	PRIS	ARIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
13	PROGRIS	RO	0	程序验证错误原始中断状态  值 描述 1 因为程序操作失败, 所以会产生一个挂起的中断。当使用 Flash 写入缓冲区时发生错误, 软件必须监测受影响的字, 以确定哪里发生了错误。 0 未产生中断。  将 FCMISC 寄存器中的 PROGMISC 位写 1 可以将该位清零。
12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
11	ERRIS	RO	0	擦除验证错误原始中断状态  值 描述 1 因为擦除操作验证失败, 所以会产生一个挂起的中断。当使用 Flash 写入缓冲区时发生错误, 软件必须监测受影响的字, 以确定哪里发生了错误。 0 未产生中断。  向 FCMISC 寄存器中的 ERMISC 位写 1 可以将该位清零。
10	INVDRIS	RO	0	无效数据原始中断状态  值 描述 1 某个位之前编程为 0, 而现在的操作试图将其编程为 1, 所以会产生一个挂起的中断。 0 未产生中断。  向 FCMISC 寄存器中的 INVMISC 位写 1 可以将该位清零。

位/域	名称	类型	复位	描述
9	VOLTRIS	RO	0	<p>泵激电压原始中断状态</p> <p>值 描述</p> <p>1 在 Flash 操作过程中，泵的调制电压超出范围，操作终止，所以会产生一个挂起的中断。</p> <p>0 未产生中断。</p> <p>向 FCMISC 寄存器中的 VOLTMISC 位写 1 可以将该位清零。</p>
8:3	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
2	ERIS	RO	0	<p>EEPROM 原始中断状态</p> <p>该位提供 EEPROM 操作状态。</p> <p>值 描述</p> <p>1 发生了 EEPROM 中断。</p> <p>0 未产生 EEPROM 中断。</p> <p>向 FCMISC 寄存器中的 EMISC 位写 1 可以将该位清零。</p>
1	PRIS	RO	0	<p>编程的原始中断状态</p> <p>该位给出编程周期的状态，编程周期是指通过 FMC 或者 FMC2 寄存器位（请参考 514页 和 522页）产生的写入或擦除操作。</p> <p>值 描述</p> <p>1 编程周期或者擦除周期已经完成。</p> <p>0 编程周期或者擦除周期没有完成。</p> <p>当 FCIM 寄存器的 PMASK 位置位时，该位状态会发送给中断控制器。清零该位可通过写 1 到 FCMISC 寄存器的 PMISC 位来实现。</p>
0	ARIS	RO	0	<p>访问原始中断状态</p> <p>值 描述</p> <p>1 试图对存储器块进行的编程或擦除行为与 FMPPEn 寄存器设定的保护策略相矛盾。</p> <p>0 没有不当的编程或擦除试图访问存储器。</p> <p>当 FCIM 寄存器的 AMASK 位置位时，该位状态会发送给中断控制器。清零该位可通过写 1 到 FCMISC 寄存器的 AMISC 位来实现。</p>

## 寄存器 5: Flash 控制器中断屏蔽寄存器 ( FCIM ) , 偏移量 0x010

该寄存器控制 Flash 控制器是否向控制器产生中断。

### Flash 控制器中断屏蔽寄存器 (FCIM)

基址 0x400F.D000  
偏移量 0x010  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		PROGMASK	保留		ERMASK	INVDMASK	VOLTMASK	保留					EMASK	PMASK	AMASK
类型	RO	RO	R/W	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
13	PROGMASK	R/W	0	PROGVER 中断屏蔽  值 描述 1 当 PROGRIS 位置位时, 向中断控制器发送一个中断。 0 PROGVER 中断被抑制, 不会发送到中断控制器。
12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
11	ERMASK	R/W	0	ERVER 中断屏蔽  值 描述 1 当 ERRIS 位置位时, 向中断控制器发送一个中断。 0 ERRIS 中断被抑制, 不会发送到中断控制器。
10	INVDMASK	R/W	0	无效数据中断屏蔽  值 描述 1 当 INVDRIS 位置位时, 向中断控制器发送一个中断。 0 INVDRIS 中断被抑制, 不会发送到中断控制器。
9	VOLTMASK	R/W	0	VOLT 中断屏蔽  值 描述 1 当 VOLTRIS 位置位时, 向中断控制器发送一个中断。 0 VOLTRIS 中断被抑制, 不会发送到中断控制器。
8:3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
2	EMASK	R/W	0	EEPROM 中断屏蔽  值 描述 1 当 ERIS 位置位时，向中断控制器发送一个中断。 0 ERIS 中断被抑制，不会发送到中断控制器。
1	PMASK	R/W	0	编程中断屏蔽 该位控制着编程原始中断状态向中断控制器的报告。  值 描述 1 当 PRIS 位置位时，向中断控制器发送一个中断。 0 PRIS 中断被抑制，不会发送到中断控制器。
0	AMASK	R/W	0	访问中断屏蔽 该位控制着访问原始中断状态向中断控制器的报告。  值 描述 1 当 ARIS 位置位时，向中断控制器发送一个中断。 0 ARIS 中断被抑制，不会发送到中断控制器。

## 寄存器 6: Flash 控制器屏蔽中断状态和清除寄存器 ( FCMISC ) , 偏移量 0x014

该寄存器提供两个功能。首先, 它可以通过指示哪个中断源或哪些中断源正在发出中断信号来报告中断产生的原因。其次, 它提供清除中断报告的办法。

### Flash 控制器屏蔽中断状态和清除寄存器 (FCMISC)

基址 0x400F.D000  
偏移量 0x014  
类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留		PROGMISC	保留		ERMISC	INVMISC	VOLTMISC	保留						EMISC	PMISC	AMISC
类型	RO	RO	R/W1C	RO	R/W1C	R/W1C	R/W1C	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
13	PROGMISC	R/W1C	0	PROGVER 屏蔽中断状态和清除  值 描述 1 读取该位为 1 表示发出了非屏蔽中断信号。 对这个位写 1 清零 PROGMISC 位以及 FCRIS 寄存器的 PROGRIS 位 ( 参考 516页 )。 0 读取该位为 0 表示中断没有发生。 写 0 对该位状态没有影响。
12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
11	ERMISC	R/W1C	0	ERVER 屏蔽中断状态和清除  值 描述 1 读取该位为 1 表示发出了非屏蔽中断信号。 对这个位写 1 清零 ERMISC 位以及 FCRIS 寄存器的 ERRIS 位 ( 参考 516页 )。 0 读取该位为 0 表示中断没有发生。 写 0 对该位状态没有影响。
10	INVMISC	R/W1C	0	无效数据屏蔽中断状态和清除  值 描述 1 读取该位为 1 表示发出了非屏蔽中断信号。 对这个位写 1 清零 INVMISC 位以及 FCRIS 寄存器的 INVDRIS 位 ( 参考 516页 )。 0 读取该位为 0 表示中断没有发生。 写 0 对该位状态没有影响。



位/域	名称	类型	复位	描述
9	VOLTMISC	R/W1C	0	<p>VOLT 屏蔽中断状态和清除</p> <p>值 描述</p> <p>1 读取该位为 1 表示发出了非屏蔽中断信号。 对这个位写 1 清零 VOLTMISC 位以及 FCRIS 寄存器的 VOLTRIS 位 (参考 516页)。</p> <p>0 读取该位为 0 表示中断没有发生。 写 0 对该位状态没有影响。</p>
8:3	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件,保留位的值在读-修改-写操作过程中应该保持不变。</p>
2	EMISC	R/W1C	0	<p>EEPROM 屏蔽中断状态和清除</p> <p>值 描述</p> <p>1 读取该位为 1 表示发出了非屏蔽中断信号。 对这个位写 1 清零 EMISC 位以及 FCRIS 寄存器的 ERIS 位 (参考 516页)。</p> <p>0 读取该位为 0 表示中断没有发生。 写 0 对该位状态没有影响。</p>
1	PMISC	R/W1C	0	<p>编程屏蔽中断状态和清除</p> <p>值 描述</p> <p>1 当读取该位为 1 时,表示一个非屏蔽中断信号被发出,原因是一个编程周期完成。 对这个位写 1 清零 PMISC 位以及 FCRIS 寄存器的 PRIS 位 (参考 516页)。</p> <p>0 当读取该位为 0 时,表示没有发生编程周期完成中断。 写 0 对该位状态没有影响。</p>
0	AMISC	R/W1C	0	<p>访问屏蔽中断状态和清除</p> <p>值 描述</p> <p>1 当读取该位为 1 时,表示一个非屏蔽中断信号被发出,原因是试图对存储器块进行的编程或擦除行为与该存储器块由 FMPPEn 寄存器置位的保护策略相矛盾。 对这个位写 1 清零 AMISC 位以及 FCRIS 寄存器的 ARIS 位 (参考 516页)。</p> <p>0 读取该位为 0 表示不适当中断没有发生。 写 0 对该位状态没有影响。</p>

## 寄存器 7: Flash 存储器控制寄存器 2 ( FMC2 ) , 偏移量 0x020

当该寄存器被写入时, Flash 存储器控制器将对 Flash 存储器地址 (FMA) 寄存器 ( 请参考 512页 ) 指定的位置启动适当周期数目的访问操作。如果访问是写操作, 那么 Flash 写缓冲器 (FWB) 寄存器中的数据将会写入指定地址。

该寄存器是启动存储器操作过程中最后被写入的寄存器。

### Flash 存储器控制寄存器 2 (FMC2)

基址 0x400F.D000  
偏移量 0x020  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WRKEY															
类型	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															WRBUF
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	WRKEY	WO	0x0000	Flash存储器写密钥 该域包含一个写密钥, 用于最大限度的减少对Flash的意外写入。必须将值 0xA442 写入该域才能进行写入操作。如果没有这个 WRKEY 值, 那么向 FMC2 寄存器中的写入操作将被忽略。读取该域将返回 0。
15:1	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	WRBUF	R/W	0	缓冲的 Flash 存储器写入 被缓存的Flash存储器写入

#### 值 描述

- 置位该位将会把 FWBn 寄存器的数据存储到 FMA 寄存器内容所指定的位置。  
当读取该位为1时, 表示之前的被缓存的Flash存储器写访问没有完成。
- 写 0 对该位状态没有影响。  
当读取该位为0时, 表示之前的被缓存的Flash存储器写访问完成。

要了解编程信息, 请参考“Flash Memory”在 1305页。

## 寄存器 8: Flash 写缓冲器有效寄存器 (FWBVAL), 偏移量 0x030

该寄存器的位显示了自从上次 Flash 存储器写缓冲器的写操作后, 哪个 FWBn 寄存器又被处理器写入过。值为 1 的寄存器会在下一次 Flash 存储器写缓冲器写操作中被写入。该寄存器在写操作后被硬件清零。写操作中违反保护也会清该状态。

在被写操作清零之后, 软件可以通过将 FWB[n] 位置位来将同样的 32 个字编程到不同的 Flash 存储器位置。下一次写入操作将使用与上次写操作相同的数据。另外, 如果一个 FWBn 寄存器的改变不应该写入 Flash 存储器, 当下一次写操作发生时, 软件可以清零相应的 FWB[n] 位来保留当前的数据。

### Flash 写缓冲器有效寄存器 (FWBVAL)

基址 0x400F.D000  
偏移量 0x030  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FWB[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FWB[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	FWB[n]	R/W	0x0	Flash 存储器写缓冲器

#### 值 描述

- 1 相应的 FWBn 寄存器已经在上次缓冲器写操作之后更新, 并准备好写入 Flash 存储器。
- 0 相应的 FWBn 寄存器没有被写入新的数据。

第 0 位对应 FWB0, 偏移量 0x100, 第 31 位对应 FWB31, 偏移量 0x13C。

## 寄存器 9: Flash 写入缓冲区寄存器 n (FWBn) , 偏移量 0x100 - 0x17C

这些 32 个寄存器包含了通过缓存 Flash 存储器写入操作写入 Flash 存储器的数据。偏移量代表某个 32 位寄存器的地址。只有在之前的缓存 Flash 存储器写操作后被更新的 FWBn 寄存器才可以写入 Flash 存储器。所以没有必要为了写 1 个或 2 个字而写满所有的寄存器存储区。FWBn 寄存器写入 Flash 存储器时，FWB0 寄存器对应 FMA 中包含的地址。FWB1 写入地址 FMA+0x4，以此类推。注意只有数据的位为 0 才会导致 Flash 存储器被修改。数据位为 1 将保持 Flash 存储器位的内容为它之前的值。

### Flash 写入缓冲区寄存器 n (FWBn)

基址 0x400F.D000  
偏移量 0x100 - 0x17C  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	DATA	R/W	0x0000.0000	数据 将要写入Flash存储器的数据。

## 寄存器 10: Flash 容量寄存器 ( FSIZE ) , 偏移量 0xFC0

该寄存器表示片内 Flash 存储器的容量大小。

**重要:** 该寄存器应该用来决定微控制器处理的 Flash 存储器容量大小。但是, 要支持传统软件, 可使用 DC0 寄存器。读取 DC0 寄存器能够了解传统存储器的准确容量。软件必须使用 FSIZE 寄存器来确定没有在 DC0 寄存器中列出的存储器的大小。

### Flash 容量寄存器 (FSIZE)

基址 0x400F.D000  
偏移量 0xFC0  
类型 RO, 复位 0x0000.007F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SIZE															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	SIZE	RO	0x7F	Flash 大小 表示片内 Flash 存储器的大小。
	值	描述		
	0x0003	8KB Flash		
	0x0007	16 KB Flash		
	0x000F	32 KB Flash		
	0x001F	64 KB Flash		
	0x002F	96 KB Flash		
	0x003F	128 KB Flash		
	0x005F	192 KB Flash		
	0x007F	256 KB Flash		

## 寄存器 11: SRAM 大小寄存器 (SSIZE), 偏移量 0xFC4

这个寄存器用来指示片内 SRAM 的大小。

**重要:** 该寄存器应该用来决定微控制器处理的 SRAM 容量大小。但是, 要支持传统软件, 可使用 DC0 寄存器。读取 DC0 寄存器能够了解传统存储器的准确容量。软件必须使用 SSIZE 寄存器来确定没有在 DC0 寄存器中列出的存储器的大小。

### SRAM 大小寄存器 (SSIZE)

基址 0x400F.D000  
偏移量 0xFC4  
类型 RO, 复位 0x0000.007F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SIZE															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	SIZE	RO	0x7F	SRAM 大小 显示了片上 SRAM 的大小。
	值	描述		
	0x0007	2KB SRAM		
	0x000F	4 KB SRAM		
	0x0017	6 KB SRAM		
	0x001F	8 KB SRAM		
	0x002F	12 KB SRAM		
	0x003F	16 KB SRAM		
	0x004F	20 KB SRAM		
	0x005F	24 KB SRAM		
	0x007F	32 KB SRAM		

## 寄存器 12: ROM 软件映射寄存器 ( ROMSWMAP ) , 偏移量 0xFCC

该寄存器用来指示片内 ROM 中第三方软件的存在。

**重要:** 该寄存器用来确定微控制器中片内 ROM 中是否存在第三方软件。然而, 为了支持传统软件, 可以使用 NVMSTAT 寄存器。读取 NVMSTAT 寄存器中的 TPSW 位可以正确地识别是否存在传统第三方软件。软件应针对不在传统器件上的软件使用 ROMTPSW 寄存器。

### ROM 软件映射寄存器 (ROMSWMAP)

基址 0x400F.D000  
偏移量 0xFCC  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	SAFERTOS	RO	0x0	SafeRTOS 存在
				值 描述
				0 SafeRTOS 没有在片内 ROM 中。
				1 SafeRTOS 在片内 ROM 中。

## 8.5 存储器寄存器描述 ( 系统控制偏移量 )

本节的剩余部分按照地址偏移的数字顺序列出并描述了系统控制地址空间中的寄存器。本节中的寄存器地址偏移量都是相对于系统控制基址 0x400F.E000 而言的。

### 寄存器 13: ROM 控制寄存器 (RMCTL) , 偏移量 0x0F0

该寄存器提供了对 ROM 控制器状态的控制。该寄存器的偏移量是相对于系统控制基址 0x400F.E000 而言的。

在复位时，执行以下操作序列：

1. 读取 BOOTCFG 寄存器。如果 EN 位清零，那么执行 ROM 的引导加载程序。
2. 在 ROM 引导加载程序中，将指定的 GPIO 管脚状态与规定的极性相比较。如果管脚状态与规定的极性匹配，那么 ROM 会映射到地址 0x0000.0000，继续执行 ROM 引导装载程序。
3. 如果 EN 位被置位或者管脚状态与指定的极性不符，那么地址 0x0000.0004 的数据就会被读取，如果该数据是 0xFFFF.FFFF，那么 ROM 被映射到地址 0x0000.0000，继续执行 ROM 引导加载程序。
4. 如果地址 0x0000.0004 的数据不是 0xFFFF.FFFF，堆栈指针 (SP) 装载 Flash 存储器地址 0x0000.0000 的数据，程序计数器 (PC) 装载地址 0x0000.0004 的数据。用户应用程序开始执行。

#### ROM 控制寄存器 (RMCTL)

基址 0x400F.E000  
偏移量 0x0F0  
类型 R/W1C, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
	BA															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	BA	R/W1C	1	启动别名  值 描述 1 微控制器的 ROM 出现在地址 0x0。 0 Flash 存储器在地址 0x0。  对该位写入 1 可以清零该位。



**寄存器 14: Flash 存储器保护读取启用寄存器 0 ( FMPRE0 )** , 偏移量 **0x130** 和 **0x200**

**寄存器 15: Flash 存储器保护读取启用寄存器 1 ( FMPRE1 )** , 偏移量 **0x204**

**寄存器 16: Flash 存储器保护读取启用寄存器 2 ( FMPRE2 )** , 偏移量 **0x208**

**寄存器 17: Flash 存储器保护读取启用寄存器 3 ( FMPRE3 )** , 偏移量 **0x20C**

注意: FMPRE0 寄存器采用别名, 以实现向后兼容。

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

这个寄存器存放的是每个 2 KB 块的只读保护位 ( FMPPE<sub>n</sub> 存放只执行位 )。

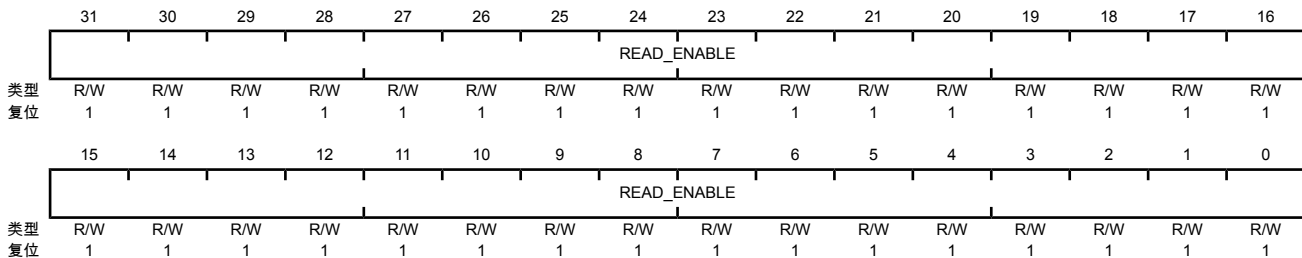
该寄存器在上电复位序列时加载。对所有已执行的 2 KB 模块来说, FMPRE<sub>n</sub> 和 FMPPE<sub>n</sub> 寄存器在出厂时都被设置为 1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是 R/W0, 用户只能将保护位从 1 变为 0 (不可以从 0 变为 1)。这种改变不是永久的, 等到寄存器被提交 (保存) 以后, 位的改变就是永久性的。如果一个位从 1 变为 0 且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复“锁定的”微控制器”在 197 页中详述的序列。

每个 FMPRE<sub>n</sub> 寄存器控制一个 64 k 的 Flash 块。更多信息请参考“Flash 存储器保护”一节。

- FMPRE0 : 0 到 64 KB
- FMPRE1 : 65 到 128 KB
- FMPRE2 : 129 到 192 KB
- FMPRE3 : 193 到 256 KB

#### Flash 存储器保护读取启用寄存器 n ( FMPRE<sub>n</sub> )

基址 0x400F.E000  
偏移量 0x130 和 0x200  
类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	READ_ENABLE	R/W	0xFFFF.FFFF	Flash 读取启用 每个位可以把 2 KB 的 Flash 块配置成只读。 这些策略可以组合, 如表“Flash 保护策略组合”所示。

**寄存器 18: Flash 存储器保护编程启用寄存器 0 ( FMPPE0 )** , 偏移量 **0x134** 和 **0x400**

**寄存器 19: Flash 存储器保护编程启用寄存器 1 ( FMPPE1 )** , 偏移量 **0x404**

**寄存器 20: Flash 存储器保护编程启用寄存器 2 ( FMPPE2 )** , 偏移量 **0x408**

**寄存器 21: Flash 存储器保护编程启用寄存器 3 ( FMPPE3 )** , 偏移量 **0x40C**

注意: FMPPE0 寄存器采用别名, 以实现向后兼容。

注意: 偏移量是相对于系统控制基址 0x400FE000 的。

该寄存器存储的是 2KB Flash 块的只执行保护位 ( FMPREn 存放只读保护位 )。

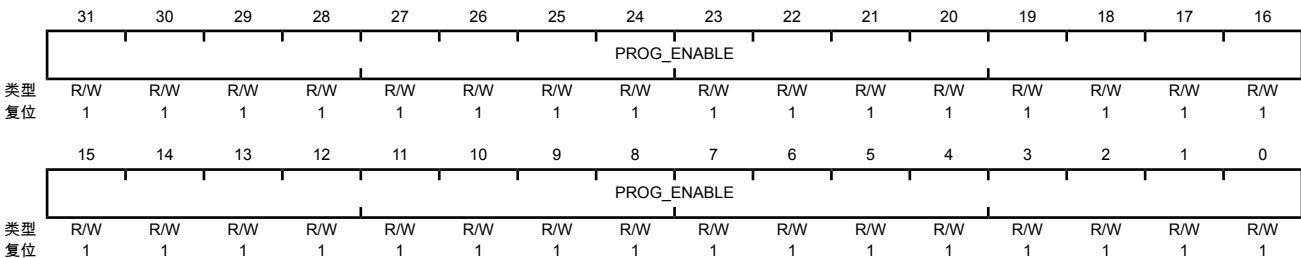
该寄存器在上电复位序列时加载。对所有执行存储块来说, FMPREn 和 FMPPEn 寄存器在出厂时都被设置为 1。这样可以实现一种开放的访问和编程策略。可以通过写特定的寄存器位来更改这个寄存器的位。但是, 这个寄存器是 R/W0, 用户只能将保护位从 1 变为 0 (不可以从 0 变为 1)。这种改变不是永久的, 等到寄存器被提交 (保存) 以后, 位的改变就是永久性的。如果一个位从 1 变为 0 且没有提交, 那么它可以通过执行一段上电复位序列来恢复。显示的复位值只适用于上电复位, 任何其它类型的复位都不会影响这个寄存器。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复“锁定的”微控制器”在 197 页中详述的序列。更多信息请参考“Flash 存储器保护”一节。

每个 FMPPEn 寄存器控制一个 64 k 的 Flash 块。更多信息请参考“Flash 存储器保护”一节。

- FMPPE0 : 0 到 64 KB
- FMPPE1 : 65 到 128 KB
- FMPPE2 : 129 到 192 KB
- FMPPE3 : 193 到 256 KB

Flash 存储器保护编程启用寄存器 n ( FMPPEn )

基址 0x400FE000  
 偏移量 0x134 和 0x400  
 类型 R/W, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	PROG_ENABLE	R/W	0xFFFF.FFFF	Flash 编程启用 每个位可以把 2 KB 的 Flash 块配置成只执行。 这些策略可以组合, 如表“Flash 保护策略组合”所示。

## 寄存器 22: 启动配置寄存器 (BOOTCFG), 偏移量 0x1D0

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

此寄存器无法直接写入, 而应通过 FMD 寄存器写入, 如“非易失性寄存器编程”在 508 页所述。该寄存器提供了对 GPIO 管脚的配置, 以启用 ROM 引导加载程序, 还提供了一次性写入机制, 以禁止外部调试器访问设备。复位时, 通过使用在该寄存器中配置的 A-Q 端口 GPIO 信号, 用户可以选择让内核直接执行 ROM 引导加载程序或执行 Flash 存储器中的应用程序。在复位时, 执行以下操作序列:

1. 读取 BOOTCFG 寄存器。如果 EN 位清零, 那么执行 ROM 的引导加载程序。
2. 在 ROM 引导加载程序中, 将指定的 GPIO 管脚状态与规定的极性相比较。如果管脚状态与规定的极性匹配, 那么 ROM 会映射到地址 0x0000.0000, 继续执行 ROM 引导装载程序。
3. 如果 EN 位被置位或者管脚状态与指定的极性不符, 那么地址 0x0000.0004 的数据就会被读取, 如果该数据是 0xFFFF.FFFF, 那么 ROM 被映射到地址 0x0000.0000, 继续执行 ROM 引导装载程序。
4. 如果地址 0x0000.0004 的数据不是 0xFFFF.FFFF, 堆栈指针 (SP) 装载 Flash 存储器地址 0x0000.0000 的数据, 程序计数器 (PC) 装载地址 0x0000.0004 的数据。用户应用程序开始执行。

出厂时 DBG0 位设为 0, DBG1 设为 1, 默认启用外部调试器。将 DBG1 位清零后, 从下一次设备上电周期开始, 任何外部调试器都将无法访问设备。NW 位表示可将寄存器的位从 1 改为 0。可通过软件清零 NW 位, 从而避免对此寄存器的位进行更多更改。

在上电序列后, 可以用 FMC 寄存器的 COMT 位来提交确认寄存器的值, 然后寄存器的值变为非易失性保存下来。在提交之前, 这些寄存器中的位只能由 1 变为 0。寄存器尚未提交时, 显示的复位值只适用于上电复位; 任何其它类型的复位都不会影响这个寄存器。一旦提交, 该寄存器在上电复位时保留自身的值。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复“锁定的”微控制器”在 197 页中详述的序列。

### 启动配置寄存器 (BOOTCFG)

基址 0x400F.E000  
偏移量 0x1D0  
类型 RO, 复位 0xFFFF.FFFE

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NW	保留														
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PORT			PIN		POL		EN	保留						DBG1	DBG0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

位/域	名称	类型	复位	描述
31	NW	RO	1	没有写入 置位时, 此位表示可将寄存器的值从 1 改为 0。清零时, 此位规定不能更改该寄存器的内容。
30:16	保留	RO	0xFFFF	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
15:13	PORT	RO	0x7	<p>启动 GPIO 端口</p> <p>该字段选择复位时启用 ROM 引导装载程序的 GPIO 端口。</p> <p>值 描述</p> <p>0x0 端口 A</p> <p>0x1 端口 B</p> <p>0x2 端口 C</p> <p>0x3 端口 D</p> <p>0x4 端口 E</p> <p>0x5 端口 F</p> <p>0x6 端口 G</p> <p>0x7 端口 H</p>
12:10	PIN	RO	0x7	<p>启动GPIO管脚</p> <p>该字段选择复位时启用 ROM 引导装载程序的 GPIO 端口管脚编号。</p> <p>值 描述</p> <p>0x0 管脚 0</p> <p>0x1 管脚 1</p> <p>0x2 管脚 2</p> <p>0x3 管脚 3</p> <p>0x4 管脚 4</p> <p>0x5 管脚 5</p> <p>0x6 管脚 6</p> <p>0x7 管脚 7</p>
9	POL	RO	1	<p>启动 GPIO 极性</p> <p>若该位为 1，那么在复位时 GPIO 管脚的高电平启用 ROM 引导装载程序。若该位为 0，那么在复位时 GPIO 端口管脚的low电平启用 ROM 引导装载程序。</p>
8	EN	RO	1	<p>启动 GPIO 启用</p> <p>清零该位将启用GPIO管脚在复位时用于启用ROM引导装载程序。若该位被置位时，地址 0x0000.0004 的内容将被检测，来查看 Flash 存储器是否被编程。如果内容不是 0xFFFF.FFFF，内核会从 Flash 存储器执行。如果 Flash 没有被编程，内核会从 ROM 执行。</p>
7:2	保留	RO	0x3F	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
1	DBG1	RO	1	<p>调试控制 1</p> <p>DBG1 位必须为 1 且 DBG0 必须为 0，才可以进行调试。</p>
0	DBG0	RO	0	<p>调试控制 0</p> <p>DBG1 位必须为 1 且 DBG0 必须为 0，才可以进行调试。</p>

寄存器 23: 用户寄存器 0 ( USER\_REG0 ) , 偏移量 0x1E0

寄存器 24: 用户寄存器 1 ( USER\_REG1 ) , 偏移量 0x1E4

寄存器 25: 用户寄存器 2 ( USER\_REG2 ) , 偏移量 0x1E8

寄存器 26: 用户寄存器 3 ( USER\_REG3 ) , 偏移量 0x1EC

注意: 偏移量是相对于系统控制基址 0x400F.E000 的。

每个此类寄存器都可提供 32 位用户定义的非易失性数据。这些寄存器中的位只能由 1 变为 0。寄存器尚未提交时, 显示的复位值只适用于上电复位; 任何其它类型的复位都不会影响这个寄存器。一旦提交, 该寄存器在上电复位时保留自身的值。一旦提交, 恢复该寄存器出厂默认值的唯一方法是执行“恢复“锁定的”微控制器”在 197 页中详述的序列。

#### 用户寄存器 n ( USER\_REGn )

基址 0x400F.E000

偏移量 0x1E0

类型 R/W, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	DATA	R/W	0xFFFF.FFFF	用户数据 包含用户数据值。该域的位全部初始化为 1, 且提交后, 该域将通过上电复位保留自身的值。

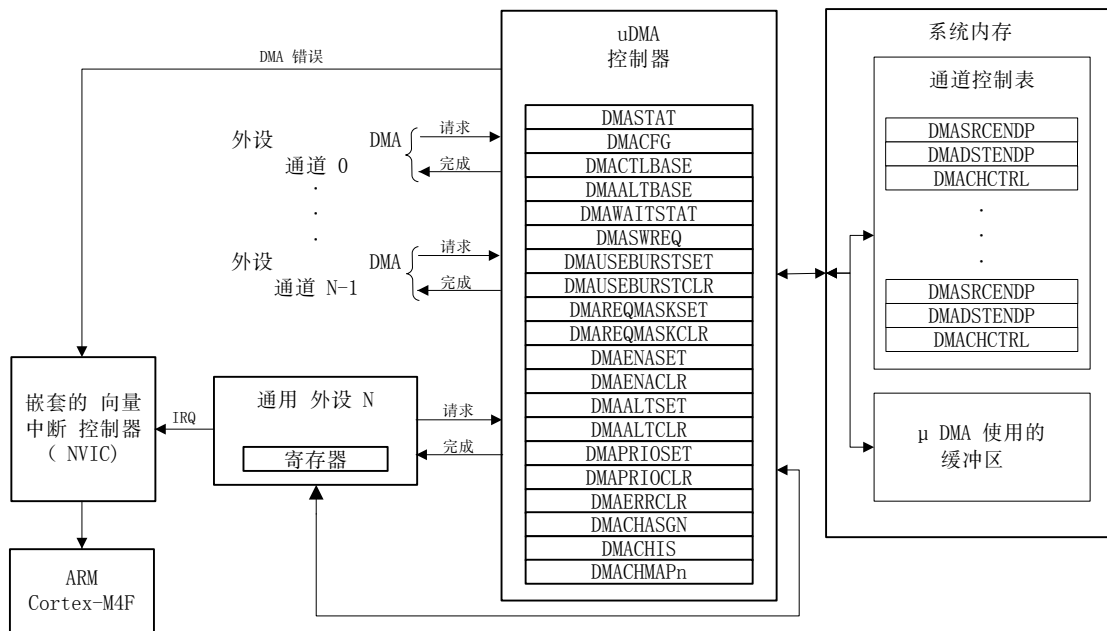
## 9 微型直接存储器访问 ( $\mu$ DMA)

LM4F232H5QD 微控制器内置一个直接存储器访问 ( Direct Memory Access , 简称为DMA ) 控制器, 我们称之为微型 DMA ( $\mu$ DMA) 控制器。 $\mu$ DMA 控制器所提供的工作方式能够分载 Cortex™-M4F 处理器参与的数据传输任务, 从而更加高效地使用内核以及总线带宽。 $\mu$ DMA 控制器能够自动执行存储器与外设之间的数据传输。片上每个支持  $\mu$ DMA 功能的外设都有专用的  $\mu$ DMA 通道, 通过合理的编程配置, 当外设需要时能够自动在外设和存储器之间传输数据。 $\mu$ DMA 控制器具有以下特性:

- ARM® PrimeCell® 32 通道的可配置  $\mu$ DMA 控制器;
- 支持存储器到存储器、存储器到外设、外设到存储器的  $\mu$ DMA 传输, 包括:
  - 基本模式, 用于简单的传输需求
  - 乒乓模式, 用于实现持续数据流
  - 散聚模式, 借助一个可编程的任务列表, 由单个请求触发一连串的指定传输
- 高度灵活的可配置的通道配置;
  - 各通道均可独立配置、独立操作
  - 每个支持  $\mu$ DMA 功能的片上模块都有其专用通道
  - 灵活的通道分配
  - 对于双向模块, 为其接收和发送各提供一个通道
  - 专用的软件通道, 可由软件启动  $\mu$ DMA 传输
  - 每通道都可分别配置优先级
  - 可选配置: 任一通道均可用作软件启动传输
- 优先级分为两级;
- 通过优化设计, 改进了  $\mu$ DMA 控制器与处理器内核之间的总线访问性能:
  - 当内核不访问总线时,  $\mu$ DMA 控制器即可占用总线
  - RAM条带处理
  - 外设总线分段
- 支持 8 位、16 位或 32 位数据宽度
- 待传输数目可编程为 2 的整数幂, 有效范围 1 到 1024
- 源地址及目的地址可自动递增, 递增单位可以是字节、半字、字、不递增
- 可屏蔽的外设请求
- 传输结束中断, 且每个通道有独立的中断

## 9.1 结构图

图 9-1.  $\mu$ DMA 结构图



## 9.2 功能描述

**功能描述**  $\mu$ DMA 控制器是一种使用方便、配置灵活的 DMA 控制器，用于同微控制器 Cortex-M4F 处理器内核配合以实现高效工作。 $\mu$ DMA 控制器支持多种数据宽度以及地址递增机制，各 DMA 通道之间具有不同的优先级，还提供了多种传输模式，能够通过预编程实现十分复杂的自动传输流程。 $\mu$ DMA 控制器对总线的占用总是次于处理器内核，因此绝不影响处理器的总线会话。由于  $\mu$ DMA 控制器只会在总线空闲时占用总线，因此它提供的数据传输带宽非常独立，不会影响系统其它部分的正常运行。此外总线架构还经过了优化，增强了处理器内核与  $\mu$ DMA 控制器高效共享片上总线的性能，从而大大提高了性能。优化的内容包括 RAM 条带处理以及外设总线分段，在大多数情况下允许处理器内核和  $\mu$ DMA 控制器同时访问总线并执行数据传输。

$\mu$ DMA 控制器可以将数据转移到片上 SRAM，也可以从片上 SRAM 将数据转移出。但是，由于 Flash 存储器和 ROM 位于不同的内部总线，所以  $\mu$ DMA 不能从 Flash 存储器或 ROM 转移数据。

$\mu$ DMA 控制器为每种支持  $\mu$ DMA 的外设功能都提供了专用的通道，可以各自独立进行配置。 $\mu$ DMA 控制器的配置方法比较独特，是通过系统存储器中的通道控制结构体进行配置的，并且该结构体由处理器维护。除支持简单传输模式之外， $\mu$ DMA 控制器也支持更加“复杂”的传输模式：在收到某个单次传输请求后，按照建立在存储器中的任务列表，可以执行向/从指定地址发送/接收指定大小数据块的传输流程。 $\mu$ DMA 控制器还支持以乒乓缓冲的方式实现与外设之间的持续数据流。

每个通道还能配置仲裁数目。所谓仲裁数目，是指  $\mu$ DMA 控制器在重新仲裁总线优先级之前，以猝发方式传输的数据单元数目。借助仲裁数目的配置，当外设产生一个  $\mu$ DMA 服务请求后，即可精准地操控与外设之间传输多少个数据单元。

### 9.2.1 通道分配

使用 DMA 通道映射选择 n (DMACHMAPn) 寄存器中的 4 位分配域可以为每个  $\mu$ DMA 通道分配最多 5 种可能的通道分配方式。

表 9-1 在 536 页 示出了 μDMA 通道映射。而在 Enc 列中示出了各自 DMACHMAPn 位域的编码。编码 0x5 - 0xF 是保留的。要支持使用 DMA 通道分配 (DMACHASGN) 寄存器的传统软件，Enc0 要与清零的 DMACHASGN 位相等，且 Enc1 要与置位的 DMACHASGN 位相等。在读取 DMACHASGN 寄存器时，如果相应的 DMACHMAPn 寄存器位域等于 0，则读取的位域返回值为 0；否则，返回值为 1 (如果相应的 DMACHMAPn 寄存器位域不为 0)。表中的类型标识栏用于表示特定外设是使用单个请求 (S)，猝发请求 (B) 还是两者都使用。

注意：表中注明“软件可用”的通道可用于未来进行外设扩展。现在这些通道仅软件可访问，不可连接外设。30号通道是软件专用通道。

映射到 0~3 号 μDMA 通道的 USB 端点可通过 USBDMASEL 寄存器 (见 1133 页) 予以更改。

表 9-1. μDMA 通道分配

Enc.	0		1		2		3		4	
Ch #	外设	类型	外设	类型	外设	类型	外设	类型	外设	类型
0	USB0 EP1 RX	SB	UART2 RX	SB	软件	B	通用定时器 4A	B	软件	B
1	USB0 EP1 TX	B	UART2 TX	SB	软件	B	通用定时器 4B	B	软件	B
2	USB0 EP2 RX	B	通用定时器 3A	B	软件	B	软件	B	软件	B
3	USB0 EP2 TX	B	通用定时器 3B	B	软件	B	软件	B	软件	B
4	USB0 EP3 RX	B	通用定时器 2A	B	软件	B	GPIO A	B	软件	B
5	USB0 EP3 TX	B	通用定时器 2B	B	软件	B	GPIO B	B	软件	B
6	软件	B	通用定时器 2A	B	UART5 RX	SB	GPIO C	B	软件	B
7	软件	B	通用定时器 2B	B	UART5 TX	SB	GPIO D	B	软件	B
8	UART0 RX	SB	UART1 RX	SB	软件	B	通用定时器 5A	B	软件	B
9	UART0 TX	SB	UART1 TX	SB	软件	B	通用定时器 5B	B	软件	B
10	SSI0 RX	SB	SSI1 RX	SB	UART6 RX	SB	通用定时器 6A	B	软件	B
11	SSI0 TX	SB	SSI1 TX	SB	UART6 TX	SB	通用定时器 6B	B	软件	B
12	软件	B	UART2 RX	SB	SSI2 RX	SB	通用定时器 7A	B	GPIO K	B
13	软件	B	UART2 TX	SB	SSI2 TX	SB	通用定时器 7B	B	GPIO L	B
14	ADC0 SS0	B	通用定时器 2A	B	SSI3 RX	SB	GPIO E	B	GPIO M	B
15	ADC0 SS1	B	通用定时器 2B	B	SSI3 TX	SB	GPIO F	B	GPIO N	B
16	ADC0 SS2	B	软件	B	UART3 RX	SB	通用定时器 8A	B	GPIO P	B
17	ADC0 SS3	B	软件	B	UART3 TX	SB	通用定时器 8B	B	软件	B
18	通用定时器 0A	B	通用定时器 1A	B	UART4 RX	SB	GPIO B	B	软件	B
19	通用定时器 0B	B	通用定时器 1B	B	UART4 TX	SB	GPIO G	B	软件	B
20	通用定时器 1A	B	软件	B	UART7 RX	SB	GPIO H	B	软件	B
21	通用定时器 1B	B	软件	B	UART7 TX	SB	GPIO J	B	软件	B
22	UART1 RX	SB	软件	B	软件	B	软件	B	软件	B
23	UART1 TX	SB	软件	B	软件	B	软件	B	软件	B
24	SSI1 RX	SB	ADC1 SS0	B	软件	B	通用定时器 9A	B	软件	B
25	SSI1 TX	SB	ADC1 SS1	B	软件	B	通用定时器 9B	B	软件	B
26	软件	B	ADC1 SS2	B	软件	B	通用定时器 10A	B	软件	B
27	软件	B	ADC1 SS3	B	软件	B	通用定时器 10B	B	软件	B
28	软件	B	软件	B	软件	B	通用定时器 11A	B	软件	B
29	软件	B	软件	B	软件	B	通用定时器 11B	B	软件	B
30	软件	B	软件	B	软件	B	软件	B	软件	B



表 9-1.  $\mu$ DMA 通道分配 (续)

Enc.	0		1		2		3		4	
Ch #	外设	类型	外设	类型	外设	类型	外设	类型	外设	类型
31	保留	B	保留	B	保留	B	保留	B	保留	B

## 9.2.2 优先级

每个通道  $\mu$ DMA 的优先级由通道的序号以及通道的优先级标志位所决定。第 0 号  $\mu$ DMA 通道的优先级最高；通道的序号越大，其优先级越低。每个  $\mu$ DMA 通道都有一个可设置的优先级标志位，由此可分为默认优先级和高优先级。若某个通道的优先级位置位，则该通道将具有高优先级，其优先于所有未将此标志位置位的通道。假如有多个通道都设为高优先级，那么仍将按照通道序号区分其相互的优先级。

通道的优先级位可通过 DMA 通道优先置位 (DMAPRIOSET) 置位，通过 DMA 通道优先清除 (DMAPRIOCLR) 清零。

## 9.2.3 仲裁数目

当某个  $\mu$ DMA 通道请求传输时， $\mu$ DMA 控制器将对所有发出请求的通道进行仲裁，并且向其中优先级最高的通道提供服务。一旦开始传输后，将持续传输一定数量的数据，之后再对发出请求的通道进行仲裁。每个通道的仲裁数目都是可设置的，其有效范围为 1~1024 个数据单元。当  $\mu$ DMA 控制器按照仲裁数目传输了若干个数据单元之后，随后将检查所有发出请求的通道，并向其中优先级最高的通道提供服务。

如果某个优先级较低的  $\mu$ DMA 通道仲裁数目设置得太大，那么高优先级通道的传输延迟将可能增加，因为  $\mu$ DMA 控制器需要等待低优先级的猝发传输完全结束之后才会重新进行仲裁，检查是否存在更高优先级的请求。基于以上原因，建议低优先级通道的仲裁数目不应设得太大，这样可以充分保障系统对高优先级  $\mu$ DMA 通道的响应速度。

仲裁数目也可以形象地看做一个猝发的大小。仲裁数目就是获得控制权后以猝发形式连续传输的数据单元数。请注意这里所说的“仲裁”是指  $\mu$ DMA 通道优先级的仲裁，而非总线的仲裁。在竞争总线时，处理器内核始终优于  $\mu$ DMA 控制器。此外，只要处理器需要在同一总线上执行总线交互， $\mu$ DMA 控制器都将失去总线控制权；即便在猝发传输的过程中， $\mu$ DMA 控制器也将被暂时中断。

## 9.2.4 请求类型

$\mu$ DMA 控制器可响应来自外设的两种请求：单次请求或猝发请求。每种外设可能支持其中一种或两种类型。单次请求表明外设已准备好传输一个数据单元，猝发请求表明外设已准备好传输多个数据单元。

取决于外设发出的是单次请求或猝发请求， $\mu$ DMA 控制器的响应也将有所不同。假如同时产生了单次请求和猝发请求，而且  $\mu$ DMA 通道已按照猝发请求建立，那么优先响应猝发请求。表 9-2 在 537 页列出了各种外设对这两种请求类型的支持情况。

表 9-2. 所支持的请求类型

外设	单个请求信号	猝发请求信号
ADC	无	序列发生器的 IE 位
通用定时器	无	触发事件
GPIO	原始中断脉冲	无
SSI TX	TX FIFO 未滿	TX FIFO 深度 (固定为 4)
SSI RX	RX FIFO 非空	RX FIFO 深度 (固定为 4)
UART TX	TX FIFO 未滿	TX FIFO 深度 (可配置)
UART RX	RX FIFO 非空	RX FIFO 深度 (可配置)

表 9-2. 所支持的请求类型 (续)

外设	单个请求信号	猝发请求信号
USB TX	无	FIFO TXRDY
USB RX	无	FIFO RXRDY

### 9.2.4.1 单个请求

当检测到单次请求、并且没有猝发请求时， $\mu$ DMA 控制器将传输一个数据单元，传输完成后停止并等待其它请求。

### 9.2.4.2 猝发请求

当检测到猝发请求后， $\mu$ DMA 控制器将执行猝发传输，传输数目是以下两者的较小值：仲裁数目；尚未传输完的数据单元数。因此，仲裁数目应与外设发出猝发请求时所包含的数据单元数相同。例如，UART 模块可基于 FIFO 触发深度产生猝发请求。此时仲裁数目应与满足触发深度条件后 FIFO 能够传输的数据单元数相同。猝发传输一旦启动就必须运行到结束，期间即使有更高优先级通道的请求也无法中断。猝发传输所需的时间通常都比数量相同、单次触发的用时总和要短。

实际使用中应尽可能地采用猝发传输，尽量避免单次传输。例如，某些数据天生就只有在作为一个数据块共同传输时才有意义，每次传输一点则毫无用处。通过 DMA 通道采用猝发置位 (DMAUSEBURSTSET) 寄存器可以禁用单次请求。当把此寄存器中对应于某个通道的标志位置位后， $\mu$ DMA 控制器将只响应该通道的猝发请求。

## 9.2.5 通道配置

$\mu$ DMA 控制器采用系统内存中保存一个控制表，表中包含若干个通道控制结构体。每个  $\mu$ DMA 通道在控制表中可能有一个或两个结构体。控制表中的每个结构体都包含：源指针、目的指针、待传输数目、传输模式。控制表可以定义到系统内存中的任意位置，但必须保证其连续并且按 1024 字节边界对齐。

表 9-3 在 538 页 列出了内存中通道控制表的内容分布布局。每个通道在控制表中都可能包含一个或两个结构体：主控制结构体及副控制结构体。在控制表中，所有主控制结构体的都在表的前半部分，所有副控制结构体都在表的下半部分。在较简单的传输模式中，对传输的连续性要求不高，允许在每次传输结束后再重新配置、重新启动。这种情况一般不需要副控制结构体，因此内存中只需放置表的前半部分，而后半部分所占用的内存可用作其它用途。如果采用更加复杂的传输模式（例如乒乓模式或散聚模式），那就需要用到副控制结构体，此时整个控制表都必须加载到内存中。

控制表中任何未用到的内存块都可留给应用程序使用，包括任何应用程序未用的通道的控制结构体，以及各个通道中未用到的控制字。

表 9-3. 控制结构体的存储器映射

偏移量	通道
0x0	通道 0 主功能
0x10	通道 1 主功能
...	...
0x1F0	通道 31 主功能
0x200	通道0副功能
0x210	通道1副功能
...	...
0x3F0	通道31副功能

表 9-4 列出了控制表中单个控制结构体项的内容。每个控制结构体项都按照 16 字节边界对齐。每个结构体项由 4 个长整型项组成：源末指针、目的末指针、控制字以及一个未用的长整型项。末指针

就是指向传输过程最末一个单元地址的指针（包含其本身）。假如源地址或目的地址并不自动递增（例如外设的寄存器），那么指针应当指向待传输的地址。

表 9-4. 通道控制结构体

偏移量	描述
0x000	源末指针
0x004	目的末指针
0x008	控制字
0x00C	未用

控制字包含以下的位域：

- 源/目的数据宽度
- 源/目的地址增量
- 总线重新仲裁之前传输的数目（仲裁数目）
- 待传输的数据单元总数
- 采用猝发传输标志
- 传输模式

关于控制字及其各个位域的详细介绍，请参阅“ $\mu$ DMA 通道控制结构体”在 556 页。 $\mu$ DMA 控制器在传输执行期间自动更新待传输大小位域以及传输模式位域。当传输结束后，待传输数目将为 0，传输模式将变为“已停止”。由于控制字是由  $\mu$ DMA 控制器自动修改的，因此在每次新建传输之前必须手动配置。源末指针和目的末指针不会被自动修改，所以只要源地址或目的地址不变，就无需再进行配置。

在启动传输之前，必须将 DMA 通道启用置位 (DMAENASET) 寄存器中的相应标志位置位，启用  $\mu$ DMA 通道。当需要禁用某个通道时，应将 DMA 通道启用清除 (DMAENACLR) 寄存器中的相应通道标志位置位。当某个  $\mu$ DMA 传输结束后，控制器会自动禁用该通道。

## 9.2.6 传输模式

$\mu$ DMA 控制器支持多种传输模式。前两种模式支持简单的单次传输。后面几种复杂的模式能够实现持续数据流。

### 9.2.6.1 停止模式

停止模式虽然是控制字中传输模式位域的有效值，但实际上这并不是一种真正的传输模式。当控制字中的传输模式是停止模式时， $\mu$ DMA 控制器并不会对此通道进行任何传输，并且一旦该通道启用， $\mu$ DMA 控制器还会自动禁用该通道。在任何  $\mu$ DMA 传输结束后， $\mu$ DMA 控制器都会自动将通道控制字的传输模式位域改写为停止模式。

### 9.2.6.2 基本模式

在基本模式下，只要有待传输的数据单元，并且收到了传输请求， $\mu$ DMA 控制器便会执行传输。这种模式适用于那些只要有数据可传输就产生  $\mu$ DMA 请求信号的外设。如果请求是瞬时的（即使整个传输尚未完成也并不保持），则不得采用基本模式。举例来说，如果将某个通道设为基本模式，并且采用软件启动传输，则启动时只会创建一个瞬时请求；此时每次软件请求所传输的数目等于 DMA 通道控制字 (DMACHCTL) 寄存器中 ARBSIZE 位域所指定的数目（即仲裁数目），即使还有更多数据需要传输也将停止。

在基本模式下，当所有数据单元传输完成后， $\mu$ DMA 控制器自动将该通道置为停止模式。

### 9.2.6.3 自动模式

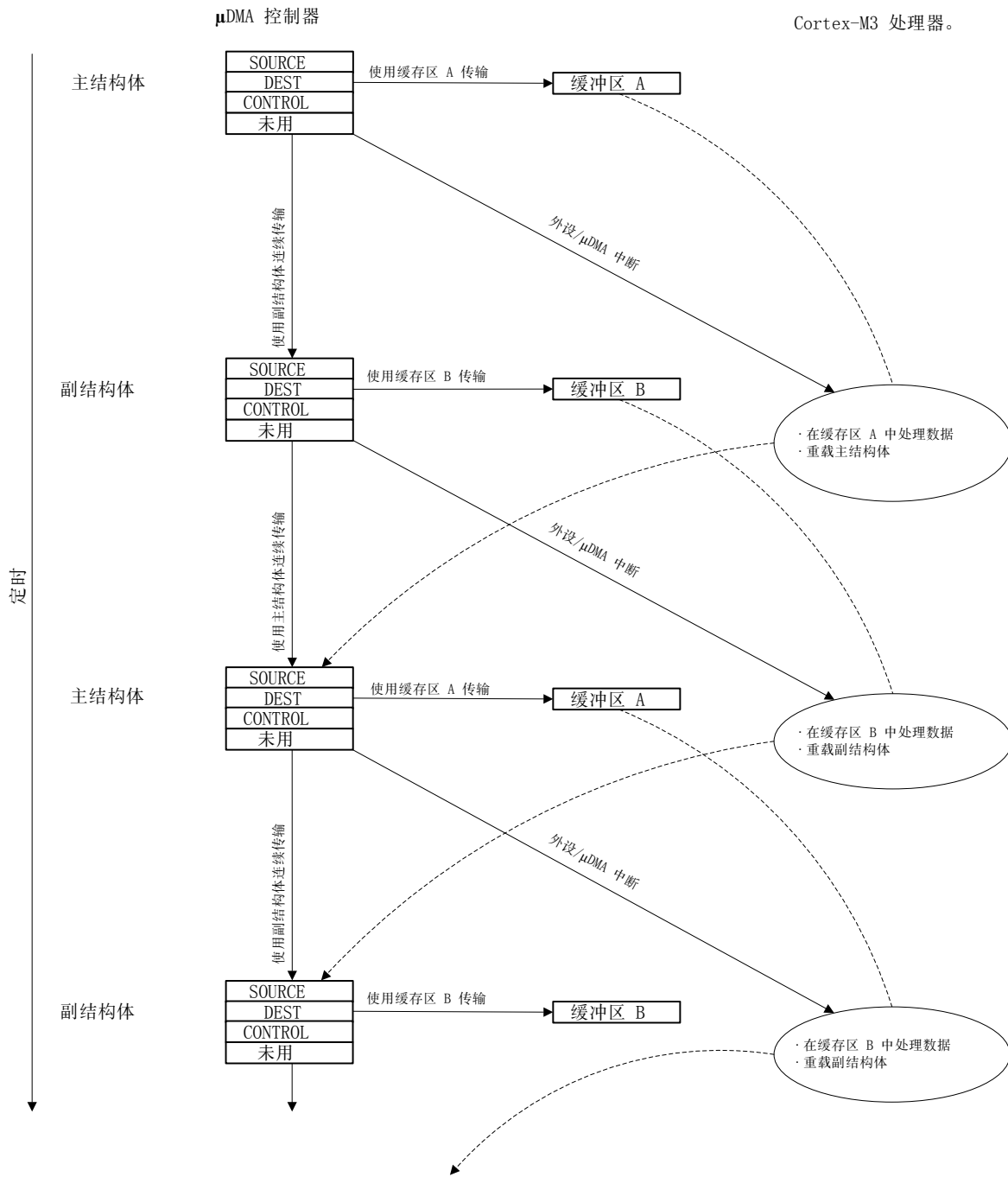
自动模式与基本模式类似，区别在于：每当收到一个传输请求后，传输过程会一直持续到整个传输结束，即使  $\mu$ DMA 请求已经消失（瞬时请求）也会持续完成。这种模式非常适用于软件触发的传输过程。一般来说外设都不使用自动模式。

在自动模式下，当所有数据单元传输完成后， $\mu$ DMA 控制器自动将该通道置为停止模式。

### 9.2.6.4 乒乓

乒乓模式用于实现内存与外设之间连续不断的数据流。要使用乒乓模式，必须同时配置主数据结构和副数据结构体。两个结构体均用于实现存储器与外设之间的数据传输，均由处理器建立。传输过程首先从主控制结构体开始。当主控制结构体所配置的传输过程结束后， $\mu$ DMA 控制器自动载入副控制结构体并按其配置继续传输。每当这时都会产生一个中断，处理器可以对刚刚结束传输过程的数据结构体进行重新配置。于是乎，主/副控制结构体交替在缓冲区与外设之间搬运数据，周而复始，川流不息。

图 9-2 在 541 页 描绘出了乒乓模式下的操作示例。

图 9-2. 乒乓式  $\mu$ DMA 数据会话的示例

### 9.2.6.5 存储器散聚

存储器散聚模式是一种较为复杂的工作模式。通常在搬运数据块时，其数据源和数据目的都是线性分布的；但有时必须将内存中某块连续的数据分散传递到几个不同的位置，或将内存中几个不同位置的数据块汇聚传递到同一个位置连续放置，此时就应当采用散聚模式。举例来说，内存中可能存储有数条遵从某种通信协议的报文，那么就可以利用  $\mu$ DMA 的汇聚模式将几个报文的有效数据内容依次读出、并连续保存到内存缓冲中的指定位置（有效内容拼装）。

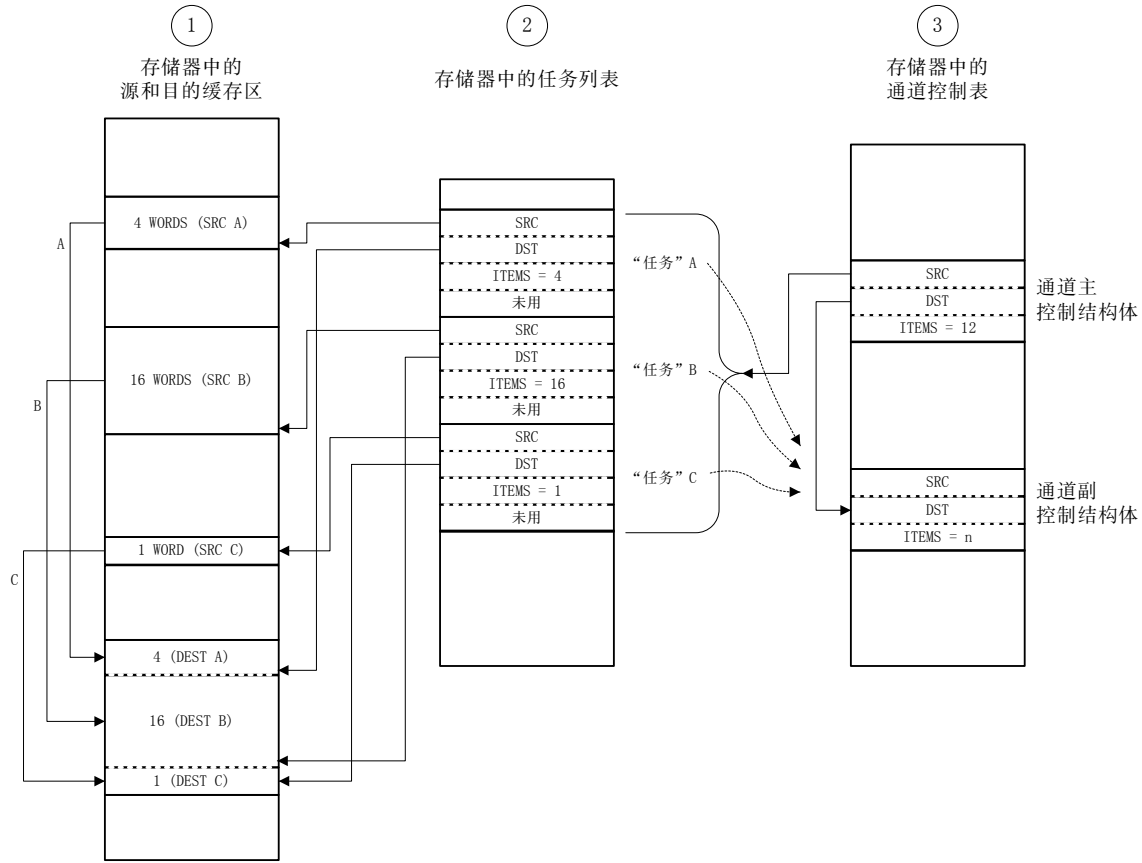
在存储器散聚模式下，主控制结构体的工作是按照内存中一个表的内容配置副控制结构体。这个表由处理器软件建立，包含若干个控制结构体，每个控制结构体中包含能够实现特定传输的源末指针、目的末指针、控制字。每个控制结构体项的控制字中必须将传输模式设置为散聚模式。主传输流程依次将表中的控制结构体项拷贝到副控制结构体中，随后予以执行。 $\mu$ DMA 控制器就这样交替切换：每次用主控制结构体从列表中將下一个传输流程配置拷贝到副控制结构体中，然后切换到副控制结构体执行相应的传输任务。在列表的最末一个控制结构体项中，应将其控制字编程为采用自动传输模式。这样在执行最后一个传输过程时是自动模式， $\mu$ DMA 控制器在执行完成后将停止此通道的运行。只有当最后一次传输过程也结束后，才会产生结束中断。如果让控制表最后一个控制结构体项拷贝覆盖主控制结构体，使其重新指向列表的起始位置（或指向一个新的列表），就可以让整个列表始终不停循环工作。此外通过编辑控制表内容，也可以触发一个或多个其它通道执行传输：比较直接的方式是编辑产生一个写操作、以软件触发其它通道；也可以采用间接的方式，通过设法让某个外设动作而产生  $\mu$ DMA 请求。

按照这种方式对  $\mu$ DMA 控制器进行配置，即可基于一个  $\mu$ DMA 请求执行一组指定的传输。

图 9-3 在 543 页 和 图 9-4 在 544 页 描绘出按照存储器散聚模式工作的示例。这个例子演示的是汇集操作：将分别位于内存中三个不同位置的数据拷贝到同一个缓冲区中并连续放置。图 9-3 在 543 页 描绘出应用程序应如何在内存中建立一个  $\mu$ DMA 任务列表，控制器按照该列表执行三组来自内存中不同位置的拷贝操作。通道的主控制结构体负责将控制结构体项从任务列表中拷贝出来，并填充到副控制结构体中。

图 9-4 在 544 页 描绘出  $\mu$ DMA 控制器执行三组拷贝操作的序列。首先， $\mu$ DMA 控制器按照主控制结构体工作，将任务 A 载入到副控制结构体中。随后  $\mu$ DMA 控制器切换到副控制结构体，按照任务 A 从源缓冲区 A 拷贝数据到目的缓冲区。随后， $\mu$ DMA 控制器再次按照主控制结构体工作，将任务 B 载入到副控制结构体中，并按照副控制结构体执行任务 B 的拷贝操作。对于任务 C 也同样重复以上步骤。

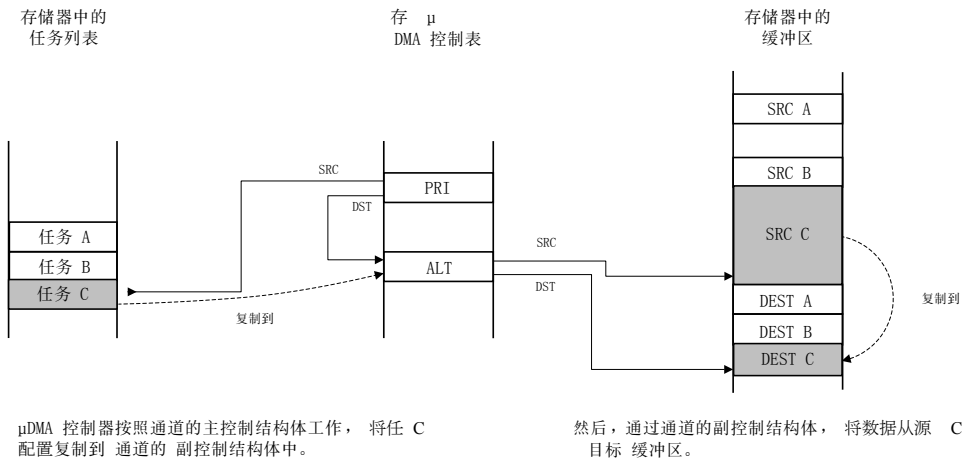
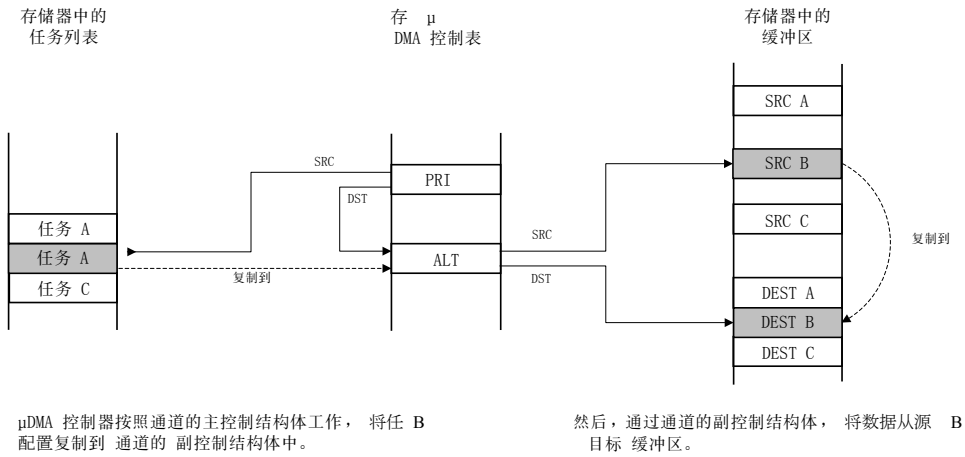
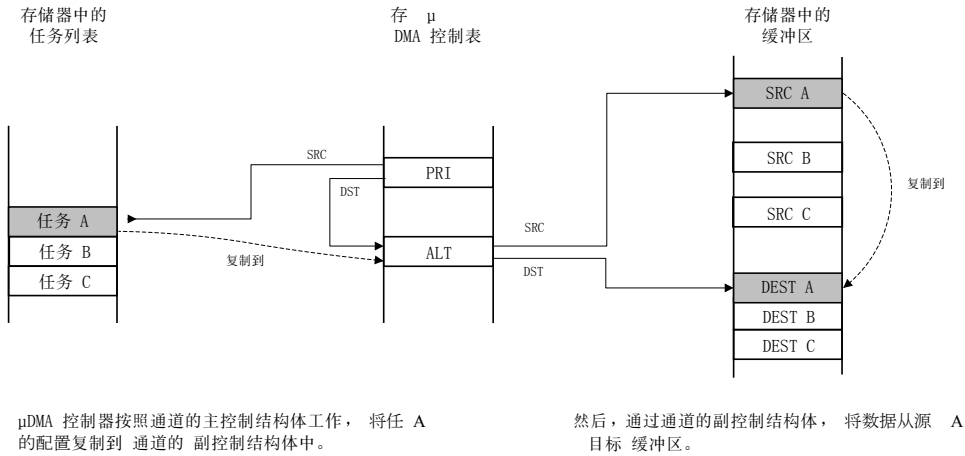
图 9-3. 存储器散聚模式：创建及配置



## 注：

1. 应用程序需要将存储器中三个不同的位置的若干个数据单元拷贝到一个缓冲区中并顺序组合。
2. 应用程序在存储器中建立  $\mu$ DMA “任务列表”，表中包括 3 个  $\mu$ DMA 控制“任务”的指针以及控制配置。
3. 应用程序设置通道的主控制结构体，每次将一个任务的配置拷贝到副控制结构体中，并且接下来由  $\mu$ DMA 控制器予以执行。
4. 任务列表中的 SRC 和 DST 指针必须指向相应缓冲区中的最后位置。

图 9-4. 存储器散聚模式：μDMA 复制序列





### 9.2.6.6 外设散聚

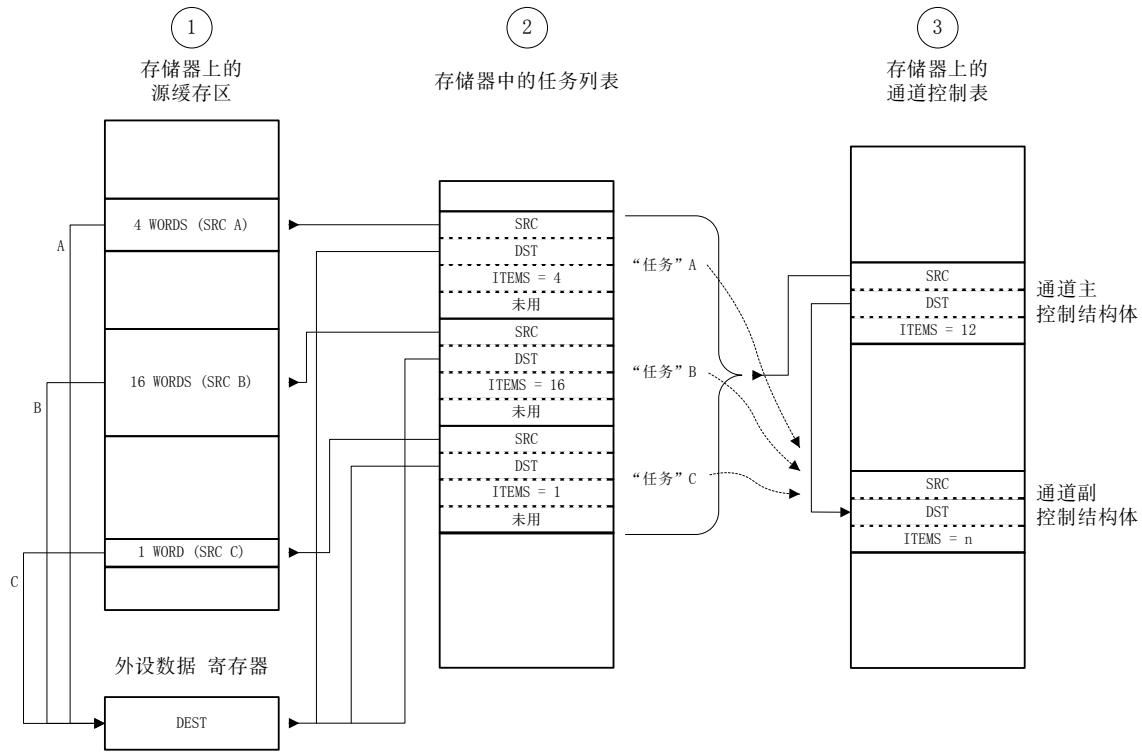
外设散聚模式与存储器散聚模式非常相似，只不过传输过程是由产生  $\mu$ DMA 请求的外设控制的。当  $\mu$ DMA 控制器检测到有来自外设的请求后，将通过主控制结构体从控制表中拷贝一个控制结构体项填充到副控制结构体中，随后执行其传输过程。此次传输过程结束后，只有当外设再次产生  $\mu$ DMA 请求后，才会开始下一个传输过程。仅当外设产生请求时， $\mu$ DMA 控制器才会继续执行控制表中的传输任务，直至完成最后一次传输。只有当最后一次传输过程也结束后，才会产生结束中断。

按照这种方式对  $\mu$ DMA 控制器进行配置，只要外设准备好传输数据，就可以在内存的若干指定地址与外设之间传输数据。

图 9-5 在 546 页 和 图 9-6 在 547 页 描绘出按照外设散聚模式工作的示例。这个例子演示的是汇集操作：将分别位于内存中三个不同位置的数据拷贝到同一个外设数据寄存器中。图 9-5 在 546 页 描绘出应用程序应如何在内存中建立一个  $\mu$ DMA 任务列表，控制器按照该列表执行三组来自内存中不同位置的拷贝操作。通道的主控制结构体负责将控制结构体项从任务列表中拷贝出来，并填充到副控制结构体中。

图 9-6 在 547 页 描绘出  $\mu$ DMA 控制器执行三组拷贝操作的序列。首先， $\mu$ DMA 控制器按照主控制结构体工作，将任务 A 载入到副控制结构体中。随后  $\mu$ DMA 控制器切换到副控制结构体，按照任务 A 从源缓冲区 A 拷贝数据到外设数据寄存器。随后， $\mu$ DMA 控制器再次按照主任务结构体工作，将任务 B 载入到副控制结构体中，并按照副控制结构体执行任务 B 的拷贝操作。对于任务 C 也同样重复以上步骤。

图 9-5. 外设散聚模式：创建及配置



注：  
 1. 应用程序需要将内存中三个不同位置的若干个数据单元复制到一个外设数据寄存器。  
 2. 应用程序在存储器中建立  $\mu$ DMA “任务列表”，表中包括 3 个  $\mu$ DMA 控制“任务”指针以及控制配置。  
 3. 应用程序设置通道的主控制结构体，每次将一个任务的配置复制到副控制结构体中，并且接下来由  $\mu$ DMA 控制器予以执行。



### 9.2.7 待传输数目及增量

$\mu$ DMA 控制器支持传输宽度为 8 位、16 位或 32 位的数据。对于任何传输，都必须保障源数据宽度与目的数据宽度一致。源地址及目的地址可以按字节、半字或字自动递增，也可以设置为不自动递增。源地址增量及目的地址增量相互无关，设置地址增量时只要保证其大于等于数据宽度即可。例如，当传输 8 位宽的数据单元时，将地址增量设置为整字（4 位）也是允许的。待传输的数据在内存中必须按照数据宽度（8 位、16 位或 32 位）对齐。

表 9-5 列出了从某个支持 8 位数据的外设进行读操作时的配置。

表 9-5.  $\mu$ DMA 读操作实例：8 位外设

位域	配置
源数据宽度	8 位
目的数据宽度	8 位
源地址增量	不递增
目的地址增量	字节
源末指针	外设读 FIFO 寄存器
目的末指针	内存中数据缓冲区的末尾

### 9.2.8 外设接口

如果某个外设支持  $\mu$ DMA 功能，则当其作好传输数据的准备时，将可产生一个单次请求信号及/或一个猝发请求信号（见 表 9-2 在 537 页）。请求信号可通过 DMA 通道请求屏蔽置位 (DMAREQMASKSET) 启用，并通过 DMA 通道请求屏蔽清零 (DMAREQMASKCLR) 禁用。若某个通道的请求屏蔽位置位，则禁用（屏蔽）该通道的  $\mu$ DMA 请求信号。假如并未屏蔽该请求信号，并且  $\mu$ DMA 通道已经正确配置并且以及启用，则当外设产生请求信号时， $\mu$ DMA 控制器将开始传输过程。

注意： 当使用  $\mu$ DMA 与外设进行数据通信时，外设必须禁用所有到 NVIC 的中断。

当  $\mu$ DMA 传输结束后， $\mu$ DMA 控制器产生一个中断，详见“中断及错误”在 548 页。

关于某种外设与  $\mu$ DMA 控制器如何相互配合工作的详细信息，请参阅该类型外设 DMA 操作的相关章节。

### 9.2.9 软件请求

在 32 个  $\mu$ DMA 通道中有一个是专用于软件启动的传输过程的。当此通道  $\mu$ DMA 传输结束时，还有专用的中断予以指示。要想正确使用软件启动的  $\mu$ DMA 传输，应首先配置并启用传输过程，之后通过 DMA 通道软件请求 (DMASWREQ) 寄存器发送软件请求。请注意，基于软件的  $\mu$ DMA 传输应当采用自动传输模式。

通过 DMASWREQ 寄存器也可以启动任意通道的  $\mu$ DMA 传输。假如在某个外设的  $\mu$ DMA 通道上采用软件启动请求，那么当传输结束时，结束中断将在该外设的中断向量处产生，而非软件中断向量。只要某个外设并不用到  $\mu$ DMA 数据传输，其  $\mu$ DMA 通道就可以用作软件传输请求。

### 9.2.10 中断及错误

当某个  $\mu$ DMA 传输过程结束时， $\mu$ DMA 控制器将在相应外设的中断向量处产生一个结束中断。因此，假如某个外设采用  $\mu$ DMA 传输数据，并且启用了该外设的中断，那么中断处理函数中必须包含对  $\mu$ DMA 传输结束中断的相关处理。假如传输过程使用了软件  $\mu$ DMA 通道，那么结束中断将在专用的软件  $\mu$ DMA 中断向量上产生（参见 表 9-6 在 549 页）。

当启用某外设的  $\mu$ DMA 后， $\mu$ DMA 控制器将禁止该外设的普通传输中断传递到中断控制器，不过这些中断的状态仍然能在外设的中断寄存器中查询到。因此，当采用  $\mu$ DMA 传输大量数据时，中断控

制器并不会随着数据流从外设频繁收到中断，而是只在数据传输过程结束后收到一个中断。请注意，未屏蔽的外设错误中断仍会正常发送到中断控制器。

当某一  $\mu$ DMA 通道发出完成中断，在 DMA 通道中断状态 (DMACHIS) 寄存器中对应该通道的 CHIS 位被置位 (参见 583 页)。外设中断处理代码也可以通过该寄存器来确定中断是由  $\mu$ DMA 通道造成的，还是由外设中断寄存器上报的出错事件造成的。当中断处理程序激活后， $\mu$ DMA 控制器所发出的完成中断请求会自动清除。

若  $\mu$ DMA 控制器在尝试进行数据传输时遇到了总线错误或存储器保护错误，将会自动关闭出错的  $\mu$ DMA 通道，并且在  $\mu$ DMA 错误中断向量处产生中断。处理器可以通过读取 DMA 总线错误清除 (DMAERRCLR) 寄存器来确定是否产生了错误中断。一旦产生错误则 ERRCLR 位将置位。向 ERRCLR 位写 1 即可清除错误状态。

表 9-6 列出了  $\mu$ DMA 控制器专用的中断。

表 9-6.  $\mu$ DMA 中断分配

中断	分配
46	$\mu$ DMA 软件通道传输中断
47	$\mu$ DMA 错误中断

## 9.3 初始化及配置

### 9.3.1 模块初始化

在使用  $\mu$ DMA 控制器之前，必须先要在系统控制模块中将其启用，并且在外设中启用  $\mu$ DMA 功能。此外，还应当先设置好通道控制结构体的位置。

系统初始化期间应执行一遍下面的步骤：

1. 使用 RCGCDMA 寄存器启用  $\mu$ DMA 时钟 (参见 305 页)。
2. 通过将 DMA 配置 (DMACFG) 中的 MASTEREN 位置位，启用  $\mu$ DMA 控制器。
3. 向 DMA 通道控制基指针 (DMACTLBASE) 寄存器写入通道控制表的基地址，可对通道控制表的位置进行编程。基地址必须按照 1024 字节对齐。

### 9.3.2 存储器到存储器传输的配置

第 30 号  $\mu$ DMA 通道是专用的软件启动传输通道。不过，只要相关外设不使用  $\mu$ DMA 功能，那么任何通道都可以用于软件启动、存储器到存储器的传输。

#### 9.3.2.1 配置通道属性

首先我们应当配置通道属性：

1. 对 DMA 通道优先置位 (DMAPRIOSET) 或 DMA 通道优先清除 (DMAPRIOCLR) 寄存器的第 30 位编程，将通道的优先级设置为最高优先级或者默认优先级。
2. 将 DMA 通道主副清除 (DMAALTCLR) 寄存器中的第 30 位置位，为此次传输选择主通道控制结构体。
3. 将 DMA 通道采用猝发清除 (DMAUSEBURSTCLR) 寄存器中的第 30 位置位，以允许  $\mu$ DMA 控制器既能响应单次请求也能响应猝发请求。
4. 将 DMA 通道请求屏蔽清零 (DMAREQMASKCLR) 寄存器中的第 30 位置位，以允许  $\mu$ DMA 控制器识别该通道的请求。

### 9.3.2.2 配置通道控制结构体

下面来配置通道控制结构体。

本示例需要实现的功能是：从某个内存缓冲区向另一缓冲区传输 256 个字。采用第 30 号通道进行软件启动传输，其控制结构体在控制表中的偏移量为 0x1E0。控制结构体内各个功能字的偏移量见表 9-7。

表 9-7. 第 30 号通道的通道控制结构体偏移量

偏移量	描述
控制表基地址 +0x1E0	第 30 号通道源末指针
控制表基地址 +0x1E4	第 30 号通道目的末指针
控制表基地址 +0x1E8	第 30 号通道控制字

#### 配置源和目的参数

源末指针和目的末指针都应当指向传输过程最后一次传输的地址（其本身包含在内）。

1. 向偏移量 0x1E0 处的源末指针写入：源缓冲地址 +0x3FC (0xFF \* 4)。
2. 向偏移量 0x1E4 处的目的末指针写入：目的缓冲地址 +0x3FC (0xFF \* 4)。

至于偏移量 0x1E8 处的控制字，应按照表 9-8 进行编程。

表 9-8. 存储器传输示例的通道控制字配置

DMACHCTL 中的位域	位	值	描述
DSTINC	31:30	2	目标地址按 32 位自动递增
DSTSIZE	29:28	2	目标数据宽度为 32 位
SRCINC	27:26	2	源地址按 32 位自动递增
SRCSIZE	25:24	2	源数据宽度为 32 位
保留	23:18	0	保留
ARBSIZE	17:14	3	传输 8 个数据单元后仲裁
XFERSIZE	13:4	255	总共传输 256 个单元
NXTUSEBURST	3	0	对本传输类型无意义
XFERMODE	2:0	2	采用自动请求传输模式

### 9.3.2.3 启动传输过程

完成通道配置后，即可启动传输过程：

1. 将 DMA 通道启用置位 (DMAENASET) 寄存器中的第 30 位置位以启用该通道。
2. 将 DMA 通道软件请求 (DMASWREQ) 寄存器中的第 30 位置位，以发送传输请求。

随后就会开始  $\mu$ DMA 传输。倘若同时开启了相关的中断，那么当传输过程全部结束后还会产生中断事件通知处理器。如果需要，还需通过读取 DMAENASET 寄存器中的第 30 位来检查状态。当传输完成后，此位自动清零。此外也可通过通道控制字（偏移量 0x1E8）的 XFERMODE 位域来检查传输状态。当传输完成后，此位自动清零。

### 9.3.3 外设简单发送的配置

在下面的示例中，我们要配置  $\mu$ DMA 控制器，将缓冲区中的数据发送给某个外设。该外设具有发送 FIFO，且触发深度为 4。此示例中的外设占用  $\mu$ DMA 第 7 号通道。

#### 9.3.3.1 配置通道属性

首先我们应当配置通道属性：

1. 配置 DMA 通道优先置位 (DMAPRIOSET) 或 DMA 通道优先清除 (DMAPRIOCLR) 寄存器的第 7 位，将通道的优先级设置为最高优先级或者默认优先级。
2. 将 DMA 通道主副清除 (DMAALTCLR) 寄存器中的第 7 位置位，为此次传输选择主通道控制结构体。
3. 将 DMA 通道采用猝发清除 (DMAUSEBURSTCLR) 寄存器中的第 7 位置位，以允许  $\mu$ DMA 控制器既能响应单次请求也能响应猝发请求。
4. 将 DMA 通道请求屏蔽清零 (DMAREQMASKCLR) 寄存器中的第 7 位置位，以允许  $\mu$ DMA 控制器识别该通道的请求。

#### 9.3.3.2 配置通道控制结构体

本示例需要实现的功能是：从某个内存缓冲区经过第 7 号通道向某个外设的发送 FIFO 寄存器传输 64 个字节。第 7 号通道的控制结构体在控制表中的偏移量为 0x070。控制结构体内各个功能字的偏移量见表 9-9。

表 9-9. 第 7 号通道的通道控制结构体偏移量

偏移量	描述
控制表基地址 +0x070	第 7 号通道源末指针
控制表基地址 +0x074	第 7 号通道目的末指针
控制表基地址 +0x078	第 7 号通道控制字

#### 配置源和目的参数

源末指针和目的末指针都应当指向传输过程最后一次传输的地址（其本身包含在内）。由于外设指针是固定的，因此只需指向外设的数据寄存器即可。

1. 向偏移量 0x070 处的源末指针写入：源缓冲地址 +0x3F。
2. 向偏移量 0x074 处的目的末指针写入：外设的发送 FIFO 寄存器地址。

至于偏移量 0x078 处的控制字，应按照表 9-10 进行编程。

表 9-10. 外设传输示例的通道控制字配置

DMACHCTL 中的位域	位	值	描述
DSTINC	31:30	3	目标地址不自动递增
DSTSIZE	29:28	0	目标数据宽度为 8 位
SRCINC	27:26	0	源地址按 8 位自动递增
SRCSIZE	25:24	0	源数据宽度为 8 位
保留	23:18	0	保留
ARBSIZE	17:14	2	传输 4 个数据单元后仲裁

表 9-10. 外设传输示例的通道控制字配置 (续)

DMACHCTL 中的位域	位	值	描述
XFERSIZE	13:4	63	总共传输 64 个单元
NXTUSEBURST	3	0	对本传输类型无意义
XFERMODE	2:0	1	采用基本传输模式

注意：注：在这个示例中，外设产生的是单次请求还是猝发请求并不重要。由于外设本身具有发送 FIFO，并且在深度达到 4 时触发，因此将仲裁数目设为 4。即使外设真的产生猝发请求，那么传输 4 字节也正好符合 FIFO 的容限。假如外设产生的是单次请求（即 FIFO 中仍然有空位），那么将每次传输 1 个字节。假如应用程序要求必须按猝发方式传输，那么应当将 DMA 通道采用猝发置位 (DMAUSEBURSTSET) 寄存器中管辖通道猝发的 SET[7] 置位。

### 9.3.3.3 启动传输过程

完成通道配置后，即可启动传输过程：

1. 将 DMA 通道启用置位 (DMAENASET) 寄存器中的第 7 位置位以启用该通道。

随后  $\mu$ DMA 控制器即可经由第 7 号通道进行传输。每当外设产生  $\mu$ DMA 请求后，控制器就会向其传输若干数据。当全部 64 个字节传输完成后，传输过程才会结束。传输过程结束后  $\mu$ DMA 控制器将自动禁用该通道，并将通道控制字的 XFERMODE 位清零（停止模式）。可以通过读取 DMA 通道启用置位 (DMAENASET) 寄存器中的第 7 位来检查传输状态。当传输完成后，此位自动清零。此外也可通过通道控制字（偏移量 0x078）的 XFERMODE 位域来检查传输状态。当传输完成后，此位自动清零。

假如启用了该外设的中断，那么当整个传输过程结束时，外设中断处理程序将接收中断。

### 9.3.4 外设乒乓接收的配置

在下面的示例中，我们要配置  $\mu$ DMA 控制器，从某个外设连续接收 8 位数据，并保存到一对 64 字节的缓冲区中。该外设具有发送 FIFO，且触发深度为 8。此示例中的外设占用  $\mu$ DMA 第 8 号通道。

#### 9.3.4.1 配置通道属性

首先我们应当配置通道属性：

1. 配置 DMA 通道优先置位 (DMAPRIOSET) 或 DMA 通道优先清除 (DMAPRIOCLR) 寄存器的第 8 位，将通道的优先级设置为最高优先级或者默认优先级。
2. 将 DMA 通道主副清除 (DMAALTCLR) 寄存器中的第 8 位置位，为此次传输选择主通道控制结构体。
3. 将 DMA 通道采用猝发清除 (DMAUSEBURSTCLR) 寄存器中的第 8 位置位，以允许  $\mu$ DMA 控制器既能响应单次请求也能响应猝发请求。
4. 将 DMA 通道请求屏蔽清零 (DMAREQMASKCLR) 寄存器中的第 8 位置位，以允许  $\mu$ DMA 控制器识别该通道的请求。

#### 9.3.4.2 配置通道控制结构体

下面来配置通道控制结构体，本示例需要实现的功能是：从外设的接收 FIFO 向两个分别为 64 字节的缓冲区传输若干字节。接收数据时，当一个缓冲区装满后， $\mu$ DMA 控制器自动切换到向另一个缓冲区填充收到的数据。



要想实现乒乓式缓冲，必须同时使用该通道的主控制结构体和副控制结构体。第 8 号通道的主控制结构体在控制表中的偏移量为 0x080，副控制结构体在控制表中的偏移量为 0x280。控制结构体内各个功能字的偏移量见 表 9-11。

表 9-11. 第 8 号通道的主控制结构体及副控制结构体偏移量

偏移量	描述
控制表基地址 +0x080	第 8 号通道主源末指针
控制表基地址 +0x084	第 8 号通道主目的末指针
控制表基地址 +0x088	第 8 号通道主控制字
控制表基地址 +0x280	第 8 号通道副源末指针
控制表基地址 +0x284	第 8 号通道副目的末指针
控制表基地址 +0x288	第 8 号通道副控制字

#### 配置源和目的参数

源末指针和目的末指针都应当指向传输过程最后一次传输的地址（其本身包含在内）。由于外设指针是固定的，因此只需指向外设的数据寄存器即可。主控制结构体和副控制结构体中的指针都必须进行配置。

1. 向偏移量 0x080 处的主源末指针写入：外设的接收缓冲地址。
2. 向偏移量 0x084 处的主目的末指针写入：乒乓缓冲区 A 地址 +0x3F。
3. 向偏移量 0x280 处的副源末指针写入：外设的接收缓冲地址。
4. 向偏移量 0x284 处的副目的末指针写入：乒乓缓冲区 B 地址 +0x3F。

至于偏移量 0x088 处的主控制字和 0x288 处的副控制字，应按照下面的方式编程：

1. 按照表 9-12 对偏移量 0x088 处的主控制字进行编程。
2. 按照表 9-12 对偏移量 0x288 处的副通道控制字进行编程。

表 9-12. 外设乒乓接收示例的通道控制字配置

DMACHCTL 中的位域	位	值	描述
DSTINC	31:30	0	目标地址按 8 位自动递增
DSTSIZE	29:28	0	目标数据宽度为 8 位
SRCINC	27:26	3	源地址不自动递增
SRCSIZE	25:24	0	源数据宽度为 8 位
保留	23:18	0	保留
ARBSIZE	17:14	3	传输 8 个数据单元后仲裁
XFERSIZE	13:4	63	总共传输 64 个单元
NXTUSEBURST	3	0	对本传输类型无意义
XFERMODE	2:0	3	采用乒乓传输模式

注意：注：在这个示例中，外设产生的是单次请求还是猝发请求并不重要。由于外设本身具有发送 FIFO，并且在深度达到 8 时触发，因此将仲裁数目设为 8。即使外设真的产生猝发请求，那么传输 8 字节也正好符合 FIFO 的容限。假如外设产生的是单次请求（即 FIFO 中仍然有数据），那么将每次传输 1 个字节。假如应用程序要求必须按猝发方式传输，那么应当将 DMA 通道采用猝发置位 (DMAUSEBURSTSET) 寄存器中管辖通道猝发的 SET[8] 置位。

### 9.3.4.3 配置外设中断

当采用  $\mu$ DMA 的乒乓模式工作时，应当配置中断服务函数。强烈建议通过中断服务函数进行相关处理。不过，乒乓模式也可以采用轮询方式进行相关处理。每当其中一个缓冲区传输完成后即会触发中断。

1. 配置并启用该外设的中断处理函数。

### 9.3.4.4 启用 $\mu$ DMA 通道

完成通道配置后，即可启动传输过程：

1. 将 DMA 通道启用置位 (DMAENASET) 寄存器中的第 8 位置位以启用该通道。

### 9.3.4.5 处理中断

当前已配置并启用了  $\mu$ DMA 控制器，在第 8 号通道上可进行传输。当外设产生  $\mu$ DMA 请求后，控制器将按照主控制结构体的配置将数据传输到缓冲区 A。当对缓冲区 A 的主传输流程结束后，控制器将自动切换到副通道控制结构体，并开始将数据搬运到缓冲区 B。与此同时，主通道控制字的模式位域将自动变为“已停止”，中断将挂起。

当产生中断后，中断处理函数首先应确认哪一缓冲区已传输完成；之后自行处理数据或置标志（有中断外的相应代码根据此标志处理缓冲区的数据）。随后设置本缓冲区下一次的传输任务。

依上所述，在中断处理函数中应当：

1. 读取偏移量 0x088 处的主通道控制字，检查其 XFERMODE 域。若该位域为 0，则表明缓冲区 A 已传输结束。如果缓冲区 A 传输完成，则应当：
  - a. 自行处理缓冲区 A 中刚收到的数据；或置标志表明缓冲区 A 有已接收完成的数据，由专用的缓冲区处理代码进行处理。
  - b. 按照表 9-12 在 553 页对偏移量 0x88 处的主控制字进行编程。
2. 读取偏移量 0x288 处的主通道控制字，检查其 XFERMODE 位域。若该位域为 0，则表明缓冲区 B 已传输结束。如果缓冲区 B 传输完成，则应当：
  - a. 自行处理缓冲区 B 中刚收到的数据；或置标志表明缓冲区 B 有已接收完成的数据，由专用的缓冲区处理代码进行处理。
  - b. 按照表 9-12 在 553 页对偏移量 0x288 处的副控制字进行编程。

### 9.3.5 通道分配的配置

通过 DMACHMAPn 寄存器可更改任一  $\mu$ DMA 通道的通道分配。本寄存器的每个 4 位域分别对应一个  $\mu$ DMA 通道。

关于通道的分配参见表 9-1 在 536 页。

## 9.4 寄存器映射

表 9-13 在 555 页列出了所有  $\mu$ DMA 通道控制结构体以及相关寄存器。通道控制结构体展示出通道控制表中每一项的详细内容。通道控制表位于系统内存中，其位置由应用程序决定，因此其基地址为 n/a（无预定义值），且寄存器描述带有如此提示。在下表中，通道控制结构体的“偏移量”一列代表该配置字相对于控制表中每个结构体项起始地址的偏移。至于通道控制表在内存中的具体排布，请参阅“通道配置”在 538 页以及表 9-3 在 538 页。 $\mu$ DMA 寄存器地址是相对于 0x400F.F000 的  $\mu$ DMA 基地址而言的，以十六进制递增的方式给出。请注意，在操作  $\mu$ DMA 寄存器之前，必须先启用  $\mu$ DMA

模块时钟 ( 请参考 305页 ) 。 $\mu$ DMA 模块时钟启用后, 必须等待至少 3 个系统时钟才可访问  $\mu$ DMA 模块寄存器。

表 9-13.  $\mu$ DMA 寄存器映射

偏移量	名称	类型	复位	描述	见页面
<b><math>\mu</math>DMA 通道控制结构体 ( 从通道控制表基地址的偏移量 )</b>					
0x000	DMASRCENDP	R/W	-	DMA 通道源地址末指针寄存器	557
0x004	DMADSTENDP	R/W	-	DMA 通道目的地址末指针寄存器	558
0x008	DMACHCTL	R/W	-	DMA 通道控制字寄存器	559
<b><math>\mu</math>DMA 寄存器 ( 从 <math>\mu</math>DMA 基地址的偏移量 )</b>					
0x000	DMASTAT	RO	0x001F.0000	DMA 状态寄存器	564
0x004	DMACFG	WO	-	DMA 配置寄存器	566
0x008	DMACTLBASE	R/W	0x0000.0000	DMA 通道控制基指针寄存器	567
0x00C	DMAALTBASE	RO	0x0000.0200	DMA 副通道控制基指针寄存器	568
0x010	DMAWAITSTAT	RO	0xFFFF.FFC0	DMA 通道等待请求状态寄存器	569
0x014	DMASWREQ	WO	-	DMA 通道软件请求寄存器	570
0x018	DMAUSEBURSTSET	R/W	0x0000.0000	DMA 通道采用突发置位寄存器	571
0x01C	DMAUSEBURSTCLR	WO	-	DMA 通道采用突发清除寄存器	572
0x020	DMAREQMASKSET	R/W	0x0000.0000	DMA 通道请求屏蔽置位寄存器	573
0x024	DMAREQMASKCLR	WO	-	DMA 通道请求屏蔽清零寄存器	574
0x028	DMAENASET	R/W	0x0000.0000	DMA 通道启用置位寄存器	575
0x02C	DMAENACLAR	WO	-	DMA 通道启用清除寄存器	576
0x030	DMAALTSET	R/W	0x0000.0000	DMA 通道主副置位寄存器	577
0x034	DMAALTCLR	WO	-	DMA 通道主副清除寄存器	578
0x038	DMAPRIOSET	R/W	0x0000.0000	DMA 通道优先置位寄存器	579
0x03C	DMAPRIOCLR	WO	-	DMA 通道优先清零寄存器	580
0x04C	DMAERRCLR	R/W	0x0000.0000	DMA 总线错误清除寄存器	581
0x500	DMACHASGN	R/W	0x0000.0000	DMA 通道分配寄存器	582
0x504	DMACHIS	R/W1C	0x0000.0000	DMA 通道中断状态寄存器	583
0x510	DMACHMAP0	R/W	0x0000.0000	DMA 通道映射选择寄存器 0	584
0x514	DMACHMAP1	R/W	0x0000.0000	DMA 通道映射选择寄存器 1	585
0x518	DMACHMAP2	R/W	0x0000.0000	DMA 通道映射选择寄存器 2	586
0x51C	DMACHMAP3	R/W	0x0000.0000	DMA 通道映射选择寄存器 3	587
0xFD0	DMAPeriphID4	RO	0x0000.0004	DMA 外设标识寄存器 4	592
0xFE0	DMAPeriphID0	RO	0x0000.0030	DMA 外设标识寄存器 0	588
0xFE4	DMAPeriphID1	RO	0x0000.00B2	DMA 外设标识寄存器 1	589

表 9-13.  $\mu$ DMA 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0xFE8	DMAPeriphID2	RO	0x0000.000B	DMA 外设标识寄存器 2	590
0xFEC	DMAPeriphID3	RO	0x0000.0000	DMA 外设标识寄存器 3	591
0xFF0	DMAPrimeCellID0	RO	0x0000.000D	DMA PrimeCell 标识寄存器 0	593
0xFF4	DMAPrimeCellID1	RO	0x0000.00F0	DMA PrimeCell 标识寄存器 1	594
0xFF8	DMAPrimeCellID2	RO	0x0000.0005	DMA PrimeCell 标识寄存器 2	595
0xFFC	DMAPrimeCellID3	RO	0x0000.00B1	DMA PrimeCell 标识寄存器 3	596

## 9.5 $\mu$ DMA 通道控制结构体

$\mu$ DMA 通道控制结构体保存每个  $\mu$ DMA 通道的传输设置。每个  $\mu$ DMA 通道具有两个控制结构体，所有控制结构体共同在系统内存中组成一个控制表。“通道配置”在 538 页给出了通道控制表以及通道控制结构体的详细解释。

通道控制表由若干个控制结构体项组成。每个通道都有一个主控制结构体和一个副控制结构体。主控制结构体位于偏移量 0x0、0x10、0x20，依此类推。副控制结构体位于偏移量 0x200、0x210、0x220，依此类推。

**寄存器 1: DMA 通道源地址末指针寄存器 ( DMASRCENDP ) , 偏移量 0x000**

DMA 通道源地址末指针 (DMASRCENDP) 是通道控制结构体的一部分, 用于定义  $\mu$ DMA 传输的源地址。

$\mu$ DMA 控制器可以将数据转移到片上 SRAM, 也可以从片上 SRAM 将数据转移出。但是, 由于 Flash 存储器和 ROM 位于不同的内部总线, 所以  $\mu$ DMA 控制器不能向/从 Flash 存储器或 ROM 转移数据。

**注意:** 此处的偏移量是指针对系统内存中控制结构体的基地址, 而不是  $\mu$ DMA 模块的基地址。

**DMA 通道源地址末指针寄存器 (DMASRCENDP)**

基址 n/a

偏移量 0x000

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:0	ADDR	R/W	-	源地址末指针 此位域的内容是 $\mu$ DMA 传输源数据块中最后一个数据单元的地址 ( 其本身包含在内)。假如源地址不自动递增 ( DMACHCTL 寄存器的 SRCINC 域为 0x3 ), 那么此域将指向源地址本身 ( 例如外设的数据寄存器 )。

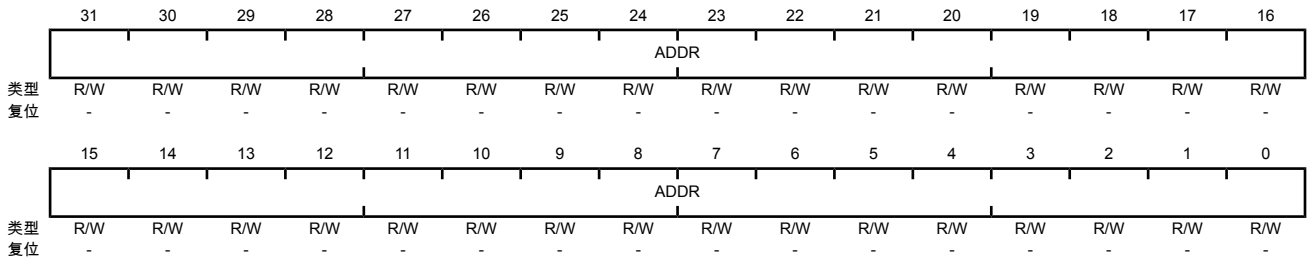
**寄存器 2: DMA 通道目的地址末指针寄存器 ( DMADSTENDP ) , 偏移量 0x004**

DMA 通道目的地址末指针 (DMADSTENDP) 是通道控制结构体的一部分, 用于定义 μDMA 传输的目的地址。

**注意:** 此处的偏移量是指针对系统内存中控制结构体的基地址, 而不是 μDMA 模块的基地址。

DMA 通道目的地址末指针寄存器 (DMADSTENDP)

基址 n/a  
偏移量 0x004  
类型 R/W, 复位 -



位/域	名称	类型	复位	描述
31:0	ADDR	R/W	-	目的地址末指针 此位域的内容是 μDMA 传输目的数据块中最后一个数据单元的地址 ( 其本身包含在内 )。假如目的地址不自动递增 ( DMACHCTL 寄存器的 DSTINC 位域为 0x3 ), 那么此位域将指向目的地址本身 ( 例如外设的数据寄存器 )。

### 寄存器 3: DMA 通道控制字寄存器 ( DMACHCTL ) , 偏移量 0x008

DMA 通道控制字 (DMACHCTL) 是通道控制结构体的一部分, 用于指定  $\mu$ DMA 传输的参数。

注意: 此处的偏移量是指针对系统内存中控制结构体的基地址, 而不是  $\mu$ DMA 模块的基地址。

#### DMA 通道控制字寄存器 (DMACHCTL)

基址 n/a

偏移量 0x008

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DSTINC		DSTSIZE		SRCINC		SRCSIZE		保留				ARBSIZE			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARBSIZE		XFERSIZE										NXTUSEBURST	XFERMODE		
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:30	DSTINC	R/W	-	<p>目的地址增量</p> <p>此位域用于配置目的地址自动递增时的增量。</p> <p>地址增量值必须等于或大于目的数据宽度 (DSTSIZE) 的值。</p> <p>值 描述</p> <p>0x0 字节 按照8位地址递增。</p> <p>0x1 半字 按照16位地址递增。</p> <p>0x2 字 按照32位地址递增。</p> <p>0x3 不递增 通道的目的地址始终等于目的地址末指针 (DMADSTENDP)。</p>
29:28	DSTSIZE	R/W	-	<p>目的数据宽度</p> <p>此位域用于配置目的数据宽度。</p> <p>注意: DSTSIZE 必须与 SRCSIZE 相等。</p> <p>值 描述</p> <p>0x0 字节 传输的每个数据单元为8位。</p> <p>0x1 半字 传输的每个数据单元为16位。</p> <p>0x2 字 传输的每个数据单元为32位。</p> <p>0x3 保留</p>

位/域	名称	类型	复位	描述
27:26	SRCINC	R/W	-	<p>源地址增量 此位域用于配置源地址自动递增时的增量。 地址增量必须大于等于源数据宽度 (SRCSIZE)。</p> <p>值 描述</p> <p>0x0 字节 按照8位地址递增。</p> <p>0x1 半字 按照16位地址递增。</p> <p>0x2 字 按照32位地址递增。</p> <p>0x3 不递增 通道的源地址始终等于源地址末指针 (DMASRCENDP)。</p>
25:24	SRCSIZE	R/W	-	<p>源数据宽度 此位域用于配置源数据宽度。 注意: DSTSIZE 必须与 SRCSIZE 相等。</p> <p>值 描述</p> <p>0x0 字节 传输的每个数据单元为 8 位。</p> <p>0x1 半字 传输的每个数据单元为 16 位。</p> <p>0x2 字 传输的每个数据单元为 32 位。</p> <p>0x3 保留</p>
23:18	保留	R/W	-	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>



位/域	名称	类型	复位	描述																								
17:14	ARBSIZE	R/W	-	<p>仲裁数目</p> <p>此位域用于配置传输了多少个数据单元后，<math>\mu</math>DMA 控制器重新进行仲裁。仲裁数目是 2 的整数幂，其可能的配置如下表所示。</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>1 个单元 <math>\mu</math>DMA 控制器每传输 1 个数据单元后即重新仲裁。</td> </tr> <tr> <td>0x1</td> <td>2 个单元</td> </tr> <tr> <td>0x2</td> <td>4 个单元</td> </tr> <tr> <td>0x3</td> <td>8 个单元</td> </tr> <tr> <td>0x4</td> <td>16 个单元</td> </tr> <tr> <td>0x5</td> <td>32 个单元</td> </tr> <tr> <td>0x6</td> <td>64 个单元</td> </tr> <tr> <td>0x7</td> <td>128 个单元</td> </tr> <tr> <td>0x8</td> <td>256 个单元</td> </tr> <tr> <td>0x9</td> <td>512 个单元</td> </tr> <tr> <td>0xA-0xF</td> <td>1024 个单元</td> </tr> </tbody> </table> <p>当配置为此数值时，由于待传输数目最多也只能是 1024 个单元，因此 <math>\mu</math>DMA 传输过程中将不再仲裁。</p>	值	描述	0x0	1 个单元 $\mu$ DMA 控制器每传输 1 个数据单元后即重新仲裁。	0x1	2 个单元	0x2	4 个单元	0x3	8 个单元	0x4	16 个单元	0x5	32 个单元	0x6	64 个单元	0x7	128 个单元	0x8	256 个单元	0x9	512 个单元	0xA-0xF	1024 个单元
值	描述																											
0x0	1 个单元 $\mu$ DMA 控制器每传输 1 个数据单元后即重新仲裁。																											
0x1	2 个单元																											
0x2	4 个单元																											
0x3	8 个单元																											
0x4	16 个单元																											
0x5	32 个单元																											
0x6	64 个单元																											
0x7	128 个单元																											
0x8	256 个单元																											
0x9	512 个单元																											
0xA-0xF	1024 个单元																											
13:4	XFERSIZE	R/W	-	<p>待传输数目 (减1)</p> <p>此位域配置要传输的数据单元总数。配置值等于实际要传输的数据单元数减 1，也就是说若此位域的值 0，则实际需要传输 1 个数据单元。此位域共有 10 位，最大可能值为 1023，因此最多允许传输 1024 个单元。</p> <p>待传输数目的单位是数据单元，而不是字节。若数据宽度为 32 位，则可传输若干个 32 位宽的字。</p> <p><math>\mu</math>DMA 控制器每次进入仲裁流程之前都会立即更新此位域，因此该位域的实际含义是剩余需要传输的数据单元数。</p>																								
3	NXTUSEBURST	R/W	-	<p>下一个采用猝发</p> <p>在外设散聚模式下，此位域控制是否在最后一次传输时自动将采用猝发的 SET[n] 位置位。通常在最后一次传输时，假如剩余待传输的数据单元数少于仲裁数目，那么 <math>\mu</math>DMA 控制器会采用单次传输来完成本次数据会话。若此标志位置位，那么控制器将采用猝发传输来完成最后一次传输。</p>																								

位/域	名称	类型	复位	描述																		
2:0	XFERMODE	R/W	-	<p><math>\mu</math>DMA 传输模式</p> <p>此位域控制 <math>\mu</math>DMA 流程的工作模式。关于传输模式，详见“传输模式”在 539页。</p> <p>由于此寄存器也位于系统内存中，因此其复位值不定。启用此通道前必须将此位域初始化为 0。</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>停止</td> </tr> <tr> <td>0x1</td> <td>基本</td> </tr> <tr> <td>0x2</td> <td>自动请求</td> </tr> <tr> <td>0x3</td> <td>乒乓</td> </tr> <tr> <td>0x4</td> <td>存储器散聚</td> </tr> <tr> <td>0x5</td> <td>副存储器散聚</td> </tr> <tr> <td>0x6</td> <td>外设散聚</td> </tr> <tr> <td>0x7</td> <td>副外设散聚</td> </tr> </tbody> </table>	值	描述	0x0	停止	0x1	基本	0x2	自动请求	0x3	乒乓	0x4	存储器散聚	0x5	副存储器散聚	0x6	外设散聚	0x7	副外设散聚
值	描述																					
0x0	停止																					
0x1	基本																					
0x2	自动请求																					
0x3	乒乓																					
0x4	存储器散聚																					
0x5	副存储器散聚																					
0x6	外设散聚																					
0x7	副外设散聚																					

**XFERMODE 位域的有效值.**

**停止**

通道已停止或配置数据无效。此时不会产生传输。

**基本**

每次触发后（不论是外设请求还是软件请求）， $\mu$ DMA 控制器都按照 ARBSIZE 位域指定的数目传输若干次。

**自动请求**

当被触发一次后（不论是外设请求还是软件请求）， $\mu$ DMA 控制器都按照 XFERSIZE 位域指定的数目完成整个传输过程，无需额外再产生请求信号。

**乒乓**

这种模式需要同时用到主控制结构体和副控制结构体。当前（主或副）控制结构体的 XFERSIZE 位域指定的数目已经传输完成后， $\mu$ DMA 控制器将切换到另一组控制结构体并按其指定的方式工作。只要控制结构体都还是乒乓模式， $\mu$ DMA 控制器就会这样不停切换交替传输下去。当其中一个控制结构体设为非乒乓模式后，乒乓传输流程才会停止。单个控制结构体配置的传输完成后将会产生中断。请参考“乒乓”在 540页。

**存储器散聚**

这种模式需要同时用到主控制结构体和副控制结构体，按照一个任务列表依次执行其各个动作。此时，主控制结构体的源地址指针应指向一个任务列表的顶部，主控制结构体负责将任务配置拷贝到副控制结构体中。副控制结构体的 XFERMODE 位域必须配置为 0x5（副存储器散聚）才能正确执行所载入的任务。当任务执行结束后， $\mu$ DMA 切换回主控制结构体，并将任务列表中的下一项任务配置拷贝到副控制结构体中。流程就这样不断进行直至整个任务列表全部完成。最后一项任务的 XFERMODE 值不得为 0x5。请注意，要想实现永不停歇的循环工作，可将最后一项任务制定为更新主控制结构体，使其重新指向任务列表的顶部，当然也可以指向其它的任务列表。请参考“存储器散聚”在 541页。

**副存储器散聚**

当  $\mu$ DMA 控制器工作于存储器散聚模式时，必须将副控制结构体的工作模式配置为此种模式。

#### 外设散聚

当  $\mu$ DMA 控制器工作于外设散聚模式时，必须将主控制结构体的工作模式配置为此种模式。此模式下  $\mu$ DMA 控制器的工作流程与存储器散聚模式基本相同，区别在于：每次传输并非按照副控制结构体 XFERSIZE 位域定义的数目执行传输，而是按照 ARBSIZE 位域定义的数目执行传输，类似于基本模式。请参考“外设散聚”在 545 页。

#### 副外设散聚

当  $\mu$ DMA 控制器工作于外设散聚模式时，必须将副控制结构体的工作模式配置为此种模式。

## 9.6 $\mu$ DMA 寄存器描述

给出的寄存器地址都是相对于 0x400F.F000 的  $\mu$ DMA 模块基地址而言的。

### 寄存器 4: DMA 状态寄存器 (DMASTAT), 偏移量 0x000

DMA 状态 (DMASTAT) 寄存器可返回 μDMA 控制器的当前状态。若 μDMA 控制器处于复位状态, 则不能读取本寄存器。

#### DMA 状态寄存器 (DMASTAT)

基址 0x400F.F000  
偏移量 0x000  
类型 RO, 复位 0x001F.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留											DMACHANS				
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								STATE				保留			MASTEN
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述																								
31:21	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。																								
20:16	DMACHANS	RO	0x1F	可用的 μDMA 通道数 (减 1) 此位域的值等于 μDMA 控制器可用的 μDMA 通道数减 1。复位值 0x1F 表示本器件有 32 个可用的 μDMA 通道。																								
15:8	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。																								
7:4	STATE	RO	0x0	控制状态机状态 此位域能够体现当前状态机的状态。可能的状态包括 :  <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr><td>0x0</td><td>空闲</td></tr> <tr><td>0x1</td><td>正在读取通道控制数据</td></tr> <tr><td>0x2</td><td>正在读取源末指针</td></tr> <tr><td>0x3</td><td>正在读取目的末指针</td></tr> <tr><td>0x4</td><td>正在从源地址读取数据</td></tr> <tr><td>0x5</td><td>正在向目的地址写数据</td></tr> <tr><td>0x6</td><td>正在等待μDMA请求清除</td></tr> <tr><td>0x7</td><td>正在写入通道控制数据</td></tr> <tr><td>0x8</td><td>已挂起</td></tr> <tr><td>0x9</td><td>已完成</td></tr> <tr><td>0xA-0xF</td><td>未定义</td></tr> </tbody> </table>	值	描述	0x0	空闲	0x1	正在读取通道控制数据	0x2	正在读取源末指针	0x3	正在读取目的末指针	0x4	正在从源地址读取数据	0x5	正在向目的地址写数据	0x6	正在等待μDMA请求清除	0x7	正在写入通道控制数据	0x8	已挂起	0x9	已完成	0xA-0xF	未定义
值	描述																											
0x0	空闲																											
0x1	正在读取通道控制数据																											
0x2	正在读取源末指针																											
0x3	正在读取目的末指针																											
0x4	正在从源地址读取数据																											
0x5	正在向目的地址写数据																											
0x6	正在等待μDMA请求清除																											
0x7	正在写入通道控制数据																											
0x8	已挂起																											
0x9	已完成																											
0xA-0xF	未定义																											
3:1	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。																								

---

位/域	名称	类型	复位	描述
0	MASTEN	RO	0	主启用状态
				值 描述
				0 $\mu$ DMA 控制器已被禁用。
				1 $\mu$ DMA 控制器已被启用。

## 寄存器 5: DMA 配置寄存器 (DMACFG), 偏移量 0x004

DMACFG 寄存器用于配置 μDMA 控制器。

### DMA 配置寄存器 (DMACFG)

基址 0x400F.F000

偏移量 0x004

类型 WO, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															MASTEN
类型	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:1	保留	WO	-	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	MASTEN	WO	-	控制器主机启用

#### 值 描述

0 禁用 μDMA 控制器。

1 启用 μDMA 控制器。

## 寄存器 6: DMA 通道控制基指针寄存器 ( DMACTLBASE ) , 偏移量 0x008

必须配置 DMACTLBASE 寄存器, 使控制表的基地址指针指向系统内存中的某个地址。

μDMA 控制器所需的内存大小并不固定, 取决于应用程序需要用到的 μDMA 通道数量, 以及是否需要用到副控制数据结构体。关于通道控制表的详细信息, 请参见“通道配置”在 538 页。基地址必须按照 1024 字节对齐。当 μDMA 处于复位状态时, 此寄存器不可读。

### DMA 通道控制基指针寄存器 (DMACTLBASE)

基址 0x400F.F000  
偏移量 0x008  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR						保留									
类型	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

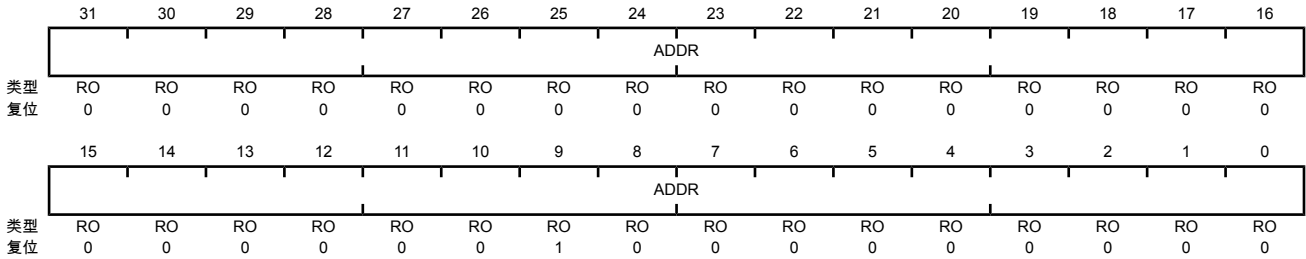
位/域	名称	类型	复位	描述
31:10	ADDR	R/W	0x0000.00	通道控制基地址 此位域包含指向通道控制表的基地址指针。基地址必须按照1024字节对齐。
9:0	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

### 寄存器 7: DMA 副通道控制基指针寄存器 ( DMAALTBASE ) , 偏移量 0x00C

DMAALTBASE 寄存器可返回副通道控制结构体数据块的基地址。提供本寄存器是为了方便软件使用，无需自行计算副控制结构体的地址。当 μDMA 处于复位状态时，此寄存器不可读。

#### DMA 副通道控制基指针寄存器 (DMAALTBASE)

基址 0x400F.F000  
 偏移量 0x00C  
 类型 RO, 复位 0x0000.0200



位/域	名称	类型	复位	描述
31:0	ADDR	RO	0x0000.0200	副通道地址指针 此位域包含副控制结构体的基地址。



**寄存器 8: DMA 通道等待请求状态寄存器 (DMAWAITSTAT) , 偏移量 0x010**

本只读寄存器用于指示出  $\mu$ DMA 控制器是否正在等待请求。为了提高  $\mu$ DMA 的性能，可以禁止在外设发出单次请求时触发  $\mu$ DMA 传输，而是只在外设产生猝发请求时才通过  $\mu$ DMA 传输。此功能需外设的设计予以支持才能正常使用，软件无论采取什么方式都无法控制。当  $\mu$ DMA 处于复位状态时，此寄存器不可读。

**DMA 通道等待请求状态寄存器 (DMAWAITSTAT)**

基址 0x400F.F000

偏移量 0x010

类型 RO, 复位 0xFFFF.FFC0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WAITREQ[n]															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WAITREQ[n]															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	WAITREQ[n]	RO	0xFFFF.FFC0	<p>通道[n]的等待状态</p> <p>此位域可返回各个通道等待请求的状态。第0位对应于第0号通道，依此类推。</p> <p>值 描述</p> <p>1 相应通道并未等待请求。</p> <p>0 相应通道正在等待请求。</p>

### 寄存器 9: DMA 通道软件请求寄存器 (DMASWREQ), 偏移量 0x014

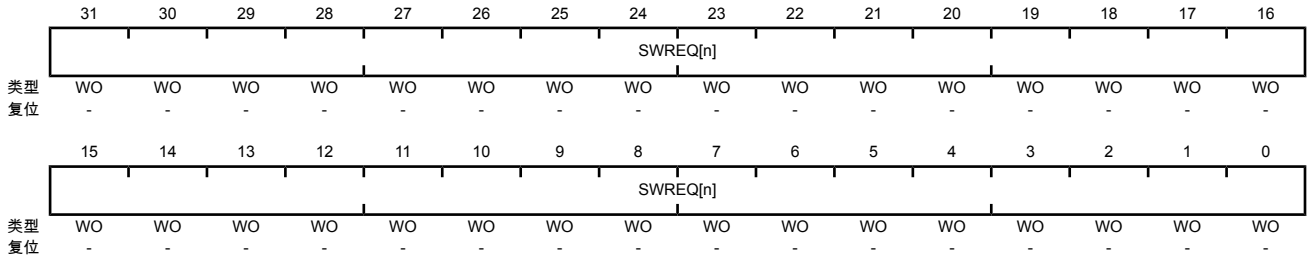
DMASWREQ 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位, 即会向相应的  $\mu$ DMA 通道产生一个请求信号。

#### DMA 通道软件请求寄存器 (DMASWREQ)

基址 0x400F.F000

偏移量 0x014

类型 WO, 复位 -



位/域	名称	类型	复位	描述
31:0	SWREQ[n]	WO	-	通道[n]的软件请求 此位域用于产生软件请求。第0位对应于第0号通道, 依此类推。

#### 值 描述

- 1 向相应通道产生软件请求。
- 0 不产生软件请求。

当软件请求完成后, 相应标志位将自动清零。

**寄存器 10: DMA 通道采用猝发置位寄存器 (DMAUSEBURSTSET)，偏移量 0x018**

DMAUSEBURSTSET 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位，即可禁止相应的  $\mu$ DMA 通道响应单次请求，只接受猝发请求。读取本寄存器可返回各个通道采用猝发的状态。

假如待传输数据的数目是仲裁数目（猝发大小）的整数倍，那么当完成最后一次传输后，相应的 SET[n] 位将会清零。假如剩余待传输的数据单元少于仲裁数目（猝发大小），那么  $\mu$ DMA 控制器会自动将 SET[n] 位清零，并且剩余的数据单元将按照单次请求的方式传输。如果想用猝发方式传输剩余的数据单元，应将相应位再次置位。假如外设不支持猝发请求模型，则此寄存器中的相关位不得置位。

详细信息和实例请参考“请求类型”在 537 页。

**DMA 通道采用猝发置位寄存器 (DMAUSEBURSTSET)**

基址 0x400F.F000

偏移量 0x018

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	SET[n]	R/W	0x0000.0000	通道[n]的采用猝发置位

**值 描述**

0  $\mu$ DMA 通道 [n] 既可响应单次请求也响应猝发请求。

1  $\mu$ DMA 通道 [n] 只响应猝发请求。

第0位对应于第0号通道，依此类推。该位如上述所描述的将自动清零也可通过将 DMAUSEBURSTCLR 寄存器中相应的 CLR[n] 位置位，手动清零该位。

## 寄存器 11: DMA 通道采用猝发清除寄存器 (DMAUSEBURSTCLR), 偏移量 0x01C

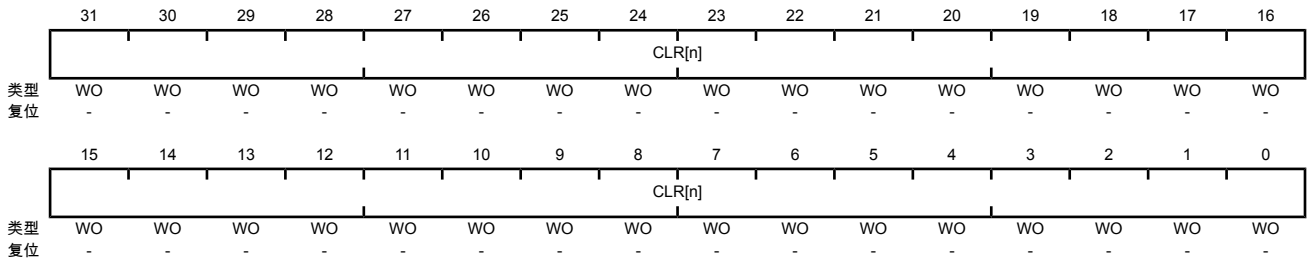
DMAUSEBURSTCLR 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位, 则 DMAUSEBURSTSET 寄存器中相应的 SET[n] 位即会清零。

### DMA 通道采用猝发清除寄存器 (DMAUSEBURSTCLR)

基址 0x400F.F000

偏移量 0x01C

类型 WO, 复位 -



位/域	名称	类型	复位	描述
31:0	CLR[n]	WO	-	通道 [n] 的采用猝发清零

#### 值 描述

0 无影响。

1 将某个位置位, 即可将 DMAUSEBURSTSET 寄存器中的 SET[n] 清零, 也就是说  $\mu$ DMA 通道 [n] 可响应单次请求和猝发请求。

## 寄存器 12: DMA 通道请求屏蔽置位寄存器 ( DMAREQMASKSET ) , 偏移量 0x020

DMAREQMASKSET 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位，相应的  $\mu$ DMA 通道即不再自动产生请求。读取本寄存器可返回请求屏蔽的状态。当屏蔽某个  $\mu$ DMA 通道的请求后，外设将无法请求  $\mu$ DMA 传输。于是该通道便可用于软件启动的  $\mu$ DMA 传输流程。

### DMA 通道请求屏蔽置位寄存器 (DMAREQMASKSET)

基址 0x400F.F000  
偏移量 0x020  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	SET[n]	R/W	0x0000.0000	通道[n]的请求屏蔽置位

#### 值 描述

0 相应通道所关联的外设可以请求  $\mu$ DMA 传输。

1 相应通道所关联的外设不能请求  $\mu$ DMA 传输。此时该通道可用于软件启动的  $\mu$ DMA 传输。

第 0 位对应于第 0 号通道，依此类推。要想将本寄存器中的某个位清零，需将 DMAREQMASKCLR 寄存器的相应 CLR[n] 位置位。

### 寄存器 13: DMA 通道请求屏蔽清零寄存器 ( DMAREQMASKCLR ) , 偏移量 0x024

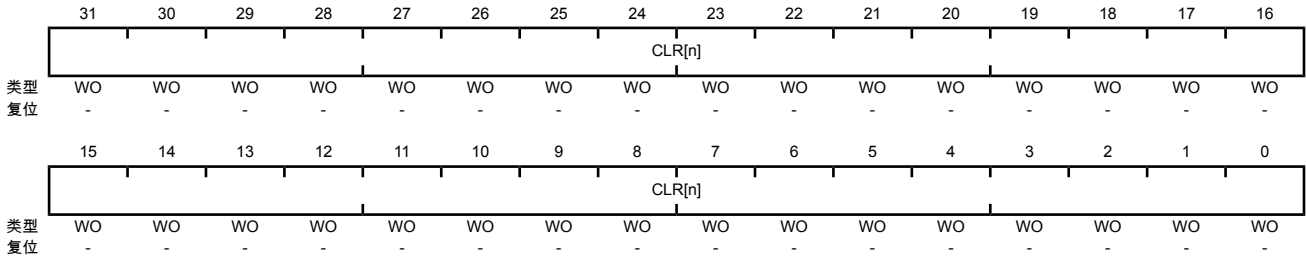
DMAREQMASKCLR 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位, 则 DMAREQMASKSET 寄存器中相应的 SET[n] 位即会清零。

#### DMA 通道请求屏蔽清零寄存器 (DMAREQMASKCLR)

基址 0x400F.F000

偏移量 0x024

类型 WO, 复位 -



位/域	名称	类型	复位	描述
31:0	CLR[n]	WO	-	通道 [n] 的请求屏蔽清零

#### 值 描述

0 无影响。

1 将某个位置位, 即可将 DMAREQMASKSET 寄存器中的 SET[n] 清零, 也就是说启用了与该通道 [n] 关联的外设, 并请求  $\mu$ DMA 传输。

## 寄存器 14: DMA 通道启用置位寄存器 (DMAENASET), 偏移量 0x028

DMAENASET 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位, 即可启用相应的  $\mu$ DMA 通道。读取本寄存器可返回各通道的启用状态。假如某个通道已经启用, 但是屏蔽自动请求 (DMAREQMASKSET), 那么这个通道就能用于软件启动的传输。

### DMA 通道启用置位寄存器 (DMAENASET)

基址 0x400F.F000  
偏移量 0x028  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	SET[n]	R/W	0x0000.0000	通道 [n] 的启用设置

值 描述  
0 禁用  $\mu$ DMA 通道 [n]。  
1 启用  $\mu$ DMA 通道 [n]。

第 0 位对应于第 0 号通道。要想将本寄存器中的某个位清零, 需将 DMAENACLR 寄存器的相应 CLR[n] 位置位, 或等到  $\mu$ DMA 传输结束。

### 寄存器 15: DMA 通道启用清除寄存器 (DMAENACLRL), 偏移量 0x02C

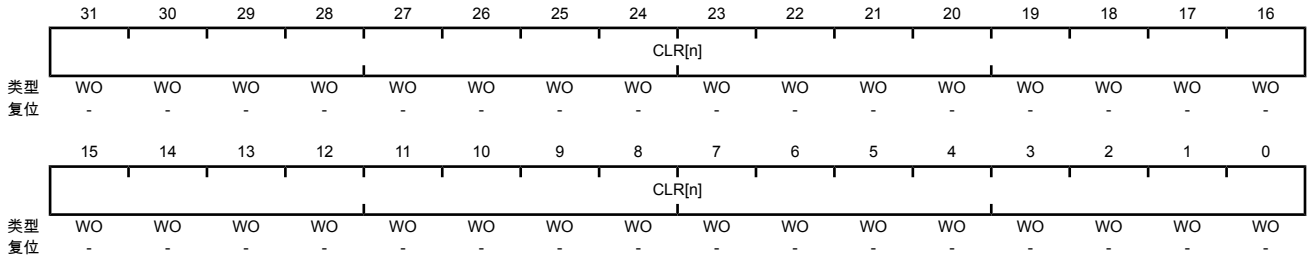
DMAENACLRL 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位, 则 DMAENASET 寄存器中相应的 SET[n] 位即会清零。

#### DMA 通道启用清除寄存器 (DMAENACLRL)

基址 0x400F.F000

偏移量 0x02C

类型 WO, 复位 -



位/域	名称	类型	复位	描述
31:0	CLR[n]	WO	-	清除通道 [n] 启用清除

值 描述

0 无影响。

1 将某个位置位, 即可将 DMAENASET 寄存器中的 SET[n] 清零, 也就是说通道 [n] 禁用了  $\mu$ DMA 传输。

注意: 当某一通道完成  $\mu$ DMA 传输时, 控制器会禁用它。



## 寄存器 16: DMA 通道主副置位寄存器 (DMAALTSET), 偏移量 0x030

DMAALTSET 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位, 相应的  $\mu$ DMA 通道即会采用其副控制结构体。读取本寄存器可返回各通道使用哪一控制结构体。

### DMA 通道主副置位寄存器 (DMAALTSET)

基址 0x400F.F000

偏移量 0x030

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	SET[n]	R/W	0x0000.0000	通道 [n] 的副控制结构体置位

#### 值 描述

0  $\mu$ DMA 通道 [n] 使用主控制结构体。

1  $\mu$ DMA 通道 [n] 相应通道使用副控制结构体。

第 0 位对应于第 0 号通道, 依此类推。要想将本寄存器中的某个位清零, 需将 DMAALTCLR 寄存器的相应 CLR[n] 位置位。

**注意:** 对于乒乓模式和散聚模式,  $\mu$ DMA 控制器会自动将相应位置位, 选择副控制结构体。

### 寄存器 17: DMA 通道主副清除寄存器 (DMAALTCLR), 偏移量 0x034

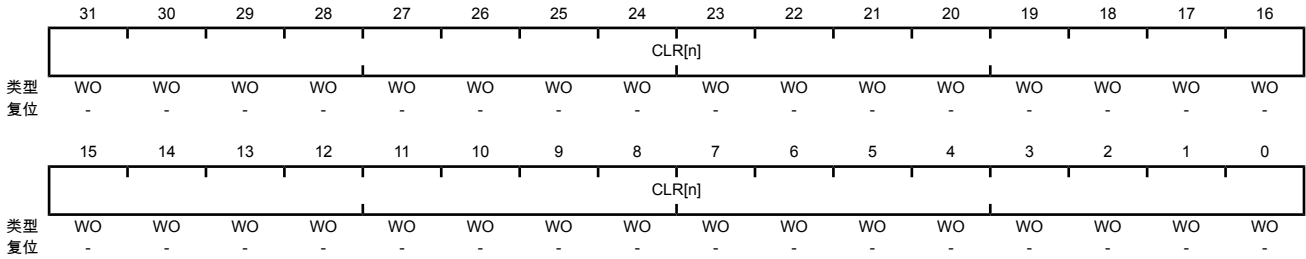
DMAALTCLR 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位, 则 DMAALTSET 寄存器中相应的 SET[n] 位即会清零。

#### DMA 通道主副清除寄存器 (DMAALTCLR)

基址 0x400F.F000

偏移量 0x034

类型 WO, 复位 -



位/域	名称	类型	复位	描述
31:0	CLR[n]	WO	-	通道 [n] 的副控制结构体清零
				<b>值 描述</b> 0 无影响。 1 将某个位置位, 即可将 DMAALTSET 寄存器中的 SET[n] 清零, 也就是说通道 [n] 正在使用主控制结构体。
				<b>注意:</b> 对于乒乓模式和散聚模式, $\mu$ DMA 控制器会自动将相应位置位, 选择副控制结构体。

## 寄存器 18: DMA 通道优先置位寄存器 ( DMAPRIOSET ) , 偏移量 0x038

DMAPRIOSET 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位，相应的  $\mu$ DMA 通道即会成为高优先级。读取本寄存器可获知各通道的优先级。

### DMA 通道优先置位寄存器 (DMAPRIOSET)

基址 0x400F.F000

偏移量 0x038

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SET[n]															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	SET[n]	R/W	0x0000.0000	通道 [n] 的优先级设置

#### 值 描述

0  $\mu$ DMA 通道 [n] 正在使用默认优先级。

1  $\mu$ DMA 通道 [n] 正在使用高优先级。

第 0 位对应于第 0 号通道，依此类推。要想将本寄存器中的某个位清零，需将 DMAPRIOCLR 寄存器的相应 CLR[n] 位置位。

### 寄存器 19: DMA 通道优先清零寄存器 ( DMAPRIOCLR ) , 偏移量 0x03C

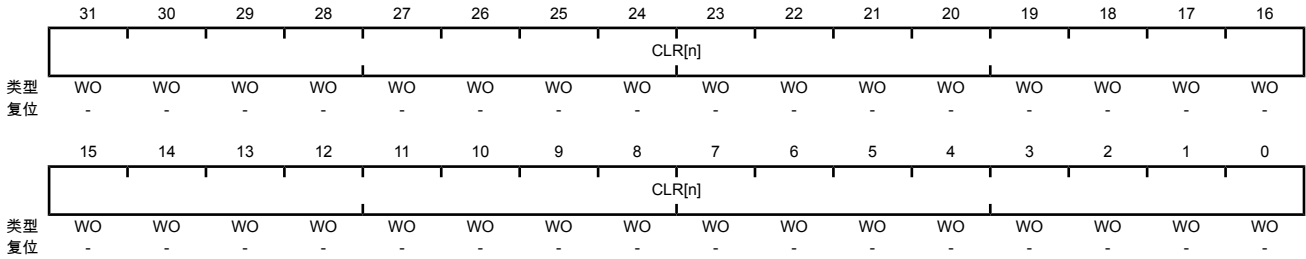
DMAPRIOCLR 寄存器的每个位分别对应一个  $\mu$ DMA 通道。将某个位置位, 则 DMAPRIOSET 寄存器中相应的 SET[n] 位即会清零。

#### DMA 通道优先清零寄存器 (DMAPRIOCLR)

基址 0x400F.F000

偏移量 0x03C

类型 WO, 复位 -



位/域	名称	类型	复位	描述
31:0	CLR[n]	WO	-	通道[n]的优先级清零
				<b>值 描述</b> 0 无影响。 1 将某个位置位, 即可将 DMAPRIOSET 寄存器中的 SET[n] 清零, 也就是说通道 [n] 正在使用默认优先级。

## 寄存器 20: DMA 总线错误清除寄存器 ( DMAERRCLR ) , 偏移量 0x04C

DMAERRCLR 寄存器用于读取并清除  $\mu$ DMA 总线错误状态。如果  $\mu$ DMA 控制器在执行传输过程中遭遇总线错误, 即将错误状态标志置位。如果在某个通道上发生了总线错误,  $\mu$ DMA 控制器会自动禁用该通道。其它通道不受影响, 仍然可以继续工作。

### DMA 总线错误清除寄存器 (DMAERRCLR)

基址 0x400F.F000

偏移量 0x04C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															ERRCLR
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	ERRCLR	R/W1C	0	<p><math>\mu</math>DMA 总线错误状态</p> <p>值 描述</p> <p>0 无挂起的总线错误。</p> <p>1 有挂起的总线错误。</p> <p>此位写 1 清零。</p>

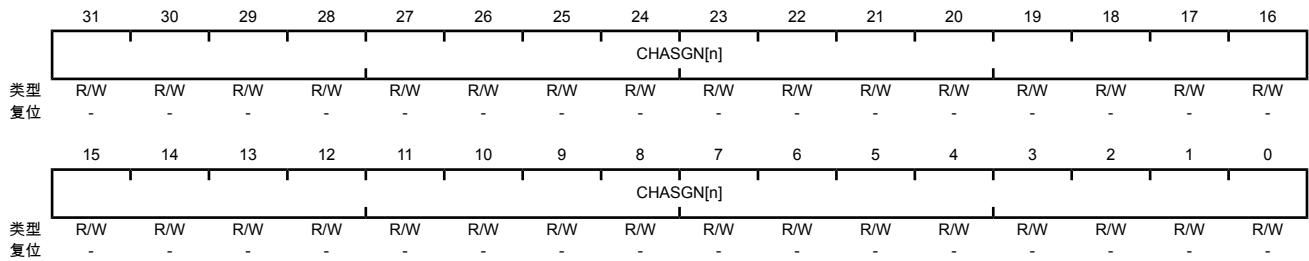
### 寄存器 21: DMA 通道分配寄存器 ( DMACHASGN ) , 偏移量 0x500

DMACHASGN 寄存器的每个位分别对应一个 μDMA 通道。将某个位置位，会选择表 9-1 在 536 页中规定的通道的次功能。

**注意:** 该寄存器用于支持传统软件。新软件应该使用 DMACHMAPn 寄存器。如果在该寄存器的某位清零，则在 DMACHMAPn 寄存器的相应位会被配置为 0x0。如果在该寄存器的某位置位，则相应位域会被配置为 0x1。在读取该寄存器时，如果相应的 DMACHMAPn 寄存器位域等于 0，则读取的位值为 0；否则，为 1 ( 如果相应的 DMACHMAPn 寄存器位域不为 0 )。

#### DMA 通道分配寄存器 (DMACHASGN)

基址 0x400F.F000  
偏移量 0x500  
类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:0	CHASGN[n]	R/W	-	通道 [n] 的分配选择
				值 描述
				0 按照通道的主功能工作。
				1 按照通道的次功能工作。

## 寄存器 22: DMA 通道中断状态寄存器 ( DMACHIS ) , 偏移量 0x504

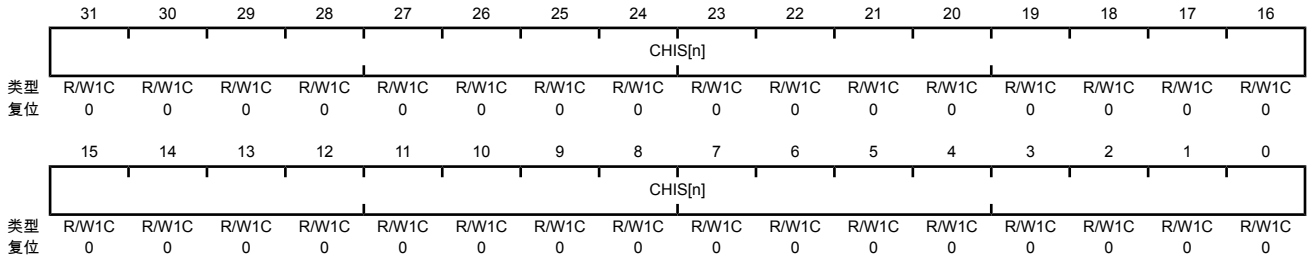
DMACHIS 寄存器的每个位分别对应一个  $\mu$ DMA 通道。当  $\mu$ DMA 通道产生了一个完成中断时, 该位置位。该位写 1 清零。

### DMA 通道中断状态寄存器 (DMACHIS)

基址 0x400F.F000

偏移量 0x504

类型 R/W1C, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:0	CHIS[n]	R/W1C	0x0000.0000	通道 [n] 的中断状态

#### 值 描述

1 相应的  $\mu$ DMA 通道产生中断。

0 相应的  $\mu$ DMA 通道没有产生中断。

此位写 1 清零。

### 寄存器 23: DMA 通道映射选择寄存器 0 ( DMACHMAP0 ) , 偏移量 0x510

DMACHMAP0 寄存器的每 4 位域用于配置表 9-1 在 536 页 所描述的 μDMA 通道分配。

注意: 要支持使用 DMA 通道分配 (DMACHASGN) 寄存器的传统软件, 值 0x0 等于清零的 DMACHASGN; 值 0x1 等于置位的 DMACHASGN 位。

#### DMA 通道映射选择寄存器 0 (DMACHMAP0)

基址 0x400F.F000  
偏移量 0x510  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CH7SEL				CH6SEL				CH5SEL				CH4SEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH3SEL				CH2SEL				CH1SEL				CH0SEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:28	CH7SEL	R/W	0x00	μDMA 通道 7 源选择 关于通道的分配参见表 9-1 在 536 页。
27:24	CH6SEL	R/W	0x00	μDMA 通道 6 源选择 关于通道的分配参见表 9-1 在 536 页。
23:20	CH5SEL	R/W	0x00	μDMA 通道 5 源选择 关于通道的分配参见表 9-1 在 536 页。
19:16	CH4SEL	R/W	0x00	μDMA 通道 4 源选择 关于通道的分配参见表 9-1 在 536 页。
15:12	CH3SEL	R/W	0x00	μDMA 通道 3 源选择 关于通道的分配参见表 9-1 在 536 页。
11:8	CH2SEL	R/W	0x00	μDMA 通道 2 源选择 关于通道的分配参见表 9-1 在 536 页。
7:4	CH1SEL	R/W	0x00	μDMA 通道 1 源选择 关于通道的分配参见表 9-1 在 536 页。
3:0	CH0SEL	R/W	0x00	μDMA 通道 0 源选择 关于通道的分配参见表 9-1 在 536 页。



## 寄存器 24: DMA 通道映射选择寄存器 1 ( DMACHMAP1 ) , 偏移量 0x514

DMACHMAP1 寄存器的每 4 位域用于配置表 9-1 在 536 页 所描述的  $\mu$ DMA 通道分配。

注意: 要支持使用 DMA 通道分配 (DMACHASGN) 寄存器的传统软件, 值 0x0 等于清零的 DMACHASGN; 值 0x1 等于置位的 DMACHASGN 位。

### DMA 通道映射选择寄存器 1 (DMACHMAP1)

基址 0x400F.F000

偏移量 0x514

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CH15SEL				CH14SEL				CH13SEL				CH12SEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH11SEL				CH10SEL				CH9SEL				CH8SEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:28	CH15SEL	R/W	0x00	$\mu$ DMA 通道 15 源选择 关于通道的分配参见表 9-1 在 536 页。
27:24	CH14SEL	R/W	0x00	$\mu$ DMA 通道 14 源选择 关于通道的分配参见表 9-1 在 536 页。
23:20	CH13SEL	R/W	0x00	$\mu$ DMA 通道 13 源选择 关于通道的分配参见表 9-1 在 536 页。
19:16	CH12SEL	R/W	0x00	$\mu$ DMA 通道 12 源选择 关于通道的分配参见表 9-1 在 536 页。
15:12	CH11SEL	R/W	0x00	$\mu$ DMA 通道 11 源选择 关于通道的分配参见表 9-1 在 536 页。
11:8	CH10SEL	R/W	0x00	$\mu$ DMA 通道 10 源选择 关于通道的分配参见表 9-1 在 536 页。
7:4	CH9SEL	R/W	0x00	$\mu$ DMA 通道 9 源选择 关于通道的分配参见表 9-1 在 536 页。
3:0	CH8SEL	R/W	0x00	$\mu$ DMA 通道 8 源选择 关于通道的分配参见表 9-1 在 536 页。

## 寄存器 25: DMA 通道映射选择寄存器 2 ( DMACHMAP2 ) , 偏移量 0x518

DMACHMAP2 寄存器的每 4 位域用于配置表 9-1 在 536 页 所描述的 μDMA 通道分配。

注意: 要支持使用 DMA 通道分配 (DMACHASGN) 寄存器的传统软件, 值 0x0 等于清零的 DMACHASGN; 值 0x1 等于置位的 DMACHASGN 位。

### DMA 通道映射选择寄存器 2 (DMACHMAP2)

基址 0x400F.F000  
偏移量 0x518  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CH23SEL				CH22SEL				CH21SEL				CH20SEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH19SEL				CH18SEL				CH17SEL				CH16SEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:28	CH23SEL	R/W	0x00	μDMA 通道 23 源选择 关于通道的分配参见表 9-1 在 536 页。
27:24	CH22SEL	R/W	0x00	μDMA 通道 22 源选择 关于通道的分配参见表 9-1 在 536 页。
23:20	CH21SEL	R/W	0x00	μDMA 通道 21 源选择 关于通道的分配参见表 9-1 在 536 页。
19:16	CH20SEL	R/W	0x00	μDMA 通道 20 源选择 关于通道的分配参见表 9-1 在 536 页。
15:12	CH19SEL	R/W	0x00	μDMA 通道 19 源选择 关于通道的分配参见表 9-1 在 536 页。
11:8	CH18SEL	R/W	0x00	μDMA 通道 18 源选择 关于通道的分配参见表 9-1 在 536 页。
7:4	CH17SEL	R/W	0x00	μDMA 通道 17 源选择 关于通道的分配参见表 9-1 在 536 页。
3:0	CH16SEL	R/W	0x00	μDMA 通道 16 源选择 关于通道的分配参见表 9-1 在 536 页。

## 寄存器 26: DMA 通道映射选择寄存器 3 ( DMACHMAP3 ) , 偏移量 0x51C

DMACHMAP3 寄存器的每 4 位域用于配置表 9-1 在 536 页 所描述的  $\mu$ DMA 通道分配。

注意: 要支持使用 DMA 通道分配 (DMACHASGN) 寄存器的传统软件, 值 0x0 等于清零的 DMACHASGN; 值 0x1 等于置位的 DMACHASGN 位。

### DMA 通道映射选择寄存器 3 (DMACHMAP3)

基址 0x400F.F000

偏移量 0x51C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CH31SEL				CH30SEL				CH29SEL				CH28SEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH27SEL				CH26SEL				CH25SEL				CH24SEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:28	CH31SEL	R/W	0x00	$\mu$ DMA 通道 31 源选择 关于通道的分配参见表 9-1 在 536 页。
27:24	CH30SEL	R/W	0x00	$\mu$ DMA 通道 30 源选择 关于通道的分配参见表 9-1 在 536 页。
23:20	CH29SEL	R/W	0x00	$\mu$ DMA 通道 29 源选择 关于通道的分配参见表 9-1 在 536 页。
19:16	CH28SEL	R/W	0x00	$\mu$ DMA 通道 28 源选择 关于通道的分配参见表 9-1 在 536 页。
15:12	CH27SEL	R/W	0x00	$\mu$ DMA 通道 27 源选择 关于通道的分配参见表 9-1 在 536 页。
11:8	CH26SEL	R/W	0x00	$\mu$ DMA 通道 26 源选择 关于通道的分配参见表 9-1 在 536 页。
7:4	CH25SEL	R/W	0x00	$\mu$ DMA 通道 25 源选择 关于通道的分配参见表 9-1 在 536 页。
3:0	CH24SEL	R/W	0x00	$\mu$ DMA 通道 24 源选择 关于通道的分配参见表 9-1 在 536 页。

## 寄存器 27: DMA 外设标识寄存器 0 ( DMAPeriphID0 ) , 偏移量 0xFE0

DMAPeriphIDn 寄存器均为硬编码寄存器，寄存器的位域决定复位值。

### DMA 外设标识寄存器 0 (DMAPeriphID0)

基址 0x400F.F000

偏移量 0xFE0

类型 RO, 复位 0x0000.0030

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID0	RO	0x30	μDMA 外设标识寄存器 [7:0] 软件可通过此寄存器实现标准的交叉外设识别。

## 寄存器 28: DMA 外设标识寄存器 1 ( DMAPeriphID1 ) , 偏移量 0xFE4

DMAPeriphIDn 寄存器均为硬编码寄存器, 寄存器的位域决定复位值。

## DMA 外设标识寄存器 1 (DMAPeriphID1)

基址 0x400F.F000

偏移量 0xFE4

类型 RO, 复位 0x0000.00B2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID1	RO	0xB2	μDMA 外设标识寄存器 [15:8] 软件可通过此寄存器实现标准的交叉外设识别。

## 寄存器 29: DMA 外设标识寄存器 2 ( DMAPeriphID2 ) , 偏移量 0xFE8

DMAPeriphIDn 寄存器均为硬编码寄存器，寄存器的位域决定复位值。

### DMA 外设标识寄存器 2 (DMAPeriphID2)

基址 0x400F.F000  
 偏移量 0xFE8  
 类型 RO, 复位 0x0000.000B

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID2	RO	0x0B	μDMA 外设 ID 寄存器 [23:16] 软件可通过此寄存器实现标准的交叉外设识别。

## 寄存器 30: DMA 外设标识寄存器 3 ( DMAPeriphID3 ) , 偏移量 0xFEC

DMAPeriphIDn 寄存器均为硬编码寄存器，寄存器的位域决定复位值。

## DMA 外设标识寄存器 3 (DMAPeriphID3)

基址 0x400F.F000

偏移量 0xFEC

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

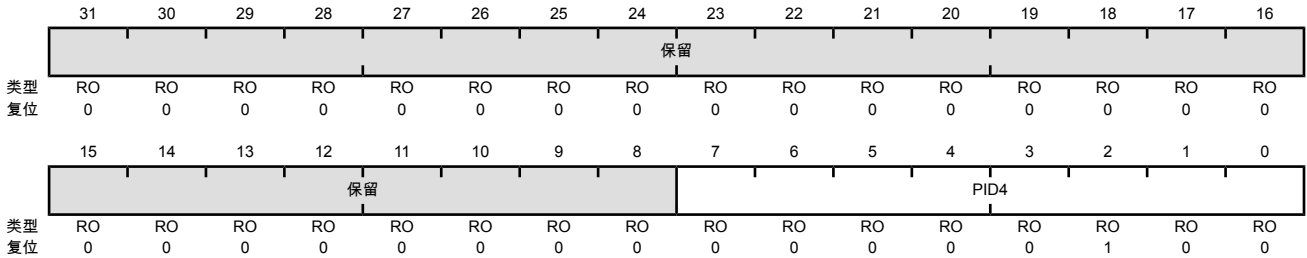
位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID3	RO	0x00	μDMA 外设标识寄存器 [31:24] 软件可通过此寄存器实现标准的交叉外设识别。

### 寄存器 31: DMA 外设标识寄存器 4 ( DMAPeriphID4 ) , 偏移量 0xFD0

DMAPeriphIDn 寄存器均为硬编码寄存器，寄存器的位域决定复位值。

#### DMA 外设标识寄存器 4 (DMAPeriphID4)

基址 0x400F.F000  
 偏移量 0xFD0  
 类型 RO, 复位 0x0000.0004



位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID4	RO	0x04	μDMA 外设 ID 寄存器 软件可通过此寄存器实现标准的交叉外设识别。



## 寄存器 32: DMA PrimeCell 标识寄存器 0 ( DMAPCellID0 ) , 偏移量 0xFF0

DMAPCellIDn 寄存器均为硬编码寄存器，寄存器的位域决定复位值。

## DMA PrimeCell 标识寄存器 0 ( DMAPCellID0 )

基址 0x400F.F000

偏移量 0xFF0

类型 RO, 复位 0x0000.000D

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID0	RO	0x0D	μDMA PrimeCell 标识寄存器 [7:0] 为软件提供一个标准的跨外设标识系统。

### 寄存器 33: DMA PrimeCell 标识寄存器 1 ( DMAPCellID1 ) , 偏移量 0xFF4

DMAPCellIDn 寄存器均为硬编码寄存器，寄存器的位域决定复位值。

#### DMA PrimeCell 标识寄存器 1 (DMAPCellID1)

基址 0x400F.F000

偏移量 0xFF4

类型 RO, 复位 0x0000.00F0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID1	RO	0xF0	μDMA PrimeCell 标识寄存器 [15:8] 为软件提供一个标准的跨外设标识系统。

## 寄存器 34: DMA PrimeCell 标识寄存器 2 ( DMAPCellID2 ) , 偏移量 0xFF8

DMAPCellIDn 寄存器均为硬编码寄存器，寄存器的位域决定复位值。

## DMA PrimeCell 标识寄存器 2 ( DMAPCellID2 )

基址 0x400F.F000

偏移量 0xFF8

类型 RO, 复位 0x0000.0005

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID2	RO	0x05	μDMA PrimeCell ID 寄存器 [23:16] 为软件提供一个标准的跨外设标识系统。

### 寄存器 35: DMA PrimeCell 标识寄存器 3 ( DMAPCellID3 ) , 偏移量 0xFFC

DMAPCellIDn 寄存器均为硬编码寄存器，寄存器的位域决定复位值。

#### DMA PrimeCell 标识寄存器 3 ( DMAPCellID3 )

基址 0x400F.F000  
 偏移量 0xFFC  
 类型 RO, 复位 0x0000.00B1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID3	RO	0xB1	μDMA PrimeCell 标识寄存器 [31:24] 为软件提供一个标准的跨外设标识系统。

## 10 通用输入/输出端口 (GPIO)

GPIO 模块包括 14 个物理 GPIO 模块，每个模块对应一个单独的 GPIO 端口（端口 A、端口 B、端口 C、端口 D、端口 E、端口 F、端口 G、端口 H、端口 J、端口 K、端口 L、端口 M、端口 N、端口 P）。GPIO 模块支持高达 105 个可编程的输入/输出管脚。具体取决于使用的外设。

GPIO 模块具有以下特性：

- 高达 105 个 GPIO，具体取决于配置
- 高度灵活的管脚复用，可配置为 GPIO 或任一外设功能
- 配置为输入模式可承受 5 V 电压
- 两种方式访问 GPIO 端口：或者通过具有更好连续访问性能的高级高性能总线 (AHB)；或者通过传统的高级外设总线 (APB)，使用现有的 A-H 和 J 端口的 AHB 访问；K-N 和 P 端口通过
- AHB 端口每个时钟周期的快速切换能力；APB 端口每两个时钟周期的快速切换能力
- 可编程控制 GPIO 中断
  - 屏蔽中断发生
  - 边沿触发（上升沿和/或下降沿）
  - （高或低）电平触发
  - 在端口 P 上，每个管脚中断都可用
- 在读和写操作中通过地址线进行位屏蔽
- 可用于启动一个 ADC 采样序列或  $\mu$ DMA 传输
- 在休眠模式中，可以保持管脚的状态
- 配置为数字输入的管脚均为施密特触发
- 可编程控制 GPIO 管脚配置
  - 弱上拉或下拉电阻
  - 数字通信时可配置为 2 mA、4 mA 或 8 mA 驱动电流；对于需要大电流的应用，最多四个管脚承载 18 mA
  - 8 mA 驱动电流的斜率控制
  - 开漏启用
  - 数字输入启用

### 10.1 信号描述

GPIO 信号具有复用硬件功能。下面的表格列出了所有 GPIO 管脚及其模拟和数字复用功能。当配置成输入时，除了 PB0 和 PB1 最高可承受 3.6 V 电压，其他所有 GPIO 管脚都可以承受 5 V 电压。将 GPIO 备用功能选择 (GPIOAFSEL) 和 GPIODEN 寄存器中相应的位置位并对 GPIO 端口控制 (GPIOCTL) 寄存器中的 PMCx 进行配置（如下表所示的数字编码），可以启用数字复用硬件功

能。下表中的模拟信号也能耐受 5 V 电压，通过清零 GPIO 数字使能 (GPIOEN) 寄存器的 DEN 位可对其进行配置。AINx 模拟信号所具备的内部电路能确保他们不会超过  $V_{DD}$  的电压（低于表 25-1 在 1291 页规定的最大值），但模拟性能规范仅适用于以下条件：I/O 管脚的输入信号在  $0 V < V_{IN} < V_{DD}$  范围之内。请注意，每个管脚必须单独编程；表格中的列并没有任何分组的意思。表中的灰色单元格代表相应 GPIO 管脚的默认值。

**重要：** 所有的 GPIO 管脚在复位时都被配置为 GPIO 功能，而且是三态的，即 (GPIOAFSEL=0、GPIOEN=0、GPIOPDR=0、GPIOPUR=0、GPIOPCTL=0)，但是下表中列出的这些管脚除外。上电复位 ( $\overline{POR}$ ) 或是外部复位  $\overline{RST}$  会让这些管脚返回默认值。

表 10-1. 具有非 0 复位值的 GPIO 管脚

GPIO 管脚	默认状态	GPIOAFSEL	GPIOEN	GPIOPDR	GPIOPUR	GPIOPCTL
PA[1:0]	UART0	0	0	0	0	0x1
PA[5:2]	SSI0	0	0	0	0	0x2
PB[3:2]	I <sup>2</sup> C0	0	0	0	0	0x3
PC[3:0]	JTAG/SWD	1	1	0	1	0x1

GPIO 提交控制寄存器提供了保护层，以防止对重要硬件信号的意外编程，其中包括 JTAG/SWD 信号和 NMI 信号。就算是不配置成 JTAG/SWD 或 NMI 信号，而是把他们配置成复用功能，这些管脚也必须遵循提交控制过程，请参考“提交控制”在 604 页。

表 10-2. GPIO 管脚和复用功能 (144LQFP)

IO	管脚	模拟功能	数字功能 ( GPIOPCTL PMCx 位域编码 ) <sup>a</sup>											
			1	2	3	4	5	6	7	8	9	14	15	
PA0	37	-	U0Rx	-	-	-	-	-	-	-	CAN1Rx	-	-	-
PA1	38	-	U0Tx	-	-	-	-	-	-	-	CAN1Tx	-	-	-
PA2	39	-	-	SSI0Clk	-	-	-	-	-	-	-	-	-	-
PA3	40	-	-	SSI0Fss	-	-	-	-	-	-	-	-	-	-
PA4	41	-	-	SSI0Rx	-	-	-	-	-	-	-	-	-	-
PA5	42	-	-	SSI0Tx	-	-	-	-	-	-	-	-	-	-
PA6	45	-	-	-	I2C1SCL	-	M1PWM2	-	-	-	-	-	-	-
PA7	46	-	-	-	I2C1SDA	-	M1PWM3	-	-	-	-	-	-	-
PB0	97	USB0ID	U1Rx	-	-	-	-	-	-	T2CCP0	-	-	-	-
PB1	98	USB0BUS	U1Tx	-	-	-	-	-	-	T2CCP1	-	-	-	-
PB2	99	-	-	-	I2C0SCL	-	-	-	-	T3CCP0	-	-	-	-
PB3	100	-	-	-	I2C0SDA	-	-	-	-	T3CCP1	-	-	-	-
PB4	136	AIN10	-	SSI2Clk	-	M0PWM2	-	-	-	T1CCP0	CAN0Rx	-	-	-
PB5	135	AIN11	-	SSI2Fss	-	M0PWM3	-	-	-	T1CCP1	CAN0Tx	-	-	-
PC0	118	-	TCK SWCLK	-	-	-	-	-	-	T4CCP0	-	-	-	-
PC1	117	-	TMS SWDIO	-	-	-	-	-	-	T4CCP1	-	-	-	-
PC2	116	-	TDI	-	-	-	-	-	-	T5CCP0	-	-	-	-
PC3	115	-	TDO SWO	-	-	-	-	-	-	T5CCP1	-	-	-	-
PC4	36	C1-	U4Rx	U1Rx	-	M0PWM6	-	IDX1	WT0CCP0	U1RTS	-	-	-	-
PC5	35	C1+	U4Tx	U1Tx	-	M0PWM7	-	PhA1	WT0CCP1	U1CTS	-	-	-	-

表 10-2. GPIO 管脚和复用功能 (144LQFP) (续)

IO	管脚	模拟功能	数字功能 (GPIOCTL PMCx 位域编码) <sup>a</sup>										
			1	2	3	4	5	6	7	8	9	14	15
PC6	34	C0+	U3Rx	-	-	-	-	PhB1	WT1CCP0	USB0EPEN	-	-	-
PC7	33	C0-	U3Tx	-	-	-	-	-	WT1CCP1	USB0PFLT	-	-	-
PD0	1	AIN15	SSI3Clk	SSI1Clk	I2C3SCL	M0PWM6	M1PWM0	-	WT2CCP0	-	-	-	-
PD1	2	AIN14	SSI3Fss	SSI1Fss	I2C3SDA	M0PWM7	M1PWM1	-	WT2CCP1	-	-	-	-
PD2	3	AIN13	SSI3Rx	SSI1Rx	-	M0FAULT0	-	-	WT3CCP0	USB0EPEN	-	-	-
PD3	4	AIN12	SSI3Tx	SSI1Tx	-	-	-	IDX0	WT3CCP1	USB0PFLT	-	-	-
PD4	141	AIN7	U6Rx	-	-	-	-	-	WT4CCP0	-	-	-	-
PD5	142	AIN6	U6Tx	-	-	-	-	-	WT4CCP1	-	-	-	-
PD6	143	AIN5	U2Rx	-	-	M0FAULT0	-	PhA0	WT5CCP0	-	-	-	-
PD7	144	AIN4	U2Tx	-	-	M0FAULT1	-	PhB0	WT5CCP1	NMI	-	-	-
PE0	15	AIN3	U7Rx	-	-	-	-	-	-	-	-	-	-
PE1	14	AIN2	U7Tx	-	-	-	-	-	-	-	-	-	-
PE2	13	AIN1	-	-	-	-	-	-	-	-	-	-	-
PE3	12	AIN0	-	-	-	-	-	-	-	-	-	-	-
PE4	139	AIN9	U5Rx	-	I2C2SCL	M0PWM4	M1PWM2	-	-	CAN0Rx	-	-	-
PE5	140	AIN8	U5Tx	-	I2C2SDA	M0PWM5	M1PWM3	-	-	CAN0Tx	-	-	-
PE6	133	AIN21	-	-	-	-	-	-	-	CAN1Rx	-	-	-
PE7	134	AIN20	U1RI	-	-	-	-	-	-	CAN1Tx	-	-	-
PF0	62	-	U1RTS	SSI1Rx	CAN0Rx	-	M1PWM4	PhA0	T0CCP0	NMI	C0o	TRD2	-
PF1	63	-	U1CTS	SSI1Tx	-	-	M1PWM5	PhB0	T0CCP1	-	C1o	TRD1	-
PF2	64	-	U1DCD	SSI1Clk	-	M0FAULT0	M1PWM6	-	T1CCP0	-	C2o	TRD0	-
PF3	65	-	U1DSR	SSI1Fss	CAN0Tx	M0FAULT1	M1PWM7	-	T1CCP1	-	-	TRCLK	-
PF4	61	-	U1DTR	-	-	M0FAULT2	M1FAULT0	IDX0	T2CCP0	USB0EPEN	-	TRD3	-
PF5	60	-	-	-	-	M0FAULT3	-	-	T2CCP1	USB0PFLT	-	-	-
PF6	59	-	-	-	I2C2SCL	-	-	-	T3CCP0	-	-	-	-
PF7	58	-	-	-	I2C2SDA	-	M1FAULT0	-	T3CCP1	-	-	-	-
PG0	55	-	-	-	I2C3SCL	-	M1FAULT1	PhA1	T4CCP0	-	-	-	-
PG1	54	-	-	-	I2C3SDA	-	M1FAULT2	PhB1	T4CCP1	-	-	-	-
PG2	53	-	-	-	I2C4SCL	M0FAULT1	M1PWM0	-	T5CCP0	-	-	-	-
PG3	52	-	-	-	I2C4SDA	M0FAULT2	M1PWM1	PhA1	T5CCP1	-	-	-	-
PG4	51	-	U2Rx	-	I2C1SCL	M0PWM4	M1PWM2	PhB1	WT0CCP0	USB0EPEN	-	-	-
PG5	50	-	U2Tx	-	I2C1SDA	M0PWM5	M1PWM3	IDX1	WT0CCP1	USB0PFLT	-	-	-
PG6	48	-	-	-	I2C5SCL	M0PWM6	-	-	WT1CCP0	-	-	-	-
PG7	47	-	-	-	I2C5SDA	M0PWM7	IDX1	-	WT1CCP1	-	-	-	-
PH0	32	-	-	SSI3Clk	-	M0PWM0	-	M0FAULT0	WT2CCP0	-	-	-	-
PH1	31	-	-	SSI3Fss	-	M0PWM1	IDX0	M0FAULT1	WT2CCP1	-	-	-	-
PH2	28	-	-	SSI3Rx	-	M0PWM2	-	M0FAULT2	WT5CCP0	-	-	-	-
PH3	27	-	-	SSI3Tx	-	M0PWM3	-	M0FAULT3	WT5CCP1	-	-	-	-
PH4	26	-	-	SSI2Clk	-	M0PWM4	PhA0	-	WT3CCP0	-	-	-	-
PH5	23	-	-	SSI2Fss	-	M0PWM5	PhB0	-	WT3CCP1	-	-	-	-
PH6	22	-	-	SSI2Rx	-	M0PWM6	-	-	WT4CCP0	-	-	-	-

表 10-2. GPIO 管脚和复用功能 (144LQFP) (续)

IO	管脚	模拟功能	数字功能 ( GPIOCTL PMCx 位域编码 ) <sup>a</sup>										
			1	2	3	4	5	6	7	8	9	14	15
PH7	21	-	-	SSI2Tx	-	MOPWM7	-	-	WT4CCP1	-	-	-	-
PJ0	120	-	U4Rx	-	-	-	-	-	T1CCP0	-	-	-	-
PJ1	121	-	U4Tx	-	-	-	-	-	T1CCP1	-	-	-	-
PJ2	122	-	U5Rx	-	-	-	IDX0	-	T2CCP0	-	-	-	-
PJ3	123	-	U5Tx	-	-	-	-	-	T2CCP1	-	-	-	-
PJ4	127	C2+	U6Rx	-	-	-	-	-	T3CCP0	-	-	-	-
PJ5	128	C2-	U6Tx	-	-	-	-	-	T3CCP1	-	-	-	-
PJ6	129	-	-	-	-	-	-	-	-	-	-	-	-
PJ7	130	-	-	-	-	-	-	-	-	-	-	-	-
PK0	16	AIN16	-	SSI3Clk	-	-	-	M1FAULT0	-	-	-	-	-
PK1	17	AIN17	-	SSI3Fss	-	-	-	M1FAULT1	-	-	-	-	-
PK2	18	AIN18	-	SSI3Rx	-	-	-	M1FAULT2	-	-	-	-	-
PK3	19	AIN19	-	SSI3Tx	-	-	-	M1FAULT3	-	-	-	-	-
PK4	112	-	U7Rx	-	-	-	-	M0FAULT0	RTCCLK	C0o	-	-	-
PK5	111	-	U7Tx	-	-	-	-	M0FAULT1	-	C1o	-	-	-
PK6	110	-	-	-	-	-	-	M0FAULT2	WT1CCP0	C2o	-	-	-
PK7	109	-	-	-	-	-	-	M0FAULT3	WT1CCP1	-	-	-	-
PL0	108	-	-	-	-	-	-	T0CCP0	WT0CCP0	-	-	-	-
PL1	107	-	-	-	-	-	-	T0CCP1	WT0CCP1	-	-	-	-
PL2	106	-	-	-	-	-	-	T1CCP0	WT1CCP0	-	-	-	-
PL3	105	-	-	-	-	-	-	T1CCP1	WT1CCP1	-	-	-	-
PL4	104	-	-	-	-	-	-	T2CCP0	WT2CCP0	-	-	-	-
PL5	103	-	-	-	-	-	-	T2CCP1	WT2CCP1	-	-	-	-
PL6	96	USB0DP	-	-	-	-	-	T3CCP0	WT3CCP0	-	-	-	-
PL7	95	USB0DM	-	-	-	-	-	T3CCP1	WT3CCP1	-	-	-	-
PM0	89	-	-	-	-	-	-	T4CCP0	WT4CCP0	-	-	-	-
PM1	88	-	-	-	-	-	-	T4CCP1	WT4CCP1	-	-	-	-
PM2	87	-	-	-	-	-	-	T5CCP0	WT5CCP0	-	-	-	-
PM3	86	-	-	-	-	-	-	T5CCP1	WT5CCP1	-	-	-	-
PM4	85	-	-	-	-	-	-	-	-	-	-	-	-
PM5	84	-	-	-	-	-	-	-	-	-	-	-	-
PM6	83	-	-	MOPWM4	-	-	-	-	WT0CCP0	-	-	-	-
PM7	82	-	-	MOPWM5	-	-	-	-	WT0CCP1	-	-	-	-
PN0	81	-	CAN0Rx	-	-	-	-	-	-	-	-	-	-
PN1	80	-	CAN0Tx	-	-	-	-	-	-	-	-	-	-
PN2	20	-	-	MOPWM6	-	-	-	-	WT2CCP0	-	-	-	-
PN3	119	-	-	MOPWM7	-	-	-	-	WT2CCP1	-	-	-	-
PN4	71	-	-	M1PWM4	-	-	-	-	WT3CCP0	-	-	-	-
PN5	70	-	-	M1PWM5	-	-	-	-	WT3CCP1	-	-	-	-
PN6	69	-	-	M1PWM6	-	-	-	-	WT4CCP0	-	-	-	-
PN7	68	-	-	M1PWM7	-	-	-	-	WT4CCP1	-	-	-	-



表 10-2. GPIO 管脚和复用功能 (144LQFP) (续)

IO	管脚	模拟功能	数字功能 (GPIOCTL PMCx 位域编码) <sup>a</sup>											
			1	2	3	4	5	6	7	8	9	14	15	
PP0	131	AIN23	MOPWM0	-	-	-	-	-	-	T4CCP0	-	-	-	-
PP1	132	AIN22	MOPWM1	-	-	-	-	-	-	T4CCP1	-	-	-	-
PP2	11	-	MOPWM2	-	-	-	-	-	-	T5CCP0	-	-	-	-

a. 带灰色阴影的数字信号是相应 GPIO 管脚的上电默认值。本器件不使用编码 10-13。

## 10.2 功能描述

每个 GPIO 端口都是同一物理模块的独立硬件实例 (请参考图 10-1 在 601 页和图 10-2 在 602 页)。LM4F232H5QD 微控制器包含 14 个端口, 因此会有 14 个此种物理 GPIO 模块。请注意, 并非每个模块都实施了所有的管脚。对于片内外设模块来说, 一些 GPIO 管脚可作为 I/O 信号使用。GPIO 管脚的复用硬件功能请参考表 23-5 在 1280 页。

图 10-1. 数字 I/O 口

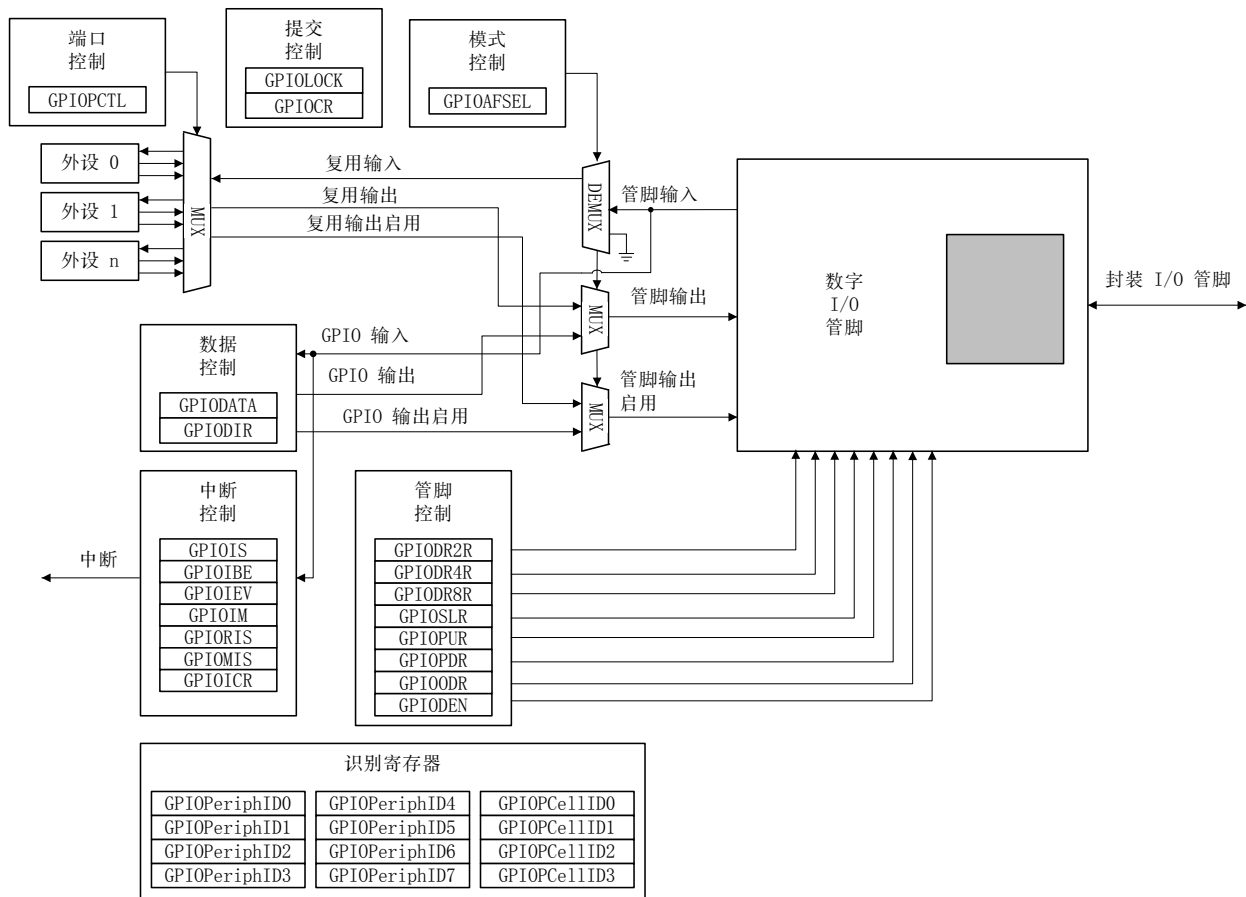
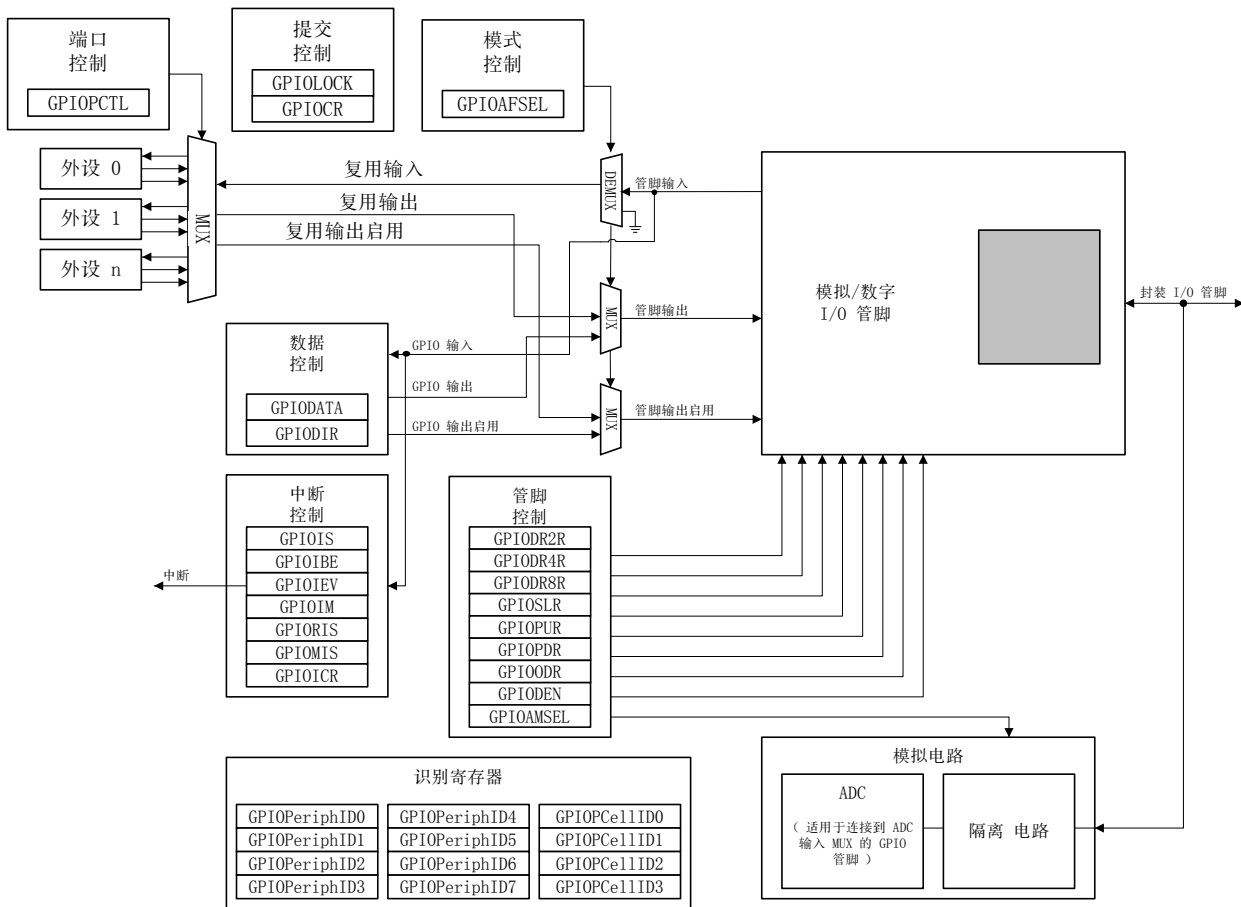


图 10-2. 模拟/数字 I/O 口



### 10.2.1 数据控制

数据控制寄存器允许软件配置 GPIO 的操作模式。当数据寄存器捕获输入数据时，数据方向寄存器将 GPIO 配置为输入；当数据寄存器通过端口输出数据时，数据方向寄存器将 GPIO 配置为输出。

小心 - 用户可以建立一个软件序列来阻止调试器连接到 **Stellaris**<sup>®</sup> 微控制器的特性。如果将程序代码加载到 **Flash** 中会立即将 **JTAG** 管脚变成其 **GPIO** 功能，那么在 **JTAG** 管脚功能切换之前，调试器将没有足够的时间去连接和终止控制器。结果调试器可能被锁定在该部分外。为了避免这个问题，可以使用一个基于外部或软件触发器的软件程序来恢复 **JTAG** 功能。

#### 10.2.1.1 数据方向操作

GPIO 方向 (GPIODIR) 寄存器 (请参考 610 页) 用来将每个独立的管脚配置为输入或输出。当数据方向寄存器里的位被清零时被配置为输入，相应的数据寄存器位便可以捕获并储存 GPIO 端口的值。当数据方向寄存器里的位被置位时被配置为输出，数据寄存器里相应的位便可以驱动 GPIO 端口。

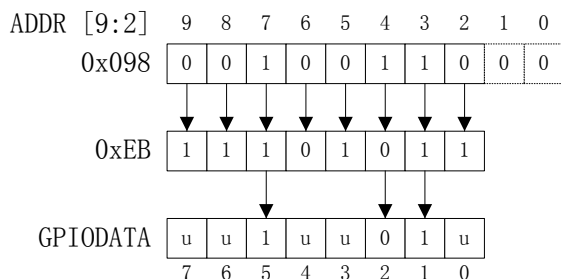
#### 10.2.1.2 数据寄存器的操作

为了提高软件的效率，通过将地址总线的位 [9:2] 用作屏蔽位，可以对 GPIO 端口的 GPIO 数据 (GPIODATA) 寄存器 (请参考 609 页) 中的各个位进行修改。通过这种方式软件驱动程序就可以以一条指令修改任何一个 GPIO 管脚，而不影响其他管脚的状态。这种方式与通过“读-修改-写”来操作 GPIO 管脚的典型做法不同。为了实现这种功能，GPIODATA 寄存器覆盖了存储器映射中的 256 个单元。

在写入操作中，如果与数据位相关联的地址位被置位，那么 GPIODATA 寄存器的值将发生变化。如果地址位被清零，那么数据位保持不变。

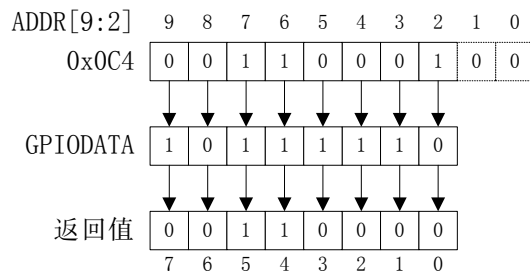
例如，将 0xEB 写入地址 GPIODATA + 0x98 处，结果将如图 10-3 所示。其中，u 表示写入操作没有改变数据。

图 10-3. GPIODATA 写入实例



在读操作过程中，如果与数据位相关联的地址位被置位，那么就可以读取到数据寄存器里的值。如果与数据位相关联的地址位被清零，那么不管数据寄存器里实际值是什么都读做0。例如，读取地址 GPIODATA+0x0C4 处的值，结果如图 10-4 所示。

图 10-4. GPIODATA 读取实例



## 10.2.2 中断控制

每个 GPIO 端口的中断能力都由 7 个寄存器控制。这些寄存器可以用于选择中断源、极性以及边沿属性。当一个或多个输入引发中断时，只有一个中断输出被送到整个 GPIO 端口的中断控制器。对于边沿触发，为了让进一步的中断可用，软件必须清除该中断。对于电平触发，必须保持住外部电平的状态才能使控制器识别中断的发生。

以下三个寄存器用来定义中断触发的边沿或感知：

- GPIO 中断检测 (GPIOIS) 寄存器 ( 请参考 611页 )
- GPIO 中断双边沿 (GPIOIBE) 寄存器 ( 请参考 612页 )
- GPIO 中断事件 (GPIOIEV) 寄存器 ( 请参考 613页 )

通过 GPIO 中断屏蔽 (GPIOIM) 寄存器 ( 请参考 614页 ) 可以启用/禁用中断。

当产生中断条件时，可以在 GPIO 原始中断状态 (GPIORIS) 和 GPIO 屏蔽后的中断状态 (GPIOMIS) 寄存器中观察到中断信号的状态 ( 请参考 615页 和 616页 )。顾名思义，GPIOMIS 寄存器仅显示允许被传送到中断控制器的中断条件。GPIORIS 寄存器则表示 GPIO 管脚满足的中断条件，但不一定发送到控制器。

向 GPIO 中断清零 (GPIOICR) 寄存器 ( 请参考 617页 ) 中的位写 1 可以清除相应的中断。

在设置中断控制寄存器 ( GPIOIS、GPIOIBE 或 GPIOIEV) 时, 应该保持中断的屏蔽状态 ( GPIOIM 清零 )。如果相应的位没有屏蔽, 那么向中断控制寄存器中写入任何值都有可能产生伪中断。

### 10.2.2.1 每个管脚的中断

GPIO PortP 的每个管脚可以捕获一个系统中断。每个管脚有一个专用中断向量, 可以被单独的中断处理器处理。PP0 和 PQ0 中断是主中断, 可以提供传统的组合中断版本。关于中断分配的详细信息, 请参考表 2-9 在 101页。

### 10.2.2.2 ADC 触发源

任何 GPIO 管脚都可以通过 GPIO ADC 控制 (GPIOADCCTL) 寄存器配置成 ADC 的外部触发源。如果 GPIO 被配置为非屏蔽的中断管脚 ( GPIOIM 中相应的位被置位 ), 该端口产生中断时, 就会发送一个触发信号到 ADC。此时如果 ADC 事件复用选择 (ADCEMUX) 寄存器被配置为使用外部触发器, 那么就会启动 ADC 转换。请参考 773页。

请注意, 如果 Port B GPIOADCCTL 寄存器被清零, PB4 也可以用作 ADC 的外部触发信号。通过这种传统模式, 用户可以向 Stellaris 以前的设备写入代码, 以在该微控制器中工作。

### 10.2.2.3 $\mu$ DMA 触发源

任何 GPIO 管脚都可以通过 GPIO DMA 控制 (GPIODMACTL) 寄存器配置成  $\mu$ DMA 的外部触发源。如果 GPIO 被配置为非屏蔽的中断管脚 ( GPIOIM 中相应的位被置位 ), 该端口产生中断, 并发送一个外部的触发信号到  $\mu$ DMA 一个 dma\_req 信号到  $\mu$ DMA。如果  $\mu$ DMA 配置为根据 GPIO 信号开始传输数据, 那么此时就会启动传输。完成传输时, dma\_done 信号从  $\mu$ DMA 发送到 GPIO 并在 GPIORIS 寄存器中报告为 DMA ( 完成 ) 中断。

### 10.2.3 模式控制

GPIO 管脚既可以被软件控制也可以被硬件控制。大部分的管脚默认是软件控制, 此时 GPIODATA 寄存器用来读写相应的管脚。当 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 ( 请参考 618页 ) 启用硬件控制时, 管脚状态将由它的复用 ( 即外设 ) 功能控制。

更多的管脚复用功能选择由 GPIO 端口控制 (GPIOPCTL) 寄存器提供, 该寄存器可以为每个 GPIO 选择其中一个外设功能。关于这些配置的详细信息, 请参阅表 23-5 在 1280页。

注意: 如果一个管脚被用作 ADC 的输入, 那么 GPIOAMSEL 寄存器中相应的位必须置位, 禁用模拟隔离电路。

### 10.2.4 提交控制

GPIO 确认控制寄存器提供的保护功能可防止对重要硬件外设的意外编程。为以下 GPIO 管脚提供保护: 能用作 JTAG/SWD 管脚 (PC[3:0]) 和 NMI 管脚 ( PD7 和 PF0 )。向 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 ( 请参阅 618页 )、GPIO 上拉电阻选择 (GPIOPUR) 寄存器 ( 请参阅 624页 )、GPIO 下拉电阻选择 (GPIOPDR) 寄存器 ( 请参阅 626页 ) 以及 GPIO 数字使能 (GPIODEN) 寄存器 ( 请参阅 629页 ) 中受保护的位写入数据将不会确认保存, 除非 GPIO 锁定 (GPIOLOCK) 寄存器 ( 请参阅 631页 ) 没有被锁定, 同时 GPIO 确认 (GPIOCR) 寄存器 ( 请参阅 632页 ) 中相应的位被置位。

### 10.2.5 管脚控制

可以根据应用程序的要求用软件来配置 GPIO 管脚。管脚控制寄存器包括 GPIODR2R、GPIODR4R、GPIODR8R、GPIODR、GPIOPUR、GPIOPDR、GPIOSLR 以及 GPIODEN 寄存器。这些寄存器控制着管脚的驱动电流大小、开漏配置、上拉下拉电阻选择、斜率控制和数字输入启用。

## 10.2.6 标识

复位时配置的标识寄存器允许软件将模块当作GPIO块进行检测和识别。标识寄存器包括 GPIOPeriphID0-GPIOPeriphID7 寄存器以及 GPIOCellID0-GPIOCellID3 寄存器。

## 10.3 初始化及配置

GPIO 模块可以通过两个不同的存储器槽访问。比较老的一种称为先进外设总线 (APB)，向后兼容以前的 Stellaris 产品。另外一种先进高端总线 (AHB)，它和 APB 一样拥有相同的寄存器映射，但是提供了比 APB 更好的访问性能。但是这两种访问方式只能选择一种使用。为指定 GPIO 端口启用的槽由 GPIOHBCTL 寄存器 ( 请参考 240页 ) 中相应的位来控制。注：GPIO 端口 K、L、M、N 和 P 只可以通过 AHB 槽访问。

要使用特定 GPIO 端口的管脚，必须先将 RCGCGPIO 寄存器 ( 请参考 302页 ) 中相应的 GPIO 端口位域 (GPIO<sub>n</sub>) 置位，以启用该端口的时钟信号。

除非另行配置，内部上电复位时，所有的 GPIO 管脚都被配置成无驱动模式 ( 三态 )：除了表 10-1 在 598页 中显示的管脚，GPIOAFSEL=0、GPIODEN=0、GPIOPDR=0 且 GPIOPUR=0。表 10-3 在 605页 列出了 GPIO 端口的所有可能的配置以及实现这些配置的控制寄存器设置。表 10-4 在 605页 显示了为 GPIO 端口的管脚 2 配置上升沿中断的方法。

表 10-3. GPIO 端口配置示例

配置	GPIO 寄存器位值 <sup>a</sup>									
	AFSEL	DIR	ODR	DEN	PUR	PDR	DR2R	DR4R	DR8R	SLR
数字输入 (GPIO)	0	0	0	1	?	?	X	X	X	X
数字输出 (GPIO)	0	1	0	1	?	?	?	?	?	?
开漏输出 (GPIO)	0	1	1	1	X	X	?	?	?	?
开漏输入/输出 (I2CSDA)	1	X	1	1	X	X	?	?	?	?
数字输入 ( 定时器 CCP )	1	X	0	1	?	?	X	X	X	X
数字输入 (QEI)	1	X	0	1	?	?	X	X	X	X
数字输出 (PWM)	1	X	0	1	?	?	?	?	?	?
数字输出 (定时器 PWM)	1	X	0	1	?	?	?	?	?	?
数字输入/输出 (SSI)	1	X	0	1	?	?	?	?	?	?
数字输入/输出 (UART)	1	X	0	1	?	?	?	?	?	?
模拟输入 ( 比较器 )	0	0	0	0	0	0	X	X	X	X
数字输出 ( 比较器 )	1	X	0	1	?	?	?	?	?	?

a. X = 忽略 ( 无关位 )

? = 可以是 0 或 1，具体取决于配置。

表 10-4. GPIO 中断配置示例

寄存器	期望的中断事件触发	管脚 2 位的值 <sup>a</sup>							
		7	6	5	4	3	2	1	0
GPIOIS	0-边沿触发	X	X	X	X	X	0	X	X
	1-电平触发								
GPIOIBE	0-单边沿触发	X	X	X	X	X	0	X	X
	1-双边沿触发								

表 10-4. GPIO 中断配置示例 (续)

寄存器	期望的中断事件触发	管脚 2 位的值 <sup>a</sup>							
		7	6	5	4	3	2	1	0
GPIOIEV	0 = 低电平, 或下降沿 1 = 高电平, 或上升沿	X	X	X	X	X	1	X	X
GPIOIM	0-屏蔽 1-没有屏蔽	0	0	0	0	0	1	0	0

a. X = 忽略 (无关位)

## 10.4 寄存器映射

表 10-6 在 607 页 列出了 GPIO 寄存器。每一个 GPIO 端口都可通过两种总线槽访问。比较老的一种称为先进外设总线 (APB), 向后兼容以前的 Stellaris 产品。另外一种先进高端总线 (AHB), 它和 APB 总线一样拥有相同的寄存器映射, 但是提供了更好的连续访问性能。

**重要:** 本章的 GPIO 寄存器在每个 GPIO 块中都是相同的, 但是根据块的不同, 8 个位可能并不是全部与 GPIO 端口相连。向未连接的位写入数据没有任何效果, 而读取未连接的位的数据没有任何意义。

所列的偏移量都是相对于 GPIO 端口基址的 16 进制寄存器地址增量:

- GPIO 端口 A (APB) : 0x4000.4000
- GPIO 端口 A (AHB) : 0x4005.8000
- GPIO 端口 B (APB) : 0x4000.5000
- GPIO 端口 B (AHB) : 0x4005.9000
- GPIO 端口 C (APB) : 0x4000.6000
- GPIO 端口 C (AHB) : 0x4005.A000
- GPIO 端口 D (APB) : 0x4000.7000
- GPIO 端口 D (AHB) : 0x4005.B000
- GPIO 端口 E (APB) : 0x4002.4000
- GPIO 端口 E (AHB) : 0x4005.C000
- GPIO 端口 F (APB) : 0x4002.5000
- GPIO 端口 F (AHB) : 0x4005.D000
- GPIO 端口 G (APB) : 0x4002.6000
- GPIO 端口 G (AHB) : 0x4005.E000
- GPIO 端口 H (APB) : 0x4002.7000
- GPIO 端口 H (AHB) : 0x4005.F000
- GPIO 端口 J (APB) : 0x4003.D000
- GPIO 端口 J (AHB) : 0x4006.0000
- GPIO 端口 K (AHB) : 0x4006.1000
- GPIO 端口 L (AHB) : 0x4006.2000
- GPIO 端口 M (AHB) : 0x4006.3000
- GPIO 端口 N (AHB) : 0x4006.4000
- GPIO 端口 P (AHB) : 0x4006.5000

注意配置这些寄存器之前必须先启用 GPIO 模块的时钟 (请参考 302 页)。启用 GPIO 模块时钟后, 必须等待三个系统时钟才能访问 GPIO 模块的寄存器。

**重要:** 所有的 GPIO 管脚在复位时都被配置为 GPIO 功能, 而且是三态的, 即 (GPIOAFSEL=0、GPIODEN=0、GPIOPDR=0、GPIOPUR=0、GPIOCTL=0), 但是下表中列出的这些管脚除外。上电复位 ( $\overline{POR}$ ) 或是外部复位  $\overline{RST}$  会让这些管脚返回默认值。

表 10-5. 具有非 0 复位值的 GPIO 管脚

GPIO 管脚	默认状态	GPIOAFSEL	GPIODEN	GPIOPDR	GPIOPUR	GPIOPCTL
PA[1:0]	UART0	0	0	0	0	0x1
PA[5:2]	SSI0	0	0	0	0	0x2
PB[3:2]	I <sup>2</sup> C0	0	0	0	0	0x3
PC[3:0]	JTAG/SWD	1	1	0	1	0x1

GPIO 提交控制寄存器提供了保护层，以防止对重要硬件信号的意外编程，其中包括 JTAG/SWD 信号和 NMI 信号。就算是不配置成 JTAG/SWD 或 NMI 信号，而是把他们配置成复用功能，这些管脚也必须遵循提交控制过程，请参考“提交控制”在 604 页。

除了 NMI 管脚和四个 JTAG/SWD 管脚 ( PD7、PF0 和 PC[3:0] ) 之外，所有 GPIO 管脚的 GPIOCR 寄存器的默认状态是 RO。GPIOCR 寄存器当前仅保护这六个 GPIO 管脚。正因为这样，GPIO 端口 D7、GPIO 端口 F0 以及 GPIO 端口 C [3:0] 的寄存器类型是 R/W。

除了 NMI 和四个 JTAG/SWD 管脚 ( PD7、PF0 和 PC[3:0] ) 之外，所有 GPIO 管脚 GPIOCR 寄存器复位值为 0x0000.00FF。为了确保 JTAG 端口不会被意外地编程为 GPIO 管脚，PC[3:0] 管脚默认是锁定的，以防止犯错。类似的，为了确保 NMI 管脚不会被意外地编程为 GPIO 管脚，PD7 和 PF0 管脚默认是锁定的，以防止犯错。因此，GPIO 端口 C 的 GPIOCR 寄存器默认复位值是 0x0000.00F0，GPIO 端口 D 为 0x0000.007F，GPIO 端口 F 为 0x0000.00FE。

表 10-6. GPIO 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	GPIODATA	R/W	0x0000.0000	GPIO 数据寄存器	609
0x400	GPIODIR	R/W	0x0000.0000	GPIO 方向寄存器	610
0x404	GPIOIS	R/W	0x0000.0000	GPIO 中断检测寄存器	611
0x408	GPIOIBE	R/W	0x0000.0000	GPIO 中断双边沿	612
0x40C	GPIOIEV	R/W	0x0000.0000	GPIO 中断事件寄存器	613
0x410	GPIOIM	R/W	0x0000.0000	GPIO 中断屏蔽寄存器	614
0x414	GPIORIS	RO	0x0000.0000	GPIO 原始中断状态寄存器	615
0x418	GPIOMIS	RO	0x0000.0000	GPIO 屏蔽中断状态寄存器	616
0x41C	GPIOICR	W1C	0x0000.0000	GPIO 中断清除寄存器	617
0x420	GPIOAFSEL	R/W	-	GPIO 备用功能选择寄存器	618
0x500	GPIODR2R	R/W	0x0000.00FF	GPIO 2-mA 驱动选择寄存器	620
0x504	GPIODR4R	R/W	0x0000.0000	GPIO 4-mA 驱动选择寄存器	621
0x508	GPIODR8R	R/W	0x0000.0000	GPIO 8-mA 驱动选择寄存器	622
0x50C	GPIOODR	R/W	0x0000.0000	GPIO 开漏选择寄存器	623
0x510	GPIOPUR	R/W	-	GPIO 上拉电阻选择寄存器	624
0x514	GPIOPDR	R/W	0x0000.0000	GPIO 下拉电阻选择寄存器	626
0x518	GPIOSLR	R/W	0x0000.0000	GPIO 斜率控制选择寄存器	628
0x51C	GPIODEN	R/W	-	GPIO 数字使能寄存器	629

表 10-6. GPIO 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x520	GPIOLOCK	R/W	0x0000.0001	GPIO 锁定寄存器	631
0x524	GPIOCR	-	-	GPIO 确认寄存器	632
0x528	GPIOAMSEL	R/W	0x0000.0000	GPIO 模拟选择寄存器	634
0x52C	GPIOCTL	R/W	-	GPIO 端口控制寄存器	636
0x530	GPIOADCCTL	R/W	0x0000.0000	GPIO ADC 控制寄存器	638
0x534	GIODMACTL	R/W	0x0000.0000	GPIO DMA 控制寄存器	639
0x538	GIOSI	R/W	0x0000.0000	GPIO 中断选择寄存器	640
0xFD0	GPIOPeriphID4	RO	0x0000.0000	GPIO 外设标识寄存器 4	641
0xFD4	GPIOPeriphID5	RO	0x0000.0000	GPIO 外设标识寄存器 5	642
0xFD8	GPIOPeriphID6	RO	0x0000.0000	GPIO 外设标识寄存器 6	643
0xFDC	GPIOPeriphID7	RO	0x0000.0000	GPIO 外设标识寄存器 7	644
0xFE0	GPIOPeriphID0	RO	0x0000.0061	GPIO 外设标识寄存器 0	645
0xFE4	GPIOPeriphID1	RO	0x0000.0000	GPIO 外设标识寄存器 1	646
0xFE8	GPIOPeriphID2	RO	0x0000.0018	GPIO 外设标识寄存器 2	647
0xFEC	GPIOPeriphID3	RO	0x0000.0001	GPIO 外设标识寄存器 3	648
0xFF0	GPIOCellID0	RO	0x0000.000D	GPIO PrimeCell 标识寄存器 0	649
0xFF4	GPIOCellID1	RO	0x0000.00F0	GPIO PrimeCell 标识寄存器 1	650
0xFF8	GPIOCellID2	RO	0x0000.0005	GPIO PrimeCell 标识寄存器 2	651
0xFFC	GPIOCellID3	RO	0x0000.00B1	GPIO PrimeCell 标识寄存器 3	652

## 10.5 寄存器描述

本节剩余部分将按地址偏移量的数字顺序逐个列出 GPIO 寄存器，并对各个寄存器进行描述。



## 寄存器 1: GPIO 数据寄存器 ( GPIODATA ) , 偏移量 0x000

GPIODATA 寄存器是数据寄存器。在软件控制模式中，如果通过 GPIO 方向 (GPIODIR) 寄存器 ( 请参考 610页 ) 将各个管脚配置成输出，那么写入 GPIODATA 寄存器的值将被发送到 GPIO 端口管脚。

为了对 GPIODATA 寄存器执行写入操作，由地址总线位 [9:2] 产生的相关屏蔽位必须被置位。否则，位的值不会被写入操作改变。

同样，从该寄存器读取的值由从访问数据寄存器的地址处获取的屏蔽位[9:2]的情况来决定。如果地址屏蔽位为 1，那么读取 GPIODATA 中相应位的值；如果地址屏蔽位为 0，那么不管 GPIODATA 中相应位的值是什么，读取它们得到的都是 0。

如果各自的管脚被配置成输出，那么读取 GPIODATA 将返回最后写入的位值；或者当这些管脚被配置成输入时，将返回相应的输入管脚上的值。复位时所有的位都是清零的。

### GPIO 数据寄存器 (GPIODATA)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x000  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DATA							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DATA	R/W	0x00	GPIO 数据 该寄存器被虚拟地映射到地址空间的 256 个单元中。为便于通过单独的驱动器读写这些寄存器，从这些寄存器读取的值和写入这些寄存器的值可以通过八条地址线 [9:2] 屏蔽。读取该寄存器将返回其当前状态。写入该寄存器仅影响那些没有被 ADDR[9:2] 屏蔽的位和被配置成输出的位。关于读写操作的实例，请参考“数据寄存器的操作”在 602页。

**寄存器 2: GPIO 方向寄存器 ( GPIODIR ) , 偏移量 0x400**

GPIODIR 寄存器是数据方向寄存器。GPIODIR 寄存器中的某位置位会将相应的管脚配置成输出，而清零的话就是配置为输入。复位时，所有的位都被清零，这意味着所有 GPIO 端口默认为输入状态。

**GPIO 方向寄存器 (GPIODIR)**

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x400  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DIR							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DIR	R/W	0x00	GPIO 数据方向

**值 描述**

0 相应的管脚为输入。

1 相应的管脚为输出。

### 寄存器 3: GPIO 中断检测寄存器 ( GPIOIS ) , 偏移量 0x404

GPIOIS 寄存器是中断感知寄存器。GPIOIS 寄存器中某位置位时, 相应的管脚被配置为电平触发, 该位清零时配置为边沿触发。复位时所有的位都是清零的。

#### GPIO 中断检测寄存器 (GPIOIS)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x404  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								IS							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	IS	R/W	0x00	GPIO中断检测类型

#### 值 描述

0 监测相应管脚的边沿 ( 边沿触发 )。

1 监测相应管脚的电平 ( 电平触发 )。

## 寄存器 4: GPIO 中断双边沿 (GPIOIBE), 偏移量 0x408

GPIOIBE 寄存器可以允许双边沿触发中断。当 GPIO 中断检测 (GPIOIS) 寄存器 (请参考 611页) 相应的位置位时, 表示边沿触发。此时, 如果 GPIOIBE 寄存器中相应的位置位, 则表示双边沿触发, 即检测上升沿和下降沿, 不必考虑 GPIO 中断事件 (GPIOIEV) 寄存器 (请参考 613页) 的设置如何。如果清零, 则表示该管脚由 GPIOIEV 寄存器控制。复位时所有的位都是清零的。

## GPIO 中断双边沿 (GPIOIBE)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x408  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								IBE							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

7:0	IBE	R/W	0x00	GPIO 双边沿中断
-----	-----	-----	------	------------

## 值 描述

0 中断发生由 GPIO 中断事件 (GPIOIEV) 寄存器 (请参考 613页) 控制。

1 相应管脚上双边沿触发中断。

## 寄存器 5: GPIO 中断事件寄存器 ( GPIOIEV ) , 偏移量 0x40C

GPIOIEV 是中断事件寄存器。当 GPIOIEV 寄存器中某位置位时, 相应的管脚由上升沿或是高电平触发中断, 具体由 GPIO 中断检测 (GPIOIS) 寄存器 ( 请参考 611页 ) 中的位控制。清零一个位则表示相应的管脚由下降沿或是低电平触发, 具体还是由 GPIOIS 中的位控制。复位时所有的位都是清零的。

### GPIO 中断事件寄存器 (GPIOIEV)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x40C  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								IEV							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	IEV	R/W	0x00	GPIO中断事件

#### 值 描述

- 0 相应管脚上的下降沿或低电平触发中断。
- 1 相应管脚上的下降沿或高电平触发中断。

## 寄存器 6: GPIO 中断屏蔽寄存器 (GPIOIM), 偏移量 0x410

GPIOIM 是中断屏蔽寄存器。GPIOIM 寄存器中某位置位时, 相应的管脚产生的中断允许被送到联合中断信号上的中断控制器。清零某位则相应的管脚产生的中断不会被送到中断控制器。复位时所有的位都是清零的。

## GPIO 中断屏蔽寄存器 (GPIOIM)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x410  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								IME							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7:0	IME	R/W	0x00	GPIO 中断屏蔽启用

## 值 描述

0 相应管脚的中断被屏蔽。

1 相应管脚的中断会被发送到中断控制器。

## 寄存器 7: GPIO 原始中断状态寄存器 ( GPIORIS ) , 偏移量 0x414

GPIORIS 寄存器是原始中断状态寄存器。当一个管脚上发生中断时GPIORIS寄存器被置位。当GPIO中断屏蔽(GPIOIM)寄存器(请参考 614页)中的某位置位时,相应的中断被送到中断控制器。读取某位为零则表示相应的输入管脚未发生中断。向GPIO中断清零(GPIOICR)寄存器中的位写入1会清零该寄存器相应的位。

### GPIO 原始中断状态寄存器 (GPIORIS)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x414  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								RIS							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件,保留位的值在读-修改-写操作过程中应该保持不变。
7:0	RIS	RO	0x00	GPIO 原始中断状态

#### 值 描述

0 相应的管脚上未发生中断条件。

1 相应的管脚上已经有中断条件发生。

向GPIOICR寄存器中的位写入1来清零相应的位。

**寄存器 8: GPIO 屏蔽中断状态寄存器 ( GPIOMIS ) ， 偏移量 0x418**

GPIOMIS 寄存器是屏蔽中断状态寄存器。如果该寄存器中的某个位置位，则说明相应的中断已经发送到中断控制器。如果某位清零，表示没有产生中断，或者中断被屏蔽。

请注意，如果 Port B GPIOADCCTL 寄存器被清零，PB4 也可以用作 ADC 的外部触发信号。通过这种传统模式，用户可以向 Stellaris 以前的设备写入代码，以在该微控制器中工作。

GPIOMIS 是屏蔽后中断的状态。

**GPIO 屏蔽中断状态寄存器 (GPIOMIS)**

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x418  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								MIS							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:0	MIS	RO	0x00	GPIO 屏蔽中断状态

**值 描述**

0 相应的管脚上未产生中断或中断已经被屏蔽。

1 相应管脚上的中断条件已经触发中断，并发送到了中断控制器。

向 GPIOICR 寄存器中的位写入 1 来清零相应的位。



## 寄存器 9: GPIO 中断清除寄存器 ( GPIOICR ) , 偏移量 0x41C

GPIOICR 是中断清除寄存器。将该寄存器中的位写入 1 会将 GPIORIS 和 GPIOMIS 寄存器中相应的中断位清零。写“0”没有任何效果。

### GPIO 中断清除寄存器 (GPIOICR)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x41C  
 类型 W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								IC							
类型	RO	RO	RO	RO	RO	RO	RO	RO	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件,保留位的值在读-修改-写操作过程中应该保持不变。
7:0	IC	W1C	0x00	GPIO 中断清除
	值	描述		
	0	相应的中断未受影响。		
	1	相应的中断被清除。		

**寄存器 10: GPIO 备用功能选择寄存器 (GPIOAFSEL)，偏移量 0x420**

GPIOAFSEL 寄存器是复用功能选择寄存器。如果某位清零，则表示相应的管脚用作 GPIO 功能，并受 GPIO 寄存器控制。某位置位则表示相应的 GPIO 线受相关外设控制。每个 GPIO 上都有几个不同的复用外设功能。可以通过 GPIO 端口控制 (GPIOCTL) 寄存器来选择其中的一个可能的功能。表 23-5 在 1280 页详细列出了每个 GPIO 管脚上的复用功能。复位时，下表中没有列出的端口对应的该寄存器的值为 0x0000.0000。

**重要:** 所有的 GPIO 管脚在复位时都被配置为 GPIO 功能，而且是三态的，即 (GPIOAFSEL=0、GPIODEN=0、GPIOPDR=0、GPIOPUR=0、GPIOCTL=0)，但是下表中列出的这些管脚除外。上电复位 (POR) 或是外部复位 RST 会让这些管脚返回默认值。

表 10-7. 具有非 0 复位值的 GPIO 管脚

GPIO 管脚	默认状态	GPIOAFSEL	GPIODEN	GPIOPDR	GPIOPUR	GPIOCTL
PA[1:0]	UART0	0	0	0	0	0x1
PA[5:2]	SSI0	0	0	0	0	0x2
PB[3:2]	I <sup>2</sup> C0	0	0	0	0	0x3
PC[3:0]	JTAG/SWD	1	1	0	1	0x1

GPIO 提交控制寄存器提供了保护层，以防止对重要硬件信号的意外编程，其中包括 JTAG/SWD 信号和 NMI 信号。就算是不配置成 JTAG/SWD 或 NMI 信号，而是把他们配置成复用功能，这些管脚也必须遵循提交控制过程，请参考“提交控制”在 604 页。

**小心** – 用户可以建立一个软件序列来阻止调试器连接到 Stellaris 微控制器的特性。如果将程序代码加载到 Flash 中会立即将 JTAG 管脚变成其 GPIO 功能，那么在 JTAG 管脚功能切换之前，调试器将没有足够的时间去连接和终止控制器。结果调试器可能被锁定在该部分外。为了避免这个问题，可以使用一个基于外部或软件触发器的软件程序来恢复 JTAG 功能。

GPIO 确认控制寄存器提供的保护功能可防止对重要硬件外设的意外编程。为以下 GPIO 管脚提供保护：能用作 JTAG/SWD 管脚 (PC[3:0]) 和 NMI 管脚 (PD7 和 PF0)。向 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (请参阅 618 页)、GPIO 上拉电阻选择 (GPIOPUR) 寄存器 (请参阅 624 页)、GPIO 下拉电阻选择 (GPIOPDR) 寄存器 (请参阅 626 页) 以及 GPIO 数字使能 (GPIODEN) 寄存器 (请参阅 629 页) 中受保护的位写入数据将不会确认保存，除非 GPIO 锁定 (GPIOLOCK) 寄存器 (请参阅 631 页) 没有被锁定，同时 GPIO 确认 (GPIOCR) 寄存器 (请参阅 632 页) 中相应的位被置位。

当使用 I<sup>2</sup>C 模块时，除了要设置 GPIOAFSEL 中的 I<sup>2</sup>C 时钟和数据管脚之外，还必须要用 GPIO 开漏选择 (GPIOODR) 寄存器 (请参考“初始化及配置”在 605 页中的例子) 将管脚设置成开漏。

## GPIO 备用功能选择寄存器 (GPIOAFSEL)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0x420

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								AFSEL							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	AFSEL	R/W	-	GPIO 复用功能选择

## 值 描述

0 相应管脚的功能是 GPIO，受 GPIO 寄存器控制。

1 相应管脚的功能是外设信号，由复用硬件功能控制。

对于未在表 10-1 在 598页 中列出的 GPIO 端口，该寄存器的复位值为 0x0000.0000。

## 寄存器 11: GPIO 2-mA 驱动选择寄存器 ( GPIODR2R ) , 偏移量 0x500

GPIODR2R 寄存器是 2-mA 驱动控制寄存器。每个 GPIO 信号都可以单独配置，而不会影响其他的管脚。当 GPIO 信号的 DRV2 置位时，GPIODR4R 寄存器中的 DRV4 位和 GPIODR8R 寄存器中的 DRV8 会被硬件自动清零。所有 GPIO 管脚默认都是 2-mA 驱动。

## GPIO 2-mA 驱动选择寄存器 (GPIODR2R)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x500  
 类型 R/W, 复位 0x0000.00FF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DRV2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DRV2	R/W	0xFF	输出管脚 2-mA 驱动启用

## 值 描述

- 1 相应的 GPIO 管脚为 2-mA 驱动。
- 0 相应的 GPIO 管脚驱动由 GPIODR4R 或 GPIODR8R 寄存器控制。

将 GPIODR4 寄存器或 GPIODR8 寄存器中的某位置位，都会将相应的 2-mA 启用位清零。如果通过 APB 存储器槽访问 GPIO，那么这种改变在写入的第二个时钟周期生效。如果使用 AHB 访问，那么在下一个时钟周期生效。

## 寄存器 12: GPIO 4-mA 驱动选择寄存器 ( GPIODR4R ) , 偏移量 0x504

GPIODR4R 寄存器是 4-mA 驱动控制寄存器。每个 GPIO 信号都可以单独配置，而不会影响其他的管脚。当 GPIO 信号的 DRV4 置位时，GPIODR2R 寄存器中的 DRV2 位和 GPIODR8R 寄存器中的 DRV8 位会被硬件自动清零。

### GPIO 4-mA 驱动选择寄存器 (GPIODR4R)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0x504

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DRV4							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DRV4	R/W	0x00	输出管脚 4-mA 电流驱动启用

#### 值 描述

- 1 相应的 GPIO 管脚为 4-mA 驱动。
- 0 相应的管脚驱动由 GPIODR2R 或 GPIODR8R 寄存器控制。

将 GPIODR2 寄存器或 GPIODR8 寄存器中的某位置位，都会将相应的 4-mA 启用位清零。如果通过 APB 存储器槽访问 GPIO，那么这种改变在写入的第二个时钟周期生效。如果使用 AHB 访问，那么在下一个时钟周期生效。

**寄存器 13: GPIO 8-mA 驱动选择寄存器 ( GPIODR8R ) ， 偏移量 0x508**

GPIODR8R 是 8-mA 驱动控制寄存器。每个 GPIO 信号都可以单独配置，而不会影响其他的管脚。当 GPIO 信号的 DRV8 置位时，GPIODR2R 寄存器中的 DRV2 位和 GPIODR4R 寄存器中的里 DRV4 会被硬件自动清零。8-mA 驱动还可被用在大电流驱动的应用上。

**注意：** 8-mA 驱动和大电流驱动在配置上没有区别。额外的电流来自  $V_{OH}/V_{OL}$  电平转换。更多信息请参考 “Recommended Operating Conditions” 在 1292 页。

**GPIO 8-mA 驱动选择寄存器 (GPIODR8R)**

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x508  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DRV8							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DRV8	R/W	0x00	输出管脚 8-mA 电流驱动启用

**值 描述**

- 1 相应的 GPIO 管脚为 8-mA 驱动。
- 0 相应的 GPIO 管脚驱动由 GPIODR2R 或 GPIODR4R 寄存器控制。

将 GPIODR2 寄存器或 GPIODR4 寄存器中的某位置位，都会将相应的 8-mA 启用位清零。如果通过 APB 存储器槽访问 GPIO，那么这种改变在写入的第二个时钟周期生效。如果使用 AHB 访问，那么在下一个时钟周期生效。

## 寄存器 14: GPIO 开漏选择寄存器 ( GPIOODR ) , 偏移量 0x50C

GPIOODR 寄存器是开漏选择寄存器。该寄存器中的某位置位可以将相应的 GPIO 管脚配置为开漏。当管脚的开漏功能启用的时候, GPIO 数字使能(GPIODEN) 寄存器( 请参考 629页 ) 中相应的位也必须置位。为了达到所需的上升和下降时间, 可以将驱动强度和斜率控制寄存器 ( GPIODR2R、GPIODR4R、GPIODR8R 和 GPIOSLR ) 中相应的位置位。如果 GPIODIR 寄存器的某位被清零, 那么相应的 GPIO 是输入。如果 GPIO 配置成输入, 而同时又选择了开漏, 那么该 GPIO 还是输入, 开漏选择不会生效, GPIO 变成输出以后才开漏才有效。

当使用 I<sup>2</sup>C 模块时, 除了将管脚配置成开漏功能, GPIO 备用功能选择 (GPIOAFSEL) 寄存器中的 I<sup>2</sup>C 数据管脚位必须置位 ( 请参考“初始化及配置”在 605页 中的示例 )。

### GPIO 开漏选择寄存器 (GPIOODR)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x50C  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								ODE							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	ODE	R/W	0x00	GPIO 开漏选择启用  值 描述 1 相应管脚配置成开漏。 0 相应管脚没有配置成开漏。

**寄存器 15: GPIO 上拉电阻选择寄存器 ( GPIOPUR ) , 偏移量 0x510**

GPIOPUR 寄存器是上拉电阻控制寄存器。当某位置位时, 相应 GPIO 信号上弱上拉电阻启用。将 GPIOPUR 中的位置位会自动将 GPIO 下拉电阻选择 (GPIOPDR) 寄存器 ( 请参考 626 页 ) 中相应的位清零。GPIOCR 寄存器保护该寄存器的写入操作。GPIOCR 寄存器某位被清零时可以阻止该寄存器中相应的位的写入操作。

**重要:** 所有的 GPIO 管脚在复位时都被配置为 GPIO 功能, 而且是三态的, 即 (GPIOAFSEL=0、GPIODEN=0、GPIOPDR=0、GPIOPUR=0、GPIOCTL=0), 但是下表中列出的这些管脚除外。上电复位 (POR) 或是外部复位 RST 会让这些管脚返回默认值。

**表 10-8. 具有非 0 复位值的 GPIO 管脚**

GPIO 管脚	默认状态	GPIOAFSEL	GPIODEN	GPIOPDR	GPIOPUR	GPIOCTL
PA[1:0]	UART0	0	0	0	0	0x1
PA[5:2]	SSI0	0	0	0	0	0x2
PB[3:2]	I <sup>2</sup> C0	0	0	0	0	0x3
PC[3:0]	JTAG/SWD	1	1	0	1	0x1

GPIO 提交控制寄存器提供了保护层, 以防止对重要硬件信号的意外编程, 其中包括 JTAG/SWD 信号和 NMI 信号。就算是不配置成 JTAG/SWD 或 NMI 信号, 而是把他们配置成复用功能, 这些管脚也必须遵循提交控制过程, 请参考“提交控制”在 604 页。

**注意:** GPIO 确认控制寄存器提供的保护功能可防止对重要硬件外设的意外编程。为以下 GPIO 管脚提供保护: 能用作 JTAG/SWD 管脚 (PC[3:0]) 和 NMI 管脚 (PD7 和 PF0)。向 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 ( 请参阅 618 页 )、GPIO 上拉电阻选择 (GPIOPUR) 寄存器 ( 请参阅 624 页 )、GPIO 下拉电阻选择 (GPIOPDR) 寄存器 ( 请参阅 626 页 ) 以及 GPIO 数字使能 (GPIODEN) 寄存器 ( 请参阅 629 页 ) 中受保护的位写入数据将不会确认保存, 除非 GPIO 锁定 (GPIOLOCK) 寄存器 ( 请参阅 631 页 ) 没有被锁定, 同时 GPIO 确认 (GPIOCR) 寄存器 ( 请参阅 632 页 ) 中相应的位被置位。



## GPIO 上拉电阻选择寄存器 (GPIOPUR)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0x510

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PUE							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PUE	R/W	-	GPIO 弱上拉电阻启用

## 值 描述

- 1 相应管脚上是一个弱上拉电阻。
- 0 相应管脚不受影响。

将 GPIOPDR 寄存器中的位置位会将 GPIOPUR 寄存器中相应的位置清零。如果通过 APB 存储器槽访问 GPIO，那么这种改变在写入的第二个时钟周期生效。如果使用 AHB 访问，那么在下一个时钟周期生效。

对于未在表 10-1 在 598 页中列出的 GPIO 端口，该寄存器的复位值为 0x0000.0000。

**寄存器 16: GPIO 下拉电阻选择寄存器 ( GPIOPDR ) , 偏移量 0x514**

GPIOPDR 缓冲区是下拉控制寄存器。置位时相应的管脚下拉电阻启用。置位 GPIOPDR 中的位会使 GPIO 上拉电阻选择 (GPIOPUR) 寄存器 ( 见 624 页 ) 中相应的位自动清零

**重要:** 所有的 GPIO 管脚在复位时都被配置为 GPIO 功能, 而且是三态的, 即 (GPIOAFSEL=0、GPIODEN=0、GPIOPDR=0、GPIOPUR=0、GPIOCTL=0), 但是下表中列出的这些管脚除外。上电复位 (POR) 或是外部复位 RST 会让这些管脚返回默认值。

**表 10-9. 具有非 0 复位值的 GPIO 管脚**

GPIO 管脚	默认状态	GPIOAFSEL	GPIODEN	GPIOPDR	GPIOPUR	GPIOCTL
PA[1:0]	UART0	0	0	0	0	0x1
PA[5:2]	SSI0	0	0	0	0	0x2
PB[3:2]	I <sup>2</sup> C0	0	0	0	0	0x3
PC[3:0]	JTAG/SWD	1	1	0	1	0x1

GPIO 提交控制寄存器提供了保护层, 以防止对重要硬件信号的意外编程, 其中包括 JTAG/SWD 信号和 NMI 信号。就算是不配置成 JTAG/SWD 或 NMI 信号, 而是把他们配置成复用功能, 这些管脚也必须遵循提交控制过程, 请参考“提交控制”在 604 页。

**注意:** GPIO 确认控制寄存器提供的保护功能可防止对重要硬件外设的意外编程。为以下 GPIO 管脚提供保护: 能用作 JTAG/SWD 管脚 (PC[3:0]) 和 NMI 管脚 (PD7 和 PF0)。向 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 ( 请参阅 618 页 )、GPIO 上拉电阻选择 (GPIOPUR) 寄存器 ( 请参阅 624 页 )、GPIO 下拉电阻选择 (GPIOPDR) 寄存器 ( 请参阅 626 页 ) 以及 GPIO 数字使能 (GPIODEN) 寄存器 ( 请参阅 629 页 ) 中受保护的位写入数据将不会确认保存, 除非 GPIO 锁定 (GPIOLOCK) 寄存器 ( 请参阅 631 页 ) 没有被锁定, 同时 GPIO 确认 (GPIOCR) 寄存器 ( 请参阅 632 页 ) 中相应的位被置位。

## GPIO 下拉电阻选择寄存器 (GPIOPDR)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0x514

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PDE							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PDE	R/W	0x00	GPIO 弱下拉电阻启用

## 值 描述

- 1 相依管脚上有一个弱下拉电阻。
- 0 相应管脚不受影响。

将 GPIOPUR 寄存器中的位置位会将 GPIOPDR 寄存器中相应的位清零。如果通过 APB 存储器槽访问 GPIO，那么这种改变在写入的第二个时钟周期生效。如果使用 AHB 访问，那么在下一个时钟周期生效。

## 寄存器 17: GPIO 斜率控制选择寄存器 ( GPIOSLR ) , 偏移量 0x518

GPIOSLR 是斜率控制选择寄存器。斜率控制只有在通过 GPIO 8-mA 驱动选择 (GPIODR8R) 寄存器 ( 请参考 622页 ) 选用 8-mA 驱动强度时才有效。

## GPIO 斜率控制选择寄存器 (GPIOSLR)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x518  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								SRL							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	SRL	R/W	0x00	斜率限制启用 ( 8-mA 驱动 )

## 值 描述

- 1 启用相应管脚的斜率控制。
- 0 禁用相应管脚的斜率控制。

**寄存器 18: GPIO 数字使能寄存器 ( GPIODEN ) , 偏移量 0x51C**

**注意:** 配置为数字输入的管脚均为施密特触发。

GPIODEN 寄存器是数字启用寄存器。默认情况下,除了下面列出的管脚外,所有的管脚复位时都被配置为非驱动(三态)的。它们的数字功能是禁止的。它们不驱动管脚上的逻辑值,而且也不允许管脚上的电压进入 GPIO 接收器。为了使用管脚的数字功能(作为数字输入/输出,或者是 GPIO 功能,或者是复用功能),GPIODEN 寄存器中相应的位必须置位。

**重要:** 所有的 GPIO 管脚在复位时都被配置为 GPIO 功能,而且是三态的,即(GPIOAFSEL=0、GPIODEN=0、GPIOPDR=0、GPIOPUR=0、GPIOCTL=0),但是下表中列出的这些管脚除外。上电复位(POR)或是外部复位 RST 会让这些管脚返回默认值。

**表 10-10. 具有非 0 复位值的 GPIO 管脚**

GPIO 管脚	默认状态	GPIOAFSEL	GPIODEN	GPIOPDR	GPIOPUR	GPIOCTL
PA[1:0]	UART0	0	0	0	0	0x1
PA[5:2]	SSI0	0	0	0	0	0x2
PB[3:2]	I <sup>2</sup> C0	0	0	0	0	0x3
PC[3:0]	JTAG/SWD	1	1	0	1	0x1

GPIO 提交控制寄存器提供了保护层,以防止对重要硬件信号的意外编程,其中包括 JTAG/SWD 信号和 NMI 信号。就算是不配置成 JTAG/SWD 或 NMI 信号,而是把他们配置成复用功能,这些管脚也必须遵循提交控制过程,请参考“提交控制”在 604页。

**注意:** GPIO 确认控制寄存器提供的保护功能可防止对重要硬件外设的意外编程。为以下 GPIO 管脚提供保护:能用作 JTAG/SWD 管脚(PC[3:0])和 NMI 管脚(PD7 和 PF0)。向 GPIO 备用功能选择(GPIOAFSEL)寄存器(请参阅 618页)、GPIO 上拉电阻选择(GPIOPUR)寄存器(请参阅 624页)、GPIO 下拉电阻选择(GPIOPDR)寄存器(请参阅 626页)以及 GPIO 数字使能(GPIODEN)寄存器(请参阅 629页)中受保护的位写入数据将不会确认保存,除非 GPIO 锁定(GPIOLOCK)寄存器(请参阅 631页)没有被锁定,同时 GPIO 确认(GPIOCR)寄存器(请参阅 632页)中相应的位被置位。

GPIO 数字使能寄存器 (GPIODEN)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0x51C  
 类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DEN							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DEN	R/W	-	数字启用

值 描述

- 0 相应管脚的数字功能被禁用。
- 1 启用管脚的数字功能。

对于未在表 10-1 在 598页 中列出的 GPIO 端口，该寄存器的复位值为 0x0000.0000。

## 寄存器 19: GPIO 锁定寄存器 ( GPIOLOCK ) , 偏移量 0x520

GPIOLOCK 寄存器可以启用对 GPIOCR 寄存器 ( 请参考 632页 ) 的写入访问。将 GPIOLOCK 寄存器中写入 0x4C4F.434B 能够解锁 GPIOCR 寄存器。向 GPIOLOCK 寄存器中写入任何数值都可以重新恢复锁定状态。读取 GPIOLOCK 寄存器返回锁定状态, 而不是先前写入的 32 位数值。因此, 当写入访问被禁用或锁定时, 读取 GPIOLOCK 寄存器返回 0x00000001。当写入访问被启用或解锁时, 读取 GPIOLOCK 寄存器返回 0x00000000。

### GPIO 锁定寄存器 (GPIOLOCK)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0x520

类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LOCK															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:0	LOCK	R/W	0x0000.0001	<p>GPIO 锁定</p> <p>写入 0x4C4F.434B 可以解锁 GPIO 确认 (GPIOCR) 寄存器, 允许写入访问。向 GPIOCR 寄存器写入任何其他的数据重新锁定, 阻止任何寄存器更新。</p> <p>读取该寄存器返回以下各值:</p> <p>值 描述</p> <p>0x1 GPIOCR 寄存器锁定, 不能修改。</p> <p>0x0 GPIOCR 解锁, 可以修改。</p>

**寄存器 20: GPIO 确认寄存器 ( GPIOCR ) ， 偏移量 0x524**

GPIOCR 寄存器是提交寄存器。GPIOCR 寄存器的值决定了 GPIOAFSEL、GPIOPUR、GPIOPDR 以及 GPIODEN 寄存器中哪些位在写入操作时被提交。如果在 GPIOCR 寄存器中的一个位被清零，那么写入 GPIOAFSEL、GPIOPUR、GPIOPDR 或 GPIODEN 寄存器中相应位的数据将不被提交并保留其原来的值。如果 GPIOCR 寄存器中的位为 1，那么写入 GPIOAFSEL、GPIOPUR、GPIOPDR 或 GPIODEN 寄存器中相应位的数据将被提交到寄存器并显示新的值。

GPIOCR 寄存器的内容只有在 GPIOLOCK 寄存器解锁时才能被修改。如果 GPIOLOCK 寄存器被锁定，那么写入 GPIOCR 寄存器将被忽略。

**重要:** 该寄存器用于防止意外设置的寄存器，该寄存器控制到 NMI 和 JTAG/SWD 调试硬件的连通性。通过将 GPIOCR 寄存器中相应的位初始化为 0 ( PD7、PF0 以及 PC[3:0] 对应的位 )，NMI 和 JTAG/SWD 调试端口只能通过对 GPIOLOCK，GPIOCR，以及其他相应寄存器的一系列写操作来转换为 GPIO。

因为这种保护当前只在 PD7、PF0 以及 PC[3:0] 的 JTAG/SWD 管脚上执行，所以 GPIOCR 寄存器中所有其它的位不能被写入 0x0。这些位硬连接 ( hardwired ) 为 0x1，确保其它管脚的 GPIOAFSEL，GPIOPUR，GPIOPDR，或 GPIODEN 寄存器位总是可以提交新的数值。

**GPIO 确认寄存器 (GPIOCR)**

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x524  
 类型 -, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CR							
类型	RO	RO	RO	RO	RO	RO	RO	RO	-	-	-	-	-	-	-	-
复位	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。



位/域	名称	类型	复位	描述
7:0	CR	-	-	<p>GPIO 提交</p> <p>值 描述</p> <p>1 相应的 GPIOAFSEL、GPIOPUR、GPIOPDR 或 GPIODEN 位可以写入。</p> <p>0 相应的 GPIOAFSEL、GPIOPUR、GPIOPDR 或 GPIODEN 位不能写入。</p> <p>注意: 除了 NMI 管脚和四个 JTAG/SWD 管脚 ( PD7、PF0 和 PC[3:0] ) 之外, 所有 GPIO 管脚的 GPIOCR 寄存器的默认状态是 RO。GPIOCR 寄存器当前仅保护这六个 GPIO 管脚。正因为这样, GPIO 端口 D7、GPIO 端口 F0 以及 GPIO 端口 C [3:0] 的寄存器类型是 R/W。</p> <p>除了 NMI 和四个 JTAG/SWD 管脚 ( PD7、PF0 和 PC[3:0] ) 之外, 所有 GPIO 管脚 GPIOCR 寄存器复位值为 0x0000.00FF。为了确保 JTAG 端口不会被意外地编程为 GPIO 管脚, PC[3:0] 管脚默认是锁定的, 以防止犯错。类似的, 为了确保 NMI 管脚不会被意外地编程为 GPIO 管脚, PD7 和 PF0 管脚默认是锁定的, 以防止犯错。因此, GPIO 端口 C 的 GPIOCR 寄存器默认复位值是 0x0000.00F0, GPIO 端口 D 为 0x0000.007F, GPIO 端口 F 为 0x0000.00FE。</p>

## 寄存器 21: GPIO 模拟选择寄存器 ( GPIOAMSEL ) , 偏移量 0x528

**重要:** 该寄存器只能用于那些可以用作 ADC AINx 输入的端口和管脚。

如果一个管脚用作 ADC 输入管脚, 那么 GPIOAMSEL 寄存器中相应的位必须置位, 禁用模拟隔离电路。

GPIOAMSEL 寄存器控制着隔离电路是否与通用 I/O 管脚的模拟一边相连。因为 GPIO 管脚可能会被 5 V 电源驱动或受模拟操作的影响, 所以模拟电路在不用的时候需要从管脚上隔离出来。

该寄存器的每一位控制着相应 GPIO 信号的隔离电路。要了解哪些 GPIO 管脚可以用作 ADC 功能, 请参考表 23-5 在 1280 页。

## GPIO 模拟选择寄存器 (GPIOAMSEL)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x528  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								GPIOAMSEL							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
7:0	GPIOAMSEL	R/W	0x00	GPIO 模拟模式选择  值 描述 1 管脚的 GPIO 模拟功能启用，隔离电路禁用，该管脚能够进行模拟信号操作。 0 管脚的 GPIO 模拟功能禁用，隔离电路启用，该管脚能够进行其他 GPIO 寄存器指定的数字信号处理。  注意： 该寄存器和位只适用于那些通过通用 I/O 口共享模拟功能的管脚。 该寄存器对所有信号的复位状态都是 0。

**寄存器 22: GPIO 端口控制寄存器 ( GPIOCTL ) , 偏移量 0x52C**

GPIOCTL 寄存器与 GPIOAFSEL 寄存器协作，为每个 GPIO 管脚（当使用复用功能模式时）选择具体的外设信号。GPIOAFSEL 寄存器中大部分的位在复位时是清零的，所以大部分 GPIO 管脚默认配置为 GPIO 功能。GPIOAFSEL 寄存器中某位置位时，表示相应的 GPIO 管脚由相关外设控制。GPIOCTL 寄存器可以为每个 GPIO 管脚选择使用哪个外设功能，因此在信号定义时提供了很高的灵活性。该寄存器中位域编码的更多信息，请参考表 23-5 在 1280 页。复位时，下表中没有列出的端口对应的该寄存器的值为 0x0000.0000。

**注意：** 如果同一信号被分配给两个不同的 GPIO 端口管脚，则该信号分配给编号更小的一个端口，而忽略编号更大端口的信号分配。

**重要：** 所有的 GPIO 管脚在复位时都被配置为 GPIO 功能，而且是三态的，即 (GPIOAFSEL=0、GPIODEN=0、GPIOPDR=0、GPIOPUR=0、GPIOCTL=0)，但是下表中列出的这些管脚除外。上电复位 (POR) 或是外部复位 RST 会让这些管脚返回默认值。

**表 10-11. 具有非 0 复位值的 GPIO 管脚**

GPIO 管脚	默认状态	GPIOAFSEL	GPIODEN	GPIOPDR	GPIOPUR	GPIOCTL
PA[1:0]	UART0	0	0	0	0	0x1
PA[5:2]	SSI0	0	0	0	0	0x2
PB[3:2]	I <sup>2</sup> C0	0	0	0	0	0x3
PC[3:0]	JTAG/SWD	1	1	0	1	0x1

GPIO 提交控制寄存器提供了保护层，以防止对重要硬件信号的意外编程，其中包括 JTAG/SWD 信号和 NMI 信号。就算是不配置成 JTAG/SWD 或 NMI 信号，而是把他们配置成复用功能，这些管脚也必须遵循提交控制过程，请参考“提交控制”在 604 页。

## GPIO 端口控制寄存器 (GPIOPCTL)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0x52C  
 类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMC7				PMC6				PMC5				PMC4			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC3				PMC2				PMC1				PMC0			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:28	PMC7	R/W	-	端口复用控制器7 该区域的值控制着端口7的配置
27:24	PMC6	R/W	-	端口复用控制器6 该区域的值控制着端口6的配置
23:20	PMC5	R/W	-	端口复用控制器5 该区域的值控制着端口5的配置
19:16	PMC4	R/W	-	端口复用控制器4 该区域的值控制着端口4的配置
15:12	PMC3	R/W	-	端口复用控制器3 该区域的值控制着端口3的配置
11:8	PMC2	R/W	-	端口复用控制器2 该区域的值控制着端口2的配置
7:4	PMC1	R/W	-	端口复用控制器1 该区域的值控制着端口1的配置
3:0	PMC0	R/W	-	端口复用控制器0 该区域的值控制着端口0的配置

**寄存器 23: GPIO ADC 控制寄存器 ( GPIOADCCTL ) , 偏移量 0x530**

该寄存器用来将 GPIO 管脚配置成 ADC 的触发源。

请注意，如果 Port B GPIOADCCTL 寄存器被清零，PB4 也可以用作 ADC 的外部触发信号。通过这种传统模式，用户可以向 Stellaris 以前的设备写入代码，以在该微控制器中工作。

**GPIO ADC 控制寄存器 (GPIOADCCTL)**

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x530  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								ADCEN							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	ADCEN	R/W	0x00	ADC 触发启用
				值 描述
				1 相应管脚用来触发 ADC。
				0 相应管脚没有用来触发 ADC。

## 寄存器 24: GPIO DMA 控制寄存器 ( GPIODMACTL ) , 偏移量 0x534

该寄存器用来将 GPIO 管脚配置成  $\mu$ DMA 的触发源。

### GPIO DMA 控制寄存器 (GPIODMACTL)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x534

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DMAEN							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DMAEN	R/W	0x00	$\mu$ DMA 触发启用

#### 值 描述

- 1 相应管脚用来触发  $\mu$ DMA。
- 0 相应管脚没有用来触发  $\mu$ DMA。

**寄存器 25: GPIO 中断选择寄存器 ( GPIOSI ) , 偏移量 0x538**

该寄存器用来启用每个管脚的单个中断。

注意: 该寄存器只在端口 P 和端口 Q 可用。

**GPIO 中断选择寄存器 (GPIOSI)**

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0x538  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	SUM	R/W	0	总结中断

**值 描述**

- 1 每个管脚有自己的中断向量。
- 0 所有的端口管脚中断进行或运算之后产生一个总结中断。



## 寄存器 26: GPIO 外设标识寄存器 4 ( GPIOPeriphID4 ) , 偏移量 0xFD0

GPIOPeriphID4、GPIOPeriphID5、GPIOPeriphID6 和 GPIOPeriphID7 这四个寄存器在概念上可以看作是一个 32 位的寄存器，每个寄存器包含了该 32 位寄存器的 8 个位，被软件用来标识外设。

### GPIO 外设标识寄存器 4 (GPIOPeriphID4)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFD0  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID4							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID4	RO	0x00	GPIO 外设 ID 寄存器 [7:0]

## 寄存器 27: GPIO 外设标识寄存器 5 ( GPIOPeriphID5 ) , 偏移量 0xFD4

GPIOPeriphID4、GPIOPeriphID5、GPIOPeriphID6 和 GPIOPeriphID7 这四个寄存器在概念上可以看作是一个 32 位的寄存器，每个寄存器包含了该 32 位寄存器的 8 个位，被软件用来标识外设。

## GPIO 外设标识寄存器 5 (GPIOPeriphID5)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFD4  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID5							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID5	RO	0x00	GPIO 外设 ID 寄存器 [15:8]

## 寄存器 28: GPIO 外设标识寄存器 6 ( GPIOPeriphID6 ) , 偏移量0xFD8

GPIOPeriphID4、GPIOPeriphID5、GPIOPeriphID6 和 GPIOPeriphID7 这四个寄存器在概念上可以看作是一个 32 位的寄存器，每个寄存器包含了该 32 位寄存器的 8 个位，被软件用来标识外设。

### GPIO 外设标识寄存器 6 (GPIOPeriphID6)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFD8  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID6							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID6	RO	0x00	GPIO 外设 ID 寄存器 [23:16]

## 寄存器 29: GPIO 外设标识寄存器 7 ( GPIOPeriphID7 ) , 偏移量 0xFDC

GPIOPeriphID4、GPIOPeriphID5、GPIOPeriphID6 和 GPIOPeriphID7 这四个寄存器在概念上可以看作是一个 32 位的寄存器，每个寄存器包含了该 32 位寄存器的 8 个位，被软件用来标识外设。

## GPIO 外设标识寄存器 7 (GPIOPeriphID7)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFDC  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID7							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID7	RO	0x00	GPIO 外设 ID 寄存器 [31:24]

### 寄存器 30: GPIO 外设标识寄存器 0 ( GPIOPeriphID0 ) , 偏移量 0xFE0

GPIOPeriphID0、GPIOPeriphID1、GPIOPeriphID2 和 GPIOPeriphID3 这四个寄存器基本可以看成是一个 32 位的寄存器，每个寄存器包含了这个 32 位寄存器其中的 8 位，软件用这些寄存器来识别外设。

#### GPIO 外设标识寄存器 0 (GPIOPeriphID0)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFE0  
 类型 RO, 复位 0x0000.0061

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID0	RO	0x61	GPIO 外设 ID 寄存器 [7:0] 软件可通过此寄存器识别外设。

**寄存器 31: GPIO 外设标识寄存器 1 ( GPIOPeriphID1 ) ， 偏移量 0xFE4**

GPIOPeriphID0、GPIOPeriphID1、GPIOPeriphID2 和 GPIOPeriphID3 这四个寄存器基本可以看成是一个 32 位的寄存器，每个寄存器包含了这个 32 位寄存器其中的 8 位，软件用这些寄存器来识别外设。

## GPIO 外设标识寄存器 1 (GPIOPeriphID1)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0xFE4

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID1	RO	0x00	GPIO 外设 ID 寄存器 [15:8] 软件可通过此寄存器识别外设。

## 寄存器 32: GPIO 外设标识寄存器 2 ( GPIOPeriphID2 ) , 偏移量 0xFE8

GPIOPeriphID0、GPIOPeriphID1、GPIOPeriphID2 和 GPIOPeriphID3 这四个寄存器基本可以看成是一个 32 位的寄存器，每个寄存器包含了这个 32 位寄存器其中的 8 位，软件用这些寄存器来识别外设。

### GPIO 外设标识寄存器 2 (GPIOPeriphID2)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0xFE8

类型 RO, 复位 0x0000.0018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID2	RO	0x18	GPIO 外设 ID 寄存器 [23:16] 软件可通过此寄存器识别外设。

**寄存器 33: GPIO 外设标识寄存器 3 ( GPIOPeriphID3 ) ， 偏移量 0xFEC**

GPIOPeriphID0、GPIOPeriphID1、GPIOPeriphID2 和 GPIOPeriphID3 这四个寄存器基本可以看成是一个 32 位的寄存器，每个寄存器包含了这个 32 位寄存器其中的 8 位，软件用这些寄存器来识别外设。

## GPIO 外设标识寄存器 3 (GPIOPeriphID3)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFEC  
 类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID3	RO	0x01	GPIO 外设 ID 寄存器 [31:24] 软件可通过此寄存器识别外设。



### 寄存器 34: GPIO PrimeCell 标识寄存器 0 ( GPIOPCellID0 ) , 偏移量 0xFF0

GPIOCellID0、GPIOCellID1、GPIOCellID2 和 GPIOCellID3 寄存器是四个 8 位寄存器，这四个寄存器概念上可以看成是一个 32 位的寄存器。寄存器将作为一个标准的交叉外设 ( cross-peripheral ) 标识系统来使用。

#### GPIO PrimeCell 标识寄存器 0 (GPIOPCellID0)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFF0  
 类型 RO, 复位 0x0000.000D

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID0	RO	0x0D	GPIO PrimeCell ID 寄存器 [7:0] 向软件提供一个标准的交叉外设标识系统。

**寄存器 35: GPIO PrimeCell 标识寄存器 1 ( GPIOPCellID1 ) , 偏移量 0xFF4**

GPIOCellID0、GPIOCellID1、GPIOCellID2 和 GPIOCellID3 寄存器是四个 8 位寄存器，这四个寄存器概念上可以看成是一个 32 位的寄存器。寄存器将作为一个标准的交叉外设 ( cross-peripheral ) 标识系统来使用。

## GPIO PrimeCell 标识寄存器 1 (GPIOCellID1)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000

偏移量 0xFF4

类型 RO, 复位 0x0000.00F0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID1	RO	0xF0	GPIO PrimeCell ID 寄存器 [15:8] 向软件提供一个标准的交叉外设标识系统。

**寄存器 36: GPIO PrimeCell 标识寄存器 2 ( GPIOPCellID2 ) , 偏移量 0xFF8**

GPIOCellID0、GPIOCellID1、GPIOCellID2 和 GPIOCellID3 寄存器是四个 8 位寄存器，这四个寄存器概念上可以看成是一个 32 位的寄存器。寄存器将作为一个标准的交叉外设 ( cross-peripheral ) 标识系统来使用。

**GPIO PrimeCell 标识寄存器 2 (GPIOPCellID2)**

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFF8  
 类型 RO, 复位 0x0000.0005

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID2	RO	0x05	GPIO PrimeCell ID 寄存器 [23:16] 向软件提供一个标准的交叉外设标识系统。

**寄存器 37: GPIO PrimeCell 标识寄存器 3 ( GPIOPCellID3 ) , 偏移量 0xFFC**

GPIOCellID0、GPIOCellID1、GPIOCellID2 和 GPIOCellID3 寄存器是四个 8 位寄存器，这四个寄存器概念上可以看成是一个 32 位的寄存器。寄存器将作为一个标准的交叉外设 ( cross-peripheral ) 标识系统来使用。

## GPIO PrimeCell 标识寄存器 3 (GPIOCellID3)

GPIO 端口 A (APB) 基址: 0x4000.4000  
 GPIO 端口 A (AHB) 基址: 0x4005.8000  
 GPIO 端口 B (APB) 基址: 0x4000.5000  
 GPIO 端口 B (AHB) 基址: 0x4005.9000  
 GPIO 端口 C (APB) 基址: 0x4000.6000  
 GPIO 端口 C (AHB) 基址: 0x4005.A000  
 GPIO 端口 D (APB) 基址: 0x4000.7000  
 GPIO 端口 D (AHB) 基址: 0x4005.B000  
 GPIO 端口 E (APB) 基址: 0x4002.4000  
 GPIO 端口 E (AHB) 基址: 0x4005.C000  
 GPIO 端口 F (APB) 基址: 0x4002.5000  
 GPIO 端口 F (AHB) 基址: 0x4005.D000  
 GPIO 端口 G (APB) 基址: 0x4002.6000  
 GPIO 端口 G (AHB) 基址: 0x4005.E000  
 GPIO 端口 H (APB) 基址: 0x4002.7000  
 GPIO 端口 H (AHB) 基址: 0x4005.F000  
 GPIO 端口 J (APB) 基址: 0x4003.D000  
 GPIO 端口 J (AHB) 基址: 0x4006.0000  
 GPIO 端口 K (AHB) 基址: 0x4006.1000  
 GPIO 端口 L (AHB) 基址: 0x4006.2000  
 GPIO 端口 M (AHB) 基址: 0x4006.3000  
 GPIO 端口 N (AHB) 基址: 0x4006.4000  
 GPIO 端口 P (AHB) 基址: 0x4006.5000  
 偏移量 0xFFC  
 类型 RO, 复位 0x0000.00B1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID3	RO	0xB1	GPIO PrimeCell ID 寄存器 [31:24] 向软件提供一个标准的交叉外设标识系统。

## 11 通用定时器

可编程定时器可对驱动定时器输入管脚的外部事件进行计数或定时。该 Stellaris® 通用定时器模块 (GPTM) 包含六个 16/32 位 GPTM 模块和六个 32/64 位宽 GPTM 模块。每个 16/32 位 GPTM 模块提供两个 16 位的定时器/计数器 (称作 Timer A 和 Timer B)，用户可以将它们配置成独立运行的定时器或事件计数器，或将它们连接成一个 32 位定时器或一个 32 位实时时钟 (RTC)。每个 32/64 位宽 GPTM 模块为 Timer A 和 Timer B 提供 32 位定时器，可以将它们连接成一个 64 位定时器来运行。它还可以触发微型直接内存访问传输 ( $\mu$ DMA)。

此外，定时器也可以用来触发模数转换 (ADC)。由于所有通用定时器的 ADC 触发信号在到达 ADC 模块前一起进行或操作，因而只需使用一个定时器来触发 ADC 事件。

通用定时器 (GPT) 模块是 Stellaris 微控制器的一个定时资源。其它定时器资源还包括系统定时器 (SysTick) (见 118) 和 PWM 模块中的 PWM 定时器 (见“PWM 定时器”在 1155 页)。

通用定时器模块 (GPTM) 包含六个 16/32 位 GPTM 模块和六个 32/64 位宽 GPTM 时钟，有以下功能选项：

- 16/32 位运行模式：
  - 16 位或 32 位可编程的单次定时器
  - 16 位或 32 位可编程的周期定时器
  - 具有 8 位预分频的 16 位通用定时器
  - 当有 32.768 KHz 的外部时钟源时可作为 32 位的实时时钟
  - 16 位输入沿计数或定时捕获模式，并带 8 位的预分频器
  - 带 8 位预分频器的 16 位 PWM 模式以及软件编程实现的 PWM 信号反相输出
- 32/64 位运行模式：
  - 32 位或 64 位可编程的单次定时器
  - 32 位或 64 位可编程的周期定时器
  - 具有 16 位预分频的 32 位通用定时器
  - 当有 32.768 KHz 的外部时钟源时可作为 64 位的实时时钟
  - 带有 16 位预分频器的 32 位输入沿计数或定时捕获模块
  - 带有 16 位预分频器的 32 位 PWM 模式以及软件编程实现的 PWM 信号反相输出
- 可以向上或向下计数
- 十二个 16/32 位捕获比较 PWM 管脚 (CCP)
- 十二个 32/64 位捕捉比较 PWM 管脚 (CCP)
- 菊花链式的定时器模块允许一个定时器开始计时多路时钟事件
- 定时器同步功能允许所选的定时器在同一时钟周期开始计数

- ADC 事件触发器
- 当调试时，CPU 出现暂停标识时，用户可以停止定时器事件（包括 RTC 模式）
- 可以确定从产生中断到进入中断服务程序所经过的时间
- 用微型直接内存访问 ( $\mu$ DMA) 有效的传输数据
  - 每个定时器具有专用通道
  - 定时器中断响应突发请求

## 11.1 结构图

在结构框图中，可用的特定的捕获比较 PWM (CCP) 管脚取决于 Stellaris 设备。请参见表 11-1 在 654 页了解可用的 CCP 管脚和它们的定时器分配。

图 11-1. GPTM 模块的结构图

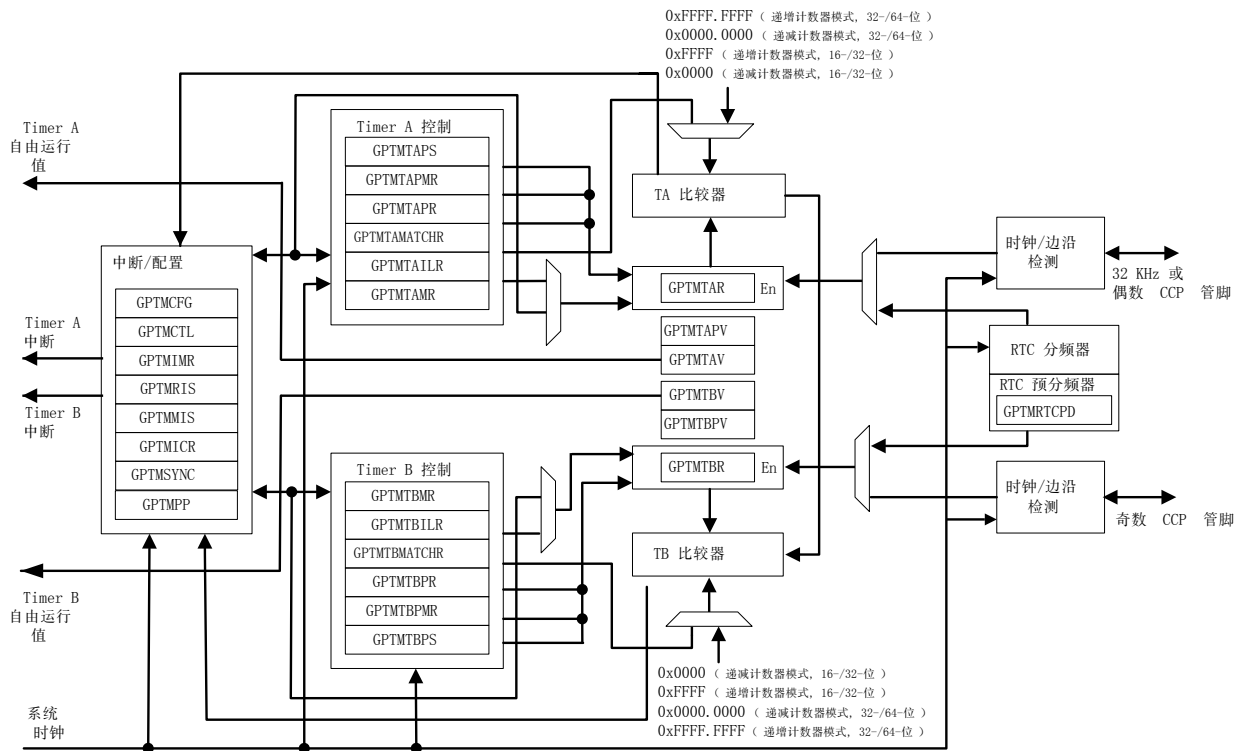


表 11-1. 可用的 CCP 管脚

定时器	递增/递减计数器	偶数 CCP 管脚	奇数 CCP 管脚
16/32 位 Timer 0	Timer A	T0CCP0	-
	Timer B	-	T0CCP1
16/32 位 Timer 1	Timer A	T1CCP0	-
	Timer B	-	T1CCP1
16/32 位 Timer 2	Timer A	T2CCP0	-
	Timer B	-	T2CCP1

表 11-1. 可用的 CCP 管脚 (续)

定时器	递增/递减计数器	偶数 CCP 管脚	奇数 CCP 管脚
16/32 位 Timer 3	Timer A	T3CCP0	-
	Timer B	-	T3CCP1
16/32 位 Timer 4	Timer A	T4CCP0	-
	Timer B	-	T4CCP1
16/32 位 Timer 5	Timer A	T5CCP0	-
	Timer B	-	T5CCP1
32/64 位宽 Timer 0	Timer A	WT0CCP0	-
	Timer B	-	WT0CCP1
32/64 位宽 Timer 1	Timer A	WT1CCP0	-
	Timer B	-	WT1CCP1
32/64 位宽 Timer 2	Timer A	WT2CCP0	-
	Timer B	-	WT2CCP1
32/64 位宽 Timer 3	Timer A	WT3CCP0	-
	Timer B	-	WT3CCP1
32/64 位宽 Timer 4	Timer A	WT4CCP0	-
	Timer B	-	WT4CCP1
32/64 位宽 Timer 5	Timer A	WT5CCP0	-
	Timer B	-	WT5CCP1

## 11.2 信号描述

以下表格列出了通用定时器模块的外部信号，并描述了各自的功能。通用定时器信号是一些 GPIO 信号的备用功能并且在复位时默认为 GPIO 信号。下表中的“复用管脚/赋值”一列给出了这些通用定时器信号的可能 GPIO 管脚位置。GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618页) 里的 AFSEL 位应被置位以选择通用定时器功能。括号中的数字是必须写入 GPIO 端口控制 (GPIOCTL) 寄存器 (636页) 里的 PMCN 域的编码，以便将通用定时器信号配置为指定的 GPIO 端口管脚。有关如何配置 GPIO 的更多信息，请参考“通用输入/输出口 (GPIO)”在 597页。

表 11-2. 通用定时器 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
T0CCP0	62	PF0 (7)	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 0。
	108	PL0 (7)			
T0CCP1	63	PF1 (7)	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 1。
	107	PL1 (7)			
T1CCP0	64	PF2 (7)	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 0。
	106	PL2 (7)			
	120	PJ0 (7)			
	136	PB4 (7)			
T1CCP1	65	PF3 (7)	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 1。
	105	PL3 (7)			
	121	PJ1 (7)			
	135	PB5 (7)			
T2CCP0	61	PF4 (7)	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 0。
	97	PB0 (7)			
	104	PL4 (7)			
	122	PJ2 (7)			

表 11-2. 通用定时器 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
T2CCP1	60 98 103 123	PF5 (7) PB1 (7) PL5 (7) PJ3 (7)	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 1。
T3CCP0	59 96 99 127	PF6 (7) PL6 (7) PB2 (7) PJ4 (7)	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 0。
T3CCP1	58 95 100 128	PF7 (7) PL7 (7) PB3 (7) PJ5 (7)	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 1。
T4CCP0	55 89 118 131	PG0 (7) PM0 (7) PC0 (7) PP0 (7)	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 0。
T4CCP1	54 88 117 132	PG1 (7) PM1 (7) PC1 (7) PP1 (7)	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 1。
T5CCP0	11 53 87 116	PP2 (7) PG2 (7) PM2 (7) PC2 (7)	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 0。
T5CCP1	52 86 115	PG3 (7) PM3 (7) PC3 (7)	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 1。
WT0CCP0	36 51 83 108	PC4 (7) PG4 (7) PM6 (7) PL0 (8)	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 0。
WT0CCP1	35 50 82 107	PC5 (7) PG5 (7) PM7 (7) PL1 (8)	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 1。
WT1CCP0	34 48 106 110	PC6 (7) PG6 (7) PL2 (8) PK6 (7)	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 0。
WT1CCP1	33 47 105 109	PC7 (7) PG7 (7) PL3 (8) PK7 (7)	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 1。
WT2CCP0	1 20 32 104	PD0 (7) PN2 (7) PH0 (7) PL4 (8)	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 0。
WT2CCP1	2 31 103 119	PD1 (7) PH1 (7) PL5 (8) PN3 (7)	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 1。



表 11-2. 通用定时器 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
WT3CCP0	3 26 71 96	PD2 (7) PH4 (7) PN4 (7) PL6 (8)	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 0。
WT3CCP1	4 23 70 95	PD3 (7) PH5 (7) PN5 (7) PL7 (8)	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 1。
WT4CCP0	22 69 89 141	PH6 (7) PN6 (7) PM0 (8) PD4 (7)	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 0。
WT4CCP1	21 68 88 142	PH7 (7) PN7 (7) PM1 (8) PD5 (7)	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 1。
WT5CCP0	28 87 143	PH2 (7) PM2 (8) PD6 (7)	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 0。
WT5CCP1	27 86 144	PH3 (7) PM3 (8) PD7 (7)	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 1。

a. TTL 表示管脚的电压水平与 TTL 一致。

### 11.3 功能描述

每个 GPTM 模块的主要元件包括两个自由运行的递增/递减计数器 ( 称作 TimerA 和 TimerB )、两个匹配寄存器、两个预分频器匹配寄存器、两个影子寄存器、两个加载/初始化寄存器和它们相关的控制功能。GPTM 的准确功能可由软件来控制，并通过寄存器接口进行配置。Timer A 和 Timer B 可以独立使用，在这种情况下，它们拥有针对 16/32 位 GPTM 模块的 16 位的计数范围和针对 32/64 位宽 GPTM 模块的 32 位的计数范围。此外，可以将 Timer A 和 Timer B 连在一起为 16/32 位 GPTM 模块提供 32 位的计数范围，以及为 32/64 位宽 GPTM 模块提供 64 位的计数范围。请注意仅可以在单独使用定时器时使用预分频器。

各个 GPTM 模块的可用模式在表 11-3 在 657 页中显示。请注意在单次触发或周期模式下递减计数时，预分频器用作真预分频器并且包含计数的最低位。在单次触发或周期模式下递增计数时，预分频器用作定时器扩展并且包含计数的最高位。在输入边沿计数、输入边沿计时和 PWM 模式下，预分频器总是用作定时器扩展，而不论计数方向。

表 11-3. 通用定时器功能

模式	定时器使用	计数方向	计数器大小		预分频器大小 <sup>a</sup>		预分频器行为 ( 计数方向 )
			16/32 位 GPTM	32/64 位宽 GPTM	16/32 位 GPTM	32/64 位宽 GPTM	
单次触发	独立	递增或递减	16 位	32 位	8 位	16 位	定时器扩展 ( 递增 )， 预分频器 ( 递减 )
	连接	递增或递减	32 位	64 位	-	-	N/A
周期	独立	递增或递减	16 位	32 位	8 位	16 位	定时器扩展 ( 递增 )， 预分频器 ( 递减 )
	连接	递增或递减	32 位	64 位	-	-	N/A
RTC	连接	递增	32 位	64 位	-	-	N/A

表 11-3. 通用定时器功能 (续)

模式	定时器使用	计数方向	计数器大小		预分频器大小 <sup>a</sup>		预分频器行为 (计数方向)
			16/32 位 GPTM	32/64 位宽 GPTM	16/32 位 GPTM	32/64 位宽 GPTM	
边沿计数	独立	递增或递减	16 位	32 位	8 位	16 位	定时器扩展 (两者均有)
边沿时间	独立	递增或递减	16 位	32 位	8 位	16 位	定时器扩展 (两者均有)
PWM	独立	递减	16 位	32 位	8 位	16 位	定时器扩展

a. 仅可以在单独使用定时器时使用预分频器

软件使用 GPTM 配置 (GPTMCFG) 寄存器 (见 675 页)、GPTM Timer A 模式 (GPTMTAMR) 寄存器 (见 676 页) 和 GPTM Timer B 模式 (GPTMTBMR) 寄存器 (见 679 页) 配置 GPTM。处于其中的一个连接模式中时, Timer A 和 Timer B 仅能在一个模式中运行。但是, 在独立模式中配置时, 可以在任何独立模式的组合中自由配置 Timer A 和 Timer B。

### 11.3.1 GPTM 复位条件

GPTM 模块复位后处于未激活状态, 所有控制寄存器均被清零, 同时进入默认状态。计数器 Timer A 和 Timer B 连同与它们对应的加载寄存器: GPTM Timer A 间隔加载 (GPTMTAILR) 寄存器 (见 699 页)、GPTM Timer B 间隔加载 (GPTMTBILR) 寄存器 (见 700 页)、影子寄存器: GPTM Timer A 值 (GPTMTAV) 寄存器 (见 709 页) 和 GPTM Timer B 值 (GPTMTBV) 寄存器 (见 710 页) 一起被初始化为 1。预分频计数器初始化为 0x00: GPTM Timer A 预分频 (GPTMTAPR) 寄存器 (见 703 页)、GPTM Timer B 预分频 (GPTMTBPR) 寄存器 (见 704 页)、GPTM Timer A 预分频快照 (GPTMTAPS) 寄存器 (见 712 页)、GPTM Timer B 预分频快照 (GPTMTBPS) 寄存器 (见 713 页)、GPTM Timer A 预分频值 (GPTMTAPV) 寄存器 (见 714 页) 和 GPTM Timer B 预分频值 (GPTMTBPV) 寄存器 (见 715 页)。

### 11.3.2 定时器模式

此部分描述了各种定时器模式的运行。在连接模式中使用 Timer A 和 Timer B 时, 仅必须使用 Timer A 控制和状态位; 不需要使用 Timer B 控制和状态位。通过向 GPTM 配置 (GPTMCFG) 寄存器写入 0x4, 可将 GPTM 配置为独立/分离模式 (见 675 页)。在以下部分中, 变量“n”用于位域和寄存器名称中, 表示 Timer A 函数或 Timer B 函数。在本节中, 递减计数模式中的超时间事件为 0x0, 而在递增模式中为 GPTM 定时器 n 间隔装载寄存器 (GPTMTnILR) 和可选的 GPTM 定时器 n 预分频寄存器 (GPTMTnPMR) 中的数值。

#### 11.3.2.1 单次触发/周期定时器模式

选择单次触发模式还是周期模式由写入 GPTM 定时器 n 模式 (GPTMTnMR) 寄存器中 TnMR 域的值来决定 (见 676 页)。是递增计数还是递减计数由 GPTMTnMR 寄存器中的 TnCDIR 位来决定。

当软件置位了 GPTM 控制 (GPTMCTL) 寄存器 (见 682 页) 中的 TnEN 位, 定时器开始从 0x0 开始递增计数或从预加载的值开始递减计数。另外, 如果 GPTMTnMR 寄存器中的 TnWOT 被置位了, 一旦 TnEN 位被置位, 定时器就会等待一个触发来开始计数 (详情请参考“等待触发模式”在 667 页)。表 11-4 在 658 页显示启用定时器时加载到定时器寄存器的值。

表 11-4. 单次触发或周期模式下启用定时器时的计数器值

寄存器	递减模式	递增模式
TnR	GPTMTnILR	0x0
TnV	GPTMTnILR	0x0
TnPS	独立模式下为 GPTMTnPR; 连接模式下不可用	独立模式下为 0x0; 连接模式下不可用

表 11-4. 单次触发或周期模式下启用定时器时的计数器值 (续)

寄存器	递减模式	递增模式
TnPv	独立模式下为 GPTMTnPR；连接模式下不可用	独立模式下为 0x0；连接模式下不可用

当定时器递减计数并且到达超时事件 (0x0) 时，定时器将会在下一个时钟周期从 GPTMTnILR 和 GPTMTnPR 寄存器重新加载其初值。当定时器递增计数并达到超时事件 (GPTMTnILR 和可选的 GPTMTnPR 寄存器中的数值) 时，定时器重新加载 0x0。如果配置为单次触发模式，则定时器停止计数并将 GPTMCTL 寄存器的 TnEN 位清零。如果配置为周期定时器，则定时器会在下一个时钟周期开始再次计数。

在周期、快照模式中 (GPTMTnMR 寄存器中的 TnMR 域为 0x2，且 TnSNAPS 已置位)，发生超时事件时的定时器值被加载到 GPTMTnR 寄存器，而预分频器的值被加载到 GPTMTnPS 寄存器。自由运行计数器的值显示在 GPTMTnV 寄存器中，自由运行预分频器的值显示在 GPTMTnPV 寄存器中。通过这种方法，软件能校验快照值和自由运行定时器的当前值，从而确定从发生中断到进入中断服务程序之间所用时间。定时器配置为单次触发模式时，快照模式不可用。

除了重新加载计数值，当达到超时事件时，GPTM 也会触发并产生中断。GPTM 将 GPTM 原始中断状态 (GPTMRIS) 寄存器 (见 691 页) 中的 TnTORIS 位置位，并保持该值直到向 GPTM 中断清除 (GPTMICR) 寄存器 (见 697 页) 执行写操作将其清零。如果在 GPTM 中断屏蔽 (GPTMIMR) 寄存器 (见 688 页) 中启用超时中断，GPTM 还将 GPTM 屏蔽的中断状态 (GPTMMIS) 寄存器 (见 694 页) 中的 TnTOMIS 位置位。通过将 GPTMTnMR 寄存器中的 TnMIE 位置位，当定时器的值与加载到 GPTM 定时器 n 匹配 (GPTMTnMATCHR) 和 GPTM 定时器 n 预分频匹配值 (GPTMTnPMR) 寄存器的值相等时，也能产生中断条件。该中断和超时中断具有同样的状态、屏蔽和清零方式，但使用匹配中断位实现功能 (例如，原始中断状态通过 GPTM 原始中断状态 (GPTMRIS) 寄存器的 TnMRIS 位监控)。注：中断状态位并不会由硬件更新，除非 GPTMTnMR 寄存器的 TnMIE 位已置位，这与超时中断的行为不同。通过将 GPTMCTL 中的 TnOTE 位置位以启用 ADC 触发。通过配置并启用合适的  $\mu$ DMA 通道来启用  $\mu$ DMA 触发。参见“通道配置”在 538 页。

如果软件在计数器递减的过程中更新了 GPTMTnILR 或 GPTMTnPR 寄存器，计数器会在下一个时钟周期加载新值，并且如果 GPTMTnMR 寄存器中的 TnILD 位清零，则从新值开始继续计数。如果 TnILD 位置位，计数器在下一个超时后加载新值。如果软件在计数器递增的过程中更新了 GPTMTnILR 或 GPTMTnPR 寄存器，超时事件会在下一个时钟周期更改为新值。如果软件在计数器递增或递减的过程中更新了 GPTM 定时器 n 值 (GPTMTnV) 寄存器，则计数器在下一个时钟周期加载新值并从新值继续计数。如果软件更新了 GPTMTnMATCHR 或 GPTMTnPMR 寄存器，新值将在下一个时钟周期得到反映 (如果 GPTMTnMR 寄存器的 TnMRSU 位已清零)。如果 TnMRSU 位置位，新值在下一个超时前不会生效。

在 64 位模式中使用 32/64 位宽定时器模块时，必须以“访问连接的 32/64 位宽 GPTM 寄存器值”在 669 页中描述的方法访问某些寄存器。

如果 GPTMCTL 寄存器的 TnSTALL 位置位，调试器将会在停止处理器的时候冻结计数器。当处理器恢复执行时定时器继续计数。

下面的表格显示了在使用预分频器时 16 位自由运行的定时器的各种配置。所有的值都是以时钟频率 80-MHz (时钟周期  $T_c=12.5$  ns) 作为标准进行计算。预分频器只能在 16/32 位定时器配置为 16 位模式和 32/64 位定时器配置为 32 位模式时使用。

表 11-5. 带预分频器的 16 位定时器配置

预分频 (8 位值)	定时器时钟 (Tc) 编号 <sup>a</sup>	最大时间	单位
00000000	1	0.8192	ms
00000001	2	1.6384	ms
00000010	3	2.4576	ms
-----	--	--	--

表 11-5. 带预分频器的 16 位定时器配置 (续)

预分频 (8 位值)	定时器时钟 (Tc) 编号 <sup>a</sup>	最大时间	单位
11111101	254	208.0768	ms
11111110	255	208.896	ms
11111111	256	209.7152	ms

a. Tc 表示时钟周期。

下面的表格显示了在使用预分频器时 32 位自由运行的定时器 (配置为 32/64 位模式) 的各种配置。所有的值都是以时钟频率 80-MHz (时钟周期 Tc=12.5 ns) 作为标准进行计算。

表 11-6. 带预分频器配置的 32 位定时器 (配置为 32/64 位模式)

预分频 (16 位值)	定时器时钟 (Tc) 编号 <sup>a</sup>	最大时间	单位
0x0000	1	53.687	s
0x0001	2	107.374	s
0x0002	3	214.748	s
-----	--	--	--
0xFFFFD	65534	0.879	10 <sup>6</sup>
0xFFFFE	65535	1.759	10 <sup>6</sup>
0xFFFFF	65536	3.518	10 <sup>6</sup>

a. Tc 表示时钟周期。

### 11.3.2.2 实时时钟模式

在实时时钟 (RTC) 模式中, Timer A 和 Timer B 寄存器连在一起被配置为递增计数器。复位后, 首次选择 RTC 模式时, 计数器加载的值为 0x1。所有后续加载的值必须写入 GPTM Timer n 间隔加载 (GPTMTnILR) 寄存器 (参考 699页)。表 11-7 在 660页 显示启用定时器时加载到定时器寄存器的值。

表 11-7. RTC 模式下启用定时器时的计数器值

寄存器	递减模式	递增模式
TnR	不可用	0x1
TnV	不可用	0x1
TnPS	不可用	不可用
TnPV	不可用	不可用

在 RTC 模式中, 要求 CCP0 输入时钟为 32.768 KHz。然后将时钟信号分频为 1 Hz, 将其传送给计数器的输入端。

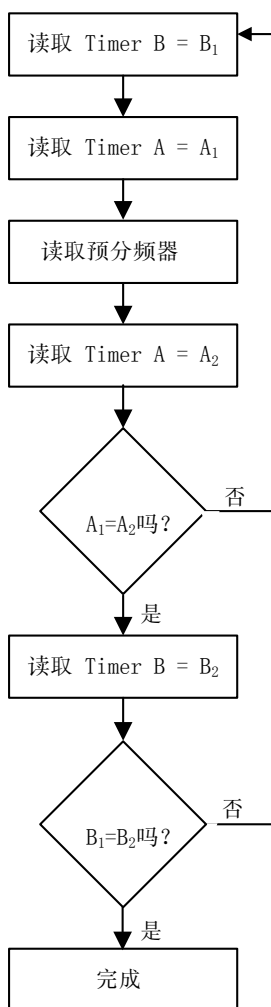
在软件写 GPTMCTL 寄存器中的 TAEN 位时, 计数器从其预加载的值 0x1 开始递增计数。如果当前的计数值与 GPTMTnMRTATCHR 寄存器中预加载的值相匹配, GPTM 将使 GPTMRIS 中的 RTCRIS 位生效, 并且继续计数, 直到发生硬件复位或者软件将其禁用 (通过清零 TAEN 位)。当定时器值达到终端计数时, 定时器将会返回到 0x0 继续递增计数。如果在 GPTMIMR 中启用 RTC 中断, 则 GPTM 还将置位 GPTMMIS 中的 RTCMIS 位并产生控制器中断。通过写 GPTMICR 中的 RTCCINT 位将状态标记清除。

在此模式中, GPTMTnR 和 GPTMTnV 寄存器总是具有相同值。

在 RTC 模式中使用 32/64 位宽定时器模块时, 必须以“访问连接的 32/64 位宽 GPTM 寄存器值”在 669页中描述的方法访问某些寄存器。

RTC 预分频器的值可通过 GPTM RTC 预分频 (GPTMRTCPD) 寄存器读取。要确保 RTC 值的一致性，软件应遵循图 11-2 在 661页中的详细过程。

图 11-2. 读取 RTC 值



除产生中断之外，还可以产生  $\mu$ DMA 触发。通过配置并启用合适的  $\mu$ DMA 通道来启用  $\mu$ DMA 触发。参见“通道配置”在 538页。

如果 GPTMCTL 寄存器的 TASTALL 位置位，在已设置 GPTMCTL 的 RTCEN 位时，调试器停止处理器将不会冻结计数器。

### 11.3.2.3 输入边沿计数模式

**注意：** 对于上升沿检测，输入信号必须在上升沿到达高电平后至少保持两个系统时钟周期。同样，下降沿检测在到达下降沿的低电平后必须至少保持两个系统时钟周期。有鉴于此，边沿检测的最大输入频率为系统频率的 1/4。

在边沿检测模式中，定时器被配置为 24 位或 48 位递增或递减计数器，包括带有存储在 GPTM Timer n 预分频 (GPTMTnPR) 寄存器中的高计数值和 GPTMTnR 寄存器中低位的可选预分频器。在此模式中，定时器能够捕获三种事件类型：上升沿、下降沿、或上升/下降沿。为了把定时器设置为边沿计数模式，GPTMTnMR 寄存器的 TnCMR 位必须清零。定时器计数时所采用的边沿类型由 GPTMCTL 寄存器的 TnEVENT 域决定。在递减模式的初始化过程中，需对 GPTMTnMATCHR 和 GPTMTnPMR

寄存器进行配置，使 GPTMTnILR 和 GPTMTnPR 寄存器以及 GPTMTnMATCHR 和 GPTMTnPMR 寄存器之间的差值等于必须计算的边沿事件的数目。在递增计数模式中，定时器从 0x0 开始计数到 GPTMTnMATCHR 和 GPTMTnPMR 寄存器中的值。表 11-8 在 662页 显示启用定时器时加载到定时器寄存器的值。

表 11-8. 输入边沿计数模式下启用定时器时的计数器值

寄存器	递减模式	递增模式
TnR	GPTMTnILR	0x0
TnV	GPTMTnILR	0x0
TnPS	GPTMTnPR	0x0
TnPV	GPTMTnPR	0x0

在软件写 GPTM 控制 (GPTMCTL) 寄存器的 TnEN 位时，定时器将启用以捕获事件。CCP 管脚上每输入一个事件，计数器的值就递减或递增 1，直到事件计数的值与 GPTMTnMATCHR 和 GPTMTnPMR 的值匹配。计数匹配时，GPTM 让 GPTM 原始中断状态 (GPTMRIS) 寄存器中的 CnMRIS 位有效，并保持该值直到向 GPTM 中断清除 (GPTMICR) 寄存器执行写操作将其清零。如果在 GPTM 中断屏蔽 (GPTMIMR) 寄存器中启用捕获模式匹配中断，GPTM 还会将 GPTM 屏蔽的中断状态 (GPTMMIS) 寄存器中的 CnMMIS 位置位。在此模式中，GPTMTnR 和 GPTMTnPS 寄存器包含输入事件的计数，同时 GPTMTnV 和 GPTMTnPV 寄存器包含自由运行定时器的值和自由运行预分频器的值。

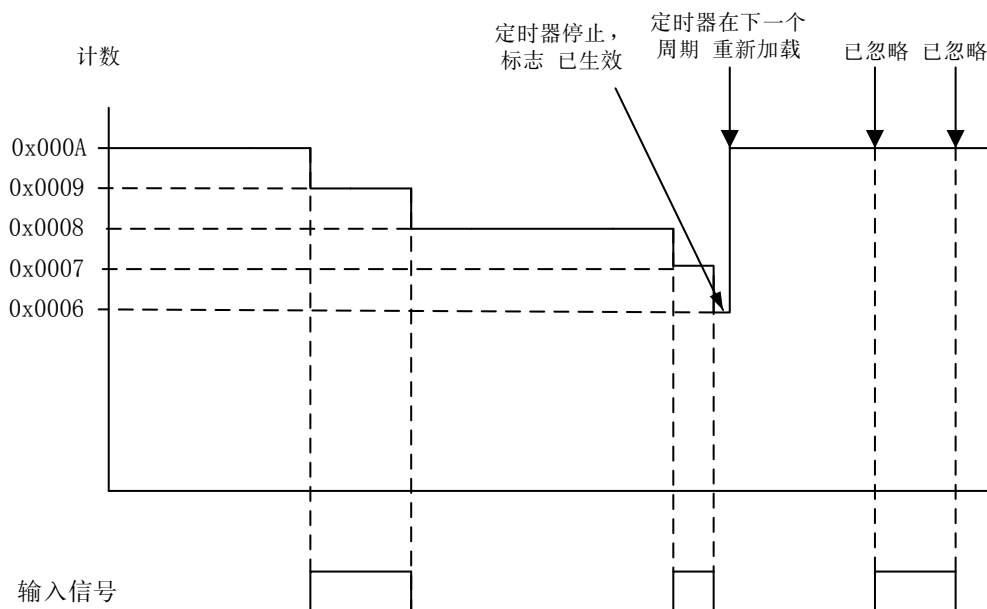
除产生中断之外，还可以产生 ADC 和/或  $\mu$ DMA 触发。将 GPTMCTL 中的 TnOTE 位置位来启用 ADC 触发。通过配置并启用合适的  $\mu$ DMA 通道可以启用  $\mu$ DMA 触发。参见“通道配置”在 538页。

在递减模式中达到匹配值后，计数器使用 GPTMTnILR 和 GPTMTnPR 寄存器中的值执行重装操作，并且由于 GPTM 自动将 GPTMCTL 寄存器的 TnEN 清零，因此计数器停止计数。一旦事件计数值满足要求，接下来的所有事件都将被忽略，直到通过软件重新将 TnEN 启用。在递增模式中，定时器重新加载 0x0 并继续计数。

图 11-3 在 663页 显示了输入边沿计数模式的工作情况。在这种情况下，定时器的初值设置为 GPTMTnILR=0x000A，匹配值 GPTMTnMATCHR=0x0006。此时，需计数四个边沿事件。计数器配置为检测输入信号的双边沿。

注：在当前计数值与 GPTMTnMATCHR 寄存器中的值匹配之后，定时器自动将 TnEN 位清零，因此最后两个边沿没有计算在内。

图 11-3. 输入边沿计数模式实例，递减计数



#### 11.3.2.4 输入边沿计时模式

**注意：** 对于上升沿检测，输入信号必须在上升沿到达高电平后至少保持两个系统时钟周期。同样，下降沿检测在到达下降沿的低电平后必须至少保持两个系统时钟周期。有鉴于此，边沿检测的最大输入频率为系统频率的 1/4。

在边沿计时模式中，定时器被配置为 24 位或 48 位递增或递减计数器，包括带有存储在 GPTMTnPR 寄存器中的高定时器值和 GPTMTnILR 寄存器中低位的可选预分频器。在此模式中，在递减计数时，定时器被初始化为 GPTMTnILR 和 GPTMTnPR 寄存器中加载的数值，递增计数时为 0x0。定时器能够捕获三种事件类型：上升沿、下降沿、或上升/下降沿。通过置位 GPTMTnMR 寄存器的 TnCMR 位可将定时器置于边沿定时模式，而定时器捕获时采用的事件类型由 GPTMCTL 寄存器的 TnEVENT 域来决定。表 11-9 在 663 页显示启用定时器时加载到定时器寄存器的值。

表 11-9. 输入事件计数模式下启用定时器时的计数器值

寄存器	递减模式	递增模式
TnR	GPTMTnILR	0x0
TnV	GPTMTnILR	0x0
TnPS	GPTMTnPR	0x0
TnPV	GPTMTnPR	0x0

在软件写 GPTMCTL 寄存器的 TnEN 位时，定时器将启用以捕获事件。在检测到所选的输入事件时，从 GPTMTnR 和 GPTMTnPS 寄存器中捕获定时器计数器的当前值，且该值可通过微控制器来读取。然后 GPTM 让 GPTM 原始中断状态 (GPTMRIS) 寄存器中的 CnERIS 位有效，并保持该值直到向 GPTM 中断清除 (GPTMICR) 寄存器执行写操作将其清零。如果在 GPTM 中断屏蔽 (GPTMIMR) 寄存器中启用捕获模式事件中断，GPTM 还会将 GPTM 屏蔽的中断状态 (GPTMMIS) 寄存器中的 CnEMIS 位置位。这种模式下，GPTMTnR 和 GPTMTnPS 寄存器包含发生选定输入事件的时间，同时 GPTMTnV 和 GPTMTnPV 寄存器包含自由运行定时器的值和自由运行预分频器的值。读取这些寄存器可以判定从发生中断到进入 ISR (中断服务程序) 所用时间。

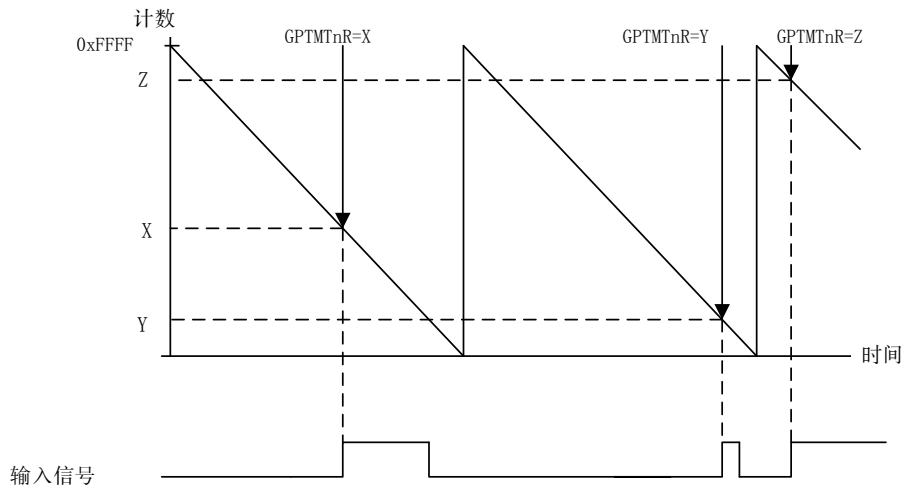
除产生中断之外，还可以产生 ADC 和/或  $\mu$ DMA 触发。将 GPTMCTL 中的 TnOTE 位置位来启用 ADC 触发。通过配置并启用合适的  $\mu$ DMA 通道可以启用  $\mu$ DMA 触发。参见“通道配置”在 538 页。

在捕获到事件之后，定时器不会停止计数。它会继续计数，直至 TnEN 位清零。当定时器达到超时值时，在递增模式中重新加载 0x0，在递减模式中重新加载来自 GPTMTnILR 和 GPTMTnPR 寄存器的值。

图 11-4 在 664 页显示了输入边沿定时模式的工作原理。在图中，假定定时器的初值为默认值 0xFFFF，定时器配置为捕获上升沿事件。

每当检测到上升沿事件时，当前计数值便加载到 GPTMTnR 和 GPTMTnPS 寄存器中，且该值一直保持在寄存器中直到检测到下一个上升沿（在此上升沿处，新的计数值加载到 GPTMTnR 和 GPTMTnPS 寄存器中）。

图 11-4. 16 位输入边沿计时模式实例



注意：在边沿计时模式下操作时，计数器在启用预分频器时按模  $2^{24}$  计数，或在未启用时按  $2^{16}$  计数。如果边沿有可能比计数更长，则可执行另一定时器，以确保检测到丢失的边沿。

### 11.3.2.5 PWM 模式

GPTM 支持简单的 PWM 生成模式。在 PWM 模式中，定时器被配置为 24 位或 48 位递减计数器，初值由 GPTMTnILR 和 GPTMTnPR 寄存器定义。在此模式中，PWM 频率和周期是同步事件，因此保证了无故障。将 GPTMTnMR 寄存器的 TnAMS 位给置为 0x1，将 nCMR 位给置为 0x0，并将 TnMR 域给置为 0x1 或 0x2 即可启用 PWM 模式。表 11-10 在 664 页显示启用定时器时加载到定时器寄存器的值。

表 11-10. PWM 模式下启用定时器时的计数器值

寄存器	递减模式	递增模式
GPTMTnR	GPTMTnILR	不可用
GPTMTnV	GPTMTnILR	不可用
GPTMTnPS	GPTMTnPR	不可用
GPTMTnPV	GPTMTnPR	不可用

在软件写 GPTMCTL 寄存器的 TnEN 位时，计数器开始递减计数，直到计数值到达 0x0。另外，如果 GPTMTnMR 寄存器中的 TnWOT 被置位了，一旦 TnEN 位被置位，定时器就会等待一个触发来开始计数（详情请参考“等待触发模式”在 667 页）。在周期模式的下一个计数周期，计数器将



GPTMTnILR 和 GPTMTnPR 寄存器中的值重新载入，作为它的初值，并继续计数直到计数器因软件将 GPTMCTL 寄存器的 TnEN 位清零而被禁用。定时器能够基于三种事件类型产生中断。这三种事件类型为：上升沿、下降沿或上升/下降沿。该事件通过 GPTMCTL 寄存器的 TnEVENT 域配置，而中断通过设置 GPTMTnMR 寄存器的 TnPWMIE 位启用。发生该事件时，将 GPTM 原始中断状态 (GPTMRIS) 寄存器中的 CnERIS 位置位，并保持该值直到向 GPTM 中断清除 (GPTMICR) 寄存器执行写操作将其清零。如果在 GPTM 中断屏蔽 (GPTMIMR) 寄存器中启用捕获模式事件中断，GPTM 还会将 GPTM 屏蔽的中断状态 (GPTMMIS) 寄存器中的 CnEMIS 位置位。注：中断状态位不会更新，除非 TnPWMIE 位已置位。

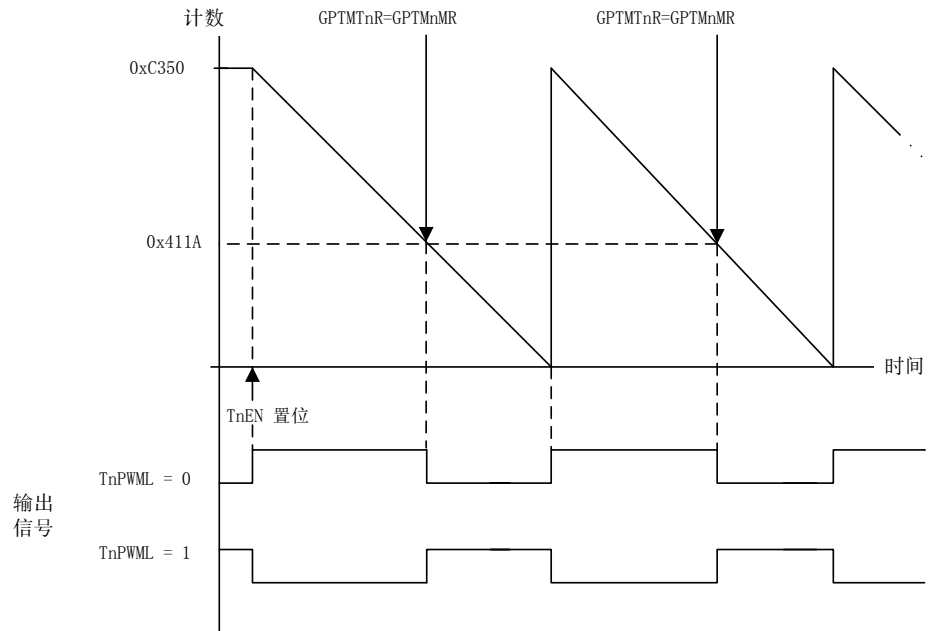
在此模式中，GPTMTnR 和 GPTMTnV 寄存器总是具有相同的值，GPTMPnPS 和 GPTMTnPV 寄存器亦然。

当计数器的值与 GPTMTnILR 和 GPTMTnPR 寄存器的值（计数器的初始状态）相等时，输出 PWM 信号生效，当计数器的值与 GPTMTnMATCHR 和 GPTMTnPMR 寄存器的值相等时，输出 PWM 信号失效。通过将 GPTMCTL 寄存器的 TnPWML 位置位，软件可实现将输出 PWM 信号反相的功能。

**注意：** 如果启用了 PWM 输出反相，将翻转边沿检测中断行为。因此，如果上升沿中断触发已置位，且 PWM 反相生成上升沿，将不存在有效的事件触发中断。相反，中断在 PWM 信号的下降沿生成。

图 11-5 在 665 页显示了在输入时钟为 50 MHz 以及 TnPWML 为 0 的情况下，如何产生周期为 1ms、占空比为 66% 的输出 PWM ( TnPWML =1 时，占空比为 33% )。在这个例子中，初值为 GPTMTnILR =0xC350，匹配值 GPTMTnMATCHR =0x411A。

图 11-5. 16 位 PWM 模式实例



使用 GPTMSYNC 寄存器同步定时器时，定时器必须进行适当配置，以避免在 CCP 输出上出现故障。GPTMTnMR 寄存器的 PLO 和 MRSU 位必须置位。图 11-6 在 666 页显示了在 PLO 和 MRSU 位被置位并且 GPTMTnMATCHR 值大于 GPTMTnILR 值时，CCP 输出如何操作。

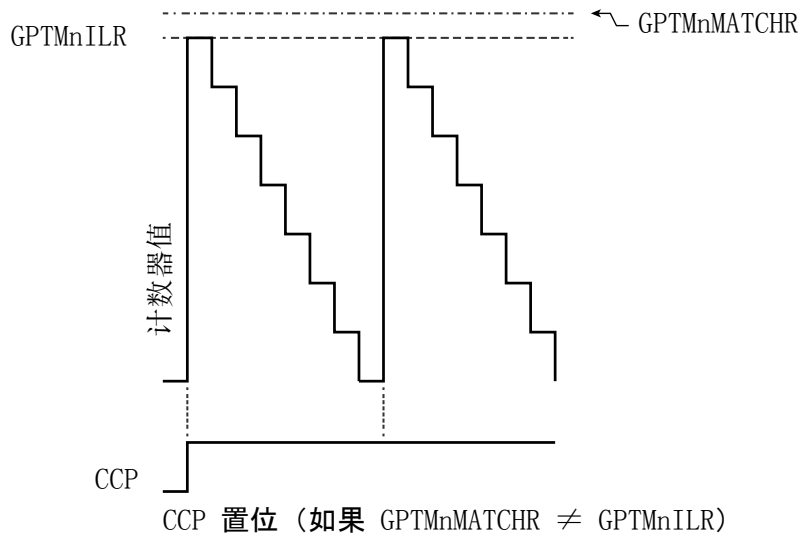
图 11-6. CCP 输出， $GPTMTnMATCHR > GPTMTnILR$ 

图 11-7 在 666 页显示在 PLO 和 MRSU 位被置位并且  $GPTMTnMATCHR$  值等于  $GPTMTnILR$  值时，CCP 输出如何操作。此时，如果 PLO 位等于 0，CCP 信号在载入  $GPTMTnILR$  值时拉高，匹配基本上将被忽略。

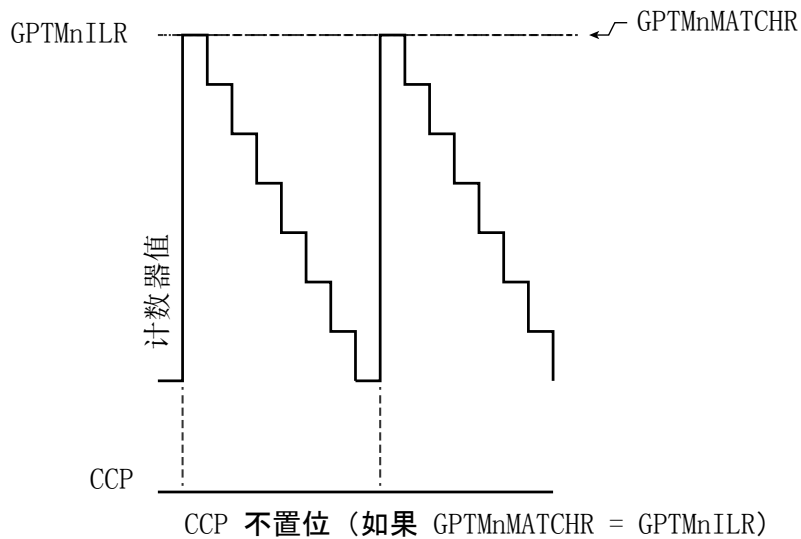
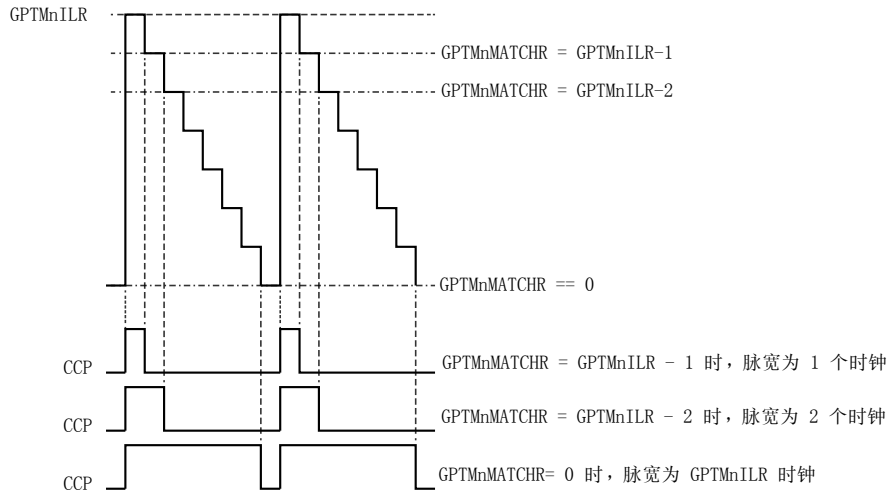
图 11-7. CCP 输出， $GPTMTnMATCHR = GPTMTnILR$ 

图 11-8 在 667 页显示在 PLO 和 MRSU 位被置位并且  $GPTMTnILR$  大于  $GPTMTnMATCHR$  值时，CCP 输出如何操作。

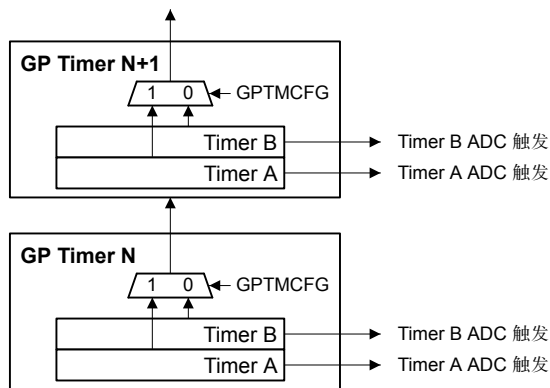
图 11-8. CCP 输出，GPTMnILR &gt; GPTMnMATCHR



### 11.3.3 等待触发模式

等待触发模式是允许使用菊花链式的定时器模块。比如，一旦配置好，一个独立的定时器可以通过定时器触发来初始化多路时钟事件。通过置位 GPTMnMR 寄存器中的 TnWOT 位以启用等待触发模式。当 TnWOT 置位时，Timer N+1 只有在等到菊花链中它的上一个 (Timer N) 接收到了超时事件时，才会开始计数。菊花链的配置一般是如下形式的：GPTM1 跟着 GPTM0，GPTM2 跟着 GPTM1，... 等等。如果 Timer A 被配置为 32 位 (16/32 位模式) 或 64 位 (32/64 位宽模式) 定时器 (通过 GPTMCFG 寄存器的 GPTMCFG 域控制)，则触发下一模块的 Timer A。如果 Timer A 被配置为 16 位 (16/32 位模式) 或 32 位 (32/64 位宽模式) 定时器，则触发同一模块中的 Timer B，由 Timer B 触发下一模块的 Timer A。必须注意：GPTM0 的 TAWOT 位永远不会置位。图 11-9 在 667 页显示了 GPTMCFG 位如何影响菊花链。这些功能对单次触发、周期和 PWM 模式都是有效的。

图 11-9. 定时器菊花链



### 11.3.4 同步通用定时器模块

GPTM0 模块中的 GPTM 同步控制 (GPTMSYNC) 寄存器可以用于同步所选的定时器，以便同时开始计数。将 GPTMSYNC 寄存器的某位置位将导致相关定时器执行超时事件的动作。在同步定时器时未产生中断。如果正在连接模式中使用定时器，仅 Timer A 的位必须在 GPTMSYNC 寄存器中置位。

表 11-11 在 668页 显示在各种定时器模式中同步定时器时所执行超时事件的动作。

表 11-11. GPTM 模式的超时动作

模式	计数目录	超时动作
32 和 64 位单次触发 (连接定时器)	—	N/A
32 和 64 位周期 (连接定时器)	递减	计数值 = ILR
	递增	计数值 = 0
32 和 64 位 RTC (连接定时器)	递增	计数值 = 0
16 和 32 位单次触发 (独立/分离定时器)	—	N/A
16 和 32 位周期 (独立/分离定时器)	递减	计数值 = ILR
	递增	计数值 = 0
16 和 32 位边沿计数 (独立/分离定时器)	递减	计数值 = ILR
	递增	计数值 = 0
16 和 32 位边沿计时 (独立/分离定时器)	递减	计数值 = ILR
	递增	计数值 = 0
16 和 32 位 PWM	递减	计数值 = ILR

### 11.3.5 DMA 操作

每一个定时器都有一个专用的  $\mu$ DMA 通道，并且可以给  $\mu$ DMA 控制器提供一个请求信号。该请求信号是突发类型，并且在定时器原始中断条件出现时都会发生。 $\mu$ DMA 传输的仲裁数目应与每次定时器事件出现时应该传输的数据单元数相同。

例如，要传输 256 个项目，每 10 ms 传输 8 个，配置定时器为周期定时器，每 10 ms 发生一次中断。配置  $\mu$ DMA 传输总量为 256 个项目，每次突发传送 8 个项目。每次定时器超时， $\mu$ DMA 控制器就会传输 8 个项目，直到所有 256 个项目传输完毕。

不需要其他特殊步骤就能使定时器用于  $\mu$ DMA 操作。关于给  $\mu$ DMA 控制器编程的更多详情，请参阅“微型直接存储器访问 ( $\mu$ DMA)”在 534页。

### 11.3.6 访问连接的 16/32 位 GPTM 寄存器值

通过向 GPTM 配置 (GPTMCFG) 寄存器中的 GPTMCFG 位域写入 0x0 或 0x1，可将 GPTM 配置为连接模式。在这两种配置下，某些 16/32 位 GPTM 寄存器会连接在一起形成伪 32 位寄存器。这些寄存器包括：

- GPTM Timer A 间隔加载 (GPTMTAILR) 寄存器 [15:0]，见 699页
- GPTM Timer B 间隔加载 (GPTMTBILR) 寄存器 [15:0]，见 700页
- GPTM Timer A (GPTMTAR) 寄存器 [15:0]，见 707页
- GPTM Timer B (GPTMTBR) 寄存器 [15:0]，见 708页
- GPTM Timer A 值 (GPTMTAV) 寄存器 [15:0]，见 709页
- GPTM Timer B 值 (GPTMTBV) 寄存器 [15:0]，见 710页
- GPTM Timer A 匹配 (GPTMTAMATCHR) 寄存器 [15:0]，见 701页

- GPTM Timer B 匹配 (GPTMTBMATCHR) 寄存器 [15:0]，见 702 页

在 32 位模式中，GPTM 把对 GPTMTAILR 的 32 位写访问转换为对 GPTMTAILR 和 GPTMTBILR 的写访问。这样，写操作最终的顺序为：

```
GPTMTBILR[15:0]:GPTMTAILR[15:0]
```

同样，对 GPTMTAR 的 32 位读操作返回的值为：

```
GPTMTBR[15:0]:GPTMTAR[15:0]
```

对 GPTMTAV 的 32 位读操作返回的值为：

```
GPTMTBV[15:0]:GPTMTAV[15:0]
```

### 11.3.7 访问连接的 32/64 位宽 GPTM 寄存器值

在 32/64 位宽 GPTM 模块上，连接的寄存器值（64 位和 48 位）并不适合作为大于处理器内核的总线宽度访问的位宽。在连接定时器模式和独立的定时器模式中，在使用预分频器时，软件必须针对该值执行原子访问，以使之一致。在读取大于 32 位的定时器值时，软件应遵循以下步骤：

1. 读取合适的 Timer B 寄存器或预分频器寄存器。
2. 读取相应的 Timer A 寄存器。
3. 重新读取 Timer B 寄存器或预分频器寄存器。
4. 对比首次和第二次读取的 Timer B 或预分频器值。如果它们相同，定时器的值一致。如果它们不同，再次重复步骤 1-4，使它们相同。

以下伪代码说明了该过程：

```
high = timer_high;
low = timer_low;
if (high != timer_high); //low overflowed into high
{
    high = timer_high;
    low = timer_low;
}
```

必须以此方法读取的寄存器如下所示：

- 64 位读取
  - GPTMTAV 和 GPTMTBV
  - GPTMTAR 和 GPTMTBR
- 48 位读取
  - GPTMTAR 和 GPTMTAPS

- GPTMTBR 和 GPTMTBPS
- GPTMTAV 和 GPTMTAPV
- GPTMTBV 和 GPTMTBPV

同样，还必须通过在写入低位之前写入高位来执行写操作，如下所示：

1. 写入合适的 Timer B 寄存器或预分频器寄存器。
2. 写入相应的 Timer A 寄存器。

必须以此方法写入的寄存器如下所示：

- 64 位写入
  - GPTMTAV 和 GPTMTBV
  - GPTMTAMATCHR 和 GPTMTBMATCHR
  - GPTMTAILR 和 GPTMTBILR
- 48 位写入
  - GPTMTAV 和 GPTMTAPV
  - GPTMTBV 和 GPTMTBPV
  - GPTMTAMATCHR 和 GPTMTAPMR
  - GPTMTBMATCHR 和 GPTMTBPMR
  - GPTMTAILR 和 GPTMTAPR
  - GPTMTBILR 和 GPTMTBPR

在写入 64 位值时，如果连续两次写入到以上“64 位写入”标题下列出的任何寄存器中，不论寄存器是在 Timer A 或 Timer B 中，或者如果在写入 Timer B 中相应的寄存器之前写入寄存器 Timer A，那么系统将通过 GPTMRIS 寄存器中的 WUERIS 位报告错误。如果该错误未被屏蔽，即可引发中断。请注意，因为预分频器的使用是可选项，所以该错误并未报告给预分频器寄存器。因此，程序员必须小心遵循上文所述的协议。

## 11.4 初始化和配置

要使用 GPTM，必须将 RCGCTIMER 或 RCGCWTIMER 寄存器中相应的 TIMERN 位置位（见 300 页和 319 页）。如果使用任何 CCP 管脚，必须通过 RCGCGPIO 寄存器启用相关 GPIO 模块的时钟（见 302 页）。欲了解需要启用哪些 GPIO 端口，请参考表 23-4 在 1271 页。在 GPIOCTL 寄存器中配置 PMCn 域，以将 CCP 信号分配到合适的管脚（见 636 页和表 23-5 在 1280 页）。

针对每种支持的定时器模式，本节提供了模块的初始化以及配置示例。

### 11.4.1 单次触发/周期定时器模式

将 GPTM 配置为单次触发和周期模式的步骤如下：

1. 确保先禁用定时器（将 GPTMCTL 寄存器的 TnEN 位清零），然后再进行更改操作。

2. 向 GPTM 配置 (GPTMCFG) 寄存器写入 0x0000.0000。
3. 配置 GPTM Timer n 模式寄存器 (GPTMTnMR) 的 TnMR 域 :
  - a. 写入 0x1 设为单次触发模式。
  - b. 写入 0x2 设为周期模式。
4. ( 可选 ) 配置 GPTMTnMR 寄存器中的 TnSNAPS、TnWOT、TnMTE 和 TnCDIR 位, 以选择是否捕获超时自由运行定时器的值、使用外部触发来启动计数、配置一个额外的触发或中断, 以及递增还是递减计数。
5. 将初值加载到 GPTM Timer n 间隔加载 (GPTMTnILR) 寄存器。
6. 如果需要中断, 将 GPTM 中断屏蔽寄存器 (GPTMIMR) 中的相应位置位。
7. 在 GPTMCTL 寄存器中将 TnEN 位置位来启用定时器并开始计数。
8. 查询 GPTMRIS 寄存器或等待中断的产生 ( 如果启用 )。在这两种情况下, 通过向 GPTM 中断清零寄存器 (GPTMICR) 的相应位写 1 将状态标志清零。

如果 GPTMTnMR 寄存器中的 TnMIE 位被置位, 则 GPTMRIS 寄存器中的 RTCRIS 位被置位, 并且定时器继续计数。在单次触发模式中, 定时器在超时事件之后停止计数。需重复上述步骤才能将定时器重新启用。在周期模式中配置的定时器在发生超时事件后将会重新载入并继续计数。

#### 11.4.2 实时时钟 (RTC) 模式

要使用 RTC 模式, 定时器必须在偶数 CCP 管脚上有一个 32.768 KHz 的输入信号。要启用 RTC 功能, 请遵循以下步骤:

1. 确保先禁用定时器 ( TAEN 位清零 ), 然后再进行更改操作。
2. 向 GPTM 配置 (GPTMCFG) 寄存器写入 0x0000.0001。
3. 向 GPTM Timer n 匹配寄存器 (GPTMTnMATCHR) 写入匹配值。
4. 根据需要将 GPTM 控制寄存器 (GPTMCTL) 中的 RTCEN 位置位/清零。
5. 如果需要中断, 将 GPTM 中断屏蔽寄存器 (GPTMIMR) 寄存器中的 RTCIM 位置位。
6. 在 GPTMCTL 寄存器中将 TAEN 位置位来启用定时器并开始计数。

当定时器计数等于 GPTMTnMATCHR 寄存器中的值时, GPTM 确认 GPTMRIS 寄存器中的 RTCRIS 位, 并且继续计数, 直到 Timer A 被禁用或发生硬件复位。通过写入 GPTMICR 寄存器中的 RTCCINT 位以清除中断。

#### 11.4.3 输入边沿计数模式

将定时器配置为输入边沿计数模式的步骤如下:

1. 确保先禁用定时器 ( TnEN 位清零 ), 然后再进行更改操作。
2. 向 GPTM 配置 (GPTMCFG) 寄存器写入 0x0000.0004。
3. 在 GPTM 定时器模式 (GPTMTnMR) 寄存器中, 写 0x0 到 TnCMR 域, 写 0x3 到 TnMR 域。

4. 通过对 GPTM 控制 (GPTMCTL) 寄存器的 TnEVENT 域进行写操作来配置定时器捕获操作的事件类型。
5. 如果使用预分频器，则将预分频值写入 GPTM 定时器 n 预分频寄存器 (GPTMTnPR)。
6. 将初值加载到 GPTM Timer n 间隔加载 (GPTMTnILR) 寄存器。
7. 将事件计数加载到 GPTM 定时器 n 匹配 (GPTMTnMATCHR) 寄存器。
8. 如果需要中断，将 GPTM 中断屏蔽 (GPTMIMR) 寄存器中的 CnMIM 位置位。
9. 置位 GPTMCTL 寄存器的 TnEN 位来启用定时器并开始等待边沿事件。
10. 查询 GPTMRIS 寄存器的 CnMRIS 位或等待中断的产生 (如果启用)。在这两种情况下，通过向 GPTM 中断清除 (GPTMICR) 寄存器的 CnMCINT 位写 1 可将状态标志清零。

在输入边沿计数模式中递减计数时，定时器在检测到定义好的边沿事件数之后停止。需确保 TnEN 位清零并重复执行 #4 在 672页 到 #9 在 672页 才能重新启用定时器。

#### 11.4.4 输入边沿定时模式

将定时器配置为输入边沿定时模式的步骤如下：

1. 确保先禁用定时器 ( TnEN 位清零 )，然后再进行更改操作。
2. 向 GPTM 配置 (GPTMCFG) 寄存器写入 0x0000.0004。
3. 在 GPTM 定时器模式 (GPTMTnMR) 寄存器中，写 0x1 到 TnCMR 域，写 0x3 到 TnMR 域。
4. 通过对 GPTM 控制 (GPTMCTL) 寄存器的 TnEVENT 域进行写操作来配置定时器捕获操作的事件类型。
5. 如果使用预分频器，则将预分频值写入 GPTM 定时器 n 预分频寄存器 (GPTMTnPR)。
6. 将初值加载到 GPTM Timer n 间隔加载 (GPTMTnILR) 寄存器。
7. 如果需要中断，将 GPTM 中断屏蔽 (GPTMIMR) 寄存器中的 CnEIM 位置位。
8. 在 GPTM 控制 (GPTMCTL) 寄存器中将 TnEN 位置位来启用定时器并开始计数。
9. 查询 GPTMRIS 寄存器的 CnERIS 位或等待中断的产生 (如果启用)。在这两种情况下，通过向 GPTM 中断清除 (GPTMICR) 寄存器的 CnECINT 位写 1 可将状态标志清零。事件发生的时间可通过读 GPTM Timer n (GPTMTnR) 寄存器来获得。

在输入边沿定时模式中，定时器在检测到边沿事件之后继续运行，但通过写 GPTMTnILR 寄存器可在任何时候改变定时器间隔。此改变在写操作的下一个周期生效。

#### 11.4.5 PWM 模式

将定时器配置为 PWM 模式的步骤如下：

1. 确保先禁用定时器 ( TnEN 位清零 )，然后再进行更改操作。
2. 向 GPTM 配置 (GPTMCFG) 寄存器写入 0x0000.0004。
3. 在 GPTM 定时器模式 (GPTMTnMR) 寄存器中，将 TnAMS 位给置为 0x1，将 TnCMR 位给置为 0x0，将 TnMR 域给置为 0x2。



4. 在 GPTM 控制 (GPTMCTL) 寄存器的 TnEVENT 域中，配置 PWM 信号的输出状态 (是否需要反相)。
5. 如果使用预分频器，则将预分频值写入 GPTM 定时器 n 预分频寄存器 (GPTMTnPR)。
6. 如果使用 PWM 中断，配置 GPTMCTL 寄存器中 TnEVENT 域的中断条件，并且通过将 GPTMTnMR 寄存器中的 TnPWMIE 位置位来启用中断。注：PWM 输出反相时，将翻转边沿检测中断行为 (参考 682 页)。
7. 将初值加载到 GPTM Timer n 间隔加载 (GPTMTnILR) 寄存器。
8. 向 GPTM 定时器 n 匹配 (GPTMTnMATCHR) 寄存器加载匹配值。
9. 将 GPTM 控制 (GPTMCTL) 寄存器的 TnEN 位置位来启用定时器并开始输出 PWM 信号。

在 PWM 定时模式中，定时器在产生 PWM 信号之后继续运行。通过写 GPTMTnILR 寄存器可在任何时候对 PWM 周期进行调整，此改变在写操作的下一个周期生效。

## 11.5 寄存器映射

表 11-12 在 673 页列出了 GPTM 寄存器。所列出的地址偏移量是寄存器相对于定时器的基址的 16 进制增量：

- 16/32 位 Timer 0 : 0x4003.0000
- 16/32 位 Timer 1 : 0x4003.1000
- 16/32 位 Timer 2 : 0x4003.2000
- 16/32 位 Timer 3 : 0x4003.3000
- 16/32 位 Timer 4 : 0x4003.4000
- 16/32 位 Timer 5 : 0x4003.5000
- 32/64 位宽 Timer 0 : 0x4003.6000
- 32/64 位宽 Timer 1 : 0x4003.7000
- 32/64 位宽 Timer 2 : 0x4004.C000
- 32/64 位宽 Timer 3 : 0x4004.D000
- 32/64 位宽 Timer 4 : 0x4004.E000
- 32/64 位宽 Timer 5 : 0x4004.F000

GPTM 外设属性 (GPTMPP) 寄存器中的 SIZE 域可识别模块是具有 16/32 位还是 32/64 位宽定时器。

请注意，要给寄存器编程，必须先启用通用定时器模块时钟 (参考 300 页或 319 页)。在访问任何定时器模块寄存器前，启用定时器模块时钟后必须有 3 个系统时钟的延迟。

表 11-12. 定时器 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	GPTMCFG	R/W	0x0000.0000	GPTM 配置寄存器	675
0x004	GPTMTAMR	R/W	0x0000.0000	GPTM Timer A 模式寄存器	676
0x008	GPTMTBMR	R/W	0x0000.0000	GPTM Timer B 模式寄存器	679
0x00C	GPTMCTL	R/W	0x0000.0000	GPTM 控制寄存器	682
0x010	GPTMSYNC	R/W	0x0000.0000	GPTM 同步寄存器	685
0x018	GPTMIMR	R/W	0x0000.0000	GPTM 中断屏蔽寄存器	688

表 11-12. 定时器 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0x01C	GPTMRIS	RO	0x0000.0000	GPTM 原始中断状态寄存器	691
0x020	GPTMMIS	RO	0x0000.0000	GPTM 屏蔽的中断状态寄存器	694
0x024	GPTMICR	W1C	0x0000.0000	GPTM 中断清除寄存器	697
0x028	GPTMTAILR	R/W	0xFFFF.FFFF	GPTM Timer A 间隔加载寄存器	699
0x02C	GPTMTBILR	R/W	-	GPTM Timer B 间隔加载寄存器	700
0x030	GPTMTAMATCHR	R/W	0xFFFF.FFFF	GPTM Timer A 匹配寄存器	701
0x034	GPTMTBMATCHR	R/W	-	GPTM Timer B 匹配寄存器	702
0x038	GPTMTAPR	R/W	0x0000.0000	GPTM Timer A 预分频寄存器	703
0x03C	GPTMTBPR	R/W	0x0000.0000	GPTM Timer B 预分频寄存器	704
0x040	GPTMTAPMR	R/W	0x0000.0000	GPTM TimerA 预分频匹配寄存器	705
0x044	GPTMTBPMR	R/W	0x0000.0000	GPTM TimerB 预分频匹配寄存器	706
0x048	GPTMTAPR	RO	0xFFFF.FFFF	GPTM Timer A 寄存器	707
0x04C	GPTMTBPR	RO	-	GPTM Timer B 寄存器	708
0x050	GPTMTAV	RW	0xFFFF.FFFF	GPTM Timer A 值寄存器	709
0x054	GPTMTBV	RW	-	GPTM Timer B 值寄存器	710
0x058	GPTMRTCPD	RO	0x0000.7FFF	GPTM RTC 预分频寄存器	711
0x05C	GPTMTAPS	RO	0x0000.0000	GPTM Timer A 预分频快照寄存器	712
0x060	GPTMTBPS	RO	0x0000.0000	GPTM Timer B 预分频快照寄存器	713
0x064	GPTMTAPV	RO	0x0000.0000	GPTM Timer A 预分频值寄存器	714
0x068	GPTMTBPV	RO	0x0000.0000	GPTM Timer B 预分频值寄存器	715
0xFC0	GPTMPP	RO	0x0000.0000	GPTM 外设属性寄存器	716

## 11.6 寄存器描述

下文将按地址偏移量的数字顺序列出 GPTM 寄存器，并对它们进行描述。

**寄存器 1: GPTM 配置寄存器 ( GPTMCFG ) , 偏移量 0x000**

该寄存器对 GPTM 模块的全局操作进行配置。写入该寄存器的值决定 GPTM 处于 32 或 64 位模式 ( 连接定时器 ) 还是处于 16 或 32 位模式 ( 独立、分离定时器 )。

**重要:** 该寄存器中的位应仅在 GPTMCTL 寄存器中的 TAEN 和 TBEN 位清零时发生更改。

**GPTM 配置寄存器 (GPTMCFG)**

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x000  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													GPTMCFG		
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

2:0	GPTMCFG	R/W	0x0	GPTM 配置
-----	---------	-----	-----	---------

GPTMCFG 值定义如下:

值	描述
0x0	对于 16/32 位定时器, 该值选择 32 位定时器配置。 对于 32/64 位宽定时器, 该值选择 64 位定时器配置。
0x1	对于 16/32 位定时器, 该值选择 32 位实时时钟 (RTC) 计数器配置。 对于 32/64 位宽定时器, 该值选择 64 位实时时钟 (RTC) 计数器配置。
0x2-0x3	保留
0x4	对于 16/32 位定时器, 该值选择 16 位定时器配置。 对于 32/64 位宽定时器, 该值选择 32 位定时器配置。 功能由 GPTMTAMR 和 GPTMTBMR 的位 1:0 来控制。
0x5-0x7	保留

## 寄存器 2: GPTM Timer A 模式寄存器 ( GPTMTAMR ) , 偏移量 0x004

该寄存器根据 GPTMCFG 寄存器中所选的配置来进一步配置 GPTM。在 PWM 模式中，将 TAAMS 位置位，将 TACMR 位清零，并配置 TAMR 域为 0x1 或 0x2。

该寄存器在 Timer A 独立使用时控制它的模式。在 Timer A 和 Timer B 连接在一起时，该寄存器控制 Timer A 和 Timer B 的模式，并忽略 GPTMTBMR 的内容。

**重要:** 除 TCACT 位域以外，该寄存器中的所有其他位应仅在 GPTMCTL 寄存器中的 TAEN 位清零时发生更改。

### GPTM Timer A 模式寄存器 (GPTMTAMR)

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x004  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				TAPLO	TAMRSU	TAPWMIE	TAILD	TASNAPS	TAWOT	TAMIE	TACDIR	TAAMS	TACMR	TAMR	
类型	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
11	TAPLO	R/W	0	GPTM Timer A PWM 传统操作  <b>值 描述</b> 0 定时器达到 0 后重新载入 GPTMTAILR 时，执行 CCP 管脚拉低的传统操作。 1 定时器达到 0 后重新载入 GPTMTAILR 时，CCP 拉高。  该位仅在 PWM 模式中有效。
10	TAMRSU	R/W	0	GPTM Timer A 匹配寄存器更新  <b>值 描述</b> 0 在下一个时钟周期更新 GPTMTAMATCHR 寄存器和 GPTMTAPR 寄存器 ( 如使用 )。 1 在下次超时时更新 GPTMTAMATCHR 寄存器和 GPTMTAPR 寄存器 ( 如使用 )。  当该位被置位时，如果定时器被禁用 ( TAEN 清零 )，则定时器启用时，GPTMTAMATCHR 和 GPTMTAPR 将更新。如果定时器暂停 ( TASTALL 置位 )，GPTMTAMATCHR 和 GPTMTAPR 将根据该位的配置更新。

位/域	名称	类型	复位	描述
9	TAPWMIE	R/W	0	<p>GPTM Timer A PWM 中断启用</p> <p>根据 GPTMCTL 寄存器中 TAEVENT 域的定义，在 PWM 模式中，该位启用在上升沿、下降沿或 CCP 输出的双边沿中断功能。</p> <p>值 描述</p> <p>0 中断已禁用。</p> <p>1 中断已启用。</p> <p>该位仅在 PWM 模式中有效。</p>
8	TAILD	R/W	0	<p>GPTM Timer A 间隔加载写操作</p> <p>值 描述</p> <p>0 在下一个周期根据 GPTMTAILR 寄存器中的值更新 GPTMTAR 和 GPTMTAV 寄存器。同时在下一个周期根据 GPTMTAPR 寄存器中的值更新 GPTMTAPS 和 GPTMTAPV 寄存器。</p> <p>1 在下次超时时根据 GPTMTAILR 和 GPTMTAPV 寄存器中的值更新 GPTMTAR 和 GPTMTAV 寄存器。在下次超时时根据 GPTMTAPR 寄存器中的值更新 GPTMTAPS 寄存器。</p> <p>请注意该位的状态在递增计数时没有影响。</p> <p>定时器已启用并正在运行时，适用上面的位描述。当该位被置位时，如果定时器被禁用（TAEN 清零），则定时器启用时，GPTMTAR、GPTMTAV、GPTMTAPS 和 GPTMTAPV 将更新。如果定时器暂停（TASTALL 置位），GPTMTAR 和 GPTMTAPS 将根据该位的配置更新。</p>
7	TASNAPS	R/W	0	<p>GPTM Timer A 快照模式</p> <p>值 描述</p> <p>0 快照模式已禁用。</p> <p>1 如果在周期模式中配置 Timer A，则 Timer A 的实际自由运行值在发生超时事件时被加载到 GPTM Timer A (GPTMTAR) 寄存器。如果使用了定时器预分频器，将把预分频器快照载入 GPTM Timer A (GPTMTAPR)。</p>
6	TAWOT	R/W	0	<p>GPTM Timer A 等待触发</p> <p>值 描述</p> <p>0 Timer A 一启用就开始计数。</p> <p>1 如果 Timer A 启用（GPTMCTL 寄存器中的 TAEN 置位），Timer A 不会马上开始计数，直到它从菊花链中上一个位置的定时器收到一个触发信号，参考图 11-9 在 667 页。这些功能对单次触发、周期和 PWM 模式都是有效的。</p> <p>通用定时器模块 0 Timer A 的该位必须清零。</p>
5	TAMIE	R/W	0	<p>GPTM Timer A 匹配中断启用</p> <p>值 描述</p> <p>0 匹配中断已禁用。</p> <p>1 在单次触发和周期模式中达到 GPTMTAMATCHR 寄存器中的匹配值时产生中断。</p>

位/域	名称	类型	复位	描述
4	TACDIR	R/W	0	<p>GPTM Timer A 计数方向</p> <p>值 描述</p> <p>0 定时器递减计数。</p> <p>1 定时器递增计数。在递增计数时，定时器从 0x0 开始。</p> <p>在 PWM 或 RTC 模式中时，忽略该位的状态。PWM 模式始终递减计数，而 RTC 模式始终递增计数。</p>
3	TAAMS	R/W	0	<p>GPTM Timer A 交替模式选择</p> <p>TAAMS 值定义如下：</p> <p>值 描述</p> <p>0 捕获模式被启用。</p> <p>1 PWM 模式被启用。</p> <p>注意： 要启用 PWM 模式，必须将 TACMR 位清零并将 TAMR 域配置为 0x1 或 0x2。</p>
2	TACMR	R/W	0	<p>GPTM Timer A 捕获模式</p> <p>TACMR 值定义如下：</p> <p>值 描述</p> <p>0 边沿计数模式</p> <p>1 边沿计时模式</p>
1:0	TAMR	R/W	0x0	<p>GPTM Timer A 模式</p> <p>TAMR 值定义如下：</p> <p>值 描述</p> <p>0x0 保留</p> <p>0x1 单次触发定时器模式</p> <p>0x2 周期定时器模式</p> <p>0x3 捕获模式</p> <p>定时器模式基于 GPTMCFG 寄存器的 2:0 位定义的定时器配置。</p>

### 寄存器 3: GPTM Timer B 模式寄存器 ( GPTMTBMR ) , 偏移量 0x008

该寄存器根据 GPTMCFG 寄存器中所选的配置来进一步配置 GPTM。在 PWM 模式中，将 TBAMS 位置位，将 TBCMR 位清零，并配置 TBMR 域为 0x1 或 0x2。

该寄存器在 Timer B 独立使用时控制它的模式。在 Timer A 和 Timer B 连接时，该寄存器将被忽略，并且 GPTMTBMR 控制 Timer A 和 Timer B 的模式。

**重要:** 除 TCACT 位域以外，该寄存器中的所有其他位应仅在 GPTMCTL 寄存器中的 TBEN 位清零时发生更改。

#### GPTM Timer B 模式寄存器 (GPTMTBMR)

16/32 位 Timer 0 基址: 0x4003.0000

16/32 位 Timer 1 基址: 0x4003.1000

16/32 位 Timer 2 基址: 0x4003.2000

16/32 位 Timer 3 基址: 0x4003.3000

16/32 位 Timer 4 基址: 0x4003.4000

16/32 位 Timer 5 基址: 0x4003.5000

32/64 位宽 Timer 0 基址: 0x4003.6000

32/64 位宽 Timer 1 基址: 0x4003.7000

32/64 位宽 Timer 2 基址: 0x4004.C000

32/64 位宽 Timer 3 基址: 0x4004.D000

32/64 位宽 Timer 4 基址: 0x4004.E000

32/64 位宽 Timer 5 基址: 0x4004.F000

偏移量 0x008

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				TBPLO	TBMRSU	TBPWMIE	TBILD	TBSNAPS	TBWOT	TBMIE	TBCDIR	TBAMS	TBCMR	TBMR	
类型	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
11	TBPLO	R/W	0	GPTM Timer B PWM 传统操作  值 描述 0 定时器达到 0 后重新载入 GPTMTAILR 时，执行 CCP 管脚拉低的传统操作。 1 定时器达到 0 后重新载入 GPTMTAILR 时，CCP 拉高。  该位仅在 PWM 模式中有效。
10	TBMRSU	R/W	0	GPTM Timer B 匹配寄存器更新  值 描述 0 在下一个时钟周期更新 GPTMTBMATCHR 寄存器和 GPTMTBPR 寄存器 ( 如使用 )。 1 在下次超时时更新 GPTMTBMATCHR 寄存器和 GPTMTBPR 寄存器 ( 如使用 )。  当该位被置位时，如果定时器被禁用 ( TBEN 清零 )，则定时器启用时，GPTMTBMATCHR 和 GPTMTBPR 将更新。如果定时器暂停 ( TBSTALL 置位 )，GPTMTBMATCHR 和 GPTMTBPR 将根据该位的配置更新。

位/域	名称	类型	复位	描述
9	TBPWMIE	R/W	0	<p>GPTM Timer B PWM 中断启用</p> <p>根据 GPTMCTL 寄存器中 TBEVENT 域的定义，在 PWM 模式中，该位启用在上沿、下降沿或 CCP 输出的双边沿中断功能。</p> <p>值 描述</p> <p>0 中断已禁用。</p> <p>1 中断已启用。</p> <p>该位仅在 PWM 模式中有效。</p>
8	TBILD	R/W	0	<p>GPTM Timer B 间隔加载写操作</p> <p>值 描述</p> <p>0 在下一个周期根据 GPTMTBILR 寄存器中的值更新 GPTMTBR 和 GPTMTBV 寄存器。同时在下一个周期根据 GPTMTBPR 寄存器中的值更新 GPTMTBPS 和 GPTMTBPV 寄存器。</p> <p>1 在下次超时时根据 GPTMTBILR 寄存器中的值更新 GPTMTBR 和 GPTMTBV 寄存器。同时在下次超时时根据 GPTMTBPR 寄存器中的值更新 GPTMTBPS 和 GPTMTBPV 寄存器。</p> <p>请注意该位的状态在递增计数时没有影响。</p> <p>定时器已启用并正在运行时，适用上面的位描述。当该位被置位时，如果定时器被禁用（TBEN 清零），则定时器启用时，GPTMTBR、GPTMTBV、GPTMTBPS 和 GPTMTBPV 将更新。如果定时器暂停（TBSTALL 置位），GPTMTBR 和 GPTMTBPS 将根据该位的配置更新。</p>
7	TBSNAPS	R/W	0	<p>GPTM Timer B 快照模式</p> <p>值 描述</p> <p>0 快照模式已禁用。</p> <p>1 如果在周期模式中配置 Timer B，则 Timer B 的实际自由运行值在发生超时事件时被加载到 GPTM Timer B (GPTMTBR) 寄存器。如果使用了定时器预分频器，将把预分频器快照载入 GPTM Timer B (GPTMTBPR)。</p>
6	TBWOT	R/W	0	<p>GPTM Timer B 等待触发</p> <p>值 描述</p> <p>0 Timer B 一启用就开始计数。</p> <p>1 如果 Timer B 启用（GPTMCTL 寄存器中的 TBEN 置位），Timer B 不会马上开始计数，直到它从菊花链中上一个位置的定时器收到一个触发信号，见图 11-9 在 667 页。这些功能对单次触发、周期和 PWM 模式都是有效的。</p>
5	TBMIE	R/W	0	<p>GPTM Timer B 匹配中断启用</p> <p>值 描述</p> <p>0 匹配中断已禁用。</p> <p>1 在单次触发和周期模式中达到 GPTMTBMATCHR 寄存器中的匹配值时产生中断。</p>



位/域	名称	类型	复位	描述
4	TBCDIR	R/W	0	<p>GPTM Timer B 计数方向</p> <p>值 描述</p> <p>0 定时器递减计数。</p> <p>1 定时器递增计数。在递增计数时，定时器从 0x0 开始。</p> <p>在 PWM 或 RTC 模式中时，忽略该位的状态。PWM 模式始终递减计数，而 RTC 模式始终递增计数。</p>
3	TBAMS	R/W	0	<p>GPTM Timer B 交替模式选择</p> <p>TBAMS 值定义如下：</p> <p>值 描述</p> <p>0 捕获模式被启用。</p> <p>1 PWM 模式被启用。</p> <p>注意： 要启用 PWM 模式，必须将 TBCMR 位清零，并将 TBMR 域配置为 0x1 或 0x2。</p>
2	TBCMR	R/W	0	<p>GPTM Timer B 捕获模式</p> <p>TBCMR 值定义如下：</p> <p>值 描述</p> <p>0 边沿计数模式</p> <p>1 边沿计时模式</p>
1:0	TBMR	R/W	0x0	<p>GPTM Timer B 模式</p> <p>TBMR 值定义如下：</p> <p>值 描述</p> <p>0x0 保留</p> <p>0x1 单次触发定时器模式</p> <p>0x2 周期定时器模式</p> <p>0x3 捕获模式</p> <p>定时器模式基于 GPTMCFG 寄存器的 2:0 位定义的定时器配置。</p>

## 寄存器 4: GPTM 控制寄存器 ( GPTMCTL ) , 偏移量 0x00C

该寄存器与 GPTMCFG 和 GMTMTnMR 寄存器一起使用, 对定时器配置进行微小调整, 并启用其它特性诸如定时器停止和输出触发信号。输出触发器可以用来启动 ADC 模块上的传输。

**重要:** 该寄存器中的位应仅在相应定时器的 TnEN 位清零时发生更改。

### GPTM 控制寄存器 (GPTMCTL)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x00C  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	TBPWML	TBOTE	保留	TBEVENT	TBSTALL	TBEN	保留	TAPWML	TAOTE	RTCEN	TAEVENT	TASTALL	TAEN		
类型	RO	R/W	R/W	RO	R/W	R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:15	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
14	TBPWML	R/W	0	GPTM Timer B PWM 输出水平 TBPWML 值定义如下:  值 描述 0 输入不受影响。 1 输入反相。
13	TBOTE	R/W	0	GPTM Timer B 输出触发启用 TBOTE 值定义如下:  值 描述 0 输出 Timer B ADC 触发被禁用。 1 输出 Timer B ADC 触发启用。  此外, 必须用 ADCEMUX 寄存器中的 EMn 位启用 ADC, 并且选择定时器为触发源 ( 见 773页 ) 。
12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
11:10	TBEVENT	R/W	0x0	<p>GPTM Timer B 事件模式</p> <p>TBEVENT 值定义如下：</p> <p>值 描述</p> <p>0x0 上升沿</p> <p>0x1 下降沿</p> <p>0x2 保留</p> <p>0x3 双边沿</p> <p>注意： 如果启用了 PWM 输出反相，将翻转边沿检测中断行为。因此，如果上升沿中断触发已置位，且 PWM 反相生成上升沿，将不存在有效的事件触发中断。相反，中断在 PWM 信号的下降沿生成。</p>
9	TBSTALL	R/W	0	<p>GPTM Timer B 停止启用</p> <p>TBSTALL 值定义如下：</p> <p>值 描述</p> <p>0 当处理器被调试器停止时，Timer B 继续计数。</p> <p>1 当处理器被调试器停止时，Timer B 停止计数。</p> <p>如果处理器正常运行，忽略 TBSTALL 位。</p>
8	TBEN	R/W	0	<p>GPTM Timer B 启用</p> <p>TBEN 值定义如下：</p> <p>值 描述</p> <p>0 Timer B 已禁用。</p> <p>1 根据 GPTMCFG 寄存器，启用 TimerB 并开始计数或启用捕获逻辑。</p>
7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
6	TAPWML	R/W	0	<p>GPTM Timer A PWM 输出电平</p> <p>TAPWML 值定义如下：</p> <p>值 描述</p> <p>0 输入不受影响。</p> <p>1 输入反相。</p>
5	TAOTE	R/W	0	<p>GPTM Timer A 输出触发启用</p> <p>TAOTE 值定义如下：</p> <p>值 描述</p> <p>0 输出 Timer A ADC 触发被禁用。</p> <p>1 输出 Timer A ADC 触发启用。</p> <p>此外，必须用 ADCMUX 寄存器中的 EMn 位启用 ADC，并且选择定时器为触发源（见 773 页）。</p>

位/域	名称	类型	复位	描述
4	RTCEN	R/W	0	<p>GPTM RTC 停止启用 RTCEN 值定义如下：</p> <p>值 描述</p> <p>0 当处理器被调试器停止时，RTC 停止计数。</p> <p>1 当处理器被调试器停止时，RTC 继续计数。</p> <p>置位后，RTCEN 位可防止所有操作模式下的定时器停止运行，即使 TnSTALL 已置位。</p>
3:2	TAEVENT	R/W	0x0	<p>GPTM Timer A 事件模式 TAEVENT 值定义如下：</p> <p>值 描述</p> <p>0x0 上升沿</p> <p>0x1 下降沿</p> <p>0x2 保留</p> <p>0x3 双边沿</p> <p>注意： 如果启用了 PWM 输出反相，将翻转边沿检测中断行为。因此，如果上升沿中断触发已置位，且 PWM 反相生成上升沿，将不存在有效的事件触发中断。相反，中断在 PWM 信号的下降沿生成。</p>
1	TASTALL	R/W	0	<p>GPTM Timer A 停止启用 TASTALL 值定义如下：</p> <p>值 描述</p> <p>0 当处理器被调试器停止时，Timer A 继续计数。</p> <p>1 当处理器被调试器停止时，Timer A 停止计数。</p> <p>如果处理器正常运行，忽略 TASTALL 位。</p>
0	TAEN	R/W	0	<p>GPTM Timer A 启用 TAEN 值定义如下：</p> <p>值 描述</p> <p>0 Timer A 已禁用。</p> <p>1 根据 GPTMCFG 寄存器，启用 Timer A 并开始计数或启用捕获逻辑。</p>

## 寄存器 5: GPTM 同步寄存器 ( GPTMSYNC ) , 偏移量 0x010

注意: 仅在 GPTM 模块 0 上执行该寄存器。

该寄存器允许软件同步多个定时器。

### GPTM 同步寄存器 (GPTMSYNC)

16/32 位 Timer 0 基址: 0x4003.0000

16/32 位 Timer 1 基址: 0x4003.1000

16/32 位 Timer 2 基址: 0x4003.2000

16/32 位 Timer 3 基址: 0x4003.3000

16/32 位 Timer 4 基址: 0x4003.4000

16/32 位 Timer 5 基址: 0x4003.5000

32/64 位宽 Timer 0 基址: 0x4003.6000

32/64 位宽 Timer 1 基址: 0x4003.7000

32/64 位宽 Timer 2 基址: 0x4004.C000

32/64 位宽 Timer 3 基址: 0x4004.D000

32/64 位宽 Timer 4 基址: 0x4004.E000

32/64 位宽 Timer 5 基址: 0x4004.F000

偏移量 0x010

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								SYNCWT5		SYNCWT4		SYNCWT3		SYNCWT2	
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SYNCWT1		SYNCWT0		SYNCT5		SYNCT4		SYNCT3		SYNCT2		SYNCT1		SYNCT0	
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:24	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
23:22	SYNCWT5	R/W	0x0	同步 GPTM 32/64 位 Timer 5 SYNCWT5 值定义如下：  值 描述 0x0 GPTM 32/64 位 Timer 5 未受影响。 0x1 触发 GPTM 32/64 位 Timer 5 的 Timer A 超时事件。 0x2 触发 GPTM 32/64 位 Timer 5 的 Timer B 超时事件。 0x3 触发 GPTM 32/64 位 Timer 5 的 Timer A 和 Timer B 超时事件。
21:20	SYNCWT4	R/W	0x0	同步 GPTM 32/64 位 Timer 4 SYNCWT4 值定义如下：  值 描述 0x0 GPTM 32/64 位 Timer 4 未受影响。 0x1 触发 GPTM 32/64 位 Timer 4 的 Timer A 超时事件。 0x2 触发 GPTM 32/64 位 Timer 4 的 Timer B 超时事件。 0x3 触发 GPTM 32/64 位 Timer 4 的 Timer A 和 Timer B 超时事件。

位/域	名称	类型	复位	描述
19:18	SYNCWT3	R/W	0x0	<p>同步 GPTM 32/64 位 Timer 3</p> <p>SYNCWT3 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 32/64 位 Timer 3 未受影响。</p> <p>0x1 触发 GPTM 32/64 位 Timer 3 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 32/64 位 Timer 3 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 32/64 位 Timer 3 的 Timer A 和 Timer B 超时事件。</p>
17:16	SYNCWT2	R/W	0x0	<p>同步 GPTM 32/64 位 Timer 2</p> <p>SYNCWT2 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 32/64 位 Timer 2 未受影响。</p> <p>0x1 触发 GPTM 32/64 位 Timer 2 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 32/64 位 Timer 2 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 32/64 位 Timer 2 的 Timer A 和 Timer B 超时事件。</p>
15:14	SYNCWT1	R/W	0x0	<p>同步 GPTM 32/64 位 Timer 1</p> <p>SYNCWT1 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 32/64 位 Timer 1 未受影响。</p> <p>0x1 触发 GPTM 32/64 位 Timer 1 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 32/64 位 Timer 1 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 32/64 位 Timer 1 的 Timer A 和 Timer B 超时事件。</p>
13:12	SYNCWT0	R/W	0x0	<p>同步 GPTM 32/64 位 Timer 0</p> <p>SYNCWT0 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 32/64 位 Timer 0 未受影响。</p> <p>0x1 触发 GPTM 32/64 位 Timer 0 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 32/64 位 Timer 0 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 32/64 位 Timer 0 的 Timer A 和 Timer B 超时事件。</p>
11:10	SYNCT5	R/W	0x0	<p>同步 GPTM 16/32 位 Timer 5</p> <p>SYNCT5 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 16/32 位 Timer 5 未受影响。</p> <p>0x1 触发 GPTM 16/32 位 Timer 5 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 16/32 位 Timer 5 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 16/32 位 Timer 5 的 Timer A 和 Timer B 超时事件。</p>

位/域	名称	类型	复位	描述
9:8	SYNCT4	R/W	0x0	<p>同步 GPTM 16/32 位 Timer 4</p> <p>SYNCT4 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 16/32 位 Timer 4 未受影响。</p> <p>0x1 触发 GPTM 16/32 位 Timer 4 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 16/32 位 Timer 4 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 16/32 位 Timer 4 的 Timer A 和 Timer B 超时事件。</p>
7:6	SYNCT3	R/W	0x0	<p>同步 GPTM 16/32 位 Timer 3</p> <p>SYNCT3 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 16/32 位 Timer 3 未受影响。</p> <p>0x1 触发 GPTM 16/32 位 Timer 3 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 16/32 位 Timer 3 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 16/32 位 Timer 3 的 Timer A 和 Timer B 超时事件。</p>
5:4	SYNCT2	R/W	0x0	<p>同步 GPTM 16/32 位 Timer 2</p> <p>SYNCT2 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 16/32 位 Timer 2 未受影响。</p> <p>0x1 触发 GPTM 16/32 位 Timer 2 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 16/32 位 Timer 2 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 16/32 位 Timer 2 的 Timer A 和 Timer B 超时事件。</p>
3:2	SYNCT1	R/W	0x0	<p>同步 GPTM 16/32 位 Timer 1</p> <p>SYNCT1 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 16/32 位 Timer 1 未受影响。</p> <p>0x1 触发 GPTM 16/32 位 Timer 1 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 16/32 位 Timer 1 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 16/32 位 Timer 1 的 Timer A 和 Timer B 超时事件。</p>
1:0	SYNCT0	R/W	0x0	<p>同步 GPTM 16/32 位 Timer 0</p> <p>SYNCT0 值定义如下：</p> <p>值 描述</p> <p>0x0 GPTM 16/32 位 Timer 0 未受影响。</p> <p>0x1 触发 GPTM 16/32 位 Timer 0 的 Timer A 超时事件。</p> <p>0x2 触发 GPTM 16/32 位 Timer 0 的 Timer B 超时事件。</p> <p>0x3 触发 GPTM 16/32 位 Timer 0 的 Timer A 和 Timer B 超时事件。</p>

## 寄存器 6: GPTM 中断屏蔽寄存器 ( GPTMIMR ) , 偏移量 0x018

该寄存器允许软件启用/禁用 GPTM 控制器级中断。置位时启用相应中断，清零时禁用相应中断。

## GPTM 中断屏蔽寄存器 (GPTMIMR)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x018  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留															WUEIM	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留				TBMIM	CBEIM	CBMIM	TBTOIM	保留				TAMIM	RTCIM	CAEIM	CAMIM	TATOIM
类型	RO	RO	RO	RO	R/W	R/W	R/W	R/W	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:17	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
16	WUEIM	R/W	0	32/64 位宽 GPTM 写操作更新错误中断屏蔽 WUEIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。
15:12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读取-修改-写入操作过程中应该保持不变。
11	TBMIM	R/W	0	GPTM Timer B 匹配中断屏蔽 TBMIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。
10	CBEIM	R/W	0	GPTM Timer B 捕获模式事件中断屏蔽 CBEIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。



位/域	名称	类型	复位	描述
9	CBMIM	R/W	0	GPTM Timer B 捕获模式匹配中断屏蔽 CBMIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。
8	TBTOIM	R/W	0	GPTM Timer B 超时中断屏蔽 TBTOIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。
7:5	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
4	TAMIM	R/W	0	GPTM Timer A 匹配中断屏蔽 TAMIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。
3	RTCIM	R/W	0	GPTM RTC 中断屏蔽 RTCIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。
2	CAEIM	R/W	0	GPTM Timer A 捕获模式事件中断屏蔽 CAEIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。
1	CAMIM	R/W	0	GPTM Timer A 捕获模式匹配中断屏蔽 CAMIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。

位/域	名称	类型	复位	描述
0	TATOIM	R/W	0	GPTM Timer A 超时中断屏蔽 TATOIM 值定义如下：  值 描述 0 中断已禁用。 1 中断已启用。

## 寄存器 7: GPTM 原始中断状态寄存器 ( GPTMRIS ) , 偏移量 0x01C

该寄存器显示了 GPTM 内部中断信号的状态。不管是否在 GPTMIMR 寄存器中将中断屏蔽, 这些位都会置位。向 GPTMICR 的某一位写 1 可将 GPTMRIS 寄存器的对应位清零。

### GPTM 原始中断状态寄存器 (GPTMRIS)

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x01C  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留															WUERIS	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留				TBMRIS	CBERIS	CBMRIS	TBTORIS	保留				TAMRIS	RTCRIS	CAERIS	CAMRIS	TATORIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:17	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
16	WUERIS	R/W	0	32/64 位宽 GPTM 写操作更新错误原始中断状态  值 描述 1 Timer A 寄存器或 Timer B 寄存器被连续写入两次, 或 Timer A 寄存器在写入相应的 Timer B 寄存器前被写入。 0 无错误。
15:12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读取-修改-写入操作过程中应该保持不变。
11	TBMRIS	RO	0	GPTM Timer B 匹配原始中断  值 描述 1 配置为单次触发或周期模式时, GPTMTBMR 寄存器的 TBMIE 位被置位, 并达到 GPTMTBMATCHR 和 ( 可选 ) GPTMTBPMR 寄存器中的匹配值。 0 尚未达到匹配值。  通过写 1 到 GPTMICR 寄存器的 TBMCINT 位可将该位清零。

位/域	名称	类型	复位	描述
10	CBERIS	RO	0	<p>GPTM Timer B 捕获模式事件原始中断</p> <p>值 描述</p> <p>1 Timer B 已发生捕获模式事件。子定时器配置为输入边沿计时模式或配置为 PWM 模式 (且通过将 GPTMTBMR 的 TBPWMIE 位置位而启用 PWM 中断) 时, 该中断有效。</p> <p>0 Timer B 尚未发生捕获模式事件。</p> <p>通过写 1 到 GPTMICR 寄存器的 CBECINT 位可将该位清零。</p>
9	CBMRIS	RO	0	<p>GPTM Timer B 捕获模式匹配原始中断</p> <p>值 描述</p> <p>1 Timer B 已发生捕获模式匹配。配置为输入边沿计时模式时, 如果 GPTMTBR 和 GPTMTBPR 中的值与 GPTMTBMATCHR 和 GPTMTBPMR 中的值相同, 则该中断有效。</p> <p>0 Timer B 尚未发生捕获模式匹配。</p> <p>通过写 1 到 GPTMICR 寄存器的 CBMCINT 位可将该位清零。</p>
8	TBTORIS	RO	0	<p>GPTM Timer B 超时原始中断</p> <p>值 描述</p> <p>1 Timer B 已超时。单次触发或周期模式定时器达到自身的计数限值 (0 或载入 GPTMTBILR 的值, 由计数方向决定) 时, 该中断有效。</p> <p>0 Timer B 未超时。</p> <p>通过写 1 到 GPTMICR 寄存器的 TBTOCINT 位可将该位清零。</p>
7:5	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。</p>
4	TAMRIS	RO	0	<p>GPTM Timer A 匹配原始中断</p> <p>值 描述</p> <p>1 配置为单次触发或周期模式时, GPTMTAMR 寄存器的 TAMIE 位被置位, 并达到 GPTMTAMATCHR 和 GPTMTAPMR 寄存器 (可选) 中的匹配值。</p> <p>0 尚未达到匹配值。</p> <p>通过写 1 到 GPTMICR 寄存器的 TAMCINT 位可将该位清零。</p>
3	RTCRIS	RO	0	<p>GPTM RTC 原始中断</p> <p>值 描述</p> <p>1 RTC 事件已发生。</p> <p>0 RTC 事件未发生。</p> <p>通过写 1 到 GPTMICR 寄存器的 RTCCINT 位可将该位清零。</p>

位/域	名称	类型	复位	描述
2	CAERIS	RO	0	<p>GPTM Timer A 捕获模式事件原始中断</p> <p>值 描述</p> <p>1 Timer A 已发生捕获模式事件。子定时器配置为输入边沿计时模式或配置为 PWM 模式 (且通过将 GPTMTAMR 的 TAPWMIE 位置位而启用 PWM 中断) 时, 该中断有效。</p> <p>0 Timer A 尚未发生捕获模式事件。</p> <p>该位通过写 1 到 GPTMICR 寄存器的 CAECINT 位来清零。</p>
1	CAMRIS	RO	0	<p>GPTM Timer A 捕获模式匹配原始中断</p> <p>值 描述</p> <p>1 Timer A 已发生捕获模式匹配。配置为输入边沿计时模式时, 如果 GPTMTAR 和 GPTMTAPR 中的值与 GPTMTAMATCHR 和 GPTMTAPMR 中的值相同, 则该中断有效。</p> <p>0 Timer A 尚未发生捕获模式匹配。</p> <p>该位通过写 1 到 GPTMICR 寄存器的 CAMCINT 位来清零。</p>
0	TATORIS	RO	0	<p>GPTM Timer A 超时原始中断</p> <p>值 描述</p> <p>1 Timer A 已超时。单次触发或周期模式定时器达到自身的计数限值 (0 或载入 GPTMTAILR 的值, 由计数方向决定) 时, 该中断有效。</p> <p>0 Timer A 未超时。</p> <p>通过写 1 到 GPTMICR 寄存器的 TATOCINT 位可将该位清零。</p>

## 寄存器 8: GPTM 屏蔽的中断状态寄存器 ( GPTMMIS ) , 偏移量 0x020

该寄存器显示了 GPTM 控制器级中断的状态。如果没有在 GPTMIMR 寄存器中将中断屏蔽，并在此时出现一个使中断有效的事件，那么该寄存器中相应的位将会置位。通过向 GPTMICR 的对应位写 1 可将所有位清零。

### GPTM 屏蔽的中断状态寄存器 (GPTMMIS)

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x020  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留															WUEMIS	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留				TBMMIS	CBEMIS	CBMMIS	TBTOMIS	保留				TAMMIS	RTCMIS	CAEMIS	CAMMIS	TATOMIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:17	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
16	WUEMIS	RO	0	32/64 位宽 GPTM 写操作更新错误屏蔽中断状态  值 描述 1 未屏蔽的写操作更新错误已发生。 0 未屏蔽的写操作更新错误未发生。
15:12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
11	TBMMIS	RO	0	GPTM Timer B 匹配屏蔽中断  值 描述 1 未屏蔽的 Timer B 模式匹配中断已发生。 0 匹配值没有到达或中断已经被屏蔽。  通过写 1 到 GPTMICR 寄存器的 TBMCINT 位可将该位清零。

位/域	名称	类型	复位	描述
10	CBEMIS	RO	0	<p>GPTM Timer B 捕获模式事件屏蔽中断</p> <p>值 描述</p> <p>1 未屏蔽的捕获 B 事件中断已发生。</p> <p>0 捕获 B 事件没有发生或中断已经被屏蔽。</p> <p>通过写 1 到 GPTMICR 寄存器的 CBECINT 位可将该位清零。</p>
9	CBMMIS	RO	0	<p>GPTM Timer B 捕获模式匹配屏蔽中断</p> <p>值 描述</p> <p>1 未屏蔽的捕获 B 匹配中断已发生。</p> <p>0 捕获 B 模式匹配值没有到达或中断已经被屏蔽。</p> <p>通过写 1 到 GPTMICR 寄存器的 CBMCINT 位可将该位清零。</p>
8	TBTOMIS	RO	0	<p>GPTM Timer B 超时屏蔽的中断</p> <p>值 描述</p> <p>1 未屏蔽的 Timer B 超时中断已发生。</p> <p>0 Timer B 没有超时或中断已经被屏蔽。</p> <p>通过写 1 到 GPTMICR 寄存器的 TBTOCINT 位可将该位清零。</p>
7:5	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
4	TAMMIS	RO	0	<p>GPTM Timer A 匹配屏蔽中断</p> <p>值 描述</p> <p>1 未屏蔽的 Timer A 模式匹配中断已发生。</p> <p>0 Timer A 模式匹配值没有到达或中断已经被屏蔽。</p> <p>通过写 1 到 GPTMICR 寄存器的 TAMCINT 位可将该位清零。</p>
3	RTCMIS	RO	0	<p>GPTM RTC 屏蔽的中断</p> <p>值 描述</p> <p>1 未屏蔽的 RTC 事件中断已发生。</p> <p>0 RTC 事件未产生中断，或此中断被屏蔽。</p> <p>通过写 1 到 GPTMICR 寄存器的 RTCCINT 位可将该位清零。</p>

位/域	名称	类型	复位	描述
2	CAEMIS	RO	0	<p>GPTM Timer A 捕获模式事件屏蔽中断</p> <p>值 描述</p> <p>1 未屏蔽的捕获 A 事件中断已发生。</p> <p>0 捕获 B 事件没有发生或中断已经被屏蔽。</p> <p>该位通过写 1 到 GPTMICR 寄存器的 CAECINT 位来清零。</p>
1	CAMMIS	RO	0	<p>GPTM Timer A 捕获模式匹配屏蔽中断</p> <p>值 描述</p> <p>1 未屏蔽的捕获 A 匹配中断已发生。</p> <p>0 捕获 A 模式匹配值没有到达或中断已经被屏蔽。</p> <p>该位通过写 1 到 GPTMICR 寄存器的 CAMCINT 位来清零。</p>
0	TATOMIS	RO	0	<p>GPTM Timer A 超时屏蔽的中断</p> <p>值 描述</p> <p>1 未屏蔽的 Timer A 超时中断已发生。</p> <p>0 Timer A 没有超时或中断已经被屏蔽。</p> <p>通过写 1 到 GPTMICR 寄存器的 TATOCINT 位可将该位清零。</p>



## 寄存器 9: GPTM 中断清除寄存器 ( GPTMICR ) , 偏移量 0x024

该寄存器用来将 GPTMRIS 和 GPTMMIS 寄存器中的状态位清零。在该寄存器写入 1 来清除 GPTMRIS 和 GPTMMIS 相应的位。

### GPTM 中断清除寄存器 (GPTMICR)

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x024  
 类型 W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留															WUECINT	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留				TBMCINT	CBECINT	CBMCINT	TBTOCINT	保留				TAMCINT	RTCCINT	CAECINT	CAMCINT	TATOCINT
类型	RO	RO	RO	RO	W1C	W1C	W1C	W1C	RO	RO	RO	W1C	W1C	W1C	W1C	W1C	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:17	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
16	WUECINT	R/W	0	32/64 位宽 GPTM 写操作更新错误中断清除 向本标志位写 1, 即可将 GPTMRIS 寄存器的 WUERIS 位以及 GPTMMIS 寄存器的 WUEMIS 位清零。
15:12	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
11	TBMCINT	W1C	0	GPTM Timer B 匹配中断清零 向本标志位写 1, 即可将 GPTMRIS 寄存器的 TBMRIS 位以及 GPTMMIS 寄存器的 TBMMIS 位清零。
10	CBECINT	W1C	0	GPTM Timer B 捕获模式事件中断清零 向本标志位写 1, 即可将 GPTMRIS 寄存器的 CBERIS 位以及 GPTMMIS 寄存器的 CBEMIS 位清零。
9	CBMCINT	W1C	0	GPTM Timer B 捕获模式匹配中断清零 向本标志位写 1, 即可将 GPTMRIS 寄存器的 CBMRIS 位以及 GPTMMIS 寄存器的 CBMMIS 位清零。
8	TBTOCINT	W1C	0	GPTM Timer B 超时中断清除 向本标志位写 1, 即可将 GPTMRIS 寄存器的 TBTORIS 位以及 GPTMMIS 寄存器的 TBTOMIS 位清零。
7:5	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。

---

位/域	名称	类型	复位	描述
4	TAMCINT	W1C	0	GPTM Timer A 匹配中断清零 向本标志位写 1，即可将 GPTMRIS 寄存器的 TAMRIS 位以及 GPTMMIS 寄存器的 TAMMIS 位清零。
3	RTCCINT	W1C	0	GPTM 实时时钟中断清除 向本标志位写 1，即可将 GPTMRIS 寄存器的 RTCRIS 位以及 GPTMMIS 寄存器的 RTCMIS 位清零。
2	CAECINT	W1C	0	GPTM Timer A 捕获模式事件中断清零 向本标志位写 1，即可将 GPTMRIS 寄存器的 CAERIS 位以及 GPTMMIS 寄存器的 CAEMIS 位清零。
1	CAMCINT	W1C	0	GPTM Timer A 捕获模式匹配中断清零 向本标志位写 1，即可将 GPTMRIS 寄存器的 CAMRIS 位以及 GPTMMIS 寄存器的 CAMMIS 位清零。
0	TATOCINT	W1C	0	GPTM Timer A 超时原始中断 向本标志位写 1，即可将 GPTMRIS 寄存器的 TATORIS 位以及 GPTMMIS 寄存器的 TATOMIS 位清零。

**寄存器 10: GPTM Timer A 间隔加载寄存器 ( GPTMTAILR ) , 偏移量 0x028**

当定时器递减计数时, 该寄存器用来将起始计数值装入定时器。当定时器递增计数时, 该寄存器为超时事件设置上边沿。

当 16/32 位 GPTM 配置为其中一种 32 位模式时, GPTMTAILR 作为 32 位寄存器使用 ( 高 16 位对应于 GPTM Timer B 间隔加载 (GPTMTBILR) 寄存器的值 )。当配置为 16 位时, 高 16 位读出来是 0, 并且不会影响 GPTMTBILR 寄存器的状态。

当 32/64 位宽 GPTM 配置为其中一种 64 位模式时, GPTMTAILR 包含 64 位计数的 31:0 位, 并且 GPTM Timer B 间隔加载 (GPTMTBILR) 寄存器包含 63:32 位。

**GPTM Timer A 间隔加载寄存器 (GPTMTAILR)**

16/32 位Timer 0 基址: 0x4003.0000

16/32 位Timer 1 基址: 0x4003.1000

16/32 位Timer 2 基址: 0x4003.2000

16/32 位Timer 3 基址: 0x4003.3000

16/32 位Timer 4 基址: 0x4003.4000

16/32 位Timer 5 基址: 0x4003.5000

32/64 位宽 Timer 0 基址: 0x4003.6000

32/64 位宽 Timer 1 基址: 0x4003.7000

32/64 位宽 Timer 2 基址: 0x4004.C000

32/64 位宽 Timer 3 基址: 0x4004.D000

32/64 位宽 Timer 4 基址: 0x4004.E000

32/64 位宽 Timer 5 基址: 0x4004.F000

偏移量 0x028

类型 R/W, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAILR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAILR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	TAILR	R/W	0xFFFF.FFFF	GPTM Timer A 间隔加载寄存器 写入该域将为 Timer A 加载计数器, 读取该域将返回 GPTMTAILR 寄存器的当前值。

**寄存器 11: GPTM Timer B 间隔加载寄存器 ( GPTMTBILR ) , 偏移量 0x02C**

当定时器递减计数时, 该寄存器用来将起始计数值装入定时器。当定时器递增计数时, 该寄存器为超时事件设置上边沿。

当 16/32 位 GPTM 配置为其中一种 32 位模式时, 该寄存器中 15:0 位的内容被加载到 GPTMTAILR 寄存器的高 16 位。读取该寄存器将返回 Timer B 的当前值并且忽略写操作。在 16 位模式中, 15:0 位用于加载值。在两种情况下都保留 31:16 位。

当 32/64 位宽 GPTM 配置为其中一种 64 位模式时, GPTMTAILR 包含 64 位计数的 31:0 位, 并且 GPTMTBILR 寄存器包含 63:32 位。

**GPTM Timer B 间隔加载寄存器 (GPTMTBILR)**

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x02C  
 类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TBILR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBILR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	1	1	0	0	1	0	0	1	1	0	0	1	0	0

位/域	名称	类型	复位	描述
31:0	TBILR	R/W	0x0000.FFFF (适用于 16/32 位) 0xFFFF.FFFF (适用于 32/64 位)	GPTM Timer B 间隔加载寄存器 写入该域将为 Timer B 加载计数器, 读取该域将返回 GPTMTBILR 寄存器的当前值。 当 16/32 位 GPTM 处于 32 位模式中, 写操作被忽略, 读操作返回 GPTMTBILR 的当前值。

**寄存器 12: GPTM Timer A 匹配寄存器 ( GPTMTAMATCHR ) , 偏移量 0x030**

该寄存器加载一个匹配值。在单次触发或周期模式中，如果定时器的值等于该寄存器的值则引发一个中断。

在边沿计数模式，该寄存器和 GPTMTAILR 寄存器一起记录共发生了多少个边沿事件。总边沿事件数等于 GPTMTAILR 的值减去该值。

在 PWM 模式中，该寄存器与 GPTMTAILR 寄存器一起决定输出信号的占空比。

当 16/32 位 GPTM 配置为其中一种 32 位模式时，GPTMTAMATCHR 作为 32 位寄存器使用（高 16 位对应于 GPTM Timer B 匹配 (GPTMTBMATCHR) 寄存器的值）。当配置为 16 位时，高 16 位读出来是 0，并且不会影响 GPTMTBMATCHR 寄存器的状态。

当 32/64 位宽 GPTM 配置为其中一种 64 位模式时，GPTMTAMATCHR 包含 64 位匹配值的 31:0 位，并且 GPTM Timer B 匹配 (GPTMTBMATCHR) 寄存器包含 63:32 位。

**GPTM Timer A 匹配寄存器 (GPTMTAMATCHR)**

16/32 位Timer 0 基址: 0x4003.0000

16/32 位Timer 1 基址: 0x4003.1000

16/32 位Timer 2 基址: 0x4003.2000

16/32 位Timer 3 基址: 0x4003.3000

16/32 位Timer 4 基址: 0x4003.4000

16/32 位Timer 5 基址: 0x4003.5000

32/64 位宽 Timer 0 基址: 0x4003.6000

32/64 位宽 Timer 1 基址: 0x4003.7000

32/64 位宽 Timer 2 基址: 0x4004.C000

32/64 位宽 Timer 3 基址: 0x4004.D000

32/64 位宽 Timer 4 基址: 0x4004.E000

32/64 位宽 Timer 5 基址: 0x4004.F000

偏移量 0x030

类型 R/W, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAMR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAMR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	TAMR	R/W	0xFFFF.FFFF	GPTM Timer A 匹配寄存器 该值与 GPTMTAR 寄存器进行比较来确定匹配事件的发生。

**寄存器 13: GPTM Timer B 匹配寄存器 ( GPTMTBMATCHR ) , 偏移量 0x034**

该寄存器加载一个匹配值。在单次触发或周期模式中，如果定时器的值等于该寄存器的值则引发一个中断。

在边沿计数模式，该寄存器和 GPTMTBILR 寄存器一起记录共发生了多少个边沿事件。总边沿事件数等于 GPTMTBILR 的值减去该值。

在 PWM 模式中，该寄存器与 GPTMTBILR 寄存器一起决定输出信号的占空比。

当 16/32 位 GPTM 配置为其中一种 32 位模式时，该寄存器中 15:0 位的内容被加载到 GPTMTAMATCHR 寄存器的高 16 位。读取该寄存器将返回 Timer B 的匹配值并且忽略写操作。在 16 位模式中，15:0 位用于匹配值。在两种情况下都保留 31:16 位。

当 32/64 位宽 GPTM 配置为其中一种 64 位模式时，GPTMTAMATCHR 包含 64 位匹配值的 31:0 位，并且 GPTMTBMATCHR 寄存器包含 63:32 位。

**GPTM Timer B 匹配寄存器 (GPTMTBMATCHR)**

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x034  
 类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TBMR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBMR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	1	1	0	0	1	0	0	1	1	0	0	1	0	0

位/域	名称	类型	复位	描述
31:0	TBMR	R/W	0x0000.FFFF (适用于 16/32 位) 0xFFFF.FFFF (适用于 32/64 位)	GPTM Timer B 匹配寄存器 该值与 GPTMTBR 寄存器进行比较来确定匹配事件的发生。

## 寄存器 14: GPTM Timer A 预分频寄存器 ( GPTMTAPR ) , 偏移量 0x038

该寄存器允许软件扩展独立使用时定时器的范围。在单次触发或周期递减计数模式中，该寄存器用作定时器计数器的真预分频器。用作真预分频器时，在 GPTMTAR 和 GPTMTAV 寄存器的值递增前，预分频器计数递减到 0。在所有其他独立/分离模式中，该寄存器是定时器计数器上限范围的一个线性扩展，在 16/32 位 GPTM 的 16 位模式下包含位 23:16，在 32/64 位宽 GPTM 的 32 位模式下包含位 47:32。

### GPTM Timer A 预分频寄存器 (GPTMTAPR)

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x038  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAPSRH								TAPSR							
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
15:8	TAPSRH	R/W	0x00	GPTM Timer A 预分频高字节 寄存器通过写操作载入该值。执行读操作将返回寄存器的当前值。 对于 16/32 位 GPTM，保留该域。对于 32/64 位宽 GPTM，该域包含 16 位预分频器的高 8 位。 详细信息和实例请参考表 11-5 在 659 页。
7:0	TAPSR	R/W	0x00	GPTM Timer A 预分频 寄存器通过写操作载入该值。执行读操作将返回寄存器的当前值。 对于 16/32 位 GPTM，该域包含全部 8 位预分频器。对于 32/64 位宽 GPTM，该域包含 16 位预分频器的低 8 位。 详细信息和实例请参考表 11-5 在 659 页。

## 寄存器 15: GPTM Timer B 预分频寄存器 ( GPTMTBPR ) , 偏移量 0x03C

该寄存器允许软件扩展独立使用时定时器的范围。在单次触发或周期递减计数模式中，该寄存器用作定时器计数器的真预分频器。用作真预分频器时，预分频器在 GPTMTBR 和 GPTMTBV 寄存器的值递增前，计数递减到 0。在所有其他独立/分离模式中，该寄存器是定时器计数器上限范围的一个线性扩展，在 16/32 位 GPTM 的 16 位模式下包含位 23:16，在 32/64 位宽 GPTM 的 32 位模式下包含位 47:32。

### GPTM Timer B 预分频寄存器 (GPTMTBPR)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x03C  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBPSRH								TBPSR							
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
15:8	TBPSRH	R/W	0x00	GPTM Timer B 预分频高字节 寄存器通过写操作载入该值。执行读操作将返回寄存器的当前值。 对于 16/32 位 GPTM，保留该域。对于 32/64 位宽 GPTM，该域包含 16 位预分频器的高 8 位。 详细信息和实例请参考表 11-5 在 659 页。
7:0	TBPSR	R/W	0x00	GPTM Timer B 预分频寄存器 寄存器通过写操作载入该值。执行读操作将返回寄存器的当前值。 对于 16/32 位 GPTM，该域包含全部 8 位预分频器。对于 32/64 位宽 GPTM，该域包含 16 位预分频器的低 8 位。 详细信息和实例请参考表 11-5 在 659 页。



## 寄存器 16: GPTM TimerA 预分频匹配寄存器 ( GPTMTAPMR ) , 偏移量 0x040

该寄存器允许软件扩展在单独使用定时器时 GPTMTAMATCHR 的范围。该寄存器保持 16/32 位 GPTM 的 16 位模式中的 23:16 位以及 32/64 位宽 GPTM 的 32 位模式中的 47:32 位。

### GPTM TimerA 预分频匹配寄存器 (GPTMTAPMR)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x040  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAPSMRH								TAPSMR							
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:8	TAPSMRH	R/W	0x00	GPTM Timer A 预分频匹配高字节 这些值与 GPTMTAMATCHR 一起使用, 以便在使用预分频器的情况下检测定时器匹配事件。 对于 16/32 位 GPTM, 保留该域。对于 32/64 位宽 GPTM, 该域包含 16 位预分频匹配值的高 8 位。
7:0	TAPSMR	R/W	0x00	GPTM Timer A 预分频匹配 这些值与 GPTMTAMATCHR 一起使用, 以便在使用预分频器的情况下检测定时器匹配事件。 对于 16/32 位 GPTM, 该域包含全部 8 位预分频器匹配值。对于 32/64 位宽 GPTM, 该域包含 16 位预分频器匹配值的低 8 位。

## 寄存器 17: GPTM TimerB 预分频匹配寄存器 ( GPTMTBPMR ) , 偏移量 0x044

该寄存器允许软件扩展在单独使用定时器时 GPTMTBMATCHR 的范围。该寄存器保持 16/32 位 GPTM 的 16 位模式中的 23:16 位以及 32/64 位宽 GPTM 的 32 位模式中的 47:32 位。

### GPTM TimerB 预分频匹配寄存器 (GPTMTBPMR)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x044  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBPSMRH								TBPSMR							
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:8	TBPSMRH	R/W	0x00	GPTM Timer B 预分频匹配高字节 这些值与 GPTMTBMATCHR 一起使用，以便在使用预分频器的情况下检测定时器匹配事件。 对于 16/32 位 GPTM，保留该域。对于 32/64 位宽 GPTM，该域包含 16 位预分频匹配值的高 8 位。
7:0	TBPSMR	R/W	0x00	GPTM Timer B 预分频匹配寄存器 这些值与 GPTMTBMATCHR 一起使用，以便在使用预分频器的情况下检测定时器匹配事件。 对于 16/32 位 GPTM，该域包含全部 8 位预分频器匹配值。对于 32/64 位宽 GPTM，该域包含 16 位预分频器匹配值的低 8 位。

## 寄存器 18: GPTM Timer A 寄存器 ( GPTMTAPR ) , 偏移量 0x048

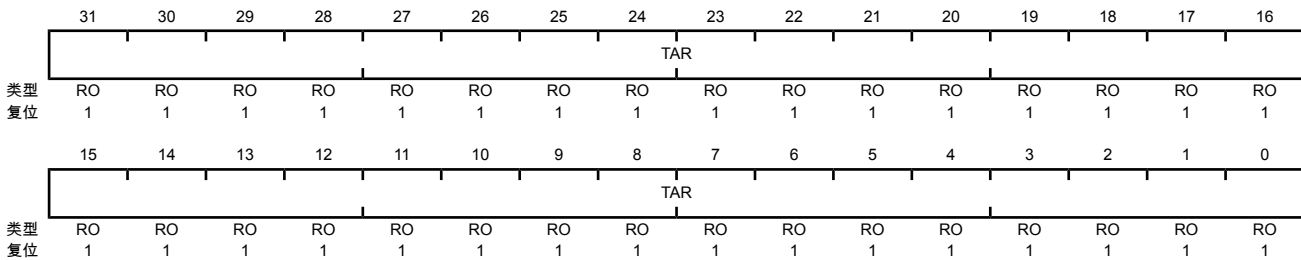
在任何情况下该寄存器显示当前 Timer A 计数器的值，输入边沿计数模式的情况除外。在输入边沿计数模式中，该寄存器记录了已经发生的边沿的个数。在输入边沿计时模式中，该寄存器包含上一次边沿事件发生的时间。

当 16/32 位 GPTM 配置为其中一种 32 位模式时，GPTMTAR 作为 32 位寄存器使用（高 16 位对应于 GPTM Timer B (GPTMTBR) 寄存器的值）。在 16 位输入边沿计数、输入边沿计时和 PWM 模式中，15:0 位包含计数器的值，23:16 位包含预分频器高 8 位的值。31:24 位的回读值始终为 0。要在 16 位单次触发和周期模式中读取预分频器的值，读取 GPTMTAV 寄存器中的 [23:16] 位。

当 32/64 位宽 GPTM 配置为其中一种 64 位模式时，GPTMTAR 包含 64 位定时器值的 31:0 位并且 GPTM Timer B (GPTMTBR) 寄存器包含 63:32 位。在 32 位模式中，预分频器的值存储在 GPTM Timer A 预分频快照 (GPTMTAPS) 寄存器中。

### GPTM Timer A 寄存器 (GPTMTAPR)

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x048  
 类型 RO, 复位 0xFFFF.FFFF



位/域	名称	类型	复位	描述
31:0	TAR	RO	0xFFFF.FFFF	GPTM Timer A 寄存器

执行读操作将返回 GPTM Timer A Count Register 的当前值，输入边沿计数和计时模式的情况除外。在输入边沿计数模式中，该寄存器记录了已经发生的边沿的个数。在输入边沿计时模式中，该寄存器包含上一次边沿事件发生的时间。

## 寄存器 19: GPTM Timer B 寄存器 ( GPTMTBPR ) , 偏移量 0x04C

在任何情况下该寄存器显示当前 Timer B 计数器的值，输入边沿计数模式的情况除外。在输入边沿计数模式中，该寄存器记录了已经发生的边沿的个数。在输入边沿计时模式中，该寄存器包含上一次边沿事件发生的时间。

当 16/32 位 GPTM 配置为其中一种 32 位模式时，该寄存器中 15:0 位的内容被加载到 GPTMTAR 寄存器的高 16 位。读取该寄存器将返回 Timer B 的当前值。在 16 位模式中，15:0 位包含计数器的值，23:16 位包含输入边沿计数、输入边沿计时和 PWM 模式中预分频器高 8 位的值。31:24 位的回读值始终为 0。要在 16 位单次触发和周期模式中读取预分频器的值，应读取 GPTMTBV 寄存器中的 [23:16] 位。

当 32/64 位宽 GPTM 配置为其中一种 64 位模式时，GPTMTAR 包含 64 位定时器值的 31:0 位并且 GPTM Timer B (GPTMTBR) 寄存器包含 63:32 位。在 32 位模式中，预分频器的值存储在 GPTM Timer B 预分频快照 (GPTMTBPS) 寄存器中。

### GPTM Timer B 寄存器 (GPTMTBPR)

16/32 位 Timer 0 基址: 0x4003.0000  
 16/32 位 Timer 1 基址: 0x4003.1000  
 16/32 位 Timer 2 基址: 0x4003.2000  
 16/32 位 Timer 3 基址: 0x4003.3000  
 16/32 位 Timer 4 基址: 0x4003.4000  
 16/32 位 Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x04C  
 类型 RO, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TBR															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBR															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	1	1	0	0	1	0	0	1	1	0	0	1	0	0

位/域	名称	类型	复位	描述
31:0	TBR	RO	0x0000.FFFF (适用于 16/32 位) 0xFFFF.FFFF (适用于 32/64 位)	GPTM Timer B 寄存器 执行读操作将返回 GPTM Timer B Count Register 的当前值，输入边沿计数和计时模式的情况除外。在输入边沿计数模式中，该寄存器记录了已经发生的边沿的个数。在输入边沿计时模式中，该寄存器包含上一次边沿事件发生的时间。

**寄存器 20: GPTM Timer A 值寄存器 ( GPTMTAV ) , 偏移量 0x050**

在所有模式下，读取该寄存器返回Timer A自由运行的值。在周期操作模式下配合使用快照特性时，软件可以根据该值来判断从发生中断到进入ISR（中断服务程序）之间所用时间。向该寄存器写入的值将会在下一个时钟周期加载到GPTMTAR寄存器里。

当16/32位GPTM配置为其中一种32位模式时，GPTMTAV作为32位寄存器使用（高16位对应于GPTM Timer B值(GPTMTBV)寄存器的值）。在16位模式中，15:0位包含计数器的值，23:16位包含预分频器当前自由运行的值，该值是输入边沿计数、输入边沿计时、PWM和单次触发或周期递增计数模式的计数高8位。在单次触发或周期递减计数模式中，存储在23:16的预分频器是真预分频器，即23:16位在15:0位的值递减之前递减计数。31:24位预分频器的回读值始终为0。

当32/64位宽GPTM配置为其中一种64位模式时，GPTMTAV包含64位定时器值的31:0位并且GPTM Timer B值(GPTMTBV)寄存器包含63:32位。在32位模式中，预分频器的当前自由运行值存储在GPTM Timer A预分频值(GPTMTAPV)寄存器.mint中

**GPTM Timer A 值寄存器 (GPTMTAV)**

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x050  
 类型 RW, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAV															
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAV															
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	TAV	RW	0xFFFF.FFFF	GPTM Timer A值寄存器 在所有模式下，读操作返回 Timer A 当前自由运行的值。向该寄存器写入的值将会在下一个时钟周期加载到 GPTMTAR 寄存器里。

## 寄存器 21: GPTM Timer B 值寄存器 ( GPTMTBV ) , 偏移量 0x054

在所有模式下, 读取该寄存器返回 Timer B 自由运行的值。软件可以根据该值来判断从发生中断到响应中断经历的时间。向该寄存器写入的值将会在下一个时钟周期加载到 GPTMTBR 寄存器里。

当 16/32 位 GPTM 配置为其中一种 32 位模式时, 该寄存器中 15:0 位的内容被加载到 GPTMTAV 寄存器的高 16 位。读取该寄存器将返回 Timer B 的当前自由运行值。在 16 位模式中, 15:0 位包含计数器的值, 23:16 位包含预分频器当前自由运行的值, 该值是输入边沿计数、输入边沿计时、PWM 和单次触发或周期递增计数模式的计数高 8 位。在单次触发或周期递减计数模式中, 存储在 23:16 的预分频器是真预分频器, 即 23:16 位在 15:0 位的值递减之前递减计数。31:24 位预分频器的回读值始终为 0。

当 32/64 位宽 GPTM 配置为其中一种 64 位模式时, GPTMTBV 包含 64 位定时器值的 63:32 位, GPTM Timer A 值 (GPTMTAV) 寄存器包含 31:0 位。在 32 位模式中, 预分频器的当前自由运行值存储在 GPTM Timer B 预分频值 (GPTMTBPV) 寄存器中。

### GPTM Timer B 值寄存器 (GPTMTBV)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x054  
 类型 RW, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TBV															
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBV															
类型	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位	0	0	1	1	0	0	1	0	0	1	1	0	0	1	0	0

位/域	名称	类型	复位	描述
31:0	TBV	RW	0x0000.FFFF (适用于 16/32 位) 0xFFFF.FFFF (适用于 32/64 位)	GPTM Timer B 值寄存器 在所有模式下, 读操作返回 Timer A 当前自由运行的值。向该寄存器写入的值将会在下一个时钟周期加载到 GPTMTAR 寄存器里。

## 寄存器 22: GPTM RTC 预分频寄存器 ( GPTMRTCPD ) , 偏移量 0x058

定时器在 RTC 模式中运行时, 该寄存器提供当前的 RTC 预分频器值。软件必须通过连续读取 GPTMTAR、GPTMTBR 和 GPTMRTCPD 寄存器执行原子访问。请参见图 11-2 在 661 页了解更多信息。

### GPTM RTC 预分频寄存器 (GPTMRTCPD)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x058  
 类型 RO, 复位 0x0000.7FFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCPD															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	RTCPD	RO	0x0000.7FFF	RTC 预分频计数器值 定时器在 RTC 模式中运行时的当前 RTC 预分频器值。该域在其他定时器模式中无意义。

## 寄存器 23: GPTM Timer A 预分频快照寄存器 ( GPTMTAPS ) , 偏移量 0x05C

该寄存器显示 16 位模式中 Timer A 预分频器的当前值。对于 32/64 位宽 GPTM , 该寄存器显示 32 位模式中 Timer A 预分频器的当前值。该寄存器不适用于 16/32 位 GPTM 模式。

### GPTM Timer A 预分频快照寄存器 (GPTMTAPS)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x05C  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSS															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	PSS	RO	0x0000	GPTM Timer A 预分频器快照 执行读操作将返回 GPTM Timer A Prescaler 当前值。



**寄存器 24: GPTM Timer B 预分频快照寄存器 ( GPTMTBPS ) , 偏移量 0x060**

该寄存器显示 16 位模式中 Timer B 预分频器的当前值。对于 32/64 位宽 GPTM，该寄存器显示 32 位模式中 Timer B 预分频器的当前值。该寄存器不适用于 16/32 位 GPTM 模式。

**GPTM Timer B 预分频快照寄存器 (GPTMTBPS)**

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x060  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSS															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	PSS	RO	0x0000	GPTM Timer A 预分频器值 执行读操作将返回 GPTM Timer A Prescaler 当前值。

## 寄存器 25: GPTM Timer A 预分频值寄存器 ( GPTMTAPV ) , 偏移量 0x064

该寄存器显示 16 位模式中 Timer A 自由运行预分频器的当前值。对于 32/64 位宽 GPTM, 该寄存器显示 32 位模式中 Timer A 预分频器的当前自由运行值。结合 GPTMTAV 寄存器, 软件可以使用该值来判断从发生中断到响应中断经历的时间。该寄存器不适用于 16/32 位 GPTM 模式。

### GPTM Timer A 预分频值寄存器 (GPTMTAPV)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x064  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSV															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	PSV	RO	0x0000	GPTM Timer A 预分频器值 执行读操作将返回 Timer A 寄存器的当前自由运行值。

**寄存器 26: GPTM Timer B 预分频值寄存器 ( GPTMTBPV ) , 偏移量 0x068**

该寄存器显示 16 位模式中 Timer B 自由运行预分频器的当前值。对于 32/64 位宽 GPTM，该寄存器显示 32 位模式中 Timer B 预分频器的当前自由运行值。结合 GPTMTBV 寄存器，软件可以使用该值来判断从发生中断到响应中断经历的时间。该寄存器不适用于 16/32 位 GPTM 模式。

**GPTM Timer B 预分频值寄存器 (GPTMTBPV)**

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0x068  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSV															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	PSV	RO	0x0000	GPTM Timer B 预分频器值 执行读操作将返回 Timer A 寄存器的当前自由运行值。

## 寄存器 27: GPTM 外设属性寄存器 ( GPTMPP ) , 偏移量 0xFC0

GPTMPP 寄存器提供关于通用定时器模块属性的信息。

## GPTM 外设属性寄存器 (GPTMPP)

16/32 位Timer 0 基址: 0x4003.0000  
 16/32 位Timer 1 基址: 0x4003.1000  
 16/32 位Timer 2 基址: 0x4003.2000  
 16/32 位Timer 3 基址: 0x4003.3000  
 16/32 位Timer 4 基址: 0x4003.4000  
 16/32 位Timer 5 基址: 0x4003.5000  
 32/64 位宽 Timer 0 基址: 0x4003.6000  
 32/64 位宽 Timer 1 基址: 0x4003.7000  
 32/64 位宽 Timer 2 基址: 0x4004.C000  
 32/64 位宽 Timer 3 基址: 0x4004.D000  
 32/64 位宽 Timer 4 基址: 0x4004.E000  
 32/64 位宽 Timer 5 基址: 0x4004.F000  
 偏移量 0xFC0  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												SIZE			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3:0	SIZE	RO	0x0	计数大小

## 值 描述

- 1 Timer A 和 Timer B 计数器为 32 位, 各自具有 16 位预分频计数器。
- 0 Timer A 和 Timer B 计数器为 16 位, 各自具有 8 位预分频计数器。

## 12 看门狗定时器 (WDT)

达到超时值时，看门狗定时器会发出不可屏蔽中断 (NMI)、常规中断或者复位信号。当系统因软件错误或外部设备未能按预期方式响应而发生故障时，看门狗定时器可令系统重新获得控制权。LM4F232H5QD 微控制器有两个看门狗定时器模块，一个模块使用系统时钟计时 (Watchdog Timer 0)，另一个模块使用 PIOSC 计时 (Watchdog Timer 1)。这两个模块是相同的，只是 WDT1 在另一个时钟域，因此需要同步器。所以，在看门狗定时器控制 (WDTCTL) 寄存器中会定义 WDT1 的某一位，用以说明 WDT1 寄存器写入操作何时完成。软件可以使用该位确保先前的访问已经完毕，然后再开始新的访问。

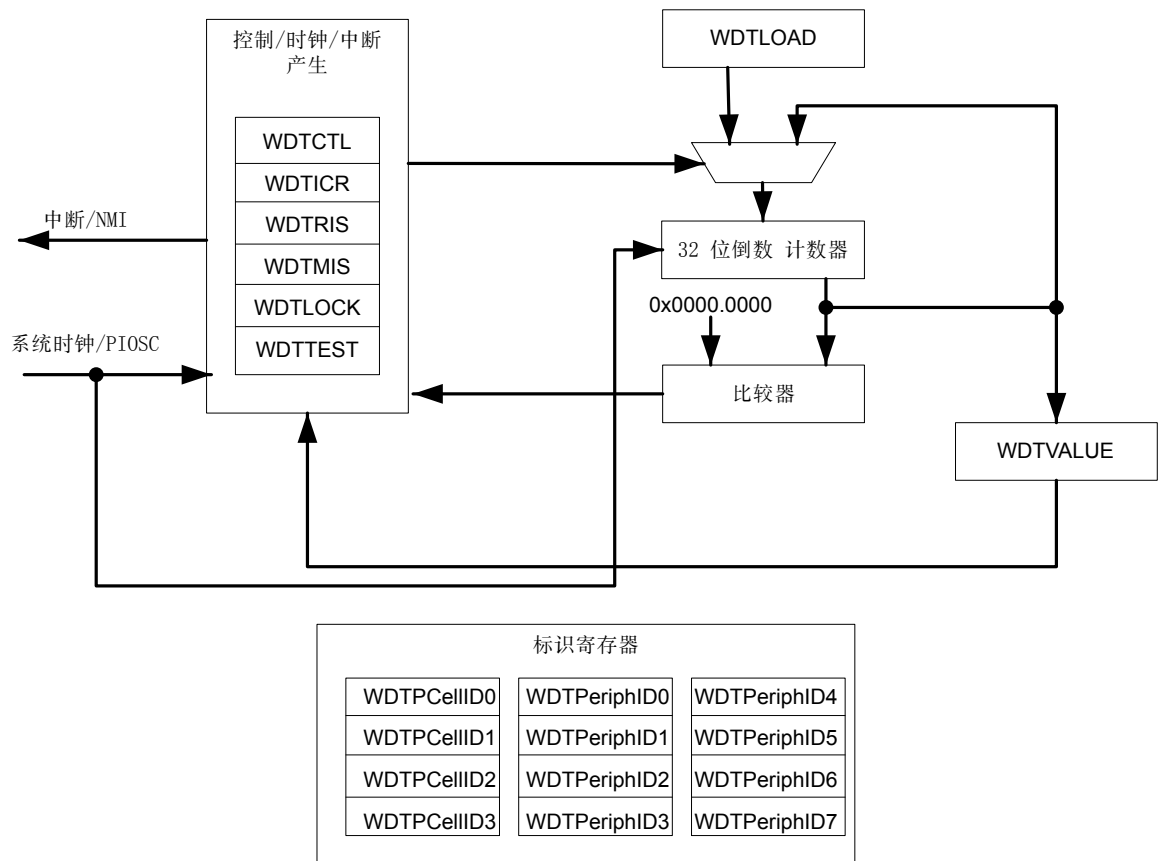
该 Stellaris® LM4F232H5QD 控制器有两个具有如下特性的看门狗定时器模块：

- 32 位递减计数器，带可编程的加载寄存器
- 带启用功能的独立看门狗时钟
- 带中断屏蔽功能和可选 NMI 功能的可编程中断产生逻辑
- 软件跑飞时可锁定寄存器以提供保护
- 带启用/禁能的复位产生逻辑
- 调试期间，微控制器的 CPU 暂停时，用户可启用的停滞

可以将看门狗定时器配置为在第一次超时的时候向控制器发出中断信号；在第二次超时的时候发出一个复位信号。配置好看门狗定时器后，即可写入锁定寄存器，从而防止定时器配置被意外更改。

## 12.1 结构图

图 12-1. WDT 模块的结构图



## 12.2 功能说明

启用 32 位计数器后，如果该计数器倒数至 0，看门狗定时器模块发出第一个超时信号；启用计数器的同时就启用了看门狗定时器的中断功能。使用 WDTCTL 寄存器的 INTTYPE 位可以将看门狗的中断配置为不可屏蔽中断 (NMI)。第一次超时后，32 位计数器重新加载看门狗定时器加载 (WDTLOAD) 寄存器的值，并从该值开始倒数。配置好看门狗定时器后，即会写入看门狗定时器锁定 (WDTLOCK) 寄存器，从而防止定时器配置被软件意外更改。

如果第一次超时中断尚未清除，计数器又倒数至 0，且对 WDTCTL 寄存器中的 RESEN 位进行了置位，从而启用了复位信号，则看门狗定时器会向系统发出复位信号。如果第二次超时发生之时，第一次超时中断已经清除，则 32 位计数器会加载 WDTLOAD 寄存器中的值，并从该值开始计数。

如果在看门狗定时器的计数器正在计数时，将一个新值写入 WDTLOAD，则计数器将加载新值并继续计数。

写入 WDTLOAD 并不会清除活动的中断。必须专门在看门狗中断清除 (WDTICR) 寄存器中进行写入，才能清除中断。

看门狗模块的中断和复位信号可以根据需要来启用或禁用。重新启用中断功能时，32 位计数器会预先载入“加载寄存器”的值，而不是上次所处的状态。

### 12.2.1 寄存器访问时序

因为看门狗定时器 1 (WDT1) 模块具有一个独立的时钟域，所以对寄存器进行写入操作时，各次访问之间必须存在时间差。软件必须保证：在连续的写入 WDT1 寄存器操作之间，或是写入然后读取寄存器操作之间插入足够的延时。对于从 WDT1 模块连续进行读取操作，则没有时间差限制。WDT1 的看门狗控制 (WDTCTL) 寄存器中的 WRC 位表示是否已经达到要求的时间差。该位在进行写入操作时被清零，写入操作完成后便被置位，从而告知软件是否可以安全地开始下一次读写操作。软件在访问另一个寄存器之前，必须查询 WDTCTL 寄存器的 WRC 是否为 1。注意：WDT0 没有该限制，因为它运行在系统时钟下。

## 12.3 初始化及配置

要使用 WDT，必须对 RCGC0n 寄存器中的 WDT 位进行置位，从而启用外设时钟，请参考 442 页。

按照下列顺序配置看门狗定时器：

1. 将所需的定时器值载入 WDTLOAD 寄存器。
2. 如果是 WDT1，则等待 WDTCTL 寄存器的 WRC 位被置位。
3. 如果将看门狗配置为触发系统复位，则对 WDTCTL 寄存器中的 RESEN 位进行置位。
4. 如果是 WDT1，则等待 WDTCTL 寄存器的 WRC 位被置位。
5. 对 WDTCTL 寄存器中的 INTEN 位进行置位，以启用看门狗定时器并锁定控制寄存器。

如果软件需要所有的看门狗寄存器都被锁定，可以向 WDTLOCK 寄存器写入任意值来完全锁定看门狗定时器模块。如要解除对看门狗定时器的锁定，请写入值 0x1ACC.E551。

要维护看门狗，需要定期将计数值重新载入 WDTLOAD 寄存器以重启计数。如果看门狗维护不及时，可通过 WDTCTL 寄存器中的 INTEN 位启用中断，让处理器尝试实施纠正操作。如果使用 ISR 无法恢复故障，可置位 WDTCTL 中的 RESEN 位来复位系统。

## 12.4 寄存器映射

表 12-1 在 719 页列出了看门狗寄存器。所列偏移量是寄存器地址相对于看门狗定时器基址的 16 进制增量，该定时器基址为：

- WDT0 : 0x4000.0000
- WDT1 : 0x4000.1000

请注意：必须先启用看门狗定时器模块的时钟，然后才能对寄存器进行编程（请参考 442 页）。

表 12-1. 看门狗定时器 (WDT) 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	WDTLOAD	R/W	0xFFFF.FFFF	看门狗加载寄存器	721
0x004	WDTVALUE	RO	0xFFFF.FFFF	看门狗当前值寄存器	722
0x008	WDTCTL	R/W	0x0000.0000 (WDT0) 0x8000.0000 (WDT1)	看门狗控制寄存器	723
0x00C	WDTICR	WO	-	看门狗中断清除寄存器	725

表 12-1. 看门狗定时器 (WDT) 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x010	WDTRIS	RO	0x0000.0000	看门狗原始中断状态寄存器	726
0x014	WDTMIS	RO	0x0000.0000	看门狗可屏蔽中断状态寄存器	727
0x418	WDTTEST	R/W	0x0000.0000	看门狗测试寄存器	728
0xC00	WDTLOCK	R/W	0x0000.0000	看门狗锁定寄存器	729
0xFD0	WDTPeriphID4	RO	0x0000.0000	看门狗外设标识寄存器 4	730
0xFD4	WDTPeriphID5	RO	0x0000.0000	看门狗外设标识寄存器 5	731
0xFD8	WDTPeriphID6	RO	0x0000.0000	看门狗外设标识寄存器 6	732
0xFDC	WDTPeriphID7	RO	0x0000.0000	看门狗外设标识寄存器 7	733
0xFE0	WDTPeriphID0	RO	0x0000.0005	看门狗外设标识寄存器 0	734
0xFE4	WDTPeriphID1	RO	0x0000.0018	看门狗外设标识寄存器 1	735
0xFE8	WDTPeriphID2	RO	0x0000.0018	看门狗外设标识寄存器 2	736
0xFEC	WDTPeriphID3	RO	0x0000.0001	看门狗外设标识寄存器 3	737
0xFF0	WDTPrimeCellID0	RO	0x0000.000D	看门狗 PrimeCell 标识寄存器 0	738
0xFF4	WDTPrimeCellID1	RO	0x0000.00F0	看门狗 PrimeCell 标识寄存器 1	739
0xFF8	WDTPrimeCellID2	RO	0x0000.0006	看门狗 PrimeCell 标识寄存器 2	740
0xFFC	WDTPrimeCellID3	RO	0x0000.00B1	看门狗 PrimeCell 标识寄存器 3	741

## 12.5 寄存器描述

本节下文按偏移位址递增的顺序列出 WDT 寄存器，并对它们加以说明。



## 寄存器 1: 看门狗加载寄存器 ( WDTLOAD ) , 偏移量 0x000

该寄存器存放的是 32 位计数器所用的 32 位间隔值。向该寄存器写入新值时，会立即加载该值并从该值开始倒数。如果向 WDTLOAD 寄存器加载值 0x0000.0000，会立即产生一个中断信号。

### 看门狗加载寄存器 (WDTLOAD)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0x000

类型 R/W, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDTLOAD															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTLOAD															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	WDTLOAD	R/W	0xFFFF.FFFF	看门狗定时器加载值

**寄存器 2: 看门狗当前值寄存器 ( WDTVALUE ) , 偏移量 0x004**

该寄存器加载的是定时器的当前计数值。

**看门狗当前值寄存器 (WDTVALUE)**

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0x004

类型 RO, 复位 0xFFFF.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDTVALUE															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTVALUE															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:0	WDTVALUE	RO	0xFFFF.FFFF	看门狗定时器当前值 32 位倒数计数器的当前值

### 寄存器 3: 看门狗控制寄存器 (WDTCTL), 偏移量 0x008

该寄存器是看门狗控制寄存器。可将该看门狗定时器配置为在超时的时候产生一个复位信号 (第二次超时) 或中断。

如果对 INTEN 位进行置位, 从而启用了看门狗的中断功能, 之后对 INTEN 位的所有写入操作都将被忽略。唯一可以对该位重新启用写入操作的方法是硬件复位。

**重要:** 因为看门狗定时器 1 (WDT1) 模块具有一个独立的时钟域, 所以对寄存器进行写入操作时, 各次访问之间必须存在时间差。软件必须保证: 在连续的写入 WDT1 寄存器操作之间, 或是写入然后读取寄存器操作之间插入足够的延时。对于从 WDT1 模块连续进行读取操作, 则没有时间差限制。WDT1 的看门狗控制 (WDTCTL) 寄存器中的 WRC 位表示是否已经达到要求的时间差。该位在进行写入操作时被清零, 写入操作完成后便被置位, 从而告知软件是否可以安全地开始下一次读写操作。软件在访问另一个寄存器之前, 必须查询 WDTCTL 寄存器的 WRC 是否为 1。注意: WDT0 没有此限制, 因为它运行在系统时钟下, 因此没有 WRC 位。

#### 看门狗控制寄存器 (WDTCTL)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0x008

类型 R/W, 复位 0x0000.0000 (WDT0) 和 0x8000.0000 (WDT1)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WRC	保留														
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													INTTYPE	RESEN	INTEN
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31	WRC	RO	1	写入操作完成 WRC 值定义如下:  值 描述 0 正在对某个 WDT1 寄存器进行写入访问。 1 目前没有写入访问, 可以读写 WDT1 寄存器。  注意: 该位是为 WDT0 保留的, 其复位值为 0。
30:3	保留	RO	0x000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2	INTTYPE	R/W	0	看门狗中断类型 INTTYPE 值定义如下:  值 描述 0 看门狗定时器中断为标准中断。 1 看门狗定时器中断为不可屏蔽中断。

位/域	名称	类型	复位	描述
1	RESEN	R/W	0	启用看门狗定时器复位 RESEN 值定义如下：  值 描述 0 禁用。 1 启用看门狗定时器模块复位输出。
0	INTEN	R/W	0	启用看门狗定时器中断 INTEN 值定义如下：  值 描述 0 禁用中断事件（此位一经给置，只能通过硬件复位加以清零）。 1 启用中断事件。一旦启用，所有的写入操作都会被忽略。

**寄存器 4: 看门狗中断清除寄存器 ( WDTICR ) , 偏移量 0x00C**

该寄存器是中断清除寄存器。向该寄存器写入任意值将清除看门狗定时器中断，并且将 WDTLOAD 寄存器中的计数值重新载入 32 位计数器。读取或复位后该寄存器的值是不确定的。

**看门狗中断清除寄存器 (WDTICR)**

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0x00C

类型 WO, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDTINTCLR															
类型	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTINTCLR															
类型	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
复位	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:0	WDTINTCLR	WO	-	看门狗定时器中断清除

**寄存器 5: 看门狗原始中断状态寄存器 ( WDTRIS ) , 偏移量 0x010**

该寄存器是看门狗的原始中断状态寄存器。如果控制器中断被屏蔽，可以通过该寄存器来监视看门狗的中断事件。

## 看门狗原始中断状态寄存器 (WDTRIS)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0x010

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
																WDTRIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	WDTRIS	RO	0	看门狗原始中断状态
				值 描述
				1 看门狗定时器超时事件已经发生。
				0 看门狗定时器尚未达到超时。

## 寄存器 6: 看门狗可屏蔽中断状态寄存器 ( WDTMIS ) , 偏移量 0x014

该寄存器是可屏蔽的中断状态寄存器。该寄存器的值是原始中断位和看门狗中断启用位的交集。

### 看门狗可屏蔽中断状态寄存器 (WDTMIS)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0x014

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
	WDTMIS															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	WDTMIS	RO	0	看门狗可屏蔽中断状态

#### 值 描述

- 1 看门狗定时器超时事件已经向中断控制器发出超时信号。
- 0 看门狗定时器尚未达到超时，或者看门狗定时器中断已经被屏蔽。

## 寄存器 7: 看门狗测试寄存器 ( WDTTEST ) , 偏移量 0x418

该寄存器可在调试期间，微控制器发出 CPU 暂停 (halt) 标志时，让用户启用暂停操作 (stall)。

## 看门狗测试寄存器 (WDTTEST)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0x418

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留							STALL	保留							
类型	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:9	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
8	STALL	R/W	0	启用看门狗定时器暂停  <b>值 描述</b> 1 如果微控制器因为调试而停止工作，则看门狗定时器停止计数。一旦微控制器重新启动，看门狗定时器也会恢复计数。 0 如果微控制器因为调试而停止工作，看门狗定时器将继续计数。
7:0	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。



## 寄存器 8: 看门狗锁定寄存器 ( WDTLOCK ) , 偏移量 0xC00

向 WDTLOCK 寄存器写入 0x1ACC.E551 可对其它所有寄存器执行写入操作。向 WDTLOCK 寄存器写入其它任意值会重新启用寄存器的锁定状态，不能对所有其它寄存器执行写入操作。读取 WDTLOCK 寄存器时返回的是锁定的状态而不是被写入的 32 位值。因此，在写入访问被禁用时，读取 WDTLOCK 寄存器将返回 0x0000.0001 ( 已锁定 ) ；否则，返回的值为 0x0000.0000 ( 未锁定 ) 。

### 看门狗锁定寄存器 (WDTLOCK)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xC00

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDTLOCK															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTLOCK															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述						
31:0	WDTLOCK	R/W	0x0000.0000	看门狗锁定 写入值 0x1ACC.E551 可解除对看门狗寄存器的写入访问锁定。写入其它任何值会重新应用锁定，防止对寄存器进行更新。 读取该寄存器返回如下值：  <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0000.0001</td> <td>锁定</td> </tr> <tr> <td>0x0000.0000</td> <td>未锁定</td> </tr> </tbody> </table>	值	描述	0x0000.0001	锁定	0x0000.0000	未锁定
值	描述									
0x0000.0001	锁定									
0x0000.0000	未锁定									

## 寄存器 9: 看门狗外设标识寄存器 4 ( WDTPeriphID4 ) , 偏移量 0xFD0

WDTPeriphIDn 寄存器为硬编码寄存器，寄存器的域决定复位值。

## 看门狗外设标识寄存器 4 (WDTPeriphID4)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFD0

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID4							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID4	RO	0x00	WDT 外设标识寄存器 [7:0]

## 寄存器 10: 看门狗外设标识寄存器 5 ( WDTPeriphID5 ) , 偏移量 0xFD4

WDTPeriphIDn 寄存器为硬编码寄存器，寄存器的域决定复位值。

### 看门狗外设标识寄存器 5 (WDTPeriphID5)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFD4

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID5							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID5	RO	0x00	WDT 外设标识寄存器 [15:8]

## 寄存器 11: 看门狗外设标识寄存器 6 ( WDTPeriphID6 ) , 偏移量 0xFD8

WDTPeriphIDn 寄存器为硬编码寄存器，寄存器的域决定复位值。

## 看门狗外设标识寄存器 6 (WDTPeriphID6)

WDT0 基址: 0x4000.0000  
 WDT1 基址: 0x4000.1000  
 偏移量 0xFD8  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID6							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID6	RO	0x00	WDT 外设标识寄存器 [23:16]

## 寄存器 12: 看门狗外设标识寄存器 7 ( WDTPeriphID7 ) , 偏移量 0xFDC

WDTPeriphIDn 寄存器为硬编码寄存器，寄存器的域决定复位值。

## 看门狗外设标识寄存器 7 (WDTPeriphID7)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFDC

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID7							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID7	RO	0x00	WDT 外设标识寄存器 [31:24]

## 寄存器 13: 看门狗外设标识寄存器 0 ( WDTPeriphID0 ) , 偏移量 0xFE0

WDTPeriphIDn 寄存器为硬编码寄存器，寄存器的域决定复位值。

## 看门狗外设标识寄存器 0 (WDTPeriphID0)

WDT0 基址: 0x4000.0000  
WDT1 基址: 0x4000.1000  
偏移量 0xFE0  
类型 RO, 复位 0x0000.0005

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID0	RO	0x05	看门狗外设标识寄存器 [7:0]

## 寄存器 14: 看门狗外设标识寄存器 1 ( WDTPeriphID1 ) , 偏移量 0xFE4

WDTPeriphIDn 寄存器为硬编码寄存器，寄存器的域决定复位值。

### 看门狗外设标识寄存器 1 (WDTPeriphID1)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFE4

类型 RO, 复位 0x0000.0018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID1	RO	0x18	看门狗外设标识寄存器 [15:8]

## 寄存器 15: 看门狗外设标识寄存器 2 ( WDTPeriphID2 ) , 偏移量 0xFE8

WDTPeriphIDn 寄存器为硬编码寄存器，寄存器的域决定复位值。

## 看门狗外设标识寄存器 2 (WDTPeriphID2)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFE8

类型 RO, 复位 0x0000.0018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID2	RO	0x18	看门狗外设标记寄存器 [23:16]



## 寄存器 16: 看门狗外设标识寄存器 3 ( WDTPeriphID3 ) , 偏移量 0xFEC

WDTPeriphIDn 寄存器为硬编码寄存器，寄存器的域决定复位值。

## 看门狗外设标识寄存器 3 (WDTPeriphID3)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFEC

类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID3	RO	0x01	看门狗外设标识寄存器 [31:24]

## 寄存器 17: 看门狗 PrimeCell 标识寄存器 0 ( WDTPCellID0 ) , 偏移量 0xFF0

WDTPCellIDn 寄存器均为硬编码寄存器，寄存器的域决定复位值。

## 看门狗 PrimeCell 标识寄存器 0 (WDTPCellID0)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFF0

类型 RO, 复位 0x0000.000D

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID0	RO	0x0D	看门狗 PrimeCell 标识寄存器 [7:0]

## 寄存器 18: 看门狗 PrimeCell 标识寄存器 1 ( WDTPCellID1 ), 偏移量 0xFF4

WDTPCellIDn 寄存器均为硬编码寄存器，寄存器的域决定复位值。

## 看门狗 PrimeCell 标识寄存器 1 (WDTPCellID1)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFF4

类型 RO, 复位 0x0000.00F0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID1	RO	0xF0	看门狗 PrimeCell 标识寄存器 [15:8]

## 寄存器 19: 看门狗 PrimeCell 标识寄存器 2 ( WDTPCellID2 ) , 偏移量 0xFF8

WDTPCellIDn 寄存器均为硬编码寄存器，寄存器的域决定复位值。

## 看门狗 PrimeCell 标识寄存器 2 (WDTPCellID2)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFF8

类型 RO, 复位 0x0000.0006

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID2	RO	0x06	看门狗 PrimeCell 标识寄存器 [23:16]

## 寄存器 20: 看门狗 PrimeCell 标识寄存器 3 ( WDTPCellID3 ) , 偏移量 0xFFC

WDTPCellIDn 寄存器均为硬编码寄存器, 寄存器的域决定复位值。

## 看门狗 PrimeCell 标识寄存器 3 (WDTPCellID3)

WDT0 基址: 0x4000.0000

WDT1 基址: 0x4000.1000

偏移量 0xFFC

类型 RO, 复位 0x0000.00B1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID3	RO	0xB1	看门狗 PrimeCell 标识寄存器 [31:24]

## 13 模数转换器 (ADC)

模数转换器 (ADC) 外设用于将连续的模拟电压转换成离散的数字量。包含两个完全相同的转换器模块，它们共用 24 个输入通道。

该 Stellaris<sup>®</sup> ADC 模块的转换分辨率为 12 位，并支持 24 个输入通道 以及一个内部温度传感器。每个 ADC 模块都包含 4 个可编程的序列发生器，无需控制器干预即可自动完成对多个模拟输入源的采样。每个采样序列发生器都可灵活配置其输入源、触发事件、中断的产生、序列发生器的优先级等内容。ADC 模块内置数字比较器功能，采样转换结果可移交给数字比较器模块。每个 ADC 模块提供 8 个数字比较器。每个数字比较器模块内置 16 路数字比较器，每路数字比较器均可将 ADC 转换结果数值与 2 个由用户定义的门限值进行比较，以确定信号的工作范围。ADC0 和 ADC1 可各自采用不同的触发源，也可采用相同的触发源；可各自采用不同的模拟输入端，也可采用同一模拟输入端。ADC 模块内部还具有移相器，可将采样开始时间（采样点）延后指定的相角。因此当两个 ADC 模块同时工作时，其采样点既可以配置为同相工作，也可以配置为相互错开一定的相角，详见“采样相位控制”在 747 页。

该 Stellaris LM4F232H5QD 微处理器提供 2 个 ADC 模块，每个模块都具有以下特性：

- 24 个共用模拟输入通道
- 12 位精度的 ADC
- 单端和差分输入配置
- 片上内置温度传感器
- 最大采样率为 1M 次/秒
- 可选的移相器，采样点以采样周期计可延后 22.5° 到 337.5°
- 4 个可编程的采样转换序列发生器，序列长度 1 到 8 个单元不等，且各自带有相应长度的转换结果 FIFO
- 灵活的触发控制
  - 控制器（软件）
  - 定时器
  - 模拟比较器
  - PWM
  - GPIO
- 硬件可对多达 64 个采样值进行平均计算
- 数字比较器模块提供 8 路数字比较器
- 转换器使用两个外部参考信号或 VDDA 和 GNDA 作为电压参考
- 模拟部分的电源/地与数字部分的电源/地相互独立
- 用微型直接内存访问 (μDMA) 有效的传输数据
  - 每个采样序列发生器各自有专用的通道

– ADC 模块的 DMA 操作均采用猝发请求

## 13.1 结构图

该 Stellaris 微控制器内置两个相同的模数转换器 (ADC) 模块。这两个模块 (ADC0 和 ADC1) 共用相同的 24 个模拟输入通道。两个 ADC 模块的工作相互独立, 因此可同时执行不同的采样序列、同时对任一模拟输入通道进行采样、并各自产生不同的中断和触发事件。图 13-1 在 743 页显示了这两个 ADC 模块是如何与模拟输入端以及系统总线连接的。

图 13-1. 两个 ADC 模块的连接结构图

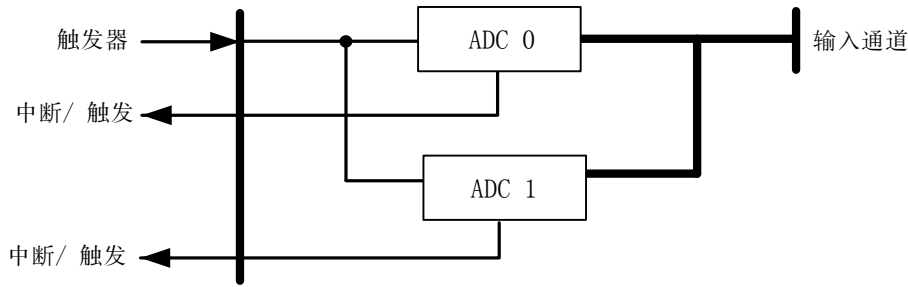
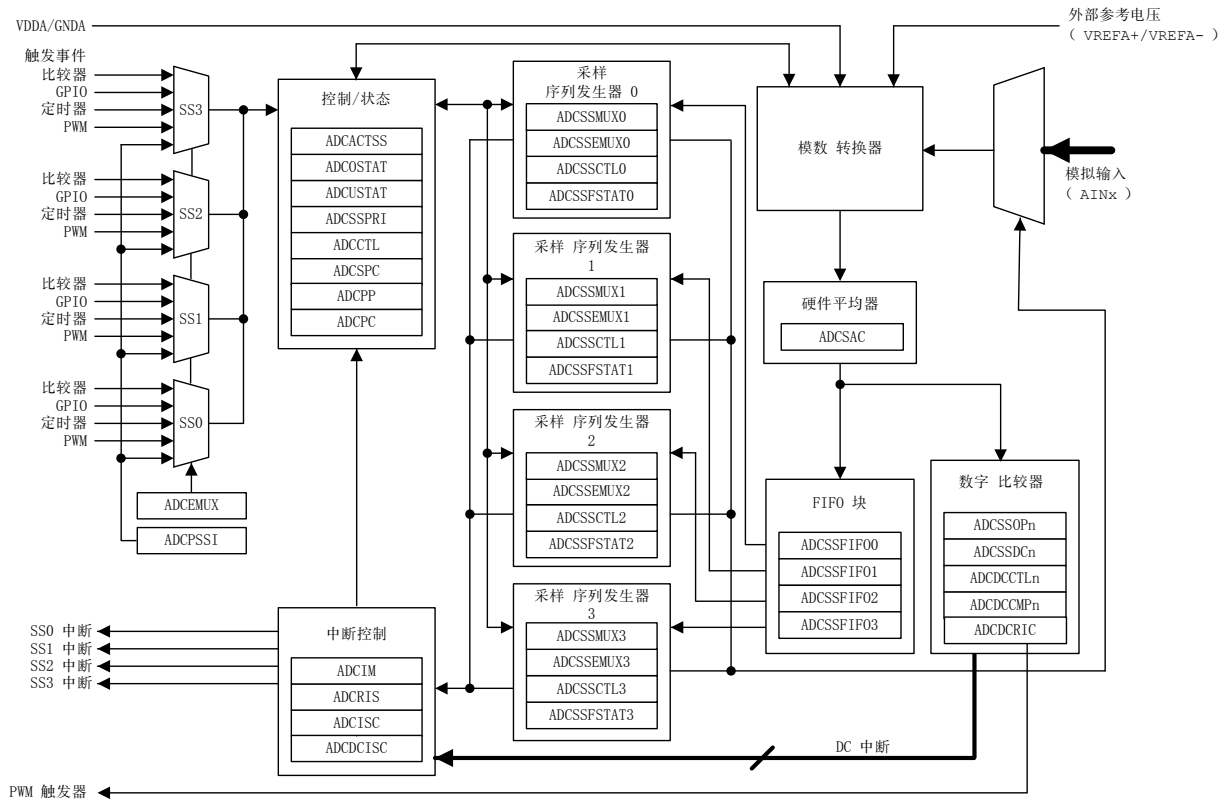


图 13-2 在 743 页显示了 ADC 模块中控制寄存器及数据寄存器的内部配置情况。

图 13-2. ADC 模块结构图



## 13.2 信号描述

下表列出了与 ADC 模块相关的所有外部信号并逐一描述其功能。AINx 信号是某些 GPIO 信号的模拟功能。表中“复用管脚/赋值”一列是各 ADC 信号所对应的 GPIO 管脚。这些管脚的配置方式为：将 GPIO 数字使能 (GPIODEN) 寄存器中相应的 DEN 位清零，并将 GPIO 模拟模式选择 (GPIOAMSEL) 寄存器中相应的 AMSEL 位置位。关于如何配置 GPIO 的更多信息，请参考“通用输入/输出端口 (GPIO)”在 597 页。VREFA+ 和 VREFA- 信号（在“复用管脚/赋值”栏中标记为“固定”）有一个固定的管脚分配和功能。该管脚不能承受 5V 的电压。

表 13-1. ADC 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
AIN0	12	PE3	I	模拟	模数转换器输入 0。
AIN1	13	PE2	I	模拟	模数转换器输入 1。
AIN2	14	PE1	I	模拟	模数转换器输入 2。
AIN3	15	PE0	I	模拟	模数转换器输入 3。
AIN4	144	PD7	I	模拟	模数转换器输入 4。
AIN5	143	PD6	I	模拟	模数转换器输入 5。
AIN6	142	PD5	I	模拟	模数转换器输入 6。
AIN7	141	PD4	I	模拟	模数转换器输入 7。
AIN8	140	PE5	I	模拟	模数转换器输入 8。
AIN9	139	PE4	I	模拟	模数转换器输入 9。
AIN10	136	PB4	I	模拟	模数转换器输入 10。
AIN11	135	PB5	I	模拟	模数转换器输入 11。
AIN12	4	PD3	I	模拟	模数转换器输入 12。
AIN13	3	PD2	I	模拟	模数转换器输入 13。
AIN14	2	PD1	I	模拟	模数转换器输入 14。
AIN15	1	PD0	I	模拟	模数转换器输入 15。
AIN16	16	PK0	I	模拟	模数转换器输入 16。
AIN17	17	PK1	I	模拟	模数转换器输入 17。
AIN18	18	PK2	I	模拟	模数转换器输入 18。
AIN19	19	PK3	I	模拟	模数转换器输入 19。
AIN20	134	PE7	I	模拟	模数转换器输入 20。
AIN21	133	PE6	I	模拟	模数转换器输入 21。
AIN22	132	PP1	I	模拟	模数转换器输入 22。
AIN23	131	PP0	I	模拟	模数转换器输入 23。
VREFA+	8	固定	-	模拟	参考电压用于指定 ADC 转换为最大值时的电压。此管脚与指定最小值的 VREFA- 结合使用。也就是说，若 AINn 输入的信号电压与 VREFA+ 电压相等，则转换结果为 4095。VREFA+ 的电压限定为表 25-25 在 1307 页中指定的范围。
VREFA-	9	固定	-	模拟	参考电压用于指定 ADC 转换为最小值时的输入电压。此管脚与指定最大值的 VREFA+ 结合使用。也就是说，若 AINn 输入的信号电压与 VREFA- 电压相等，则转换结果为 0。而若 AINn 输入的信号电压与 VREFA+ 电压相等，则转换结果为 4095。VREFA- 的电压限定为表 25-25 在 1307 页中指定的范围。

a. TTL 表示管脚的电压水平与 TTL 一致。



## 13.3 功能描述

该 Stellaris ADC 通过使用一种基于序列的可编程方法来收集采样数据，取代了传统 ADC 模块使用的单次采样或双采样的方法。每个采样序列 (Sample Sequence) 均由一组连续的采样动作组成，因此 ADC 模块可以自动从多个输入源采集数据，无需微控制器对其重新配置或进行干预。采样序列中的每个采样动作都可灵活编程，可配置的参数包括选择输入源和输入模式（单端输入或差分输入）、采样结束时是否产生中断、是否是队列中最后一个采样动作的标识符等等。此外，若结合  $\mu$ DMA 工作，ADC 模块能够更加高效地从采样序列中获取数据，同时无需 CPU 进行任何干预。

### 13.3.1 采样序列发生器

采样控制和数据采集都是由采样序列发生器 (Sample Sequencer, 简称为 SS) 处理的。所有序列发生器的实现方法都是相同的，区别仅在于能够捕捉的采样数以及 FIFO 深度有所不同。表 13-2 在 745 页给出了每个序列发生器可捕获的最大采样数及其相对应的 FIFO 深度。捕捉到的每个采样都要存入 FIFO 中。在本实现方案中，每个 FIFO 入口均为一个 32 位的字，低 12 位包含的是转换结果。

表 13-2. 采样序列发生器的采样数和 FIFO 深度

采样序列器	采样数目	FIFO 深度
SS3	1	1
SS2	4	4
SS1	4	4
SS0	8	8

对于指定的采样序列，若以  $n$  代表其序号，则采样序列  $n$  中的每个采样动作分别以 ADC 采样序列输入多路复用器选择 (ADCSSMUX $n$ )、ADC 采样序列扩展输入复用选择 (ADCSEMUX $n$ ) 及 ADC 采样序列控制 (ADCSSCTL $n$ ) 中的 1 个半字节予以定义。该 ADCSSMUX $n$  和 ADCSEMUX $n$  用于选择输入管脚，而 ADCSSCTL $n$  包含采样控制位，这些控制位分别与参数（例如，温度传感器的选择、中断启用、序列末端和差分输入模式）对应。采样序列发生器可以通过置位 ADC 活动采样序列发生器 (ADCACTSS) 寄存器中相应的 ASEN $n$  位进行启用，但也可以在启用之前进行配置。软件可通过置位 ADC 处理器采样序列启动 (ADCPSSI) 寄存器的 SS $n$  位来启动采样。此外，通过配置 ADCPSSI 寄存器的 GSYNC 和 SYNCWAIT 位，可同时启动多个 ADC 模块的多个采样序列。关于这些配置位的详细信息，请参阅 783 页。

配置采样序列时，允许同一序列中的多个采样动作对同一输入端进行采样。ADCSSCTL $n$  寄存器中的 IEn 位可任意组合置位，如果有必要的话能够在采样序列的每个采样动作时都产生中断。同样，END 位也可以放置在采样序列中的任意位置。举例来说，假设使用采样序列 0，并且将与第 5 个采样动作相关的半字中的 END 位置位，那么采样序列 0 将在完成第 5 个采样动作后结束执行。

当采样序列执行结束后，可从 ADC 采样序列结果 FIFO 寄存器 (ADCSSFIFO $n$ ) 中读取采样结果数据。ADC 模块的 FIFO 均为简单的环型缓冲区，反复读取同一地址 (ADCSSFIFO $n$ ) 即可依次“弹出”结果数据。为了方便软件调试，通过 ADC 采样序列 FIFO 状态寄存器 (ADCSSFSTAT $n$ ) 可查询到 FIFO 头指针、尾指针的位置以及 FULL (满) 和 EMPTY (空) 状态标志。如果 FIFO 已满，再进行写操作时，该写操作会失败，该 FIFO 会出现上溢状况。通过 ADCOSTAT 和 ADCUSTAT 寄存器可监控上溢和下溢状态。

### 13.3.2 模块控制

控制逻辑单元中除采样序列发生器的剩余部分负责执行以下任务：

- 中断的产生
- DMA 操作
- 采样序列按优先级执行

- 触发事件的配置
- 比较器的配置
- 外部参考电压
- 采样相位控制
- 模块计时

大多数的 ADC 控制逻辑都以 16 MHz 的 ADC 时钟速率运行。当系统 XTAL 选择 PLL 时，硬件将自动配置内部的 ADC 分频器，并按照 16 MHz 频率工作。

### 13.3.2.1 中断

采样序列发生器和数字比较器数字比较器的寄存器配置可以监控产生原始中断的事件，但对中断是否真正发送给中断控制器没有控制权。ADC 模块是否产生中断信号是由 ADC 中断掩码 (ADCIM) 寄存器的 MASK 位决定的。中断状态可以从以下两个位置查询：ADC 原始中断状态寄存器 (ADCRIS) 显示各个中断信号的原始状态；ADC 中断及清除寄存器 (ADCISC) 显示经 ADCIM 寄存器启用后的实际中断状态。通过向 ADCISC 对应的 IN 位写 1 来清除中断。请注意，数字比较器中断不是通过本寄存器清除的，而是通过向 ADC 数字比较器中断状态及清除寄存器 (ADCDCISC) 的对应位写 1 来清除的。

### 13.3.2.2 DMA操作

如果使用 DMA，则每个采样序列发生器能够独立工作，无需微控制器干预或重新配置即可传输数据，从而提高了效率。每个采样序列发生器都可向  $\mu$ DMA 控制器中相关的专用通道发送请求。ADC 不支持单次的传输请求。当采样序列的中断标志置位时 (ADCSSCTLn 寄存器的 IE 位置位) 产生猝发传输请求。

$\mu$ DMA 传输的仲裁大小必须是 2 的整数幂，并且 ADCSSCTLn 寄存器的相关 IE 位必须置位。例如，若 SS0 的  $\mu$ DMA 通道大小为 4，那么必须将 IE3 (第 4 个采样动作) 和 IE7 (第 8 个采样动作) 置位。因此，每 4 个采样动作后会触发 1 次  $\mu$ DMA 请求。除此之外不需要其它特殊步骤，ADC 模块已经能够进行  $\mu$ DMA 工作。

关于  $\mu$ DMA 控制器编程的更多信息，请参阅“微型直接存储器访问 ( $\mu$ DMA)”在 534 页。

### 13.3.2.3 优先级

当同时出现多个采样事件 (触发条件) 时，按照 ADC 采样序列器优先级寄存器 (ADCSSPRI) 所定义的优先级顺序依次处理。优先级的有效值为 0~3，其中 0 代表最高优先级、3 代表最低优先级。如果多个活动的采样序列具有相同的优先级，将导致转换结果数据不连续，因此软件必须确保当前活动的所有采样序列各自具有唯一的优先级。

### 13.3.2.4 采样事件

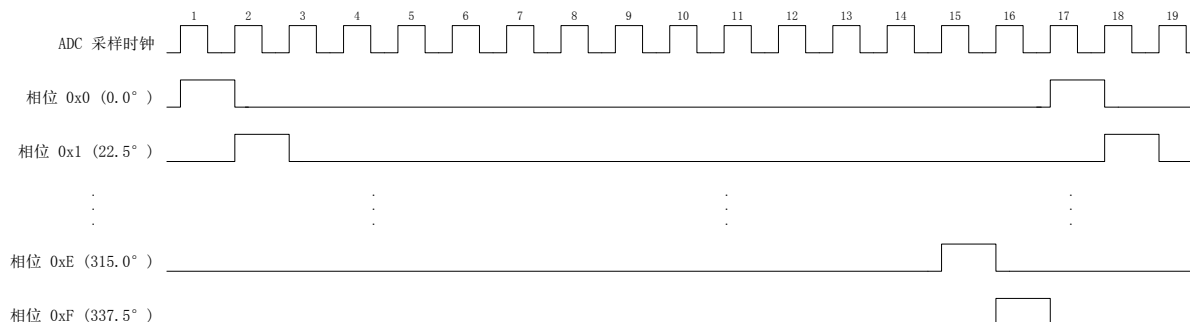
每个采样序列发生器的采样触发条件可通过 ADC 事件复用选择寄存器 (ADCEMUX) 予以定义。触发事件源包括处理器触发 (默认)、模拟比较器触发、GPIO ADC 控制 (GPIOADCCTL) 寄存器指定的 GPIO 外部信号触发、通用定时器触发、PWM 发生器触发以及持续采样触发。软件可以将 ADC 处理器采样序列启动 (ADCPSSI) 的 SSx 位置位来启动采样序列。

配置持续采样触发条件时务必慎重。假如某个采样序列的优先级过高，可能导致其它低优先级采样序列始终无法运行。通常，要将使用连续采样模式的采样序列器设置位最低优先级。当输入接口上的电压达到了某一特定值，连续采样可以和数字比较器配合使用以产生中断。

### 13.3.2.5 采样相位控制

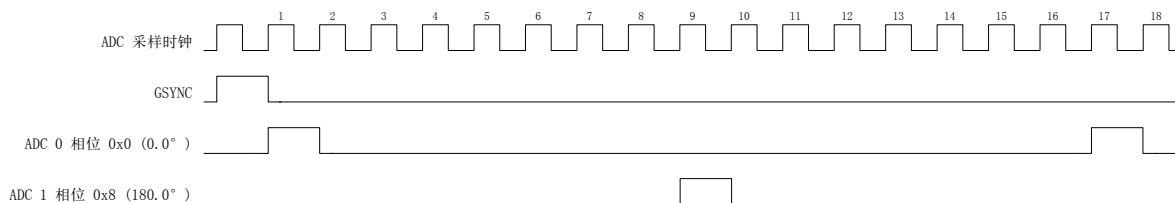
ADC0 和 ADC1 可各自采用不同的触发源，也可采用相同的触发源；可各自采用不同的模拟输入端，也可采用同一模拟输入端。假如两个转换器以相同的采样率工作，其采样点既可以配置为同相，也可以配置为相互错开一定的相角（可实现15种离散的相位差）。采样点延后的相位从  $0^\circ$  到  $337.5^\circ$  之间以  $22.5^\circ$  逐步递增，并通过 ADC 采样相位控制寄存器（ADCSPC）予以设置。图 13-3 在 747 页显示了 1 Msps 采样率时各种不同的相位关系示例。

图 13-3. ADC 采样相位



借助此功能可对单个输入通道实现双倍采样率。将 ADC0 和 ADC1 模块配置为采用同一个输入通道，ADC0 模块可以按照标准相位采样（ADCSPC 寄存器的 PHASE = 0）。ADC1 模块可以配置为延后  $180^\circ$  相位采样（PHASE = 0x8）。通过 ADC 处理器采样序列启动（ADCPSSI）寄存器的 GSYNC 和 SYNCWAIT 位可以将两个模块配置位同步运行。然后由软件将来自两个模块的结果数据进行组合，就能在 16 MHz（参见图 13-4 在 747 页）工作频率下实现 2M 的采样率。

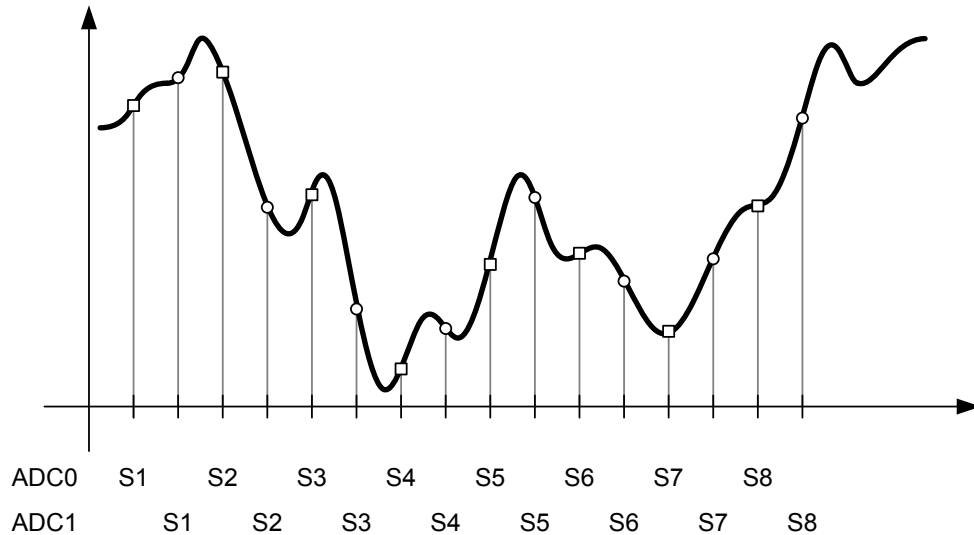
图 13-4. ADC 采样率倍增



运用 ADCSPC 寄存器还能实现许多有趣的应用：

- 不同信号的同相采样。两个转换器的采样序列同相进行。
  - ADC 模块 0：ADCSPC = 0x0，对 AIN0 采样
  - ADC 模块 1：ADCSPC = 0x0，对 AIN1 采样
- 同一信号的交错采样。两个模块的采样序列按照 ADC 时钟的 1/2（当采样率 1 Msps 时为  $500\mu\text{s}$ ）交错进行采样。如果软件将两个模块的转换结果进行交错组合，即可将单个输入通道的转换带宽加倍，如图 13-5 在 748 页所示。
  - ADC 模块 0：ADCSPC = 0x0，对 AIN0 采样
  - ADC 模块 1：ADCSPC = 0x8，对 AIN0 采样

图 13-5. 交错采样



### 13.3.2.6 模块计时

该模块由一个 16-MHz 的时钟计时，该时钟可通过分频的 PLL 输出、PIOSC 或连接到 MOSC 的外部信号（PLL 处于旁路模式）获得。PLL 工作时，ADC 时钟通过  $PLL \div 25$  得到（默认）。但是，PIOSC 可用于使用 ADC 时钟配置 (ADCC) 寄存器的模块时钟。PLL 处于旁路模式时，连接到 MOSC 的模块时钟源时钟必须为 16 MHz，除非该时钟源使用 PIOSC。如果 PIOSC 是 ADC 模块的时钟源，则 ADC 模块可以继续以深度睡眠模式工作。

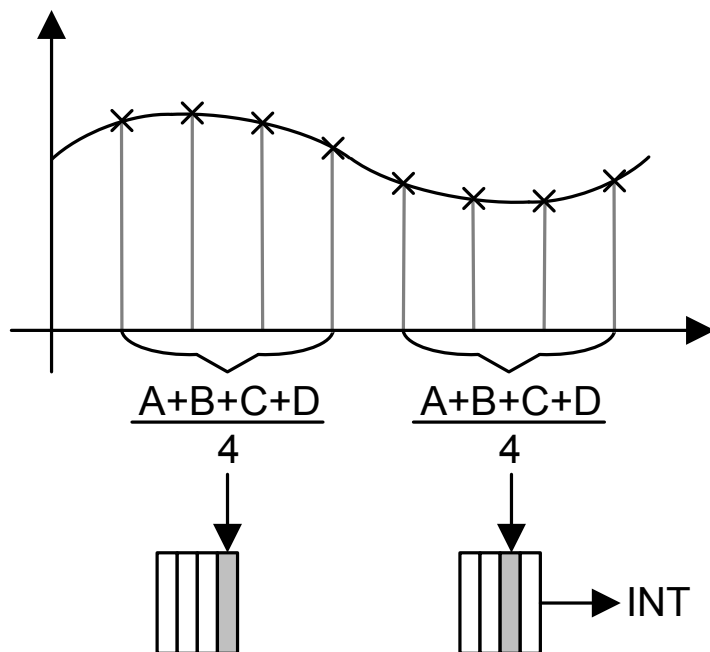
### 13.3.3 硬件采样平均电路

启用硬件采样平均电路可以获得更高的精度，与此同时付出的代价是吞吐率将成比例地降低。硬件采样平均电路最高可将 64 次采样结果累加并计算出平均值，以平均值作为单次采样的数据写入序列发生器 FIFO 的 1 个单元中。由于是算术平均值，因此吞吐率与求平均值的采样数目成反比。例如，若取 16 次采样进行平均值计算，那么吞吐率将降为 1/16。

默认情况下，硬件采样平均电路是关闭的，转换器捕捉的所有数据直接送入序列发生器的 FIFO 中。进行平均计算的硬件由 ADC 采样平均控制 (ADCSAC) 寄存器进行控制（见 785 页）。每个 ADC 模块只有一个平均电路，不论单端输入还是差分输入都会被执行相同的求平均值操作。

图 13-6 中显示一个实例。在该实例中，ADCSAC 寄存器设置为 0x2 以进行 4x 硬件过采样；IE1 被置位以提供采样队列；第 2 个平均值储存进 FIFO 后，将产生中断信号。

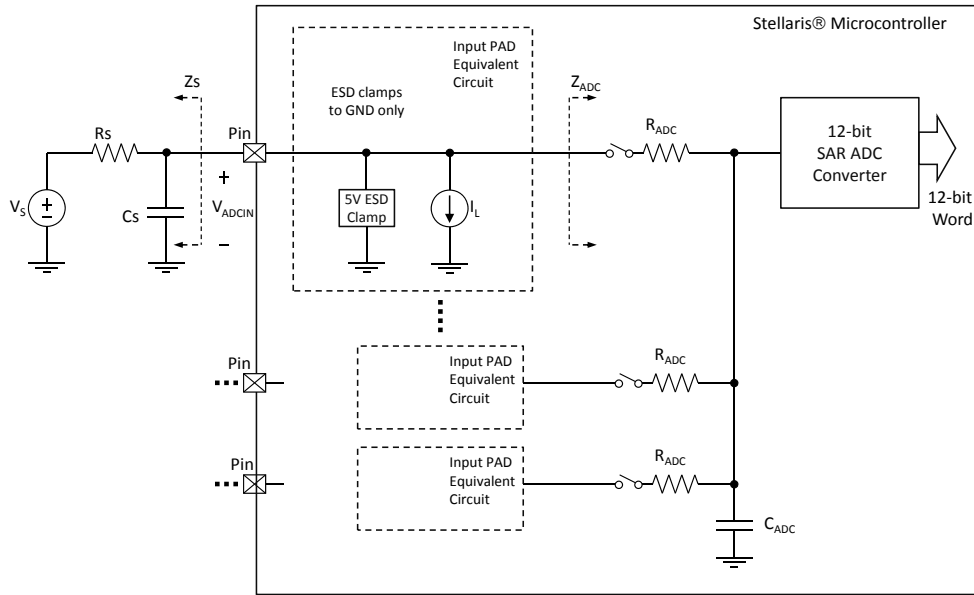
图 13-6. 采样平均的实例



### 13.3.4 模数转换器

模-数转换器 (ADC) 模块采用逐次逼近寄存器 (Successive Approximation Register, 简称为 SAR) 架构实现低功耗、高精度的 12 位 A/D 转换。该逐次逼近架构使用开关电容阵列执行两种功能：采集和保持信号，提供 12 位 DAC 操作。ADC 要求使用 16 MHz 时钟。该时钟可以是分频的 PLL 输出、PIOSC 或连接到 MOSC 的 16 MHz 时钟源。MOSC 的效果最佳，其次是分频的 PLL，最后是 PIOSC。图 13-7 显示 ADC 输入端等效框图；参数值请参考“Analog-to-Digital Converter (ADC)”在 1307 页。

图 13-7. ADC 输入端等效框图

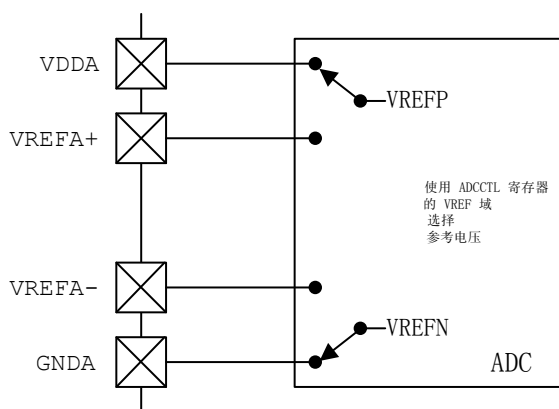


ADC模块同时从3.3V模拟电源和1.2V数字电源取电。在不要求ADC转换精度时，可以将ADC时钟配置为低功耗（请参阅“系统控制”在215页）。模拟信号通过特殊的平衡输入通道连接到ADC，尽量减少输入信号的失真和串扰。关于ADC模块供电及模拟输入的详细信息，请参阅“Analog-to-Digital Converter (ADC)”在1307页。

#### 13.3.4.1 参考电压

ADC使用内部信号VREFP和VREFN作为参考电压源，以对选定的模拟输入电压进行转换。根据ADC控制(ADCCTL)寄存器中VREF位的配置，VREFP既可以连接到VREFA+也可以连接到VDDA；VREFN既可以连接到VREFA-也可以连接到GNDA，请参见图13-8。

图 13-8. ADC 参考电压

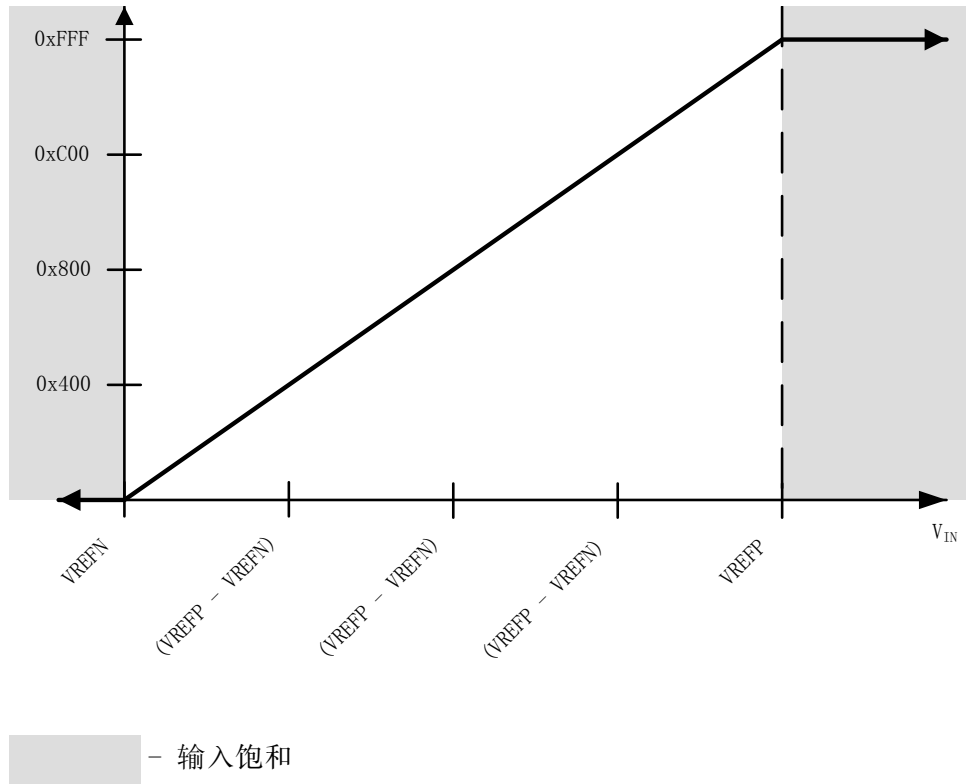


转换值的范围为 0x000~0xFFF。在单端输入模式下，0x000 对应于 VREFN 上的电平；0xFFF 对应于 VREFP 上的电平。通过这样的配置，分辨率就可以使用以下等式计算：

$$\text{mV per ADC code} = (\text{VREFP} - \text{VREFN}) / 4096$$

虽然模拟输入管脚能够承受超出此范围的电压，但在欠压及过压时 A/D 转换结果将会饱和。VREFP 上的模拟输入电压的饱和值为 0xFFF；VREFN 上的模拟输入电压的饱和值为 0x000。V<sub>REF+</sub> 和 V<sub>REF-</sub> 规格定义了 VDDA+ 和 VDDA- 上的外部参考电压的有效范围，请参考表 25-25 在 1307 页。当采用外部参考电压时务必慎重，参考电压源的质量必须符合要求。图 13-9 在 752 页显示 ADC 转换与输入模拟电压的函数关系。

图 13-9. ADC 转换结果



### 13.3.5 差分采样

除了传统的单端采样，ADC模块还支持对两个模拟输入通道进行差分采样。要启用差分采样功能，需在 ADCSSCTL0n 寄存器中将某个采样步骤配置半字节的 Dn 位置位。

若采样序列中某个采样动作配置为差分采样，必须通过 ADCSSMUXn 寄存器选择输入的差分信号对。差分信号对 0 对模拟输入端 0 和 1 进行采样，差分信号对 1 对模拟输入端 2 和 3 进行采样，依此类推（见表 13-3 在 752 页）。ADC 不支持差分信号对的随意组合，如模拟输入端 0 和模拟输入端 3 无法作为一对差分信号输入。

表 13-3. 差分采样对

差分信号对	模拟输入
0	0 和 1
1	2 和 3
2	4 和 5
3	6 和 7
4	8 和 9
5	10 和 11
6	12 和 13
7	14 和 15



表 13-3. 差分采样对 (续)

差分信号对	模拟输入
8	16 和 17
9	18 和 19
10	20 和 21
11	22 和 23

差分模式下采样电压是奇数通道与偶数通道电压的差值：

- 正向输入电压： $V_{IN+} = V_{IN\_EVEN}$  (偶数通道电压)
- 负向输入电压： $V_{IN-} = V_{IN\_ODD}$  (奇数通道电压)

差分输入电压定义为： $V_{IN_D} = V_{IN+} - V_{IN-}$ ，因此：

- 若  $V_{IN_D} = 0$ ，则转换结果 = 0x800
- 若  $V_{IN_D} > 0$ ，则转换结果 > 0x800 (范围是 0x800 ~ 0xFFFF)
- 若  $V_{IN_D} < 0$ ，则转换结果 < 0x800 (范围是 0-0x800)

使用差分采样时，还需考虑以下定义：

- 输入共模电压： $V_{IN_{CM}} = (V_{IN+} + V_{IN-}) / 2$
- 正向参考电压： $V_{REFP}$
- 负向参考电压： $V_{REFN}$
- 差分参考电压： $V_{REF_D} = V_{REFP} - V_{REFN}$
- 参考共模电压： $V_{REF_{CM}} = (V_{REFP} + V_{REFN}) / 2$

差分模式具备以下条件时效果最佳：

- $V_{IN\_EVEN}$  和  $V_{IN\_ODD}$  必须在 ( $V_{REFP}$  至  $V_{REFN}$ ) 范围内，否则无法得到有效的转换结果
- 最大可能的差分输入摆幅或最大的差分范围为： $-V_{REF_D}$  至  $+V_{REF_D}$ ，因此最大的峰峰差分输入信号为  $(+V_{REF_D} - -V_{REF_D}) = 2 * V_{REF_D} = 2 * (V_{REFP} - V_{REFN})$
- 为了利用最大可能的差分输入摆幅， $V_{IN_{CM}}$  应非常接近  $V_{REF_{CM}}$ ，请参考表 25-25 在 1307页。

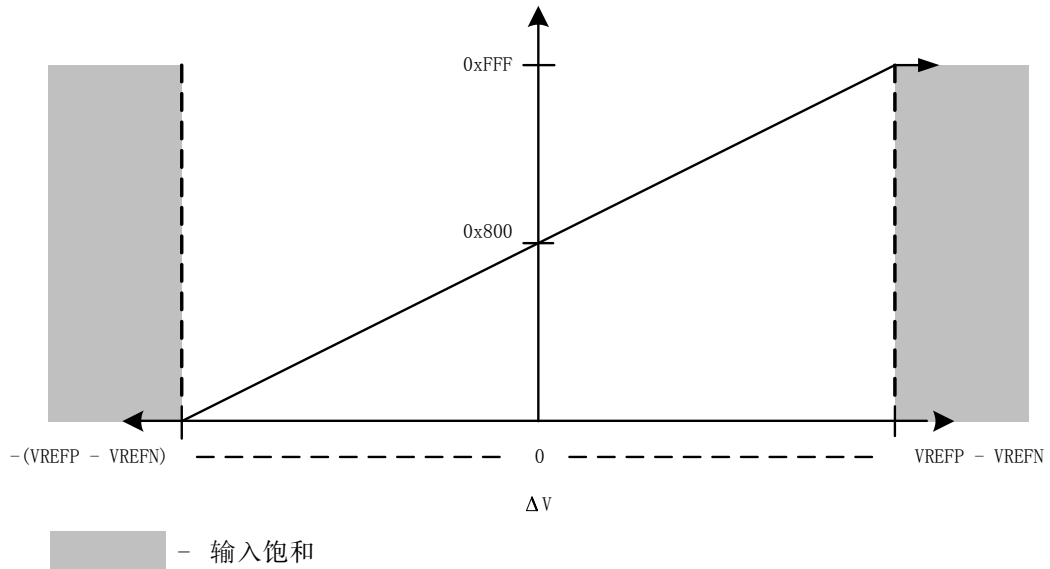
如果  $V_{IN_{CM}}$  不等于  $V_{REF_{CM}}$ ，那么在最大或最小电压下，差分输入信号可能减弱，(这是因为任何单端输入都不能大于  $V_{REFP}$  或小于  $V_{REFN}$ )，而且无法实现全摆幅。因此输入电压和参考电压之间的任何共模差异都会限制 ADC 的差分动态范围。

由于最大的峰峰差分信号电压为  $2 * (V_{REFP} - V_{REFN})$ ，ADC 读数表示为：

$$\text{mV per ADC code} = (2 * (V_{REFP} - V_{REFN})) / 4096$$

图 13-10 显示如何通过 ADC 读数来表示差分电压  $\Delta V$ 。

图 13-10. 差分电压表达式



### 13.3.6 内部温度传感器

温度传感器的主要作用有两个：1) 内部温度过高或过低时，向系统给予提示。2) 提供温度测试方法来校准休眠模块 RTC 调整值。

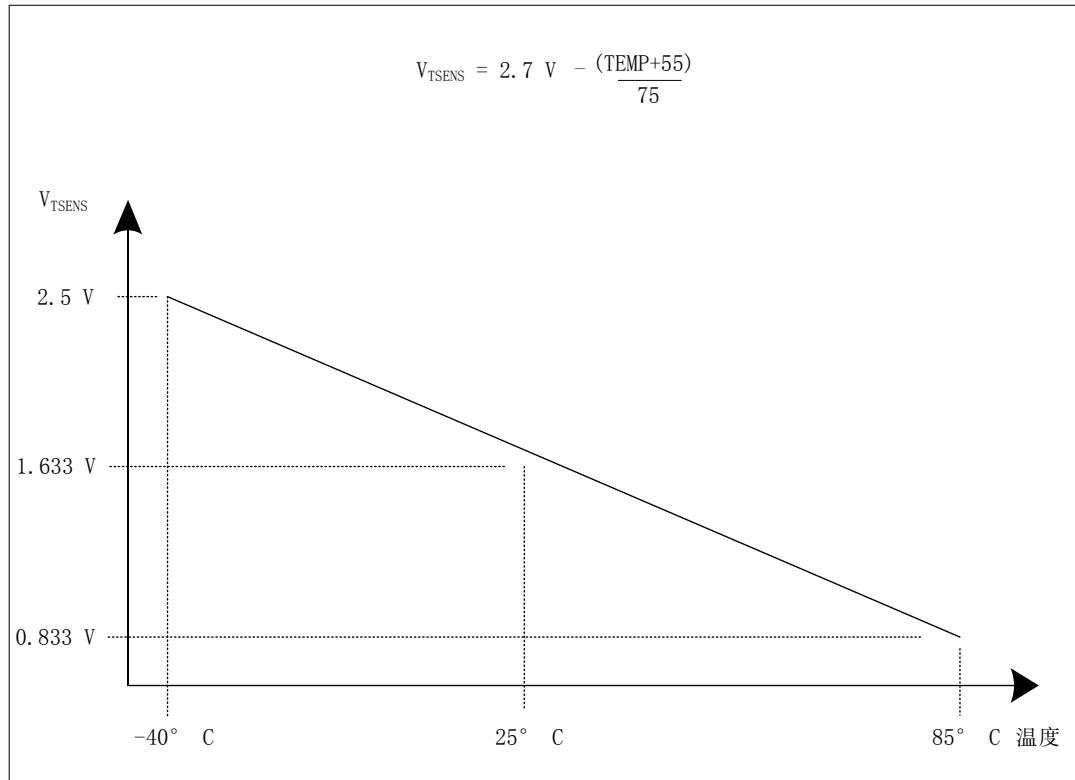
温度传感器没有单独的启用/禁用操作，因为它还关系到带隙参考电压的产生，必须始终启用。该参考电压不仅提供给 ADC 模块，还需要提供给其它所有模拟模块。此外，该温度传感器还带有 3.3 V 的辅助掉电输入功能，可以由休眠模块进行控制。

内部温度传感器将温度测量值转换为电压。此电压值  $V_{TSENS}$  可以通过以下公式得出（其中 TEMP 指温度，单位  $^{\circ}C$ ）：

$$V_{TSENS} = 2.7 - ((TEMP + 55) / 75)$$

这种关系在图 13-11 在 755 页中表示出来。

图 13-11. 内部温度传感器特性



通过将 ADCSSCTLn 中的 TSn 位置位，即可在采样队列中得到温度感应器的读数。也可以从温度传感器的 ADC 结果通过函数转换得到温度读数。以下公式根据 ADC 读数 (ADC<sub>CODE</sub>，定义为 0 至 4095 的一个不带正负号的十进制数) 和最大的 ADC 电压范围 (VREFP - VREFN) 计算温度 (TEMP，单位 °C)：

$$TEMP = 147.5 - ((75 * (VREFP - VREFN) * ADC_{CODE}) / 4096)$$

### 13.3.7 数字比较器

ADC 通常用于对外部信号采样并监控其数值的变动，确保其保持在给定的范围内。为了实现此监控过程的自动化、减少所需的处理器开销，ADC 模块内置有每个模块提供 8 个数字比较器。

ADC 转换结果可直接发送给数字比较器，与用户编程的门限进行比较。门限通过 ADC 数字比较器范围寄存器 (ADCDCMPn) 配置。ADC 可配置为根据 ADC 是在低值带、中值带还是高值带 (可在 ADCDCMPn 位域进行配置) 运行而生成中断。另外可将数字比较器的 4 种工作模式 (单次触发，持续触发，迟滞单次触发，迟滞持续触发) 应用于中断配置。

#### 13.3.7.1 输出功能

取决于 ADC 采样序列工作寄存器 n (ADCCSOPn) 中 SnDCOP 位的设置，ADC 转换结果可以保存到 ADC 采样序列 FIFO 中，也可以供给数字比较器进行比较。选定的 ADC 转换结果将被其对应的数字比较器用于监控外部信号。每个数字比较器可以有两种输出功能：处理器中断或 PWM 触发事件。

每种输出功能都有其状态机对被监控的信号实施追踪。中断功能和触发事件功能既可以分别启用，也可以同时启用；两种功能将根据同一转换数据判断其条件是否已经满足、并据此产生相应的输出。

## 中断

将 ADC 数字比较器控制寄存器 (ADCDCCTLn) 的 CIE 位置位即可启用数字比较器的中断功能。此时中断功能状态机开始运行，并监控输入的 ADC 转换结果。当某组条件得到满足并且 ADCIM 寄存器的 DCONSSx 位置位时，将向中断控制器发送一个中断。

## 触发事件

将 ADCDCCTLn 寄存器中的 CTE 位置位就可以启用数字比较器触发功能。此时触发功能状态机开始运行，并监控输入的 ADC 转换结果。当某组条件得到满足时，将相应产生一个数字比较器触发事件并发送给 PWM 模块。

### 13.3.7.2 工作模式

数字比较器有 4 种工作模式，能够支持类型广泛的应用、满足各种信号的要求。这 4 种工作模式分别是：持续触发、单次触发、迟滞持续触发、迟滞单次触发。工作模式通过 ADCDCCTLn 寄存器的 CIM 或 CTM 域选取。

#### 持续触发模式

在持续触发工作模式中，只要 ADC 转换值满足比较条件即会产生相应的中断或触发事件。因此，如果 A/D 转换结果处于规定的范围内，将产生一连串的中断或触发事件。

#### 单次触发模式

在单次触发工作模式中，只有当前 ADC 转换值满足比较条件并且前一个 ADC 转换值不满足比较条件时，才会产生相应的中断或触发事件。因此，如果 A/D 转换结果处于规定的范围内，将产生单个中断或触发事件。

#### 迟滞持续触发模式

迟滞持续触发工作模式只能结合低值带或高值带工作，只有跨越中值带进入相反的区域时才会清除迟滞条件。在迟滞持续触发工作模式中，满足以下条件时才会产生相应的中断或触发事件：ADC 转换值满足其比较条件，或之前的某个 ADC 结果满足比较条件，并且迟滞条件尚未清除（ADC 转换值尚未落入相反的区域）。因此，在 ADC 转换值进入相反的区域之前，将不断产生一连串的中断或触发事件。

#### 迟滞单次触发模式

迟滞单次触发工作模式只能结合低值带或高值带工作，只有跨越中值带进入相反的区域时才会清除迟滞条件。在迟滞单次触发工作模式中，满足以下条件时才会产生相应的中断或触发事件：ADC 转换值满足其比较条件，且前一个 ADC 转换值不满足比较条件，且迟滞条件已清除。因此将产生单个中断或触发事件。

### 13.3.7.3 功能作用范围

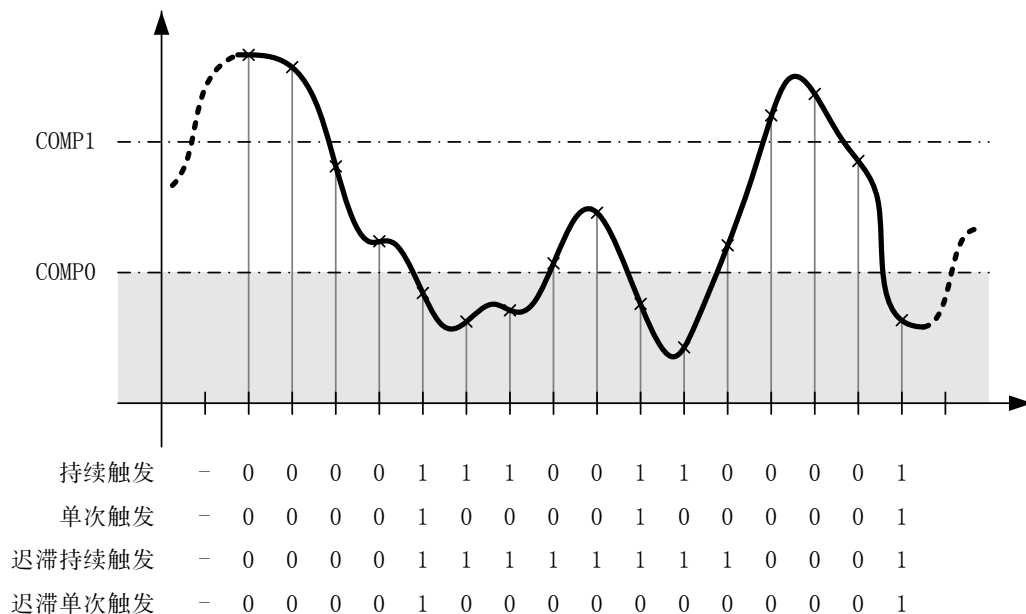
ADC 数字比较器范围 (ADCDCCMPn) 寄存器中的两组比较门限 COMP0 和 COMP1 可将转换结果有效划分为 3 个不同区域。这些区域分别称为低值带（小于等于 COMP0）、中值带（大于 COMP0 但小于等于 COMP1）以及高值带（大于 COMP1）。允许将 COMP0 和 COMP1 编程为相等的值，也就是只划分出两个区域。但是，COMP1 必须大于或等于 COMP0。若 COMP1 小于 COMP0，其后果将难以预料。

#### 低值带工作

要让数字比较器在低值带内工作，必须将 ADCDCCTLn 寄存器的 CIC 域或 CTC 域设为 0x0。此设置会在低值带内按照编程的工作模式产生中断或触发事件。图 13-12 在 757 页中显示了在高值带内各种工作模式下产生中断/触发事件的状态示例。其中每行代表一种工作模式（持续触发、单次触

发、迟滞持续触发、迟滞单次触发)，每列中的“0”表示不产生中断或触发事件信号，“1”表示产生中断或触发事件信号。

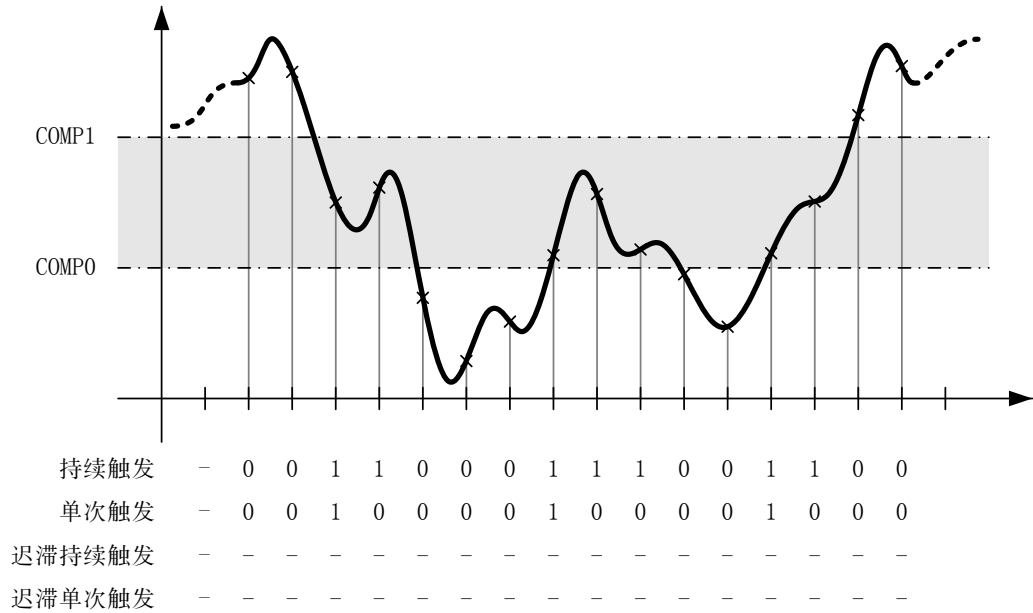
图 13-12. 低值带工作 (CIC = 0x0 和/或 CTC = 0x0)



### 中值带工作

要让数字比较器在中值带内工作，必须将 ADCDCCTLn 寄存器的 CIC 域或 CTC 域设为 0x1。此设置会在中值带内按照工作模式产生中断或触发事件。只有持续触发工作模式和单次触发工作模式能够在中值带内工作。图 13-13 在 758 页中显示了在中值带内各种允许的工作模式下产生中断/触发信号的状态示例。工作模式名（持续触发、单次触发）后每列中的“0”表示不产生中断或触发事件信号，“1”表示产生中断或触发事件信号。

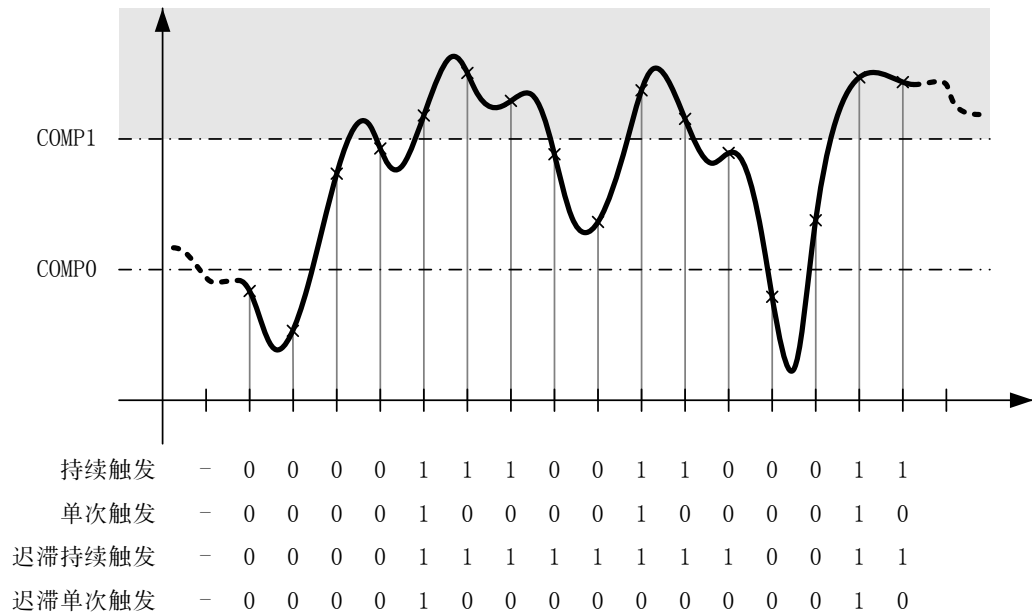
图 13-13. 中值带工作 (CIC = 0x1 和/或 CTC = 0x1)



### 高值带工作

要让数字比较器在高值带内工作，必须将 ADCDCCTLn 寄存器的 CIC 域或 CTC 域设为 0x3。此设置会在高值带内按照工作模式产生中断或触发事件。图 13-14 在 759 页中显示了在高值带内各种允许的工作模式下产生中断/触发信号的状态示例。其中每行代表一种工作模式（持续触发、单次触发、迟滞持续触发、迟滞单次触发），每列中的“0”表示不产生中断或触发事件信号，“1”表示产生中断或触发事件信号。

图 13-14. 高值带工作 (CIC = 0x3 和/或 CTC = 0x3)



## 13.4 初始化及配置

要想正常使用 ADC 模块，必须通过 RCC 寄存器（见第 236 页）启用 PLL，并且将工作频率编程为 ADC 模块支持的数值。采用不支持的频率有可能造成 ADC 模块的工作发生错误。

### 13.4.1 模块初始化

ADC 模块的初始化流程比较简单，只有很少几个步骤：启用 ADC 的时钟、禁用待用模拟输入脚的模拟隔离电路、配置采样序列发生器优先级（如果有必要的话）。

ADC 初始化的顺序如下：

1. 使用 RCGCADC 寄存器启用 ADC 时钟（参见 315 页）。
2. 通过 RCGCGPIO 寄存器（见 302 页）启用相应 GPIO 模块的时钟。欲了解需要启用哪些 GPIO 端口，请参阅“信号描述”在 744 页。
3. 将 ADC 输入管脚的 AFSEL 位置位（参见 618 页）。欲了解需要配置哪些 GPIO，请参阅表 23-4 在 1271 页。
4. 通过将 GPIO 数字使能 (GPIODEN) 寄存器中相应 DEN 位清零，将 AINx 管脚配置为模拟输出（参考 629 页）。
5. 通过为 GPIOAMSEL 寄存器（参见 634 页）的相应位写 1，禁用待用模拟输入脚的模拟隔离电路。
6. 假如应用有相关需求，则应通过 ADCSSPRI 寄存器配置采样序列发生器的优先级。默认配置是采样序列发生器 0 的优先级最高，采样序列发生器 3 的优先级最低。

### 13.4.2 采样序列发生器的配置

与模块的初始化流程相比，采样序列发生器的配置稍微复杂一些，这大概是因为每个采样序列发生器都是完全可编程的。

每个采样序列发生器的配置步骤应如下：

1. 将ADCACTSS寄存器的ASENn位清零，禁用采样序列发生器。采样序列器不用启用也可以进行配置。不过如果在配置期间禁用采样序列发生器，可以有效防止在此期间因满足触发条件而造成的误执行；
2. 通过ADCEMUX寄存器配置采样序列发生器的触发事件；
3. 在ADCSSMUXn和ADCSEMUXn寄存器中为采样序列的每个采样配置相应的输入源。
4. 通过ADCSSCTLn寄存器中的相应半字节来为采样序列中每个采样动作配置控制位。在配置最后一个采样动作的半字节时，应确保END位置位。如果未设置END位将导致不可预测的执行结果；
5. 假如要使用中断，则应在ADCIM寄存器中设置相应的MASK位；
6. 将ADCACTSS寄存器的ASENn位置位，启用采样序列发生器逻辑单元。

## 13.5 寄存器映射

表 13-4 在 760页 列出了所有 ADC 寄存器。表中偏移量一列是指相对于ADC模块基地址的十六进制地址增量，两个ADC模块的基地址分别为：

- ADC0 : 0x4003.8000
- ADC1 : 0x4003.9000

注意配置这些寄存器之前必须启用 ADC 模块的时钟（请参考 315页）。ADC 模块时钟启用后，必须等待至少 3 个系统时钟才可访问 ADC 模块寄存器。

表 13-4. ADC 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	ADCACTSS	R/W	0x0000.0000	ADC 有效采样序列发生器寄存器	763
0x004	ADCRIS	RO	0x0000.0000	ADC原始中断状态寄存器	764
0x008	ADCIM	R/W	0x0000.0000	ADC 中断掩码寄存器	766
0x00C	ADCISC	R/W1C	0x0000.0000	ADC 中断状态及清除寄存器	768
0x010	ADCOSTAT	R/W1C	0x0000.0000	ADC 上溢状态寄存器	771
0x014	ADCEMUX	R/W	0x0000.0000	ADC 事件复用选择寄存器	773
0x018	ADCUSTAT	R/W1C	0x0000.0000	ADC 下溢状态寄存器	778
0x01C	ADCTSSEL	R/W	0x0000.0000	ADC 触发源选择寄存器	779
0x020	ADCSSPRI	R/W	0x0000.3210	ADC 采样序列发生器优先级寄存器	781
0x024	ADCSPC	R/W	0x0000.0000	ADC 采样相位控制寄存器	782
0x028	ADCPSSI	R/W	-	ADC 处理器采样序列启动寄存器	783
0x030	ADCSAC	R/W	0x0000.0000	ADC 采样平均控制寄存器	785



表 13-4. ADC 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x034	ADCDCISC	R/W1C	0x0000.0000	ADC 数字比较器中断状态及清除寄存器	786
0x038	ADCCTL	R/W	0x0000.0010	ADC 控制寄存器	788
0x040	ADCSSMUX0	R/W	0x0000.0000	ADC 采样序列输入复用选择寄存器 0	789
0x044	ADCSSCTL0	R/W	0x0000.0000	ADC 采样序列控制寄存器 0	791
0x048	ADCSSFIFO0	RO	-	ADC 采样序列结果 FIFO 寄存器 0	794
0x04C	ADCSSFSTAT0	RO	0x0000.0100	ADC 采样序列 FIFO 0 状态寄存器	795
0x050	ADCSSOP0	R/W	0x0000.0000	ADC 采样序列寄存器 0	797
0x054	ADCSSDC0	R/W	0x0000.0000	ADC 采样序列数字比较器选择寄存器 0	799
0x058	ADCSSEMUX0	R/W	0x0000.0000	ADC 样序列扩展输入复用选择寄存器	801
0x060	ADCSSMUX1	R/W	0x0000.0000	ADC 采样序列输入复用选择寄存器 1	803
0x064	ADCSSCTL1	R/W	0x0000.0000	ADC 采样序列控制寄存器 1	804
0x068	ADCSSFIFO1	RO	-	ADC 采样序列结果 FIFO 寄存器 1	794
0x06C	ADCSSFSTAT1	RO	0x0000.0100	ADC 采样序列 FIFO 1 状态寄存器	795
0x070	ADCSSOP1	R/W	0x0000.0000	ADC 采样序列 1 工作寄存器	806
0x074	ADCSSDC1	R/W	0x0000.0000	ADC 采样序列数字比较器选择寄存器 1	807
0x078	ADCSSEMUX1	R/W	0x0000.0000	ADC 采样序列扩展输入复用选择寄存器 1	808
0x080	ADCSSMUX2	R/W	0x0000.0000	ADC 采样序列输入复用选择寄存器 2	803
0x084	ADCSSCTL2	R/W	0x0000.0000	ADC 采样序列控制寄存器 2	804
0x088	ADCSSFIFO2	RO	-	ADC 采样序列结果 FIFO 寄存器 2	794
0x08C	ADCSSFSTAT2	RO	0x0000.0100	ADC 采样序列 FIFO 2 状态寄存器	795
0x090	ADCSSOP2	R/W	0x0000.0000	ADC 采样序列 2 工作寄存器	806
0x094	ADCSSDC2	R/W	0x0000.0000	ADC 采样序列数字比较器选择寄存器 2	807
0x098	ADCSSEMUX2	R/W	0x0000.0000	ADC 采样序列扩展输入复用选择寄存器 2	808
0x0A0	ADCSSMUX3	R/W	0x0000.0000	ADC 采样序列输入复用选择寄存器 3	810
0x0A4	ADCSSCTL3	R/W	0x0000.0000	ADC 采样序列控制寄存器 3	811
0x0A8	ADCSSFIFO3	RO	-	ADC 采样序列结果 FIFO 寄存器 3	794
0x0AC	(ADCSSFSTAT3	RO	0x0000.0100	ADC 采样序列 FIFO 3 状态寄存器	795
0x0B0	ADCSSOP3	R/W	0x0000.0000	ADC 采样序列器 3 工作寄存器	812
0x0B4	ADCSSDC3	R/W	0x0000.0000	ADC 采样序列 3 数字比较器选择寄存器	813
0x0B8	ADCSSMUX3	R/W	0x0000.0000	ADC 采样序列扩展输入复用选择寄存器 3	814
0xD00	ADCDCRIC	WO	0x0000.0000	ADC 数字比较器复位启动条件寄存器	815
0xE00	ADCDCCTL0	R/W	0x0000.0000	ADC 数字比较器控制寄存器 0	819
0xE04	ADCDCCTL1	R/W	0x0000.0000	ADC 数字比较器控制寄存器 1	819

表 13-4. ADC 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0xE08	ADCDCCTL2	R/W	0x0000.0000	ADC 数字比较器控制寄存器 2	819
0xE0C	ADCDCCTL3	R/W	0x0000.0000	ADC 数字比较器控制寄存器 3	819
0xE10	ADCDCCTL4	R/W	0x0000.0000	ADC 数字比较器控制寄存器 4	819
0xE14	ADCDCCTL5	R/W	0x0000.0000	ADC 数字比较器控制寄存器 5	819
0xE18	ADCDCCTL6	R/W	0x0000.0000	ADC 数字比较器控制寄存器 6	819
0xE1C	ADCDCCTL7	R/W	0x0000.0000	ADC 数字比较器控制寄存器 7	819
0xE40	ADCDCCMP0	R/W	0x0000.0000	ADC 数字比较器范围寄存器 0	822
0xE44	ADCDCCMP1	R/W	0x0000.0000	ADC 数字比较器范围寄存器 1	822
0xE48	ADCDCCMP2	R/W	0x0000.0000	ADC 数字比较器范围寄存器 2	822
0xE4C	ADCDCCMP3	R/W	0x0000.0000	ADC 数字比较器范围寄存器 3	822
0xE50	ADCDCCMP4	R/W	0x0000.0000	ADC 数字比较器范围寄存器 4	822
0xE54	ADCDCCMP5	R/W	0x0000.0000	ADC 数字比较器范围寄存器 5	822
0xE58	ADCDCCMP6	R/W	0x0000.0000	ADC 数字比较器范围寄存器 6	822
0xE5C	ADCDCCMP7	R/W	0x0000.0000	ADC 数字比较器范围寄存器 7	822
0xFC0	ADCPP	RO	0x00B0.2187	ADC 外设属性寄存器	823
0xFC4	ADCPC	R/W	0x0000.0007	ADC 外设配置寄存器	825
0xFC8	ADCCC	R/W	0x0000.0000	ADC 时钟配置寄存器	826

## 13.6 寄存器描述

下文将按照地址偏移量的数字顺序列出 ADC 寄存器，并对这些寄存器进行描述。

## 寄存器 1: ADC 有效采样序列发生器寄存器 ( ADCACTSS ) , 偏移量 0x000

本寄存器控制采样序列发生器是否有效。每个采样序列发生器可分别设置启用或禁用。

### ADC 有效采样序列发生器寄存器 (ADCACTSS)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x000

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												ASEN3	ASEN2	ASEN1	ASEN0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	ASEN3	R/W	0	ADC SS3 启用  值 描述 1 启用采样序列器3。 0 禁用采样序列器3。
2	ASEN2	R/W	0	ADC SS2 启用  值 描述 1 启用采样序列器2。 0 禁用采样序列器2。
1	ASEN1	R/W	0	ADC SS1 启用  值 描述 1 启用采样序列器1。 0 禁用采样序列器1。
0	ASEN0	R/W	0	ADC SS0 启用  值 描述 1 启用采样序列器0。 0 禁用采样序列器0。

## 寄存器 2: ADC原始中断状态寄存器 (ADCRIS), 偏移量 0x004

本寄存器用于指示每个采样序列发生器的原始中断状态。软件可以通过这些标志位查看那些并不向中断控制器发送的中断状态。

## ADC原始中断状态寄存器 (ADCRIS)

ADC0 基址: 0x4003.8000  
ADC1 基址: 0x4003.9000  
偏移量 0x004  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															INRDC
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												INR3	INR2	INR1	INR0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:17	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
16	INRDC	RO	0	数字比较器原始中断状态  值 描述 1 ADCDCISC 寄存器中至少有一个位置位, 即至少产生了一个数字比较器中断。 0 将 ADCDCISC 寄存器的所有位都清零。
15:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	INR3	RO	0	SS3 原始中断状态  值 描述 1 采样已完成了转换, 且 ADCSSCTL3 寄存器中对应的 IEn 位已置位, 将启用原始中断。 0 未产生中断。  向 ADCISC 寄存器的 IN3 位写 1, 可清除本标志位。
2	INR2	RO	0	SS2 原始中断状态  值 描述 1 采样已完成了转换, 且 ADCSSCTL2 寄存器中对应的 IEn 位已置位, 将启用原始中断。 0 未产生中断。  向 ADCISC 寄存器的 IN2 位写 1, 可清除本标志位。

位/域	名称	类型	复位	描述
1	INR1	RO	0	SS1 原始中断状态  值 描述 1 采样已完成了转换，且 ADCSSCTL1 寄存器中对应的 IEn 位已置位，将启用原始中断。 0 未产生中断。  向 ADCISC 寄存器的 IN1 位写 1，可清除本标志位。
0	INR0	RO	0	SS0 原始中断状态  值 描述 1 采样已完成了转换，且 ADCSSCTL0 寄存器中对应的 IEn 位已置位，将启用原始中断。 0 未产生中断。  向 ADCISC 寄存器的 IN0 位写 1，可清除本标志位。

## 寄存器 3: ADC 中断掩码寄存器 (ADCIM) , 偏移量 0x008

本寄存器控制采样序列发生器和数字比较器原始中断信号是否能发送给中断控制器。每个原始中断信号都能分别予以掩码。任何时刻只允许 1 个 DCONSSn 位置位。如果几个 DCONSSn 位同时置位，会导致 ADCRIS 寄存器的 INRDC 位被屏蔽，并且任何采样序列器的中断线都不再产生中断。

## ADC 中断掩码寄存器 (ADCIM)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x008

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留												DCONSS3	DCONSS2	DCONSS1	DCONSS0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												MASK3	MASK2	MASK1	MASK0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:20	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
19	DCONSS3	R/W	0	SS3 数字比较器中断  值 描述 1 数字比较器产生的原始中断信号 ( ADCRIS 寄存器的 INRDC 位 ) 将从 SS3 中断线发送给中断控制器。 0 数字比较器的状态不影响 SS3 中断状态。
18	DCONSS2	R/W	0	SS2 数字比较器中断  值 描述 1 数字比较器产生的原始中断信号 ( ADCRIS 寄存器的 INRDC 位 ) 将从 SS2 中断线发送给中断控制器。 0 数字比较器的状态不影响 SS2 中断状态。
17	DCONSS1	R/W	0	SS1 数字比较器中断  值 描述 1 数字比较器产生的原始中断信号 ( ADCRIS 寄存器的 INRDC 位 ) 将从 SS1 中断线发送给中断控制器。 0 数字比较器的状态不影响 SS1 中断状态。
16	DCONSS0	R/W	0	SS0 数字比较器中断  值 描述 1 数字比较器产生的原始中断信号 ( ADCRIS 寄存器的 INRDC 位 ) 将从 SS0 中断线发送给中断控制器。 0 数字比较器的状态不影响 SS0 中断状态。

位/域	名称	类型	复位	描述
15:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
3	MASK3	R/W	0	SS3 中断掩码  值 描述 1 采样序列器 3 的原始中断信号 ( ADCRIS 寄存器的 INR3 位 ) 将发送给中断控制器。 0 采样序列器 3 的状态不影响 SS3 中断状态。
2	MASK2	R/W	0	SS2 中断掩码  值 描述 1 采样序列器 2 的原始中断信号 ( ADCRIS 寄存器的 INR2 位 ) 将发送给中断控制器。 0 采样序列器 2 的状态不影响 SS2 中断状态。
1	MASK1	R/W	0	SS1 中断掩码  值 描述 1 采样序列器 1 的原始中断信号 ( ADCRIS 寄存器的 INR1 位 ) 将发送给中断控制器。 0 采样序列器 1 的状态不影响 SS1 中断状态
0	MASK0	R/W	0	SS0 中断掩码  值 描述 1 采样序列器 0 的原始中断信号 ( ADCRIS 寄存器的 INR0 位 ) 将发送给中断控制器。 0 采样序列器 0 的状态不影响 SS0 中断状态。

## 寄存器 4: ADC 中断状态及清除寄存器 (ADCISC), 偏移量 0x00C

本寄存器能够显示出由采样序列发生器以及数字比较器以及数字比较器产生、已经发送给中断控制器的中断状态，并提供清除采样序列发生器中断状态的机制。读本寄存器时，每个位域实际上是相应的 INR 位和 MASK 位的逻辑与结果。向采样序列器中断位写 1，可清除相应的中断。请注意，数字比较器中断不是通过本寄存器清除的，而是通过向 ADCDCISC 寄存器的对应位写 1 来清除的。假如软件以轮询 ADCRIS 寄存器的方式（而不是通过中断方式）处理事件，那么即使 INn 位并未置位也能通过 ADCISC 寄存器来清除 INRn 位。

## ADC 中断状态及清除寄存器 (ADCISC)

ADC0 基址: 0x4003.8000  
 ADC1 基址: 0x4003.9000  
 偏移量 0x00C  
 类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留												DCINSS3	DCINSS2	DCINSS1	DCINSS0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												IN3	IN2	IN1	IN0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:20	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
19	DCINSS3	RO	0	SS3 数字比较器的中断状态  值 描述 1 如果 ADCRIS 寄存器的 INRDC 位和 ADCIM 寄存器的 DCONSS3 位都置位，就会向中断控制器产生一个基于电平的中断。 0 未发生中断，或该中断被屏蔽。  该位写 1 清零。如果将此位清零，则 ADCRIS 寄存器中的 INRDC 位也会被清零。
18	DCINSS2	RO	0	SS2 数字比较器的中断状态  值 描述 1 如果 ADCRIS 寄存器的 INRDC 位和 ADCIM 寄存器的 DCONSS2 位都置位，就会向中断控制器产生一个基于电平的中断。 0 未发生中断，或该中断被屏蔽。  该位写 1 清零。如果将此位清零，则 ADCRIS 寄存器中的 INRDC 位也会被清零。



位/域	名称	类型	复位	描述
17	DCINSS1	RO	0	<p>SS1 数字比较器的中断状态</p> <p>值 描述</p> <p>1 如果 ADCRIS 寄存器的 INRDC 位和 ADCIM 寄存器的 DCONSS1 位都置位，就会向中断控制器产生一个基于电平的中断。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>该位写 1 清零。如果将此位清零，则 ADCRIS 寄存器中的 INRDC 位也会被清零。</p>
16	DCINSS0	RO	0	<p>SS0 数字比较器中断状态</p> <p>值 描述</p> <p>1 如果 ADCRIS 寄存器的 INRDC 位和 ADCIM 寄存器的 DCONSS0 位都置位，就会向中断控制器产生一个基于电平的中断。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>该位写 1 清零。如果将此位清零，则 ADCRIS 寄存器中的 INRDC 位也会被清零。</p>
15:4	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
3	IN3	R/W1C	0	<p>SS3 中断状态及清除</p> <p>值 描述</p> <p>1 如果 ADCRIS 寄存器的 INR3 位和 ADCIM 寄存器的 MASK3 位都置位，就会向中断控制器产生一个基于电平的中断。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>对该位写入 1，即可将其清零。如果将此位清零，则寄存器 ADCRIS 中的 INR3 位也会被清零。</p>
2	IN2	R/W1C	0	<p>SS2 中断状态及清除</p> <p>值 描述</p> <p>1 如果 ADCRIS 寄存器的 INR2 位和 ADCIM 寄存器的 MASK2 位都置位，就会向中断控制器产生一个基于电平的中断。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>对该位写入 1，即可将其清零。如果将此位清零，则寄存器 ADCRIS 中的 INR2 位也会被清零。</p>
1	IN1	R/W1C	0	<p>SS1 中断状态及清除</p> <p>值 描述</p> <p>1 如果 ADCRIS 寄存器的 INR1 位和 ADCIM 寄存器的 MASK1 位都置位，就会向中断控制器产生一个基于电平的中断。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>对该位写入 1，即可将其清零。如果将此位清零，则寄存器 ADCRIS 中的 INR1 位也会被清零。</p>

---

位/域	名称	类型	复位	描述
0	IN0	R/W1C	0	SS0 中断状态及清除  值 描述 1 如果 ADCRIS 寄存器的 INR0 位和 ADCIM 寄存器的 MASK0 位都置位，就会向中断控制器产生一个基于电平的中断。 0 未发生中断，或该中断被屏蔽。  对该位写入 1，即可将其清零。如果将此位清零，则寄存器 ADCRIS 中的 INR0 位也会被清零。

## 寄存器 5: ADC 上溢状态寄存器 ( ADCOSTAT ) , 偏移量 0x010

本寄存器用于指示采样序列发生器FIFO的上溢状况。当软件处理完FIFO上溢状况后，可向相应的位写1予以清除。

### ADC 上溢状态寄存器 (ADCOSTAT)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x010

类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													OV3	OV2	OV1	OV0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	OV3	R/W1C	0	SS3 FIFO上溢  值 描述 1 采样序列发生器 3 的 FIFO 达到了上溢条件，也就是说 FIFO 满且要求进行写操作。当 FIFO 出现上溢后，最近一次的写操作会丢失。 0 FIFO 未出现上溢。  对该位写入 1，即可将其清零。
2	OV2	R/W1C	0	SS2 FIFO上溢  值 描述 1 采样序列发生器 2 的 FIFO 达到了上溢条件，也就是说 FIFO 满且要求进行写操作。当 FIFO 出现上溢后，最近一次的写操作会丢失。 0 FIFO 未出现上溢。  对该位写入 1，即可将其清零。
1	OV1	R/W1C	0	SS1 FIFO上溢  值 描述 1 采样序列发生器 1 的 FIFO 达到了上溢条件，也就是说 FIFO 满且要求进行写操作。当 FIFO 出现上溢后，最近一次的写操作会丢失。 0 FIFO 未出现上溢。  对该位写入 1，即可将其清零。

位/域	名称	类型	复位	描述
0	OV0	R/W1C	0	SS0 FIFO上溢  值 描述 1 采样序列发生器 0 的 FIFO 达到了上溢条件，也就是说 FIFO 满且要求进行写操作。当 FIFO 出现上溢后，最近一次的写操作会丢失。 0 FIFO 未出现上溢。  对该位写入 1，即可将其清零。

## 寄存器 6: ADC 事件复用选择寄存器 ( ADCEMUX ) , 偏移量 0x014

ADCEMUX 寄存器用于选择每个采样序列启动采样的触发条件。每个采样序列器可配置为使用唯一的触发源。当使用 PWM 发生器作为触发源时，要使用 ADC 触发源选择 (ADCTSSEL) 寄存器来指定该 PWM 发生器在哪个 PWM 模块中。

### ADC 事件复用选择寄存器 (ADCEMUX)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x014

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EM3				EM2				EM1				EM0			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述																										
15:12	EM3	R/W	0x0	<p>SS3触发条件选择</p> <p>此位域选择采样序列发生器 3 的触发源。</p> <p>此位域的有效配置包括：</p> <table border="1"> <thead> <tr> <th>值</th> <th>事件</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>           处理器 (默认)            将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。         </td> </tr> <tr> <td>0x1</td> <td>           模拟比较器 0            该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x2</td> <td>           模拟比较器 1            该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x3</td> <td>           模拟比较器 2            该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x4</td> <td>           外部 (GPIO 管脚)            该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。         </td> </tr> <tr> <td>0x5</td> <td>           定时器            此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。         </td> </tr> <tr> <td>0x6</td> <td>           PWM0            使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x7</td> <td>           PWM1            使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x8</td> <td>           PWM2            使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x9</td> <td>           PWM3            使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0xA-0xE</td> <td>保留</td> </tr> <tr> <td>0xF</td> <td>持续 (始终采样)</td> </tr> </tbody> </table>	值	事件	0x0	处理器 (默认) 将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。	0x1	模拟比较器 0 该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。	0x2	模拟比较器 1 该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。	0x3	模拟比较器 2 该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。	0x4	外部 (GPIO 管脚) 该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。	0x5	定时器 此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。	0x6	PWM0 使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。	0x7	PWM1 使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。	0x8	PWM2 使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。	0x9	PWM3 使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。	0xA-0xE	保留	0xF	持续 (始终采样)
值	事件																													
0x0	处理器 (默认) 将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。																													
0x1	模拟比较器 0 该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。																													
0x2	模拟比较器 1 该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。																													
0x3	模拟比较器 2 该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。																													
0x4	外部 (GPIO 管脚) 该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。																													
0x5	定时器 此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。																													
0x6	PWM0 使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。																													
0x7	PWM1 使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。																													
0x8	PWM2 使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。																													
0x9	PWM3 使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。																													
0xA-0xE	保留																													
0xF	持续 (始终采样)																													

位/域	名称	类型	复位	描述																																
11:8	EM2	R/W	0x0	<p>SS2 触发器选择</p> <p>此位域选择采样序列发生器 2 的触发源。</p> <p>此位域的有效配置包括：</p> <table border="1"> <thead> <tr> <th>值</th> <th>事件</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>           处理器 (默认)            将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。         </td> </tr> <tr> <td>0x1</td> <td>           模拟比较器 0            该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x2</td> <td>           模拟比较器 1            该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x3</td> <td>           模拟比较器 2            该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x4</td> <td>           外部 (GPIO 管脚)            该触发器连接到相应 GPIO 的 GPIO 中断上 (参考“ADC 触发源”在 604页)。         </td> </tr> <tr> <td>0x5</td> <td>           定时器            此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。         </td> </tr> <tr> <td>0x6</td> <td>           PWM0            使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x7</td> <td>           PWM1            使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x8</td> <td>           PWM2            使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x9</td> <td>           PWM3            使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。         </td> </tr> <tr> <td></td> <td>0xA-0xE</td> <td>保留</td> <td></td> <td></td> </tr> <tr> <td></td> <td>0xF</td> <td>持续 (始终采样)</td> <td></td> <td></td> </tr> </tbody> </table>	值	事件	0x0	处理器 (默认) 将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。	0x1	模拟比较器 0 该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。	0x2	模拟比较器 1 该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。	0x3	模拟比较器 2 该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。	0x4	外部 (GPIO 管脚) 该触发器连接到相应 GPIO 的 GPIO 中断上 (参考“ADC 触发源”在 604页)。	0x5	定时器 此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。	0x6	PWM0 使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。	0x7	PWM1 使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。	0x8	PWM2 使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。	0x9	PWM3 使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。		0xA-0xE	保留				0xF	持续 (始终采样)		
值	事件																																			
0x0	处理器 (默认) 将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。																																			
0x1	模拟比较器 0 该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。																																			
0x2	模拟比较器 1 该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。																																			
0x3	模拟比较器 2 该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。																																			
0x4	外部 (GPIO 管脚) 该触发器连接到相应 GPIO 的 GPIO 中断上 (参考“ADC 触发源”在 604页)。																																			
0x5	定时器 此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。																																			
0x6	PWM0 使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。																																			
0x7	PWM1 使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。																																			
0x8	PWM2 使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。																																			
0x9	PWM3 使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。																																			
	0xA-0xE	保留																																		
	0xF	持续 (始终采样)																																		

位/域	名称	类型	复位	描述																										
7:4	EM1	R/W	0x0	<p>SS1 触发器选择</p> <p>此位域选择采样序列发生器 1 的触发源。</p> <p>此位域的有效配置包括：</p> <table border="1"> <thead> <tr> <th>值</th> <th>事件</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>           处理器 (默认)            将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。         </td> </tr> <tr> <td>0x1</td> <td>           模拟比较器 0            该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x2</td> <td>           模拟比较器 1            该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x3</td> <td>           模拟比较器 2            该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x4</td> <td>           外部 (GPIO 管脚)            该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。         </td> </tr> <tr> <td>0x5</td> <td>           定时器            此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。         </td> </tr> <tr> <td>0x6</td> <td>           PWM0            使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x7</td> <td>           PWM1            使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x8</td> <td>           PWM2            使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x9</td> <td>           PWM3            使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0xA-0xE</td> <td>保留</td> </tr> <tr> <td>0xF</td> <td>持续 (始终采样)</td> </tr> </tbody> </table>	值	事件	0x0	处理器 (默认) 将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。	0x1	模拟比较器 0 该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。	0x2	模拟比较器 1 该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。	0x3	模拟比较器 2 该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。	0x4	外部 (GPIO 管脚) 该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。	0x5	定时器 此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。	0x6	PWM0 使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。	0x7	PWM1 使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。	0x8	PWM2 使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。	0x9	PWM3 使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。	0xA-0xE	保留	0xF	持续 (始终采样)
值	事件																													
0x0	处理器 (默认) 将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。																													
0x1	模拟比较器 0 该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。																													
0x2	模拟比较器 1 该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。																													
0x3	模拟比较器 2 该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。																													
0x4	外部 (GPIO 管脚) 该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。																													
0x5	定时器 此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。																													
0x6	PWM0 使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。																													
0x7	PWM1 使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。																													
0x8	PWM2 使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。																													
0x9	PWM3 使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。																													
0xA-0xE	保留																													
0xF	持续 (始终采样)																													



位/域	名称	类型	复位	描述																																
3:0	EM0	R/W	0x0	<p>SS0 触发器选择</p> <p>该字段选择采样序列发生器0的触发源</p> <p>此位域的有效配置包括：</p> <table border="1"> <thead> <tr> <th>值</th> <th>事件</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>           处理器 (默认)            将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。         </td> </tr> <tr> <td>0x1</td> <td>           模拟比较器 0            该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x2</td> <td>           模拟比较器 1            该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x3</td> <td>           模拟比较器 2            该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。         </td> </tr> <tr> <td>0x4</td> <td>           外部 (GPIO 管脚)            该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。         </td> </tr> <tr> <td>0x5</td> <td>           定时器            此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。         </td> </tr> <tr> <td>0x6</td> <td>           PWM0            使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x7</td> <td>           PWM1            使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x8</td> <td>           PWM2            使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。         </td> </tr> <tr> <td>0x9</td> <td>           PWM3            使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。         </td> </tr> <tr> <td></td> <td>0xA-0xE</td> <td></td> <td></td> <td>保留</td> </tr> <tr> <td></td> <td>0xF</td> <td></td> <td></td> <td>持续 (始终采样)</td> </tr> </tbody> </table>	值	事件	0x0	处理器 (默认) 将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。	0x1	模拟比较器 0 该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。	0x2	模拟比较器 1 该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。	0x3	模拟比较器 2 该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。	0x4	外部 (GPIO 管脚) 该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。	0x5	定时器 此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。	0x6	PWM0 使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。	0x7	PWM1 使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。	0x8	PWM2 使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。	0x9	PWM3 使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。		0xA-0xE			保留		0xF			持续 (始终采样)
值	事件																																			
0x0	处理器 (默认) 将 ADCPSSI 寄存器中的 SSn 位置位, 启动该触发器。																																			
0x1	模拟比较器 0 该触发器由模拟比较器控制 0 (ACCTL0) 寄存器配置 (参考 1147页)。																																			
0x2	模拟比较器 1 该触发器由模拟比较器控制 1 (ACCTL1) 寄存器配置 (参考 1147页)。																																			
0x3	模拟比较器 2 该触发器由模拟比较器控制 2 (ACCTL2) 寄存器配置 (参考 1147页)。																																			
0x4	外部 (GPIO 管脚) 该触发器连接到相应 GPIO 的 GPIO 中断上 (参考 "ADC 触发源" 在 604页)。																																			
0x5	定时器 此外, 还必须通过 GPTMCTL 寄存器的 TnOTE 位来启用该触发器 (参考 682页)。																																			
0x6	PWM0 使用 PWM0 中断和触发使能 (PWM0INTEN) 寄存器来配置 PWM 发生器 0 触发器 (参考 1190页)。																																			
0x7	PWM1 使用 PWM1INTEN 寄存器来配置 PWM 发生器 1 触发器 (参考 1190页)。																																			
0x8	PWM2 使用 PWM2INTEN 寄存器来配置 PWM 发生器 2 触发器 (参考 1190页)。																																			
0x9	PWM3 使用 PWM3INTEN 寄存器来配置 PWM 发生器 3 触发器 (参考 1190页)。																																			
	0xA-0xE			保留																																
	0xF			持续 (始终采样)																																

## 寄存器 7: ADC 下溢状态寄存器 ( ADCUSTAT ) , 偏移量 0x018

本寄存器用于指示采样序列发生器FIFO的下溢状况。当软件处理完FIFO下溢状况后，可向相应的位写1予以清除。

## ADC 下溢状态寄存器 (ADCUSTAT)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x018

类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												UV3	UV2	UV1	UV0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	UV3	R/W1C	0	SS3 FIFO下溢 此位域的有效配置如下所示。对该位写入 1，即可将其清零。  值 描述 1 采样序列发生器的 FIFO 达到了下溢条件，也就是说 FIFO 空且要求进行读操作。此次读操作并不会移动 FIFO 指针，返回值为 0。 0 FIFO未出现下溢。
2	UV2	R/W1C	0	SS2 FIFO下溢 此域的有效配置与对 UV3 域的配置相同。对该位写入 1，即可将其清零。
1	UV1	R/W1C	0	SS1 FIFO 下溢 此域的有效配置与对 UV3 域的配置相同。对该位写入 1，即可将其清零。
0	UV0	R/W1C	0	SS0 FIFO 下溢 此域的有效配置与对 UV3 域的配置相同。对该位写入 1，即可将其清零。

## 寄存器 8: ADC 触发源选择寄存器 ( ADCTSSEL ) , 偏移量 0x01C

ADCTSSEL 寄存器通过 ADC 事件多路复用器选择 (ADCEMUX) 寄存器中的 EMn 位来具体指定选择哪一个 PWM 模块。该寄存器复位为 0x0000.0000 , 并为所有发生器选择 PWM 模块 0。

### ADC 触发源选择寄存器 (ADCTSSEL)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x01C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留		PS3		保留						PS2		保留			
类型	RO	RO	R/W	R/W	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		PS1		保留						PS0		保留			
类型	RO	RO	R/W	R/W	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述								
31:30	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。								
29:28	PS3	R/W	0x0	EM3 PWM 发生器选择 该域用于选择在 ADCEMUX 寄存器的 EM3 位域的 PWM0-PWM3 编码中使用哪些 PWM 发生器管脚。  <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>PWM 发生器 0</td> </tr> <tr> <td>0x1</td> <td>PWM 发生器 1</td> </tr> <tr> <td>0x2 - 0x3</td> <td>保留</td> </tr> </tbody> </table>	值	描述	0x0	PWM 发生器 0	0x1	PWM 发生器 1	0x2 - 0x3	保留
值	描述											
0x0	PWM 发生器 0											
0x1	PWM 发生器 1											
0x2 - 0x3	保留											
27:22	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。								
21:20	PS2	R/W	0x0	EM2 PWM 发生器选择 该域用于选择在 ADCEMUX 寄存器的 EM2 位域的 PWM0-PWM3 编码中使用哪些 PWM 发生器管脚。  <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>PWM 发生器 0</td> </tr> <tr> <td>0x1</td> <td>PWM 发生器 1</td> </tr> <tr> <td>0x2 - 0x3</td> <td>保留</td> </tr> </tbody> </table>	值	描述	0x0	PWM 发生器 0	0x1	PWM 发生器 1	0x2 - 0x3	保留
值	描述											
0x0	PWM 发生器 0											
0x1	PWM 发生器 1											
0x2 - 0x3	保留											
19:14	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。								

位/域	名称	类型	复位	描述								
13:12	PS1	R/W	0x0	<p>EM1 PWM 发生器选择</p> <p>该域用于选择在 ADCEMUX 寄存器的 EM1 位域的 PWM0-PWM3 编码中使用哪些 PWM 发生器管脚。</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>PWM 发生器 0</td> </tr> <tr> <td>0x1</td> <td>PWM 发生器 1</td> </tr> <tr> <td>0x2 - 0x3</td> <td>保留</td> </tr> </tbody> </table>	值	描述	0x0	PWM 发生器 0	0x1	PWM 发生器 1	0x2 - 0x3	保留
值	描述											
0x0	PWM 发生器 0											
0x1	PWM 发生器 1											
0x2 - 0x3	保留											
11:6	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。								
5:4	PS0	R/W	0x0	<p>EM0 PWM 发生器选择</p> <p>该域用于选择在 ADCEMUX 寄存器的 EM0 位域的 PWM0-PWM3 编码中使用哪些 PWM 发生器管脚。</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>PWM 发生器 0</td> </tr> <tr> <td>0x1</td> <td>PWM 发生器 1</td> </tr> <tr> <td>0x2 - 0x3</td> <td>保留</td> </tr> </tbody> </table>	值	描述	0x0	PWM 发生器 0	0x1	PWM 发生器 1	0x2 - 0x3	保留
值	描述											
0x0	PWM 发生器 0											
0x1	PWM 发生器 1											
0x2 - 0x3	保留											
3:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。								

## 寄存器 9: ADC 采样序列发生器优先级寄存器 (ADCSSPRI), 偏移量 0x020

本寄存器用于设置各采样序列发生器的优先级。复位后默认情况下采样序列发生器0具有最高优先级、采样序列发生器3具有最低优先级。修改优先级时务必确保每个采样序列发生器具有唯一的优先级, 否则ADC可能无法正常工作。

## ADC 采样序列发生器优先级寄存器 (ADCSSPRI)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x020

类型 R/W, 复位 0x0000.3210

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		SS3		保留		SS2		保留		SS1		保留		SS0	
类型	RO	RO	R/W	R/W	RO	RO	R/W	R/W	RO	RO	R/W	R/W	RO	RO	R/W	R/W
复位	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
13:12	SS3	R/W	0x3	SS3 优先级 此字段包含确定采样序列发生器 3 的优先级编码的二进制编码值。优先级编码 0x0 的优先级最高; 0x3 最低。分配给序列发生器的优先级必须被唯一地映射。如果两个或两个以上的域相等, 那么 ADC 就不能正常工作。
11:10	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
9:8	SS2	R/W	0x2	SS2 优先级 此字段包含确定采样序列发生器 2 的优先级编码的二进制编码值。优先级编码 0x0 的优先级最高; 0x3 最低。分配给序列发生器的优先级必须被唯一地映射。如果两个或两个以上的域相等, 那么 ADC 就不能正常工作。
7:6	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5:4	SS1	R/W	0x1	SS1 优先级 此字段包含确定采样序列发生器 1 的优先级编码的二进制编码值。优先级编码 0x0 的优先级最高; 0x3 最低。分配给序列发生器的优先级必须被唯一地映射。如果两个或两个以上的域相等, 那么 ADC 就不能正常工作。
3:2	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
1:0	SS0	R/W	0x0	SS0 优先级 此字段包含确定采样序列发生器 0 的优先级编码的二进制编码值。优先级编码 0x0 的优先级最高; 0x3 最低。分配给序列发生器的优先级必须被唯一地映射。如果两个或两个以上的域相等, 那么 ADC 就不能正常工作。

## 寄存器 10: ADC 采样相位控制寄存器 (ADCSPC) , 偏移量 0x024

本寄存器可设置采样点的相位, 从0.0°到337.5°可选。例如, 借助于芯片内的两路ADC模块可实现双倍采样率, 其方法便是将两路ADC模块配置为对同一模拟输入端采样, 并将一路ADC模块采样相位配置为0°、另一路ADC模块采样相位配置为180°。

**注意:** 注意: 当 PHASE 域非 0 时应当小心, 因为对 AINx 的延迟采样可能导致采样结果有别于系统设计的预期。从 ADC 触发到采样的时间会增加, 可能会使得响应时间要比预期要长。因此而增加的延迟可能会对系统的设计带来负面影响。设计人员必须慎重考虑此延时的影响。

### ADC 采样相位控制寄存器 (ADCSPC)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x024

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												PHASE			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3:0	PHASE	R/W	0x0	相位差 此位域设置采样点与基准采样时间的滞后相位差。
	值	描述		
	0x0	ADC 采样滞后 0.0°		
	0x1	ADC 采样滞后 22.5°		
	0x2	ADC 采样滞后 45.0°		
	0x3	ADC 采样滞后 67.5°		
	0x4	ADC 采样滞后 90.0°		
	0x5	ADC 采样滞后 112.5°		
	0x6	ADC 采样滞后 135.0°		
	0x7	ADC 采样滞后 157.5°		
	0x8	ADC 采样滞后 180.0°		
	0x9	ADC 采样滞后 202.5°		
	0xA	ADC 采样滞后 225.0°		
	0xB	ADC 采样滞后 247.5°		
	0xC	ADC 采样滞后 270.0°		
	0xD	ADC 采样滞后 292.5°		
	0xE	ADC 采样滞后 315.0°		
	0xF	ADC 采样滞后 337.5°		

## 寄存器 11: ADC 处理器采样序列启动寄存器 (ADCPSSI), 偏移量 0x028

本寄存器提供了由软件启动采样序列工作的手段。采样序列可以各自分别启动,也可以按照任意组合同时启动。当同时触发多个采样序列时,其执行顺序将由ADCSSPRI寄存器的优先级编码决定。

本寄存器还能配置并同时启动对所有ADC模块的采样。方法是:首先配置第一个ADC模块。然后对该模块的ADCPSSI寄存器进行写操作。将相应的SS位与SYNCWAIT位一同置位。其余ADC模块同样按此流程逐个进行配置。当配置完最后一个ADC模块后,其ADCPSSI寄存器同样应写入,写操作时应将相应的SS位与GSYNC位一同置位。随后所有ADC模块将同时开始按照其配置执行采样。

### ADC 处理器采样序列启动寄存器 (ADCPSSI)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x028

类型 R/W, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	GSYNC	保留			SYNCWAIT	保留											
类型	R/W	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												SS3	SS2	SS1	SS0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	-

位/域	名称	类型	复位	描述
31	GSYNC	R/W	0	全局同步  值 描述 1 此位用于同时启动多个ADC模块的采样。只要有ADC模块因SSn和SYNCWAIT位置位而得以启用,则对本标志位进行写操作时,这些模块将开始采样。 0 一旦开始采样则此标志位清零。
30:28	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。
27	SYNCWAIT	R/W	0	同步等待  值 描述 1 若此标志位置位,则在启动某个采样序列时,该序列会等GSYNC位置位时再执行。 0 启动某个采样序列时,将立即执行。
26:4	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
3	SS3	WO	-	<p>SS3启动</p> <p>值 描述</p> <p>1 如果采样序列发生器 3 已在 ADCACTSS 寄存器中启用，则将执行采样。</p> <p>0 无影响。</p> <p>此标志位只能进行写操作；读操作的返回值无意义。</p>
2	SS2	WO	-	<p>SS2 启动</p> <p>值 描述</p> <p>1 如果采样序列发生器 2 已在 ADCACTSS 寄存器中启用，则将执行采样。</p> <p>0 无影响。</p> <p>此标志位只能进行写操作；读操作的返回值无意义。</p>
1	SS1	WO	-	<p>SS1 启动</p> <p>值 描述</p> <p>1 如果采样序列发生器 1 已在 ADCACTSS 寄存器中启用，则将执行采样。</p> <p>0 无影响。</p> <p>此标志位只能进行写操作；读操作的返回值无意义。</p>
0	SS0	WO	-	<p>SS0 启动</p> <p>值 描述</p> <p>1 如果采样序列发生器 0 已在 ADCACTSS 寄存器中启用，则将执行采样。</p> <p>0 无影响。</p> <p>此标志位只能进行写操作；读操作的返回值无意义。</p>



## 寄存器 12: ADC 采样平均控制寄存器 ( ADCSAC ) , 偏移量 0x030

本寄存器控制硬件对转换结果进行取平均值时的次数。最终保存入 FIFO 的结果实际上是按照 ADC 速率进行  $2^{AVG}$  次连续 A/D 采样结果的平均值。若 AVG 为 0, 则每次采样将直接写入 FIFO, 不进行任何平均运算。若 AVG = 6, 那么 ADC 模块将连续采集 64 个采样后求其算术平均值, 并将计算结果存入 FIFO 中。若 AVG = 7, 其结果无法预料。

### ADC 采样平均控制寄存器 (ADCSAC)

ADC0 基址: 0x4003.8000  
 ADC1 基址: 0x4003.9000  
 偏移量 0x030  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													AVG			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2:0	AVG	R/W	0x0	<p>硬件平均控制</p> <p>确定硬件平均计算的 ADC 采样数。该 AVG 域可以是 0 到 6 之间的任意值。如果 AVG = 7, 其结果无法预测。</p> <p>值 描述</p> <p>0x0 无硬件过采样</p> <p>0x1 2x 硬件过采样</p> <p>0x2 4x 硬件过采样</p> <p>0x3 8x 硬件过采样</p> <p>0x4 16x 硬件过采样</p> <p>0x5 32x 硬件过采样</p> <p>0x6 64x 硬件过采样</p> <p>0x7 保留</p>

**寄存器 13: ADC 数字比较器中断状态及清除寄存器 ( ADCDCISC ) , 偏移量 0x034**

本寄存器可指示数字比较器中断状态，并能对其确认操作。每个数字比较器占用 1 个位。

**ADC 数字比较器中断状态及清除寄存器 (ADCDCISC)**

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x034

类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DCINT7	DCINT6	DCINT5	DCINT4	DCINT3	DCINT2	DCINT1	DCINT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7	DCINT7	R/W1C	0	数字比较器 7 中断状态及清除  值 描述 1 数字比较器 7 产生中断。 0 无中断。  对该位写入 1，即可将其清零。
6	DCINT6	R/W1C	0	数字比较器 6 中断状态及清除  值 描述 1 数字比较器 6 产生中断。 0 无中断。  对该位写入 1，即可将其清零。
5	DCINT5	R/W1C	0	数字比较器 5 中断状态及清除  值 描述 1 数字比较器 5 产生中断。 0 无中断。  对该位写入 1，即可将其清零。
4	DCINT4	R/W1C	0	数字比较器 4 中断状态及清除  值 描述 1 数字比较器 4 产生中断。 0 无中断。  对该位写入 1，即可将其清零。

位/域	名称	类型	复位	描述
3	DCINT3	R/W1C	0	<p>数字比较器 3 中断状态及清除</p> <p>值 描述</p> <p>1 数字比较器 3 产生中断。</p> <p>0 无中断。</p> <p>对该位写入 1，即可将其清零。</p>
2	DCINT2	R/W1C	0	<p>数字比较器 2 中断状态及清除</p> <p>值 描述</p> <p>1 数字比较器 2 产生中断。</p> <p>0 无中断。</p> <p>对该位写入 1，即可将其清零。</p>
1	DCINT1	R/W1C	0	<p>数字比较器 1 中断状态及清除</p> <p>值 描述</p> <p>1 数字比较器 1 产生中断。</p> <p>0 无中断。</p> <p>对该位写入 1，即可将其清零。</p>
0	DCINT0	R/W1C	0	<p>数字比较器 0 中断状态及清除</p> <p>值 描述</p> <p>1 数字比较器 0 产生中断。</p> <p>0 无中断。</p> <p>对该位写入 1，即可将其清零。</p>

## 寄存器 14: ADC 控制寄存器 (ADCCTL), 偏移量 0x038

本寄存器用于配置参考电压。用于转换的参考电压可为 VREFA+ 和 VREFA- 或 VDDA 和 GNDA。

## ADC 控制寄存器 (ADCCTL)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x038

类型 R/W, 复位 0x0000.0010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												保留	保留	VREF	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

位/域	名称	类型	复位	描述
31:5	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
4	保留	RO	1	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3:2	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
1:0	VREF	R/W	0x0	参考电压源选择
	值	描述		
	0x0	VDDA 和 GNDA 即参考电压源		
	0x1 - 0x3	以外部 VREFA+ 和 VREFA- 输入的电压值作为参考电压源。		

## 寄存器 15: ADC 采样序列输入复用选择寄存器 0 ( ADCSSMUX0 ) , 偏移量 0x040

该寄存器协同 ADCSSEMUX0 寄存器一起定义序列中每个采样的模拟输入配置。该序列由采样序列发生器 0 执行。如果 ADCSSEMUX0 寄存器中相应的 EMUXn 位也被置位, 则在该寄存器中的 MUXn 域将从 AIN[23:16] 中选择。当相应的 EMUXn 位清零, MUXn 域从 AIN[15:0] 中选择。本寄存器为 32 位宽, 可分别定义 8 个采样的信息。

**注意:** 在本微处理器上没有 AIN[31:24] 通道。当相应的 EMUXn 位被置位时, 将 MUXn 配置为 0x8-0xF 会导致未知行为。

### ADC 采样序列输入复用选择寄存器 0 (ADCSSMUX0)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x040

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MUX7				MUX6				MUX5				MUX4			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MUX3				MUX2				MUX1				MUX0			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:28	MUX7	R/W	0x0	第8个采样动作的输入选择 MUX7 字段将在采样序列发生器所执行序列的第 8 个采样中使用。它确定在进行模数转换时采样哪个模拟输入。此处设置的值表示相应的管脚。例如, 当 EMUX7 清零时, 0x1 表示输入管脚是为 AIN1。当 EMUX7 位置位时, 0x1 表示输入是 AIN17。 如果启用了差分采样 ( ADCSSCTL0 寄存器中的 D7 位置位 ), 必须将该域设置为对编号 i, 即成对的输入是“2i”和“2i+1”。
27:24	MUX6	R/W	0x0	第7个采样动作的输入选择 MUX6 字段将在采样序列发生器所执行序列的第 7 个采样中使用。它确定在进行模数转换时采样哪个模拟输入。
23:20	MUX5	R/W	0x0	第 6 个采样的输入选择 MUX5 字段将在采样序列发生器所执行序列的第 6 个采样中使用。它确定在进行模数转换时采样哪个模拟输入。
19:16	MUX4	R/W	0x0	第 5 个采样的输入选择 MUX4 字段将在采样序列发生器所执行序列的第 5 个采样中使用。它确定在进行模数转换时采样哪个模拟输入。
15:12	MUX3	R/W	0x0	第4个采样的输入选择 MUX3 字段将在采样序列发生器所执行序列的第 4 个采样中使用。它确定在进行模数转换时采样哪个模拟输入。
11:8	MUX2	R/W	0x0	第 3 个采样的输入选择 MUX2 字段将在采样序列发生器所执行序列的第 3 个采样中使用。它确定在进行模数转换时采样哪个模拟输入。
7:4	MUX1	R/W	0x0	第 2 个采样的输入选择 MUX1 字段将在采样序列发生器所执行序列的第 2 个采样中使用。它确定在进行模数转换时采样哪个模拟输入。

位/域	名称	类型	复位	描述
3:0	MUX0	R/W	0x0	第 1 个采样的输入选择 MUX0 字段将在采样序列发生器所执行序列的第 1 个采样中使用。它确定在进行模数转换时采样哪个模拟输入。

## 寄存器 16: ADC 采样序列控制寄存器 0 ( ADCSSCTL0 ) , 偏移量 0x044

本寄存器包含采样序列中每个采样动作的配置信息。在配置一个采样序列时，不管该序列中包含几个有效的采样动作，其最后1个有效采样动作的 END 位必须置位。本寄存器为 32 位宽，可分别定义 8 个采样的信息。

### ADC 采样序列控制寄存器 0 (ADCSSCTL0)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x044

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TS7	IE7	END7	D7	TS6	IE6	END6	D6	TS5	IE5	END5	D5	TS4	IE4	END4	D4
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31	TS7	R/W	0	第 8 个采样动作的温度传感器选择  值 描述 1 在采样序列的第 8 个采样动作读取温度传感器的内容。 0 在采样序列的第 8 个采样动作读取由 ADCSSMUXn 寄存器指定的输入管脚。
30	IE7	R/W	0	第 8 个采样动作的中断启用  值 描述 1 在第 8 个采样转换结束后，原始中断信号 ( INR0 位 ) 生效。若 ADCIM 寄存器的 MASK0 位置位，就会向中断控制器提交中断。 0 不产生原始中断。  允许同一序列中多个采样动作各自产生中断。
29	END7	R/W	0	第 8 个采样动作是序列末端  值 描述 1 第 8 个采样为序列的最后一个采样。 0 序列中的另一个采样是最后采样。  可在任意的采样位置结束序列。软件必须在队列中的某个位置处设置一个 ENDn 位。当某一采样已设置了 ENDn 位，其后的采样就不会发出转换请求 ( 即使该域值不为零 ) 。

位/域	名称	类型	复位	描述
28	D7	R/W	0	第 8 个采样动作的差分输入选择  值 描述 1 该模拟输入是差分采样。相应的 ADCSSMUXn 半字节必须被设置为对编号 i，即成对的输入为“2i”和“2i+1”。 0 模拟输入信号为单端信号。  由于温度传感器是不能以差分形式输入的，因此当 TS7 位置位时，此标志位必须为 0。
27	TS6	R/W	0	第 7 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 7 个采样中使用。
26	IE6	R/W	0	第 7 个采样动作的中断启用 定义和 IE7 一样，但在第 7 个采样中使用。
25	END6	R/W	0	第 7 个采样是序列末端 定义和 END7 一样，但在第 7 个采样中使用。
24	D6	R/W	0	第 7 个采样动作的差分输入选择 定义和 D7 一样，但在第 7 个采样中使用。
23	TS5	R/W	0	第 6 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 6 个采样中使用。
22	IE5	R/W	0	第 6 个采样动作的中断启用 定义和 IE7 一样，但在第 6 个采样中使用。
21	END5	R/W	0	第 6 个采样是序列末端 定义和 END7 一样，但在第 6 个采样中使用。
20	D5	R/W	0	第 6 个采样动作的差分输入选择 定义和 D7 一样，但在第 6 个采样中使用。
19	TS4	R/W	0	第 5 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 5 个采样中使用。
18	IE4	R/W	0	第 5 个采样动作的中断启用 定义和 IE7 一样，但在第 5 个采样中使用。
17	END4	R/W	0	第 5 个采样是序列末端 定义和 END7 一样，但在第 5 个采样中使用。
16	D4	R/W	0	第 5 个采样动作的差分输入选择 定义和 D7 一样，但在第 5 个采样中使用。
15	TS3	R/W	0	第 4 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 4 个采样中使用。
14	IE3	R/W	0	第 4 个采样动作的中断启用 定义和 IE7 一样，但在第 4 个采样中使用。
13	END3	R/W	0	第 4 个采样是序列末端 定义和 END7 一样，但在第 4 个采样中使用。



位/域	名称	类型	复位	描述
12	D3	R/W	0	第 4 个采样动作的差分输入选择 定义和 D7 一样，但在第 4 个采样中使用。
11	TS2	R/W	0	第 3 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 3 个采样中使用。
10	IE2	R/W	0	第 3 个采样动作的中断启用 定义和 IE7 一样，但在第 3 个采样中使用。
9	END2	R/W	0	第 3 个采样是序列末端 定义和 END7 一样，但在第 3 个采样中使用。
8	D2	R/W	0	第 3 个采样动作的差分输入选择 定义和 D7 一样，但在第 3 个采样中使用。
7	TS1	R/W	0	第 2 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 2 个采样中使用。
6	IE1	R/W	0	第 2 个采样动作的中断启用 定义和 IE7 一样，但在第 2 个采样中使用。
5	END1	R/W	0	第 2 个采样是序列末端 定义和 END7 一样，但在第 2 个采样中使用。
4	D1	R/W	0	第 2 个采样动作的差分输入选择 定义和 D7 一样，但在第 2 个采样中使用。
3	TS0	R/W	0	第 1 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 1 个采样中使用。
2	IE0	R/W	0	第 1 个采样动作的中断启用 定义和 IE7 一样，但在第 1 个采样中使用。
1	END0	R/W	0	第 1 个采样是序列末端 定义和 END7 一样，但在第 1 个采样中使用。
0	D0	R/W	0	第 1 个采样动作的差分输入选择 定义和 D7 一样，但在第 1 个采样中使用。

寄存器 17: ADC 采样序列结果 FIFO 寄存器 0 ( ADCSSFIFO0 ) , 偏移量 0x048

寄存器 18: ADC 采样序列结果 FIFO 寄存器 1 ( ADCSSFIFO1 ) , 偏移量 0x068

寄存器 19: ADC 采样序列结果 FIFO 寄存器 2 ( ADCSSFIFO2 ) , 偏移量 0x088

寄存器 20: ADC 采样序列结果 FIFO 寄存器 3 ( ADCSSFIFO3 ) , 偏移量 0x0A8

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

本寄存器用于保存采样序列发生器采集的转换结果 ( ADCSSFIFO0 寄存器用于采样序列发生器 0 ; ADCSSFIFO1 寄存器用于采样序列发生器 1 ; ADCSSFIFO2 寄存器用于采样序列发生器 2 ; ADCSSFIFO3 寄存器用于采样序列发生器 3 )。每读1次本寄存器可返回1个转换结果数据, 依次返回第1个采样结果、第2个采样结果.....依此类推, 直到FIFO被读空。如果 FIFO 因未能及时得到软件处理而出现了上溢或下溢状况, 可通过 ADCOSTAT 寄存器和 ADCUSTAT 寄存器进行查询。

#### ADC 采样序列结果 FIFO 寄存器 n ( ADCSSFIFOn )

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x048

类型 RO, 复位 -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				DATA											
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
11:0	DATA	RO	-	转换结果数值

寄存器 21: ADC 采样序列 FIFO 0 状态寄存器 ( ADCSSFSTAT0 ) , 偏移量 0x04C

寄存器 22: ADC 采样序列 FIFO 1 状态寄存器 ( ADCSSFSTAT1 ) , 偏移量 0x06C

寄存器 23: ADC 采样序列 FIFO 2 状态寄存器 ( ADCSSFSTAT2 ) , 偏移量 0x08C

寄存器 24: ADC 采样序列 FIFO 3 状态寄存器 ( ADCSSFSTAT3 ) , 偏移量 0x0AC

本寄存器为采样序列发生器的状态提供了一个窗口, 可藉此查询满/空状态信息以及头指针、尾指针的位置。复位值 0x100 表示 FIFO 为空, 其头尾指针都指向索引值 0。ADCSSFSTAT0 寄存器代表 FIFO 0 ( 8 个单元 ) 的状态; ADCSSFSTAT1 寄存器代表 FIFO 1 ( 4 个单元 ) 的状态; ADCSSFSTAT2 寄存器代表 FIFO 2 ( 4 个单元 ) 的状态; ADCSSFSTAT3 寄存器代表 FIFO 3 ( 1 个单元 ) 的状态。

#### ADC 采样序列 FIFO n 状态寄存器 ( ADCSSFSTATn )

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x04C

类型 RO, 复位 0x0000.0100

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留			FULL	保留			EMPTY	HPTR				TPTR			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:13	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
12	FULL	RO	0	FIFO 满 值 描述 1 FIFO 当前已满。 0 FIFO 当前未滿。
11:9	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
8	EMPTY	RO	1	FIFO 为空 值 描述 1 FIFO 当前已空。 0 FIFO 当前非空。
7:4	HPTR	RO	0x0	FIFO 头指针 该字段包含当前 FIFO 的“头”指针索引, 也就是下一个要执行的写操作的位。 FIFO0 的有效值是 0x0-0x7; FIFO1 和 FIFO2 的有效值是 0x0-0x3; FIFO3 的有效值是 0x0。

位/域	名称	类型	复位	描述
3:0	TPTR	RO	0x0	FIFO尾指针 该字段包含当前 FIFO 的“尾”指针索引，也就是下一个要读取的位。 FIFO0 的有效值是 0x0-0x7；FIFO1 和 FIFO2 的有效值是 0x0-0x3； FIFO3 的有效值是 0x0。

## 寄存器 25: ADC 采样序列寄存器 0 ( ADCSSOP0 ) , 偏移量 0x050

本寄存器用于设置如何处理采样序列发生器0的各个采样结果，是保存到采样序列FIFO0中还是传递给数字比较器。

## ADC 采样序列寄存器 0 (ADCSSOP0)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x050

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			S7DCOP	保留			S6DCOP	保留			S5DCOP	保留			S4DCOP
类型	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留			S3DCOP	保留			S2DCOP	保留			S1DCOP	保留			S0DCOP
类型	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
28	S7DCOP	R/W	0	采样 7 数字比较器的操作  值 描述 1 第 8 个采样动作的结果不写入 FIFO，而是发送给指定的数字比较器（由 ADCSSDC0 寄存器的 S7DCSEL 位指定）。 0 第 8 个采样动作的结果保存在采样序列发生器 FIFO0 中。
27:25	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
24	S6DCOP	R/W	0	采样 6 数字比较器的操作 定义和 S7DCOP 一样，但在第 7 个采样中使用。
23:21	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
20	S5DCOP	R/W	0	采样 5 数字比较器的操作 定义和 S7DCOP 一样，但在第 6 个采样中使用。
19:17	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
16	S4DCOP	R/W	0	采样 4 数字比较器的操作 定义和 S7DCOP 一样，但在第 5 个采样中使用。
15:13	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
12	S3DCOP	R/W	0	采样 3 数字比较器的操作 定义和 S7DCOP 一样，但在第 4 个采样中使用。
11:9	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

---

位/域	名称	类型	复位	描述
8	S2DCOP	R/W	0	采样 2 数字比较器的操作 定义和 S7DCOP 一样，但在第 3 个采样中使用。
7:5	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
4	S1DCOP	R/W	0	采样 1 数字比较器的操作 定义和 S7DCOP 一样，但在第 2 个采样中使用。
3:1	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	S0DCOP	R/W	0	采样 0 数字比较器的操作 定义和 S7DCOP 一样，但在第 1 个采样中使用。

## 寄存器 26: ADC 采样序列数字比较器选择寄存器 0 ( ADCSSDC0 ) , 偏移量 0x054

若 ADCSSOP0 寄存器的 SnDCOP 位置位, 通过本寄存器就可确定哪一个数字比较器从序列发生器 0 的给定转换上接收到采样。

## ADC 采样序列数字比较器选择寄存器 0 (ADCSSDC0)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x054

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	S7DCSEL				S6DCSEL				S5DCSEL				S4DCSEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	S3DCSEL				S2DCSEL				S1DCSEL				S0DCSEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:28	S7DCSEL	R/W	0x0	<p>采样 7 数字比较器的选择</p> <p>若 ADCSSOP0 寄存器的 S7DCOP 位置位, 则通过本位域选择将采样序列器 0 的第 8 个采样发送给哪个数字比较器 ( 及其相关的控制寄存器 )。</p> <p>注意: 表中未列出的值是保留值。</p> <p>值 描述</p> <p>0x0 数字比较器单元 0 ( ADCDCCMP0 和 ADCDCCTL0 )</p> <p>0x1 数字比较器单元 1 ( ADCDCCMP1 和 ADCDCCTL1 )</p> <p>0x2 数字比较器单元 2 ( ADCDCCMP2 和 ADCDCCTL2 )</p> <p>0x3 数字比较器单元 3 ( ADCDCCMP3 和 ADCDCCTL3 )</p> <p>0x4 数字比较器单元 4 ( ADCDCCMP4 和 ADCDCCTL4 )</p> <p>0x5 数字比较器单元 5 ( ADCDCCMP5 和 ADCDCCTL5 )</p> <p>0x6 数字比较器单元 6 ( ADCDCCMP6 和 ADCDCCTL6 )</p> <p>0x7 数字比较器单元 7 ( ADCDCCMP7 和 ADCDCCTL7 )</p>
27:24	S6DCSEL	R/W	0x0	<p>采样 6 数字比较器的选择</p> <p>此位域的编码方式与 S7DCSEL 相同, 但用于第 7 个采样。</p>
23:20	S5DCSEL	R/W	0x0	<p>采样 5 数字比较器的选择</p> <p>此位域的编码方式与 S7DCSEL 相同, 但用于第 6 个采样。</p>
19:16	S4DCSEL	R/W	0x0	<p>采样 4 数字比较器的选择</p> <p>此位域的编码方式与 S7DCSEL 相同, 但用于第 5 个采样。</p>
15:12	S3DCSEL	R/W	0x0	<p>采样 3 数字比较器的选择</p> <p>此位域的编码方式与 S7DCSEL 相同, 但用于第 4 个采样。</p>
11:8	S2DCSEL	R/W	0x0	<p>采样 2 数字比较器的选择</p> <p>此位域的编码方式与 S7DCSEL 相同, 但用于第 3 个采样。</p>

位/域	名称	类型	复位	描述
7:4	S1DCSEL	R/W	0x0	采样 1 数字比较器的选择 此位域的编码方式与 S7DCSEL 相同，但用于第 2 个采样。
3:0	S0DCSEL	R/W	0x0	采样 0 数字比较器的选择 此位域的编码方式与 S7DCSEL 相同，但用于第 1 个采样。



## 寄存器 27: ADC 样序列扩展输入复用选择寄存器 ( ADCSSEMUX0 ) , 偏移量 0x058

该寄存器协同 ADCSSMUX0 寄存器一起定义序列中每个采样的模拟输入配置。该序列由采样序列发生器 0 执行。如果该寄存器中的某一位被置位, 则 ADCSSMUX0 寄存器中的相应 MUXn 域将从 AIN[23:16] 中选择。如果该寄存器中的某一位被清零, 相应的 MUXn 域从 AIN[15:0] 中选择。本寄存器为 32 位宽, 可分别定义 8 个采样的信息。

请注意: 当使用差分通道标识时 ( ADCSSCTL0 位中的 Dn 位置位 ) , 不使用该寄存器, 因为 ADCSSMUX0 寄存器可以选择所有的可用对。

### ADC 样序列扩展输入复用选择寄存器 (ADCSSEMUX0)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x058

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留			EMUX7	保留			EMUX6	保留			EMUX5	保留			EMUX4
类型	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留			EMUX3	保留			EMUX2	保留			EMUX1	保留			EMUX0
类型	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:29	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
28	EMUX7	R/W	0x0	第 8 个采样动作的输入选择 ( 高位 ) EMUX7 字段在采样序列发生器所执行序列的第 8 个采样中使用。  值 描述 1 使用 ADCSSMUX0 寄存器从 AIN[23:16] 中选取第 8 个采样输入。 例如, 如果 MUX7 域为 0x0, 选择 AIN16。 0 使用 ADCSSMUX0 寄存器从 AIN[15:0] 中选取第 8 个采样输入。 例如, 如果 MUX7 域为 0x0, 选择 AIN0。
27:25	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
24	EMUX6	R/W	0x0	第 7 个采样动作的输入选择 ( 高位 ) EMUX6 字段在采样序列发生器所执行序列的第 7 个采样中使用。对该位的描述同 EMUX7 相同。
23:21	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
20	EMUX5	R/W	0x0	第 6 个采样动作的输入选择 ( 高位 ) EMUX5 字段在采样序列发生器所执行序列的第 6 个采样中使用。对该位的描述同 EMUX7 相同。
19:17	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
16	EMUX4	R/W	0x0	第 5 个采样动作的输入选择 (高位) EMUX4 字段在采样序列发生器所执行序列的第 5 个采样中使用。对该位的描述同 EMUX7 相同。
15:13	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。
12	EMUX3	R/W	0x0	第 4 个采样动作的输入选择 (高位) EMUX3 字段在采样序列发生器所执行序列的第 4 个采样中使用。对该位的描述同 EMUX7 相同。
11:9	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。
8	EMUX2	R/W	0x0	第 3 个采样动作的输入选择 (高位) EMUX2 字段在采样序列发生器所执行序列的第 3 个采样中使用。对该位的描述同 EMUX7 相同。
7:5	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。
4	EMUX1	R/W	0x0	第 2 个采样动作的输入选择 (高位) EMUX1 字段在采样序列发生器所执行序列的第 2 个采样中使用。对该位的描述同 EMUX7 相同。
3:1	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。
0	EMUX0	R/W	0x0	第 1 个采样动作的输入选择 (高位) EMUX0 字段在采样序列发生器所执行序列的第 1 个采样中使用。对该位的描述同 EMUX7 相同。

**寄存器 28: ADC 采样序列输入复用选择寄存器 1 ( ADCSSMUX1 ) ， 偏移量 0x060**

**寄存器 29: ADC 采样序列输入复用选择寄存器 2 ( ADCSSMUX2 ) ， 偏移量 0x080**

该寄存器协同 ADCSSEMUX1 或 ADCSSEMUX2 寄存器一起定义序列中每个采样的模拟输入配置。该序列由采样序列发生器 1 或 2 执行。如果在 ADCSSEMUX1 或 ADCSSEMUX2 寄存器中的 EMUXn 位被置位，则在该寄存器中的 MUXn 域将从 AIN[23:16] 中选择。当相应的 EMUXn 位清零，MUXn 域从 AIN[15:0] 中选择。本寄存器为 16 位宽，可分别定义 4 个采样的信息。详细的位描述请见在 789 页上的 ADCSSMUX0 寄存器。ADCSSMUX1 寄存器配置采样序列发生器 1，ADCSSMUX2 寄存器配置采样序列发生器 2。

**注意：** 在本微处理器上没有 AIN[31:24] 通道。当相应的 EMUXn 位被置位时，将 MUXn 配置为 0x8-0xF 会导致未知行为。

#### ADC 采样序列输入复用选择寄存器 n ( ADCSSMUXn )

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x060

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MUX3				MUX2				MUX1				MUX0			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:12	MUX3	R/W	0x0	第4个采样的输入选择
11:8	MUX2	R/W	0x0	第3个采样的输入选择
7:4	MUX1	R/W	0x0	第2个采样的输入选择
3:0	MUX0	R/W	0x0	第1个采样的输入选择

**寄存器 30: ADC 采样序列控制寄存器 1 ( ADCSSCTL1 ) , 偏移量 0x064****寄存器 31: ADC 采样序列控制寄存器 2 ( ADCSSCTL2 ) , 偏移量 0x084**

这些寄存器包含在序列中每个采样的配置信息, 使用采样序列发生器 1 或 2 来执行。在配置一个采样序列时, 不管该序列中包含几个有效的采样动作, 其最后一个有效采样动作的 END 位必须置位。本寄存器为 16 位宽, 可分别定义 4 个采样的信息。详细的位描述请见在 791 页上的 ADCSSCTL0 寄存器。ADCSSCTL1 寄存器配置采样序列发生器 1, ADCSSCTL2 寄存器配置采样序列发生器 2。

## ADC 采样序列控制寄存器 n ( ADCSSCTLn )

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x064

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15	TS3	R/W	0	第 4 个采样动作的温度传感器选择 定义和 TS7 一样, 但在第 4 个采样中使用。
14	IE3	R/W	0	第 4 个采样动作的中断启用 定义和 IE7 一样, 但在第 4 个采样中使用。
13	END3	R/W	0	第 4 个采样是序列末端 定义和 END7 一样, 但在第 4 个采样中使用。
12	D3	R/W	0	第 4 个采样动作的差分输入选择 定义和 D7 一样, 但在第 4 个采样中使用。
11	TS2	R/W	0	第 3 个采样动作的温度传感器选择 定义和 TS7 一样, 但在第 3 个采样中使用。
10	IE2	R/W	0	第 3 个采样动作的中断启用 定义和 IE7 一样, 但在第 3 个采样中使用。
9	END2	R/W	0	第 3 个采样是序列末端 定义和 END7 一样, 但在第 3 个采样中使用。
8	D2	R/W	0	第 3 个采样动作的差分输入选择 定义和 D7 一样, 但在第 3 个采样中使用。
7	TS1	R/W	0	第 2 个采样动作的温度传感器选择 定义和 TS7 一样, 但在第 2 个采样中使用。

位/域	名称	类型	复位	描述
6	IE1	R/W	0	第 2 个采样动作的中断启用 定义和 IE7 一样，但在第 2 个采样中使用。
5	END1	R/W	0	第 2 个采样是序列末端 定义和 END7 一样，但在第 2 个采样中使用。
4	D1	R/W	0	第 2 个采样动作的差分输入选择 定义和 D7 一样，但在第 2 个采样中使用。
3	TS0	R/W	0	第 1 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 1 个采样中使用。
2	IE0	R/W	0	第 1 个采样动作的中断启用 定义和 IE7 一样，但在第 1 个采样中使用。
1	END0	R/W	0	第 1 个采样是序列末端 定义和 END7 一样，但在第 1 个采样中使用。
0	D0	R/W	0	第 1 个采样动作的差分输入选择 定义和 D7 一样，但在第 1 个采样中使用。

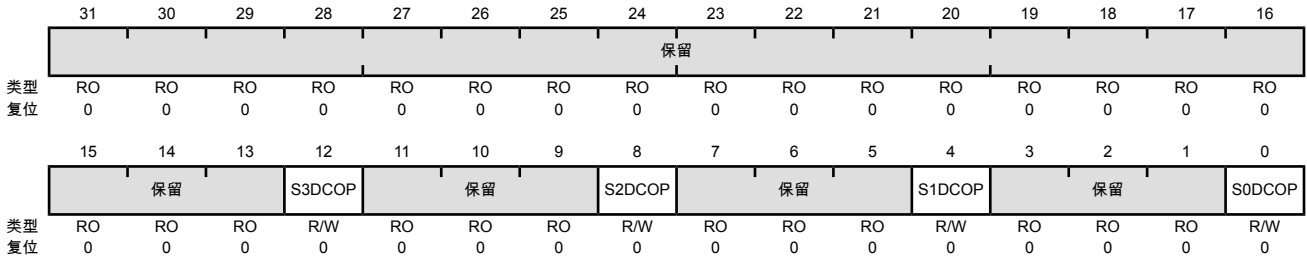
**寄存器 32: ADC 采样序列 1 工作寄存器 ( ADCSSOP1 ) , 偏移量 0x070**

**寄存器 33: ADC 采样序列 2 工作寄存器 ( ADCSSOP2 ) , 偏移量 0x090**

本寄存器用于设置如何处理采样序列发生器 1 或 2 的各个采样结果, 是保存到采样序列 FIFO 中还是传递给数字比较器。ADCSSOP1 寄存器配置采样序列发生器 1;ADCSSOP2 寄存器配置采样序列发生器 2。

ADC 采样序列 n 工作寄存器 ( ADCSSOPn )

ADC0 基址: 0x4003.8000  
 ADC1 基址: 0x4003.9000  
 偏移量 0x070  
 类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:13	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
12	S3DCOP	R/W	0	采样 3 数字比较器的操作  值 描述 1 第 4 个采样会被发送到由 ADCSSDC0n 寄存器中的 S3DCSEL 位指定的数字比较器单元, 且该值不会写入 FIFO。 0 第 4 个采样储存到采样序列器 FIFO 中。
11:9	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
8	S2DCOP	R/W	0	采样 2 数字比较器的操作 定义和 S3DCOP 一样, 但在第 3 个采样中使用。
7:5	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
4	S1DCOP	R/W	0	采样 1 数字比较器的操作 定义和 S3DCOP 一样, 但在第 2 个采样中使用。
3:1	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	S0DCOP	R/W	0	采样 0 数字比较器的操作 定义和 S3DCOP 一样, 但在第 1 个采样中使用。

**寄存器 34: ADC 采样序列数字比较器选择寄存器 1 ( ADCSSDC1 ) ， 偏移量 0x074**

**寄存器 35: ADC 采样序列数字比较器选择寄存器 2 ( ADCSSDC2 ) ， 偏移量 0x094**

若 ADCSSOPn 寄存器的 SnDCOP 位置位，通过本寄存器就可确定哪一个数字比较器从序列发生器 0 的给定转换上接收到采样。ADCSSDC1 寄存器配置采样序列发生器 1；ADCSSDC2 寄存器配置采样序列发生器 2。

#### ADC 采样序列数字比较器选择寄存器 n ( ADCSSDCn )

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x074

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	S3DCSEL				S2DCSEL				S1DCSEL				S0DCSEL			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:12	S3DCSEL	R/W	0x0	<p>采样 3 数字比较器的选择</p> <p>若 ADCSSOPn 寄存器的 S3DCOP 位置位，则通过本位域选择将采样序列器 n 的第 8 个采样发送给哪个数字比较器（及其相关的控制寄存器）。</p> <p>注意：表中未列出的值是保留值。</p> <p>值 描述</p> <p>0x0 数字比较器单元 0 ( ADCDCCMP0 和 ADCCCTL0 )</p> <p>0x1 数字比较器单元 1 ( ADCDCCMP1 和 ADCCCTL1 )</p> <p>0x2 数字比较器单元 2 ( ADCDCCMP2 和 ADCCCTL2 )</p> <p>0x3 数字比较器单元 3 ( ADCDCCMP3 和 ADCCCTL3 )</p> <p>0x4 数字比较器单元 4 ( ADCDCCMP4 和 ADCCCTL4 )</p> <p>0x5 数字比较器单元 5 ( ADCDCCMP5 和 ADCCCTL5 )</p> <p>0x6 数字比较器单元 6 ( ADCDCCMP6 和 ADCCCTL6 )</p> <p>0x7 数字比较器单元 7 ( ADCDCCMP7 和 ADCCCTL7 )</p>
11:8	S2DCSEL	R/W	0x0	<p>采样 2 数字比较器的选择</p> <p>此位域的编码方式与 S3DCSEL 相同，但用于第 3 个采样。</p>
7:4	S1DCSEL	R/W	0x0	<p>采样 1 数字比较器的选择</p> <p>此位域的编码方式与 S3DCSEL 相同，但用于第 2 个采样。</p>
3:0	S0DCSEL	R/W	0x0	<p>采样 0 数字比较器的选择</p> <p>此位域的编码方式与 S3DCSEL 相同，但用于第 1 个采样。</p>

**寄存器 36: ADC 采样序列扩展输入复用选择寄存器 1 ( ADCSSEMUX1 ) , 偏移量 0x078**

**寄存器 37: ADC 采样序列扩展输入复用选择寄存器 2 ( ADCSSEMUX2 ) , 偏移量 0x098**

该寄存器协同 ADCSSMUX1 或 ADCSSMUX2 寄存器一起定义序列中每个采样的模拟输入配置。该序列由采样序列发生器 1 或 2 执行。如果该寄存器中的某一位被置位, 则在 ADCSSMUX1 或 ADCSSMUX2 寄存器中的相应 MUXn 域从 AIN[23:16] 中选择。如果该寄存器中的某一位被清零, 相应的 MUXn 域从 AIN[15:0] 中选择。该寄存器为 16 位宽, 并且包含 4 个可能采样的信息。ADCSSEMUX1 寄存器控制采样序列发生器 1; ADCSSEMUX2 寄存器控制采样序列发生器 2。

请注意: 当使用差分通道标识时 ( ADCSSCTL1 或 ADCSSCTL2 寄存器中的 Dn 位置位 ), 不使用该寄存器, 因为 ADCSSMUX1 或 ADCSSMUX2 寄存器可以选择所有的可用对。

#### ADC 采样序列扩展输入复用选择寄存器 n ( ADCSSEMUXn )

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x078

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		EMUX3		保留			EMUX2		保留			EMUX1		保留	EMUX0
类型	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:13	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
12	EMUX3	R/W	0x0	第 4 个采样动作的输入选择 (高位) EMUX3 字段在采样序列发生器所执行序列的第 4 个采样中使用。  值 描述 1 使用 ADCSSMUX1 或 ADCSSMUX2 寄存器从 AIN[23:16] 中选取第 4 个采样输入。例如, 如果 MUX3 域为 0x0, 选择 AIN16。 0 使用 ADCSSMUX1 或 ADCSSMUX2 寄存器从 AIN[15:0] 中选取第 4 个采样输入。例如, 如果 MUX3 域为 0x0, 选择 AIN0。
11:9	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
8	EMUX2	R/W	0x0	第 3 个采样动作的输入选择 (高位) EMUX2 字段在采样序列发生器所执行序列的第 3 个采样中使用。对该位的描述同 EMUX3 相同。
7:5	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
4	EMUX1	R/W	0x0	第 2 个采样动作的输入选择 (高位) EMUX1 字段在采样序列发生器所执行序列的第 2 个采样中使用。对该位的描述同 EMUX3 相同。



---

位/域	名称	类型	复位	描述
3:1	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	EMUX0	R/W	0x0	第 1 个采样动作的输入选择 (高位) EMUX0 字段在采样序列发生器所执行序列的第 1 个采样中使用。对该位的描述同 EMUX3 相同。

**寄存器 38: ADC 采样序列输入复用选择寄存器 3 ( ADCSSMUX3 ) , 偏移量 0x0A0**

该寄存器协同 ADCSSEMUX3 寄存器一起定义序列中每个采样的模拟输入配置。该序列由采样序列发生器 3 执行。如果 ADCSSEMUX3 寄存器中的 EMUX0 位被置位，则该寄存器中的 MUX0 域从 AIN[23:16] 中选择。当 EMUX0 位被清零，MUX0 域从 AIN[15:0] 中选择。本寄存器为 4 位宽，并且包含 1 个可能采样的信息。详细的位描述请见在 789 页上的 ADCSSMUX0 寄存器。

## ADC 采样序列输入复用选择寄存器 3 (ADCSSMUX3)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x0A0

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												MUX0			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3:0	MUX0	R/W	0	第 1 个采样的输入选择

### 寄存器 39: ADC 采样序列控制寄存器 3 ( ADCSSCTL3 ) , 偏移量 0x0A4

该寄存器包含采样序列发生器 3 所执行的采样配置信息。本寄存器为 4 位宽，并且包含 1 个可能采样的信息。详细的位描述请见在 791 页 上的 ADCSSCTL0 寄存器。

#### ADC 采样序列控制寄存器 3 (ADCSSCTL3)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x0A4

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												TS0	IE0	END0	D0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	TS0	R/W	0	第 1 个采样动作的温度传感器选择 定义和 TS7 一样，但在第 1 个采样中使用。
2	IE0	R/W	0	第 1 个采样动作的中断启用 定义和 IE7 一样，但在第 1 个采样中使用。
1	END0	R/W	0	第 1 个采样是序列末端 定义和 END7 一样，但在第 1 个采样中使用。
0	D0	R/W	0	第 1 个采样动作的差分输入选择 定义和 D7 一样，但在第 1 个采样中使用。

**寄存器 40: ADC 采样序列器 3 工作寄存器 ( ADCSSOP3 ) , 偏移量 0x0B0**

本寄存器用于设置如何处理采样序列发生器3的各个采样结果，是保存到采样序列FIFO0中还是传递给数字比较器。

**ADC 采样序列器 3 工作寄存器 (ADCSSOP3)**

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x0B0

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
	S0DCOP															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	S0DCOP	R/W	0	采样 0 数字比较器的操作

**值 描述**

- 1 采样会被发送到由 ADCSSDC03 寄存器中的 S0DCSEL 位指定的数字比较器单元，且该值不会写入 FIFO。
- 0 采样会被保存到采样序列 FIFO 3 中。

**寄存器 41: ADC 采样序列 3 数字比较器选择寄存器 ( ADCSSDC3 ) , 偏移量 0x0B4**

若 ADCSSOP3 寄存器的 SnDCOP 位置位, 通过本寄存器就可确定哪一个数字比较器从序列发生器 3 的给定转换上接收到采样。

**ADC 采样序列 3 数字比较器选择寄存器 (ADCSSDC3)**

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x0B4

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												S0DCSEL			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3:0	S0DCSEL	R/W	0x0	采样 0 数字比较器的选择 若 ADCSSOP3 寄存器的 S0DCOP 位置位, 则通过本位域选择将采样序列器 3 的采样发送给哪个数字比较器 ( 及其相关的控制寄存器 )。 注意: 表中未列出的值是保留值。

**值 描述**

0x0 数字比较器单元 0 ( ADCDCCMP0 和 ADCCCTL0 )

0x1 数字比较器单元 1 ( ADCDCCMP1 和 ADCCCTL1 )

0x2 数字比较器单元 2 ( ADCDCCMP2 和 ADCCCTL2 )

0x3 数字比较器单元 3 ( ADCDCCMP3 和 ADCCCTL3 )

0x4 数字比较器单元 4 ( ADCDCCMP4 和 ADCCCTL4 )

0x5 数字比较器单元 5 ( ADCDCCMP5 和 ADCCCTL5 )

0x6 数字比较器单元 6 ( ADCDCCMP6 和 ADCCCTL6 )

0x7 数字比较器单元 7 ( ADCDCCMP7 和 ADCCCTL7 )

## 寄存器 42: ADC 采样序列扩展输入复用选择寄存器 3 ( ADCSSMUX3 ) , 偏移量 0x0B8

该寄存器协同 ADCSSMUX3 寄存器一起定义序列中每个采样的模拟输入配置。该序列由采样序列发生器 3 执行。如果 EMUX0 被置位，则 ADCSSMUX3 寄存器中的相应 MUX0 域从 AIN[23:16] 中选择。当 EMUX0 位被清零，MUX0 域从 AIN[15:0] 中选择。本寄存器为 1 位宽，并且包含 1 个可能采样的信息。

请注意：当使用差分通道标识时 ( ADCSSCTL3 位中的 Dn 位置位 )，不使用该寄存器，因为 ADCSSMUX0 寄存器可以选择所有的可用对。

### ADC 采样序列扩展输入复用选择寄存器 3 (ADCSSMUX3)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0x0B8

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															EMUX0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	EMUX0	R/W	0x0	第 1 个采样动作的输入选择 ( 高位 ) EMUX0 字段在采样序列发生器所执行序列的唯一采样中使用。

#### 值 描述

- 1 使用 ADCSSMUX3 寄存器从 AIN[23:16] 中选取采样输入。例如，如果 MUX0 域为 0x0，选择 AIN16。
- 0 使用 ADCSSMUX3 寄存器从 AIN[15:0] 中选取采样输入。例如，如果 MUX0 域为 0x0，选择 AIN0。

## 寄存器 43: ADC 数字比较器复位启动条件寄存器 ( ADCDCRIC ) , 偏移量 0xD00

本寄存器能够将任一数字比较器中断或触发事件恢复到其初始状态。通过这种方式及时恢复中断，可保障数字比较器中断或触发事件所用到的数据不会已经过期失效。

### ADC 数字比较器复位启动条件寄存器 (ADCDCRIC)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0xD00

类型 WO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								DCTRIG7	DCTRIG6	DCTRIG5	DCTRIG4	DCTRIG3	DCTRIG2	DCTRIG1	DCTRIG0
类型	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO	WO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DCINT7	DCINT6	DCINT5	DCINT4	DCINT3	DCINT2	DCINT1	DCINT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO	WO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:24	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
23	DCTRIG7	WO	0	数字比较器触发事件 7  值 描述 1 将数字比较器 7 触发事件单元恢复到其初始状态。 0 无影响。  当触发事件被清除时，此标志位将自动清零。 由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生触发事件，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。该位置位后，软件应当等到该位清零后才可以继续。
22	DCTRIG6	WO	0	数字比较器触发事件 6  值 描述 1 将数字比较器 6 触发事件单元恢复到其初始状态。 0 无影响。  当触发事件被清除时，此标志位将自动清零。 由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生触发事件，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。

位/域	名称	类型	复位	描述
21	DCTRIG5	WO	0	<p>数字比较器触发事件 5</p> <p>值 描述</p> <p>1 将数字比较器 5 触发事件单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当触发事件被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生触发事件，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
20	DCTRIG4	WO	0	<p>数字比较器触发事件 4</p> <p>值 描述</p> <p>1 将数字比较器 4 触发事件单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当触发事件被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生触发事件，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
19	DCTRIG3	WO	0	<p>数字比较器触发事件 3</p> <p>值 描述</p> <p>1 将数字比较器 3 触发事件单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当触发事件被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生触发事件，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
18	DCTRIG2	WO	0	<p>数字比较器触发事件 2</p> <p>值 描述</p> <p>1 将数字比较器 2 触发事件单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当触发事件被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生触发事件，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
17	DCTRIG1	WO	0	<p>数字比较器触发事件 1</p> <p>值 描述</p> <p>1 将数字比较器 1 触发事件单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当触发事件被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生触发事件，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>



位/域	名称	类型	复位	描述
16	DCTRIG0	WO	0	<p>数字比较器触发事件 0</p> <p>值 描述</p> <p>1 将数字比较器 0 触发事件单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当触发事件被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生触发事件，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
15:8	保留	RO	0x00	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
7	DCINT7	WO	0	<p>数字比较器中断 7</p> <p>值 描述</p> <p>1 将数字比较器 7 中断单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当中断被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生中断，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
6	DCINT6	WO	0	<p>数字比较器中断 6</p> <p>值 描述</p> <p>1 将数字比较器 6 中断单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当中断被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生中断，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
5	DCINT5	WO	0	<p>数字比较器中断 5</p> <p>值 描述</p> <p>1 将数字比较器 5 中断单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当中断被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生中断，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>

位/域	名称	类型	复位	描述
4	DCINT4	WO	0	<p>数字比较器中断 4</p> <p>值 描述</p> <p>1 将数字比较器 4 中断单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当中断被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生中断，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
3	DCINT3	WO	0	<p>数字比较器中断 3</p> <p>值 描述</p> <p>1 将数字比较器 3 中断单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当中断被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生中断，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
2	DCINT2	WO	0	<p>数字比较器中断 2</p> <p>值 描述</p> <p>1 将数字比较器 2 中断单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当中断被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生中断，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
1	DCINT1	WO	0	<p>数字比较器中断 1</p> <p>值 描述</p> <p>1 将数字比较器 1 中断单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当中断被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生中断，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>
0	DCINT0	WO	0	<p>数字比较器中断 0</p> <p>值 描述</p> <p>1 将数字比较器 0 中断单元恢复到其初始状态。</p> <p>0 无影响。</p> <p>当中断被清除时，此标志位将自动清零。</p> <p>由于数字比较器采用ADC的当前转换值以及前一转换值来决定何时产生中断，所以在启动新的采样序列时，务必将数字比较器恢复到其初始状态，避免仍然使用已失效的数据。</p>

寄存器 44: ADC 数字比较器控制寄存器 0 ( ADCDCCTL0 ) , 偏移量 0xE00

寄存器 45: ADC 数字比较器控制寄存器 1 ( ADCDCCTL1 ) , 偏移量 0xE04

寄存器 46: ADC 数字比较器控制寄存器 2 ( ADCDCCTL2 ) , 偏移量 0xE08

寄存器 47: ADC 数字比较器控制寄存器 3 ( ADCDCCTL3 ) , 偏移量 0xE0C

寄存器 48: ADC 数字比较器控制寄存器 4 ( ADCDCCTL4 ) , 偏移量 0xE10

寄存器 49: ADC 数字比较器控制寄存器 5 ( ADCDCCTL5 ) , 偏移量 0xE14

寄存器 50: ADC 数字比较器控制寄存器 6 ( ADCDCCTL6 ) , 偏移量 0xE18

寄存器 51: ADC 数字比较器控制寄存器 7 ( ADCDCCTL7 ) , 偏移量 0xE1C

本寄存器配置用于产生处理器中断和/或 PWM 触发事件的比较编码。关于使用 ADC 数字比较器触发 PWM 发生器的更多信息, 请参阅“中断/ADC-触发选择器”在 1157页。

#### ADC 数字比较器控制寄存器 n ( ADCDCCTLn )

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0xE00

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		CTE	CTC			CTM			保留			CIE	CIC		CIM
类型	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:13	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
12	CTE	R/W	0	比较触发事件启用
	值 描述			
	1			启用触发事件功能状态机。将根据 CTC 及 CTM 域的设置使用 ADC 转换结果判断是否产生触发事件。
	0			禁用触发事件功能状态机。ADC 转换结果将被触发事件所忽略。

位/域	名称	类型	复位	描述
11:10	CTC	R/W	0x0	<p>比较触发事件条件</p> <p>当 ADC 转换结果与 COMP0 值以及 COMP1 值进行比较时，此位域定义哪个工作区域产生触发事件。COMP0 和 COMP1 域均在 ADCDCMPx 寄存器中定义。</p> <p>值 描述</p> <p>0x0 低值带 ADC 结果 &lt; COMP0 ≤ COMP1</p> <p>0x1 中值带 COMP0 ≤ ADC 结果 &lt; COMP1</p> <p>0x2 保留</p> <p>0x3 高值带 COMP0 ≤ COMP1 ≤ ADC 结果</p>
9:8	CTM	R/W	0x0	<p>比较触发事件模式</p> <p>此位域定义比较器触发事件的工作模式。</p> <p>值 描述</p> <p>0x0 持续触发模式 只要ADC的转换结果处于选定的工作区域内，就产生触发事件</p> <p>0x1 单次触发模式 当ADC的转换结果首次处于选定的工作区域内时，将产生一次触发事件</p> <p>0x2 迟滞持续触发模式 只要ADC的转换结果处于选定的工作区域内，就会持续产生触发事件；只有当转换结果到达相反的工作区域内时才会清除迟滞状态。 请注意，迟滞模式只适用于 CTC = 0x1 和 0x3 的情况。</p> <p>0x3 迟滞单次触发模式 当 ADC 的转换结果首次处于选定的工作区域内时，将产生一次触发事件。只有当转换结果到达相反的工作区域时，才会清除迟滞状态。此后才可能再次触发。 请注意，迟滞模式只适用于 CTC = 0x1 和 0x3 的情况。</p>
7:5	保留	RO	0x0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
4	CIE	R/W	0	<p>比较中断启用</p> <p>值 描述</p> <p>1 启用比较中断。根据 CIC 和 CIM 域的设置并通过 ADC 转换数据来判断是否应当产生中断。</p> <p>0 禁用比较中断。ADC 转换结果始终不会产生中断。</p>

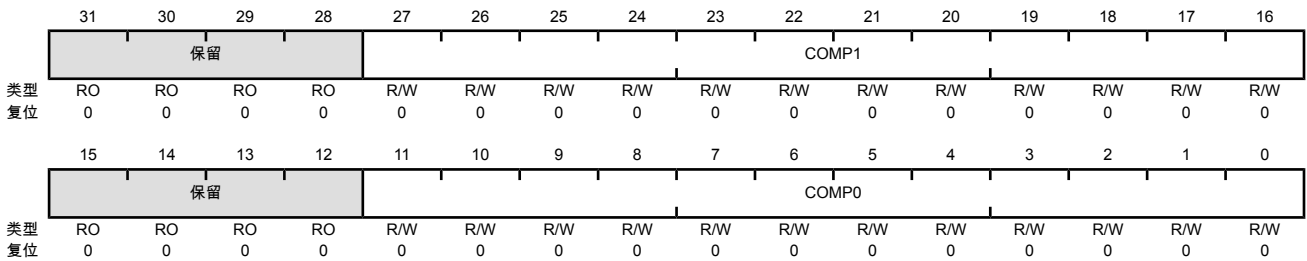
位/域	名称	类型	复位	描述
3:2	CIC	R/W	0x0	<p>比较中断条件</p> <p>当 ADC 转换结果与 COMP0 值以及 COMP1 值进行比较时，此位域定义哪个工作区域产生中断。COMP0 和 COMP1 域均在 ADCDCCMPx 寄存器中定义。</p> <p>值 描述</p> <p>0x0 低值带 ADC 结果 &lt; COMP0 ≤ COMP1</p> <p>0x1 中值带 COMP0 ≤ ADC 结果 &lt; COMP1</p> <p>0x2 保留</p> <p>0x3 高值带 COMP0 &lt; COMP1 ≤ ADC 结果</p>
1:0	CIM	R/W	0x0	<p>比较中断模式</p> <p>此位域定义比较器产生中断的工作模式。</p> <p>值 描述</p> <p>0x0 持续触发模式 只要ADC的转换结果处于选定的工作区域内，就产生中断</p> <p>0x1 单次触发模式 当ADC的转换结果首次处于选定的工作区域内时，将产生一次中断</p> <p>0x2 迟滞持续触发模式 只要ADC的转换结果处于选定的工作区域内，就会持续产生中断；只有当转换结果到达相反的工作区域内时才会清除迟滞状态。 请注意，迟滞模式只适用于 CTC = 0x1 和 0x3 的情况。</p> <p>0x3 迟滞单次触发模式 当ADC的转换结果首次处于选定的工作区域内时，将产生一次中断。只有当转换结果到达相反的工作区域时，才会清除迟滞状态。此后才可能再次中断。 请注意，迟滞模式只适用于 CTC = 0x1 和 0x3 的情况。</p>

- 寄存器 52: ADC 数字比较器范围寄存器 0 ( ADCDCCMP0 ) , 偏移量 0xE40
  - 寄存器 53: ADC 数字比较器范围寄存器 1 ( ADCDCCMP1 ) , 偏移量 0xE44
  - 寄存器 54: ADC 数字比较器范围寄存器 2 ( ADCDCCMP2 ) , 偏移量 0xE48
  - 寄存器 55: ADC 数字比较器范围寄存器 3 ( ADCDCCMP3 ) , 偏移量 0xE4C
  - 寄存器 56: ADC 数字比较器范围寄存器 4 ( ADCDCCMP4 ) , 偏移量 0xE50
  - 寄存器 57: ADC 数字比较器范围寄存器 5 ( ADCDCCMP5 ) , 偏移量 0xE54
  - 寄存器 58: ADC 数字比较器范围寄存器 6 ( ADCDCCMP6 ) , 偏移量 0xE58
  - 寄存器 59: ADC 数字比较器范围寄存器 7 ( ADCDCCMP7 ) , 偏移量 0xE5C
- 本寄存器定义两个比较门限值, ADC 模块据此判断转换的数据属于哪一工作区域。

注意: 注意: COMP1 域的值必须大于等于 COMP0 域的值, 否则可能产生无法预料的后果。

ADC 数字比较器范围寄存器 n ( ADCDCCMPn )

ADC0 基址: 0x4003.8000  
 ADC1 基址: 0x4003.9000  
 偏移量 0xE40  
 类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:28	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
27:16	COMP1	R/W	0x000	门限1 该域中的值与 ADC 转换结果进行比较。比较的结果用于决定转换结果是否在高值带范围内。 请注意, COMP1 位域的值必须大于等于 COMP0 位域的值。
15:12	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
11:0	COMP0	R/W	0x000	门限0 该域中的值与 ADC 转换结果进行比较。比较的结果用于决定转换数据是否在低值带范围内。

## 寄存器 60: ADC 外设属性寄存器 (ADCPP) , 偏移量 0xFC0

ADCPP 寄存器提供关于 ADC 模块属性的寄存器。

## ADC 外设属性寄存器 (ADCPP)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0xFC0

类型 RO, 复位 0x00B0.2187

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留								TS	RSL				类型			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	DC				CH				MSR								
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	1	

位/域	名称	类型	复位	描述
31:24	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
23	TS	RO	0x1	温度传感器  值 描述 1 该 ADC 模块有一个温度传感器。 0 该 ADC 模块没有温度传感器。  该域提供与传统 DC1 寄存器的 TEMPSNS 位相似的功能。
22:18	RSL	RO	0xC	分辨率 该域指定一个最大的二进制数值，用以表示已转换的采样。该域编码为一个二进制的数，范围在 0 到 32 位之间。
17:16	类型	RO	0x0	ADC 的架构  值 描述 0x0 SAR 0x1 - 0x3 保留
15:10	DC	RO	0x8	数字比较器计数 该域指定可用于转换器的 ADC 数字比较器的数量。该域编码为一个二进制的数，范围在 0 到 63 之间。 该域提供与传统 DC9 寄存器的 ADCnDCn 位相似的功能。
9:4	CH	RO	0x18	ADC 通道计数 该域指定可用于转换器的 ADC 输入通道的数量。本域编码为一个二进制的数，范围在 0 到 63 之间。 该域提供与传统 DC3 和 DC8 寄存器的 ADCnAINn 位相似的功能。

位/域	名称	类型	复位	描述																				
3:0	MSR	RO	0x7	<p>最大 ADC 采样率</p> <p>该域指定最大 ADC 转换数/秒。该 MSR 域编码如下：</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>保留</td> </tr> <tr> <td>0x1</td> <td>125 ksps</td> </tr> <tr> <td>0x2</td> <td>保留</td> </tr> <tr> <td>0x3</td> <td>250 ksps</td> </tr> <tr> <td>0x4</td> <td>保留</td> </tr> <tr> <td>0x5</td> <td>500 ksps</td> </tr> <tr> <td>0x6</td> <td>保留</td> </tr> <tr> <td>0x7</td> <td>1 Msps</td> </tr> <tr> <td>0x8 - 0xF</td> <td>保留</td> </tr> </tbody> </table>	值	描述	0x0	保留	0x1	125 ksps	0x2	保留	0x3	250 ksps	0x4	保留	0x5	500 ksps	0x6	保留	0x7	1 Msps	0x8 - 0xF	保留
值	描述																							
0x0	保留																							
0x1	125 ksps																							
0x2	保留																							
0x3	250 ksps																							
0x4	保留																							
0x5	500 ksps																							
0x6	保留																							
0x7	1 Msps																							
0x8 - 0xF	保留																							



## 寄存器 61: ADC 外设配置寄存器 (ADCPC), 偏移量 0xFC4

ADCPC 寄存器提供关于外设配置的信息。

### ADC 外设配置寄存器 (ADCPC)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0xFC4

类型 R/W, 复位 0x0000.0007

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												SR			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

位/域	名称	类型	复位	描述																				
31:4	保留	RO	0x0000.0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。																				
3:0	SR	R/W	0x7	<p>ADC 采样率</p> <p>该域指定 ADC 转换数/秒，适用于运行模式、睡眠模式和深度睡眠模式。该域编码是以传统的 RCGC0 寄存器编码为基础的。该可编程的采样速率不能超过由 ADCPP 寄存器中的 MSR 位指定的最大采样速率。该 SR 域编码如下：</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr><td>0x0</td><td>保留</td></tr> <tr><td>0x1</td><td>125 ksps</td></tr> <tr><td>0x2</td><td>保留</td></tr> <tr><td>0x3</td><td>250 ksps</td></tr> <tr><td>0x4</td><td>保留</td></tr> <tr><td>0x5</td><td>500 ksps</td></tr> <tr><td>0x6</td><td>保留</td></tr> <tr><td>0x7</td><td>1 Msps</td></tr> <tr><td>0x8 - 0xF</td><td>保留</td></tr> </tbody> </table>	值	描述	0x0	保留	0x1	125 ksps	0x2	保留	0x3	250 ksps	0x4	保留	0x5	500 ksps	0x6	保留	0x7	1 Msps	0x8 - 0xF	保留
值	描述																							
0x0	保留																							
0x1	125 ksps																							
0x2	保留																							
0x3	250 ksps																							
0x4	保留																							
0x5	500 ksps																							
0x6	保留																							
0x7	1 Msps																							
0x8 - 0xF	保留																							

## 寄存器 62: ADC 时钟配置寄存器 (ADCCC), 偏移量 0xFC8

ADCCC 寄存器控制 ADC 模块的时钟源。

## ADC 时钟配置寄存器 (ADCCC)

ADC0 基址: 0x4003.8000

ADC1 基址: 0x4003.9000

偏移量 0xFC8

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												CS			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述								
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。								
3:0	CS	R/W	0	ADC 时钟源 下表指定了可以产生 ADC 时钟输入的时钟源, 请参考图 5-5 在 211 页。  <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>可以是 16 MHz 系统时钟 (如果 PLL 被旁路), 也可以是通过 <math>PLL \div 25</math> 而得到的 16 MHz 时钟 (默认)。 请注意: 如果不使用 PLL, 则系统时钟必须大于等于 16 MHz。</td> </tr> <tr> <td>0x1</td> <td>PIOSC PIOSC 为 ADC 提供 16 MHz 时钟源。如果 PIOSC 用作时钟源, ADC 模块可继续以深度睡眠模式运行。</td> </tr> <tr> <td>0x2 - 0xF</td> <td>保留</td> </tr> </tbody> </table>	值	描述	0x0	可以是 16 MHz 系统时钟 (如果 PLL 被旁路), 也可以是通过 $PLL \div 25$ 而得到的 16 MHz 时钟 (默认)。 请注意: 如果不使用 PLL, 则系统时钟必须大于等于 16 MHz。	0x1	PIOSC PIOSC 为 ADC 提供 16 MHz 时钟源。如果 PIOSC 用作时钟源, ADC 模块可继续以深度睡眠模式运行。	0x2 - 0xF	保留
值	描述											
0x0	可以是 16 MHz 系统时钟 (如果 PLL 被旁路), 也可以是通过 $PLL \div 25$ 而得到的 16 MHz 时钟 (默认)。 请注意: 如果不使用 PLL, 则系统时钟必须大于等于 16 MHz。											
0x1	PIOSC PIOSC 为 ADC 提供 16 MHz 时钟源。如果 PIOSC 用作时钟源, ADC 模块可继续以深度睡眠模式运行。											
0x2 - 0xF	保留											

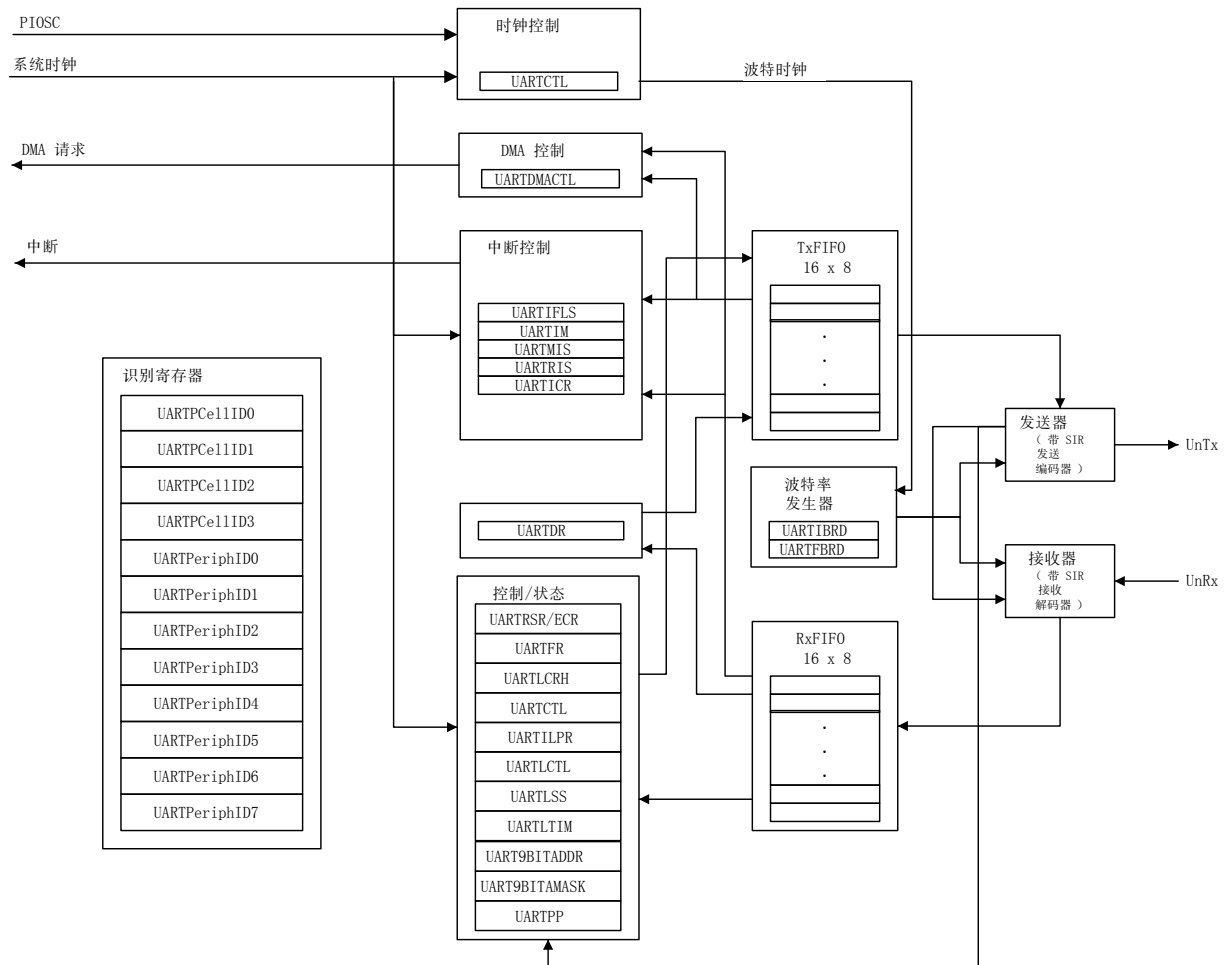
## 14 通用异步收发器 (UART)

该 Stellaris® LM4F232H5QD 控制器包括八个具有以下特征的通用异步收发器 (UART)：

- 可编程的波特率发生器，在常规模式（16 分频）下最高可达 5 Mbps，在高速模式（8 分频）下最高可达 10 Mbps
- 相互独立的 16×8 发送 (TX) FIFO 和接收 (RX) FIFO，可降低中断服务对 CPU 的占用
- FIFO 长度可编程，包括提供传统双缓冲接口的 1 字节深的操作
- FIFO 触发水平可设为 1/8, 1/4, 1/2, 3/4 和 7/8
- 标准的异步通信位：起始位、停止位和奇偶校验位
- 行中止的产生和检测
- 完全可编程的串行接口特性
  - 5、6、7 或 8 个数据位
  - 偶校验、奇校验、粘着或无奇偶校验位的产生/检测
  - 产生 1 或 2 个停止位
- IrDA 串行红外 (SIR) 编解码器
  - 可选择采用 IrDA 串行红外 (SIR) 输入输出或普通 UART 输入输出
  - 支持 IrDA SAR 编解码功能，半双工时数据传输率最高 115.2 Kbps
  - 支持标准的 3/16 位时间以及低功耗位时间 (1.41~2.23 μs)
  - 可编程的内部时钟发生器，允许对参考时钟进行 1 到 256 分频以得到低功耗模式的位持续时间。
- 支持与 ISO 7816 智能卡的通讯
- 调制解调器流量控制和状态 (在 UART1 模块上)
- 支持 LIN 协议
- 支持 EIA-485 (9 位)
- 提供标准的基于 FIFO 深度的中断以及发送结束中断
- 用微型直接内存访问 (μDMA) 有效的传输数据
  - 相互独立的发送通道和接收通道
  - 当接收 FIFO 中有数据时产生单次请求；当接收 FIFO 到达预设的触发深度时产生猝发请求
  - 当发送 FIFO 中有空闲单元时产生单次请求；当发送 FIFO 到达预设的触发深度时产生猝发请求

## 14.1 结构图

图 14-1. UART 模块的结构图



## 14.2 信号描述

下表列出了 UART 模块的外部信号并描述了每个信号的功能。UART 信号是某些 GPIO 信号的备用功能，复位时默认为 GPIO 信号，但是 U0Rx 和 U0Tx 管脚除外，这两个管脚默认为 UART 功能。下表中的“复用管脚/赋值”一栏列出了各 UART 信号所对应的 GPIO 管脚。应对 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618页) 的 AFSEL 位进行置位以选择 UART 功能。括号中的数字表示必须写入 GPIO 端口控制 (GPIOCTL) 寄存器 (636页) 中 PMCn 位域的编码，以便向指定的 GPIO 端口引脚分配 UART 信号。有关如何配置 GPIO 的更多信息，请参考“通用输入/输出端口 (GPIO)”在 597页。

表 14-1. UART 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
U0Rx	37	PA0 (1)	I	TTL	UART 模块 0 接收信号。
U0Tx	38	PA1 (1)	O	TTL	UART 模块 0 发送信号。

表 14-1. UART 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
U1CTS	35 63	PC5 (8) PF1 (1)	I	TTL	UART 模块 1 CTS ( Clear to Send, 允许发送 ) 调制解调器流控输入信号。
U1DCD	64	PF2 (1)	I	TTL	UART 模块 1 DCD ( Data Carrier Detect, 数据载波检测 ) 调制解调器状态输入信号。
U1DSR	65	PF3 (1)	I	TTL	UART 模块 1 DSR ( Data Set Ready, 数据设备就绪 ) 调制解调器输出控制线。
U1DTR	61	PF4 (1)	O	TTL	UART 模块 1 DTR ( Data Terminal Ready, 数据终端就绪 ) 调制解调器状态输入信号。
U1RI	134	PE7 (1)	I	TTL	UART 模块 1 RI ( Ring Indicator, 振铃指示 ) 调制解调器状态输入信号。
U1RTS	36 62	PC4 (8) PF0 (1)	O	TTL	UART 模块 1 RTS ( Request to Send, 请求发送 ) 调制解调器流控输出线。
U1Rx	36 97	PC4 (2) PB0 (1)	I	TTL	UART 模块 1 接收信号。
U1Tx	35 98	PC5 (2) PB1 (1)	O	TTL	UART 模块 1 发送信号。
U2Rx	51 143	PG4 (1) PD6 (1)	I	TTL	UART 模块 2 接收信号。
U2Tx	50 144	PG5 (1) PD7 (1)	O	TTL	UART 模块 2 发送信号。
U3Rx	34	PC6 (1)	I	TTL	UART 模块 3 接收信号。
U3Tx	33	PC7 (1)	O	TTL	UART 模块 3 发送信号。
U4Rx	36 120	PC4 (1) PJ0 (1)	I	TTL	UART 模块 4 接收信号。
U4Tx	35 121	PC5 (1) PJ1 (1)	O	TTL	UART 模块 4 发送信号。
U5Rx	122 139	PJ2 (1) PE4 (1)	I	TTL	UART 模块 5 接收信号。
U5Tx	123 140	PJ3 (1) PE5 (1)	O	TTL	UART 模块 5 发送信号。
U6Rx	127 141	PJ4 (1) PD4 (1)	I	TTL	UART 模块 6 接收信号。
U6Tx	128 142	PJ5 (1) PD5 (1)	O	TTL	UART 模块 6 发送信号。
U7Rx	15 112	PE0 (1) PK4 (1)	I	TTL	UART 模块 7 接收信号。
U7Tx	14 111	PE1 (1) PK5 (1)	O	TTL	UART 模块 7 发送信号。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 14.3 功能描述

每个 Stellaris UART 执行“并 - 串”和“串 - 并”转换功能。其功能与 16C550 UART 类似，但两者的寄存器不兼容。

用户可通过 UART 控制 (UARTCTL) 寄存器 (见 852 页) 的 TXE 和 RXE 位对 UART 进行发送和/或接收配置。不在复位状态下时，发送和接收均可进行。必须先将 UARTCTL 中的 UARTEEN 位清零，从而禁用 UART，然后才能对控制寄存器进行配置。如果在进行 TX 或 RX 操作期间禁用 UART，则会先完成当前处理，然后再停止 UART。

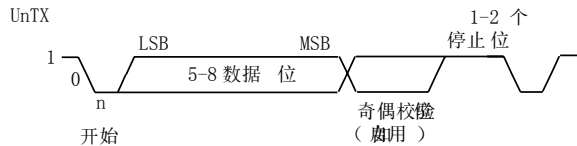
UART 模块还包含一个串行红外 (SIR) 编码器/解码器模块，该模块可以与一个红外收发器连接以实现 IrDA SIR 物理层连接。使用 UARTCTL 寄存器对 SIR 功能进行设定。

### 14.3.1 发送/接收逻辑

发送逻辑对从发送 FIFO 读取的数据执行“并 - 串”转换。控制逻辑输出串行位流时，最先输出起始位，之后按照控制寄存器的配置依次输出若干数据位（最低有效位在前）、奇偶校验位和停止位。详见图 14-2 在 830 页。

检测到有效的起始脉冲后，接收逻辑会对接收到的位流执行“串 - 并”转换。此外还会进行溢出、奇偶校验和线路中止检测以及帧错误检查，并将这些状态随数据一同写入接收 FIFO 中。

图 14-2. UART 字符帧



### 14.3.2 波特率的产生

波特率分频值是一个 22 位数，它由 16 位整数和 6 位小数组成。波特率发生器使用这两个值组成的数字来确定周期。波特率分频值支持小数部分，使得 UART 可以产生各种标准波特率。

16 位整数通过 UART 波特率分频值整数 (UARTIBRD) 寄存器（见 848 页）进行加载；而 6 位小数则通过 UART 波特率分频值小数 (UARTFBRD) 寄存器（见 849 页）进行加载。波特率分频值 (BRD) 和系统时钟之间具有以下关系（其中  $BRDI$  是 BRD 的整数部分， $BRDF$  是小数部分，之间用一个小数点隔开。）

$$BRD = BRDI + BRDF = \text{UARTSysClk} / (\text{ClkDiv} * \text{波特率})$$

其中  $\text{UARTSysClk}$  是与 UART 相连的系统时钟， $\text{ClkDiv}$  为 16（如果 UARTCTL 中的 HSE 清零）或 8（如果对 HSE 进行置位）。默认情况下，该系统时钟为“时钟控制”在 209 页中描述的主系统时钟。另外，UART 可根据内部精确振荡器 (PIOSC) 计时，不受系统时钟选择的影响。这使 UART 时钟能够独立于系统时钟 PLL 设置编程。请参考 UARTCC 寄存器获取更多详细信息。

6 位小数部分（即写入 UARTFBRD 寄存器的 DIVFRAC 位域的数值）的计算方法是：将波特率分频值的小数部分乘以 64，之后加 0.5 以抵消舍入误差：

$$\text{UARTFBRD}[\text{DIVFRAC}] = \text{取整}(\text{BRDF} * 64 + 0.5)$$

UART 产生一个内部波特率基准时钟，其频率为波特率的 8 倍或 16 倍（称为 Baud8 或 Baud16，具体视 UARTCTL 中 HSE 位 [第 5 位] 的设置而定）。这个基准时钟经过 8 或 16 分频后便可产生发送时钟，还可以在接收操作期间用于检测错误。请注意，在 ISO 7816 智能卡模式（在 UARTCTL 寄存器中对 SMART 位进行置位时）下，HSE 位的状态对时钟产生没有影响。

UARTIBRD 和 UARTFBRD 寄存器与 UART 线控，高字节 (UARTLCRH) 寄存器（见 850 页）一起组成一个 30 位内部寄存器。这个内部寄存器仅在对 UARTLCRH 执行了写入操作后才会更新，所以在对波特率分频值进行更改后，必须再执行一个写入 UARTLCRH 寄存器的操作，才能使更改生效。

可以采用四种顺序来更新波特率寄存器：

- 写入 UARTIBRD、写入 UARTFBRD 然后写入 UARTLCRH
- 写入 UARTFBRD、写入 UARTIBRD 然后写入 UARTLCRH
- 写入 UARTIBRD 然后写入 UARTLCRH

- 写入 UARTFBRD 然后写入 UARTLCRH

### 14.3.3 数据传输

接收或发送的数据存放在两个 16 字节 FIFO 中，不过接收 FIFO 的每个字符还额外包含四位状态信息。发送时，数据被写入发送 FIFO。如果启用了 UART，它会让数据帧按照 UARTLCRH 寄存器中设置的参数开始发送，直至发送 FIFO 中的数据全部发送完毕。数据一经写入发送 FIFO（即，如果该 FIFO 不为空），UART 标志 (UARTFR) 寄存器（见 845 页）中的 BUSY 位即会生效，并且在数据发送期间一直保持有效。BUSY 位仅在发送 FIFO 为空且包括停止位在内的最后一个字符已经从移位寄存器中发出时，才会变无效。即便 UART 不再处于启用状态，也可以指示自身正“忙”。

在接收器空闲（UnRx 信号持续为 1）且数据输入变为“低电平”（收到起始位）时，接收计数器开始运行，并且根据 UARTCTL（详见“发送/接收逻辑”在 830 页）中的 HSE 位（第 5 位）设置，在 Baud16 的第八个周期或者 Baud8 的第四个周期对数据进行采样。

如果 UnRx 信号在 Baud16 的第 8 个周期（HSE 清零）或者 Baud 8 的第 4 个周期（HSE 置位）仍然为低电平，则起始位有效且可以识别，否则即忽略该起始位。检测到有效起始位后，会按照设定的数据字符长度和 UARTCTL 中 HSE 位的值，每 16 个 Baud16 周期或每 8 个 Baud8 周期（即：每个位周期）对后续数据位进行一次采样。如果启用了奇偶校验模式，则接着检查奇偶校验位。数据长度和奇偶校验都在 UARTLCRH 寄存器中定义。

最后，如果 UnRx 信号为高电平，则可确认停止位有效，否则即表示发生了帧错误。接收到一个完整的字时，相关数据会连同与该字相关的错误位一起存放到接收 FIFO 中。

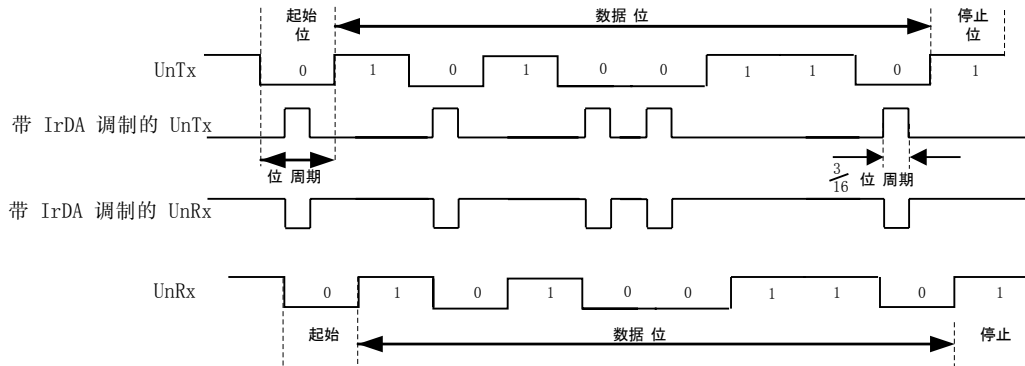
### 14.3.4 串行红外 (SIR)

UART 外设包含一个 IrDA 串行红外 (SIR) 编码器/解码器模块。IrDA SIR 模块的功能是在异步 UART 数据流和半双工串行 SIR 接口之间进行转换。不在片上执行任何模拟信号处理。SIR 模块的作用是向 UART 提供数字编码输出和解码输入。启用 SIR 模块后，该模块将使用 UnTx 和 UnRx 管脚实施 SIR 协议。这些信号应与一个红外收发器连接以实现 IrDA SIR 物理层连接。SIR 模块可以接收和发送，但是只能以半双工方式进行红外通信，所以它不能同时进行接收和发送。发送必须停止，然后才能接收数据。IrDA SIR 物理层规定在发送和接收之间至少有 10ms 的延时。SIR 模块有两种工作模式：

- 在标准 IrDA 模式下，输出管脚上发送的逻辑 0 电平为高脉冲，其位宽为所选波特率位周期的 3/16，而发送的逻辑 1 电平则是一个静态低电平信号。这些电平控制红外发送器的驱动装置，逢 0 电平便发送光脉冲。在接收端，接收到的光脉冲给接收器的光敏晶体管基极加电，将其输出拉至低电平并将 UART 输入管脚置为低电平。
- 在 IrDA 低功耗模式下，更改 UARTCR 寄存器中相应的位后，即可将发送红外脉冲的脉宽设置为内部产生的 IrLPBaud16 信号周期的三倍（在额定频率 1.8432 MHz 下为 1.63μs）。关于 IrDA 低功耗脉冲宽度配置的详细信息，请参阅 847 页。

图 14-3 在 832 页显示了含有 IrDA 调制和不含 IrDA 调制时的 UART 发送和接收信号。

图 14-3. IrDA 数据调制



在正常 IrDA 模式和低功耗 IrDA 模式下：

- 发送时，以 UART 数据位为基础进行编码
- 接收时，译码位被传输到 UART 接收逻辑电路

IrDA SIR 物理层规定了一个半双工通信链接，在发送和接收之间至少有 10 ms 延迟。必须使用软件来设定该延迟，因为 UART 本身并不自动支持延迟。之所以需要该延迟，是因为红外接收器的电子在耦合了相邻发送器 LED 的光能后会产生偏置，乃至饱和。这种延迟称为等待时间或接收器建立时间。

### 14.3.5 对 ISO 7816 的支持

UART 为与 ISO 7816 智能卡之间的通信提供了一些基本支持。对 UARTCTL 寄存器的第 3 位 (SMART) 进行置位时，UnTx 信号用作位时钟信号，而 UnRx 信号用作连接到智能卡的半双工通信线路。可将 GPIO 信号用于向智能卡发出复位信号。其余智能卡信号应由系统设计提供。此模式下的最大时钟速率是系统时钟/16。

使用 ISO 7816 模式时，UARTLCRH 寄存器必须设置为发送 8 位字 (WLEN 位域 6:5 配置为 0x3)，并使用偶校验 (对 PEN 和 EPS 进行置位)。在这种模式下，UART 自动使用 2 个停止位，而忽略 UARTLCRH 寄存器中的 STP2 位。

如果在发送期间检测到奇偶校验错误，UnRx 将在第二个停止位期间拉至低电平。此时，UART 将终止发送、清空发送 FIFO 并丢弃其中的所有数据，同时产生一个奇偶校验错误中断，以便让软件检测到该问题，并重新发送受影响的数据。请注意，在这种情况下，UART 并不支持自动重新发送。

### 14.3.6 对调制解调器握手信号的支持

本节介绍在将 UART 作为数据终端设备 (DTE) 或数据通信设备 (DCE) 连接时，如何为 UART1 配置并使用调制解调器的流控和状态信号。一般来说，调制解调器为 DCE，与调制解调器相连的计算设备为 DTE。

#### 14.3.6.1 信号

根据 UART 是用作 DCE 还是 DTE，UART1 所提供的状态信号有所不同。用作 DTE 时，调制解调器的流控和状态信号定义如下：

- $\overline{\text{UICTS}}$  为允许发送信号



- $\overline{\text{UIDSR}}$  为数据集就绪信号
- $\overline{\text{UIDCD}}$  为数据载波检测信号
- $\overline{\text{UIRI}}$  为振铃指示信号
- $\overline{\text{UIRTS}}$  为请求发送信号
- $\overline{\text{UIDTR}}$  为数据终端就绪信号

用作 DCE 时，调制解调器的流控和状态信号定义如下：

- $\overline{\text{UICTS}}$  为请求发送信号
- $\overline{\text{UIDSR}}$  为数据终端就绪信号
- $\overline{\text{UIRTS}}$  为允许发送信号
- $\overline{\text{UIDTR}}$  为数据集就绪信号

请注意，用作 DCE 时没有数据载波检测信号和振铃指示信号。如果需要用到这些信号，可使用通用 I/O 信号并提供软件支持来模拟这些功能。

#### 14.3.6.2 流控

流控既可以通过硬件实现也可以通过软件实现。下面将分别介绍这两种方法。

##### 硬件流控 (RTS/CTS)

可将  $\overline{\text{UIRTS}}$  输出端连接到接收设备的“允许发送”输入端，将接收设备上的“请求发送”输出端连接到  $\overline{\text{UICTS}}$  输入端，从而在两台设备之间实现硬件流控。

$\overline{\text{UICTS}}$  输入端控制发送器。发送器只有在  $\overline{\text{UICTS}}$  输入端生效的情况下才能发送数据。 $\overline{\text{UIRTS}}$  输出信号指示接收 FIFO 的状态。 $\overline{\text{UICTS}}$  将保持生效，直至达到预设水印电平，表明接收 FIFO 再无空间存储更多字符。

UARTCTL 寄存器的第 15 位 (CTSEN) 和第 14 位 (RTSEN) 指定流控模式，如表 14-2 在 833 页中所述。

表 14-2. 流控模式

CTSEN	RTSEN	描述
1	1	启用 RTS 和 CTS 流控
1	0	仅启用 CTS 流控
0	1	仅启用 RTS 流控
0	0	RTS 与 CTS 流控均禁用

请注意，当 RTSEN 为 1 时，软件无法通过 UARTCTL 寄存器的请求发送 (RTS) 位更改  $\overline{\text{UIRTS}}$  输出值，因此应当忽略 RTS 位的状态。

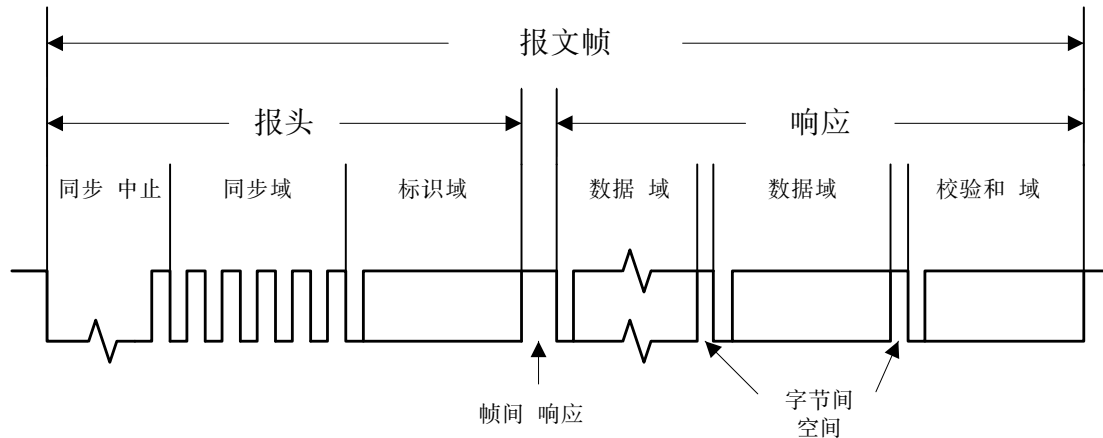
##### 软件流控 (调制解调器状态中断)

可以使用中断来指示 UART 的状态，从而实现两台设备间的软件流控。使用 UARTIM 寄存器的 3:0 位，可分别为  $\overline{\text{UIDSR}}$ 、 $\overline{\text{UIDCD}}$ 、 $\overline{\text{UICTS}}$  和  $\overline{\text{UIRI}}$  信号发出中断。可使用 UARTRIS 寄存器和 UARTRIS 寄存器查看原始中断状态和屏蔽中断状态。以上中断可通过 UARTRIS 寄存器予以清除。

### 14.3.7 对 LIN 的支持

无论作为主机还是从机，UART 模块都可对 LIN 协议提供硬件支持。可通过对 UARTCTL 寄存器的 LIN 位进行置位来启用 LIN 模式。LIN 报文以报文开头的同步间隔来加以识别。同步间隔传输一组 0。同步间隔之后是同步数据域 (0x55)。图 14-4 在 834 页显示了 LIN 报文的结构。

图 14-4. LIN 报文



应对 UART 进行如下配置，才能在 LIN 模式下工作：

1. 将 UART 配置为 1 个起始位、8 个数据位、无奇偶校验位以及 1 个停止位。启用发送 FIFO。
2. 对 UARTCTL 寄存器中的 LIN 位进行置位。

准备发送 LIN 报文时，TXFIFO 的 FIFO 位置 0 应包含同步数据 (0x55)，位置 1 应包含标识符数据，然后是待发送的数据，并且在最后 1 个 FIFO 位置写入校验和。

#### 14.3.7.1 LIN 主机模式

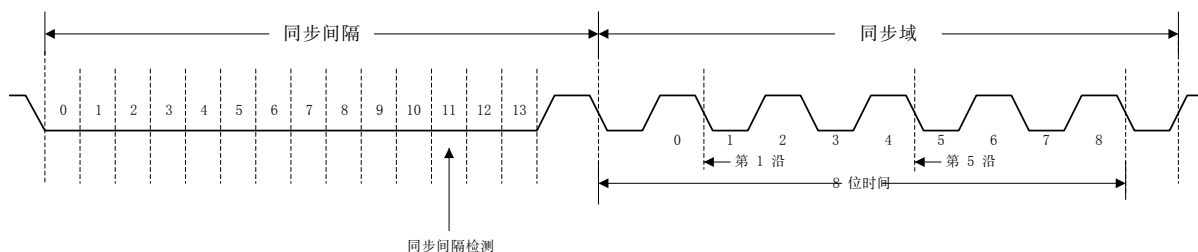
对 UARTLCTL 寄存器中的 MASTER 位进行置位，即可将 UART 启用为 LIN 主机模式。可通过 UARTLCTL 寄存器的 BLEN 位域对同步间隔的长度进行设置，允许的长度为 13-16 位（波特率时钟周期）。

#### 14.3.7.2 LIN 从机模式

需要将 LIN UART 从机模式的波特率调节为与 LIN 主机模式的波特率一致。在从机模式下，LIN UART 所能识别的同步间隔长度必须至少为 13 位。提供一个定时器来捕获同步域的第 1 个下降沿和第 5 个下降沿的时间数据，以便据此将波特率调节为与主机模式一致。

检测到同步间隔后，UART 模块将等待同步数据场。对 UARTRIS 寄存器的 LME1RIS 位进行置位，可在同步域的第一个下降沿产生中断，与此同时捕捉定时器的值并将其保存到 UARTLSS 寄存器 (T1) 中。对 UARTRIS 寄存器的 LME5RIS 位进行置位，可在同步域的第五个下降沿产生第二个中断，与此同时再次捕捉定时器的值 (T2)。可使用  $(T2-T1)/8$  计算出实际波特率，并应根据需要调节本机波特率。图 14-5 在 835 页显示了同步域。

图 14-5. LIN 同步域



### 14.3.8 9 位 UART 模式

UART 提供 9 位模式，可利用 UART9BITADDR 寄存器中的 9BITEN 位将其启用。此功能在 UART 的多分支配置中非常有用，在这种配置下，与多个从机相连的单个主机可以通过某个从机的地址或带地址字节限定符的地址集，与该从机相互通信。所有从机都会在奇偶校验位查找是否有该地址限定符，如果设置了该限定符，则对接收到的字节和预设地址进行比较。如果地址匹配，则继续接收或发送数据。如果地址不匹配，则丢弃该地址字节以及后续的所有数据字节。如果 UART 处于 9 位模式，则接收器在没有奇偶校验的模式下工作。可将该地址预先定义为与接收到的字节匹配，可通过 UART9BITADDR 寄存器进行配置。通过使用 UART9BITAMASK 寄存器中的地址屏蔽，匹配范围可扩大为地址集。默认情况下，UART9BITAMASK 为 0xFF，表示仅匹配指定的地址。

找不到匹配项时，会丢弃其余数据字节和清零的第 9 位。如果找到匹配项，则向 NVIC 发出一个中断信号，以便进一步操作。后续数据字节和清零的第 9 位存放在 FIFO 中。如果在这种情况下启用了  $\mu$ DMA 和/或 FIFO 操作，软件可以屏蔽该中断，无需处理器干预。9 位模式下的所有发送操作均为数据字节且第 9 位清零。软件可以将奇偶校验设定改写为粘着奇偶校验，对某个字节启用奇校验，从而改写将置位（以指示地址）的第 9 位。要让发送时间与正确的奇偶校验设定匹配，可将地址字节作为单次传输发送，而不是突发传输。发送 FIFO 并不含地址/数据位，因此软件应该相应地启用地址位。

### 14.3.9 FIFO 操作

UART 有 2 个 16x8 的 FIFO；一个用于发送，另一个用于接收。这两个 FIFO 都通过 UART 数据 (UARTDR) 寄存器（见 840 页）进行访问。UARTDR 寄存器的读取操作返回一个 12 位的值，该值由 8 个数据位和 4 个错误标志组成，而写操作则将 8 位数据放入发送 FIFO。

不在复位状态下时，两个 FIFO 均处于禁用状态，并充当 1 字节深的保持寄存器。可通过对 UARTLCRH 中的 FEN 位进行置位，从而启用这两个 FIFO（850 页）。

可通过 UART 标志 (UARTFR) 寄存器（见 845 页）和 UART 接收状态 (UARTRSR) 寄存器监控 FIFO 的状态。而对空、满和溢出条件的监控则是由硬件来完成的。UARTFR 寄存器包含空和满的标志 (TXFE、TXFF、RXFE 和 RXFF 位)，而 UARTRSR 寄存器则通过 OE 位指示溢出状态。如果 FIFO 被禁用，将根据 1 字节深的保持寄存器的状态设置空和满标志。

令 FIFO 产生中断的触发点是通过 UART 中断 FIFO 深度选择 (UARTIFLS) 寄存器（见 856 页）来控制的。可将两个 FIFO 分别配置为在不同的深度触发中断。可选深度配置包括  $\frac{1}{8}$ 、 $\frac{1}{4}$ 、 $\frac{1}{2}$ 、 $\frac{3}{4}$  和  $\frac{7}{8}$ 。例如，为接收 FIFO 选择了  $\frac{1}{4}$ ，UART 则会在接收了 4 个数据字节之后产生接收中断。不在复位状态下时，两个 FIFO 都被配置为在  $\frac{1}{2}$  的深度触发中断。

### 14.3.10 中断

在观察到以下情况时，UART 会产生中断：

- 溢出错误

- 中止错误
- 奇偶校验错误
- 帧错误
- 接收超时
- 发送 (在满足 UARTIFLS 寄存器中 TXIFLSEL 位所定义的条件时, 或对 UARTCTL 寄存器的 EOT 位进行置位且所有发送数据的最后 1 位已经从串行移位寄存器发出时)
- 接收 (在满足 UARTIFLS 寄存器中 RXIFLSEL 位所定义的条件时)

会对所有中断事件执行逻辑或操作, 然后再将它们发送到中断控制器, 因此 UART 一次只能向控制器发送一个中断请求。通过读取 UART 屏蔽中断状态 (UARTMIS) 寄存器 (见 865 页), 软件可以在一个中断服务例程中处理多个中断事件。

对 UART 中断屏蔽 (UARTIM) 寄存器 (见 858 页) 中相应的 IM 位进行置位, 可以定义能够触发控制器级别中断的中断事件。如果不使用中断, 通过 UART 原始中断状态 (UARTRIS) 寄存器 (见 861 页) 可始终查看到原始中断状态。

向 UART 中断清除 (UARTICR) 寄存器 (见 869 页) 的相应位写 1, 即可 (为 UARTMIS 寄存器和 UARTRIS 寄存器) 清除中断。

当接收 FIFO 不为空时接收超时中断有效, 在 32 位周期内不再接收数据。当 FIFO 读取了所有数据 (或读取了保持寄存器) 后变为空时, 或将 1 写入 UARTICR 寄存器的相应位时, 接收超时中断被清除。

发生以下事件之一时, 接收中断将更改状态:

- 如果 FIFO 启用且接收 FIFO 到达设置的触发级别, RXRIS 位被置位。通过从接收 FIFO 读取数据直至其低于触发级别, 或向 RXIC 位写 1 清除中断, 即可将接收中断清除。
- 如果 FIFO 禁用 (拥有一个位置的深度) 且接收数据已填充该位置, 则 RXRIS 位被置位。通过对接收 FIFO 执行一次读取, 或向 RXIC 位写 1 清除中断来清零, 即可将接收中断清除。

发生以下事件之一时, 发送中断将更改状态:

- 如果 FIFO 启用且发送 FIFO 到达设置的触发级别, TXRIS 位被置位。发送中断通过向发送 FIFO 写入数据直至其高于触发级别来清零, 或通过向 TXIC 位写 1 清除中断来清零。
- 如果 FIFO 禁用 (拥有一个位置的深度) 且发送器单个位置中无数据存在, 则 TXRIS 位被置位。发送中断通过对发送 FIFO 执行一次写入来清零, 或通过向 TXIC 位写 1 清除中断来清零。

### 14.3.11 回送操作

可对 UARTCTL 寄存器 (见 852 页) 的 LBE 位进行置位, 从而使 UART 处于内部回送模式, 以便进行诊断和调试。在回送模式下, UnTx 输出端发送的数据将被 UnRx 输入端接收。请注意, 应先对 LBE 位进行置位, 然后再启用 UART。

### 14.3.12 DMA 操作

UART 向  $\mu$ DMA 控制器接口提供独立的发送通道和接收通道。UART 的 DMA 操作通过 UART DMA 控制 (UARTDMACTL) 寄存器来启用。启用 DMA 操作后, UART 在相应的 FIFO 可以传输数据时, 向接收通道或发送通道发出 DMA 请求。对于接收通道, 只要接收 FIFO 中有数据, 就会发出单次传输请求。只要接收 FIFO 中的数据量达到或超过 UARTIFLS 寄存器中配置的 FIFO 触发水平, 就会发出突发传输请求。对于发送通道, 只要发送 FIFO 中至少有一个空位, 就会发出单次传输请求。

只要发送 FIFO 中所含的字符少于 FIFO 触发水平，就会发出突发请求。μDMA 控制器会根据 DMA 通道的配置自动处理单次和突发 DMA 传输请求。

如需为接收通道启用 DMA 操作，请对 DMA 控制 (UARTDMACTL) 寄存器中的 RXDMAE 位进行置位。如需为发送通道启用 DMA 操作，请对 UARTDMACTL 寄存器中的 TXDMAE 位进行置位。还可以将 UART 配置为在发生接收错误时，令接收通道停止使用 DMA。如果对 UARTDMACR 寄存器的 DMAERR 位进行了置位且发生接收错误，会自动禁用 DMA 接收请求。这种情况可通过清除相应的 UART 错误中断予以解除。

如果启用了 μDMA，μDMA 控制器会在传输完成后自动触发中断。此中断使用 UART 中断向量。因此，如果 UART 操作使用了中断，且启用了 DMA，UART 中断处理函数中必须包含对 μDMA 完成中断的处理。

请参见“微型直接存储器访问 (μDMA)”在 534 页以了解有关对 μDMA 控制器进行配置的更多信息。

## 14.4 初始化及配置

要启用并初始化 UART，必须执行以下步骤：

1. 使用 RCGCUART 寄存器（见 307 页）启用 UART 模块。
2. 通过 RCGCGPIO 寄存器（请参考 302 页）启用相应 GPIO 模块的时钟。欲了解需要启用哪些 GPIO 端口，请参考表 23-5 在 1280 页。
3. 为相应的管脚设置 GPIOAFSEL 位（请参考 618 页）。请参考表 23-4 在 1271 页以确定配置哪些 GPIO 端口。
4. 按所选工作模式分别配置 GPIO 的限流和/或斜率（见 620 页和 628 页）。
5. 配置 GPIOCTL 寄存器的 PMCn 位域，以将 UART 信号赋给相应的管脚（见 636 页和表 23-5 在 1280 页）。

要使用 UART，必须将 RCGCUART 寄存器（307 页）中相应的位置位，从而启用外设时钟。另外，相应 GPIO 模块的时钟也必须通过系统控制模块中的 RCGCGPIO 寄存器（302 页）来启用。欲了解需要启用哪些 GPIO 端口，请参考表 23-5 在 1280 页。

本节讨论了使用 UART 模块所需的步骤。在本例中，假定 UART 时钟为 20MHz，且所需的 UART 配置为：

- 波特率 115200
- 数据长度 8 位
- 1 个停止位
- 无奇偶校验
- 禁用 FIFO
- 无中断

由于必须先对 UARTIBRD 和 UARTFBRD 寄存器执行写入操作，然后才能对 UARTECRH 寄存器执行写入操作，所以在对 UART 进行编程时，首先要考虑的是波特率分频值 (BRD)。可以通过“波特率的产生”在 830 页中所述的等式计算出 BRD：

$$BRD = 20,000,000 / (16 * 115,200) = 10.8507$$

即 UARTIBRD 寄存器 (见 848 页) 的 DIVINT 位域应设为 10 (十进制) 或 0xA。加载到 UARTFBRD 寄存器 (见 849 页) 的值是通过以下等式得出的:

$$\text{UARTFBRD}[\text{DIVFRAC}] = \text{integer}(0.8507 * 64 + 0.5) = 54$$

如此便得到了 BRD 的值, 接着要按照以下顺序将 UART 配置写入模块:

1. 将 UARTCTL 寄存器中的 UARTEN 位清零, 以便禁用 UART。
2. 将 BRD 的整数部分写入 UARTIBRD 寄存器。
3. 将 BRD 的小数部分写入 UARTFBRD 寄存器。
4. 将所需的串行工作参数写入 UARTLCRH 寄存器 (在本例中为 0x0000.0060)。
5. 通过写 UARTCC 寄存器来配置 UART 时钟源。
6. 还可以选择配置  $\mu$ DMA 通道 (见“微型直接存储器访问 ( $\mu$ DMA)”在 534 页) 并在 UARTDMACTL 寄存器中启用 DMA 选项。
7. 对 UARTCTL 寄存器中的 UARTEN 位进行置位, 以便启用 UART。

## 14.5 寄存器映射

表 14-3 在 838 页列出了 UART 寄存器。所列偏移量是寄存器地址相对于 UART 基址的 16 进制增量, UART 基址分别为:

- UART0:0x4000.C000
- UART1:0x4000.D000
- UART2:0x4000.E000
- UART3 : 0x4000.F000
- UART4 : 0x4001.0000
- UART5 : 0x4001.1000
- UART6 : 0x4001.2000
- UART7 : 0x4001.3000

注意: 必须先启用 UART 模块的时钟, 然后才能配置寄存器 (请参考 307 页)。启用 UART 模块的时钟后, 必须有 3 个系统时钟单位的延迟, 然后才能访问 UART 模块的寄存器。

注意: 必须先禁用 UART (见 852 页“UARTCTL 寄存器”中对 UARTEN 位的说明), 然后才能重新对控制寄存器进行编程。如果在进行 TX 或 RX 操作期间禁用 UART, 则会先完成当前处理, 然后再停止 UART。

表 14-3. UART 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	UARTDR	R/W	0x0000.0000	UART 数据寄存器	840
0x004	UARTRSR/UARTECR	R/W	0x0000.0000	UART 接收状态/错误清除寄存器	842
0x018	UARTFR	RO	0x0000.0090	UART 标志寄存器	845
0x020	UARTILPR	R/W	0x0000.0000	UART IrDA 低功耗寄存器	847
0x024	UARTIBRD	R/W	0x0000.0000	UART 波特率分频值整数寄存器	848
0x028	UARTFBRD	R/W	0x0000.0000	UART 波特率分频值小数寄存器	849

表 14-3. UART 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x02C	UARTLCRH	R/W	0x0000.0000	UART 线控寄存器	850
0x030	UARTCTL	R/W	0x0000.0300	UART 控制寄存器	852
0x034	UARTIFLS	R/W	0x0000.0012	UART 中断 FIFO 深度选择寄存器	856
0x038	UARTIM	R/W	0x0000.0000	UART 中断屏蔽寄存器	858
0x03C	UARTRIS	RO	0x0000.000F	UART 原始中断状态寄存器	861
0x040	UARTMIS	RO	0x0000.0000	UART 屏蔽中断状态寄存器	865
0x044	UARTICR	W1C	0x0000.0000	UART 中断清除寄存器	869
0x048	UARTDMACTL	R/W	0x0000.0000	UART DMA 控制寄存器	871
0x090	UARTLCTL	R/W	0x0000.0000	UART LIN 控制寄存器	872
0x094	UARTLSS	RO	0x0000.0000	UART LIN 快照寄存器	873
0x098	UARTLTIM	RO	0x0000.0000	UART LIN 定时寄存器	874
0x0A4	UART9BITADDR	R/W	0x0000.0000	UART 9 位模式自身地址寄存器	875
0x0A8	UART9BITAMASK	R/W	0x0000.00FF	UART 9 位模式自身地址屏蔽寄存器	876
0xFC0	UARTPP	RO	0x0000.0003	UART 外设属性寄存器	877
0xFC8	UARTCC	R/W	0x0000.0000	UART 时钟配置寄存器	878
0xFD0	UARTPeriphID4	RO	0x0000.0000	UART 外设标识寄存器 4	879
0xFD4	UARTPeriphID5	RO	0x0000.0000	UART 外设标识寄存器 5	880
0xFD8	UARTPeriphID6	RO	0x0000.0000	UART 外设标识寄存器 6	881
0xFDC	UARTPeriphID7	RO	0x0000.0000	UART 外设标识寄存器 7	882
0xFE0	UARTPeriphID0	RO	0x0000.0060	UART 外设标识寄存器 0	883
0xFE4	UARTPeriphID1	RO	0x0000.0000	UART 外设标识寄存器 1	884
0xFE8	UARTPeriphID2	RO	0x0000.0018	UART 外设标识寄存器 2	885
0xFEC	UARTPeriphID3	RO	0x0000.0001	UART 外设标识寄存器 3	886
0xFF0	UARTPCellID0	RO	0x0000.000D	UART PrimeCell 标识寄存器 0	887
0xFF4	UARTPCellID1	RO	0x0000.00F0	UART PrimeCell 标识寄存器 1	888
0xFF8	UARTPCellID2	RO	0x0000.0005	UART PrimeCell 标识寄存器 2	889
0xFFC	UARTPCellID3	RO	0x0000.00B1	UART PrimeCell 标识寄存器 3	890

## 14.6 寄存器描述

本小节按照地址偏移量的数字顺序列出并描述了 UART 寄存器。

## 寄存器 1: UART 数据寄存器 (UARTDR), 偏移量 0x000

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

该寄存器是数据寄存器 (FIFO 的接口)。

对于传输的数据来说, 如果启用了 FIFO, 则写入该位置的数据被移入发送 FIFO。如果 FIFO 被禁用, 数据将存放在发送保持寄存器 (发送 FIFO 底位字) 中。对该寄存器进行写操作会启动一个 UART 发送操作。

对于接收到的数据来说, 如果启用了 FIFO, 数据字节和 4 位状态 (中止、帧、奇偶校验和溢出) 被移入 12 位宽的接收 FIFO。如果 FIFO 被禁用, 数据字节和状态则存放在接收保持寄存器 (接收 FIFO 底位字) 中。读取该寄存器可以检索到接收的数据。

## UART 数据寄存器 (UARTDR)

UART0:0x4000.C000 (0x4000.CFFF 的结束地址) 基址: 0x4000.C000  
 UART1:0x4000.D000 (0x4000.DFFF 的结束地址) 基址: 0x4000.D000  
 UART2:0x4000.E000 (0x4000.EFFF 的结束地址) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x000  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				OE	BE	PE	FE	DATA							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
11	OE	RO	0	UART 溢出错误  值 描述 1 FIFO 已满时接收到新数据, 导致数据丢失。 0 没有因 FIFO 溢出而导致数据丢失。
10	BE	RO	0	UART 中止错误  值 描述 1 检测到中止条件, 表示接收数据输入处于低电平的时间比一个完整字的传输时间 (定义为起始位、数据位、奇偶校验位和停止位) 还要长。 0 没有出现中止条件  在 FIFO 模式下, 该错误与 FIFO 顶部的字符有关。发生中止时, 只有一个 0 字符被加载到 FIFO。仅在接收数据输入变为 1 (标记状态) 且接收到下一个有效的起始位后, 才启用下一字符。



位/域	名称	类型	复位	描述
9	PE	RO	0	<p>UART 奇偶校验错误</p> <p>值 描述</p> <p>1 所接收数据字符的奇偶与 UARTLCRH 寄存器中第 2 位和第 7 位所定义的奇偶校验不匹配。</p> <p>0 未发生奇偶校验错误</p> <p>在 FIFO 模式下，该错误与 FIFO 顶部的字符有关。</p>
8	FE	RO	0	<p>UART 帧错误</p> <p>值 描述</p> <p>1 接收的字符未含有效的停止位（有效的停止位为 1）。</p> <p>0 未发生帧错误</p>
7:0	DATA	R/W	0x00	<p>发送或接收的数据</p> <p>要通过 UART 发送的数据应写入此位域。</p> <p>读取时，该位域包含 UART 所接收的数据。</p>

## 寄存器 2: UART 接收状态/错误清除寄存器 ( UARTRSR/UARTECR ) , 偏移量 0x004

UARTRSR/UARTECR 寄存器分别为接收状态寄存器和错误清除寄存器。

接收状态除了可以从 UARTDR 寄存器中读取之外, 还可以从 UARTRSR 寄存器中读取。如果从该寄存器读取状态, 则状态信息先与从 UARTDR 读取的条目对应, 其次才读取 UARTRSR。达到溢出条件时, 会立即给置溢出状态信息。

不能对 UARTRSR 寄存器进行写入操作。

向 UARTECR 寄存器写入任何值都会将帧错误、奇偶校验错误、中止错误和溢出错误清除。在复位后所有位都会清零。

### 只读状态寄存器

#### UART 接收状态/错误清除寄存器 (UARTRSR/UARTECR)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000  
 UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000  
 UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x004  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													OE	BE	PE	FE
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

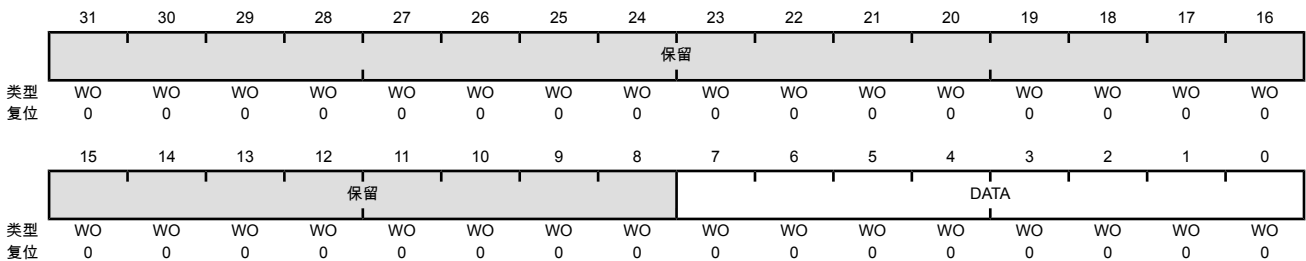
位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3	OE	RO	0	UART 溢出错误  值 描述 1 FIFO 已满时接收到新数据, 导致数据丢失。 0 没有因 FIFO 溢出而导致数据丢失。  向 UARTECR 进行写入操作会将该位清零。 由于 FIFO 已满时不再有数据写入, 所以 FIFO 的内容依然有效, 只是移位寄存器的内容被覆盖了。此时, CPU 必须读取数据以便将 FIFO 清空。

位/域	名称	类型	复位	描述
2	BE	RO	0	<p>UART 中止错误</p> <p><b>值 描述</b></p> <p>1 检测到中止条件，表示接收数据输入处于低电平的时间比一个完整字的传输时间（定义为起始位、数据位、奇偶校验位和停止位）还要长。</p> <p>0 没有出现中止条件</p> <p>向 UARTECR 进行写操作会将该位清零。</p> <p>在 FIFO 模式下，该错误与 FIFO 顶部的字符有关。发生中止时，只有一个 0 字符被加载到 FIFO。仅在接收数据输入变为 1（标记状态）且接收到下一个有效的起始位后，才启用下一字符。</p>
1	PE	RO	0	<p>UART 奇偶校验错误</p> <p><b>值 描述</b></p> <p>1 所接收数据字符的奇偶与 UARTECR 寄存器中第 2 位和第 7 位所定义的奇偶校验不匹配。</p> <p>0 未发生奇偶校验错误</p> <p>向 UARTECR 进行写操作会将该位清零。</p>
0	FE	RO	0	<p>UART 帧错误</p> <p><b>值 描述</b></p> <p>1 接收的字符未含有效的停止位（有效的停止位为 1）。</p> <p>0 未发生帧错误</p> <p>向 UARTECR 进行写操作会将该位清零。</p> <p>在 FIFO 模式下，该错误与 FIFO 顶部的字符有关。</p>

## 只写错误清除寄存器

### UART 接收状态/错误清除寄存器 (UARTRSR/UARTECR)

UART0:0x4000.C000 (0x4000.CFFF 的结束地址) 基址: 0x4000.C000  
 UART1:0x4000.D000 (0x4000.DFFF 的结束地址) 基址: 0x4000.D000  
 UART2:0x4000.E000 (0x4000.EFFF 的结束地址) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x004  
 类型 WO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:8	保留	WO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DATA	WO	0x00	错误清除 将任意数据写入该寄存器会将帧、奇偶校验、中止和溢出标志清零。

### 寄存器 3: UART 标志寄存器 (UARTFR), 偏移量 0x018

UARTFR 寄存器是标志寄存器。复位后, TXFF、RXFF 和 BUSY 位均为 0; 而 TXFE 和 RXFE 位均为 1。RI、DCD、DSR 和 CTS 位指示调制解调器流控及状态。请注意, 调制解调器位仅对 UART1 有效, 对于 UART0 和 UART2, 该位是保留的。

#### UART 标志寄存器 (UARTFR)

UART0:0x4000.C000 (0x4000.CFFF 的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF 的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF 的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x018

类型 RO, 复位 0x0000.0090

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留								RI	TXFE	RXFF	TXFF	RXFE	BUSY	DCD	DSR	CTS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	

位/域	名称	类型	复位	描述
31:9	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
8	RI	RO	0	振铃指示  值 描述 1 U1RI 信号有效。 0 U1RI 信号无效。  该位仅对 UART1 有效, 对于 UART0 和 UART2, 该位是保留的。
7	TXFE	RO	1	UART 发送 FIFO 为空 该位的具体意思取决于 UARTLCRH 寄存器中 FEN 位的状态。  值 描述 1 如果 FIFO 处于禁用状态 (FEN 为 0), 则发送保持寄存器为空。 如果 FIFO 处于启用状态 (FEN 为 1), 则发送 FIFO 为空。 0 发送器有数据要发送。
6	RXFF	RO	0	UART 接收 FIFO 已满 该位的具体意思取决于 UARTLCRH 寄存器中 FEN 位的状态。  值 描述 1 如果 FIFO 处于禁用状态 (FEN 为 0), 则接收保持寄存器为满。 如果 FIFO 处于启用状态 (FEN 为 1), 则接收 FIFO 为满。 0 接收器可以接收数据。

位/域	名称	类型	复位	描述
5	TXFF	RO	0	<p>UART 发送 FIFO 已满</p> <p>该位的具体意思取决于 UARTLCRH 寄存器中 FEN 位的状态。</p> <p>值 描述</p> <p>1 如果 FIFO 处于禁用状态 ( FEN 为 0 ) , 则发送保持寄存器为满。 如果 FIFO 处于启用状态 ( FEN 为 1 ) , 则发送 FIFO 为满。</p> <p>0 发送器未满。</p>
4	RXFE	RO	1	<p>UART 接收 FIFO 为空</p> <p>该位的具体意思取决于 UARTLCRH 寄存器中 FEN 位的状态。</p> <p>值 描述</p> <p>1 如果 FIFO 处于禁用状态 ( FEN 为 0 ) , 则接收保持寄存器为空。 如果 FIFO 处于启用状态 ( FEN 为 1 ) , 则接收 FIFO 为空。</p> <p>0 接收器不为空。</p>
3	BUSY	RO	0	<p>UART 忙</p> <p>值 描述</p> <p>1 UART 正忙于发送数据。该位一直处于置位状态, 直至包括所有停止位在内的全部字节都从移位寄存器发出为止。</p> <p>0 UART 未处于忙状态。</p> <p>只要发送 FIFO 不为空 ( 不管是否启用了 UART ) , 就会对该位进行置位。</p>
2	DCD	RO	0	<p>数据载波检测</p> <p>值 描述</p> <p>1 发出 U1DCD 信号。</p> <p>0 没有发出 U1DCD 信号。</p> <p>该位仅对 UART1 有效, 对于 UART0 和 UART2, 该位是保留的。</p>
1	DSR	RO	0	<p>数据集就绪</p> <p>值 描述</p> <p>1 发出 U1DSR 信号。</p> <p>0 没有发出 U1DSR 信号。</p> <p>该位仅对 UART1 有效, 对于 UART0 和 UART2, 该位是保留的。</p>
0	CTS	RO	0	<p>允许发送</p> <p>值 描述</p> <p>1 发出 U1CTS 信号。</p> <p>0 没有发出 U1CTS 信号。</p> <p>该位仅对 UART1 有效, 对于 UART0 和 UART2, 该位是保留的。</p>

## 寄存器 4: UART IrDA 低功耗寄存器 (UARTILPR), 偏移量 0x020

UARTILPR 寄存器保存 8 位的低功耗计数器分频值, 该值用于将系统时钟 (SysClk) 分频, 从而得到低功耗 SIR 脉宽所需的时钟。它的所有位在复位后都会被清零。

根据写入 UARTILPR 中的低功耗分频值对 SysClk 进行分频, 从而产生 IrLPBaud16 内部时钟。启用低功耗模式时, SIR 脉冲的宽度是 IrLPBaud16 时钟周期的三倍。低功耗分频值根据以下等式计算:

$$ILPDVSR = \text{SysClk} / F_{\text{IrLPBaud16}}$$

其中  $F_{\text{IrLPBaud16}}$  标称值为 1.8432 MHz。

配置分频值时必须保证  $1.42 \text{ MHz} < F_{\text{IrLPBaud16}} < 2.12 \text{ MHz}$ , 从而使低功耗脉宽保持在 1.41–2.11  $\mu\text{s}$  范围内 (IrLPBaud16 时钟周期的三倍)。IrLPBaud16 的最低频率须确保脉冲少于一个 IrLPBaud16 时钟周期时被忽略, 但高于 1.4 $\mu\text{s}$  时则为有效脉冲。

注意: 0 是非法值。如果设定值 0, 将不会产生 IrLPBaud16 脉冲。

### UART IrDA 低功耗寄存器 (UARTILPR)

UART0:0x4000.C000 (0x4000.CFFF 的结束地址) 基址: 0x4000.C000  
 UART1:0x4000.D000 (0x4000.DFFF 的结束地址) 基址: 0x4000.D000  
 UART2:0x4000.E000 (0x4000.EFFF 的结束地址) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x020  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								ILPDVSR							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	ILPDVSR	R/W	0x00	IrDA 低功耗分频值 此位域包含 8 位低功耗分频值。

**寄存器 5: UART 波特率分频值整数寄存器 ( UARTIBRD ) , 偏移量 0x024**

UARTIBRD 寄存器是波特率分频值的整数部分。在复位后所有位都会清零。允许的最小分频比为 1 (当 UARTIBRD=0 时), 在这种情况下会忽略 UARTFBRD 寄存器。更改 UARTIBRD 寄存器时, 除非当前字符的发送/接收已经结束, 否则新值不会生效。更改完波特率分频值后, 必须再写入 UARTLCRH 寄存器。详细配置信息请参考“波特率的产生”在 830 页。

**UART 波特率分频值整数寄存器 (UARTIBRD)**

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x024

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DIVINT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	DIVINT	R/W	0x0000	波特率分频值的整数部分



## 寄存器 6: UART 波特率分频值小数寄存器 (UARTFBRD), 偏移量 0x028

UARTFBRD 寄存器是波特率分频值的小数部分。在复位后所有位都会清零。更改 UARTFBRD 寄存器时, 除非当前字符的发送/接收已经结束, 否则新值不会生效。更改完波特率分频值后, 必须再写入 UARTLCRH 寄存器。详细配置信息请参考“波特率的产生”在 830页。

### UART 波特率分频值小数寄存器 (UARTFBRD)

UART0:0x4000.C000 (0x4000.CFFF的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x028

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										DIVFRAC					
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:6	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5:0	DIVFRAC	R/W	0x0	波特率分频值的小数部分

## 寄存器 7: UART 线控寄存器 (UARTLCRH), 偏移量 0x02C

UARTLCRH 寄存器是线控寄存器。数据长度、奇偶校验和停止位选择等串行通信参数都是通过本寄存器给置的。

更新波特率分频值 (UARTIBRD 和/或 UARTIFRD) 时, 还必须对 UARTLCRH 寄存器执行写入操作。波特率分频值寄存器的写入选通信号与 UARTLCRH 寄存器相关联。

## UART 线控寄存器 (UARTLCRH)

UART0:0x4000.C000 (0x4000.CFFF的结束地址) 基址: 0x4000.C000  
 UART1:0x4000.D000 (0x4000.DFFF的结束地址) 基址: 0x4000.D000  
 UART2:0x4000.E000 (0x4000.EFFF的结束地址) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x02C  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								SPS	WLEN		FEN	STP2	EPS	PEN	BRK
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	SPS	R/W	0	UART 粘着奇偶校验位选择 如果对 UARTLCRH 的第 1、2 和 7 位进行了置位, 则发送奇偶校验位, 且校验结果为 0。如果对第 1 和 7 位进行了置位且第 2 位清零, 则发送奇偶校验位, 且校验结果为 1。 该位清零时, 粘着奇偶校验处于禁用状态。
6:5	WLEN	R/W	0x0	UART 字长度 这些位指示每帧中发送或接收多少个数据位, 具体如下:  值 描述 0x0 5 个数据位 (默认) 0x1 6 个数据位 0x2 7 个数据位 0x3 8 个数据位
4	FEN	R/W	0	UART 启用 FIFO  值 描述 1 启用发送和接收 FIFO 缓冲区 (FIFO 模式)。 0 禁用 FIFO (字符模式)。FIFO 变成 1 字节深的保持寄存器。

位/域	名称	类型	复位	描述
3	STP2	R/W	0	<p>UART 双停止位选择</p> <p>值 描述</p> <p>1 每帧传输结束时发送双停止位。接收逻辑不会检查接收的是否为双停止位。</p> <p>在 7816 智能卡模式下 ( 对 UARTCTL 寄存器的 SMART 位进行置位 ) , 强制采用 2 个停止位。</p> <p>0 每帧传输结束时发送 1 个停止位。</p>
2	EPS	R/W	0	<p>UART 偶校验位选择</p> <p>值 描述</p> <p>1 在发送和接收期间会生成并进行偶校验, 从而检查数据位和奇偶校验位中“1”的个数是否为偶数。</p> <p>0 执行奇校验, 检查“1”的个数是否为奇数。</p> <p>如果通过 PEN 位禁用奇偶校验, 则该位无效。</p>
1	PEN	R/W	0	<p>启用 UART 奇偶校验</p> <p>值 描述</p> <p>1 启用奇偶校验位的生成和校验。</p> <p>0 禁用奇偶校验, 且没有将奇偶校验位加入数据帧。</p>
0	BRK	R/W	0	<p>UART 发送中止</p> <p>值 描述</p> <p>1 当前字符发送完毕后, UnTx 信号的输出一直处于低电平状态。为了正确执行中止命令, 软件必须令该位处于置位状态, 并且持续至少 2 帧 ( 字符周期 ) 。</p> <p>0 正常使用。</p>

**寄存器 8: UART 控制寄存器 ( UARTCTL ) ， 偏移量 0x030**

UARTCTL 寄存器是控制寄存器。除了启用发送位 (TXE) 和启用接收位 (RXE) 处于置位状态外，其他所有位都在复位后清零。

要想启用 UART 模块，必须对 UARTEN 位进行置位。如果软件需要更改模块的配置，则必须先将 UARTEN 位清零，然后才能写入配置更改。如果在进行发送或接收操作期间禁用 UART，则会先完成当前处理，然后再停止 UART。

请注意，[15:14,11:10] 位仅对 UART1 有效。对于 UART0 和 UART2，这些位是保留的。

**注意：** 启用 UART 后便不得修改 UARTCTL 寄存器，否则会产生无法预料的后果。建议按照以下顺序对 UARTCTL 寄存器进行更改。

1. 禁用 UART。
2. 等待当前字符发送或接收完毕。
3. 将线控寄存器 (UARTLCRH) 的第 4 位 (FEN) 清零，清空发送 FIFO。
4. 重新设定控制寄存器的内容。
5. 启用 UART。

**UART 控制寄存器 (UARTCTL)**

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000  
 UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000  
 UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x030  
 类型 R/W, 复位 0x0000.0300

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTSEN	RTSEN	保留		RTS	DTR	RXE	TXE	LBE	LIN	HSE	EOT	SMART	SIRLP	SIREN	UARTEN
类型	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15	CTSEN	R/W	0	启用允许发送  <b>值 描述</b> 1 启用 CTS 硬件流控。仅当发出 U1CTS 信号时才发送数据。 0 禁用 CTS 硬件流控。  该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。

位/域	名称	类型	复位	描述
14	RTSEN	R/W	0	<p>启用请求发送</p> <p>值 描述</p> <p>1 启用 RTS 硬件流控。仅当接收 FIFO 当前有条目时，方可（通过发出 U1RTS）来请求数据。</p> <p>0 禁用 RTS 硬件流控。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>
13:12	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
11	RTS	R/W	0	<p>请求发送</p> <p>RTSEN 清零后，该位的状态通过 U1RTS 信号反映。如果对 RTSEN 进行置位，则进行写入操作时会忽略该位，进行读取操作时也应忽略该位。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>
10	DTR	R/W	0	<p>数据终端就绪信号</p> <p>该位给置 U1DTR 输出的状态。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>
9	RXE	R/W	1	<p>启用 UART 接收</p> <p>值 描述</p> <p>1 启用 UART 的接收部分。</p> <p>0 禁用 UART 的接收部分。</p> <p>如果 UART 在接收中途被禁用，它会先接收完当前字符，然后再停止。</p> <p>注意：要启用接收，还必须对 UARTEN 位进行置位。</p>
8	TXE	R/W	1	<p>启用 UART 发送</p> <p>值 描述</p> <p>1 启用 UART 的发送部分。</p> <p>0 禁用 UART 的发送部分。</p> <p>如果 UART 在发送中途被禁用，它会先发送完当前字符，然后再停止。</p> <p>注意：要启用发送，还必须对 UARTEN 位进行置位。</p>
7	LBE	R/W	0	<p>启用 UART 回送</p> <p>值 描述</p> <p>1 UnTx 路径馈送到 UnRx 路径上。</p> <p>0 正常工作。</p>
6	LIN	R/W	0	<p>启用 LIN 模式</p> <p>值 描述</p> <p>1 UART 在 LIN 模式下工作。</p> <p>0 正常工作。</p>

位/域	名称	类型	复位	描述
5	HSE	R/W	0	<p>启用高速</p> <p>值 描述</p> <p>0 UART 以系统时钟的 16 分频工作。</p> <p>1 UART 以系统时钟的 8 分频工作。</p> <p>注意: 使用的系统时钟也取决于波特率分频值配置 ( 见 848页 ) 和 849 页 ) 。</p> <p>在 ISO 7816 智能卡模式 ( 对 SMART 位进行置位 ) 下, 该位的状态对时钟产生没有影响。</p>
4	EOT	R/W	0	<p>传输结束</p> <p>该位确定 UARTRIS 寄存器中 TXRIS 位的操作。</p> <p>值 描述</p> <p>1 仅当包括停止位在内的所有待发送数据都从串行移位寄存器中发送出去后, 才会对 TXRIS 位进行置位。</p> <p>0 满足 UARTIFLS 中指定的发送 FIFO 条件时, 对 TXRIS 位进行置位。</p>
3	SMART	R/W	0	<p>支持 ISO 7816 智能卡</p> <p>值 描述</p> <p>1 UART 在智能卡模式下工作。</p> <p>0 正常工作。</p> <p>使用 ISO 7816 模式时, 应用程序必须确保将 UARTLCRH 设置为 8 位字长 ( 将 WLEN 设置为 0x3 ) 且进行偶校验 ( 将 PEN 设置为 1, EPS 设置为 1, SPS 设置为 0 ) 。</p> <p>在此模式下, 会忽略 UARTLCRH 中 STP2 位的值并强制采用 2 个停止位。请注意, 在出现奇偶校验错误的情况下, UART 并不支持自动重新发送。如果在发送过程中检测到奇偶校验错误, 会中止所有后续发送操作, 必须由软件来处理受影响字节或报文的重新发送。</p>
2	SIRLP	R/W	0	<p>UART SIR 低功耗模式</p> <p>该位用来选择 IrDA 编码模式。</p> <p>值 描述</p> <p>1 UART 在 SIR 低功耗模式下工作。无论选定的比特率是多少, 低电平位将以 3 倍于 IrLPBaud16 输入信号周期的脉宽发送。</p> <p>0 低电平位将作为一个活动高电平脉冲发送, 其脉宽为位周期的 3/16。</p> <p>对该位进行置位可降低功率消耗, 但可能会缩短发送距离。请参考 847 页获取更多信息。</p>
1	SIREN	R/W	0	<p>启用 UART SIR</p> <p>值 描述</p> <p>1 启用 IrDA SIR 模块, 且 UART 将使用 SIR 协议来发送和接收数据。</p> <p>0 正常工作。</p>

---

位/域	名称	类型	复位	描述
0	UARTEN	R/W	0	启用 UART  值 描述 1 启用 UART。 0 禁用 UART。  如果 UART 在发送或接收中途被禁用，它会先处理完当前字符，然后再停止。

### 寄存器 9: UART 中断 FIFO 深度选择寄存器 ( UARTIFLS ) , 偏移量 0x034

UARTIFLS 寄存器是中断 FIFO 深度选择寄存器。可使用该寄存器来定义 UARTRIS 寄存器中 TXRIS 位和 RXRIS 位触发时的 FIFO 深度。

中断是在 FIFO 深度从不满足触发条件到满足触发条件的跳变沿产生的。也就是说, FIFO 所装的数据量 ( 深度 ) 超过规定触发深度时, 就产生中断。例如, 接收 FIFO 的触发深度如果设为 1/2, 则在模块接收到第 9 个字符时触发中断。

不在复位状态下时, 会将 TXIFLSEL 位和 RXIFLSEL 位配置为使 FIFO 以 1/2 为触发深度触发中断。

#### UART 中断 FIFO 深度选择寄存器 (UARTIFLS)

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000  
 UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000  
 UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x034  
 类型 R/W, 复位 0x0000.0012

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										RXIFLSEL			TXIFLSEL		
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

位/域	名称	类型	复位	描述														
31:6	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。														
5:3	RXIFLSEL	R/W	0x2	UART 接收中断 FIFO 深度选择 接收中断的触发深度设置如下:  <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>RX FIFO ≥ 1/8 满</td> </tr> <tr> <td>0x1</td> <td>RX FIFO ≥ 1/4 满</td> </tr> <tr> <td>0x2</td> <td>RX FIFO ≥ 1/2 满 ( 默认 )</td> </tr> <tr> <td>0x3</td> <td>RX FIFO ≥ 3/4 满</td> </tr> <tr> <td>0x4</td> <td>RX FIFO ≥ 7/8 满</td> </tr> <tr> <td>0x5-0x7</td> <td>保留</td> </tr> </tbody> </table>	值	描述	0x0	RX FIFO ≥ 1/8 满	0x1	RX FIFO ≥ 1/4 满	0x2	RX FIFO ≥ 1/2 满 ( 默认 )	0x3	RX FIFO ≥ 3/4 满	0x4	RX FIFO ≥ 7/8 满	0x5-0x7	保留
值	描述																	
0x0	RX FIFO ≥ 1/8 满																	
0x1	RX FIFO ≥ 1/4 满																	
0x2	RX FIFO ≥ 1/2 满 ( 默认 )																	
0x3	RX FIFO ≥ 3/4 满																	
0x4	RX FIFO ≥ 7/8 满																	
0x5-0x7	保留																	



位/域	名称	类型	复位	描述
2:0	TXIFLSEL	R/W	0x2	UART 发送中断 FIFO 深度选择 发送中断的触发深度设置如下：
				值 描述
				0x0 TX FIFO $\leq \frac{1}{8}$ 空
				0x1 TX FIFO $\leq \frac{1}{4}$ 空
				0x2 TX FIFO $\leq \frac{1}{2}$ 空 (默认)
				0x3 TX FIFO $\leq \frac{3}{4}$ 空
				0x4 TX FIFO $\leq \frac{7}{8}$ 空
				0x5-0x7 保留

**注意:** 如果对 UARTCTL (见 852页) 中的 EOT 位进行置位, 只要 FIFO 完全变空且包括停止位在内的所有数据均从发送串行移位器发出, 就会产生发送中断。在这种情况下, 会忽略 TXIFLSEL 的设置。

**寄存器 10: UART 中断屏蔽寄存器 ( UARTIM ) ， 偏移量 0x038**

UARTIM 寄存器是中断屏蔽设置/清除寄存器。

读取时，该寄存器会给出相关中断屏蔽位的当前值。对某个位进行置位时，可将相应的原始中断信号传递到中断控制器。将某个位清零时，可阻止该原始中断信号传递到中断控制器。

请注意，[3:0] 位仅对 UART1 有效。对于 UART0 和 UART2，这些位是保留的。

**UART 中断屏蔽寄存器 (UARTIM)**

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x038

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LME5IM	LME1IM	LMSBIM	9BITIM	保留	OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	DSRIM	DCDIM	CTSIM	RIIM
类型	R/W	R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
15	LME5IM	R/W	0	LIN 模式第 5 沿中断屏蔽  值 描述 1 对 UARTRIS 寄存器的 LME5RIS 位进行了置位时，会向中断控制器发送中断。 0 LME5RIS 中断被抑制，不会发送到中断控制器。
14	LME1IM	R/W	0	LIN 模式第 1 沿中断屏蔽  值 描述 1 对 UARTRIS 寄存器的 LME1RIS 位进行了置位时，会向中断控制器发送中断。 0 LME1RIS 中断不会发送到中断控制器。
13	LMSBIM	R/W	0	LIN 模式同步间隔中断屏蔽  值 描述 1 对 UARTRIS 寄存器的 LMSBRIS 位进行了置位时，会向中断控制器发送中断。 0 LMSBRIS 中断不会发送到中断控制器。

位/域	名称	类型	复位	描述
12	9BITIM	R/W	0	<p>9 位模式中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 9BITRIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 9BITRIS 中断不会发送到中断控制器。</p>
11	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
10	OEIM	R/W	0	<p>UART 溢出错误中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 OERIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 OERIS 中断不会发送到中断控制器。</p>
9	BEIM	R/W	0	<p>UART 中止错误中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 BERIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 BERIS 中断不会发送到中断控制器。</p>
8	PEIM	R/W	0	<p>UART 奇偶校验错误中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 PERIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 PERIS 中断不会发送到中断控制器。</p>
7	FEIM	R/W	0	<p>UART 帧错误中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 FERIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 FERIS 中断不会发送到中断控制器。</p>
6	RTIM	R/W	0	<p>UART 接收超时中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 RTRIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 RTRIS 中断不会发送到中断控制器。</p>

位/域	名称	类型	复位	描述
5	TXIM	R/W	0	<p>UART 发送中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 TXRIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 TXRIS 中断不会发送到中断控制器。</p>
4	RXIM	R/W	0	<p>UART 接收中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 RXRIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 RXRIS 中断不会发送到中断控制器。</p>
3	DSRIM	R/W	0	<p>UART 数据集就绪调制解调器中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 DSRRIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 DSRRIS 中断不会发送到中断控制器。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>
2	DCDIM	R/W	0	<p>UART 数据载波检测调制解调器中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 DCDRIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 DCDRIS 中断不会发送到中断控制器。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>
1	CTSIM	R/W	0	<p>UART 允许发送调制解调器中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 CTSRIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 CTSRIS 中断不会发送到中断控制器。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>
0	RIIM	R/W	0	<p>UART 振铃指示调制解调器中断屏蔽</p> <p>值 描述</p> <p>1 对 UARTRIS 寄存器的 RIRIS 位进行了置位时，会向中断控制器发送中断。</p> <p>0 RIRIS 中断不会发送到中断控制器。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>

## 寄存器 11: UART 原始中断状态寄存器 (UARTRIS), 偏移量 0x03C

UARTRIS 寄存器是原始中断状态寄存器。读取时, 该寄存器给出相应中断的当前原始状态值。写操作对本寄存器无效。

请注意, [3:0] 位仅对 UART1 有效。对于 UART0 和 UART2, 这些位是保留的。

### UART 原始中断状态寄存器 (UARTRIS)

UART0:0x4000.C000 (0x4000.CFFF 的结束地址) 基址: 0x4000.C000  
 UART1:0x4000.D000 (0x4000.DFFF 的结束地址) 基址: 0x4000.D000  
 UART2:0x4000.E000 (0x4000.EFFF 的结束地址) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x03C  
 类型 RO, 复位 0x0000.000F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LME5RIS	LME1RIS	LMSBRIS	9BITRIS	保留	OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS	DSRRIS	DCDRIS	CTSRIS	RIRIS
类型	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
15	LME5RIS	RO	0	LIN 模式第 5 沿原始中断状态  值 描述 1 捕获了 LIN 同步域第 5 个下降沿的定时器值。 0 无中断  向 UARTICR 寄存器的 LME5IC 位写入 1 即可将该位清零。
14	LME1RIS	RO	0	LIN 模式第 1 沿原始中断状态  值 描述 1 捕获了 LIN 同步域的第 1 个下降沿的定时器值。 0 无中断  向 UARTICR 寄存器的 LME1IC 位写入 1 即可将该位清零。
13	LMSBRIS	RO	0	LIN 模式同步间隔原始中断状态  值 描述 1 检测到 LIN 同步间隔。 0 无中断  向 UARTICR 寄存器的 LMSBIC 位写入 1 即可将该位清零。

位/域	名称	类型	复位	描述
12	9BITRIS	R/W	0	<p>9 位模式原始中断状态</p> <p>值 描述</p> <p>1 出现了接收地址匹配。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 9BITIC 位写入 1 即可将该位清零。</p>
11	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
10	OERIS	RO	0	<p>UART 溢出错误原始中断状态</p> <p>值 描述</p> <p>1 发生了溢出错误。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 OEIC 位写入 1 即可将该位清零。</p>
9	BERIS	RO	0	<p>UART 中止错误原始中断状态</p> <p>值 描述</p> <p>1 发生了中止错误。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 BEIC 位写入 1 即可将该位清零。</p>
8	PERIS	RO	0	<p>UART 奇偶校验错误原始中断状态</p> <p>值 描述</p> <p>1 发生了奇偶校验错误。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 PEIC 位写入 1 即可将该位清零。</p>
7	FERIS	RO	0	<p>UART 帧错误原始中断状态</p> <p>值 描述</p> <p>1 发生了帧错误。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 FEIC 位写入 1 即可将该位清零。</p>
6	RTRIS	RO	0	<p>UART 接收超时原始中断状态</p> <p>值 描述</p> <p>1 发生接收超时。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 RTIC 位写入 1 即可将该位清零。</p>

位/域	名称	类型	复位	描述
5	TXRIS	RO	0	<p>UART 发送原始中断状态</p> <p>值 描述</p> <p>1 如果 UARTCTL 寄存器的 EOT 位清零，则表示发送 FIFO 深度已经达到 UARTIFLS 寄存器所定义的条件。</p> <p>如果对 EOT 位进行了置位，则表示所有待发送数据和标志的最后一位已经从串行移位寄存器中发出。</p> <p>0 无中断</p> <p>FIFO 启用时，通过向 UARTICR 寄存器中的 TXIC 位写 1 或向发送 FIFO 写入数据直至其高于触发深度，以将该位清零，或在 FIFO 禁用时写入一个字节将该位清零。</p>
4	RXRIS	RO	0	<p>UART 接收原始中断状态</p> <p>值 描述</p> <p>1 接收 FIFO 深度已经达到 UARTIFLS 寄存器所定义的条件。</p> <p>0 无中断</p> <p>FIFO 启用时，通过向 UARTICR 寄存器中的 RXIC 位写 1 或从接收 FIFO 读取数据直至其低于触发深度，以将该位清零；FIFO 禁用时，通过读取一个字节将该位清零。</p>
3	DSRRIS	RO	0	<p>UART 数据集就绪调制解调器原始中断状态</p> <p>值 描述</p> <p>1 用于软件流控的数据集就绪标志。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 DSRIC 位写入 1 即可将该位清零。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>
2	DCDRIS	RO	0	<p>UART 数据载波检测调制解调器原始中断状态</p> <p>值 描述</p> <p>1 用于软件流控的数据载波检测标志。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 DCDIC 位写入 1 即可将该位清零。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>
1	CTSRIS	RO	0	<p>UART 允许发送调制解调器原始中断状态</p> <p>值 描述</p> <p>1 用于软件流控的允许发送标志。</p> <p>0 无中断</p> <p>向 UARTICR 寄存器的 CTSIC 位写入 1 即可将该位清零。</p> <p>该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。</p>

位/域	名称	类型	复位	描述
0	RIRIS	RO	0	UART 振铃指示调制解调器原始中断状态  值 描述 1 用于软件流控的振铃指示。 0 无中断  向 UARTICR 寄存器的 RIIC 位写入 1 即可将该位清零。 该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。



## 寄存器 12: UART 屏蔽中断状态寄存器 (UARTMIS) , 偏移量 0x040

UARTMIS 寄存器是屏蔽中断状态寄存器。读取时, 该寄存器给出相应中断的当前屏蔽状态值。写操作对本寄存器无效。

请注意, [3:0] 位仅对 UART1 有效。对于 UART0 和 UART2, 这些位是保留的。

### UART 屏蔽中断状态寄存器 (UARTMIS)

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000  
 UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000  
 UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x040  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LME5MIS	LME1MIS	LMSBMIS	9BITMIS	保留	OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS	DSRMIS	DCDMIS	CTSMIS	RIMIS
类型	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
15	LME5MIS	RO	0	LIN 模式第 5 沿屏蔽中断状态  值 描述 1 由于对 LIN 同步域的第 5 个下降沿进行了置位, 因此发出未屏蔽的中断信号。 0 未产生中断或中断被屏蔽。  向 UARTICR 寄存器的 LME5IC 位写入 1 即可将该位清零。
14	LME1MIS	RO	0	LIN 模式第 1 沿屏蔽中断状态  值 描述 1 由于对 LIN 同步域的第 1 个下降沿进行了置位, 因此发出未屏蔽的中断信号。 0 未产生中断或中断被屏蔽。  向 UARTICR 寄存器的 LME1IC 位写入 1 即可将该位清零。
13	LMSBMIS	RO	0	LIN 模式同步间隔屏蔽中断状态  值 描述 1 由于接收到一个 LIN 同步间隔, 因此发出未屏蔽的中断信号。 0 未产生中断或中断被屏蔽。  向 UARTICR 寄存器的 LMSBIC 位写入 1 即可将该位清零。

位/域	名称	类型	复位	描述
12	9BITMIS	R/W	0	<p>9 位模式屏蔽中断状态</p> <p>值 描述</p> <p>1 由于接收地址匹配，因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 9BITIC 位写入 1 即可将该位清零。</p>
11	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>
10	OEMIS	RO	0	<p>UART 溢出错误屏蔽中断状态</p> <p>值 描述</p> <p>1 由于产生溢出错误，因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 OEIC 位写入 1 即可将该位清零。</p>
9	BEMIS	RO	0	<p>UART 中止错误屏蔽中断状态</p> <p>值 描述</p> <p>1 由于产生中止错误，因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 BEIC 位写入 1 即可将该位清零。</p>
8	PEMIS	RO	0	<p>UART 奇偶校验错误屏蔽中断状态</p> <p>值 描述</p> <p>1 由于产生奇偶校验错误，因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 PEIC 位写入 1 即可将该位清零。</p>
7	FEMIS	RO	0	<p>UART 帧错误屏蔽中断状态</p> <p>值 描述</p> <p>1 由于产生帧错误，因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 FEIC 位写入 1 即可将该位清零。</p>
6	RTMIS	RO	0	<p>UART 接收超时屏蔽中断状态</p> <p>值 描述</p> <p>1 由于接收超时，因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 RTIC 位写入 1 即可将该位清零。</p>

位/域	名称	类型	复位	描述
5	TXMIS	RO	0	<p>UART 发送屏蔽中断状态</p> <p>值 描述</p> <p>1 由于超过指定的发送 FIFO 深度 ( 如果 EOT 位清零 ) 或发送了最后一个数据位 ( 如果对 EOT 位进行了置位 ) , 因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>FIFO 启用时, 通过向 UARTICR 寄存器中的 TXIC 位写 1 或向发送 FIFO 写入数据直至其高于触发深度, 以将该位清零, 或在 FIFO 禁用时写入一个字节将该位清零。</p>
4	RXMIS	RO	0	<p>UART 接收屏蔽中断状态</p> <p>值 描述</p> <p>1 由于超过了指定的接收 FIFO 深度, 因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>FIFO 启用时, 通过向 UARTICR 寄存器中的 RXIC 位写 1 或从接收 FIFO 读取数据直至其低于触发深度, 以将该位清零; FIFO 禁用时, 通过读取一个字节将该位清零。</p>
3	DSRMIS	RO	0	<p>UART 数据集就绪调制解调器屏蔽中断状态</p> <p>值 描述</p> <p>1 由于数据集就绪, 因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 DSRIC 位写入 1 即可将该位清零。 该位仅对 UART1 有效, 对于 UART0 和 UART2, 该位是保留的。</p>
2	DCDMIS	RO	0	<p>UART 数据载波检测调制解调器屏蔽中断状态</p> <p>值 描述</p> <p>1 由于数据载波检测, 因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 DCDIC 位写入 1 即可将该位清零。 该位仅对 UART1 有效, 对于 UART0 和 UART2, 该位是保留的。</p>
1	CTSMIS	RO	0	<p>UART 允许发送调制解调器屏蔽中断状态</p> <p>值 描述</p> <p>1 由于允许发送, 因此发出未屏蔽的中断信号。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 UARTICR 寄存器的 CTSIC 位写入 1 即可将该位清零。 该位仅对 UART1 有效, 对于 UART0 和 UART2, 该位是保留的。</p>

位/域	名称	类型	复位	描述
0	RIMIS	RO	0	UART 振铃指示调制解调器屏蔽中断状态  值 描述 1 由于有振铃指示，因此发出未屏蔽的中断信号。 0 未产生中断或中断被屏蔽。  向 UARTICR 寄存器的 RIIC 位写入 1 即可将该位清零。 该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。

## 寄存器 13: UART 中断清除寄存器 (UARTICR), 偏移量 0x044

UARTICR 寄存器是中断清除寄存器。写入 1 时, 相应的中断 (包括原始中断, 如果启用了屏蔽中断, 则还包括屏蔽中断) 被清除。对本寄存器写 0 无效。

请注意, [3:0] 位仅对 UART1 有效。对于 UART0 和 UART2, 这些位是保留的。

### UART 中断清除寄存器 (UARTICR)

UART0:0x4000.C000 (0x4000.CFFF 的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF 的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF 的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x044

类型 W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LME5IC	LME1IC	LMSBIC	9BITIC	保留	OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	DSRMIC	DCDMIC	CTSMIC	RIMIC
类型	W1C	W1C	W1C	R/W	RO	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
15	LME5IC	W1C	0	LIN 模式第 5 沿中断清除 向该位写入 1, 即可将 UARTRIS 寄存器的 LME5RIS 位以及 UARTMIS 寄存器的 LME5MIS 位清零。
14	LME1IC	W1C	0	LIN 模式第 1 沿中断清除 向该位写入 1, 即可将 UARTRIS 寄存器的 LME1RIS 位以及 UARTMIS 寄存器的 LME1MIS 位清零。
13	LMSBIC	W1C	0	LIN 模式同步间隔中断清除 向该位写入 1, 即可将 UARTRIS 寄存器的 LMSBRIS 位以及 UARTMIS 寄存器的 LMSBMIS 位清零。
12	9BITIC	R/W	0	9 位模式中断清除 向该位写入 1, 即可将 UARTRIS 寄存器的 9BITRIS 位以及 UARTMIS 寄存器的 9BITMIS 位清零。
11	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
10	OEIC	W1C	0	溢出错误中断清除 向该位写入 1, 即可将 UARTRIS 寄存器的 OERIS 位以及 UARTMIS 寄存器的 OEMIS 位清零。
9	BEIC	W1C	0	中止错误中断清除 向该位写入 1, 即可将 UARTRIS 寄存器的 BERIS 位以及 UARTMIS 寄存器的 BEMIS 位清零。

位/域	名称	类型	复位	描述
8	PEIC	W1C	0	奇偶校验错误中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 PERIS 位以及 UARTMIS 寄存器的 PEMIS 位清零。
7	FEIC	W1C	0	帧错误中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 FERIS 位以及 UARTMIS 寄存器的 FEMIS 位清零。
6	RTIC	W1C	0	接收超时中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 RTRIS 位以及 UARTMIS 寄存器的 RTMIS 位清零。
5	TXIC	W1C	0	发送中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 TXRIS 位以及 UARTMIS 寄存器的 TXMIS 位清零。
4	RXIC	W1C	0	接收中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 RXRIS 位以及 UARTMIS 寄存器的 RXMIS 位清零。
3	DSRMIC	W1C	0	UART 数据集就绪调制解调器中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 DSRRIS 位以及 UARTMIS 寄存器的 DSRMIS 位清零。 该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。
2	DCDMIC	W1C	0	UART 数据载波检测调制解调器中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 DCDRIS 位以及 UARTMIS 寄存器的 DCDMIS 位清零。 该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。
1	CTSMIC	W1C	0	UART 允许发送调制解调器中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 CTSRIS 位以及 UARTMIS 寄存器的 CTSMIS 位清零。 该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。
0	RIMIC	W1C	0	UART 振铃指示调制解调器中断清除 向该位写入 1，即可将 UARTRIS 寄存器的 RIRIS 位以及 UARTMIS 寄存器的 RIMIS 位清零。 该位仅对 UART1 有效，对于 UART0 和 UART2，该位是保留的。

## 寄存器 14: UART DMA 控制寄存器 (UARTDMACTL), 偏移量 0x048

UARTDMACTL 寄存器是 DMA 控制寄存器。

## UART DMA 控制寄存器 (UARTDMACTL)

UART0:0x4000.C000 (0x4000.CFFF的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x048

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													DMAERR	TXDMAE	RXDMAE	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x00000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2	DMAERR	R/W	0	出现错误时的 DMA  值 描述 1 发生接收错误时, 自动禁用 $\mu$ DMA 接收请求。 0 接收错误不影响 $\mu$ DMA 接收请求。
1	TXDMAE	R/W	0	启用发送 DMA  值 描述 1 启用发送 FIFO 的 $\mu$ DMA 通道。 0 禁用发送 FIFO 的 $\mu$ DMA 通道。
0	RXDMAE	R/W	0	启用接收 DMA  值 描述 1 启用接收 FIFO 的 $\mu$ DMA 通道。 0 禁用接收 FIFO 的 $\mu$ DMA 通道。

## 寄存器 15: UART LIN 控制寄存器 (UARTLCTL), 偏移量 0x090

UARTLCTL 寄存器用于配置 UART 在 LIN 模式下工作的相关参数。

## UART LIN 控制寄存器 (UARTLCTL)

UART0:0x4000.C000 (0x4000.CFFF 的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF 的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF 的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x090

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										BLEN		保留			MASTER
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:6	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5:4	BLEN	R/W	0x0	同步间隔长度 值 描述 0x3 同步间隔长度为 16T 位 0x2 同步间隔长度为 15T 位 0x1 同步间隔长度为 14T 位 0x0 同步间隔长度为 13T 位 (默认值)
3:1	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	MASTER	R/W	0	启用 LIN 主机 值 描述 1 UART 作为 LIN 主机 0 UART 作为 LIN 从机



## 寄存器 16: UART LIN 快照寄存器 ( UARTLSS ) , 偏移量 0x094

UARTLSS 寄存器用于捕捉 LIN 模式下在同步域中检测到第 1 个下降沿以及第 5 个下降沿时的自由运行定时器值。

### UART LIN 快照寄存器 (UARTLSS)

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x094

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSS															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	TSS	RO	0x0000	定时器快照 此位域包含系统检测到同步域第 5 沿或同步域第 1 沿时的自由运行定时器值。

**寄存器 17: UART LIN 定时寄存器 (UARTLTIM) , 偏移量 0x098**

UARTLTIM 寄存器包含自由运行定时器的当前值, 该定时器用于在 LIN 从机模式下计算波特率。该寄存器的值与 UART LIN 快照 (UARTLSS) 寄存器的值一起用于调节波特率, 以便与主机波特率一致。

**注意:** 如果 PIOSC 被用作 UART 波特时钟 (UART 时钟配置 (UARTCC) 寄存器的 CS 域为 0x1), 则应对该寄存器中的值读取两次以确保数据正确。如果数据不匹配, 应再次读取该值直至两个连续的值匹配。将系统时钟用作 UART 波特时钟时 (CS 域为 0x0), 无需执行此步骤。

**UART LIN 定时寄存器 (UARTLTIM)**

UART0:0x4000.C000 (0x4000.CFFF的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x098

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIMER															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	TIMER	RO	0x0000	定时器值 该位域包含自由运行定时器的值。

**寄存器 18: UART 9 位模式自身地址寄存器 ( UART9BITADDR ) , 偏移量 0x0A4**

UART9BITADDR 寄存器用于写入在 9 位地址屏蔽 (UART9BITAMASK) 给置为 0xFF 时, 应与接收字节匹配的具体地址。该寄存器与 UART9BITAMASK 配合使用, 从而与接收到的地址字节匹配。

**UART 9 位模式自身地址寄存器 (UART9BITADDR)**

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0x0A4

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	9BITEN		保留						ADDR							
类型	R/W	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15	9BITEN	R/W	0	启用 9 位模式  值 描述 1 启用 9 位模式。 0 禁用 9 位模式。
14:8	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	ADDR	R/W	0x00	9 位模式的自身地址 此位域中包含在 UART9BITAMASK 给置为 0xFF 时, 应与接收字节相匹配的地址。

## 寄存器 19: UART 9 位模式自身地址屏蔽寄存器 ( UART9BITAMASK ) , 偏移量 0x0A8

UART9BITAMASK 寄存器用于为 9 位模式启用地址屏蔽。屏蔽地址位，从而创建一组将与接收到的地址字节相匹配的地址。

### UART 9 位模式自身地址屏蔽寄存器 (UART9BITAMASK)

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000  
 UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000  
 UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0x0A8  
 类型 R/W, 复位 0x0000.00FF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								MASK							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	MASK	R/W	0xFF	9 位模式的自身地址屏蔽 此位域包含地址屏蔽，可用于创建一组应匹配的地址。

## 寄存器 20: UART 外设属性寄存器 (UARTPP), 偏移量 0xFC0

UARTPP 寄存器提供有关 UART 模块属性的信息。

### UART 外设属性寄存器 (UARTPP)

UART0:0x4000.C000 (0x4000.CFFF 的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF 的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF 的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFC0

类型 RO, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															NB	SC
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
1	NB	RO	0x1	9 位模式支持  值 描述 1 UART 模块支持 9 位数据的发送, 从而支持 RS-485。 0 UART 模块不支持 9 位数据的发送, 因而不支持 RS-485。
0	SC	RO	0x1	智能卡支持  值 描述 1 UART 模块支持智能卡。 0 UART 模块不支持智能卡。

**寄存器 21: UART 时钟配置寄存器 ( UARTCC ) ， 偏移量 0xFC8**

UARTCC 寄存器控制 UART 模块的波特时钟源。更多信息，请参考“通信时钟源”一节在 211页。

注意： 如果将 PIOSC 用作 UART 波特时钟，则系统时钟频率在运行模式下必须至少为 9 MHz。

**UART 时钟配置寄存器 (UARTCC)**

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000  
 UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000  
 UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000  
 UART3 基址: 0x4000.F000  
 UART4 基址: 0x4001.0000  
 UART5 基址: 0x4001.1000  
 UART6 基址: 0x4001.2000  
 UART7 基址: 0x4001.3000  
 偏移量 0xFC8  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												CS			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述										
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。										
3:0	CS	R/W	0	UART 波特时钟源 下面的表格指定了产生 UART 波特时钟的源： <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>系统时钟 ( 基于时钟源和分频因子 )</td> </tr> <tr> <td>0x1-0x4</td> <td>保留</td> </tr> <tr> <td>0x5</td> <td>PIOSC</td> </tr> <tr> <td>0x5-0xF</td> <td>保留</td> </tr> </tbody> </table>	值	描述	0x0	系统时钟 ( 基于时钟源和分频因子 )	0x1-0x4	保留	0x5	PIOSC	0x5-0xF	保留
值	描述													
0x0	系统时钟 ( 基于时钟源和分频因子 )													
0x1-0x4	保留													
0x5	PIOSC													
0x5-0xF	保留													

## 寄存器 22: UART 外设标识寄存器 4 ( UARTPeriphID4 ) , 偏移量 0xFD0

UARTPeriphIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

## UART 外设标识寄存器 4 (UARTPeriphID4)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFD0

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID4							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID4	RO	0x00	UART 外设标识寄存器 [7:0] 可由软件用来识别该外设是否存在。

## 寄存器 23: UART 外设标识寄存器 5 ( UARTPeriphID5 ) , 偏移量 0xFD4

UARTPeriphIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

## UART 外设标识寄存器 5 (UARTPeriphID5)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFD4

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID5							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID5	RO	0x00	UART 外设标识寄存器 [15:8] 可由软件用来识别该外设是否存在。



## 寄存器 24: UART 外设标识寄存器 6 ( UARTPeriphID6 ) , 偏移量 0xFD8

UARTPeriphIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

## UART 外设标识寄存器 6 (UARTPeriphID6)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFD8

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID6							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID6	RO	0x00	UART 外设标识寄存器 [23:16] 可由软件用来识别该外设是否存在。

## 寄存器 25: UART 外设标识寄存器 7 ( UARTPeriphID7 ) , 偏移量 0xFDC

UARTPeriphIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

## UART 外设标识寄存器 7 (UARTPeriphID7)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFDC

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID7							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID7	RO	0x00	UART 外设标识寄存器 [31:24] 可由软件用来识别该外设是否存在。

## 寄存器 26: UART 外设标识寄存器 0 ( UARTPeriphID0 ) , 偏移量 0xFE0

UARTPeriphIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

### UART 外设标识寄存器 0 (UARTPeriphID0)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFE0

类型 RO, 复位 0x0000.0060

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID0	RO	0x60	UART 外设标识寄存器 [7:0] 可由软件用来识别该外设是否存在。

## 寄存器 27: UART 外设标识寄存器 1 ( UARTPeriphID1 ) , 偏移量 0xFE4

UARTPeriphIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

## UART 外设标识寄存器 1 (UARTPeriphID1)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFE4

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID1	RO	0x00	UART 外设标识寄存器 [15:8] 可由软件用来识别该外设是否存在。

## 寄存器 28: UART 外设标识寄存器 2 ( UARTPeriphID2 ) , 偏移量 0xFE8

UARTPeriphIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

## UART 外设标识寄存器 2 (UARTPeriphID2)

UART0:0x4000.C000 ( 0x4000.CFFF 的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF 的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF 的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFE8

类型 RO, 复位 0x0000.0018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID2	RO	0x18	UART 外设标识寄存器 [23:16] 可由软件用来识别该外设是否存在。

## 寄存器 29: UART 外设标识寄存器 3 ( UARTPeriphID3 ) , 偏移量 0xFEC

UARTPeriphIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

## UART 外设标识寄存器 3 (UARTPeriphID3)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFEC

类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID3	RO	0x01	UART 外设标识寄存器 [31:24] 可由软件用来识别该外设是否存在。

## 寄存器 30: UART PrimeCell 标识寄存器 0 (UARTPCellID0), 偏移量 0xFF0

UARTPCellIDn 寄存器均为硬编码寄存器, 寄存器中的各位域决定复位值。

## UART PrimeCell 标识寄存器 0 (UARTPCellID0)

UART0:0x4000.C000 (0x4000.CFFF的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFF0

类型 RO, 复位 0x0000.000D

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID0	RO	0x0D	UART PrimeCell 标志寄存器 [7:0] 向软件提供一个标准的跨外设标识系统。

## 寄存器 31: UART PrimeCell 标识寄存器 1 (UARTPCellID1), 偏移量 0xFF4

UARTPCellIDn 寄存器均为硬编码寄存器, 寄存器中的各位域决定复位值。

## UART PrimeCell 标识寄存器 1 (UARTPCellID1)

UART0:0x4000.C000 (0x4000.CFFF的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFF4

类型 RO, 复位 0x0000.00F0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID1	RO	0xF0	UART PrimeCell 标志寄存器 [15:8] 向软件提供一个标准的跨外设标识系统。



## 寄存器 32: UART PrimeCell 标识寄存器 2 ( UARTPCelIID2 ) , 偏移量 0xFF8

UARTPCelIIDn 寄存器均为硬编码寄存器，寄存器中的各位域决定复位值。

### UART PrimeCell 标识寄存器 2 (UARTPCelIID2)

UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000

UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000

UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFF8

类型 RO, 复位 0x0000.0005

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID2	RO	0x05	UART PrimeCell 标志寄存器 [23:16] 向软件提供一个标准的跨外设标识系统。

## 寄存器 33: UART PrimeCell 标识寄存器 3 (UARTPCellID3), 偏移量 0xFFC

UARTPCellIDn 寄存器均为硬编码寄存器, 寄存器中的各位域决定复位值。

## UART PrimeCell 标识寄存器 3 (UARTPCellID3)

UART0:0x4000.C000 (0x4000.CFFF的结束地址) 基址: 0x4000.C000

UART1:0x4000.D000 (0x4000.DFFF的结束地址) 基址: 0x4000.D000

UART2:0x4000.E000 (0x4000.EFFF的结束地址) 基址: 0x4000.E000

UART3 基址: 0x4000.F000

UART4 基址: 0x4001.0000

UART5 基址: 0x4001.1000

UART6 基址: 0x4001.2000

UART7 基址: 0x4001.3000

偏移量 0xFFC

类型 RO, 复位 0x0000.00B1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID3	RO	0xB1	UART PrimeCell 标志寄存器 [31:24] 向软件提供一个标准的跨外设标识系统。

## 15 同步串行接口 (SSI)

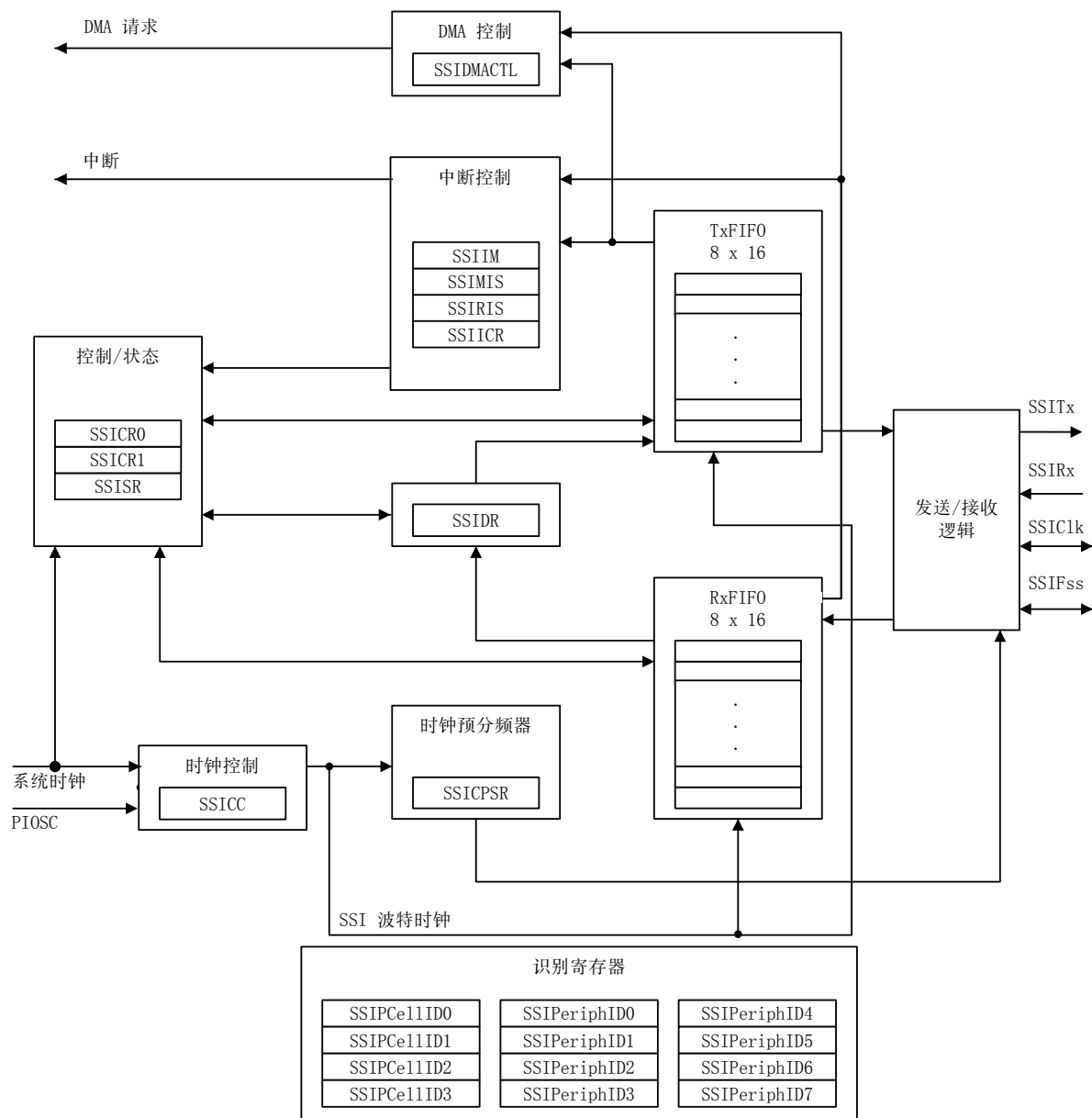
该 Stellaris® 微控制器包含四个同步串行接口 (SSI) 模块。每个 SSI 模块都能以主机或从机方式与片外器件进行同步串行通信，可以支持飞思卡尔 (Freescale) SPI、MICROWIRE 或者 Texas Instruments 的同步串行接口 (SSI)。

该 Stellaris LM4F232H5QD 控制器包括四个具有以下功能的 SSI 模块：

- 提供可编程控制的接口，可与 Freescale 的 SPI 接口、MICROWIRE 或者 Texas Instruments 同步串行接口相连
- 主机或者从机方式
- 可编程的时钟位速率以及预分频器；
- 相互独立的发送 FIFO 和接收 FIFO，二者均为 16 位宽、8 个单元深；
- 可编程决定数据帧大小，范围为 4 到 16 位
- 内部回送测试 (loopback test) 模式，可进行诊断/调试测试
- 标准 FIFO 中断以及发送结束中断；
- 用微型直接内存访问 ( $\mu$ DMA) 有效的传输数据
  - 相互独立的发送通道和接收通道
  - 当接收 FIFO 中有数据时产生单次请求；当接收 FIFO 中包含 4 个数据单元时产生猝发请求
  - 当发送 FIFO 中有空闲单元时产生单次请求；当发送 FIFO 中包含 4 个空闲单元时产生猝发请求

## 15.1 结构图

图 15-1. SSI 模块的结构图



## 15.2 信号描述

下面的表格列出了 SSI 模块的外部信号，并描述了每个信号的功能。大多数 SSI 信号是某些 GPIO 信号的备用功能，在复位的时候默认为 GPIO 信号。但 `SSI0Clk`、`SSI0Fss`、`SSI0Rx` 和 `SSI0Tx` 管脚例外，它们将默认为 SSI 功能。下表中“复用管脚/赋值”一栏列出了 SSI 信号可能的 GPIO 管脚布局。将 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618页) 中的 AFSEL 位置位，以便选择 SSI 功能。必须将括号中的数字写入 GPIO 端口控制 (GPIOCTL) 寄存器 (636页) 的 PMCn 域中，以便把 SSI 信号分配给指定 GPIO 端口管脚。有关如何配置 GPIO 的更多信息，请参阅“通用输入/输出端口 (GPIO)”在 597页。

表 15-1. SSI 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
SSI0Clk	39	PA2 (2)	I/O	TTL	SSI模块0时钟
SSI0Fss	40	PA3 (2)	I/O	TTL	SSI 模块 0 帧信号
SSI0Rx	41	PA4 (2)	I	TTL	SSI模块0接收
SSI0Tx	42	PA5 (2)	O	TTL	SSI模块0发送
SSI1Clk	1 64	PD0 (2) PF2 (2)	I/O	TTL	SSI模块1时钟
SSI1Fss	2 65	PD1 (2) PF3 (2)	I/O	TTL	SSI 模块 1 帧信号
SSI1Rx	3 62	PD2 (2) PF0 (2)	I	TTL	SSI模块1接收
SSI1Tx	4 63	PD3 (2) PF1 (2)	O	TTL	SSI模块1发送
SSI2Clk	26 136	PH4 (2) PB4 (2)	I/O	TTL	SSI模块2时钟
SSI2Fss	23 135	PH5 (2) PB5 (2)	I/O	TTL	SSI 模块 2 帧信号
SSI2Rx	22	PH6 (2)	I	TTL	SSI模块2接收
SSI2Tx	21	PH7 (2)	O	TTL	SSI模块2发送
SSI3Clk	1 16 32	PD0 (1) PK0 (2) PH0 (2)	I/O	TTL	SSI模块3时钟
SSI3Fss	2 17 31	PD1 (1) PK1 (2) PH1 (2)	I/O	TTL	SSI 模块 3 帧信号
SSI3Rx	3 18 28	PD2 (1) PK2 (2) PH2 (2)	I	TTL	SSI模块3接收
SSI3Tx	4 19 27	PD3 (1) PK3 (2) PH3 (2)	O	TTL	SSI模块3发送

a. TTL 表示管脚的电压水平与 TTL 一致。

## 15.3 功能描述

SSI 对从片外器件接收的数据进行串-并转换。CPU 可访问数据、控制信息以及状态信息。发送及接收通道均内置 FIFO 存储器缓冲区，在发送及接收模式下各自最多可以存储 8 个 16 位数。SSI 还支持  $\mu$ DMA 接口。可以将发送和接收 FIFO 配置成  $\mu$ DMA 模块的目的地址/源地址。启用  $\mu$ DMA 操作需要将 SSIDMACTL 寄存器中相应的位置位（请参考 919 页）。

### 15.3.1 位速率的产生

SSI 模块内置可编程的位速率时钟分频器以及预分频器，通过分频产生串行输出时钟。SSI 模块支持 2 MHz 及更高的位速率，实际使用时最高位速率通常由片外器件的性能决定。

串行位速率由输入时钟 (SysClk) 分频后得到。首先，使用 2~254 之间的偶数分频值 CPSDVSR 对输入时钟进行分频，该值在 SSI 时钟预分频 (SSICPSR) 寄存器中设置（请参考 912 页）。然后再使用 1~256 之间的一个数（即 1 + SCR）对时钟进一步分频，此处的 SCR 在 SSI 控制 0 (SSICR0) 寄存器中设置（请参考 905 页）。

输出时钟 SSIClk 的频率由以下公式定义：

$$\text{SSIClk} = \text{SysClk} / (\text{CPSDVSR} * (1 + \text{SCR}))$$

注意：当 SSI 时钟配置 (SSICC) 寄存器的 CS 域被配置为 0x1 时，PIOSC 将作为 SSIClk 的源。在主机模式下，系统时钟或 PIOSC 的速度必须至少是 SSIClk 的两倍，而 SSIClk 不能超过 25 MHz。在从机模式下，系统时钟或者 PIOSC 必须至少是 SSIClk 的 6 倍。

有关 SSI 的时序参数，请参考“Synchronous Serial Interface (SSI)”在 1310 页。

## 15.3.2 FIFO 操作

### 15.3.2.1 发送 FIFO

常用发送 FIFO 是一组 16 位宽、8 单元深的先入先出缓冲区。CPU 通过写 SSI 数据 (SSIDR) 寄存器 (请参考 909 页) 将数据写入发送 FIFO，数据在由发送逻辑读出之前一直保存在发送 FIFO 中。

当 SSI 配置为主机或从机时，并行数据在进行串行转换并通过 SSITx 管脚分别发送到相关的从机或主机之前，即已写入发送 FIFO。

当工作于从机模式时，SSI 模块在每次主机启动通信后才发送数据。若发送 FIFO 为空，那么在主机启动会话时 SSI 模块会把发送 FIFO 中最早的一个数据发送出去。通过 RCGCSSI 寄存器的 Rn 位启用 SSI 模块时钟后，如果写入发送 FIFO 的有效数据不足 8 个，则 SSI 模块将发送 0。因此，必须按照会话要求确保 FIFO 内包含有效数据。SSI 模块可以在发送 FIFO 为空时产生中断或  $\mu$ DMA 请求。

### 15.3.2.2 接收 FIFO

常用 SSI 接收 FIFO 是一组 16 位宽、8 单元深的先入先出缓冲区。从串行接口接收到的数据在由 CPU 读出之前一直保存在缓冲区，CPU 通过读 SSIDR 寄存器来访问读取 FIFO。

当 SSI 配置为主机或从机时，从 SSIRx 管脚接收到的串行数据在并行加载到相关的从机或主机接收 FIFO 之前将被分别记录。

## 15.3.3 中断

SSI 可在出现下列情况时产生中断：

- 发送 FIFO 服务 (发送 FIFO 半满或更低)
- 接收 FIFO 服务 (接收 FIFO 半满或更多)
- 接收 FIFO 超时
- 接收 FIFO 溢出
- 传输结束

在发送给中断控制器之前，所有中断事件先进行一次逻辑或操作，因此同一时刻不管实际发生了多少 SSI 中断事件，SSI 模块都只向中断控制器产生一个中断请求。将 SSI 中断屏蔽 (SSIIM) 寄存器 (请参考 913 页) 中相应的位清零，可以单独屏蔽这四个中断。将相应的屏蔽位置位来启用中断。

SSI 模块不但提供组合的中断输出，还分别提供各个中断源的输出，因此在处理中断时既可采用全局中断服务子程序，也可采用模块化的设备驱动程序。动态的发送/接收数据流中断与静态的状态中断相互独立，方便即时响应 FIFO 触发深度进行读写操作。独立中断源的状态可以查询 SSI 原始中断状态 (SSIRIS) 和 SSI 屏蔽中断状态 (SSIMIS) 寄存器 (请分别参考 914 页和 916 页)。

接收 FIFO 设有 32 个 SSIClk 时钟周期 (不管此时 SSIClk 是否激活) 的超时周期，并且只要接收 FIFO 从空状态变为非空状态即会启动。假如 RX FIFO 在接下来的 32 个时钟内再次变为空状态，超时周期也会复位。因此中断服务子程序 (ISR) 应在读取接收 FIFO 之后及时将 SSI 中断清除 (SSIICR)

寄存器的 RTIC 位置位，以清除接收 FIFO 超时中断。此清除操作不能执行得太晚，否则有可能中断服务子程序 (ISR) 在中断实际清除之前已经返回，此外也可能造成不必要地重复进入中断。

传输结束 (EOT) 中断表示数据已经传输完成。该中断可以用来指示什么时候关闭 SSI 模块的时钟或者进入休眠模式。另外，由于数据的发送和接收是同时完成的，此中断也能即时指示接收 FIFO 中的数据已经就绪，无需等待接收 FIFO 超时了。

### 15.3.4 帧格式

根据所设置的数据大小，每个数据帧的长度均在 4~16 位之间，并且从最高有效位 (MSB) 开始发送。通过对 SSICR0 寄存器中的 FRF 位进行设置，用户可选择三种基本的帧类型：

- Texas Instruments 同步串行
- 飞思卡尔 (Freescale) SPI
- MICROWIRE

对于上述三种帧格式，串行时钟 (SSIClk) 在 SSI 空闲时保持不活动状态，只有当数据的发送或接收处于活动状态时，SSIClk 才在设置好的频率下工作。利用 SSIClk 的空闲状态可提供接收超时指示，如果一个超时周期之后接收 FIFO 仍含有数据，则产生超时指示。

对于 Freescale SPI 和 MICROWIRE 这两种帧格式，串行帧 (SSIFss) 管脚为低电平有效，并在整个帧的传输过程中保持有效 (被下拉)。

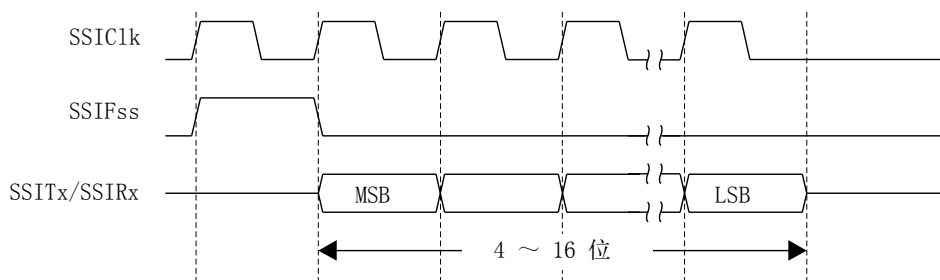
而对于 Texas Instruments 同步串行帧格式，在发送每个帧之前，SSIFss 管脚会发出一个以上升沿开始并持续一个时钟周期的脉冲。在这种帧格式中，SSI 和片外从器件在 SSIClk 的上升沿驱动各自的输出数据，并在下降沿锁存另一个器件的数据。

与另外两种全双工的帧格式不同，MICROWIRE 是半双工工作的，采用特殊的主-从传输技术。在该模式中，当帧开始传输时向片外从机发送一个 8 位的控制报文。在发送过程中，SSI 不会接收到任何输入数据。当报文发送结束后，片外从机即对报文进行译码。发送 8 位报文的最后一位扣，该从机将等待 1 个串行时钟周期，然后开始响应请求的数据。应答数据长度为 4 位到 16 位，于是单次会话的总帧长为 13 位到 25 位。

#### 15.3.4.1 Texas Instruments 同步串行帧格式

图 15-2 在 895 页显示了 Texas Instruments 同步串行单次传输的帧格式。

图 15-2. TI 同步串行帧格式 ( 单次传输 )

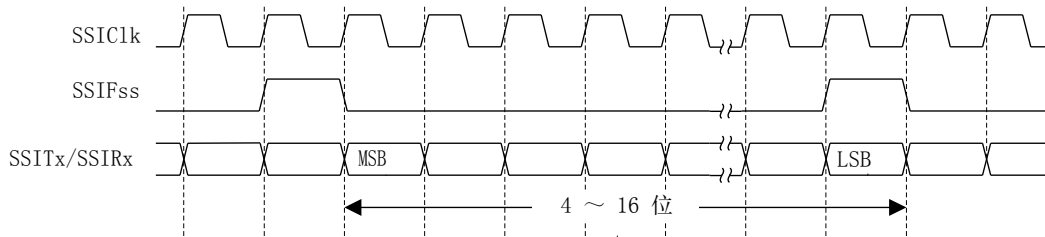


在此模式中，当 SSI 模块处于空闲状态时，SSIClk 和 SSIFss 被强制拉低，发送数据线 SSITx 被置为三态。一旦发送 FIFO 的底部入口包含数据，SSIFss 就会变为高电平并持续一个 SSIClk 周期。要发送的值也从发送 FIFO 传输到发送逻辑的串行移位寄存器中。在 SSIClk 的下一个上升沿，4~16 位数据帧的 MSB 从 SSITx 管脚移出。同样，接收数据的 MSB 也被片外串行从器件移到 SSIRx 管脚上。

然后，SSI 和片外从器件在 SSIClk 的每一个下降沿时将数据位逐个移入各自的串行移位器中。在锁存了 LSB 之后的第一个 SSIClk 上升沿，接收数据从串行移位器传输到接收 FIFO。

图 15-3 在 896 页显示了 Texas Instruments 同步串行帧格式的连续的传输情况。

图 15-3. TI 同步串行的帧格式 (连续传输)



#### 15.3.4.2 飞思卡尔 SPI 格式

Freescale SPI 接口是一个 4 线接口，其中 SSIFss 信号用于从机选择。Freescale SPI 格式的主要特征是其 SSIClk 信号定义相当灵活，其非活动状态及相位可分别通过 SSISCR0 控制寄存器的 SPO 和 SPH 位进行设置。

##### SPO 时钟极性位

当 SPO 时钟极性控制位清零时，它在 SSIClk 管脚上产生稳定的低电平值。如果 SPO 位置位，则在没有进行数据传输的情况下，它在 SSIClk 管脚上产生一个稳定的高电平值。

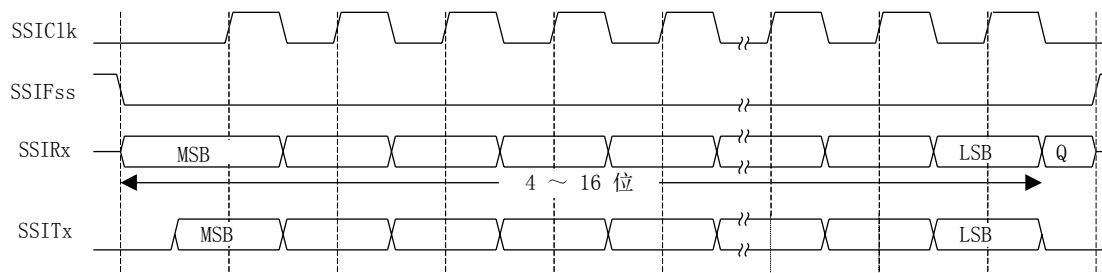
##### SPH 相位控制位

SPH 相位控制位用于选择捕捉数据的时钟沿，允许其改变状态。该位的状态对传输的首位影响最大，即是否在首次捕捉数据之前忽略 1 次时钟跳变。当 SPH 相位控制位为低时，在第一个时钟边沿转换时捕获数据。如果 SPH 位为高，则在第二个时钟边沿转换时捕获数据。

#### 15.3.4.3 SPO = 0 和 SPH = 0 时的 Freescale SPI 帧格式

SPO = 0 和 SPH = 0 时，Freescale SPI 帧格式的单次和连续传输信号序列如图 15-4 在 896 页和图 15-5 在 897 页所示。

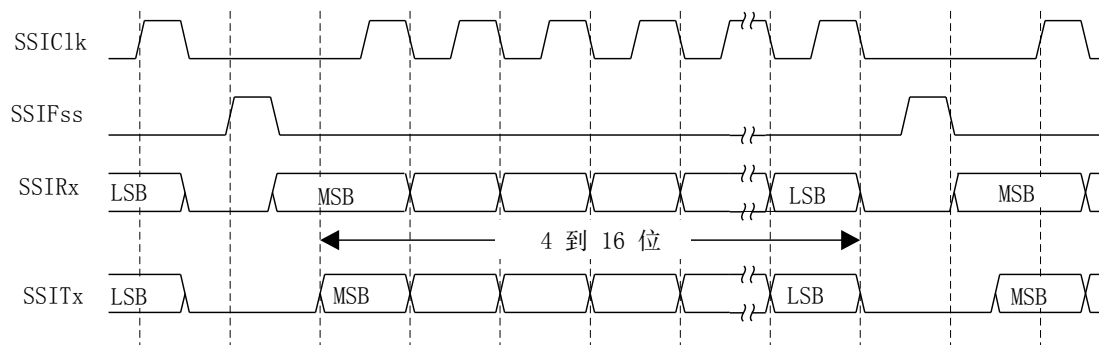
图 15-4. SPO = 0 和 SPH = 0 时 Freescale SPI 帧格式 (单次传输)



注意： Q 未定义。



图 15-5. SPO = 0 和 SPH = 0 时 Freescale SPI 的帧格式 (连续传输)



在此配置中，当 SSI 处于空闲时：

- SSIClk 被强制变为低电平
- SSIFss 被强制变为高电平
- 发送数据线 SSITx 被仲裁强制变为低电平
- 当 SSI 配置为主机时，它将启用 SSIClk 管脚
- 当 SSI 配置为从机时，它将禁用 SSIClk 管脚

若 SSI 模块启用，并且发送 FIFO 中已经填入有效数据，那么 SSIFss 主机信号拉低标志传输开始，从机数据可以传输到主机的 SSIRx 输入线上。主机的 SSITx 输出管脚也被启用。

在半 SSIClk 周期之后，有效的主机数据传输到 SSITx 管脚。一旦主机和从机数据设定好，另外半个 SSIClk 时钟周期后，SSIClk 主机时钟管脚被拉高。

这时，数据在 SSIClk 信号的上升沿被捕获，在下降沿进行传输。

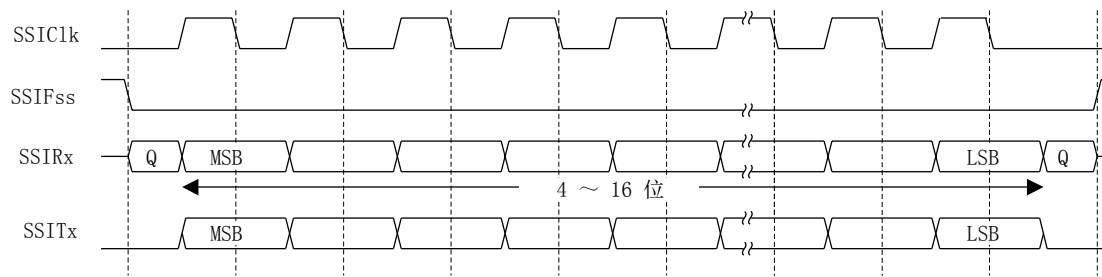
如果传输一个字，则在数据字的所有位都已传输完之后，SSIFss 线在捕获到最后一位之后的一个 SSIClk 周期返回到其空闲的高电平状态。

然而在连续传输过程中，必须让 SSIFss 信号在相邻两次数据传输之间输出高脉冲，因为当 SPH 位清零时，从机选通管脚将锁住串行外设寄存器中的数据，不允许对其修改。因此主机必须在相邻两次数据传输之间拉高从机的 SSIFss 管脚，确保能够正确写入串行外设数据。当连续传输完成时，SSIFss 管脚将在捕获到最后一位之后的一个 SSIClk 周期后返回到空闲状态。

#### 15.3.4.4 SPO = 0 和 SPH = 1 时的 Freescale SPI 帧格式

SPO = 0 和 SPH = 1 时，Freescale SPI 帧格式的传输信号序列如图 15-6 在 898 页所示，该图涵盖了单次和连续传输这两种情况。

图 15-6. SPO = 0 和 SPH = 1 时的 Freescale SPI 帧格式



注意: Q 未定义。

在此配置中，当 SSI 处于空闲时：

- SSIClk 被强制变为低电平
- SSIFss 被强制变为高电平
- 发送数据线 SSITx 被仲裁强制变为低电平
- 当 SSI 配置为主机时，它将启用 SSIClk 管脚
- 当 SSI 配置为从机时，它将禁用 SSIClk 管脚

如果 SSI 启用并且在发送 FIFO 中含有有效的数据，则通过将 SSIFss 主机信号驱动为低电平表示发送操作开始。主机 SSITx 输出被启用。半个 SSIClk 时钟周期后，主机和从机的数据都已在各自的发送线上就绪。同时，利用一个上升沿跳变将 SSIClk 启用。

这时，数据在 SSIClk 信号的下降沿被捕获，在上升沿进行传输。

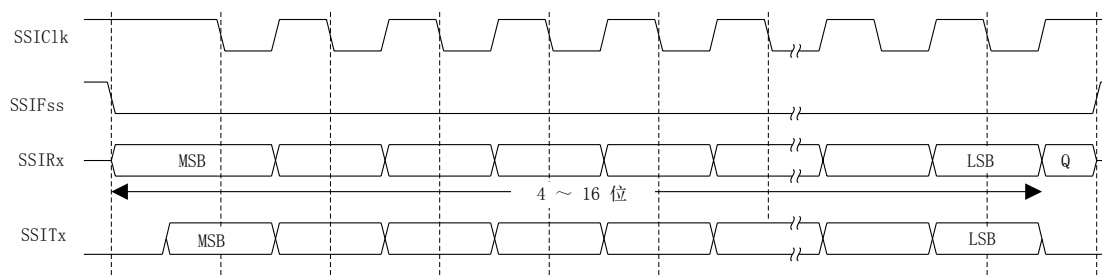
如果传输一个字，则在所有位传输完之后，SSIFss 线在捕获到最后一个位之后的一个 SSIClk 周期返回到其空闲的高电平状态。

如果是连续的传输，则 SSIFss 管脚在连续的数据字之间保持为低电平，连续传输的结束情况与单字传输的相同。

#### 15.3.4.5 SPO = 1 和 SPH = 0 时的 Freescale SPI 帧格式

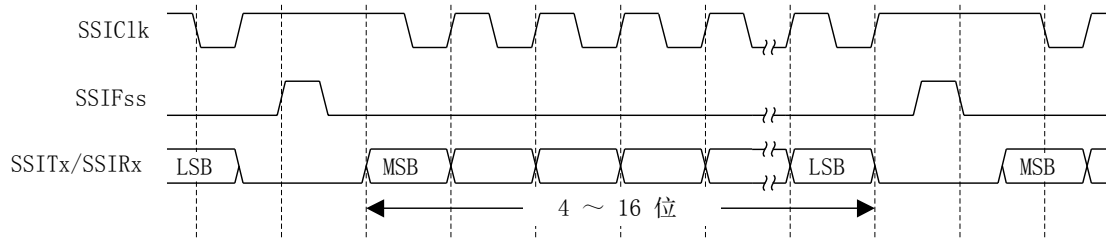
SPO = 1 和 SPH = 0 时，Freescale SPI 帧格式的单次和连续传输信号序列如图 15-7 在 898 页和图 15-8 在 899 页所示。

图 15-7. SPO = 1 和 SPH = 0 时 Freescale SPI 的帧格式 ( 单次传输 )



注意: Q 未定义。

图 15-8. SPO = 1 和 SPH = 0 时 Freescale SPI 的帧格式 (连续传输)



在此配置中，当 SSI 处于空闲时：

- SSIClk 被强制变为高电平
- SSIFss 被强制变为高电平
- 发送数据线 SSITx 被仲裁强制变为低电平
- 当 SSI 配置为主机时，它将启用 SSIClk 管脚
- 当 SSI 配置为从机时，它将禁用 SSIClk 管脚

若 SSI 模块启用，并且发送 FIFO 中已经填入有效数据，SSIFss 主机信号拉低表示传输过程开始，而从机也立即往主机 SSIRx 线上传输数据。主机的 SSITx 输出管脚也被启用。

半个时钟周期后，有效主机数据输出到 SSITx 线上。一旦主机和从机的数据都已就绪，另外半个 SSIClk 时钟周期后 SSIClk 主机时钟管脚被拉低。之后双方将在每个 SSIClk 时钟的下降沿捕捉数据、在上升沿进行输出。

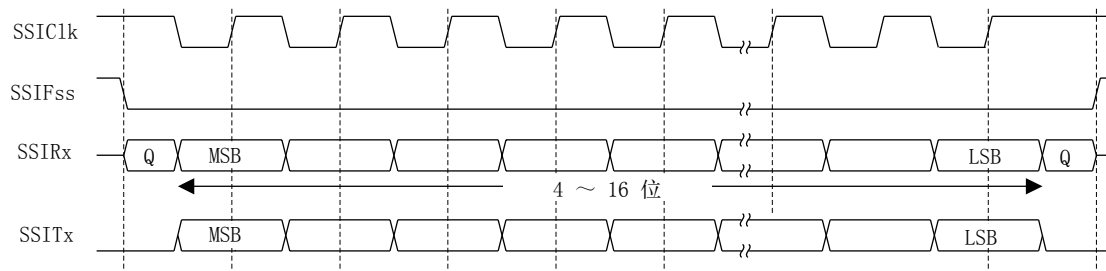
如果传输一个字，则在数据字的所有位传输完之后，SSIFss 线在捕获到最后一位之后的一个 SSIClk 周期后返回到其空闲的高电平状态。

然而在连续传输过程中，必须让 SSIFss 信号在相邻两次数据传输之间输出高脉冲，因为当 SPH 位清零时，从机选通管脚将锁住串行外设寄存器中的数据，不允许对其修改。因此主机必须在相邻两次数据传输之间拉高从机的 SSIFss 管脚，确保能够正确写入串行外设数据。当连续传输完成时，SSIFss 管脚将在捕获到最后一位之后的一个 SSIClk 周期后返回到空闲状态。

#### 15.3.4.6 SPO = 1 和 SPH = 1 时的 Freescale SPI 帧格式

SPO = 1 和 SPH = 1 时，Freescale SPI 帧格式的传输信号序列如图 15-9 在 899 页所示，该图涵盖了单次和连续传输这两种情况。

图 15-9. SPO = 1 和 SPH = 1 时的 Freescale SPI 帧格式



注意： Q 未定义。

在此配置中，当 SSI 处于空闲时：

- SSIClk 被强制变为高电平
- SSIFss 被强制变为高电平
- 发送数据线 SSITx 被仲裁强制变为低电平
- 当 SSI 配置为主机时，它将启用 SSIClk 管脚
- 当 SSI 配置为从机时，它将禁用 SSIClk 管脚

如果 SSI 启用并且在发送 FIFO 中含有有效的数据，则通过将 SSIFss 主机信号驱动为低电平表示发送操作开始。主机的 SSITx 输出管脚也被启用。另外半个 SSIClk 时钟周期后，主设备和从设备的数据都已在各自的发送线上就绪。同时，利用下降沿跳变将 SSIClk 启用。之后双方将在每个 SSIClk 时钟的上升沿捕捉数据、在下降沿进行输出。

在所有位传输完之后，如果是单个字传输，则在最后一个位传输完之后的一个 SSIClk 周期中，SSIFss 线返回到其空闲的高电平状态。

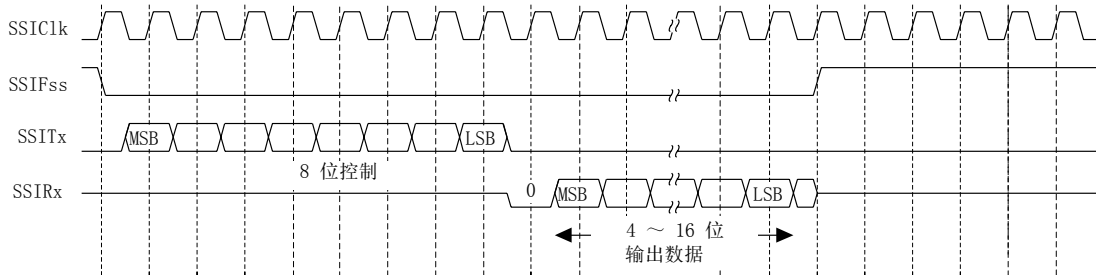
而对于连续传输，SSIFss 管脚保持其有效的低电平状态，直至最后一个字的最后一位捕获完成，再返回其上述的空闲状态。

如果是连续传输，则 SSIFss 管脚在连续的数据字之间保持为低电平，连续传输的结束情况与单字传输的相同。

#### 15.3.4.7 MICROWIRE 帧格式

图 15-10 在 900 页 显示了单次传输的 MICROWIRE 帧格式图 15-11 在 901 页 显示了该格式的连续的传输情况

图 15-10. MICROWIRE 帧格式 (单帧)



MICROWIRE 的帧格式与 SPI 非常相近，区别之处在于 MICROWIRE 是半双工而非全双工，而且还采用了主-从报文传递技术。每次串行传输都由 SSI 向片外从器件发送 8 位控制字开始。在此传输过程中，SSI 不会接收到输入的数据。当控制字发送结束（8 位的最后 1 位发送完成）后，片外从设备即对控制字进行译码。总线等待 1 个时钟周期之后，片外从设备开始发送应答数据。应答数据长度为 4 位到 16 位，于是单次会话的总帧长为 13 位到 25 位。

在此配置中，当 SSI 处于空闲时：

- SSIClk 被强制变为低电平
- SSIFss 被强制变为高电平
- 发送数据线 SSITx 被仲裁强制变为低电平

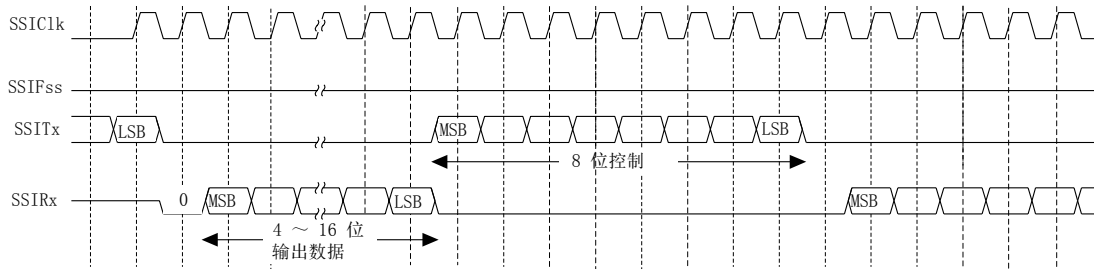
向发送 FIFO 内写入控制字节即可触发发送。在 SSIFss 的下降沿，发送 FIFO 底部入口包含的值被传输到发送逻辑的串行移位寄存器中，而 8 位控制帧的 MSB 被移出到 SSITx 管脚上。在该控制帧的传输期间 SSIFss 保持低电平。SSIRx 管脚保持三态。

片外从设备会在每个 SSIClk 时钟的上升沿将控制字的位依次锁存入串行移位寄存器。在将最后一位锁存之后，从器件在一个时钟周期的等待状态期间对控制字节进行译码，并且从机将数据发送回 SSI 进行应答。每个数据位在 SSIClk 的下降沿时刻被驱动到 SSIRx 线上。SSI 在 SSIClk 的上升沿时依次将每个位锁存。在帧传输结束时，对于单次传输，SSIFss 信号在最后 1 位被锁存入串行移位寄存器（上升沿）后保持 1 个时钟周期，然后重新拉高，使接收的数据传递至接收 FIFO。

**注意：** 在接收移位器将 LSB 锁存之后的 SSIClk 的下降沿上或在 SSIFss 管脚变为高电平时，片外从器件能够将接收线置为三态。

连续传输的开始及结束均与单次传输相同。但 SSIFss 线持续有效（保持低电平），并且数据传输以连续的方式产生。下一帧的控制字节紧跟当前接收数据的最低有效位（LSB）。在当前帧的 LSB 锁存到 SSI 之后，所接收到的每个值在 SSIClk 的下降沿时刻从接收移位器中进行传输。

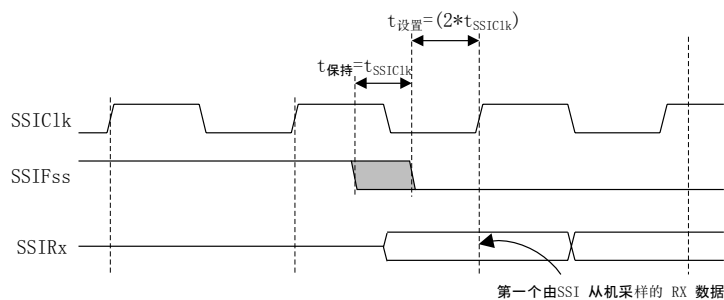
图 15-11. MICROWIRE 帧格式（连续传输）



在 MICROWIRE 模式中，当 SSIFss 变为低电平之后，SSI 从机在 SSIClk 的上升沿时刻对接收数据的第一个位进行采样。用来驱动自由运行的 SSIClk 的主机必须确保 SSIFss 信号相对于 SSIClk 的上升沿具有足够的建立时间和保持时间裕量。

图 15-12 在 901 页 阐明了建立和保持时间要求。相对于 SSI 从机对接收数据的第一位进行采样时所在的 SSIClk 上升沿，SSIFss 的建立时间至少必须是 SSI 操作时钟 (SSIClk) 周期的两倍。相对于该边沿之前的 SSIClk 上升沿，SSIFss 至少必须具有一个 SSIClk 周期的保持时间。

图 15-12. MICROWIRE 帧格式，SSIFss 输入建立和保持时间要求



### 15.3.5 DMA 操作

SSI 模块可与  $\mu$ DMA 控制器接口实现相互独立的发送通道和接收通道。SSI 的  $\mu$ DMA 操作通过 SSI DMA 控制 (SSIDMACTL) 寄存器启用。在启用  $\mu$ DMA 操作后，SSI 模块在接收 FIFO 或发送 FIFO 可以传输数据时向接收或发送通道产生  $\mu$ DMA 请求。对于接收通道，只要接收 FIFO 中存在数据，

单次传输请求就会发出。如果接收 FIFO 中的数据是 4 个或者更多，就会发出多个连续传输请求。对于发送通道，只要发送 FIFO 中存在一个空位，单次传输请求就会发出。如果发送 FIFO 中的空位有 4 个或者更多，就会发出多个连续传输请求。 $\mu$ DMA 控制器会根据  $\mu$ DMA 通道的配置自动处理单次传输请求和连续传输请求。为了启用接收通道的 DMA 操作，寄存器 DMA 控制 (SSIDMACTL) 中的 RXDMAE 位应置位。为了启用发送通道的 DMA 操作，寄存器 SSIDMACTL 中的 TXDMAE 位应置位。如果已经启用了  $\mu$ DMA，那么  $\mu$ DMA 控制器会在传输结束时自动触发中断。此中断产生到 SSI 中断向量。因此，如果启用了 SSI 操作使用的中断和  $\mu$ DMA，必须设计 SSI 中断处理函数来处理  $\mu$ DMA 的完成中断。

更多的  $\mu$ DMA 控制器编程信息请参考“微型直接存储器访问 ( $\mu$ DMA)”在 534 页。

## 15.4 初始化及配置

请按照以下步骤启用并初始化 SSI 模块：

1. 用 RCGCSSI 寄存器（请参考 309 页）启用 SSI 模块。
2. 通过 RCGCGPIO 寄存器（请参考 302 页）启用相应 GPIO 模块的时钟。欲了解需要启用哪些 GPIO 端口，请参考表 23-5 在 1280 页。
3. 为相应的管脚设置 GPIO AFSEL 位（请参考 618 页）。欲了解需要配置哪些 GPIO，请参阅表 23-4 在 1271 页。
4. 配置 GPIOCTL 寄存器的 PMCn 域，将 SSI 信号赋给相应的管脚。请参阅 636 页和表 23-5 在 1280 页。

针对不同的帧格式，SSI 可通过以下步骤进行配置：

1. 在更改配置前，应确保 SSICR1 寄存器的 SSE 位清零。
2. 确定 SSI 为主机还是从机：
  - a. 若工作于主机模式，应将 SSICR1 寄存器配置为 0x0000.0000；
  - b. 若工作于从机模式（启用输出），应将 SSICR1 寄存器配置为 0x0000.0004。
  - c. 若工作于从机模式（禁用输出），应将 SSICR1 寄存器配置为 0x0000.000C。
3. 通过写 SSICC 寄存器来配置 SSI 时钟源。
4. 通过写 SSICPSR 寄存器来配置时钟预分频除数。
5. 写 SSICR0 寄存器，实现以下配置：
  - 串行时钟速率 (SCR)
  - 如果使用 Freescale SPI 模式，则配置所需的时钟相位/极性 (SPH 和 SPO)
  - 协议模式：Freescale SPI、TI SSF、MICROWIRE (FRF)
  - 数据宽度 (DSS)
6. 另外，还可以配置  $\mu$ DMA 通道（请参考“微型直接存储器访问 ( $\mu$ DMA)”在 534 页），并在 SSIDMACTL 寄存器中启用 DMA 选项。
7. 通过置位 SSICR1 寄存器的 SSE 位来启用 SSI。

举例：假定 SSI 的配置如下：

- 主机操作
- Freescale SPI 模式 (SPO = 1, SPH = 1)
- 1 Mbps 位速率
- 8 个数据位

如果系统时钟为 20 MHz，则位速率的计算如下：

$$\text{SSIClk} = \text{SysClk} / (\text{CPSDVSR} * (1 + \text{SCR}))$$

$$1 \times 10^6 = 20 \times 10^6 / (\text{CPSDVSR} * (1 + \text{SCR}))$$

在这种情况下，如果 CPSDVSR = 0x2，则 SCR 必须为 0x9。

具体的配置序列如下：

1. 确保 SSICR1 寄存器的 SSE 位已经清零。
2. 将 SSICR1 寄存器写入 0x0000.0000。
3. 将 SSICPSR 寄存器写入 0x0000.0002。
4. 将 SSICR0 寄存器写入 0x0000.09C7。
5. 将 SSICR1 寄存器的 SSE 位置位，启用 SSI 模块。

## 15.5 寄存器映射

表 15-2 在 903 页 列出了 SSI 寄存器。所列出的地址偏移量是寄存器相对于 SSI 的基址的 16 进制增量：

- SSI0 : 0x4000.8000
- SSI1 : 0x4000.9000
- SSI2 : 0x4000.A000
- SSI3 : 0x4000.B000

请注意配置这些寄存器之前必须先启用 SSI 模块时钟（请参考 309 页）。SSI 模块时钟启用之后必须等待三个系统时钟的延迟，SSI 模块的寄存器才能访问。

注意： 在对任何控制寄存器重新编程前，必须将 SSI 禁用（请参考 SSICR1 寄存器的 SSE 位）。

表 15-2. SSI 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	SSICR0	R/W	0x0000.0000	SSI 控制寄存器0	905
0x004	SSICR1	R/W	0x0000.0000	SSI 控制寄存器1	907
0x008	SSIDR	R/W	0x0000.0000	SSI 数据寄存器	909
0x00C	SSISR	RO	0x0000.0003	SSI 状态寄存器	910
0x010	SSICPSR	R/W	0x0000.0000	SSI 时钟预分频寄存器	912
0x014	SSIIM	R/W	0x0000.0000	SSI 中断屏蔽寄存器	913

表 15-2. SSI 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x018	SSIRIS	RO	0x0000.0008	SSI 原始中断状态寄存器	914
0x01C	SSIMIS	RO	0x0000.0000	SSI 屏蔽中断状态寄存器	916
0x020	SSIICR	W1C	0x0000.0000	SSI 中断清除寄存器	918
0x024	SSIDMACTL	R/W	0x0000.0000	SSI DMA 控制寄存器	919
0xFC8	SSICC	R/W	0x0000.0000	SSI 时钟配置寄存器	920
0xFD0	SSIPeriphID4	RO	0x0000.0000	SSI 外设标识寄存器 4	921
0xFD4	SSIPeriphID5	RO	0x0000.0000	SSI 外设标识寄存器 5	922
0xFD8	SSIPeriphID6	RO	0x0000.0000	SSI 外设标识寄存器 6	923
0xFDC	SSIPeriphID7	RO	0x0000.0000	SSI 外设标识寄存器 7	924
0xFE0	SSIPeriphID0	RO	0x0000.0022	SSI 外设标识寄存器 0	925
0xFE4	SSIPeriphID1	RO	0x0000.0000	SSI 外设标识寄存器 1	926
0xFE8	SSIPeriphID2	RO	0x0000.0018	SSI 外设标识寄存器 2	927
0xFEC	SSIPeriphID3	RO	0x0000.0001	SSI 外设标识寄存器 3	928
0xFF0	SSIPCellID0	RO	0x0000.000D	SSI PrimeCell 标识寄存器 0	929
0xFF4	SSIPCellID1	RO	0x0000.00F0	SSI PrimeCell 标识寄存器 1	930
0xFF8	SSIPCellID2	RO	0x0000.0005	SSI PrimeCell 标识寄存器 2	931
0xFFC	SSIPCellID3	RO	0x0000.00B1	SSI PrimeCell 标识寄存器 3	932

## 15.6 寄存器描述

下文将按地址偏移量的数字顺序列举并描述 SSI 寄存器。



## 寄存器 1: SSI 控制寄存器0 (SSICR0), 偏移量 0x000

SSICR0 寄存器中包含的位域用来控制 SSI 模块内的各种功能。诸如协议模式、时钟速率和数字宽度等功能都在该寄存器中配置。

## SSI 控制寄存器0 (SSICR0)

SSI0 基址: 0x4000.8000

SSI1 基址: 0x4000.9000

SSI2 基址: 0x4000.A000

SSI3 基址: 0x4000.B000

偏移量 0x000

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCR								SPH	SPO	FRF		DSS			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:8	SCR	R/W	0x00	SSI 串行时钟速率 该域的值用来产生 SSI 的发送和接收位速率位速率： $BR = \text{SysClk} / (\text{CPSDVSR} * (1 + \text{SCR}))$ 其中 CPSDVSR 是 SSICPSR 寄存器的位域，取值为 2~254 之间的偶数，SCR 取值为 0~255。
7	SPH	R/W	0	SSI 串行时钟相位 此位仅用于 Freescale SPI 格式。 SPH 控制位用来选择捕获数据的时钟边沿并允许边沿改变状态。该位对第一个传输位影响最大，因为它可以在第一个数据捕获边沿之前允许或不允许一次时钟转换。  值 描述 0 数据在首个时钟跳变沿捕捉。 1 数据在第 2 个时钟跳变沿捕捉。
6	SPO	R/W	0	SSI 串行时钟极性  值 描述 0 SSIClk 管脚为稳定的低电平。 1 当无数据传输时，SSIClk 管脚为稳定的高电平。

位/域	名称	类型	复位	描述
5:4	FRF	R/W	0x0	SSI 帧格式选择  值 帧格式 0x0 Freescale SPI 格式 0x1 Texas Instruments 同步串行帧格式 0x2 MICROWIRE 帧格式 0x3 保留
3:0	DSS	R/W	0x0	SSI 数据大小选择  值 系统数据宽度 0x0-0x2 保留 0x3 传输的每个数据单元为 4 位。 0x4 传输的每个数据单元为 5 位。 0x5 传输的每个数据单元为 6 位。 0x6 传输的每个数据单元为 7 位。 0x7 传输的每个数据单元为 8 位。 0x8 传输的每个数据单元为 9 位。 0x9 传输的每个数据单元为 10 位。 0xA 传输的每个数据单元为 11 位。 0xB 传输的每个数据单元为 12 位。 0xC 传输的每个数据单元为 13 位。 0xD 传输的每个数据单元为 14 位。 0xE 传输的每个数据单元为 15 位。 0xF 传输的每个数据单元为 16 位。

## 寄存器 2: SSI 控制寄存器1 ( SSICR1 ) , 偏移量 0x004

SSICR0 寄存器中包含的位域用来控制 SSI 模块内的各种功能。主机和从机的模式功能由该寄存器控制。

## SSI 控制寄存器1 (SSICR1)

SSIO 基址: 0x4000.8000  
 SS11 基址: 0x4000.9000  
 SS12 基址: 0x4000.A000  
 SS13 基址: 0x4000.B000  
 偏移量 0x004  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										SLBY6	EOT	SOD	MS	SSE	LBM
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:6	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
5	SLBY6	R/W	0	从机旁路模式  值 描述 0 SSI 模块在正常模式下工作，从机与 SSInCLK 的系统时钟比例为 12:1。 1 SSI 模块启用，从机按与 SSInCLK 的比例为 6:1 的系统时钟工作。
4	EOT	R/W	0	传输结束  值 描述 0 TXRIS 中断表示发送 FIFO 中的数据半满或者更少。 1 当传输结束时产生 TXRIS 中断。
3	SOD	R/W	0	SSI 从机模式输出禁用 该位只在从机模式 ( MS = 1 ) 下使用。在多从机系统中，SSI 主机可以向系统中的所有从机广播一则报文，同时能保证只有一个从机将数据驱动到串行输出线上。在这样的系统中，多个从机的 TXD 线可以连在一起在这样的系统中操作时，需对 SOD 位进行配置以使 SSI 从机不驱动 SSITx 管脚。  值 描述 0 从机模式下 SSI 模块可以驱动 SSITx 输出。 1 从机模式下 SSI 模块不能驱动 SSITx 输出。

位/域	名称	类型	复位	描述
2	MS	R/W	0	<p>SSI 主机/从机选择</p> <p>此位用于选择主机模式还是从机模式，只有当 SSI 禁用 (SSE = 0) 时才能设置。</p> <p>值 描述</p> <p>0 SSI 模块工作于主机模式。</p> <p>1 SSI 模块工作于从机模式。</p>
1	SSE	R/W	0	<p>SSI 同步串行接口启用</p> <p>值 描述</p> <p>0 禁用 SSI 操作</p> <p>1 启用 SSI 操作</p> <p>注意： 在对任何控制器重新编程之前必须先将该位清零。</p>
0	LBM	R/W	0	<p>SSI 回送模式</p> <p>值 描述</p> <p>0 启用正常的串口操作。</p> <p>1 启用环回模式：发送串行移位寄存器的输出端将在芯片内部连接到接收串行移位寄存器的输入端。</p>

### 寄存器 3: SSI 数据寄存器 (SSIDR), 偏移量 0x008

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

SSIDR 寄存器是一个 16 位宽的寄存器。当读取 SSIDR 寄存器时, 将访问接收 FIFO 中由当前 FIFO 读指针所指向的单元。当 SSI 接收逻辑单元将输入的数据帧移出后, 该数据将放入接收 FIFO 中由当前 FIFO 写指针所指向的单元。

当写入 SSIDR 寄存器时, 数据将放入发送 FIFO 中由当前 FIFO 写指针所指向的单元。发送逻辑从发送 FIFO 中一次移出一个数据值。每个数据值被加载到发送串行移位器中, 然后以设置好的位速率串行移出到 SSITx 管脚。

当所选的数据大小小于 16 位时, 用户必须将写入发送 FIFO 的数据右对齐。发送逻辑忽略未使用的位。而在接收时, 若收到的数据长度不足 16 位, 在接收缓冲中能自动右对齐。

当 SSI 设置为 MICROWIRE 帧格式时, 发送数据的默认宽度为 8 位 (忽略最高有效字节)。接收数据的宽度由程序员控制。即使 SSICR1 寄存器的 SSE 位清零, 发送 FIFO 和接收 FIFO 也不会清零, 因此软件可以在启用 SSI 模块之前先填充发送 FIFO。

#### SSI 数据寄存器 (SSIDR)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0x008  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

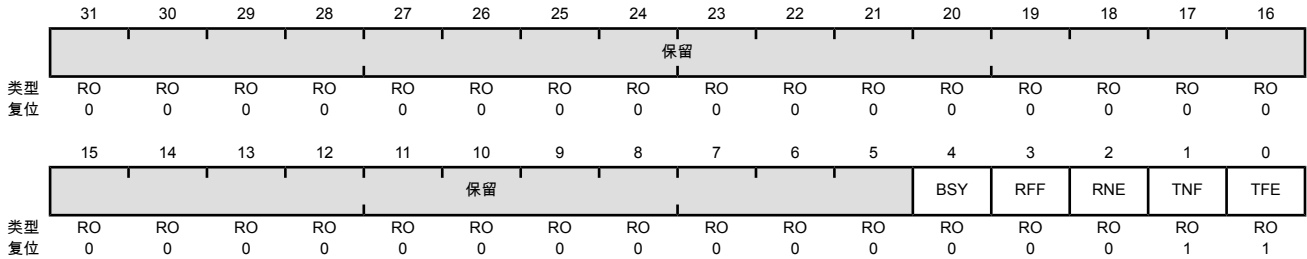
位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	DATA	R/W	0x0000	SSI 接收/发送数据 一个读取操作读取接收 FIFO 的数据。一个写入操作将数据写入发送 FIFO。 当 SSI 设置的数据宽度小于 16 位时, 软件必须将数据右对齐。发送逻辑将忽略顶部未使用的位。接收逻辑自动右对齐数据。

### 寄存器 4: SSI 状态寄存器 ( SSISR ) , 偏移量 0x00C

SSISR 寄存器包含若干标志位, 用于指示 FIFO 填充状态和 SSI 忙状态。

#### SSI 状态寄存器 (SSISR)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0x00C  
 类型 RO, 复位 0x0000.0003



位/域	名称	类型	复位	描述
31:5	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
4	BSY	RO	0	SSI 忙标志位  值 描述 0 SSI 空闲。 1 SSI 正在发送/接收数据帧, 或发送 FIFO 非空。
3	RFF	RO	0	SSI 接收 FIFO 满  值 描述 0 接收 FIFO 未滿。 1 接收 FIFO 满
2	RNE	RO	0	SSI FIFO 不为空  值 描述 0 接收 FIFO 为空 1 接收 FIFO 不为空
1	TNF	RO	1	SSI 发送 FIFO 未滿  值 描述 0 发送 FIFO 已滿 1 发送 FIFO 未滿

---

位/域	名称	类型	复位	描述
0	TFE	RO	1	SSI 发送 FIFO 为空 值 描述 0 发送 FIFO 不为空。 1 发送 FIFO 为空。

### 寄存器 5: SSI 时钟预分频寄存器 (SSICPSR), 偏移量 0x010

SSICPSR 寄存器指定一个分频因子, 以根据系统时钟产生 SSIClk 时钟。该时钟进一步分频, 分频因子为 1 到 256, 即  $1 + SCR$ 。SCR 通过 SSICR0 寄存器编程配置。SSIClk 频率的计算公式如下:

$$SSIClk = SysClk / (CPSDVSR * (1 + SCR))$$

该寄存器必须写入 2 到 254 之间的偶数。所设的值的最低有效位硬编码为 0。如果向该寄存器写入奇数, 则该寄存器读操作返回的值中, 最低有效位为 0。

#### SSI 时钟预分频寄存器 (SSICPSR)

SS10 基址: 0x4000.8000  
 SS11 基址: 0x4000.9000  
 SS12 基址: 0x4000.A000  
 SS13 基址: 0x4000.B000  
 偏移量 0x010  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CPSDVSR							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CPSDVSR	R/W	0x00	SSI 时钟预分频除数 该值必须为 2-254 之间的一个偶数, 具体取值由 SSIClk 的频率决定。执行读操作时, LSB 的返回值始终为 0。



## 寄存器 6: SSI 中断屏蔽寄存器 (SSIIM), 偏移量 0x014

SSIIM 寄存器为中断屏蔽置位或清零寄存器。该寄存器可读可写, 复位后所有位清零。

对本寄存器进行读操作, 可获取各个中断的当前屏蔽状态。将某位置位会屏蔽相应的中断, 该中断就不会发送到中断控制器。将某位清零会清除相应中断的屏蔽, 该中断就会发送到中断控制器。

### SSI 中断屏蔽寄存器 (SSIIM)

SSI0 基址: 0x4000.8000

SSI1 基址: 0x4000.9000

SSI2 基址: 0x4000.A000

SSI3 基址: 0x4000.B000

偏移量 0x014

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												TXIM	RXIM	RTIM	RORIM
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	TXIM	R/W	0	SSI 发送 FIFO 中断屏蔽  值 描述 0 发送 FIFO 中断被屏蔽。 1 发送 FIFO 中断没有被屏蔽。
2	RXIM	R/W	0	SSI 接收 FIFO 中断屏蔽  值 描述 0 接收 FIFO 中断被屏蔽。 1 接收 FIFO 中断没有被屏蔽。
1	RTIM	R/W	0	SSI 接收超时中断屏蔽  值 描述 0 接收 FIFO 超时中断被屏蔽。 1 接收 FIFO 超时中断没有被屏蔽。
0	RORIM	R/W	0	SSI 接收溢出中断屏蔽  值 描述 0 接收 FIFO 溢出中断被屏蔽。 1 接收 FIFO 溢出中断没有被屏蔽。

## 寄存器 7: SSI 原始中断状态寄存器 ( SSIRIS ) , 偏移量 0x018

SSIRIS 寄存器是原始中断状态寄存器。读操作时，此寄存器给出屏蔽前的当前原始中断状态值。对本寄存器的写操作无效。

### SSI 原始中断状态寄存器 (SSIRIS)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0x018  
 类型 RO, 复位 0x0000.0008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													TXRIS	RXRIS	RTRIS	RORRIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
3	TXRIS	RO	1	SSI 发送 FIFO 原始中断状态  值 描述 0 无中断。 1 若 SSI1CR1 寄存器的 EOT 位清零，则表示发送 FIFO 为半空或更少。 若 EOT 置位，则表示发送 FIFO 为空，且最后 1 位已经通过串口发出。  当发送 FIFO 中有效数据超过 4 个单元 ( EOT 清零 ) 或非空 ( EOT 置位 ) 时，该位清零。
2	RXRIS	RO	0	SSI 接收 FIFO 原始中断状态  值 描述 0 无中断。 1 接收 FIFO 中有效数据不少于 4 个单元  当接收 FIFO 中有效数据少于 4 个单元时，此标志位清零。
1	RTRIS	RO	0	SSI 接收超时原始中断状态  值 描述 0 无中断。 1 发生接收超时  当对 SSI 中断清除 ( SSI1CR ) 寄存器的 RTIC 位写入 1 时，本标志位清零。

---

位/域	名称	类型	复位	描述
0	RORRIS	RO	0	SSI 接收溢出原始中断状态  值 描述 0 无中断。 1 接收 FIEO 发生溢出  当对 SSI 中断清除 (SSIICR) 的 RORIC 位写入 1 时，该位清零。

## 寄存器 8: SSI 屏蔽中断状态寄存器 (SSIMIS), 偏移量 0x01C

SSIMIS 寄存器是屏蔽的中断状态寄存器。读操作时, 此寄存器给出相应中断的当前屏蔽状态值。对本寄存器的写操作无效。

### SSI 屏蔽中断状态寄存器 (SSIMIS)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0x01C  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												TXMIS	RXMIS	RTMIS	RORMIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
3	TXMIS	RO	0	SSI 发送 FIFO 屏蔽中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 发送 FIFO 中有效数据超过 4 个单元 (EOT 位清零) 或者最后 1 位已经通过串口发出 (EOT 位置位), 这两种情况会发出未屏蔽的中断信号。  当发送 FIFO 中有效数据少于 4 个单元 (EOT 清零) 或非空 (EOT 置位) 时, 该位清零。
2	RXMIS	RO	0	SSI 接收 FIFO 屏蔽中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 由于接收 FIFO 中有效数据超过 4 个单元, 因此发出了未屏蔽的中断信号。  当接收 FIFO 中有效数据少于 4 个单元时, 此标志位清零。
1	RTMIS	RO	0	SSI 接收超时屏蔽中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 接收超时产生中断。  当对 SSI 中断清除 (SSIICR) 寄存器的 RTIC 位写入 1 时, 本标志位清零。

位/域	名称	类型	复位	描述
0	RORMIS	RO	0	SSI 接收溢出屏蔽中断状态  值 描述 0 未产生中断或中断被屏蔽。 1 接收 FIFO 溢出产生中断。  当对 SSI 中断清除 (SSIICR) 的 RORIC 位写入 1 时，该位清零。

### 寄存器 9: SSI 中断清除寄存器 ( SSIICR ) , 偏移量 0x020

SSIICR 寄存器是中断清除寄存器。当对本寄存器的某位写 1 时, 即会清除相应的中断。对本寄存器写 0 无效。

#### SSI 中断清除寄存器 (SSIICR)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0x020  
 类型 W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留														RTIC	RORIC	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	W1C	W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	RTIC	W1C	0	SSI 接收超时中断清除 向该位写入 1 会将 SSIRIS 寄存器的 RTRIS 位和 SSIMIS 寄存器的 RTMIS 位清零。
0	RORIC	W1C	0	SSI 接收溢出中断清除 向该位写入 1 会将 SSIRIS 寄存器的 RORRIS 位和 SSIMIS 寄存器的 RORMIS 位清零。



### 寄存器 11: SSI 时钟配置寄存器 (SSICC), 偏移量 0xFC8

SSICC 寄存器控制 SSI 模块的波特时钟源。

注意: 如果 PIOSC 用于 SSI 波特时钟, 那么在运行模式下系统时钟频率至少为 16 MHz。

#### SSI 时钟配置寄存器 (SSICC)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0xFC8  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												CS			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

3:0 CS R/W 0 SSI 波特时钟源

下面的表格指定了产生 SSI 波特时钟的源:

值	描述
0x0	系统时钟 (基于时钟源和分频因子)
0x1-0x4	保留
0x5	PIOSC
0x6 - 0xF	保留



## 寄存器 12: SSI 外设标识寄存器 4 ( SSIPeriphID4 ) , 偏移量 0xFD0

SSIPeriphIDn 寄存器为硬编码，寄存器内的域决定了复位值。

### SSI 外设标识寄存器 4 (SSIPeriphID4)

SSI0 基址: 0x4000.8000

SSI1 基址: 0x4000.9000

SSI2 基址: 0x4000.A000

SSI3 基址: 0x4000.B000

偏移量 0xFD0

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID4							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

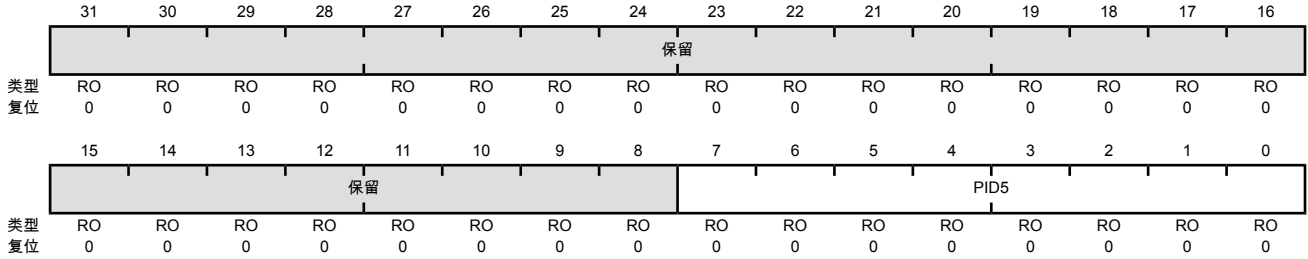
位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID4	RO	0x00	SSI 外设 ID 寄存器 [7:0] 软件可通过此寄存器识别外设。

### 寄存器 13: SSI 外设标识寄存器 5 ( SSIPeriphID5 ) , 偏移量 0xFD4

SSIPeriphIDn 寄存器为硬编码，寄存器内的域决定了复位值。

#### SSI 外设标识寄存器 5 (SSIPeriphID5)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0xFD4  
 类型 RO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID5	RO	0x00	SSI 外设 ID 寄存器 [15:8] 软件可通过此寄存器识别外设。

## 寄存器 14: SSI 外设标识寄存器 6 ( SSIPeriphID6 ) , 偏移量 0xFD8

SSIPeriphIDn 寄存器为硬编码，寄存器内的域决定了复位值。

## SSI 外设标识寄存器 6 (SSIPeriphID6)

SSI0 基址: 0x4000.8000

SSI1 基址: 0x4000.9000

SSI2 基址: 0x4000.A000

SSI3 基址: 0x4000.B000

偏移量 0xFD8

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID6							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

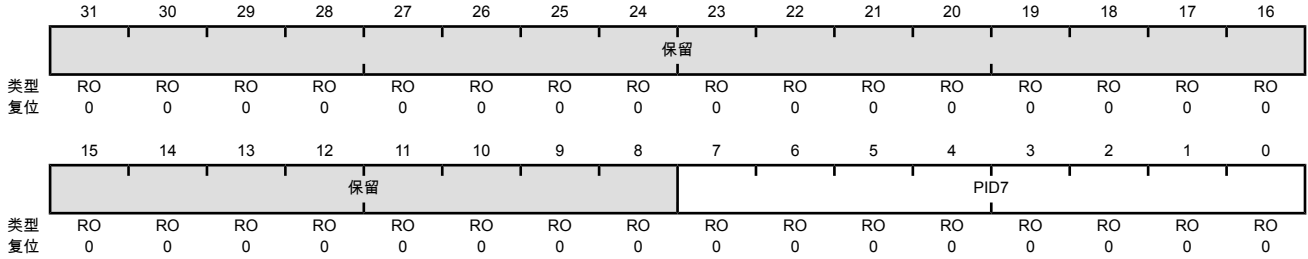
位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID6	RO	0x00	SSI 外设 ID 寄存器 [23:16] 软件可通过此寄存器识别外设。

## 寄存器 15: SSI 外设标识寄存器 7 ( SSIPeriphID7 ) , 偏移量 0xFDC

SSIPeriphIDn 寄存器为硬编码，寄存器内的域决定了复位值。

### SSI 外设标识寄存器 7 (SSIPeriphID7)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0xFDC  
 类型 RO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID7	RO	0x00	SSI 外设 ID 寄存器 [31:24] 软件可通过此寄存器识别外设。

## 寄存器 16: SSI 外设标识寄存器 0 ( SSIPeriphID0 ) , 偏移量 0xFE0

SSIPeriphIDn 寄存器为硬编码，寄存器内的域决定了复位值。

### SSI 外设标识寄存器 0 (SSIPeriphID0)

SSI0 基址: 0x4000.8000

SSI1 基址: 0x4000.9000

SSI2 基址: 0x4000.A000

SSI3 基址: 0x4000.B000

偏移量 0xFE0

类型 RO, 复位 0x0000.0022

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0

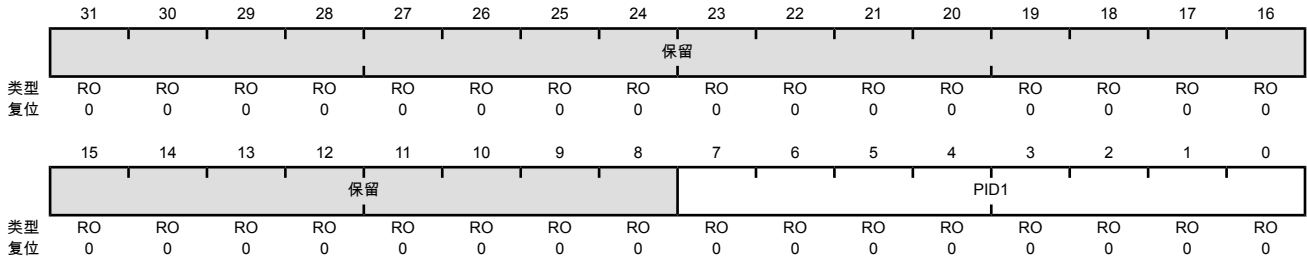
位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID0	RO	0x22	SSI 外设 ID 寄存器 [7:0] 软件可通过此寄存器识别外设。

## 寄存器 17: SSI 外设标识寄存器 1 ( SSIPeriphID1 ) , 偏移量 0xFE4

SSIPeriphIDn 寄存器为硬编码，寄存器内的域决定了复位值。

### SSI 外设标识寄存器 1 (SSIPeriphID1)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0xFE4  
 类型 RO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID1	RO	0x00	SSI 外设 ID 寄存器 [15:8] 软件可通过此寄存器识别外设。

## 寄存器 18: SSI 外设标识寄存器 2 ( SSIPeriphID2 ) , 偏移量 0xFE8

SSIPeriphIDn 寄存器为硬编码，寄存器内的域决定了复位值。

### SSI 外设标识寄存器 2 (SSIPeriphID2)

SSI0 基址: 0x4000.8000

SSI1 基址: 0x4000.9000

SSI2 基址: 0x4000.A000

SSI3 基址: 0x4000.B000

偏移量 0xFE8

类型 RO, 复位 0x0000.0018

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

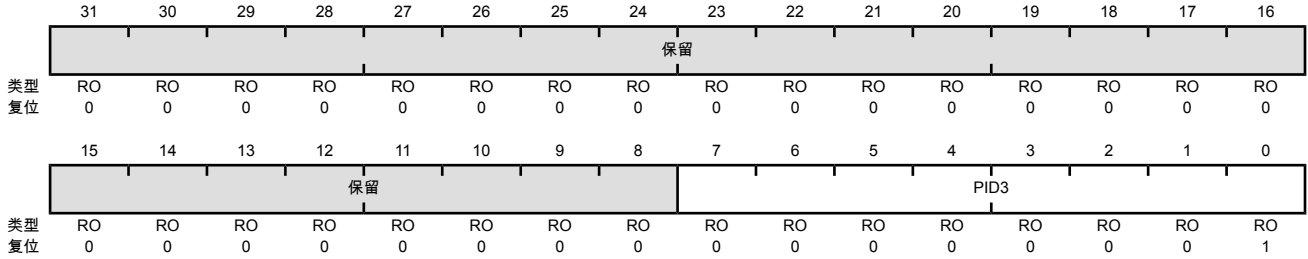
位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID2	RO	0x18	SSI 外设 ID 寄存器 [23:16] 软件可通过此寄存器识别外设。

## 寄存器 19: SSI 外设标识寄存器 3 ( SSIPeriphID3 ) , 偏移量 0xFEC

SSIPeriphIDn 寄存器为硬编码，寄存器内的域决定了复位值。

### SSI 外设标识寄存器 3 (SSIPeriphID3)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0xFEC  
 类型 RO, 复位 0x0000.0001



位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	PID3	RO	0x01	SSI 外设 ID 寄存器 [31:24] 软件可通过此寄存器识别外设。



## 寄存器 20: SSI PrimeCell 标识寄存器 0 ( SSIPCellID0 ) , 偏移量 0xFF0

SSIPCellIDn 寄存器为硬编码，寄存器内的域决定了复位值。

## SSI PrimeCell 标识寄存器 0 (SSIPCellID0)

SSI0 基址: 0x4000.8000

SSI1 基址: 0x4000.9000

SSI2 基址: 0x4000.A000

SSI3 基址: 0x4000.B000

偏移量 0xFF0

类型 RO, 复位 0x0000.000D

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID0							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID0	RO	0x0D	SSI PrimeCell 标识寄存器 [7:0] 向软件提供一个标准的交叉外设标识系统。

## 寄存器 21: SSI PrimeCell 标识寄存器 1 ( SSIPCellID1 ) , 偏移量 0xFF4

SSIPCellIDn 寄存器为硬编码，寄存器内的域决定了复位值。

### SSI PrimeCell 标识寄存器 1 (SSIPCellID1)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0xFF4  
 类型 RO, 复位 0x0000.00F0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID1							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID1	RO	0xF0	SSI PrimeCell 标识寄存器 [15:8] 向软件提供一个标准的交叉外设标识系统。

## 寄存器 22: SSI PrimeCell 标识寄存器 2 ( SSIPCellID2 ) , 偏移量 0xFF8

SSIPCellIDn 寄存器为硬编码，寄存器内的域决定了复位值。

## SSI PrimeCell 标识寄存器 2 (SSIPCellID2)

SSI0 基址: 0x4000.8000

SSI1 基址: 0x4000.9000

SSI2 基址: 0x4000.A000

SSI3 基址: 0x4000.B000

偏移量 0xFF8

类型 RO, 复位 0x0000.0005

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID2							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID2	RO	0x05	SSI PrimeCell 标识寄存器 [23:16] 向软件提供一个标准的交叉外设标识系统。

### 寄存器 23: SSI PrimeCell 标识寄存器 3 ( SSIPCellID3 ) , 偏移量 0xFFC

SSIPCellIDn 寄存器为硬编码，寄存器内的域决定了复位值。

#### SSI PrimeCell 标识寄存器 3 (SSIPCellID3)

SSI0 基址: 0x4000.8000  
 SSI1 基址: 0x4000.9000  
 SSI2 基址: 0x4000.A000  
 SSI3 基址: 0x4000.B000  
 偏移量 0xFFC  
 类型 RO, 复位 0x0000.00B1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CID3							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CID3	RO	0xB1	SSI PrimeCell 标识寄存器 [31:24] 向软件提供一个标准的交叉外设标识系统。

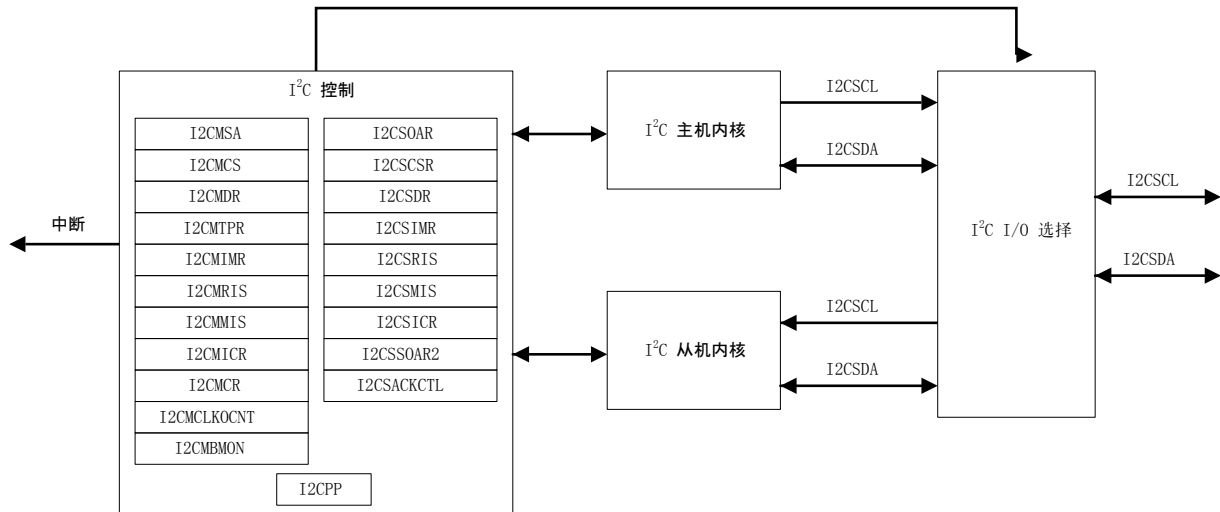
## 16 内部集成电路 (I<sup>2</sup>C) 接口

内部集成电路 (I<sup>2</sup>C) 总线通过双线设计 ( 串行数据线 SDA 和串行时钟线 SCL ) 提供双向数据传输以及连接外部 I<sup>2</sup>C 设备的接口, 例如串行存储器 ( RAM 和 ROM )、网络设备、LCD、音频发生器等。I<sup>2</sup>C 总线还可在产品开发 and 制造过程中用于系统测试和诊断。LM4F232H5QD 微控制器包括能够与总线上的其他 I<sup>2</sup>C 设备进行互动 ( 发送和接收数据 )。

该 Stellaris® LM4F232H5QD 控制器包含 I<sup>2</sup>C 模块, 具有以下特点:

- I<sup>2</sup>C 总线上的设备可被指定为主机或从机
  - 在主机或从机模式下都支持发送和接受数据
  - 支持它们作为主机和从机的同步操作
- 四种 I<sup>2</sup>C 模式:
  - 主机传送
  - 主机接收
  - 从机传送
  - 从机接收
- 四种传输速度:
  - 标准 (100 Kbps)
  - 快速 (400 Kbps)
  - 超快速 (1 Mbps)
  - 高速 (3.33 Mbps)
- 时钟低电平超时中断
- 双从机地址功能
- 时钟低电平超时中断
- 双从机地址功能
- 主机和从机产生中断
  - 主机因为传送或接收数据结束(或者是因为错误而取消)产生中断
  - 从机在主机向其发送数据或发出请求时, 或检测到 START 或 STOP 信号时产生中断
- 主机带有仲裁和时钟同步功能, 支持多主机以及 7 位寻址模式

## 16.1 结构图

图 16-1. I<sup>2</sup>C 结构图

## 16.2 信号描述

下表列出了 I<sup>2</sup>C 接口的外部信号及其功能。I<sup>2</sup>C 接口信号是某些 GPIO 信号的复用功能，复位时将重置为默认的 GPIO 信号，而 I2C0SCL 和 I2CSDA 管脚例外，这两个管脚复位时将重置为默认的 I<sup>2</sup>C 功能。下表中“复用管脚/赋值”一栏列出了 I<sup>2</sup>C 信号的 GPIO 管脚的各种可能布局。应将 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618页) 的 AFSEL 置位，以便选择 I<sup>2</sup>C 功能。必须将括号里的数字写入 GPIO 端口控制 (GPIOPTCL) 寄存器 (636页) 的 PMCn 域中，以便将 I<sup>2</sup>C 信号分配给特定的 GPIO 端口管脚。请注意，应通过 GPIO 开漏选择 (GPIOODR) 寄存器将 I2CSCL 管脚设置成开漏。如何配置 GPIO 的更多信息，请参考“通用输入/输出端口 (GPIO)”在 597页。

表 16-1. I2C 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
I2C0SCL	99	PB2 (3)	I/O	OD	I <sup>2</sup> C 模块 0 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C0SDA	100	PB3 (3)	I/O	OD	I <sup>2</sup> C 模块 0 数据。
I2C1SCL	45 51	PA6 (3) PG4 (3)	I/O	OD	I <sup>2</sup> C 模块 1 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C1SDA	46 50	PA7 (3) PG5 (3)	I/O	OD	I <sup>2</sup> C 模块 1 数据。
I2C2SCL	59 139	PF6 (3) PE4 (3)	I/O	OD	I <sup>2</sup> C 模块 2 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C2SDA	58 140	PF7 (3) PE5 (3)	I/O	OD	I <sup>2</sup> C 模块 2 数据。
I2C3SCL	1 55	PD0 (3) PG0 (3)	I/O	OD	I <sup>2</sup> C 模块 3 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C3SDA	2 54	PD1 (3) PG1 (3)	I/O	OD	I <sup>2</sup> C 模块 3 数据。
I2C4SCL	53	PG2 (3)	I/O	OD	I <sup>2</sup> C 模块 4 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。

表 16-1. I2C 信号 (144LQFP) (续)

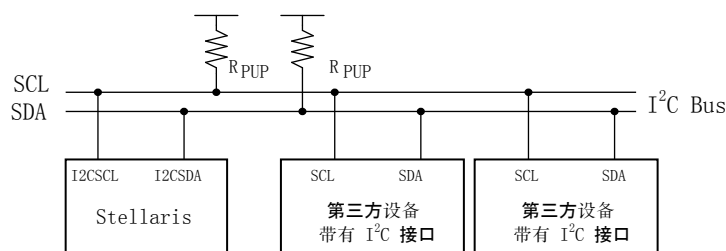
管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
I2C4SDA	52	PG3 (3)	I/O	OD	I <sup>2</sup> C 模块 4 数据。
I2C5SCL	48	PG6 (3)	I/O	OD	I <sup>2</sup> C 模块 5 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C5SDA	47	PG7 (3)	I/O	OD	I <sup>2</sup> C 模块 5 数据。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 16.3 功能描述

每个 I<sup>2</sup>C 模块都有主机和从机功能。为了实现正确操作，SDA 管脚必须配置为开漏信号。由于内部电路支持高速操作，SCL 管脚不得配置为开漏信号，即使内部电路可导致其发挥开漏信号的作用。典型的 I<sup>2</sup>C 总线配置请参阅图 16-2。有关如何确定正确操作所需的上拉大小，请参考“I2C 总线规范和用户手册”。

I<sup>2</sup>C 时序结构图请参阅“Inter-Integrated Circuit (I<sup>2</sup>C) Interface”在 1312 页。

图 16-2. I<sup>2</sup>C 总线配置

### 16.3.1 I<sup>2</sup>C 总线功能概览

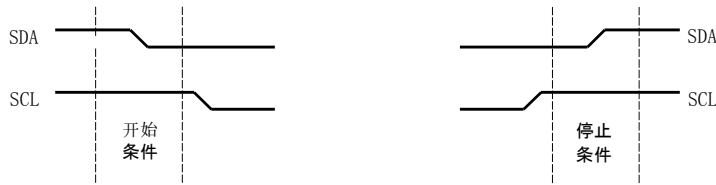
I<sup>2</sup>C 总线只使用两个信号：SDA 和 SCL。在 Stellaris 微控制器上，它们的名称分别是 I2CSDA 和 I2CSCL 信号<sup>®</sup>。SDA 是双向串行数据线，SCL 是双向串行时钟线。当两根线都为高电平时，总线即处于空闲状态。

I<sup>2</sup>C 总线每次传输的数据长度为九位，其中包括八位数据位和一位应答位。每次传输的字节数（定义为有效开始和停止条件之间的时间，请参阅“开始和停止条件”在 935 页）没有限制，但是每个字节后面必须紧跟一位应答位，而且数据传输时必须首先传送最高有效位 (MSB) 位。当接收器不能接收另一个完整的字节时，它会保持时钟线 SCL 为低电平，并迫使发送器进入等待状态。接收器释放时钟线 SCL 之后，数据传送将继续进行。

#### 16.3.1.1 开始和停止条件

I<sup>2</sup>C 总线协议定义了两种状态，以便开始和结束数据传输：开始和停止。SCL 为高电平时，SDA 线上由高电平变为低电平被定义为开始条件；SCL 为高电平时，SDA 线上由低电平变为高电平被定义为停止条件。达到开始条件以后，总线将占线；达到停止条件以后，总线将空闲。请参考图 16-3。

图 16-3. 开始和停止条件



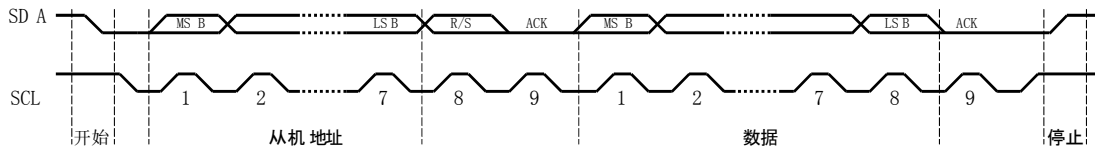
STOP 位决定数据周期在结束的时候是停止还是继续重复开始条件。要产生单次传输，应在 I<sup>2</sup>C 主机从机地址 (I2CMSA) 寄存器中写入所需的地址，并将 R/S 位清零，在控制寄存器中写入 ACK= X (0 或 1)、STOP= 1、START=1 以及 RUN= 1，以便执行单次传输并停止。操作完成后 (或者因为错误退出)，中断管脚将激活，数据可能从 I<sup>2</sup>C 主机数据 (I2CMDR) 寄存器中读出。I<sup>2</sup>C 模块以主接收器模式运行时，ACK 位通常会被置位，这会让 I<sup>2</sup>C 总线控制器在每个字节接收完之后自动发送一个应答。当 I<sup>2</sup>C 总线控制器无需接收从发送器发送的数据时，该位必须清零。

当此模块以从机模式运行时，I<sup>2</sup>C 从机原始中断状态 (I2CSRIS) 寄存器中的两位用于监测总线上的开始和停止条件；而 I<sup>2</sup>C 从机屏蔽中断状态 (I2CSMIS) 寄存器中的两位会将开始和停止条件转变成控制器中断 (前提是启用了中断功能)。

### 16.3.1.2 带 7 位地址的数据格式

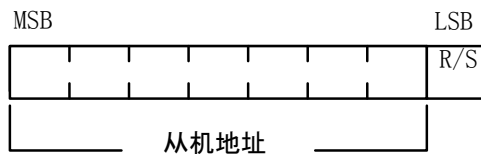
传输数据格式请参阅图 16-4。在达到开始条件之后，从机地址将被发送。该地址长度为 7 位，第八位是数据方向位 (I2CMSA 寄存器中的 R/S 位)。如果 R/S 位被清零，则表示传输操作 (发送)；如果被置位，则表示数据请求 (接收)。数据传送总是会被主机产生的停止条件终止，然而，通过重复的开始条件以及向其他从机发送地址，主机无需首先产生停止条件即可与总线上的其他设备发起通信。因此，在单次传输中可以存在许多种接收/发送格式组合。

图 16-4. 带 7 位地址的完整数据传输



第一字节中的前七位即构成从机地址 (请参阅图 16-5)。第八位确定报文的方向。第一字节中的 R/S 位为零则表示主机向选定的从机传输 (发送) 数据，该位为 1 则表示主机接收从机的数据。

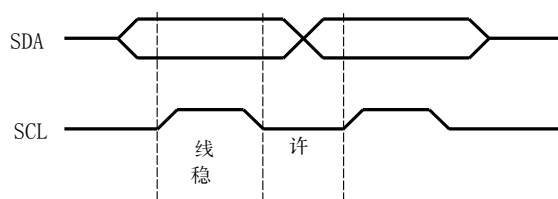
图 16-5. 第一字节中的 R/S 位



### 16.3.1.3 数据有效性

SDA 线上的数据在时钟的高电平期间必须稳定，只有在 SCL 为低电平的时候，数据线才能改变 (请参阅图 16-6)。



图 16-6. I<sup>2</sup>C 总线位传输过程中的数据有效性

#### 16.3.1.4 应答

所有的总线传输活动都包含一个由主机产生的应答时钟周期。在应答周期中，发送器（可以是主机或者从机）会释放 SDA 线。要应答数据传输活动，在应答时钟周期内，接收器必须将 SDA 拉低。在应答周期内，接收器发出的数据必须遵循数据有效性要求，请参阅“数据有效性”在 936 页。

如果从机接收器不应答从机地址，SDA 必须被从机置高电平，以便主机产生停止条件并终止当前传输活动。如果主机设备作为接收器，那么它需要应答每一个由从机发起的数据传输。由于主机控制着传输过程中的字节数量，通过不应答最后一个字节，主机会将数据的末尾告知从机发送器。接下来从机发送器必须释放 SDA，以便让主机产生停止条件或者重复的开始条件。

如果从机需要提供手动的应答或者否定应答，I<sup>2</sup>C 从机应答控制 (I2CSACKCTL) 寄存器可以让从机对无效数据或无效指令做出否定应答，或者对有效数据或有效指令做出应答。当启用该功能时，在最后一个数据位之后，I<sup>2</sup>C 时钟会被拉低，直到该寄存器写入指示响应。

#### 16.3.1.5 时钟低电平超时

I<sup>2</sup>C 从机可以将时钟周期性地拉低，以产生较低的位传输速率，从而延长数据传输。I<sup>2</sup>C 模块有一个 12 位的可编程计数器，它可以跟踪时钟被拉低了多长时间。通过 I<sup>2</sup>C 主机时钟低电平超时计数 (I2CMCLKOCNT) 寄存器，该计数器的高 8 位可通过软件编程。应用程序能对计数器最重要的 8 位进行编程，以反映可接受的传输累计低电平时间。该计数在 START 条件时载入，并且在主机内部总线时钟的每个下降沿进行递减计数。注：即使总线上的 SCL 被拉低，为此计数器生成的内部总线时钟将按编程的 I<sup>2</sup>C 速度一直运行。达到终端计数时，主机状态机在 SCL 和 SDA 释放时通过发布 STOP 条件，在总线上强制执行中止。

例如，如果一个 I<sup>2</sup>C 模块正按 100 kHz 速度操作，将 I2CMCLKOCNT 寄存器编程为 0xDA 将产生 34.88 ms 的累计时钟低电平时间。

当到达时钟超时期限时，I<sup>2</sup>C 主机原始中断状态 (I2CMRIS) 寄存器中的 CLKRIS 位被置位，以便让主机开始纠正措施，解决远程从机状态问题。另外，I<sup>2</sup>C 主机控制/状态 (I2CMCS) 寄存器中的 CLKTO 位将被置位；在发送停止条件时或者 I<sup>2</sup>C 主机复位期间，该位将被清零。软件可以读取 I<sup>2</sup>C 主机总线监视 (I2CMBMON) 寄存器中的 SDA 和 SCL 位以获得 SDA 和 SCL 信号的原始状态，从而帮助确定远程从机的状态。

如果从机连续拉低时钟，唯一的解决办法就是利用 CLKRIS 位给处理器发送一个中断，并且通过复位主机和远程从机，在更高的协议层解决条件问题。

#### 16.3.1.6 双地址

I<sup>2</sup>C 接口支持从机双地址功能。系统提供额外的可编程地址，启用后也可以进行地址匹配。在传统模式中，双地址功能将被禁用，如果地址与 I2CSOAR 寄存器中的 OAR 域相匹配，I<sup>2</sup>C 从机会在总线上提供应答。在双地址模式下，如果 I2CSOAR 寄存器中的 OAR 域或者 I2CSOAR2 寄存器中的 OAR2 域匹配，I<sup>2</sup>C 从机会在总线上提供应答。双地址功能通过对 I2CSOAR2 寄存器中的 OAR2EN 位进行编程而启用，且传统地址不会被禁用。

I2CSCSR 寄存器中的 OAR2SEL 位可以显示出应答地址是否是复用地址。该位被清零时，表示处于传统操作，或者无地址匹配。

### 16.3.1.7 仲裁

只有在总线空闲时，主机才可以开始传输。在开始条件的最小保持时间内，可能会有两个或者更多的主机产生开始条件。在这些情况下，当 SCL 为高电平时，SDA 线上将产生仲裁机制。在仲裁过程中，第一个竞争的主机在 SDA 上设置“1”（高电平），而另一个主机发送“0”（低电平），则发送“1”的这个主机将关闭其数据输出阶段并退出，直至总线再次空闲。

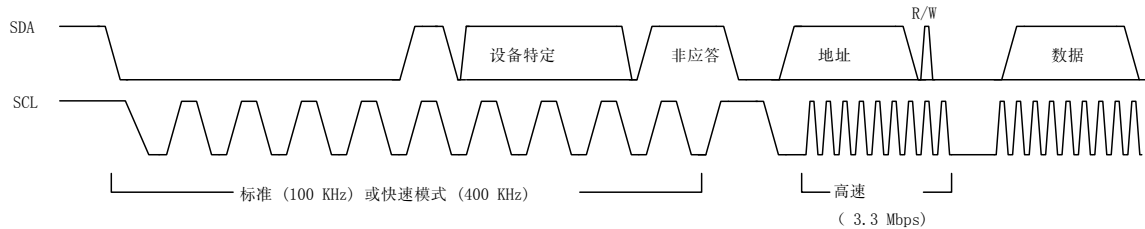
仲裁可以在多个位上发生。仲裁的第一个阶段是比较地址位，如果两个主机都试图寻址相同的设备，则仲裁继续比较数据位。

### 16.3.1.8 高速操作

Stellaris I<sup>2</sup>C 外设支持既用作主机又用作从机的高速操作。用作主机时，协议如图 16-7 所示。在开始高速模式传输前，主机负责发送标准模式 (100 Kbps) 或快速模式 (400 Kbps) 的主机代码字节。主机代码字节必须包含 0000.1XXX 格式的数据，并用于告诉从机设备准备进行高速传输。主机代码字节不应发送到从机，而只用于表示即将传输的数据将以更高的数据速率传输。要发送主机代码字节，软件应将主机代码字节的值置入 I2CMSA 寄存器，并将 0x11 写入 I2CMCS 寄存器。这会将 I<sup>2</sup>C 主机外设置于高速模式，所有之后的传输（直至 STOP 指令）都通过常规的 I2CMCS 命令位在高速数据速率进行，而无需将 I2CMCS 寄存器的 HS 位置位。同样的，仅在主机代码字节中需要将 I2CMCS 寄存器的 HS 位置位。

用作高速从机时，无需使用其他软件。

图 16-7. 高速数据格式



注意：高速模式为 3.4 Mbps，条件是设置了正确的系统时钟频率，且 SCL 和 SDA 线具有适当拉力。

## 16.3.2 可用的速度模式

I<sup>2</sup>C 总线能够以标准模式 (100 kbps)、快速模式 (400 kbps)、超快模式 (1 Mbps) 或者高速模式 (3.33 Mbps) 运行。所选速度模式必须与总线上的其他 I<sup>2</sup>C 设备相同。

### 16.3.2.1 标准、快速和超快模式

通过 I<sup>2</sup>C 主机定时器周期 (I2CMTPR) 寄存器中的数值可以选择标准、快速和超快模式，其 SCL 频率为标准模式 100 kbps、快速模式 400 kbps 或超快模式 1 Mbps。

I<sup>2</sup>C 时钟速率取决于以下参数：*CLK\_PRD*、*TIMER\_PRD*、*SCL\_LP* 和 *SCL\_HP*，其中：

*CLK\_PRD* 是系统时钟周期

*SCL\_LP* 是 SCL 的低电平相位（固定为 6）

*SCL\_HP* 是 SCL 的高电平相位（固定为 4）

*TIMER\_PRD* 是 I2CMTPR 寄存器的编程值（请参考 959 页）。

I<sup>2</sup>C 时钟周期计算方法如下：

$$SCL\_PERIOD = 2 \times (1 + TIMER\_PRD) \times (SCL\_LP + SCL\_HP) \times CLK\_PRD$$

例如：

$CLK\_PRD = 50ns$

$TIMER\_PRD = 2$

$SCL\_LP=6$

$SCL\_HP=4$

将产生以下频率的 SCL：

$1/SCL\_PERIOD = 333\text{ Khz}$

表 16-2 给出了不同的系统时钟频率下产生标准、快速和超快模式 SCL 频率的定时器周期的示例。

表 16-2. I<sup>2</sup>C 主机定时器周期与速度模式示例

系统时钟	定时器周期	标准模式	定时器周期	快速模式	定时器周期	超快模式
4 MHz	0x01	100 Kbps	-	-	-	-
6 MHz	0x02	100 Kbps	-	-	-	-
12.5 MHz	0x06	89 Kbps	0x01	312 Kbps	-	-
16.7 MHz	0x08	93 Kbps	0x02	278 Kbps	-	-
20 MHz	0x09	100 Kbps	0x02	333 Kbps	-	-
25 MHz	0x0C	96.2 Kbps	0x03	312 Kbps	-	-
33 MHz	0x10	97.1 Kbps	0x04	330 Kbps	-	-
40 MHz	0x13	100 Kbps	0x04	400 Kbps	0x01	1000 Kbps
50 MHz	0x18	100 Kbps	0x06	357 Kbps	0x02	833 Kbps
80 MHz	0x27	100 Kbps	0x09	400 Kbps	0x03	1000 Kbps

### 16.3.2.2 高速模式

高速模式的配置方法是将 I<sup>2</sup>C 主机控制/状态 (I2CMCS) 寄存器的 HS 位置位。高速模式将以高位速率传输数据，其占空比为 66.6%/33.3%，但是通信和仲裁将以标准、快速或者超快模式的速度进行，用户可以选择其中一种模式。如果 I2CMCS 寄存器中的 HS 位被置位，那么当前模式的上拉功能将启用。

可以使用上述公式选择时钟周期，但是在这种情况下， $SCL\_LP=2$ ， $SCL\_HP=1$ 。例如：

$CLK\_PRD = 25\text{ ns}$

$TIMER\_PRD = 1$

$SCL\_LP=2$

$SCL\_HP=1$

将产生以下频率的 SCL：

$1/T = 3.33\text{ Mhz}$

表 16-3 在 939页给出了高速模式下定时器周期和系统时钟的示例。

表 16-3. 高速模式下 I<sup>2</sup>C 主机定时器周期的示例

系统时钟	定时器周期	传输模式
40 MHz	0x01	3.33 Mbps
50 MHz	0x02	2.77 Mbps
80 MHz	0x03	3.33 Mbps

### 16.3.3 中断

I<sup>2</sup>C 会在发生以下条件时产生中断：

- 主机通信完成
- 主机仲裁失败
- 主机通信错误
- 主机总线超时
- 从机通信接收
- 从机通信请求
- 监测到总线上发生停止条件
- 监测到总线上发生开始条件

I<sup>2</sup>C 主机和 I<sup>2</sup>C 从机模块具有单独的中断信号。虽然很多条件下两个模块都可以产生中断，但是最后只有一个中断信号将发送至中断控制器。

#### 16.3.3.1 I<sup>2</sup>C 主机中断

当通信结束（发送或者接收）、仲裁失败或者通信发生错误时，I<sup>2</sup>C 主机模块将产生一个中断。要启用 I<sup>2</sup>C 主机中断，应通过软件将 I<sup>2</sup>C 主机中断屏蔽 (I2CMIMR) 寄存器中的 IM 位置位。当满足中断条件时，必须通过软件检查 I<sup>2</sup>C 主机控制/状态 (I2CMCS) 寄存器中的 ERROR 和 ARBLST 位，以验证错误并未发生在最后一个通信期间，以及确保没有输掉仲裁。如果最后一个通信没有得到从机的应答，那么就会发生错误。如果没有错误发生，而且主机也没有输掉仲裁，那么应用程序就会继续进行数据传输。将 I<sup>2</sup>C 主机中断清除 (I2CMICR) 寄存器中的 IC 位置位，该中断就会被清除。

如果应用程序无需使用中断，那么通过 I<sup>2</sup>C 主机原始中断状态 (I2CMRIS) 寄存器可以随时查看原始中断状态。

#### 16.3.3.2 I<sup>2</sup>C 从机中断

当接收到数据或者请求时，从机模块可以产生中断。将 I<sup>2</sup>C 从机中断屏蔽 (I2CSIMR) 寄存器中的 DATAIM 位置位即可启用该中断。应通过软件检查 I<sup>2</sup>C 从机控制/状态 (I2CSCSR) 寄存器中的 RREQ 和 TREQ 位，以确定该模块应该写入（发送）还是读取（接收）来自 I<sup>2</sup>C 从机数据 (I2CSDR) 寄存器的数据。如果从机模块处于接收模式，并且已接收第一个字节，则 FBR 和 RREQ 位将同时置位。将 I<sup>2</sup>C 从机中断清除 (I2CSICR) 寄存器中的 DATAIC 位置位即可清除该中断。

另外，当监测到开始和停止条件时，从机模块也可以产生中断。要启用这些中断，请将 I<sup>2</sup>C 从机中断屏蔽 (I2CSIMR) 寄存器中的 STARTIM 和 STOPIM 位置位；要清除这些中断，请将 I<sup>2</sup>C 从机中断清除 (I2CSICR) 寄存器中的 STOPIC 和 STARTIC 位置位。

如果应用程序无需使用中断，那么通过 I<sup>2</sup>C 从机原始中断状态 (I2CSRIS) 寄存器可以随时查看原始中断状态。

### 16.3.4 回送操作

将 I<sup>2</sup>C 主机配置 (I2CMCR) 寄存器中的 LPBK 位置位即可让 I<sup>2</sup>C 模块进入内部回送模式，以便进行诊断或者调试工作。在回送模式下，主机和从机发出的 SDA 和 SCL 信号被绑定在一起。

### 16.3.5 指令序列流程图

本节描述了主机和从机模式下进行各种类型的 I<sup>2</sup>C 传输的详细步骤。

#### 16.3.5.1 I<sup>2</sup>C 主机指令序列

下图显示了 I<sup>2</sup>C 主机的可用指令序列。

图 16-8. 主机单次传输

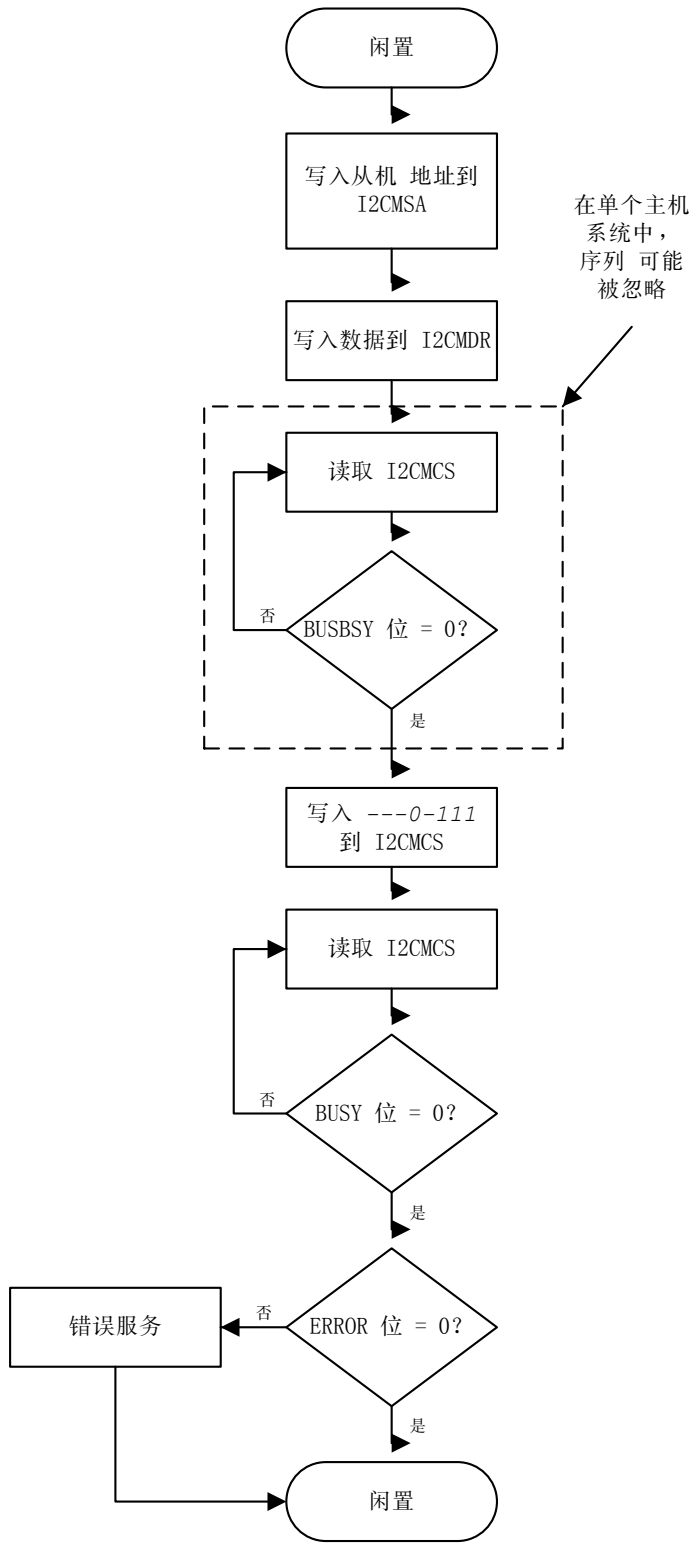


图 16-9. 主机单次接收

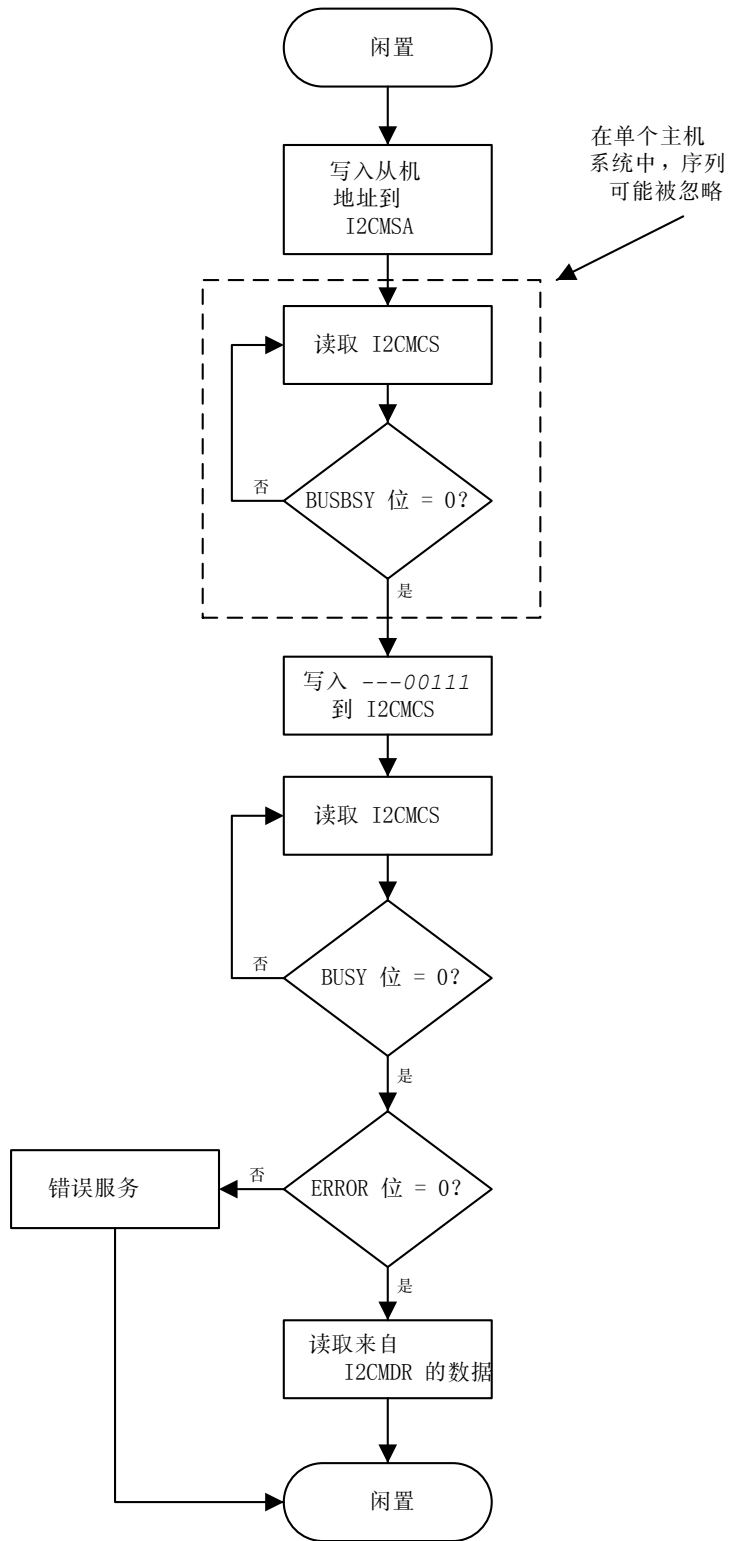


图 16-10. 多次重复开始条件下的主机传输

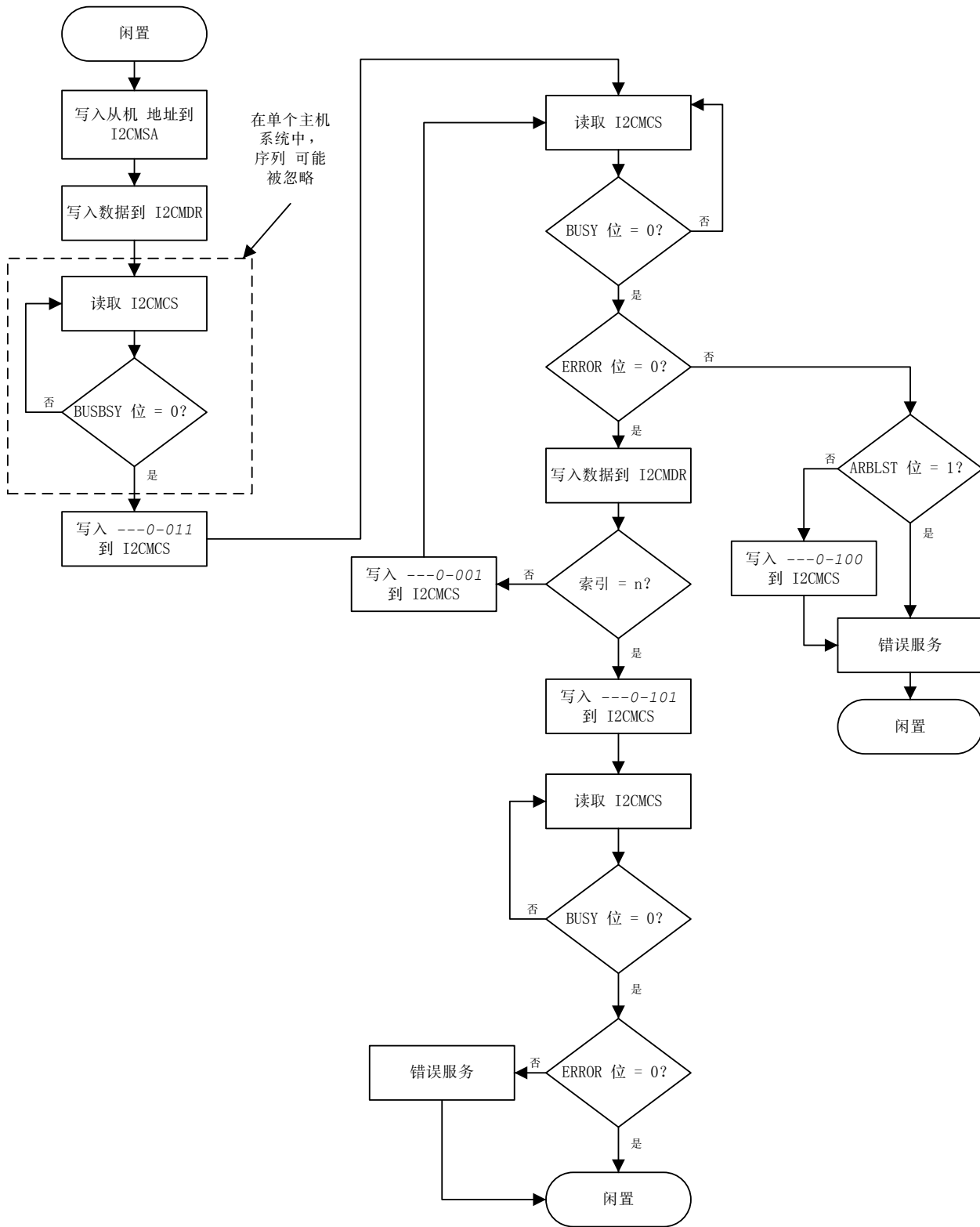




图 16-11. 多次重复开始条件下的主机接收

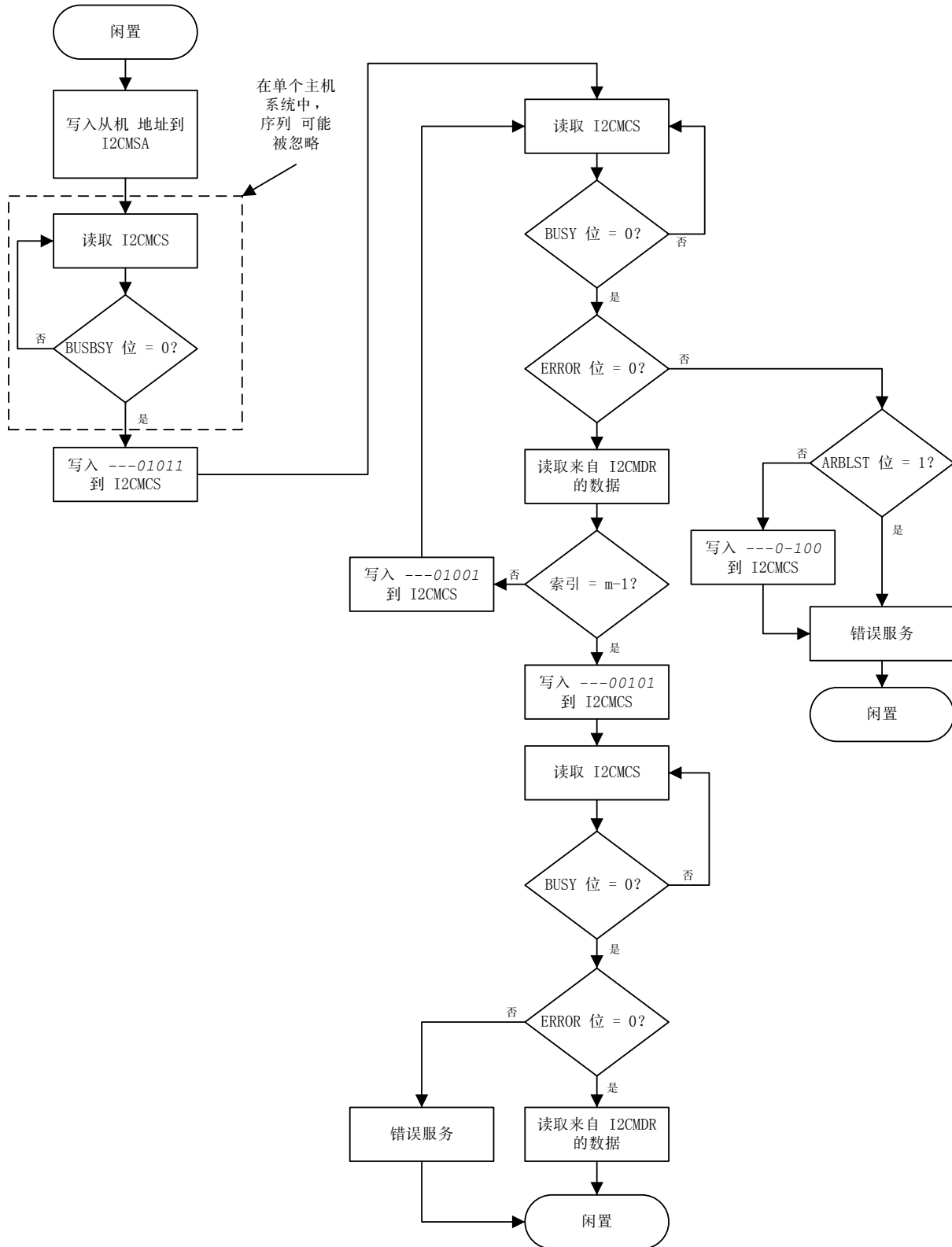


图 16-12. 多次重复开始条件下的主机传输之后多次重复开始条件下的主机接收

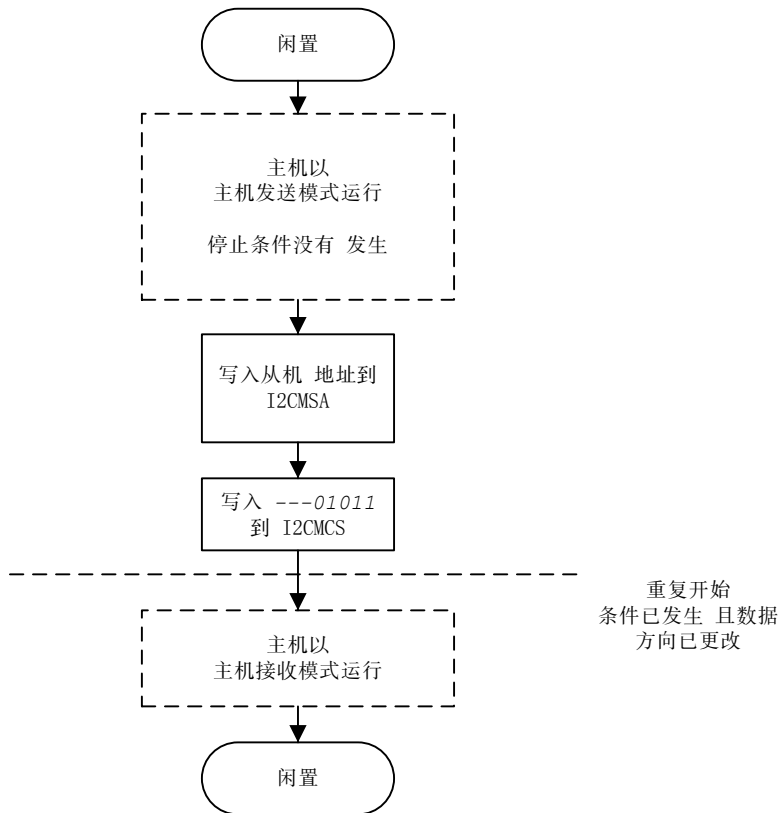


图 16-13. 多次重复开始条件下的主机接收之后多次重复开始条件下的主机传输

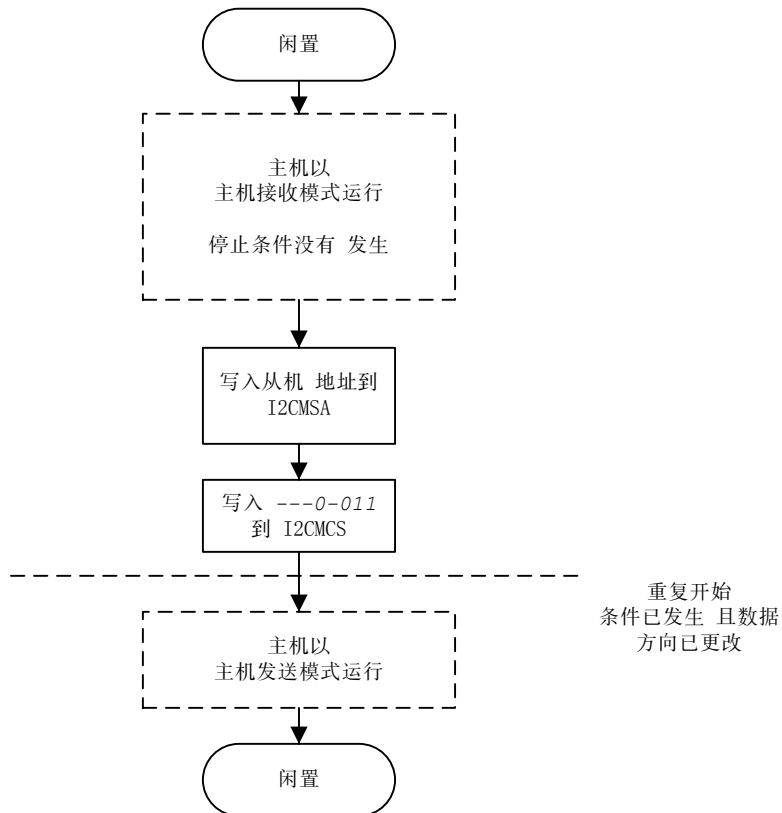
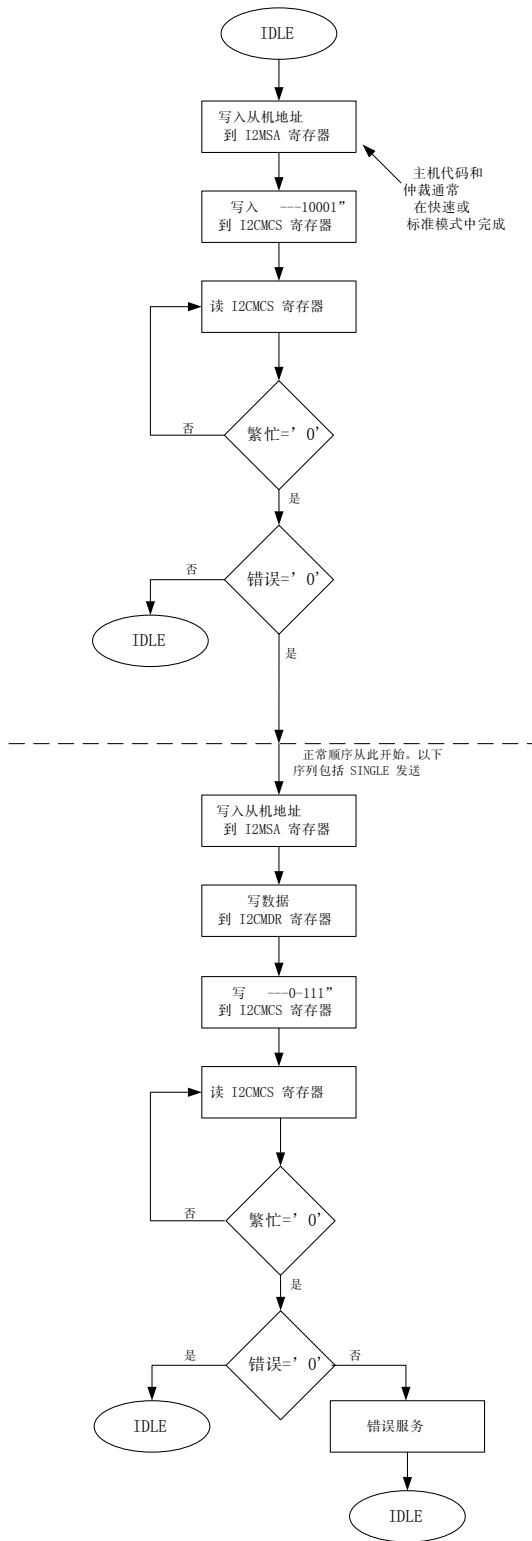


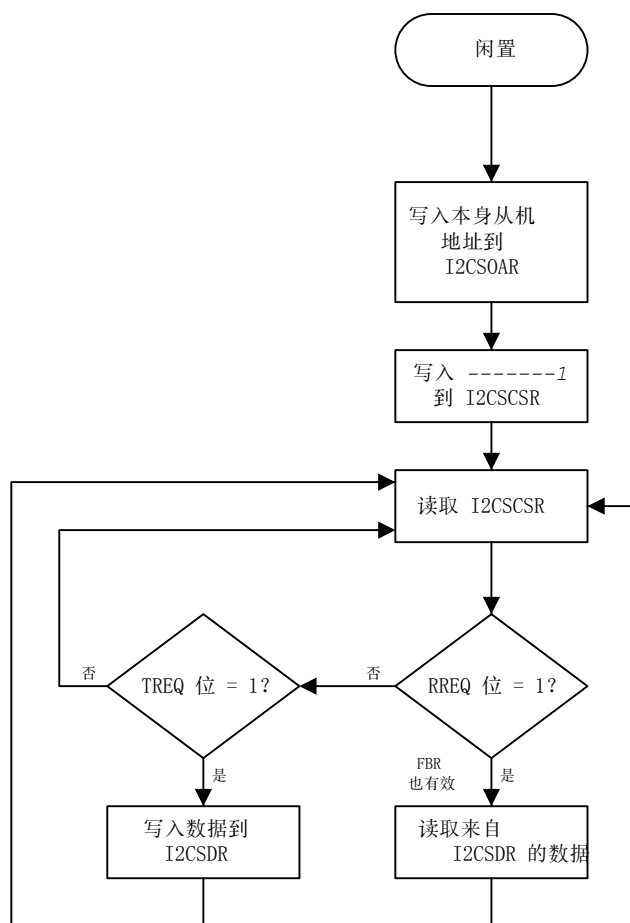
图 16-14. 高速模式主机传输



### 16.3.5.2 I<sup>2</sup>C 从机指令序列

图 16-15 在 949 页显示了 I<sup>2</sup>C 从机的可用指令序列。

图 16-15. 从机指令序列



## 16.4 初始化及配置

以下面示例显示了如何将 I<sup>2</sup>C 模块配置为主机去，以传输单字节数据。假设系统时钟为 20 MHz。

1. 用系统控制模块中的 RCGCI2C 寄存器启用 I<sup>2</sup>C 时钟 ( 请参阅 311 页 )。
2. 通过系统时钟模块中的 RCGCGPIO 寄存器启用相应 GPIO 模块的时钟 ( 请参考 302 页 )。欲了解需要启用哪些 GPIO 端口，请参考表 23-5 在 1280 页。
3. 在 GPIO 模块中，通过 GPIOAFSEL 寄存器启用相应管脚的复用功能 ( 请参考 618 页 )。欲了解需要配置哪些 GPIO，请参考表 23-4 在 1271 页。
4. 启用 I2CSDA 管脚以执行开漏操作。请参考 623 页。
5. 设置 GPIOCTL 寄存器中的 PMCn 域，将 I<sup>2</sup>C 信号分配到适当的管脚。请参阅 636 页和表 23-5 在 1280 页。
6. 在 I2CMCR 寄存器中写入 0x0000.0010，以初始化 I<sup>2</sup>C 主机。

7. 将 I2CMTPR 寄存器写入适当的数值，把 SCL 时钟速率设置为 100 Kbps。I2CMTPR 寄存器中写入的数据表示一个 SCL 时钟周期所包含的系统时钟周期数量。TPR 数值由以下公式确定：

$$\begin{aligned} \text{TPR} &= (\text{System Clock} / (2 * (\text{SCL\_LP} + \text{SCL\_HP}) * \text{SCL\_CLK})) - 1; \\ \text{TPR} &= (20\text{MHz} / (2 * (6+4) * 100000)) - 1; \\ \text{TPR} &= 9 \end{aligned}$$

在 I2CMTPR 寄存器中写入 0x0000.0009。

8. 在 I2CMSA 寄存器中写入 0x0000.0076，以指定从机地址，并指明下一个操作是发送。这会把从机地址设置为 0x3B。
9. 在 I2CMDR 寄存器中写入所需的数据，以待发送的数据（字节）加载到数据寄存器中。
10. 在 I2CMCS 寄存器中写入 0x0000.0007（停止、开始和运行），以启动从主机到从机的单字节数据传输。
11. 等待传输完成。可以查询 I2CMCS 寄存器中的 BUSBSY 位，该清零即表示传输完成。
12. 检查 I2CMCS 寄存器中的 ERROR 位，确认传输得到应答。

## 16.5 寄存器映射

表 16-4 在 950 页列出了各个 I<sup>2</sup>C 寄存器。所有给出的地址都相对于 I<sup>2</sup>C 基础地址：

- I<sup>2</sup>C 0 : 0x4002.0000
- I<sup>2</sup>C 1 : 0x4002.1000
- I<sup>2</sup>C 2 : 0x4002.2000
- I<sup>2</sup>C 3 : 0x4002.3000
- I<sup>2</sup>C 4 : 0x400C.0000
- I<sup>2</sup>C 5 : 0x400C.1000

请注意，在对寄存器编程之前必须启用 I<sup>2</sup>C 模块的时钟（请参考 311 页）。I<sup>2</sup>C 模块时钟启用之后必须延迟 3 个系统时钟，I<sup>2</sup>C 模块的寄存器才能访问。

驱动程序库中的 hw\_i2c.h 文件 StellarisWare<sup>®</sup> 将 0x800 作为 I<sup>2</sup>C 从机寄存器的基础地址。请注意，如果使用偏移量介于 0x800 和 0x818 之间的寄存器，StellarisWare 将使用从机基础地址，且偏移量介于 0x000 和 0x018 之间。

表 16-4. 内部集成电路 (I<sup>2</sup>C) 接口 寄存器映射

偏移量	名称	类型	复位	描述	见页面
<b>I<sup>2</sup>C 主机</b>					
0x000	I2CMSA	R/W	0x0000.0000	I2C 主机从机地址寄存器	952
0x004	I2CMCS	R/W	0x0000.0020	I2C 主机控制/状态寄存器	953
0x008	I2CMDR	R/W	0x0000.0000	I2C 主机数据寄存器	958
0x00C	I2CMTPR	R/W	0x0000.0001	I2C 主机定时器周期寄存器	959
0x010	I2CMIMR	R/W	0x0000.0000	I2C 主机中断屏蔽寄存器	960
0x014	I2CMRIS	RO	0x0000.0000	I2C 主机原始中断状态寄存器	961
0x018	I2CMMIS	RO	0x0000.0000	I2C 主机屏蔽中断状态寄存器	962

表 16-4. 内部集成电路 (I<sup>2</sup>C) 接口 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x01C	I2CMICR	WO	0x0000.0000	I2C 主机中断清除寄存器	963
0x020	I2CMCR	R/W	0x0000.0000	I2C 主机配置寄存器	964
0x024	I2CMCLKOCNT	R/W	0x0000.0000	I2C 主机时钟低电平超时计数寄存器	965
0x02C	I2CMBMON	RO	0x0000.0003	I2C 主机总线监视寄存器	966
<b>I<sup>2</sup>C 从机</b>					
0x800	I2CSOAR	R/W	0x0000.0000	I2C 从机本身地址寄存器	967
0x804	I2CSCSR	RO	0x0000.0000	I2C 从机控制/状态寄存器	968
0x808	I2CSDR	R/W	0x0000.0000	I2C 从机数据寄存器	970
0x80C	I2CSIMR	R/W	0x0000.0000	I2C 从机中断屏蔽	971
0x810	I2CSRIS	RO	0x0000.0000	I2C 从机原始中断状态寄存器	972
0x814	I2CSMIS	RO	0x0000.0000	I2C 从机屏蔽中断状态寄存器	973
0x818	I2CSICR	WO	0x0000.0000	I2C 从机中断清除寄存器	974
0x81C	I2CSOAR2	R/W	0x0000.0000	I2C 从机自身地址寄存器 2	975
0x820	I2CSACKCTL	R/W	0x0000.0000	I2C 从机应答控制寄存器	976
<b>I<sup>2</sup>C 状态和控制</b>					
0xFC0	I2CPP	RO	0x0000.0001	I2C 外设属性寄存器	977
0xFC4	I2CPC	RO	0x0000.0001	I2C 外配置寄存器	978

## 16.6 寄存器描述 (I<sup>2</sup>C 主机)

本章的剩余部分按照地址偏移量由小到大的顺序依次详细介绍各 I<sup>2</sup>C 主机寄存器。

寄存器 1: I<sup>2</sup>C 主机从机地址寄存器 (I2CMSA), 偏移量 0x000

该寄存器包含 8 位 : 7 位地址 (A6:A0) 和 1 位接收/发送位。接收/发送位决定了下一个操作是接收 (高电平) 还是发送 (低电平)。

## I2C 主机从机地址寄存器 (I2CMSA)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x000  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								SA							R/S
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:1	SA	R/W	0x00	I <sup>2</sup> C 从机地址 该域指定从机地址的 A6 到 A0 位。
0	R/S	R/W	0	接收/发送 R/S 位说明下一个操作是接收 (高电平) 还是发送 (低电平)。

## 值 描述

0 传输

1 接收



## 寄存器 2: I<sup>2</sup>C 主机控制/状态寄存器 (I2CMCS), 偏移量 0x004

该寄存器在读取时访问状态位, 在写入时访问控制位。读取时, 状态寄存器可指示 I<sup>2</sup>C 总线控制器的状态。写入时, 控制器寄存器可用来设置 I<sup>2</sup>C 控制器的操作。

START 位产生开始条件或者重复开始条件。STOP 位决定数据周期在结束的时候是停止还是继续重复开始条件。要产生单次传输, 应在 I<sup>2</sup>C 主机从机地址 (I2CMSA) 寄存器中写入所需的地址, 并将 R/S 位清零, 在控制寄存器中写入 ACK=X (0 或 1)、STOP=1、START=1 以及 RUN=1, 以便执行单次传输并停止。当操作完成时(或者由于错误退出), 中断将激活, 数据可能从 I2CMDR 寄存器中读出。I<sup>2</sup>C 模块以主机接收器模式运行时, ACK 位通常会被置位, 这会让 I<sup>2</sup>C 总线控制器在每个字节接收完成之后自动发送一个应答。当 I<sup>2</sup>C 总线控制器无需接收从发送器发送的数据时, 该位必须清零。

### 只读状态寄存器

#### I2C 主机控制/状态寄存器 (I2CMCS)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x004  
 类型 RO, 复位 0x0000.0020

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CLKTO	BUSBSY	IDLE	ARBLST	DATACK	ADRACK	ERROR	BUSY
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	CLKTO	RO	0	时钟超时错误  值 描述 0 无时钟超时错误。 1 发生时钟超时错误。  当主机发送停止条件或者 I <sup>2</sup> C 主机复位时, 该位被清零。
6	BUSBSY	RO	0	总线占线  值 描述 0 I <sup>2</sup> C 总线空闲。 1 I <sup>2</sup> C 总线占线。  该位根据开始和停止条件变化。

位/域	名称	类型	复位	描述
5	IDLE	RO	1	I <sup>2</sup> C 空闲  值 描述 0 I <sup>2</sup> C 控制器没有闲置。 1 I <sup>2</sup> C 控制器空闲。
4	ARBLST	RO	0	仲裁失败  值 描述 0 I <sup>2</sup> C 控制器赢得仲裁。 1 I <sup>2</sup> C 控制器输掉仲裁。
3	DATAACK	RO	0	应答数据  值 描述 0 传输数据得到应答。 1 传输数据没有得到应答。
2	ADRACK	RO	0	应答地址  值 描述 0 传输地址得到应答。 1 传输地址没有得到应答。
1	ERROR	RO	0	错误  值 描述 0 上一个操作没有发生错误。 1 上一个操作发生了错误。  该错误可能源于从机地址没有得到应答或者传输数据没有得到应答。
0	BUSY	RO	0	I <sup>2</sup> C 占线  值 描述 0 控制器空闲。 1 控制器忙。  如果 BUSY 位被置位，其他状态位将无效。

## 只写控制寄存器

## I2C 主机控制/状态寄存器 (I2CMCS)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x004  
 类型 WO, 复位 0x0000.0020

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												HS	ACK	STOP	START	RUN
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:5	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
4	HS	WO	0	高速模式启用  值 描述 0 在 I2CMTPR 寄存器中写入相应数值即可选择以标准、快速或者超快模式运行主机。SCL 频率分别是标准模式 100 kbps、快速模式 400 kbps 或超快模式 1 Mbps。 1 主机以高速模式运行时，传输速率可高达 3.33 Mbps。
3	ACK	WO	0	数据应答启用  值 描述 0 主机不会自动应答接收到的数据字节。 1 主机会自动应答接收到的数据字节。域解码请参阅表 16-5 在 956 页。
2	STOP	WO	0	产生停止条件  值 描述 0 控制器没有产生停止条件。 1 控制器产生了停止条件。域解码请参阅表 16-5 在 956 页。
1	START	WO	0	产生开始条件  值 描述 0 控制器没有产生开始条件。 1 控制器产生或者重复开始条件。域解码请参阅表 16-5 在 956 页。

位/域	名称	类型	复位	描述
0	RUN	WO	0	I <sup>2</sup> C 主机启用
				值 描述
				0 主机禁用。
				1 主机可以传输或发送数据。域解码请参阅表 16-5 在 956 页。

表 16-5. 为 I2CMCS[3:0] 域写入域解码

目前状态	I2CMSA[0]	I2CMCS[3:0]				描述
	R/S	ACK	STOP	START	RUN	
闲置	0	X <sup>a</sup>	0	1	1	开始条件后进行传输 (主机进入主机传输状态)。
	0	X	1	1	1	产生开始条件后紧跟传输和停止条件 (主机保持空闲状态)。
	1	0	0	1	1	开始条件以后紧跟接收操作, 否定应答 (主机进入主机接收状态)。
	1	0	1	1	1	开始条件以后紧跟接收操作和停止条件 (主机保持空闲状态)。
	1	1	0	1	1	开始条件以后紧跟接收操作 (主机进入主机接收状态)。
	1	1	1	1	1	非法
	所有其他未列出的组合都不可用。					NOF
主机传送	X	X	0	0	1	发送操作 (主机保持主机发送状态)。
	X	X	1	0	0	停止条件 (主机进入空闲状态)。
	X	X	1	0	1	发送操作以后紧跟停止条件 (主机进入空闲状态)。
	0	X	0	1	1	重复开始条件以后紧跟发送操作 (主机保持主机发送状态)。
	0	X	1	1	1	重复开始条件以后紧跟发送操作和停止条件 (主机进入空闲状态)。
	1	0	0	1	1	重复开始条件以后紧跟接收操作, 否定应答 (主机进入主机接收状态)。
	1	0	1	1	1	重复开始条件以后紧跟发送操作和停止条件 (主机进入空闲状态)。
	1	1	0	1	1	重复开始条件以后紧跟接收操作 (主机进入主机接收状态)。
	1	1	1	1	1	非法。
所有其他未列出的组合都不可用。					NOF.	

表 16-5. 为 I2CMCS[3:0] 域写入域解码 (续)

目前状态	I2CMSA[0]	I2CMCS[3:0]				描述
	R/S	ACK	STOP	START	RUN	
主机接收	X	0	0	0	1	接收操作, 否定应答 (主机保持主机接收状态)。
	X	X	1	0	0	停止条件 (主机进入空闲状态)。 <sup>b</sup>
	X	0	1	0	1	接收紧跟停止条件 (主机进入空闲状态)。
	X	1	0	0	1	接收操作 (主机保持主机接收状态)。
	X	1	1	0	1	非法。
	1	0	0	1	1	重复开始条件后紧跟接收操作, 否定应答 (主机保持主机接收状态)。
	1	0	1	1	1	重复开始条件后紧跟接收操作和停止条件 (主机进入空闲状态)。
	1	1	0	1	1	重复开始条件后紧跟接收操作 (主机保持主机接收状态)。
	0	X	0	1	1	重复开始条件后紧跟发送操作 (主机进入主机发送状态)。
	0	X	1	1	1	重复开始条件以后紧跟发送操作和停止条件 (主机进入空闲状态)。
所有其他未列出的组合都不可用。						NOP.

a. 表格中的 X 表示该位可以是 0 或者 1。

b. 在主机接收模式下, 停止条件只能在主机发出数据否定应答或者从机发出地址否定应答以后产生。

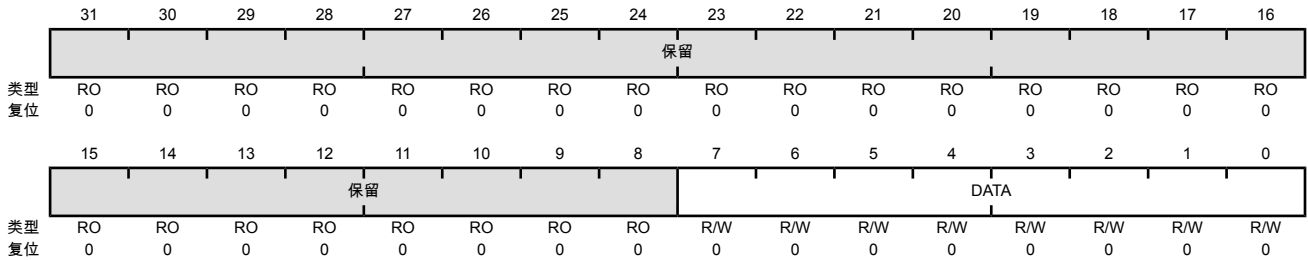
### 寄存器 3: I<sup>2</sup>C 主机数据寄存器 (I2CMDR) , 偏移量 0x008

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

处于主机发送状态时, 该寄存器包含待发送的数据; 处于主机接收状态时, 该寄存器包含已经接收的数据。

#### I2C 主机数据寄存器 (I2CMDR)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x008  
 类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DATA	R/W	0x00	传输的数据 通信时传输的数据。

寄存器 4: I<sup>2</sup>C 主机定时器周期寄存器 (I2CMTPR), 偏移量 0x00C

编程该寄存器以设置 SCL 时钟的定时器周期, 并将 SCL 时钟指定为标准或高速模式。

## I2C 主机定时器周期寄存器 (I2CMTPR)

I2C 0 基址: 0x4002.0000

I2C 1 基址: 0x4002.1000

I2C 2 基址: 0x4002.2000

I2C 3 基址: 0x4002.3000

I2C 4 基址: 0x400C.0000

I2C 5 基址: 0x400C.1000

偏移量 0x00C

类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								HS	TPR						
类型	RO	RO	RO	RO	RO	RO	RO	RO	WO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

7	HS	WO	0x0	高速模式启用
---	----	----	-----	--------

## 值 描述

1 TPR 设置的 SCL 时钟周期适用于高速模式 (3.33 Mbps)。

0 TPR 设置的 SCL 时钟周期适用于标准模式 (100 Kbps)、快速模式 (400 Kbps) 或超快模式 (1 Mbps)。

6:0	TPR	R/W	0x1	SCL 时钟周期 该域指定了 SCL 时钟周期。 $SCL\_PRD = 2 \times (1 + TPR) \times (SCL\_LP + SCL\_HP) \times CLK\_PRD$
-----	-----	-----	-----	-----------------------------------------------------------------------------------------------------------

其中:

$SCL\_PRD$  是 SCL 线周期 (I<sup>2</sup>C 时钟)。

TPR 是定时器周期寄存器值 (范围 1 到 127)。

$SCL\_LP$  是 SCL 低电平周期 (固定为 6)。

$SCL\_HP$  是 SCL 高电平周期 (固定为 4)。

$CLK\_PRD$  是系统时钟周期, 单位 ns。

寄存器 5: I<sup>2</sup>C 主机中断屏蔽寄存器 (I2CMIMR), 偏移量 0x010

该寄存器决定原始中断是否可以传送到中断控制器。

## I2C 主机中断屏蔽寄存器 (I2CMIMR)

I2C 0 基址: 0x4002.0000

I2C 1 基址: 0x4002.1000

I2C 2 基址: 0x4002.2000

I2C 3 基址: 0x4002.3000

I2C 4 基址: 0x400C.0000

I2C 5 基址: 0x400C.1000

偏移量 0x010

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														CLKIM	IM
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	CLKIM	R/W	0	时钟超时中断屏蔽  值 描述 1 当 I2CMRIS 寄存器中的 CLKRIS 位被置位时, 时钟超时中断就会被发送到中断控制器。 0 CLKRIS 中断被抑制, 不会发送到中断控制器。
0	IM	R/W	0	主机中断屏蔽寄存器  值 描述 1 当 I2CMRIS 寄存器中的 RIS 位被置位时, 主机中断会发送到中断控制器。 0 RIS 中断被抑制, 不会发送到中断控制器。



## 寄存器 6: I<sup>2</sup>C 主机原始中断状态寄存器 ( I2CMRIS ) , 偏移量 0x014

该寄存器决定中断是否被挂起。

### I2C 主机原始中断状态寄存器 (I2CMRIS)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x014  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															CLKRIS	RIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	CLKRIS	RO	0	时钟超时原始中断状态  值 描述 1 时钟超时中断被挂起。 0 无中断。  如果在 I2CMICR 寄存器中的 CLKIC 位写入 1, 该位会被清零。
0	RIS	RO	0	主机原始中断状态寄存器  值 描述 1 主机中断被挂起。 0 无中断。  如果在 I2CMICR 寄存器中的 IC 位写入 1, 则该位会被清零。

寄存器 7: I<sup>2</sup>C 主机屏蔽中断状态寄存器 (I2CMMIS), 偏移量 0x018

该寄存器决定中断是否可以发出信号。

## I2C 主机屏蔽中断状态寄存器 (I2CMMIS)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x018  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														CLKMIS	MIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	CLKMIS	RO	0	时钟超时屏蔽中断状态  值 描述 1 未经屏蔽的时钟超时中断信号已经发出, 已经挂起。 0 无中断。  如果在 I2CMICR 寄存器中的 CLKIC 位写入 1, 该位会被清零。
0	MIS	RO	0	屏蔽中断状态  值 描述 1 未经屏蔽的主机中断信号已经发出, 已经挂起。 0 未产生中断或中断被屏蔽。  如果在 I2CMICR 寄存器中的 IC 位写入 1, 则该位会被清零。

寄存器 8: I<sup>2</sup>C 主机中断清除寄存器 ( I2CMICR ) , 偏移量 0x01C

该寄存器可清除原始中断和屏蔽中断。

## I2C 主机中断清除寄存器 (I2CMICR)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x01C  
 类型 WO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															CLKIC	IC
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
1	CLKIC	WO	0	时钟超时中断清除 在该位写入 1 即可将 I2CMRIS 寄存器中的 CLKRIS 位和 I2CMMIS 寄存器中的 CLKMIS 位清零。 读取该寄存器将返回无意义的数。
0	IC	WO	0	主机中断清除寄存器 在该位写入 1 即可将 I2CMRIS 寄存器中的 RIS 位和 I2CMMIS 寄存器中的 MIS 位清零。 读取该寄存器将返回无意义的数。

寄存器 9: I<sup>2</sup>C 主机配置寄存器 (I2CMCR), 偏移量 0x020

该寄存器用于配置设备模式 (主机或者从机), 以及为测试模式回送设置接口。

## I2C 主机配置寄存器 (I2CMCR)

I2C 0 基址: 0x4002.0000

I2C 1 基址: 0x4002.1000

I2C 2 基址: 0x4002.2000

I2C 3 基址: 0x4002.3000

I2C 4 基址: 0x400C.0000

I2C 5 基址: 0x400C.1000

偏移量 0x020

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留											SFE	MFE	保留		LPBK
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:6	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
5	SFE	R/W	0	I <sup>2</sup> C 从机功能启用  值 描述 1 从机模式启用。 0 从机模式禁用。
4	MFE	R/W	0	I <sup>2</sup> C 主机功能启用  值 描述 1 主机模式启用。 0 主机模式禁用。
3:1	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	LPBK	R/W	0	I <sup>2</sup> C 回送  值 描述 1 控制器处于测试模式回送状态。 0 正常工作。

**寄存器 10: I<sup>2</sup>C 主机时钟低电平超时计数寄存器 ( I2CMCLKOCNT ) ， 偏移量 0x024**

该寄存器包含一个 12 位计数器的高 8 位， 可用来保持远程从机拉低时钟的超时限制。 用户无法查看该计数器的低四位。

**I2C 主机时钟低电平超时计数寄存器 (I2CMCLKOCNT)**

I2C 0 基址: 0x4002.0000

I2C 1 基址: 0x4002.1000

I2C 2 基址: 0x4002.2000

I2C 3 基址: 0x4002.3000

I2C 4 基址: 0x400C.0000

I2C 5 基址: 0x400C.1000

偏移量 0x024

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CNTL							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	CNTL	R/W	0	I <sup>2</sup> C 主机计数值 该域包含一个 12 位计数器的高 8 位， 可用来进行时钟低电平超时计数。

## 寄存器 11: I<sup>2</sup>C 主机总线监视寄存器 ( I2CMBMON ) , 偏移量 0x02C

该寄存器用来确定 SCL 和 SDA 的信号状态。

### I2C 主机总线监视寄存器 (I2CMBMON)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x02C  
 类型 RO, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															SDA	SCL
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	

位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
1	SDA	RO	1	I <sup>2</sup> C SDA 状态  值 描述 1 I2CSDA 信号为高电平。 0 I2CSDA 信号为低电平。
0	SCL	RO	1	I <sup>2</sup> C SCL 状态  值 描述 1 I2CSCL 信号为高电平。 0 I2CSCL 信号为低电平。

## 16.7 寄存器描述 ( I<sup>2</sup>C 从机 )

本章的剩余部分按照地址偏移量由小到大的顺序依次详细介绍各 I<sup>2</sup>C 从机寄存器。

## 寄存器 12: I<sup>2</sup>C 从机本身地址寄存器 ( I2CSOAR ) , 偏移量 0x800

该寄存器包含 7 位地址位, 用以识别 Stellaris I<sup>2</sup>C 总线上的 I<sup>2</sup>C 设备。

### I2C 从机本身地址寄存器 (I2CSOAR)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x800  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留									OAR						
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:7	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
6:0	OAR	R/W	0x00	I <sup>2</sup> C 从机本身地址 该域指定从机地址的 A6 到 A0 位。

寄存器 13: I<sup>2</sup>C 从机控制/状态寄存器 (I2CSCSR), 偏移量 0x804

写入时, 该寄存器是控制寄存器; 读取时, 该寄存器是状态寄存器。

## 只读状态寄存器

## I2C 从机控制/状态寄存器 (I2CSCSR)

I2C 0 基址: 0x4002.0000  
I2C 1 基址: 0x4002.1000  
I2C 2 基址: 0x4002.2000  
I2C 3 基址: 0x4002.3000  
I2C 4 基址: 0x400C.0000  
I2C 5 基址: 0x400C.1000  
偏移量 0x804  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												OAR2SEL	FBR	TREQ	RREQ
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3	OAR2SEL	RO	0	OAR2 地址匹配  值 描述 1 OAR2 地址已匹配, 而且得到从机应答。 0 地址没有匹配, 或者匹配是传统模式。  比较每个地址之后, 该位会被重新评估。
2	FBR	RO	0	第一个字节已接收  值 描述 1 从机本身地址后面的第一个字节已经被接收。 0 第一个字节没有被接收。  只有 RREQ 位被置位, 该位才有效, 而且数据从 I2CSDR 寄存器读取之后, 该位将被自动清零。 注意: 从机发送操作不会使用该位。
1	TREQ	RO	0	发送请求  值 描述 1 I <sup>2</sup> C 控制器被设置成从发送器, 正在使用时钟拉低来延迟主机, 直到数据写入 I2CSDR 寄存器中。 0 没有未完成的发送请求。



位/域	名称	类型	复位	描述
0	RREQ	RO	0	接收请求

## 值 描述

- 1 I<sup>2</sup>C 控制器有未完成的来自 I<sup>2</sup>C 主机的接收数据，正在使用时钟拉低来延迟主机，直到 I2CSDR 寄存器的数据被读取。
- 0 没有未完成的接收数据。

## 只写控制寄存器

## I2C 从机控制/状态寄存器 (I2CSCSR)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x804  
 类型 WO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															DA	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	DA	WO	0	设备激活

## 值 描述

- 0 禁用 I<sup>2</sup>C 从机操作。
- 1 启用 I<sup>2</sup>C 从机操作。

一旦将该位置位，则在通过写 0 或复位而清零该位前，不得再次将该位置位，否则可能发生传输故障。

## 寄存器 14: I<sup>2</sup>C 从机数据寄存器 (I2CSDR) , 偏移量 0x808

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

处于从机发送状态时, 该寄存器包含待发送的数据; 处于从机接收状态时, 该寄存器包含已经接收的数据。

### I2C 从机数据寄存器 (I2CSDR)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x808  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DATA							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7:0	DATA	R/W	0x00	要发送的数据 该地段包含了从机接收或者发送操作期间需要发送的数据。

寄存器 15: I<sup>2</sup>C 从机中断屏蔽 ( I2CSIMR ) , 偏移量 0x80C

该寄存器决定原始中断是否可以传送到中断控制器。

## I2C 从机中断屏蔽 (I2CSIMR)

I2C 0 基址: 0x4002.0000

I2C 1 基址: 0x4002.1000

I2C 2 基址: 0x4002.2000

I2C 3 基址: 0x4002.3000

I2C 4 基址: 0x400C.0000

I2C 5 基址: 0x400C.1000

偏移量 0x80C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													STOPIM	STARTIM	DATAIM
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
2	STOPIM	R/W	0	停止条件中断屏蔽  值 描述 1 当 I2CSRIS 寄存器中的 STOPRIS 位被置位时, 停止条件中断会被发送到中断控制器。 0 STOPRIS 中断被抑制, 不会发送到中断控制器。
1	STARTIM	R/W	0	开始条件中断屏蔽  值 描述 1 当 I2CSRIS 寄存器中的 STARTRIS 位被置位时, 开始条件中断会被发送到中断控制器。 0 STARTRIS 中断被抑制, 不会发送到中断控制器。
0	DATAIM	R/W	0	数据中断屏蔽  值 描述 1 当 I2CSRIS 寄存器中的 DATARIS 位被置位时, 数据接收或者数据请求中断会被发送到中断控制器。 0 DATARIS 中断被抑制, 不会发送到中断控制器。

寄存器 16: I<sup>2</sup>C 从机原始中断状态寄存器 ( I2CSRIS ) , 偏移量 0x810

该寄存器决定中断是否被挂起。

## I2C 从机原始中断状态寄存器 (I2CSRIS)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x810  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													STOPRIS	STARTRIS	DATARIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
2	STOPRIS	RO	0	停止条件原始中断状态  值 描述 1 有一个停止条件中断已挂起。 0 无中断。  将 I2CSICR 寄存器中的 STOPIC 位置位, 该位就会被清零。
1	STARTRIS	RO	0	开始条件原始中断状态  值 描述 1 有一个开始条件中断已挂起。 0 无中断。  将 I2CSICR 寄存器中的 STARTIC 位写 1, 该位就会被清零。
0	DATARIS	RO	0	数据原始中断状态  值 描述 1 有一个数据接收或者数据请求中断已挂起。 0 无中断。  在 I2CSICR 寄存器中的 DATAIC 位写入 1, 该位就会被清零。

寄存器 17: I<sup>2</sup>C 从机屏蔽中断状态寄存器 ( I2CSMIS ) , 偏移量 0x814

该寄存器决定中断是否可以发出信号。

## I2C 从机屏蔽中断状态寄存器 (I2CSMIS)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x814  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													STOPMIS	STARTMIS	DATAMIS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
2	STOPMIS	RO	0	<p>停止条件屏蔽中断状态</p> <p>值 描述</p> <p>1 有一个未屏蔽的停止条件中断已挂起。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>将 I2CSICR 寄存器中的 STOPIC 位置位, 该位就会被清零。</p>
1	STARTMIS	RO	0	<p>开始条件屏蔽中断状态</p> <p>值 描述</p> <p>1 有一个未屏蔽的开始条件中断已挂起。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>向 I2CSICR 寄存器中的 STARTIC 位写 1, 该位就会被清零。</p>
0	DATAMIS	RO	0	<p>数据屏蔽中断状态</p> <p>值 描述</p> <p>1 有一个未屏蔽的数据接收或者数据请求中断已挂起。</p> <p>0 未产生中断或中断被屏蔽。</p> <p>在 I2CSICR 寄存器中的 DATAIC 位写入 1, 该位就会被清零。</p>

寄存器 18: I<sup>2</sup>C 从机中断清除寄存器 (I2CSICR), 偏移量 0x818

该寄存器可清除原始中断。读取该寄存器将返回无意义的数。

## I2C 从机中断清除寄存器 (I2CSICR)

I2C 0 基址: 0x4002.0000

I2C 1 基址: 0x4002.1000

I2C 2 基址: 0x4002.2000

I2C 3 基址: 0x4002.3000

I2C 4 基址: 0x400C.0000

I2C 5 基址: 0x400C.1000

偏移量 0x818

类型 WO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													STOPIC	STARTIC	DATAIC	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
2	STOPIC	WO	0	停止条件中断清除 在该位写入 1 即可将 I2CSRIS 寄存器中的 STOPRIS 位和 I2CSMIS 寄存器中的 STOPMIS 位清零。 读取该寄存器将返回无意义的数。
1	STARTIC	WO	0	开始条件中断清除 在该位写入 1 即可将 I2CSRIS 寄存器中的 STARTRIS 位和 I2CSMIS 寄存器中的 STARTMIS 位清零。 读取该寄存器将返回无意义的数。
0	DATAIC	WO	0	数据中断清除 在该位写入 1 即可将 I2CSRIS 寄存器中的 STOPRIS 位和 I2CSMIS 寄存器中的 STOPMIS 位清零。 读取该寄存器将返回无意义的数。

## 寄存器 19: I<sup>2</sup>C 从机自身地址寄存器 2 ( I2CSOAR2 ) , 偏移量 0x81C

该寄存器包含 7 位地址位, 用以识别 I<sup>2</sup>C 总线上 I<sup>2</sup>C 设备的复用地址。

### I2C 从机自身地址寄存器 2 (I2CSOAR2)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x81C  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								OAR2EN	OAR2						
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	OAR2EN	R/W	0	I <sup>2</sup> C 从机自身地址 2 启用  值 描述 1 启用 OAR2 域中的复用地址功能。 0 复用地址禁用。
6:0	OAR2	R/W	0x00	I <sup>2</sup> C 从机自身地址 2 该域指定复用 OAR2 地址。

## 寄存器 20: I<sup>2</sup>C 从机应答控制寄存器 ( I2CSACKCTL ) , 偏移量 0x820

该寄存器能够让 I<sup>2</sup>C 从机否定应答无效数据和指令, 或应答有效数据和指令。最后一个数据位传输完成以后, I<sup>2</sup>C 时钟被拉低, 直到该寄存器写入数据。

### I2C 从机应答控制寄存器 (I2CSACKCTL)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0x820  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														ACKOVAL	ACKOEN
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
1	ACKOVAL	R/W	0	I <sup>2</sup> C 从机应答覆盖值  值 描述 1 发送否定应答, 表示数据或者指令无效。 0 发送应答, 表示数据或者指令有效。
0	ACKOEN	R/W	0	I <sup>2</sup> C 从机应答取代启用  值 描述 1 设备会根据 ACKOVAL 位写入的值发送应答或否定应答。 0 没有响应。

## 16.8 寄存器描述 ( I<sup>2</sup>C 状态和控制寄存器 )

本章的剩余部分按照地址偏移量由小到大的顺序依次详细介绍各 I<sup>2</sup>C 状态和控制寄存器。



寄存器 21: I<sup>2</sup>C 外设属性寄存器 (I2CPP) , 偏移量 0xFC0I2CPP 寄存器提供关于 I<sup>2</sup>C 模块的属性信息。

## I2C 外设属性寄存器 (I2CPP)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0xFC0  
 类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															HS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	HS	RO	0x1	可以进行高速传输

## 值 描述

- 1 该接口可以进行高速操作。
- 0 该接口可以进行标准、快速或超快操作。

寄存器 22: I<sup>2</sup>C 外设配置寄存器 (I2CPC) , 偏移量 0xFC4I2CPC 寄存器提供关于 I<sup>2</sup>C 模块的配置信息。

## I2C 外设配置寄存器 (I2CPC)

I2C 0 基址: 0x4002.0000  
 I2C 1 基址: 0x4002.1000  
 I2C 2 基址: 0x4002.2000  
 I2C 3 基址: 0x4002.3000  
 I2C 4 基址: 0x400C.0000  
 I2C 5 基址: 0x400C.1000  
 偏移量 0xFC4  
 类型 RO, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															HS
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
0	HS	RO	1	可以进行高速传输

## 值 描述

- 1 该接口设置为进行高速操作。注: 该编码只能在 I2CPP HS 位已置位时使用。否则, 该编码不可用。
- 0 该接口设置为进行标准、快速或超快操作。

## 17 控制器局域网 (CAN) 模块

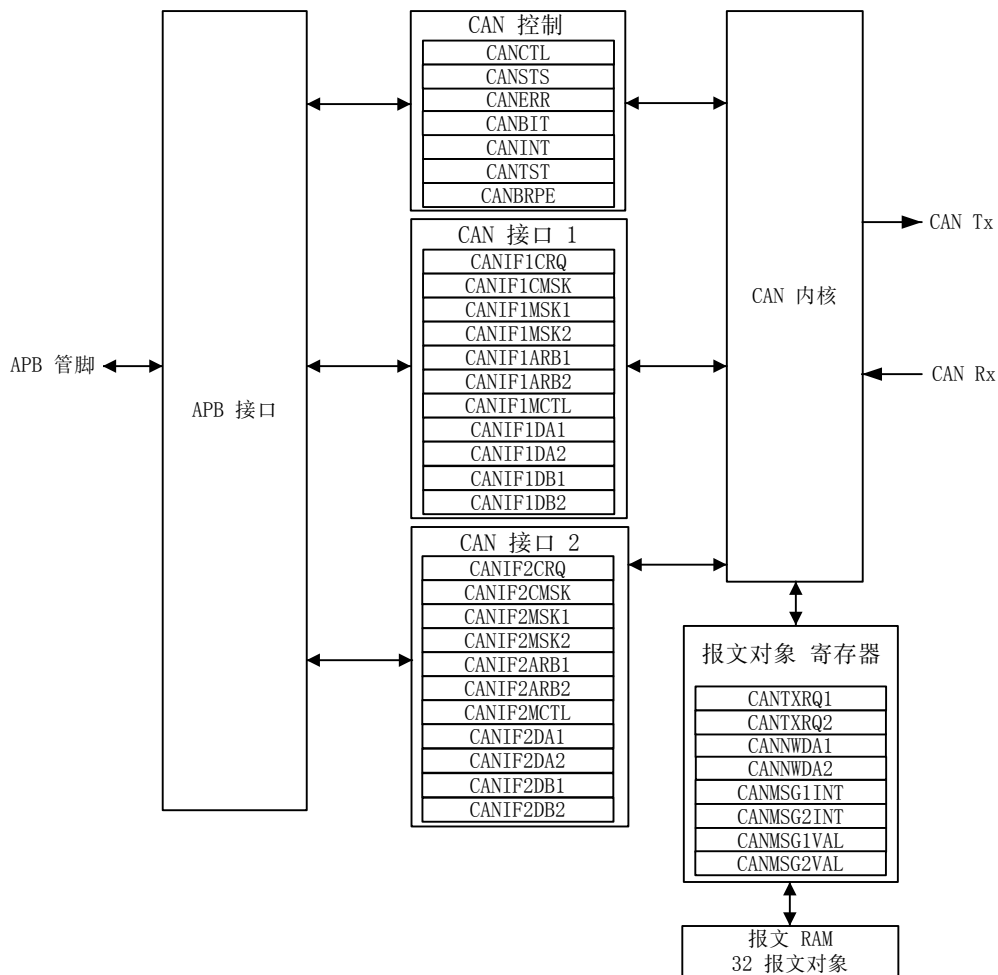
控制器局域网 ( Controller Area Network , 简称为CAN ) 是一种用于连接电子控制设备 ( Electronic Control Unit , 简称为ECU ) 的多主共享型串行总线标准。CAN 总线针对抗电磁干扰进行了专门设计, 适用于具有较强电磁干扰的环境, 不但可以使用与诸如 RS-485 的差分平衡传输线, 也可以使用更加可靠的双绞线。CAN 总线最初是针对汽车应用而研发的, 不过时至今日已经广泛应用于各种嵌入式控制领域 ( 例如工业方面和医疗方面 )。网络接线长度小于 40 米时, CAN 总线的位速率最高可达 1 Mbps。位速率越低, 则有效通信距离越远 ( 例如, 位速率为 125 kbps 时, 通信距离可达 500 米 )。

该 Stellaris® LM4F232H5QD 微控制器包含两个 CAN 模块, 具有以下特性:

- 支持 CAN 总线协议 2.0 A/B
- 位速率最高可达 1 Mbps
- 32 个报文对象, 每个报文对象都具有独立的标识符掩码
- 可屏蔽中断
- 支持禁用自动重新发送 ( Disable Automatic Retransmission , 简称为 DAR ) 模式, 因此可用于时间触发 CAN ( Time Triggered CAN , 简称为 TTCAN ) 应用
- 自测试操作具有可编程的回送模式
- 可编程的FIFO模式, 能存储多个报文对象
- 提供 CANnTX、CANnRX 管脚, 可无缝连接片外 CAN 收发器。

## 17.1 结构图

图 17-1. CAN 控制器结构图



## 17.2 信号描述

下表列出了 CAN 控制器的外部信号及其功能。CAN 控制器信号是某些 GPIO 信号的复用功能，因此在复位时，它们将重置为默认的 GPIO 信号。下表中“复用管脚/赋值”一栏列出了 CAN 信号的 GPIO 管脚的各种可能布局。应将 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 ( 618页 ) 中的 AFSEL 位置位，以便选择 CAN 控制器功能。必须将括号中的数字写入 GPIO 端口控制 (GPIOCTL) 寄存器 ( 636页 ) 的 PMCn 域中，以便把 CAN 信号分配给指定 GPIO 端口管脚。有关如何配置 GPIO 的更多信息，请参考“通用输入/输出端口 (GPIO)”在 597页。

表 17-1. 控制器局域网 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
CAN0Rx	62	PF0 (3)	I	TTL	CAN模块0接收信号。
	81	PN0 (1)			
	136	PB4 (8)			
	139	PE4 (8)			

表 17-1. 控制器局域网 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
CAN0Tx	65 80 135 140	PF3 (3) PN1 (1) PB5 (8) PE5 (8)	O	TTL	CAN模块0发送信号。
CAN1Rx	37 133	PA0 (8) PE6 (8)	I	TTL	CAN模块1接收信号。
CAN1Tx	38 134	PA1 (8) PE7 (8)	O	TTL	CAN模块1发送信号。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 17.3 功能描述

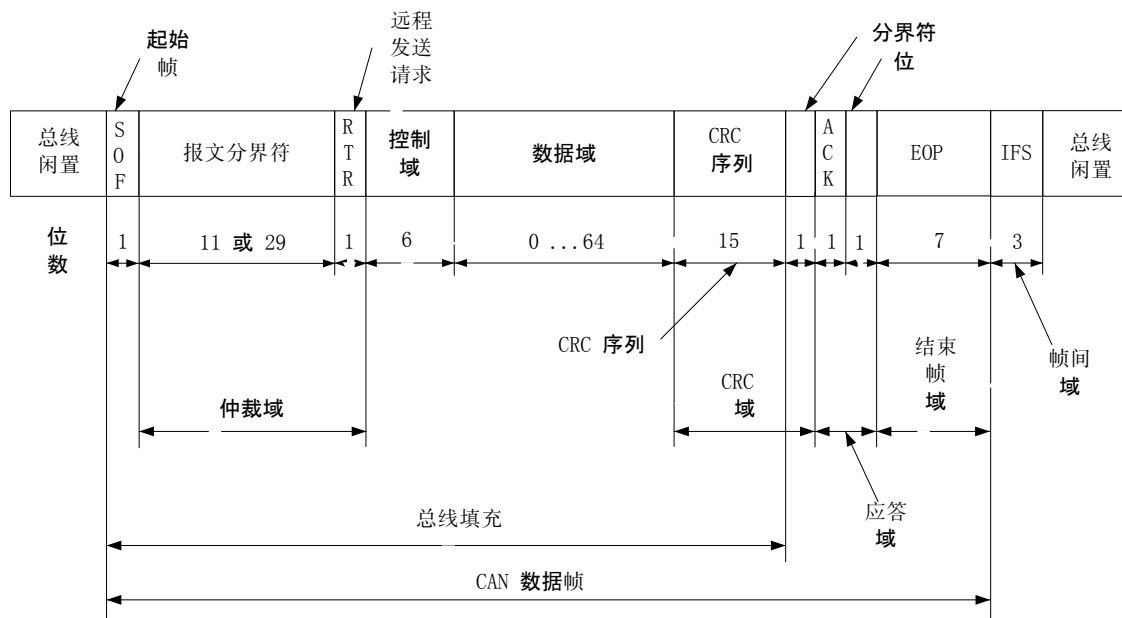
该 Stellaris 控制器支持 CAN 2.0 版协议 (A/B)。支持传输的报文类型包括带有 11 位标识符 (标准) 或 29 位标识符 (扩展) 的数据帧、远程帧、错误帧以及过载帧。通过编程, 传输速率最高可达到 1 Mbps。

CAN 模块由 3 个主要部分组成:

- CAN 协议控制器和报文处理器
- 报文存储器
- CAN 寄存器接口

数据帧中包含要发送的数据; 而远程帧不包含任何数据, 用于请求传输特定报文对象。CAN 数据帧/远程帧的结构如图 17-2 所示。

图 17-2. CAN 数据帧/远程帧



协议控制器通过 CAN 总线传输和接收串行数据，并将数据传递给报文处理器。根据报文对象存储器中当前的滤波设置和标识符，报文处理器会将此信息加载到相应的报文对象。报文处理器还负责根据 CAN 总线上的事件产生中断。

报文对象存储器由一组 32 个完全相同的存储块构成，可保存每个报文对象的当前配置、状态和实际数据。用户可通过 CAN 报文对象寄存器接口之一访问这些存储块。

由于不能通过 Stellaris 存储器映射直接访问报文存储器，所以 Stellaris CAN 控制器将通过和报文对象进行通信的两组 CAN 接口寄存器提供接口，以与报文存储器通信。由于不能直接访问报文存储器，所以必须使用这两组接口寄存器来读取或写入所有报文对象。当多个 CAN 控制器报文对象同时包含亟待处理的新信息时，用户可通过两组报文对象接口并行访问它们。通常情况下，我们用一组接口专司发送，另一组接口专司接收。

### 17.3.1 初始化

要使用 CAN 控制器，必须先通过 RCGC0 寄存器（请参考 442 页）启用外设时钟。此外，还必须通过 RCGC2 寄存器（请参考 448 页）启用相关 GPIO 模块的时钟。欲了解需要启用哪些 GPIO 端口，请参考表 23-4 在 1271 页。将相应管脚的 GPIOAFSEL 位置位（请参考 618 页）。配置 PMCN 寄存器的 GPIOCTL 域，以将 CAN 信号分配给相应的管脚。参考 636 页和表 23-5 在 1280 页。

要开始软件初始化，请（通过软件或以硬件复位的方式）将 CAN 控制 (CANCTL) 寄存器的 INIT 位置位，或进入脱离总线状态（当发送器的错误计数值超过 255 时即进入此状态）。将 INIT 位置位后，CAN 总线上的报文传输都将终止，而且 CANnTX 信号将保持为高电平。进入初始化状态并不会改变 CAN 控制器、报文对象或错误计数器的配置。不过，某些配置寄存器仅可在初始化状态下访问。

要初始化 CAN 控制器，请将 CAN 位时序 (CANBIT) 寄存器置位，并配置每个报文对象。如果无需使用某个报文对象，请将 CAN IFn 仲裁 2 (CANIFnARB2) 寄存器的 MSGVAL 位清零，即可使其标记为无效。否则，请务必将整个报文对象初始化，因为报文对象的某些域可能包含无效的信息，而这些信息将会导致无法预料的结果。必须将 CANCTL 寄存器的 INIT 位和 CCE 位都置位，这样才能访问 CANBIT 寄存器以及 CAN 波特率预分频器扩展 (CANBRPE) 寄存器以配置位定时参数。要退出初始化状态，必须将 INIT 位清零。此后，内部流处理器 (Bit Stream Processor, 简称为 BSP) 首先会等待连续出现 11 个隐性位（表明总线处于空闲状态），以与 CAN 总线数据传输同步，然后才会参与总线活动，开始报文传输。报文对象随时都可以初始化，与 CAN 控制器的初始化状态无关。然而，在开始传输报文之前，必须确保为报文对象配置特殊标识符或将其设置为无效。在正常工作时，如果需要修改某个报文对象的配置，应先将 CANIFnARB2 寄存器中的 MSGVAL 位清零，暂时将该报文对象设为无效。待修改好配置后，重新将 MSGVAL 位置位，以将报文对象设为有效。

### 17.3.2 基本操作

CAN 模块提供两组 CAN 接口寄存器 (CANIF1x 和 CANIF2x)，以用于访问报文 RAM 中的报文对象。CAN 控制器负责协调报文 RAM 与寄存器之间的数据传输。这两组寄存器相互独立而又完全相同，可用于实现连续会话。通常情况下，我们用一组接口专司发送数据，另一组接口专司接收数据。

一旦 CAN 模块完成初始化并且 CANCTL 寄存器中的 INIT 位被清零，CAN 模块将自动与 CAN 总线同步，然后开始传输报文。收到的每个报文都需要经过报文处理器的滤波，如果能够通过，将会保存到由 CAN IFn 命令请求 (CANIFnCRQ) 寄存器中的 MNUM 位所指定的报文对象中。整个报文（包括所有仲裁位、数据长度码以及 8 个数据字节）都将保存在报文对象中。如果使用标识符屏蔽 (CAN IFn 屏蔽 1 和 CAN IFn 屏蔽 2 (CANIFnMSKn) 寄存器中的 MSK 位)，那么报文对象中被屏蔽为“无关”的仲裁位可能会被覆盖。

CPU 随时可以通过 CAN 接口寄存器读写任一报文。当出现并发访问的情况时，由报文处理器负责保障数据的一致性。

报文对象的发送受管理 CAN 硬件的软件所控制。报文对象既可以用于单次数据传输，也可以作为永久报文对象，以实现周期性的响应。永久报文对象的所有仲裁部分及控制部分都已预先设置好，用户只需刷新各数据字节。开始发送时，CAN 发送请求 n (CANTXRQn) 寄存器的 TXRQST 位以及

CAN 新数据 n (CANNWDAn) 寄存器的 NEWDAT 位将置位。假如多个要发送的报文都被分配给同一报文对象 (当报文对象不够用时)，必须在请求发送此报文之前配置好整个报文对象。

CAN 控制器允许同时请求发送任意数量的报文对象；当多个报文对象同时等待发送时，发送顺序是按照内部优先级定义的，即按照报文对象的编号 (MNUM) 排序，其中 1 是最高优先级，32 是最低优先级。报文可以随时刷新或设置为无效，即使其发送请求仍处于挂起状态。当报文的发送请求已挂起并且尚未开始时，刷新报文内容将丢弃旧内容。按照报文对象的配置，当收到标识符相同的远程帧时，系统将自动请求发送报文。

当收到匹配的远程帧时，系统将自动开始发送。要启用该模式，请将 CAN IFn 报文控制 (CANIFnMCTL) 寄存器的 RMTEN 位置位。当收到匹配的远程帧时，TXRQST 位将被置位，并且报文对象将自动发送其数据，或产生中断以表示有远程帧请求。远程帧可以是严格限定的某个报文标识符，也可以是报文对象所定义的一个数值范围。CAN 屏蔽寄存器 (CANIFnMSKn) 用于配置哪些帧组将被确定为远程帧请求。通过 CANIFnMCTL 寄存器的 UMASK 位启用 CANIFnMSKn 寄存器的 MSK 位，以确定哪些帧被识别为远程帧请求。如果准备以 29 位扩展标识符触发远程帧请求，则应将 CANIFnMSK2 寄存器的 MXTD 位置位。

### 17.3.3 报文对象的发送

假如 CAN 模块的内部发送移位寄存器已经准备好加载，并且 CAN 接口寄存器与报文 RAM 之间未发生数据传输，那么已挂起发送请求并且优先级最高的有效报文对象将会被报文处理器加载到发送移位寄存器中，并开始发送。该报文对象在 CANNWDAn 寄存器中的 NEWDAT 位将清零。当发送成功后，如果自开始发送以后该数据对象没有写入新的数据，则 CANTXRQn 寄存器的对应 TXRQST 位将清零。如果 CAN 控制器配置为每成功发送一个报文对象即产生一次中断 (CAN IFn 报文控制 (CANIFnMCTL) 寄存器的 TXIE 位置位)，那么 CANIFnMCTL 寄存器的 INTPND 位会在每次成功发送一个报文对象后被置位。如果 CAN 模块输掉仲裁或在发送期间产生错误，该报文将在 CAN 总线空闲时立即重新发送。如果与此同时有更高优先级的报文请求发送，那么将按照优先级的顺序依次发送报文。

### 17.3.4 待发送报文对象的配置

可按照以下步骤配置要发送的报文对象：

- 在 CAN IFn 命令屏蔽 (CANIFnCMASK) 寄存器中：
  - 将 WRNRD 位置位，以指定对 CANIFnCMASK 寄存器进行写入操作；通过 MASK 位指定是否将报文对象的 IDMASK、DIR 和 MXTD 发送给 CAN IFn 寄存器；
  - 通过 ARB 位指定是否将报文对象的 ID、DIR、XTD 和 MSGVAL 发送给接口寄存器；
  - 通过 CONTROL 位指定是否将控制位发送给接口寄存器；
  - 通过 CLRINTPND 位指定是否将 CANIFnMCTL 寄存器的 INTPND 位清零；
  - 通过 NEWDAT 位指定是否将 CANNWDAn 寄存器的 NEWDAT 位清零；
  - 通过 DATAA 和 DATAB 位指定要发送哪些数据位；
- 通过 CANIFnMSK1 寄存器的 MSK[15:0] 位指定将 29 位或 11 位报文标识符中的哪些位用于验收滤波。请注意，此寄存器的 MSK[15:0] 位将用于 29 位标识符的 [15:0] 位，而不会用于 11 位标识符。若此寄存器写入 0x00，则所有报文均可通过验收滤波。此外还要注意，为了将这些位用于验收滤波，必须将 CANIFnMCTL 寄存器的 UMASK 位置位，以启用这些位。
- 通过 CANIFnMSK1 寄存器的 MSK[12:0] 位指定将 29 位或 11 位报文标识符中的哪些位用于验收滤波。请注意，MSK[12:0] 位将用于 29 位报文标识符的 [28:16] 位；而 MSK[12:2] 位将用于 11 位报文标识符的 [10:0] 位。通过 MXTD 和 MDIR 位指定是否将 XTD 位和 DIR 位也用于验收

滤波。若此寄存器写入 0x00，则所有报文均可通过验收滤波。此外还要注意，为了将这些位用于验收滤波，必须将 CANIFnMCTL 寄存器的 UMASK 位置位，以启用这些位。

4. 对于 29 位的标识符而言，配置 CANIFnARB1 寄存器的 ID[15:0] 以用于报文标识符的 [15:0] 位，而配置 CANIFnARB2 寄存器的 ID[12:0] 以用于报文标识符的 [28:16] 位。将 XTD 位置位即表示采用扩展标识符；将 DIR 位置位即表示发送；将 MSGVAL 位置位即表示该报文对象有效。
5. 对于 11 位标识符，不考虑 CANIFnARB1 寄存器并配置 CANIFnARB2 寄存器的 ID[12:2] 以用于报文标识符的 [10:0] 位。将 XTD 清零即表示采用标准标识符；将 DIR 位置位即表示发送；将 MSGVAL 位置位即表示该报文对象有效。
6. 在 CANIFnMCTL 寄存器中：
  - 可选设置：将 UMASK 位置位即可为验收滤波启用掩码（CANIFnMSK1 和 CANIFnMSK2 寄存器中指定的 MSK、MXTD 和 MDIR）；
  - 可选设置：将 TXIE 位置位，以在成功发送一帧后，让 INTPND 位自动置位；
  - 可选设置：将 RMTEN 位置位，以在收到匹配的远程帧后，让 TXRQST 位自动置位，从而允许自动传输；
  - 将单个报文对象的 EOB 位置位；
  - 配置 DLC[3:0] 域以指定数据帧的长度。配置时应注意不要将 NEWDAT、MSGLST、INTPND 和 TXRQST 位置位。
7. 将待发送数据加载到 CAN IFn 数据(CANIFnDA1, CANIFnDA2, CANIFnDB1, CANIFnDB2) 寄存器中。CAN 数据帧的数据字节 0 将保存在 CANIFnDA1 寄存器的 DATA[7:0] 位中。
8. 在 CAN IFn 指令请求 (CANIFnCRQ) 寄存器的 MNUM 域填写要发送的报文对象编号。
9. 当所有配置都成功完成后，将 CANIFnMCTL 寄存器的 TXRQST 位置位。将此位置位后，报文对象将在总线空闲时按照优先级开始发送。请注意，如果 CANIFnMCTL 寄存器的 RMTEN 位置位，则在接收到匹配的远程帧后也会自动开始发送报文。

### 17.3.5 待发送报文对象的刷新

CPU 随时可以通过 CAN 接口寄存器刷新待发送报文对象的数据字节，而且在刷新之前并不一定需将 CANIFnARB2 寄存器的 MSGVAL 位或者 CANIFnMCTL 寄存器的 TXRQST 位清零。

在 CANIFnDAn/CANIFnDBn 寄存器的内容传递给报文对象之前，即使只需刷新其中的几个数据字节，也必须保证 4 个字节全都有效。通过 CPU 写入新数据字节时，应将所有 4 个字节写入 CANIFnDAn/CANIFnDBn 寄存器中，或者先将报文对象传输到 CANIFnDAn/CANIFnDBn 寄存器，然后再写入数据字节。

要想只刷新报文对象中的数据，应将 CANIFnMSKn 寄存器的 WRNRD、DATAA 和 DATAB 位置位，接着将刷新数据写入 CANIFnDA1、CANIFnDA2、CANIFnDB1 和 CANIFnDB2 寄存器，然后将报文对象编号写入 CAN IFn 命令请求 (CANIFnCRQ) 寄存器的 MNUM 域。要想尽快开始发送新数据，应将 CANIFnMSKn 寄存器的 TXRQST 位置位。

刷新数据时，如果报文对象已开始发送，那么在发送完成后，CANIFnMCTL 寄存器的 TXRQST 位可能被清零。为了避免出现这种情况，应同时将 CANIFnMCTL 寄存器的 NEWDAT 和 TXRQST 位置位。只要这两个位同时置位，则一旦开始新发送，NEWDAT 位将立即清零。



### 17.3.6 已接收报文对象的验收

当已接收报文的仲裁域和控制域 (CANIFnARB2 寄存器的 ID 位域和 XTD 位、CANIFnMCTL 寄存器的 RMTEN 位和 DLC[3:0] 位域) 已经全部移入 CAN 控制器后, 报文处理器将开始扫描报文 RAM 以搜索与之匹配的有效报文对象。要扫描报文 RAM 以搜索匹配报文对象, 控制器将使用通过 CANIFnMSKn 寄存器屏蔽位设置的验收滤波功能, 并通过 CANIFnMCTL 寄存器的 UMASK 位启用滤波功能。控制器将从报文对象 1 开始逐个将有效报文对象与收到的报文进行比对, 以在报文 RAM 中找到匹配的报文对象。当找到匹配的报文对象后, 扫描过程就此结束, 随后报文处理器将根据收到的是数据帧还是远程帧予以分别处理。

### 17.3.7 接收数据帧

报文处理器将来自 CAN 控制器接收移位寄存器的报文存储到报文 RAM 中匹配的报文对象中。而数据字节、所有仲裁位和 DLC 位都存储到相应的报文对象中。在此情况下, 即使采用了仲裁掩码, 数据字节也将与标识符相关联的。CANIFnMCTL 寄存器的 NEWDAT 位置位时, 即表示已经收到新数据。CPU 在读取报文对象之后应将此位清零, 告诉控制器已经接收此报文, 且缓冲区可以开始接收新报文。当 NEWDAT 位已经置位时, 假如 CAN 控制器又接收到一条新报文, 那么 CANIFnMCTL 寄存器的 MSGLST 位将自动置位, 表示前一条数据丢失。假如系统需要在成功接收完一帧后产生中断, 则应将 CANIFnMCTL 寄存器的 RXIE 位置位。在这种情况下, 当某个报文对象成功接收一帧后, 该寄存器的 INTPND 位将自动置位, 使得 CANINT 寄存器指向该报文对象。请注意, 该报文对象的 TXRQST 位应当清零, 以防止发送远程帧。

### 17.3.8 接收远程帧

远程帧中不包含数据, 而只通知应当发送哪一报文对象。当收到远程帧时, 应考虑匹配报文对象的 3 种不同配置:

表 17-2. 报文对象的配置

CANIFnMCTL 寄存器的配置	描述
<ul style="list-style-type: none"> <li>■ DIR = 1 (方向为发送); 通过 CANIFnARB2 寄存器设置</li> <li>■ RMTEN = 1 (在收到帧时, 将 CANIFnMCTL 寄存器的 TXRQST 位置位, 以启用发送)</li> <li>■ UMASK = 1 或 0</li> </ul>	在接收到匹配的远程帧时, 将该报文对象的 TXRQST 位置位。其他报文对象保持不变。控制器将尽快开始自动发送该报文对象的数据。
<ul style="list-style-type: none"> <li>■ DIR = 1 (方向为发送); 通过 CANIFnARB2 寄存器设置</li> <li>■ RMTEN = 0 (在收到帧时, 不改变 CANIFnMCTL 寄存器的 TXRQST 位的设置)</li> <li>■ UMASK = 0 (忽略 CANIFnMSKn 寄存器中的屏蔽设置)</li> </ul>	在接收到匹配的远程帧时, 该报文对象的 TXRQST 位保持不变; 而远程帧被忽略。该远程帧将被禁用, 数据不会被传输, 且看不到任何能表明该远程帧曾经存在过的迹象。
<ul style="list-style-type: none"> <li>■ DIR = 1 (方向为发送); 通过 CANIFnARB2 寄存器设置</li> <li>■ RMTEN = 0 (在收到帧时, 不改变 CANIFnMCTL 寄存器的 TXRQST 位的设置)</li> <li>■ UMASK = 1 (将掩码 [CANIFnMSKn 寄存器的 MSK、MXTD 和 MDIR] 用于验收滤波)</li> </ul>	在接收到匹配的远程帧时, 将该报文对象的 TXRQST 位清零。移位寄存器的仲裁域和控制域 (ID + XTD + RMTEN + DLC) 将存储到报文 RAM 中的报文对象中, 并且该报文对象的 NEWDAT 位置位。报文对象的数据域保持不变; 像接收到的数据帧那样处理远程帧。当另一 CAN 设备的 Stellaris 控制器并未包含可用数据时, 如果它要发出远程数据请求, 这种模式即非常有用。软件必须填充数据并人工响应此帧。

### 17.3.9 接收/发送优先级

报文对象的接收/发送优先级由报文编号决定。报文对象 1 的优先级最高，报文对象 32 的优先级最低。假如当前挂起的发送请求不止一个，那么将优先发送编号最小的那个报文对象。请注意，这里所说的优先级与 CAN 总线上通过报文标识符强制实现的总线优先级毫无关系。举例来说，假如报文对象 1 和报文对象 2 同时有待发送的有效报文，那么报文对象 1 将总是优先发送，不管其报文标识符的总线优先级如何。

### 17.3.10 接收报文对象的配置

可按照以下步骤配置接收报文对象：

1. 按照“待发送报文对象的配置”在 983 页中所描述的方法对 CAN IFn 命令屏蔽 (CANIFnCMASK) 寄存器进行配置，但 WRNRD 位要被置位，表明是对报文 RAM 的写入操作。
2. 按照“待发送报文对象的配置”在 983 页中描述的方法对 CANIFnMSK1 和 CANIFnMSK2 寄存器进行配置，以设置将哪些位用作验收滤波。注意，要将这些位用于验收滤波，必须将 CANIFnMCTL 寄存器的 UMASK 位置位，以启用这些位。
3. 通过 CANIFnMSK1 寄存器的 MSK[12:0] 位指定将 29 位或 11 位报文标识符中的哪些位用于验收滤波。请注意，MSK[12:0] 位将用于 29 位报文标识符的 [28:16] 位；而 MSK[12:2] 位将用于 11 位报文标识符的 [10:0] 位。通过 MXTD 和 MDIR 位指定是否将 XTD 位和 DIR 位也用于验收滤波。若此寄存器写入 0x00，则所有报文均可通过验收滤波。此外还要注意，为了将这些位用于验收滤波，必须将 CANIFnMCTL 寄存器的 UMASK 位置位，以启用这些位。
4. 按照“待发送报文对象的配置”在 983 页中描述的方法配置 CANIFnARB1 和 CANIFnARB2 寄存器的 XTD 和 ID 位以接收报文标识符，并将 MSGVAL 位置位以表示报文有效，同时将 DIR 位清零以表示接收。
5. 在 CANIFnMCTL 寄存器中：
  - 可选设置：将 UMASK 位置位即可为验收滤波启用掩码 (CANIFnMSK1 和 CANIFnMSK2 寄存器中指定的 MSK、MXTD 和 MDIR)；
  - 可选设置：将 RXIE 位置位，以在成功接收一帧后，让 INTPND 位自动置位；
  - 将 RMTEN 位清零，使得 TXRQST 位保持不变；
  - 将单个报文对象的 EOB 位置位；
  - 配置 DLC[3:0] 位域，以指定数据帧的长度；
 配置时应注意不要将 NEWDAT、MSGLST、INTPND 和 TXRQST 位置位。
6. 在 CAN IFn 指令请求 (CANIFnCRQ) 寄存器的 MNUM 域中写入要接收报文对象的编号。当 CAN 总线上有匹配的帧时，将立即开始接收报文对象。

当报文处理器向数据对象保存一个数据帧时，会将收到的数据长度码 (Data Length Code，简称为 DLC) 以及 8 个数据字节写入 CANIFnDA1、CANIFnDA2、CANIFnDB1 和 CANIFnDB2 寄存器。CAN 数据帧的数据字节 0 将保存在 CANIFnDA1 寄存器的 DATA[7:0] 位中。假如数据长度码小于 8，那么报文对象的剩余数据字节将以随机值予以覆盖。

通过 CAN 屏蔽寄存器 (CANIFnMSKn)，可以允许某个报文对象接收某些数据帧组。CAN 屏蔽寄存器 (CANIFnMSKn) 用于配置哪些帧组将被报文对象接收。通过 CANIFnMCTL 寄存器的 UMASK 位启用 CANIFnMSKn 寄存器的 MSK 位，以过滤哪些帧将被接收。如果报文对象仅使用 29 位扩展标识符，则应将 CANIFnMSK2 寄存器中的 MXTD 位置位。

### 17.3.11 已接收报文对象的处理

因为报文处理器状态机保证了数据的一致性，所以 CPU 可以随时通过 CAN 接口寄存器读取接收的报文。

一般来说，CPU 应先向 CANIFnCMSK 寄存器写入 0x007F，随后向 CANIFnCRQ 寄存器写入报文对象的编号。这个操作组合将使已接收的整条报文从报文 RAM 移入报文缓冲寄存器中

(CANIFnMSKn、CANIFnARBn 和 CANIFnMCTL 寄存器)。此外，报文 RAM 中的 NEWDAT 位和 INTPND 位也会清零，以确认该报文对象已经被读取，并清除该报文对象所产生的挂起的中断。

假如报文对象使用掩码进行验收滤波，那么可以通过 CANIFnARBn 寄存器查询已接收报文在屏蔽操作之前的完整 ID。

CANIFnMCTL 寄存器的 NEWDAT 位能够指示出自上一次读取此报文对象之后是否又收到了新的报文。CANIFnMCTL 寄存器的 MSGLST 位能够指示出自上一次读取此报文对象之后是否又收到了新的报文。MSGLST 位不会自动清零，因此软件必须在每次读取其状态后手动将其清零。

通过远程帧，CPU 可以向 CAN 总线的其它节点请求新的数据。将已接收报文对象的 TXRQST 位置位时，系统会以其报文标识符开始发送远程帧。此远程帧将使其它 CAN 节点开始发送匹配的数据帧。假如在远程帧发送完成之前收到了匹配的数据帧，那么 TXRQST 位将自动复位。当 CAN 总线上的其它设备提前发送数据帧时，该操作可以避免数据丢失。

#### 17.3.11.1 FIFO 缓冲区的配置

除 CANIFnMCTL 寄存器的 EOB 位以外，属于 FIFO 缓冲区的接收报文对象的配置方式与单个接收报文对象的配置方法完全相同（请参考“接收报文对象的配置”在 986 页）。要将 2 个或更多报文对象连续进入同一个 FIFO 缓冲区中，请为这些报文对象的标识符及掩码（如果有的话）设置相同的值。由于报文对象已经有固定的优先级，因此编号最小的报文对象必然是 FIFO 缓冲区中的首个报文对象。除最后一个报文对象外，FIFO 中其它报文对象的 EOB 位必须清零。最后一个报文对象的 EOB 位应置位，以表明它是缓冲区中的最后一个单元。

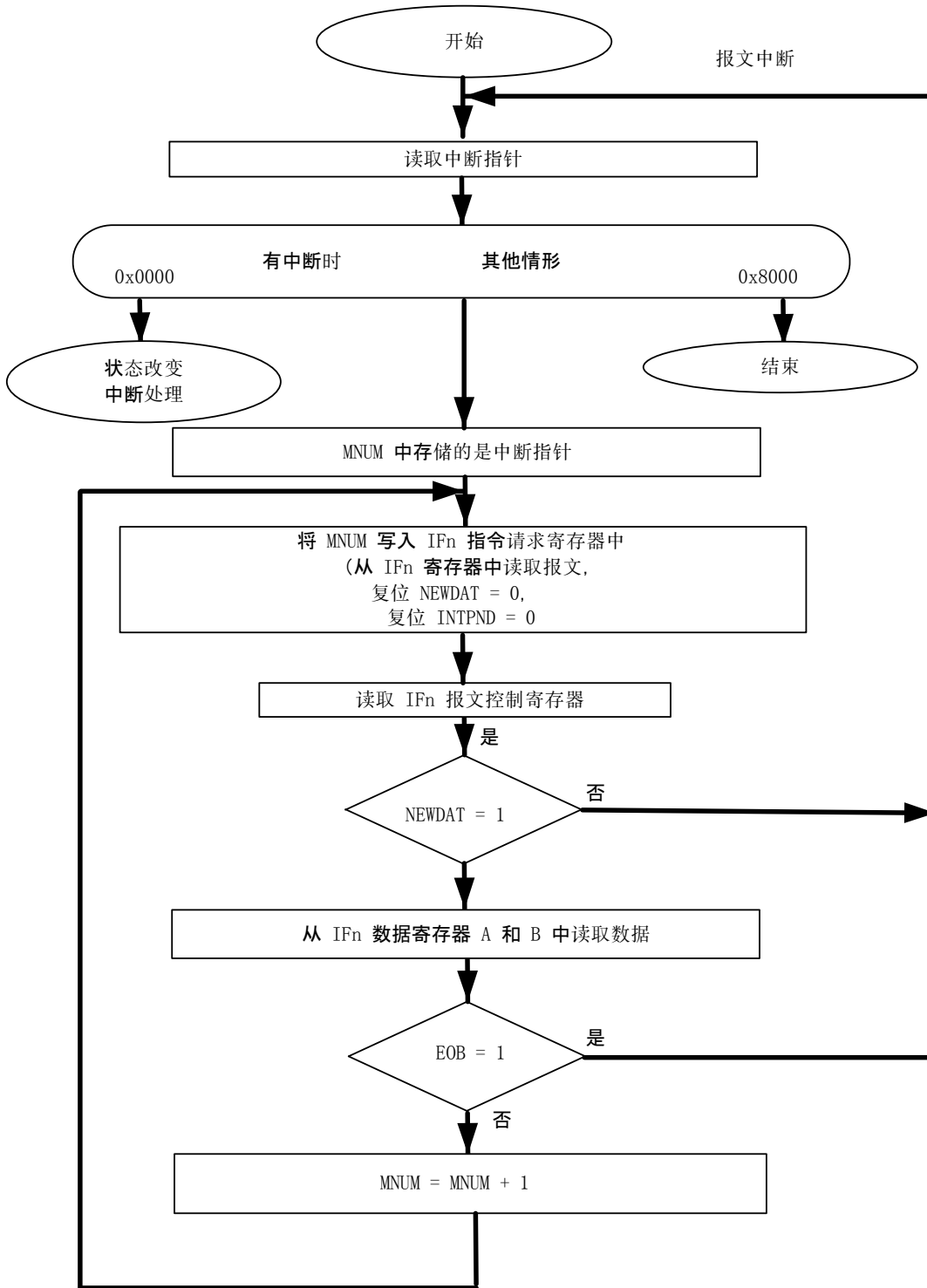
#### 17.3.11.2 采用 FIFO 缓冲区接收报文

当接收到若干个标识符与 FIFO 缓冲区匹配的报文时，这些报文将从编号最低的报文对象开始依次保存。当报文保存到 FIFO 缓冲区中的某个报文对象中时，该报文对象的 CANIFnMCTL 寄存器的 NEWDAT 位便会置位。若在 EOB 清零时将 NEWDAT 位置位，则该报文对象将锁定，报文处理器将无法对其写入，直到 CPU 将 NEWDAT 位清零。匹配的报文将依次保存入 FIFO 缓冲区中，直至达到 FIFO 缓冲区的最后一个报文对象。以后收到的匹配报文都将写入 FIFO 缓冲区的最后一个报文对象中，因此，该最后一个报文对象的内容可能会被反复覆盖，直到将 NEWDAT 位清零以释放前面的所有报文对象。

#### 17.3.11.3 FIFO 缓冲区的读取

当 CPU 向 CANIFnCRQ 寄存器写入某个报文对象编号以传输其内容时，应将 CANIFnCMSK 寄存器的 TXRQST 和 CLRINTPND 位置位，这样，CANIFnMCTL 寄存器的 NEWDAT 和 INTPEND 位将在读取后自动清零。CANIFnMCTL 寄存器中的这两个位将始终反映报文对象的当前状态，直到用户手动将其清零。要确保 FIFO 缓冲区正常工作，CPU 应当按照编号由小到大的顺序依次读取各报文对象的内容。当从 FIFO 缓冲区中读取数据时，用户必须明白：此时接收到的新报文将写入 CANIFnMCTL 寄存器的 NEWDAT 位已被清零的编号最小的报文对象中。因此，无法确定新接收的报文在 FIFO 缓冲区中的顺序。图 17-3 在 988 页描述了 CPU 可如何处理连续进入 FIFO 缓冲区中的一组报文对象。

图 17-3. FIFO 缓冲区中的报文对象



### 17.3.12 中断的处理

如果多个中断被挂起，CAN 中断 (CANINT) 寄存器将指向优先级最高的挂起中断，而不考虑它们的时间顺序。状态中断具有最高优先级。就报文中断而言，编号最小的报文对象中断具有最高优先级。

报文中断既可以通过将相应报文对象 CANIFnMCTL 寄存器的 INTPND 位清零予以清除，也可以通过读取 CAN 状态 (CANSTS) 寄存器予以清除。状态中断只能通过读取 CANSTS 寄存器予以清除。

CANINT 寄存器中的中断标识符 INTID 可表明中断原因。当没有任何中断挂起时，该寄存器的读取值为 0x0000。只要 INTID 域的值不是 0，即表明有中断挂起。如果将 CANCTL 寄存器的 IE 位置位，则中断控制器的中断行将激活。中断线将保持处于活动状态，直到 INTID 域为 0（表示所有中断源都已清除，即中断条件已经复位），或者 IE 被清零（即禁用 CAN 控制器的中断）。

CANINT 寄存器的 INTID 域将指向挂起的具有最高优先级的报文中断。CANCTL 寄存器的 SIE 位用于控制 CANSTS 寄存器中 RXOK、TXOK 和 LEC 位的改变是否会导致中断。CANCTL 寄存器的 EIE 位用于控制 CANSTS 寄存器中 BOFF 和 EWARN 位的改变是否会导致中断。CANCTL 寄存器中的 IE 位用于控制 CAN 控制器产生的中断是否会导致中断控制器中断。即使 CANCTL 寄存器的 IE 位清零，CANINT 寄存器也会自动更新，只不过中断不会指示给 CPU。

当 CANINT 寄存器的值为 0x8000 时，表示 CAN 模块刷新了 CANSTS 寄存器（但并不一定改变了其内容）并挂起了中断，因此产生的中断是错误中断或状态中断。对 CANSTS 寄存器执行写入操作即可将其中的 RXOK、TXOK 和 LEC 位清零；但如果想要清除状态中断源，则唯一的办法是读取 CANSTS 寄存器。

在进行中断处理时可通过以下两种方法确定中断源：一种方法是通过读取 CANINT 寄存器的 INTID 位，确定目前挂起的最高优先级中断；另一种方法是通过读取 CAN 报文中断挂起 (CANMSGnINT) 寄存器，查看所有拥有挂起中断的报文对象。

负责读取作为中断源的报文的 中断服务程序在读取报文时还可能同时将 CANIFnCMSK 寄存器的 CLRINTPND 位置位，以将报文对象的 INTPND 位清零。一旦 INTPND 位被清零，CANINT 寄存器的内容将变成下一个挂起中断的报文对象编号。

### 17.3.13 测试模式

CAN 模块提供测试模式，方便进行各种诊断。将 CANCTL 寄存器的 TEST 位置位即可进入测试模式。进入测试模式后，可通过 CAN 测试 (CANTST) 寄存器的 TX[1:0]、LBACK、SILENT 和 BASIC 位将 CAN 控制器置于各种不同的诊断模式下。CANTST 寄存器的 RX 位用于监控 CANnRX 信号。当 TEST 位清零时，CANTST 寄存器的所有设置将被禁用。

#### 17.3.13.1 安静模式

安静模式可用于分析 CAN 总线上的通信，它不会传输显性位（确认位、错误帧等等），因此不会对通信造成任何影响。将 CANTST 寄存器的 SILENT 位置位即可让 CAN 控制器进入安静模式。在安静模式下，CAN 控制器能够接收有效的数据帧和远程帧，但本身只能在 CAN 总线上发送隐性位，而无法发送任何报文。即使 CAN 控制器必须发送显性位（应答位、过载位或活动错误标志位），也仅能在控制器内部传输。也就是说，即便 CAN 总线处于隐性状态，CAN 控制器也能够监控显性位。

#### 17.3.13.2 回送模式

回送模式用于实现自检功能。在回送模式下，CAN 控制器在内部将 CANnTX 信号传输到 CANnRX 信号，并将自身发送的报文像接收到的报文一样存储到报文缓冲区中（假如这些报文能够通过验收滤波）。将 CANTST 寄存器的 LBACK 位置位即可进入回送模式。为了彻底隔绝外部事件的影响，在回送模式下，CAN 控制器将会忽略应答错误（即在数据帧/远程帧的应答时间内采样到的隐性位）。CANnRX 信号的实际值会被 CAN 控制器忽略。不过，仍然能够在 CANnTX 信号上监控发送的报文。

#### 17.3.13.3 回送 + 安静模式

如果混合使用回送模式与安静模式，CAN 控制器既能实现在线自检，也不会影响与 CANnTX 和 CANnRX 信号连接的 CAN 系统的运行。在这种模式下，CANnRX 信号并不连入 CAN 控制器，并且 CANnTX 信号始终保持隐性位状态。将 CANTST 寄存器的 LBACK 和 SILENT 位同时置位即可进入这种模式。

### 17.3.13.4 基本模式

在基本模式下，CAN 控制器在运行过程中将不使用报文 RAM。在基本模式下，CANIF1 寄存器将用作发送缓冲区。将 CANIF1CRQ 寄存器的 BUSY 位置位即会请求发送 IF1 寄存器的内容。当 BUSY 位置位时，CANIF1 寄存器即被锁定。此时，BUSY 位可用于指示挂起的发送。当 CAN 总线空闲时，CANIF1 寄存器就会把内容载入 CAN 控制器的移位寄存器中，并开始发送。发送结束后 BUSY 位将自动清零，并且 CANIF1 寄存器也将解除锁定状态。当 CANIF1 寄存器被锁定时，将 CANIF1CRQ 寄存器的 BUSY 位清零即可随时终止挂起的发送。当报文因为输掉仲裁或者发送出错而等待重发时，只要 CPU 将 BUSY 位清零，就能禁止自动重发此报文。

CANIF2 寄存器用作接收缓冲区。当接收一条报文后，移位寄存器中的内容将不经过任何验收滤波而直接保存到 CANIF2 寄存器中。此外，在报文传输期间，可以监视移位寄存器的实际内容。每当 CANIF2CRQ 寄存器的 BUSY 位置位（开始读取报文对象）时，移位寄存器的内容都将保存到 CANIF2 寄存器中。

在基本模式下，所有与报文对象相关的控制位、状态位以及 CANIFnCMSK 寄存器的控制位都不会被赋值。CANIFnCRQ 寄存器的报文编号也不会被赋值。在 CANIF2MCTL 寄存器中，NEWDAT 和 MSGLST 位将保持其原有功能，DLC[3:0] 域将显示收到的 DLC，其余控制位均被清零。

将 CANTST 寄存器的 BASIC 位置位，即可启用基本模式。

### 17.3.13.5 发送控制

软件可以通过如下 4 种方式直接取代 CANnTX 信号的控制：

- CANnTX 由 CAN 控制器控制；
- 采样点通过 CANnTX 信号驱动输出，用以监控位时序；
- CANnTX 始终驱动输出低值；
- CANnTX 始终驱动输出高值；

结合可读的 CAN 接收管脚 CANnRX，最后两项功能可用于校验 CAN 总线物理层是否工作正常。

对 CANTST 寄存器的 TX[1:0] 域进行设置即可启用发送控制功能。CANnTX 信号的三种测试功能可连接所有 CAN 协议功能。TX[1:0] 位必须清零（当选择 CAN 报文传输、回送模式、安静模式或基本模式时）。

### 17.3.14 位时序配置错误的注意事项

如果在配置 CAN 的位时序参数时出现微小错误，即使其并未立即导致总线故障，也可能会严重降低 CAN 总线的性能。在很多情况下，CAN 总线本身的位同步功能将会修正 CAN 位时序参数配置的不当，将其影响降低到只会偶尔产生错误帧的程度。但在仲裁时，如果两个或两个以上的 CAN 节点同时试图发送帧，则偏移的采样点可能导致其中一个发送器进入错误认可状态。这种偶发错误非常难于分析，需要对 CAN 节点内的位同步原理以及 CAN 节点与 CAN 总线的互动过程有详尽的了解才能予以断定。

### 17.3.15 位时间与位速率

CAN 总线支持的位速率范围是 1kbps 到 1000kbps。CAN 网络中的每个成员都有自己的时钟发生器。即使各 CAN 节点所采用的振荡器频率并不相同，也能够通过单独设置各个 CAN 节点的位时间参数，最终实现位速率的统一。

温度/电压波动以及元件本身老化将导致频率总会存在微小偏差，因此这些振荡器不可能始终保持稳定。不过，只要偏差保持在指定的振荡器容差范围内，CAN 节点都可通过定期与总线上的比特流重新同步的方式为各种位速率进行纠偏。

根据 CAN 规范，位时间分成 4 个时间段（请参阅图 17-4 在 991 页）：同步段、传播时间段、相位缓冲段 1 和相位缓冲段 2。每个段由具体、可编程的时间份额数量组成（请参阅表 17-3 在 991 页）。时间份额是位时间配置的基本单位，时间份额的长度 ( $t_q$ ) 由 CAN 控制器的输入时钟 ( $f_{sys}$ ) 以及波特率预分频器（Baud Rate Prescaler，简称为 BRP）共同确定：

$$t_q = BRP / f_{sys}$$

其中  $f_{sys}$  就是通过 RCC 或 RCC2 寄存器（请参考 236 页或 243 页）设置的系统时钟频率。

同步段 (Sync) 是将产生 CAN 总线电平跳变沿的位时间部分；若跳变沿位于 Sync 之外，则将跳变沿与 Sync 的间隔称为该跳变沿的相位误差。

传播时间段 (Prop) 用于补偿 CAN 网络中的物理延迟时间。

相位缓冲段 1 (Phase1) 和相位缓冲段 2 (Phase2) 共同确定采样点在位时间中的位置。

（重）同步跳转宽度 (SJW) 决定重新同步会将采样点移动多远，移动距离的上限由用于补偿跳变沿相位误差的相位缓冲段确定。

通过不同的位时间配置可以得到指定的位速率，但是要让 CAN 网络正常工作，必须考虑物理延迟时间和振荡器的容限。

图 17-4. CAN 位时间

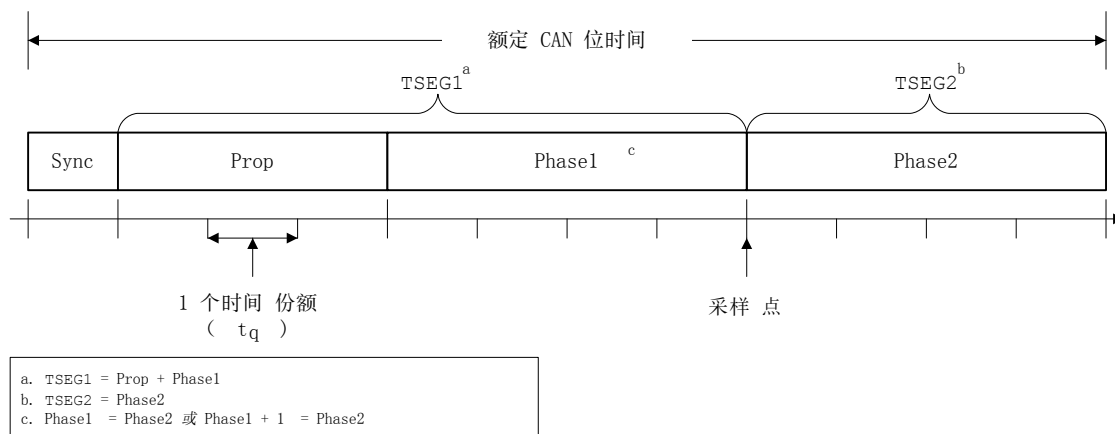


表 17-3. CAN 协议范围<sup>a</sup>

参数	有效范围	备注
BRP	[1..64]	定义时间份额 $t_q$ 的长度。通过 CANBRPE 寄存器可将此范围扩展到 1024。
Sync	1 $t_q$	固定长度，将总线输入与系统时钟进行同步
Prop	[1..8] $t_q$	补偿物理延时
Phase1	[1..8] $t_q$	根据同步的情况可能会临时延长
Phase2	[1..8] $t_q$	根据同步的情况可能会临时缩短
SJW	[1..4] $t_q$	不得超过任一相位缓冲段的长度

a. 该表描述了 CAN 协议要求的最小可编程范围。

位定时的配置是通过 CANBIT 寄存器中两个字节设置的。在 CANBIT 寄存器中，TSEG2、TSEG1、SJW 和 BRP 这 4 个位域的值必须填写为其实际取值减去 1；因此虽然上表中定义的有效范围是 [1..n]，实际对寄存器编程值的有效范围是 [0..n-1]。举例来说，同步跳转宽度 SJW（有效范围 [1..4]）在 SJW 位域中只需两个位即可表示。表 17-4 列出了 CANBIT 寄存器位域值与实际参数的关系。

表 17-4. CANBIT 寄存器值

CANBIT 寄存器域	设置
TSEG2	Phase2 - 1
TSEG1	Prop + Phase1 - 1
SJW	SJW - 1
BRP	BRP

因此，位时间的长度为（编程值）：

$$[TSEG1 + TSEG2 + 3] \times t_q$$

或（函数值）：

$$[Sync + Prop + Phase1 + Phase2] \times t_q$$

CANBIT 寄存器中的数值将作为 CAN 协议控制器的配置输入。波特率预分频器（由 BRP 域配置）定义了每个时间份额（位时间的基本时间单元）的长度；位时序逻辑（由 TSEG1、TSEG2 和 SJW 配置）定义了位时间中包含多少个时间份额。

位时间的处理、采样点的位置计算以及偶发的同步动作均由 CAN 控制器控制，并且在每个时间份额都要进行判定。

CAN 控制器负责将报文转译为帧以及将帧转译为报文。此外，控制器还要负责以下任务：产生/丢弃帧中的固定格式位；插入/抽取填充位；计算/校验 CRC 校验码；执行错误管理以及确定采用何种类型的同步。位值将在采样点接收或发送。信息处理时间 (IPT) 是采样点后控制器做好向 CAN 总线发送下一个位所需的时间。IPT 期间控制器需要进行的工作包括：获取下一个数据位、处理 CRC 位、判断是否需要位填充以及根据需要产生错误标志或进入空闲状态。

IPT 根据应用程序的不同而不同，但不会长于  $2 t_q$ ；CAN 的 IPT 是  $0 t_q$ 。IPT 长度是 Phase2 可编程长度的下限值。进行同步时，缩短后的 Phase2 可能会小于 IPT，但这并不影响总线时序。

### 17.3.16 位时序参数的计算

在计算位时序参数时，通常应首先确定所需的位速率或位时间。位时间（即位速率的倒数）必须是系统时钟周期的整数倍。

位时间由 4 到 25 个时间份额组成。通过将各个时间段进行组合，就能得到所需的位时间，即重复以下步骤。

首先要确定位时间的 Prop 段。Prop 段的长度取决于系统中测得的延迟时间。为保证 CAN 总线系统具有可扩展性，应明确定义最长总线长度和最大节点延时。Prop 的最终时间应转换为时间份额（进位得到最接近的整数值，即  $t_q$  的整数倍）。

Sync 段的长度是 1 个  $t_q$ （固定），从而两个相位缓冲段的长度为（位时间 - Prop - 1） $t_q$ 。假如剩余的  $t_q$  数是偶数值，则将平均分配给两个相位缓冲段，即 Phase2 = Phase1；假如是奇数值，则 Phase2 = Phase1 + 1。

还必须考虑 Phase2 的最小额定长度。Phase2 不能比 CAN 控制器的信息处理时间短。其长度范围为  $[0..2] t_q$ ，具体视实际的执行情况而定。

同步跳转宽度的长度可设置为以下 3 个值中的最小值：4、Phase1 和 Phase2。

结果配置所需的振荡器容限范围通过以下公式计算得到：

$$(1 - df) \times f_{nom} \leq f_{osc} \leq (1 + df) \times f_{nom}$$



其中：

- $df$  = 振荡器频率的最大容限
- $fosc$  = 实际的振荡器频率
- $fnom$  = 额定的振荡器频率

最大频率容限必须考虑以下公式：

$$df \leq \frac{(Phase\_seg1, Phase\_seg2) \min}{2 \times (13 \times tbit - Phase\_Seg2)}$$

$$df \max = 2 \times df \times fnom$$

其中：

- Phase1 和 Phase2 取自表 17-3 在 991 页
- $tbit$  = 位时间
- $dfmax$  = 两个振荡器之间的最大差异

如果可以包含一个以上的配置，那么应选择允许最高振荡器容限范围的配置。

如果多个系统时钟频率不同的 CAN 节点需要达到相同的位速率，需要对其各自进行不同的配置。在计算传输时间时，应在整个 CAN 网络内选取距离最远的节点计算最大延迟时间。

CAN 系统的振荡器容限范围受容限范围最低的节点限制。

这个计算表明：为了找到一种协议兼容的 CAN 位时序配置，必须缩短总线长度或、降低位速率，或增强振荡器频率的稳定性。

### 17.3.16.1 高波特率下的位时序参数计算

在此示例中，假设 CAN 模块输入时钟为 25 MHz，CAN 总线位速率为 1 Mbps。

$$\text{bit time} = 1 \mu\text{s} = n * t_q = 5 * t_q$$

$$t_q = 200 \text{ ns}$$

$$t_q = (\text{Baud rate Prescaler}) / \text{CAN Clock}$$

$$\text{Baud rate Prescaler} = t_q * \text{CAN Clock}$$

$$\text{Baud rate Prescaler} = 200\text{E}-9 * 25\text{E}6 = 5$$

$$t_{\text{Sync}} = 1 * t_q = 200 \text{ ns} \quad \backslash\backslash \text{fixed at 1 time quanta}$$

delay of bus driver 50 ns

delay of receiver circuit 30 ns

delay of bus line (40m) 220 ns

$$t_{\text{Prop}} 400 \text{ ns} = 2 * t_q \quad \backslash\backslash 400 \text{ is next integer multiple of } t_q$$

$$\text{bit time} = t_{\text{Sync}} + t_{\text{TSeg1}} + t_{\text{TSeg2}} = 5 * t_q$$

$$\text{bit time} = t_{\text{Sync}} + t_{\text{Prop}} + t_{\text{Phase 1}} + t_{\text{Phase 2}}$$

```

tPhase 1 + tPhase2 = bit time - tSync - tProp
tPhase 1 + tPhase2 = (5 * tq) - (1 * tq) - (2 * tq)
tPhase 1 + tPhase2 = 2 * tq
tPhase1 = 1 * tq
tPhase2 = 1 * tq                \\tPhase2 = tPhase1

tTSeg1 = tProp + tPhase1
tTSeg1 = (2 * tq) + (1 * tq)
tTSeg1 = 3 * tq

tTSeg2 = tPhase2
tTSeg2 = (Information Processing Time + 1) * tq
tTSeg2 = 1 * tq                \\Assumes IPT=0

tSJW = 1 * tq                \\Least of 4, Phase1 and Phase2
    
```

在上面的例子中，CANBIT 寄存器各位域的值为：

TSEG2	= TSeg2 -1 = 1-1 = 0
TSEG1	= TSeg1 -1 = 3-1 = 2
SJW	= SJW -1 = 1-1 = 0
BRP	= 波特率预分频器 - 1 = 5-1 = 4

因此最终写入 CANBIT 寄存器的值是 0x0204。

### 17.3.16.2 低波特率下的位时序参数计算

在此计算中，假定 CAN 时钟的频率 50 MHz，CAN 总线位速率为 100kbps。

```

bit time = 10 μs = n * tq = 10 * tq
tq = 1 μs
tq = (Baud rate Prescaler)/CAN Clock
Baud rate Prescaler = tq * CAN Clock
Baud rate Prescaler = 1E-6 * 50E6 = 50

tSync = 1 * tq = 1 μs                \\fixed at 1 time quanta

delay of bus driver 200 ns
delay of receiver circuit 80 ns
delay of bus line (40m) 220 ns
tProp 1 μs = 1 * tq                \\1 μs is next integer multiple of tq

bit time = tSync + tTSeg1 + tTSeg2 = 10 * tq
bit time = tSync + tProp + tPhase 1 + tPhase2
    
```

```

tPhase 1 + tPhase2 = bit time - tSync - tProp
tPhase 1 + tPhase2 = (10 * tq) - (1 * tq) - (1 * tq)
tPhase 1 + tPhase2 = 8 * tq
tPhase1 = 4 * tq
tPhase2 = 4 * tq                                \\tPhase1 = tPhase2

tTSeg1 = tProp + tPhase1
tTSeg1 = (1 * tq) + (4 * tq)
tTSeg1 = 5 * tq
tTSeg2 = tPhase2
tTSeg2 = (Information Processing Time + 4) * tq
tTSeg2 = 4 * tq                                \\Assumes IPT=0

tSJW = 4 * tq                                \\Least of 4, Phase1, and Phase2

```

TSEG2	= TSeg2 - 1 = 4-1 = 3
TSEG1	= TSeg1 - 1 = 5-1 = 4
SJW	= SJW - 1 = 4-1 = 3
BRP	= 波特率预分频器 - 1 = 50-1 =49

因此最终写入 CANBIT 寄存器的值是 0x34F1。

## 17.4 寄存器映射

表 17-5 在 995页 列出了各寄存器。给出的地址均相对以下 CAN 基地址：

- CAN0 : 0x4004.0000
- CAN1 : 0x4004.1000

请注意，在设置寄存器之前，应先启用 CAN 控制器时钟（请参考 314页）。启用 CAN 模块时钟后，必须要延迟 3 个系统时钟才可以访问 CAN 模块寄存器。

表 17-5. CAN 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	CANCTL	R/W	0x0000.0001	CAN 控制寄存器	998
0x004	CANSTS	R/W	0x0000.0000	CAN 状态寄存器	1000
0x008	CANERR	RO	0x0000.0000	CAN 错误计数寄存器	1003
0x00C	CANBIT	R/W	0x0000.2301	CAN 位时序寄存器	1004
0x010	CANINT	RO	0x0000.0000	CAN 中断寄存器	1005
0x014	CANTST	R/W	0x0000.0000	CAN 测试寄存器	1006

表 17-5. CAN 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x018	CANBRPE	R/W	0x0000.0000	CAN 波特率预分频器扩展寄存器	1008
0x020	CANIF1CRQ	R/W	0x0000.0001	CAN IF1 指令请求寄存器	1009
0x024	CANIF1CMSK	R/W	0x0000.0000	CAN IF1 指令屏蔽寄存器	1010
0x028	CANIF1MSK1	R/W	0x0000.FFFF	CAN IF1 屏蔽寄存器 1	1013
0x02C	CANIF1MSK2	R/W	0x0000.FFFF	CAN IF1 屏蔽寄存器 2	1014
0x030	CANIF1ARB1	R/W	0x0000.0000	CAN IF1 仲裁寄存器 1	1015
0x034	CANIF1ARB2	R/W	0x0000.0000	CAN IF1 仲裁寄存器 2	1016
0x038	CANIF1MCTL	R/W	0x0000.0000	CAN IF1 报文控制寄存器	1018
0x03C	CANIF1DA1	R/W	0x0000.0000	CAN IF1 数据寄存器 A1	1021
0x040	CANIF1DA2	R/W	0x0000.0000	CAN IF1 数据寄存器 A2	1021
0x044	CANIF1DB1	R/W	0x0000.0000	CAN IF1 数据寄存器 B1	1021
0x048	CANIF1DB2	R/W	0x0000.0000	CAN IF1 数据寄存器 B2	1021
0x080	CANIF2CRQ	R/W	0x0000.0001	CAN IF2 指令请求寄存器	1009
0x084	CANIF2CMSK	R/W	0x0000.0000	CAN IF2 指令屏蔽寄存器	1010
0x088	CANIF2MSK1	R/W	0x0000.FFFF	CAN IF2 屏蔽寄存器 1	1013
0x08C	CANIF2MSK2	R/W	0x0000.FFFF	CAN IF2 屏蔽寄存器 2	1014
0x090	CANIF2ARB1	R/W	0x0000.0000	CAN IF2 仲裁寄存器 1	1015
0x094	CANIF2ARB2	R/W	0x0000.0000	CAN IF2 仲裁寄存器 2	1016
0x098	CANIF2MCTL	R/W	0x0000.0000	CAN IF2 报文控制寄存器	1018
0x09C	CANIF2DA1	R/W	0x0000.0000	CAN IF2 数据寄存器 A1	1021
0x0A0	CANIF2DA2	R/W	0x0000.0000	CAN IF2 数据寄存器 A2	1021
0x0A4	CANIF2DB1	R/W	0x0000.0000	CAN IF2 数据寄存器 B1	1021
0x0A8	CANIF2DB2	R/W	0x0000.0000	CAN IF2 数据寄存器 B2	1021
0x100	CANTXRQ1	RO	0x0000.0000	CAN 传输请求寄存器 1	1022
0x104	CANTXRQ2	RO	0x0000.0000	CAN 传输请求寄存器 2	1022
0x120	CANNWDA1	RO	0x0000.0000	CAN 新数据寄存器 1	1023
0x124	CANNWDA2	RO	0x0000.0000	CAN 新数据寄存器 2	1023
0x140	CANMSG1INT	RO	0x0000.0000	CAN 报文 1 中断挂起寄存器	1024
0x144	CANMSG2INT	RO	0x0000.0000	CAN 报文 2 中断挂起寄存器	1024
0x160	CANMSG1VAL	RO	0x0000.0000	CAN 报文 1 有效寄存器	1025
0x164	CANMSG2VAL	RO	0x0000.0000	CAN 报文 2 有效寄存器	1025

## 17.5 CAN 寄存器描述

本章的剩余部分按照地址偏移量由小到大的顺序依次详细介绍各寄存器。CAN 模块提供两组接口寄存器以用于访问报文 RAM 中的报文对象：CANIF1x 和 CANIF2x。这两组寄存器的功能完全相同，可用于实现连续会话。

## 寄存器 1: CAN 控制寄存器 (CANCTL), 偏移量 0x000

该控制寄存器用于初始化模块以及启用测试模式和中断。

即使将 INIT 位置位或清零, 也无法缩短脱离总线序列 (请参阅 CAN 规范 2.0)。设备一旦进入脱离总线状态, 便会将 INIT 位置位, 并终止所有总线活动。当 CPU 将 INIT 位清零后, 设备需等待总线上出现 129 个空闲态 (129 \* 11 个连续的隐形位), 然后才能恢复正常的总线操作。脱离总线恢复序列结束时, 错误管理计数器将会复位。

在 INIT 清零后的等待期间, 每当监视到 11 个连续的高电平位, CANSTS 寄存器即被写入一个 BITERROR0 码 (LEC 域 = 0x5), 这样, CPU 即可随时检查 CAN 总线是否被拉低或受到持续干扰, 同时还能监视脱离总线恢复序列的进度。

### CAN 控制寄存器 (CANCTL)

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x000  
 类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								TEST	CCE	DAR	保留	EIE	SIE	IE	INIT
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述						
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。						
7	TEST	R/W	0	测试模式启用  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>CAN 控制器工作正常。</td> </tr> <tr> <td>1</td> <td>CAN 控制器处于测试模式。</td> </tr> </table>	值	描述	0	CAN 控制器工作正常。	1	CAN 控制器处于测试模式。
值	描述									
0	CAN 控制器工作正常。									
1	CAN 控制器处于测试模式。									
6	CCE	R/W	0	配置修改启用  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>禁止对 CANBIT 寄存器进行写入操作。</td> </tr> <tr> <td>1</td> <td>若 INIT 位为 1, 则允许对 CANBIT 寄存器进入写入操作。</td> </tr> </table>	值	描述	0	禁止对 CANBIT 寄存器进行写入操作。	1	若 INIT 位为 1, 则允许对 CANBIT 寄存器进入写入操作。
值	描述									
0	禁止对 CANBIT 寄存器进行写入操作。									
1	若 INIT 位为 1, 则允许对 CANBIT 寄存器进入写入操作。									
5	DAR	R/W	0	禁用自动重发功能  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>启用自动重发被打断的报文。</td> </tr> <tr> <td>1</td> <td>禁用自动重发功能。</td> </tr> </table>	值	描述	0	启用自动重发被打断的报文。	1	禁用自动重发功能。
值	描述									
0	启用自动重发被打断的报文。									
1	禁用自动重发功能。									
4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。						

位/域	名称	类型	复位	描述
3	EIE	R/W	0	出错中断启用  值        描述 0        未产生错误状态中断。 1        当 CANSTS 寄存器的 BOFF 或 EWARN 位发生变化时， 产生了一个中断。
2	SIE	R/W	0	状态中断启用  值        描述 0        不产生状态中断。 1        成功发送或接收报文或检测到 CAN 总线错误时，即产生一 个中断。当 CANSTS 寄存器的 TXOK、RXOK 或 LEC 位发 生变化时，将产生一个中断。
1	IE	R/W	0	CAN 中断启用  值        描述 0        中断禁用。 1        中断启用。
0	INIT	R/W	1	初始化  值        描述 0        正常工作。 1        初始化开始。

## 寄存器 2: CAN 状态寄存器 (CANSTS), 偏移量 0x004

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

该状态寄存器中包含用于中断服务的信息, 如脱离总线、错误计数阈值和错误类型。

LEC 域保存的代码能够表明 CAN 总线的上一个错误的类型。当成功完成一个报文传输 (接收或发送) 后, 此域将自动清零。CPU 可以向此域写入未用的错误码 0x7, 以手动将此域设置为无效错误, 从而方便检查它日后是否发生变化。

若 CAN 控制 (CANCTL) 寄存器中的 BOFF 和 EWARN 位被置位, 即会产生错误中断; 若 RXOK、TXOK 和 LEC 位被置位, 即会产生状态中断。EPASS 位的更改或者对 RXOK、TXOK 或 LEC 位执行写入操作都不会产生中断。

在 CAN 中断挂起时, 读取 CAN 状态 (CANSTS) 寄存器会将 CAN 中断启用 (CANINT) 寄存器清零。

### CAN 状态寄存器 (CANSTS)

CAN0 基址: 0x4004.0000  
CAN1 基址: 0x4004.1000  
偏移量 0x004  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								BOFF	EWARN	EPASS	RxOK	TxOK	LEC		
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	BOFF	RO	0	脱离总线状态  值            描述 0            CAN 控制器未处于脱离总线状态。 1            CAN 控制器处于脱离总线状态。
6	EWARN	RO	0	警告状态  值            描述 0            两组错误计数器都低于错误警告上限 (96 次)。 1            至少有一组错误计数器已达到错误警告上限 (96 次)。



位/域	名称	类型	复位	描述
5	EPASS	RO	0	<p>错误认可</p> <p>值      描述</p> <p>0      CAN 模块处于错误激活状态，也就是说，接收或发送错误计数都小于或等于 127。</p> <p>1      CAN 模块处于错误认可状态，也就是说，接收或发送错误计数至少有一个大于 127。</p>
4	RxOK	R/W	0	<p>接收报文成功</p> <p>值      描述</p> <p>0      自从该位上一次清零以来，并未成功接收到新的报文。</p> <p>1      自从该位上一次清零以来，至少成功接收到一条新的报文。此位与验收滤波结果无关。</p> <p>该位必须手动写入 0 予以清零。</p>
3	TxOK	R/W	0	<p>发送报文成功</p> <p>值      描述</p> <p>0      自从该位上一次清零以来，没有报文发送成功。</p> <p>1      自从该位上一次清零以来，至少成功发送了一条新的报文。该操作没有发生错误，并且至少被另外一个节点应答。</p> <p>该位必须手动写入 0 予以清零。</p>

位/域	名称	类型	复位	描述
2:0	LEC	R/W	0x0	<p>上一次的错误代码 即 CAN 总线上次发生的错误类型。</p> <p>值      描述</p> <p>0x0      无错误</p> <p>0x1      填充错误 接收的报文中某个部分包含连续 5 个以上的相同位，而这在 CAN 协议中是不允许的。</p> <p>0x2      格式错误 接收的报文中某个本该具有固定格式的部分格式错误。</p> <p>0x3      应答错误 发送的报文未收到任何节点的应答。</p> <p>0x4      位 1 错误 在发送报文时，CAN 控制器对数据线进行监控以探测是否存在冲突。在发送仲裁域时，数据冲突是仲裁协议的一部分。在发送其它帧域时，数据冲突被视为错误。 第 1 位错误表示设备企图发送高电平（逻辑 1），但是监视到的总线电平为低电平（逻辑 0）。</p> <p>0x5      位 0 错误 第 0 位错误表示设备企图发送低电平（逻辑 0），但是监视到的总线电平为高电平（逻辑 1）。 在脱离总线恢复期间，每次监控到含 11 个高电平位的序列后，该状态就会置位。通过对该状态的检查，软件可以在干扰总线的情况下对脱离总线恢复序列的进程进行监控。</p> <p>0x6      CRC 错误 接收的报文中的 CRC 校验有误，表明收到的计算值与数据的经计算的 CRC 不符。</p> <p>0x7      无事件发生 当 LEC 位显示该值时，表明自上一次对 LEC 域写该值以来，未检测出任何 CAN 总线事件。</p>

### 寄存器 3: CAN 错误计数寄存器 (CANERR) , 偏移量 0x008

该寄存器包含了错误计数器值, 可以用来分析错误原因。

#### CAN 错误计数寄存器 (CANERR)

CAN0 基址: 0x4004.0000

CAN1 基址: 0x4004.1000

偏移量 0x008

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RP	REC						TEC								
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述						
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。						
15	RP	RO	0	接收到的错误认可 <table border="0" style="margin-left: 20px;"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>接收错误计数低于错误认可水平 ( 127 或更低 )。</td> </tr> <tr> <td>1</td> <td>接收错误计数已达到错误认可水平 ( 128 或更高 )。</td> </tr> </table>	值	描述	0	接收错误计数低于错误认可水平 ( 127 或更低 )。	1	接收错误计数已达到错误认可水平 ( 128 或更高 )。
值	描述									
0	接收错误计数低于错误认可水平 ( 127 或更低 )。									
1	接收错误计数已达到错误认可水平 ( 128 或更高 )。									
14:8	REC	RO	0x00	接收错误计数器 此域包含接收错误计数器的状态值 ( 0 到 127 )。						
7:0	TEC	RO	0x00	发送错误计数器 此位包含发送错误计数器的值 ( 0 到 255 )。						

## 寄存器 4: CAN 位时序寄存器 (CANBIT) , 偏移量 0x00C

本寄存器用于定义位宽度和位份额。写入的数值应按照系统时钟频率进行计算。只有将 CANCTL 寄存器的 CCE 位和 INIT 位置位时，此寄存器才允许写入。更多信息请参阅“位时间与位速率”在 990 页。

### CAN 位时序寄存器 (CANBIT)

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x00C  
 类型 R/W, 复位 0x0000.2301

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留	TSEG2			TSEG1				SJW		BRP					
类型	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述
31:15	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
14:12	TSEG2	R/W	0x2	采样点后的时间段 0x00-0x07：硬件在实际应用中对该值的解释要比此处编程的值大 1。因此，例如复位 0x2 表示 Phase2 包含 3 (即 2 + 1) 个位时间份额 (请参阅图 17-4 在 991 页)。位时间份额由 BRP 域定义。
11:8	TSEG1	R/W	0x3	采样点前的时间段 0x00-0x0F：硬件在实际应用中对该值的解释要比此处编程的值大 1。因此，例如复位 0x3 表示 Phase1 包含 4 (即 3 + 1) 个位时间份额 (请参阅图 17-4 在 991 页)。位时间份额由 BRP 域定义。
7:6	SJW	R/W	0x0	(重) 同步跳转宽度 0x00-0x03：硬件在实际应用中对该值的解释要比此处编程的值大 1。在帧起始 (SOF) 过程中，如果 CAN 控制器检测到相位误差 (偏差)，它可以通过改变 SJW 的值来调整 TSEG2 或 TSEG1 的长度因此复位 0 调节了 1 个位时间份额的长度。
5:0	BRP	R/W	0x1	波特率预分频器 该值是通过振荡器频率分频获得的，用于产生位时间份额。位时间由多个这种份额组成。 0x00-0x3F：硬件在实际应用中对该值的解释要比此处编程的值大 1。BRP 定义每个时间份额由多少个 CAN 时钟周期组成，因此复位为 2 (即 1+1) 个位时间份额。 CANBRPE 寄存器可对位时间进行进一步细分。

## 寄存器 5: CAN 中断寄存器 (CANINT) , 偏移量 0x010

该寄存器用于指示中断源。

假如多个中断同时挂起，则 CAN 中断寄存器 (CANINT) 始终指向挂起的最高优先级中断，而不考虑中断产生的顺序。中断在被 CPU 清除之前将保持挂起。假如 INTID 域的值不是 0x0000 (默认值)，并且 CANCTL 寄存器的 IE 位被置位，则该中断将保持激活。在通过读取 CANSTS 寄存器以将 INTID 域清零之前，或在将 CANCTL 寄存器的 IE 位清零之前，中断线将始终保持激活。

**注意：** 在 CAN 中断挂起时，读取 CAN 状态 (CANSTS) 寄存器会将 CAN 中断启用 (CANINT) 寄存器清零。

### CAN 中断寄存器 (CANINT)

CAN0 基址: 0x4004.0000

CAN1 基址: 0x4004.1000

偏移量 0x010

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTID															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	INTID	RO	0x0000	中断标识符 此位域中的数值代表中断源。
			值	描述
			0x0000	无中断挂起
			0x0001-0x0020	导致中断的报文对象编号
			0x0021-0x7FFF	保留
			0x8000	状态中断
			0x8001-0xFFFF	保留

## 寄存器 6: CAN 测试寄存器 (CANTST) , 偏移量 0x014

该寄存器用于自检以及访问外部管脚。只有当 CANCTL 寄存器的 TEST 位置位时, 才允许对本寄存器进行写入操作。用户可以按需组合不同的测试功能, 不过应当注意到: 假如 TX 位的值不是 0, 将可能影响 CAN 的发送功能。

### CAN 测试寄存器 (CANTST)

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x014  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								RX	TX		LBACK	SILENT	BASIC	保留	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述										
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。										
7	RX	RO	0	接收观测  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>CANnRx 管脚为低电平。</td> </tr> <tr> <td>1</td> <td>CANnRx 管脚为高电平。</td> </tr> </table>	值	描述	0	CANnRx 管脚为低电平。	1	CANnRx 管脚为高电平。				
值	描述													
0	CANnRx 管脚为低电平。													
1	CANnRx 管脚为高电平。													
6:5	TX	R/W	0x0	发送控制 取代 CANnTx 管脚的控制。  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0x0</td> <td>CAN 模块控制 CANnTx 是默认操作, 由 CAN 模块控制。</td> </tr> <tr> <td>0x1</td> <td>采样点 采样点在 CANnTx 信号驱动输出。该模式适用于监控位时序。</td> </tr> <tr> <td>0x2</td> <td>驱动输出低电平 CANnTX 始终驱动输出低电平。该模式适用于检查 CAN 总线的线物理层。</td> </tr> <tr> <td>0x3</td> <td>驱动输出高电平 CANnTX 始终驱动输出高电平。该模式适用于检查 CAN 总线的线物理层。</td> </tr> </table>	值	描述	0x0	CAN 模块控制 CANnTx 是默认操作, 由 CAN 模块控制。	0x1	采样点 采样点在 CANnTx 信号驱动输出。该模式适用于监控位时序。	0x2	驱动输出低电平 CANnTX 始终驱动输出低电平。该模式适用于检查 CAN 总线的线物理层。	0x3	驱动输出高电平 CANnTX 始终驱动输出高电平。该模式适用于检查 CAN 总线的线物理层。
值	描述													
0x0	CAN 模块控制 CANnTx 是默认操作, 由 CAN 模块控制。													
0x1	采样点 采样点在 CANnTx 信号驱动输出。该模式适用于监控位时序。													
0x2	驱动输出低电平 CANnTX 始终驱动输出低电平。该模式适用于检查 CAN 总线的线物理层。													
0x3	驱动输出高电平 CANnTX 始终驱动输出高电平。该模式适用于检查 CAN 总线的线物理层。													
4	LBACK	R/W	0	回送模式  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>回送模式禁用。</td> </tr> <tr> <td>1</td> <td>回送模式启用。在回送模式下, 发送器所发出的数据将被路由到接收器。而接收输入将被忽略。</td> </tr> </table>	值	描述	0	回送模式禁用。	1	回送模式启用。在回送模式下, 发送器所发出的数据将被路由到接收器。而接收输入将被忽略。				
值	描述													
0	回送模式禁用。													
1	回送模式启用。在回送模式下, 发送器所发出的数据将被路由到接收器。而接收输入将被忽略。													

位/域	名称	类型	复位	描述
3	SILENT	R/W	0	<p>安静模式</p> <p>值      描述</p> <p>0      安静模式禁用。</p> <p>1      安静模式启用。在安静模式下，CAN 控制器不发送任何数据，而只监控总线。此模式通常也称为总线监控模式。</p>
2	BASIC	R/W	0	<p>基本模式</p> <p>值      描述</p> <p>0      基本模式禁用。</p> <p>1      基本模式启用。在基本模式下，软件应当采用 CANIF1 寄存器作为发送缓冲区，并采用 CANIF2 寄存器作为接收缓冲区。</p>
1:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

### 寄存器 7: CAN 波特率预分频器扩展寄存器 ( CANBRPE ) , 偏移量 0x018

本寄存器用于对通过 CANBIT 寄存器中的 BRP 位设置的位时间实现进一步分频。只有当 CANCTL 寄存器的 CCE 位置位时，才允许对本寄存器进行写入操作。

#### CAN 波特率预分频器扩展寄存器 (CANBRPE)

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x018  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												BRPE			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3:0	BRPE	R/W	0x0	波特率预分频器扩展 0x00-0x0F：最高可将 CANBIT 寄存器中的 BRP 位扩展到 1023。硬件在实际应用中对该位的解释要比 BRPE (MSB) 和 BRP (LSB) 编程的值大 1。



**寄存器 8: CAN IF1 指令请求寄存器 (CANIF1CRQ), 偏移量 0x020****寄存器 9: CAN IF2 指令请求寄存器 (CANIF2CRQ), 偏移量 0x080**

当 CANIF1MCTL 寄存器的 TXRQST 位置位时, 如果对本寄存器的 MNUM 域写入某个报文对象的编号, 即会启动报文传输。进行写入操作时, BUSY 位也会自动置位, 表明在 CAN 接口寄存器与内部报文 RAM 之间正在进行数据传输。等待 3 到 6 个 CAN\_CLK 周期后, 接口寄存器与报文 RAM 之间的传输将结束, 随后 BUSY 位将清零。

**CAN IFn 指令请求寄存器 (CANIFnCRQ)**

CAN0 基址: 0x4004.0000

CAN1 基址: 0x4004.1000

偏移量 0x020

类型 R/W, 复位 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BUSY	保留										MNUM				
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

位/域	名称	类型	复位	描述								
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。								
15	BUSY	RO	0	繁忙标志								
				<table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>当读/写活动结束后, 此位将自动清零</td> </tr> <tr> <td>1</td> <td>当向本寄存器中写入报文对象编号时, 此位即自动置位。</td> </tr> </tbody> </table>	值	描述	0	当读/写活动结束后, 此位将自动清零	1	当向本寄存器中写入报文对象编号时, 此位即自动置位。		
值	描述											
0	当读/写活动结束后, 此位将自动清零											
1	当向本寄存器中写入报文对象编号时, 此位即自动置位。											
14:6	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。								
5:0	MNUM	R/W	0x01	报文编号 在报文 RAM 中的 32 个报文对象中挑出一个进行数据传输。报文对象的编号是 1 到 32。								
				<table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>保留 0 不是有效的报文编号; 它被解释为 0x20 或对象 32。</td> </tr> <tr> <td>0x01-0x20</td> <td>报文编号 代表 1 到 32 号报文对象</td> </tr> <tr> <td>0x21-0x3F</td> <td>保留 不是有效的报文编号; 这些值将被替换, 并且被解释为 0x01-0x1F。</td> </tr> </tbody> </table>	值	描述	0x00	保留 0 不是有效的报文编号; 它被解释为 0x20 或对象 32。	0x01-0x20	报文编号 代表 1 到 32 号报文对象	0x21-0x3F	保留 不是有效的报文编号; 这些值将被替换, 并且被解释为 0x01-0x1F。
值	描述											
0x00	保留 0 不是有效的报文编号; 它被解释为 0x20 或对象 32。											
0x01-0x20	报文编号 代表 1 到 32 号报文对象											
0x21-0x3F	保留 不是有效的报文编号; 这些值将被替换, 并且被解释为 0x01-0x1F。											

**寄存器 10: CAN IF1 指令屏蔽寄存器 ( CANIF1CMSK ) , 偏移量 0x024**

**寄存器 11: CAN IF2 指令屏蔽寄存器 ( CANIF2CMSK ) , 偏移量 0x084**

读取指令屏蔽寄存器即可获得各种功能的当前状态。写入指令屏蔽寄存器即可指定传输方向, 并选择缓冲寄存器作为数据传输的源或目标。

请注意, 若 WRNRD 位清零且 CLRINTPND 和/或 NEWDAT 位置位, 则在读取报文对象缓冲区时, 缓冲区中的中断挂起和/或新数据标志将被清除。

**CAN IFn 指令屏蔽寄存器 ( CANIFnCMSK )**

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x024  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								WRNRD	MASK	ARB	CONTROL	CLRINTPND	NEWDAT / TXRQST	DATAA	DATAB
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述						
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。						
7	WRNRD	R/W	0	写入, 非读取  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>将 CANIFnCRQ 寄存器的 MNUM 域指定的 CAN 报文对象中的数据传送到 CANIFn 寄存器中。</td> </tr> <tr> <td>1</td> <td>将 CANIFn 寄存器中的数据移动到 CAN 指令请求 (CANIFnCRQ) 寄存器中的 MNUM 域所指定的 CAN 报文对象中。</td> </tr> </table> 注意: CLRINTPND 和/或 NEWDAT 位置位时, 可通过读取报文缓冲区 (WRNRD = 0) 来清除缓冲区中的中断挂起和新数据状况。	值	描述	0	将 CANIFnCRQ 寄存器的 MNUM 域指定的 CAN 报文对象中的数据传送到 CANIFn 寄存器中。	1	将 CANIFn 寄存器中的数据移动到 CAN 指令请求 (CANIFnCRQ) 寄存器中的 MNUM 域所指定的 CAN 报文对象中。
值	描述									
0	将 CANIFnCRQ 寄存器的 MNUM 域指定的 CAN 报文对象中的数据传送到 CANIFn 寄存器中。									
1	将 CANIFn 寄存器中的数据移动到 CAN 指令请求 (CANIFnCRQ) 寄存器中的 MNUM 域所指定的 CAN 报文对象中。									
6	MASK	R/W	0	访问掩码位  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>掩码位无变化。</td> </tr> <tr> <td>1</td> <td>将报文对象的 IDMASK + DIR + MXTD 传递给接口寄存器。</td> </tr> </table>	值	描述	0	掩码位无变化。	1	将报文对象的 IDMASK + DIR + MXTD 传递给接口寄存器。
值	描述									
0	掩码位无变化。									
1	将报文对象的 IDMASK + DIR + MXTD 传递给接口寄存器。									
5	ARB	R/W	0	访问仲裁位  <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>仲裁位无变化。</td> </tr> <tr> <td>1</td> <td>将报文对象的 ID + DIR + XTD + MSGVAL 传递给接口寄存器。</td> </tr> </table>	值	描述	0	仲裁位无变化。	1	将报文对象的 ID + DIR + XTD + MSGVAL 传递给接口寄存器。
值	描述									
0	仲裁位无变化。									
1	将报文对象的 ID + DIR + XTD + MSGVAL 传递给接口寄存器。									

位/域	名称	类型	复位	描述
4	CONTROL	R/W	0	配置控制位  值 描述 0 控制位无变化。 1 将 CANIFnMCTL 寄存器的控制位传递给接口寄存器。
3	CLRINTPND	R/W	0	清除中断挂起位 此位的功能取决于 WRNRD 位的配置。  值 描述 0 若将 WRNRD 位清零，则中断挂起状态将从报文缓冲区传递给 CANIFnMCTL 寄存器。 若将 WRNRD 位置位，报文对象中的 INTPND 位将保持不变。 1 若将 WRNRD 位清零，报文缓冲区中的中断挂起状态也会被清除。请注意：传递给 CANIFnMCTL 寄存器的该位的值总是会反映其被清除之前的真实状态。 若将 WRNRD 位置位，报文对象中的 INTPND 位将被清零。
2	NEWDAT / TXRQST	R/W	0	NEWDAT / TXRQST 位 此位的功能取决于 WRNRD 位的配置。  值 描述 0 若 WRNRD 被清零，则新数据状态值将从报文缓冲区传递给 CANIFnMCTL 寄存器。 若将 WRNRD 位置位，将不会请求发送。 1 若将 WRNRD 位清零，报文缓冲区中的新数据状态将被清除。请注意：传递给 CANIFnMCTL 寄存器的该位的值总是会反映其被清除之前的真实状态。 若将 WRNRD 位置位，将会请求发送。请注意：当该位置位时，CANIFnMCTL 寄存器中的 TXRQST 位将被忽略。
1	DATAA	R/W	0	访问数据字节 0 到 3 此位的功能取决于 WRNRD 位的配置。  值 描述 0 数据字节 0-3 不会改变。 1 若将 WRNRD 位清零，则 CANIFnDA1 和 CANIFnDA2 寄存器中的数据字节 0-3 将传递给报文对象。 若将 WRNRD 位置位，则报文对象中的数据字节 0-3 将传递给 CANIFnDA1 和 CANIFnDA2 寄存器。

位/域	名称	类型	复位	描述						
0	DATAB	R/W	0	访问数据字节 4 到 7 此位的功能取决于 WRNRD 位的配置。						
				<table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>数据字节 4-7 不会改变。</td> </tr> <tr> <td>1</td> <td>若将 WRNRD 位清零，则 CANIFnDA1 和 CANIFnDA2 寄存器中的数据字节 4-7 将传递给报文对象； 若将 WRNRD 位置位，则报文对象中的数据字节 4-7 将传递给 CANIFnDA1 和 CANIFnDA2 寄存器。</td> </tr> </tbody> </table>	值	描述	0	数据字节 4-7 不会改变。	1	若将 WRNRD 位清零，则 CANIFnDA1 和 CANIFnDA2 寄存器中的数据字节 4-7 将传递给报文对象； 若将 WRNRD 位置位，则报文对象中的数据字节 4-7 将传递给 CANIFnDA1 和 CANIFnDA2 寄存器。
值	描述									
0	数据字节 4-7 不会改变。									
1	若将 WRNRD 位清零，则 CANIFnDA1 和 CANIFnDA2 寄存器中的数据字节 4-7 将传递给报文对象； 若将 WRNRD 位置位，则报文对象中的数据字节 4-7 将传递给 CANIFnDA1 和 CANIFnDA2 寄存器。									

**寄存器 12: CAN IF1 屏蔽寄存器 1 ( CANIF1MSK1 ) , 偏移量 0x028****寄存器 13: CAN IF2 屏蔽寄存器 1 ( CANIF2MSK1 ) , 偏移量 0x088**

本寄存器中包含的掩码信息将连同数据信息 (CANIFnDAn)、仲裁信息 (CANIFnARn) 和控制信息 (CANIFnMCTL) 一起写入报文 RAM 中的报文对象中。本寄存器中的掩码可结合 CANIFnARn 寄存器的 ID 位实现验收滤波。其他掩码信息在 CANIFnMSK2 寄存器中。

**CAN IFn 屏蔽寄存器 1 ( CANIFnMSK1 )**

CAN0 基址: 0x4004.0000  
CAN1 基址: 0x4004.1000  
偏移量 0x028  
类型 R/W, 复位 0x0000.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSK															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述						
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。						
15:0	MSK	R/W	0xFFFF	标识符掩码 在使用 29 位的标识符时，这些位用作 ID 的 [15:0] 位。CANIFnMSK2 寄存器中的 MSK 域用作 ID 的 [28 : 16] 位。对于 11 位的报文标识符，本位无意义。  <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>报文对象中的对应标识符域 (ID) 不能禁用验收滤波中的匹配。</td> </tr> <tr> <td>1</td> <td>报文对象中的对应标识符域 (ID) 用于验收滤波。</td> </tr> </tbody> </table>	值	描述	0	报文对象中的对应标识符域 (ID) 不能禁用验收滤波中的匹配。	1	报文对象中的对应标识符域 (ID) 用于验收滤波。
值	描述									
0	报文对象中的对应标识符域 (ID) 不能禁用验收滤波中的匹配。									
1	报文对象中的对应标识符域 (ID) 用于验收滤波。									

**寄存器 14: CAN IF1 屏蔽寄存器 2 ( CANIF1MSK2 ) , 偏移量 0x02C**

**寄存器 15: CAN IF2 屏蔽寄存器 2 ( CANIF2MSK2 ) , 偏移量 0x08C**

该寄存器保存了 CANIFnMSK1 寄存器相关的掩码信息。

**CAN IFn 屏蔽寄存器 2 ( CANIFnMSK2 )**

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x02C  
 类型 R/W, 复位 0x0000.FFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MXTD	MDIR	保留													
类型	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述						
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。						
15	MXTD	R/W	1	掩码扩展标识符 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>扩展标识符位 ( CANIFnARB2 寄存器的 XTD 位 ) 不影响验收滤波。</td> </tr> <tr> <td>1</td> <td>扩展标识符位 XTD 用于验收滤波。</td> </tr> </tbody> </table>	值	描述	0	扩展标识符位 ( CANIFnARB2 寄存器的 XTD 位 ) 不影响验收滤波。	1	扩展标识符位 XTD 用于验收滤波。
值	描述									
0	扩展标识符位 ( CANIFnARB2 寄存器的 XTD 位 ) 不影响验收滤波。									
1	扩展标识符位 XTD 用于验收滤波。									
14	MDIR	R/W	1	屏蔽报文方向 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>报文方向位 ( CANIFnARB2 寄存器的 DIR 位 ) 不影响验收滤波。</td> </tr> <tr> <td>1</td> <td>报文方向位 DIR 用于验收滤波。</td> </tr> </tbody> </table>	值	描述	0	报文方向位 ( CANIFnARB2 寄存器的 DIR 位 ) 不影响验收滤波。	1	报文方向位 DIR 用于验收滤波。
值	描述									
0	报文方向位 ( CANIFnARB2 寄存器的 DIR 位 ) 不影响验收滤波。									
1	报文方向位 DIR 用于验收滤波。									
13	保留	RO	1	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。						
12:0	MSK	R/W	0xFF	标识符掩码 在使用 29 位的标识符时，这些位用作 ID 的 [28:16] 位。CANIFnMSK2 寄存器中的 MSK 域用作 ID 的 [15 : 0] 位。在使用 11 位的标识符时，MSK[12:2] 用作 ID 的 [10:0] 位。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>报文对象中的对应标识符域 (ID) 不能禁用验收滤波中的匹配。</td> </tr> <tr> <td>1</td> <td>报文对象中的对应标识符域 (ID) 用于验收滤波。</td> </tr> </tbody> </table>	值	描述	0	报文对象中的对应标识符域 (ID) 不能禁用验收滤波中的匹配。	1	报文对象中的对应标识符域 (ID) 用于验收滤波。
值	描述									
0	报文对象中的对应标识符域 (ID) 不能禁用验收滤波中的匹配。									
1	报文对象中的对应标识符域 (ID) 用于验收滤波。									

**寄存器 16: CAN IF1 仲裁寄存器 1 ( CANIF1ARB1 ) , 偏移量 0x030**

**寄存器 17: CAN IF2 仲裁寄存器 1 ( CANIF2ARB1 ) , 偏移量 0x090**

这些寄存器保存着用于验收滤波的标识符。

CAN IFn 仲裁寄存器 1 ( CANIFnARB1 )

CAN0 基址: 0x4004.0000

CAN1 基址: 0x4004.1000

偏移量 0x030

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ID															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	ID	R/W	0x0000	报文标识符 此位域结合 CANIFnARB2 寄存器的 ID 域，可用于创建报文标识符。 在使用 29 位的标识符时，CANIFnARB1 寄存器的 15:0 位即 ID 的 [15:0]，而 CANIFnARB2 寄存器的 12:0 位即 ID 的 [28:16]。 对于 11 位报文标识符，本位域无意义。

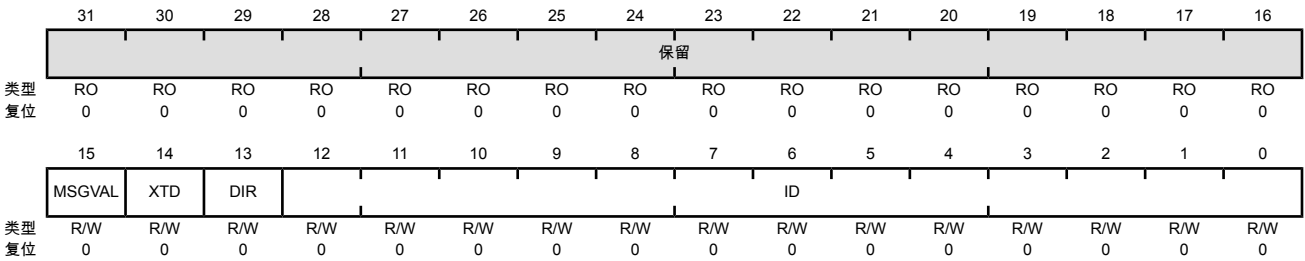
**寄存器 18: CAN IF1 仲裁寄存器 2 ( CANIF1ARB2 ) , 偏移量 0x034**

**寄存器 19: CAN IF2 仲裁寄存器 2 ( CANIF2ARB2 ) , 偏移量 0x094**

这些寄存器保存着用于验收滤波的信息。

**CAN IFn 仲裁寄存器 2 ( CANIFnARB2 )**

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x034  
 类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述						
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。						
15	MSGVAL	R/W	0	报文有效  <table border="1"> <tr> <th>值</th> <th>描述</th> </tr> <tr> <td>0</td> <td>报文对象被报文处理器忽略。</td> </tr> <tr> <td>1</td> <td>报文对象已经配置完成，可随时由 CAN 控制器的报文处理器处理。</td> </tr> </table>	值	描述	0	报文对象被报文处理器忽略。	1	报文对象已经配置完成，可随时由 CAN 控制器的报文处理器处理。
值	描述									
0	报文对象被报文处理器忽略。									
1	报文对象已经配置完成，可随时由 CAN 控制器的报文处理器处理。									
14	XTD	R/W	0	扩展标识符  <table border="1"> <tr> <th>值</th> <th>描述</th> </tr> <tr> <td>0</td> <td>报文对象采用 11 位的标准标识符。</td> </tr> <tr> <td>1</td> <td>报文对象采用 29 位的扩展标识符。</td> </tr> </table>	值	描述	0	报文对象采用 11 位的标准标识符。	1	报文对象采用 29 位的扩展标识符。
值	描述									
0	报文对象采用 11 位的标准标识符。									
1	报文对象采用 29 位的扩展标识符。									
13	DIR	R/W	0	报文方向  <table border="1"> <tr> <th>值</th> <th>描述</th> </tr> <tr> <td>0</td> <td>接收。当 CANIFnMCTL 寄存器中的 TXRQST 位置位时，即表明接收到带有此报文对象标识符的远程帧。当接收到标识符匹配的数据帧时，报文将存储与该报文对象中。</td> </tr> <tr> <td>1</td> <td>发送。当 CANIFnMCTL 寄存器中的 TXRQST 位置位时，相应的报文对象将作为数据帧发送。当接收到标识符匹配的远程帧时，该报文对象的 TXRQST 位将置位（如果 RMTEN = 1）。</td> </tr> </table>	值	描述	0	接收。当 CANIFnMCTL 寄存器中的 TXRQST 位置位时，即表明接收到带有此报文对象标识符的远程帧。当接收到标识符匹配的数据帧时，报文将存储与该报文对象中。	1	发送。当 CANIFnMCTL 寄存器中的 TXRQST 位置位时，相应的报文对象将作为数据帧发送。当接收到标识符匹配的远程帧时，该报文对象的 TXRQST 位将置位（如果 RMTEN = 1）。
值	描述									
0	接收。当 CANIFnMCTL 寄存器中的 TXRQST 位置位时，即表明接收到带有此报文对象标识符的远程帧。当接收到标识符匹配的数据帧时，报文将存储与该报文对象中。									
1	发送。当 CANIFnMCTL 寄存器中的 TXRQST 位置位时，相应的报文对象将作为数据帧发送。当接收到标识符匹配的远程帧时，该报文对象的 TXRQST 位将置位（如果 RMTEN = 1）。									

在初始化过程中以及在 CANCTL 寄存器的 INIT 位清零之前，所有未用的报文对象会将该位清零。在 CANIFnARBn 寄存器中的 ID 位、CANIFnARB2 寄存器中的 XTD 和 DIR 位或者 CANIFnMCTL 寄存器中的 DLC 位中的任一位被修改之前，或者不再需要报文对象时，也必须将 MSGVAL 位清零。



---

位/域	名称	类型	复位	描述
12:0	ID	R/W	0x000	<p>报文标识符</p> <p>此位域结合 CANIFnARB2 寄存器的 ID 域，可用于创建报文标识符。</p> <p>在使用 29 位的标识符时，CANIFnARB1 寄存器的 ID[15:0] 位域即 ID 的 [15:0]，而 ID[12:0] 位即 ID 的 [28:16]。</p> <p>在使用 11 位的标识符时，ID[12:2] 将用作 ID 的 [10:0] 位。CANIFnARB1 寄存器的 ID 域会被忽略。</p>

**寄存器 20: CAN IF1 报文控制寄存器 ( CANIF1MCTL ) , 偏移量 0x038**

**寄存器 21: CAN IF2 报文控制寄存器 ( CANIF2MCTL ) , 偏移量 0x098**

该寄存器保存了与将要发送给报文 RAM 的报文对象相关的控制信息。

CAN IFn 报文控制寄存器 ( CANIFnMCTL )

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x038  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB	保留			DLC			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述						
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。						
15	NEWDAT	R/W	0	<p>新数据</p> <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>自从此标志位上一次由 CPU 清零之后，报文控制器并未向此报文对象的数据部分写入新的数据。</td> </tr> <tr> <td>1</td> <td>报文处理器或 CPU 已经向此报文对象的数据部分写入了新的数据。</td> </tr> </table>	值	描述	0	自从此标志位上一次由 CPU 清零之后，报文控制器并未向此报文对象的数据部分写入新的数据。	1	报文处理器或 CPU 已经向此报文对象的数据部分写入了新的数据。
值	描述									
0	自从此标志位上一次由 CPU 清零之后，报文控制器并未向此报文对象的数据部分写入新的数据。									
1	报文处理器或 CPU 已经向此报文对象的数据部分写入了新的数据。									
14	MSGLST	R/W	0	<p>报文丢失</p> <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>此位自上一次由 CPU 清零之后，并未丢失新的报文。</td> </tr> <tr> <td>1</td> <td>NEWDAT 位置位时，报文处理器将新的报文存入这个对象中；CPU 丢失了报文。</td> </tr> </table> <p>仅当报文对象 CANIFnARB2 寄存器的 DIR 位清零 (接收方向) 时，此标志位才有效。</p>	值	描述	0	此位自上一次由 CPU 清零之后，并未丢失新的报文。	1	NEWDAT 位置位时，报文处理器将新的报文存入这个对象中；CPU 丢失了报文。
值	描述									
0	此位自上一次由 CPU 清零之后，并未丢失新的报文。									
1	NEWDAT 位置位时，报文处理器将新的报文存入这个对象中；CPU 丢失了报文。									
13	INTPND	R/W	0	<p>中断挂起</p> <table border="0"> <tr> <td>值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>此报文对象不是中断源。</td> </tr> <tr> <td>1</td> <td>此报文对象是中断源。如果此时不存在优先级更高的中断，那么 CANINT 寄存器中的中断标识符将指向该报文对象。</td> </tr> </table>	值	描述	0	此报文对象不是中断源。	1	此报文对象是中断源。如果此时不存在优先级更高的中断，那么 CANINT 寄存器中的中断标识符将指向该报文对象。
值	描述									
0	此报文对象不是中断源。									
1	此报文对象是中断源。如果此时不存在优先级更高的中断，那么 CANINT 寄存器中的中断标识符将指向该报文对象。									

位/域	名称	类型	复位	描述						
12	UMASK	R/W	0	<p>使用验收掩码</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>掩码被忽略。</td> </tr> <tr> <td>1</td> <td>使用掩码 ( CANIFnMSKn 寄存器的 MSK、MXTD 和 MDIR 位 ) 进行验收滤波。</td> </tr> </tbody> </table>	值	描述	0	掩码被忽略。	1	使用掩码 ( CANIFnMSKn 寄存器的 MSK、MXTD 和 MDIR 位 ) 进行验收滤波。
值	描述									
0	掩码被忽略。									
1	使用掩码 ( CANIFnMSKn 寄存器的 MSK、MXTD 和 MDIR 位 ) 进行验收滤波。									
11	TXIE	R/W	0	<p>传输中断启用</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>成功发送一帧后，CANIFnMCTL 寄存器的 INTPND 位不会变化。</td> </tr> <tr> <td>1</td> <td>成功发送一帧后，CANIFnMCTL 寄存器的 INTPND 位将自动置位。</td> </tr> </tbody> </table>	值	描述	0	成功发送一帧后，CANIFnMCTL 寄存器的 INTPND 位不会变化。	1	成功发送一帧后，CANIFnMCTL 寄存器的 INTPND 位将自动置位。
值	描述									
0	成功发送一帧后，CANIFnMCTL 寄存器的 INTPND 位不会变化。									
1	成功发送一帧后，CANIFnMCTL 寄存器的 INTPND 位将自动置位。									
10	RXIE	R/W	0	<p>接收中断启用</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>成功接收一帧后，CANIFnMCTL 寄存器的 INTPND 位不会变化。</td> </tr> <tr> <td>1</td> <td>成功接收一帧后，CANIFnMCTL 寄存器的 INTPND 位将自动置位。</td> </tr> </tbody> </table>	值	描述	0	成功接收一帧后，CANIFnMCTL 寄存器的 INTPND 位不会变化。	1	成功接收一帧后，CANIFnMCTL 寄存器的 INTPND 位将自动置位。
值	描述									
0	成功接收一帧后，CANIFnMCTL 寄存器的 INTPND 位不会变化。									
1	成功接收一帧后，CANIFnMCTL 寄存器的 INTPND 位将自动置位。									
9	RMTEN	R/W	0	<p>远程启用</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>收到远程帧后，CANIFnMCTL 寄存器的 TXRQST 位不会变化。</td> </tr> <tr> <td>1</td> <td>收到远程帧后，CANIFnMCTL 寄存器的 TXRQST 位将自动置位。</td> </tr> </tbody> </table>	值	描述	0	收到远程帧后，CANIFnMCTL 寄存器的 TXRQST 位不会变化。	1	收到远程帧后，CANIFnMCTL 寄存器的 TXRQST 位将自动置位。
值	描述									
0	收到远程帧后，CANIFnMCTL 寄存器的 TXRQST 位不会变化。									
1	收到远程帧后，CANIFnMCTL 寄存器的 TXRQST 位将自动置位。									
8	TXRQST	R/W	0	<p>发送请求</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>此报文对象并未等待发送。</td> </tr> <tr> <td>1</td> <td>已经请求发送此报文对象，并且尚未发送完成。</td> </tr> </tbody> </table> <p>注意：若 CANIFnCMSK 寄存器的 WRNRD 和 TXRQST 位置位，则此标志位将被忽略。</p>	值	描述	0	此报文对象并未等待发送。	1	已经请求发送此报文对象，并且尚未发送完成。
值	描述									
0	此报文对象并未等待发送。									
1	已经请求发送此报文对象，并且尚未发送完成。									
7	EOB	R/W	0	<p>缓冲区末端</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>此报文对象属于 FIFO 缓冲区，但不是此缓冲区的最后一个报文对象。</td> </tr> <tr> <td>1</td> <td>此报文对象是 FIFO 缓冲区的最后一个报文对象，或只是单个报文对象。</td> </tr> </tbody> </table> <p>该位用于串联两个或多个报文对象 ( 多达 32 个 ) 以创建 FIFO 缓冲区。当为单个报文对象 ( 因而不属于 FIFO 缓冲区 ) 时，该位必须被置位。</p>	值	描述	0	此报文对象属于 FIFO 缓冲区，但不是此缓冲区的最后一个报文对象。	1	此报文对象是 FIFO 缓冲区的最后一个报文对象，或只是单个报文对象。
值	描述									
0	此报文对象属于 FIFO 缓冲区，但不是此缓冲区的最后一个报文对象。									
1	此报文对象是 FIFO 缓冲区的最后一个报文对象，或只是单个报文对象。									

位/域	名称	类型	复位	描述						
6:4	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。						
3:0	DLC	R/W	0x0	数据长度码						
				<table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x0-0x8</td> <td>指定数据帧中的字节数。</td> </tr> <tr> <td>0x9-0xF</td> <td>默认数据帧长度为 8 字节。</td> </tr> </tbody> </table>	值	描述	0x0-0x8	指定数据帧中的字节数。	0x9-0xF	默认数据帧长度为 8 字节。
值	描述									
0x0-0x8	指定数据帧中的字节数。									
0x9-0xF	默认数据帧长度为 8 字节。									
				<p>报文对象的 CANIFnMCTL 寄存器中的 DLC 域的定义必须和含相同标识符的其它节点上的相应对象相同。在报文处理器存储数据帧时，它将 DLC 写入由接收报文给定的值中。</p>						

寄存器 22: CAN IF1 数据寄存器 A1 ( CANIF1DA1 ) , 偏移量 0x03C

寄存器 23: CAN IF1 数据寄存器 A2 ( CANIF1DA2 ) , 偏移量 0x040

寄存器 24: CAN IF1 数据寄存器 B1 ( CANIF1DB1 ) , 偏移量 0x044

寄存器 25: CAN IF1 数据寄存器 B2 ( CANIF1DB2 ) , 偏移量 0x048

寄存器 26: CAN IF2 数据寄存器 A1 ( CANIF2DA1 ) , 偏移量 0x09C

寄存器 27: CAN IF2 数据寄存器 A2 ( CANIF2DA2 ) , 偏移量 0x0A0

寄存器 28: CAN IF2 数据寄存器 B1 ( CANIF2DB1 ) , 偏移量 0x0A4

寄存器 29: CAN IF2 数据寄存器 B2 ( CANIF2DB2 ) , 偏移量 0x0A8

这些寄存器包含将要发送或刚刚收到的数据。在 CAN 数据帧中, 数据字节 0 是发送或接收的第一个字节, 数据字节 7 是发送或接收的最后一个字节。在 CAN 的串行比特流中, 每个数据字节的最高有效位 (MSB) 都将首先发送。

#### CAN IFn 数据寄存器 nn ( CANIFnDnn )

CAN0 基址: 0x4004.0000

CAN1 基址: 0x4004.1000

偏移量 0x03C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	DATA	R/W	0x0000	数据 CANIFnDA1 寄存器包含数据字节 1 和 0; CANIFnDA2 寄存器包含数据字节 3 和 2; CANIFnDB1 寄存器包含数据字节 5 和 4; CANIFnDB2 寄存器包含数据字节 7 和 6。

**寄存器 30: CAN 传输请求寄存器 1 ( CANTXRQ1 ) , 偏移量 0x100**

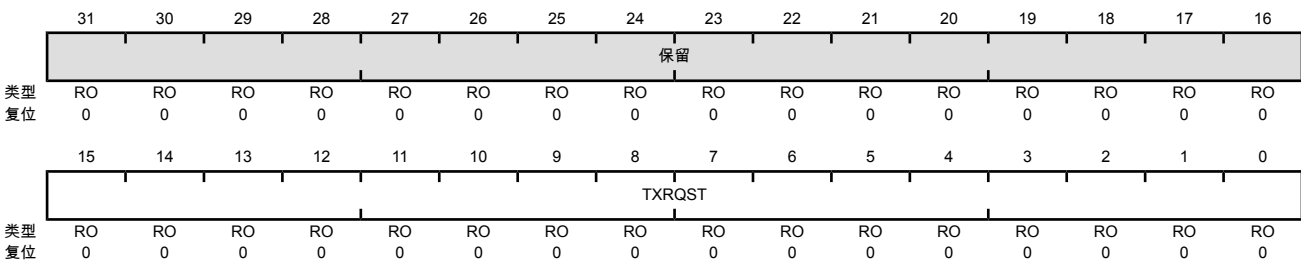
**寄存器 31: CAN 传输请求寄存器 2 ( CANTXRQ2 ) , 偏移量 0x104**

CANTXRQ1 和 CANTXRQ2 寄存器保存了 32 个报文对象的 TXRQST 位。通过读取这些位, CPU 可以检查哪些报文对象有挂起的发送请求。某一特定报文对象的 TXRQST 位可以通过以下三种源变更: (1) CPU ( 通过 CANIFnMCTL 寄存器修改 ); (2) 报文处理器状态机 ( 收到远程帧后 ); (3) 报文处理器状态机 ( 在成功发送报文后 )。

CANTXRQ1 寄存器包含报文 RAM 中前 16 个报文对象的 TXRQST 位; CANTXRQ2 寄存器包含报文 RAM 中后 16 个报文对象的 TXRQST 位。

CAN 传输请求寄存器 n ( CANTXRQn )

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x100  
 类型 RO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	TXRQST	RO	0x0000	传输请求位
			值	描述
			0	相应的报文对象并未等待发送。
			1	相应的报文对象已请求发送, 并且尚未发送完成。

**寄存器 32: CAN 新数据寄存器 1 ( CANNWDA1 ) , 偏移量 0x120**

**寄存器 33: CAN 新数据寄存器 2 ( CANNWDA2 ) , 偏移量 0x124**

CANNWDA1 和 CANNWDA2 寄存器总共保存 32 个报文对象的 NEWDAT 位状态。通过读取这些位，CPU 可以检查哪些报文对象的数据部分有更新。某一特定报文对象的 NEWDAT 位可以通过以下三种源变更：(1) CPU ( 通过 CANIFnMCTL 寄存器修改 )；(2) 报文处理器状态机 ( 在收到数据帧后 )；(3) 报文处理器状态机 ( 在成功发送报文后 )。

CANNWDA1 寄存器包含报文 RAM 中前 16 个报文对象的 NEWDAT 位；CANNWDA2 寄存器包含报文 RAM 中后 16 个报文对象的 NEWDAT 位。

#### CAN 新数据寄存器 n ( CANNWDA<sub>n</sub> )

CAN0 基址: 0x4004.0000

CAN1 基址: 0x4004.1000

偏移量 0x120

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NEWDAT															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	NEWDAT	RO	0x0000	新数据位
	值	描述		
	0	自从此标志位上次被 CPU 清零以来，相应报文对象的数据部分并未写入新的数据。		
	1	报文处理器或 CPU 向相应报文对象的数据部分写入了新的数据。		

**寄存器 34: CAN 报文 1 中断挂起寄存器 ( CANMSG1INT ) , 偏移量 0x140**

**寄存器 35: CAN 报文 2 中断挂起寄存器 ( CANMSG2INT ) , 偏移量 0x144**

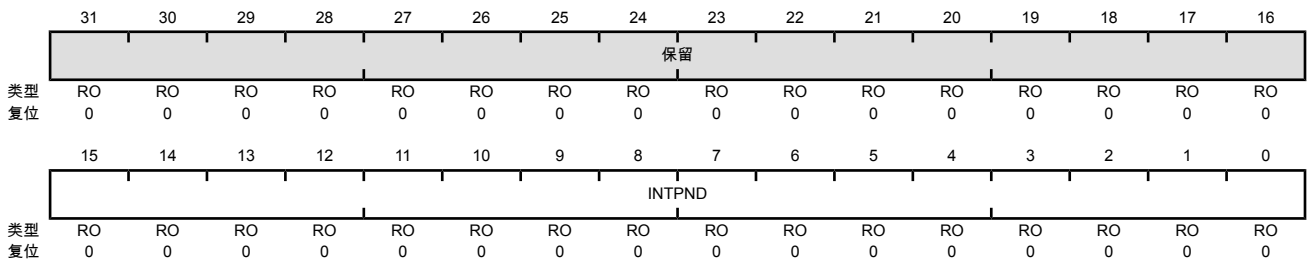
CANMSG1INT 和 CANMSG2INT 寄存器保存了 32 个报文对象的 INTPND 位。通过读取这些位，CPU 可以检查哪些报文对象有挂起的中断。某一特定报文对象的 INTPND 位可以通过以下 2 种方式改变：(1) CPU ( 通过 CANIFnMCTL 寄存器修改 )；(2) 报文处理器状态机 ( 在成功接收或发送一帧后 )。

该域也可以通过 CANINT 寄存器中编码。

CANMSG1INT 寄存器包含报文 RAM 中前 16 个报文对象的 INTPND 位；CANMSG2INT 寄存器包含报文 RAM 中后 16 个报文对象的 INTPND 位。

**CAN 报文 n 中断挂起寄存器 ( CANMSGnINT )**

CAN0 基址: 0x4004.0000  
 CAN1 基址: 0x4004.1000  
 偏移量 0x140  
 类型 RO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	INTPND	RO	0x0000	中断挂起位
			值	描述
			0	相应报文对象并非中断源。
			1	相应报文对象是中断源。



**寄存器 36: CAN 报文 1 有效寄存器 ( CANMSG1VAL ) , 偏移量 0x160**

**寄存器 37: CAN 报文 2 有效寄存器 ( CANMSG2VAL ) , 偏移量 0x164**

CANMSG1VAL 和 CANMSG2VAL 寄存器共同保存 32 个报文对象的 MSGVAL 位。通过读取这些位, CPU 可以检查哪些报文对象有效。某一特定报文对象的报文有效位可以通过 CANIFnARB2 寄存器予以变更。

CANMSG1VAL 寄存器包含报文 RAM 中前 16 个报文对象的 MSGVAL 位; CANMSG2VAL 寄存器包含报文 RAM 中后 16 个报文对象的 MSGVAL 位。

#### CAN 报文 n 有效寄存器 ( CANMSGnVAL )

CAN0 基址: 0x4004.0000  
CAN1 基址: 0x4004.1000  
偏移量 0x160  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSGVAL															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	MSGVAL	RO	0x0000	报文有效位
	值			描述
	0			相应报文对象并未配置, 将被报文处理器忽略。
	1			相应报文对象已完成配置, 应由报文处理器处理。

## 18 通用串行总线 (USB) 控制器

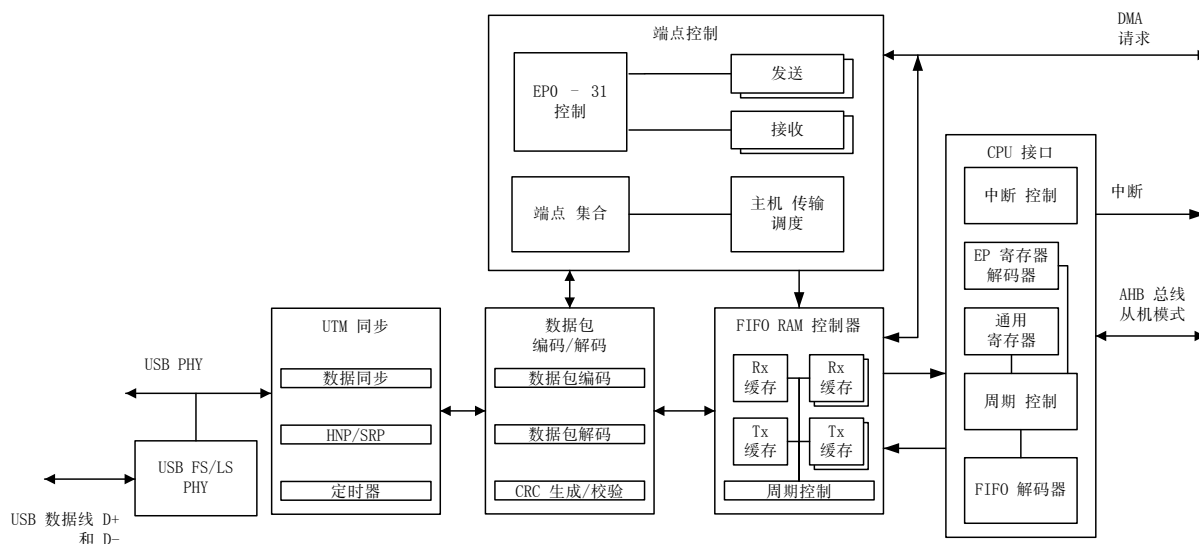
该 Stellaris® USB 控制器支持 USB 主机/设备/OTG 功能，在点对点通信过程中可运行在全速和低速模式。它符合 USB2.0 标准，包含挂起和恢复信号。该控制器提供 16 个端点，其中两个是控制传输的专用连接端点（一个用于输入，一个用于输出），另外 14 个是固件定义的端点。此外，它还带有一个大小可动态变化的 FIFO，以支持多包队列。可通过  $\mu$ DMA 来访问 FIFO，将对系统软件的依赖降至最低。USB 设备启动方式灵活，可软件控制是否在启动时连接。USB 控制器遵从 OTG 标准的会话请求协议 (SRP) 和主机协商协议 (HNP)。

该 Stellaris USB 模块有如下特性：

- 符合 USB-IF 认证标准
- 支持 USB 2.0 全速模式 (12 Mbps) 和低速模式 (1.5 Mbps)
- 四种传输类型：控制传输、中断传输、批次传输和等时传输
- 16 个端点
  - 一个专用的输入控制端点和一个专用输出控制端点
  - 7 个可配置的输入端点和 7 个可配置的输出端点
- 4 KB 专用端点内存空间：一个端点可定义为双缓存的 1023 字节最大包长的等时传输
- 支持 VBUS 电压浮动 (droop) 和有效 ID 检测，并产生中断信号
- 用微型直接内存访问 ( $\mu$ DMA) 有效的传输数据
  - 用于发送和接收的独立通道多达 3 个输入端点和 3 个输出端点
  - 当发送 FIFO 中包含所需数量的数据后，可产生通道请求

## 18.1 结构图

图 18-1. USB 模块结构图



## 18.2 信号描述

下表列出了与 USB 控制器的外部信号并逐一描述其功能。一些 USB 控制器的信号是 GPIO 的复用功能，这些管脚在复位时默认设置为 GPIO 信号。表中“复用管脚/赋值”一列列出了 USB 信号的可能管脚位置。应当将 GPIO 复用功能选择 (GPIOAFSEL) 寄存器 (618 页) 中的 AFSEL 位置位以选择 USB 功能。括号里边的数字表示必须编入 GPIO 端口控制 (GPIOCTL) 寄存器 (636 页) 中的 PMCn 位域里的编码，用以将 USB 信号分配给指定的 GPIO 端口管脚。通过将 GPIO 数字启用 (GPIODEN) 寄存器中的 DEN 位清零来配置 USB0VBUS 和 USB0ID 信号。有关如何配置 GPIO 的更多信息，请参考“通用输入/输出端口 (GPIO)”在 597 页。其余的管脚 (在管脚复用/赋值栏中标记为“固定”的管脚) 都有一个固定的管脚赋值和功能。

**注意：** 在 OTG 模式，由于 USB0VBUS 和 USB0ID 是 USB 控制器专用的管脚，所以无需配置，可直接连接到 USB 连接器的 VBUS 和 ID 信号。如果 USB 控制器用作专用的主机或设备，USB 通用控制和状态 (USBGPCS) 寄存器中的 DEVMODOTG 和 DEVMOD 位可用于将 USB0VBUS 和 USB0ID 输入连接到内部固定电平，从而释放 PB0 和 PB1 管脚，以用于 GPIO。当用作自供电的设备时，需要监测 VBUS，来确定主机是否断开 VBUS，以及自供电设备是否禁用 D+/D- 上拉电阻。此功能可通过将一个标准 GPIO 连接到 VBUS 实现。

表 18-1. USB 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
USB0DM	95	PL7	I/O	模拟	USB0 的双向差分数据管脚 (USB 规范中的 D-)。
USB0DP	96	PL6	I/O	模拟	USB0 的双向差分数据管脚 (USB 规范中的 D+)。
USB0EPEN	3 34 51 61	PD2 (8) PC6 (8) PG4 (8) PF4 (8)	O	TTL	可选择性地以主机模式控制外部电源，向 USB 总线供电。

表 18-1. USB 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
USB0ID	97	PB0	I	模拟	此信号用于检测 USB ID 信号的状态。此时 USB PHY 将在内部启用一个上拉电阻，通过外部元件（USB 连接器）检测 USB 控制器的初始状态（即电缆的 A 侧设置下拉电阻，B 侧设置上拉电阻）。
USB0PFLT	4 33 50 60	PD3 (8) PC7 (8) PG5 (8) PF5 (8)	I	TTL	可选择性地用于主机模式的外部电源，指示外部电源的错误状态。
USB0VBUS	98	PB1	I/O	模拟	此信号用于会话请求协议。USB PHY 可通过此信号检测 VBUS 的电平，并在 VBUS 脉冲期间短时下拉。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 18.3 功能描述

该 Stellaris 支持 OTG 标准的会话请求协议 (SRP) 和主机协商协议 (HNP)，提供完整的 OTG 协商。会话请求协议 (SRP) 允许连接在 USB 线缆 B 端的 B 类设备向连接在 A 端的 A 类设备发出请求，通知 A 类设备打开 VBUS 电源。主机协商协议 (HNP) 用于在初始会话请求协议提供供电后，决定 USB 线缆哪端的设备作为 USB Host 主控制器。当该设备连接到非 OTG 外设或设备时，控制器可以检测出线缆终端所使用的设备类型，并提供一个寄存器来指示该控制器是用作主机控制器还是用作设备控制器。之上所提供的动作都是由 USB 控制器自动处理的。基于这种自动探测机制，系统使用 A 类/B 类连接器取代 A/B 类连接器，可支持与另外的 OTG 设备完整的 OTG 协商。

另外，USB 控制器还提供对连接到非 OTG 的外设或主机控制器的支持。可将 USB 控制器配置为专用的主机或设备模式，此时，USB0VBUS 和 USB0ID 信号可作为 GPIO 使用。但当 USB 控制器用作自供电设备时，必须将 GPIO 输入管脚或模拟比较器输入管脚连接到 VBUS 管脚上，并配置为在 VBUS 掉电时会产生中断。该中断用于禁用 USB0DP 信号上的的上拉电阻。

注意：当 USB 模块运行时，MOSC 必须为时钟源（无论是否使用 PLL），并且系统时钟必须至少为 30 MHz。

### 18.3.1 用作设备时的操作

本节描述了该 Stellaris USB 控制器用作 USB 设备时的操作。在 USB 控制器的运行模式由设备模式改变为主机模式或由主机模式改变为设备模式之前，必须通过设置软件复位控制 2 (SRCR2) 寄存器（请参考 440 页）中的 USB0 位来复位 USB 控制器。输入端点、输出端点、进入和退出挂起 (SUSPEND) 模式、帧起始 (SOF) 的识别都在本节有所描述。

运行在设备模式时，输入事务通过使用端点的发送端点寄存器，由端点的发送接口进行控制。输出事务通过使用端点的接收端点寄存器，由端点的接收接口进行控制。

当配置端点的 FIFO 大小时，需要考虑最大数据包大小。

- 批量传输. 批量端点的 FIFO 可配置为最大包长 (最大 64 字节)，如果使用双包缓存，则需要配置为 2 倍于最大包长大小 (后面的章节将详细描述)。
- 中断. 中断端点的 FIFO 可配置为最大包长 (最大 64 字节)，如果使用双包缓存，则需要配置为两倍的包长大小。
- 等时传输. 等时端点的 FIFO 比较灵活，最大支持 1023 字节。
- 控制. 也可以用于为 USB 设备指定一个独立的控制端点。但是在大多数情况下，USB 设备应该在 USB 控制器的端点 0 上使用专用的控制端点。

### 18.3.1.1 集合

当运行在设备模式时，USB 控制器提供两个专用的控制端点（输入和输出）以及可用于与主机控制器进行通信的 14 个可配置的端点（7 个输入和 7 个输出）。端点的端点号和方向与对应的相关寄存器有直接联系。比如，当主机发送到端点 1，所有的配置和数据存在于端点 1 发送寄存器接口中。

端点 0 是专用的控制端点，用于在枚举过程中所有的对端点 0 的控制传输事务，或者是对端点 0 的任意其它的控制请求。端点 0 使用 USB 控制器 FIFO RAM 的前 64 字节，此内存对于输入传输事务和输出传输事务是共用的。

其余的 14 个端点可配置为控制端点、批量端点、中断端点或等时端点。应将他们视为 7 个可配置的输入端点和 7 个可配置的输出端点。这些成对的端点的输入和输出端点的端点类型可以不同。比如端点对的输出部分可以设置为批量端点，而输入部分可以设置为中断端点。每个端点的 FIFO 的地址和大小可以根据应用需求来修改。

### 18.3.1.2 用作设备时的输入传输

USB 控制器作为设备运行时，输入传输的数据通过发送端点的 FIFO 来处理。7 个可配置的输入端点的 FIFO 大小由 USB 发送 FIFO 起始地址 (USBTXFIFOADD) 寄存器决定。传输时发送端点 FIFO 中的最大数据包大小可编程配置，该大小由写入该端点的 USB 端点 n 发送最大传输数据 (USBTXMAXPn) 寄存器中的值决定。端点的 FIFO 可配置为双包缓存或单包缓存。当启用双包缓存启用时，FIFO 中可缓冲两个数据包，这就需要 FIFO 的大小至少为两个数据包大小。当不使用双包缓存时，即使数据包的大小小于 FIFO 大小的一半，也只能缓冲一个数据包。

**注意：** 端点的最大包长不能超过 FIFO 的大小。当数据位于 FIFO 中时，不对 USBTXMAXPn 寄存器执行写操作，否则会造成异常的结果。

#### 单包缓存

如果发送端点 FIFO 的大小小于该端点最大包长的两倍（依照 USB 发送动态 FIFO 大小 [USBTXFIFOSZ] 寄存器设置），只能使用单包缓冲，在 FIFO 中缓冲一个数据包。当将每个数据包都成功地加载到发送 FIFO 时，USB 端点 n 发送控制和状态低字节 (USBTXCSRLn) 寄存器中的 TXRDY 位必须置位。如果 USB 端点 n 发送控制和状态高字节 (USBTXCSRHn) 寄存器中的 AUTOSET 位置位，则当将最大的数据包装载进 FIFO 时，TXRDY 位将自动置位。对于包长小于最大包长的数据包，必须对 TXRDY 位手动置位。当 TXRDY 位置位（无论是手动置位还是自动置位）时，表明要发送的数据包已准备好。如果数据包成功发送，TXRDY 位和 FIFONE 位将被清零，同时产生相应的中断信号。此时下一包数据可装载到 FIFO 中。

#### 双包缓存

如果发送端点 FIFO 的大小至少两倍于该端点最大包长时，允许使用双包缓存，FIFO 中可以缓冲两个数据包。当将每个数据包都加载到发送 FIFO 时，USBTXCSRLn 寄存器中的 TXRDY 位必须置位。如果 USBTXCSRHn 寄存器中的 AUTOSET 位置位，则当将最大的数据包装载进 FIFO 时，TXRDY 位将自动置位。对于包长小于最大包长的数据包，必须对 TXRDY 位手动设置。当 TXRDY 位置位（无论是手动置位还是自动置位）时，表明要发送的数据包已准备好。在装载完第一个包后，TXRDY 位会被立即清零，同时产生一个中断信号。此时，第二个数据包可装载到发送 FIFO 中，TXRDY 位会被重新置位（手动置位或当该数据包为最大包长时自动置位）。就此来说，两个要发送的包都已准备就绪。如果所有的都成功发送，会将 TXRDY 位自动清零，同时产生相应的发送端点中断信号，此时下一包数据可装载到发送 FIFO 中。USBTXCSRLn 寄存器中的 FIFONE 位的状态表明此时可以装载几个数据包。如果 FIFONE 位置位，表明 FIFO 中还有一个包未发送，只能装载一个数据包。如果 FIFONE 位清零，表明 FIFO 中没有未发送的包，可以装载两个数据包。

**注意：** 如果 USB 发送双包缓存禁用 (USBTXDPKTBUFFDIS) 寄存器中的 EPn 位置位，相应的端点禁用双包缓存。此位默认为置位，需要将该位清零才可启用双包缓存时。

### 18.3.1.3 用作设备时的输出事务

USB 控制器处于设备模式时，输出的数据通过接收 FIFO 来处理。7 个可配置的输出端点的接收 FIFO 大小由 USB 接收 FIFO 起始地址 (USBXFIFOADD) 寄存器决定。任何数据包中端点能接收到的最大数据量是由写入 USB 端点 n 接收最大传输数据 (USBXMAXPn) 寄存器中的值决定。端点的 FIFO 可配置为双包缓存或单包缓存，当双包缓存启用时，FIFO 中可缓冲两个数据包。当不使用双包缓存时，即使数据包的大小小于 FIFO 大小的一半，也只能缓冲一个数据包。

注意：最大包大小不能超过 FIFO 的大小。

#### 单包缓存

如果接收端点 FIFO 的大小小于该端点最大包长的两倍时，只能使用单包缓冲，在 FIFO 中缓冲一个数据包。当接收 FIFO 中接收到数据包时，USB 接收端点 n 控制和状态低字节 (USBXCSSLn) 寄存器的 RXRDY 和 FULL 位置位，同时对相应的接收端点发出信号，指示此时正在从 FIFO 中卸下数据包。当数据包卸下后，必须将 RXRDY 位清零以允许接收后续的数据包。此动作会向主机控制器发送一个确认信号。如果 USB 端点 n 接收控制和状态高字节 (USBXCSSLn) 寄存器中的 AUTOCL 位置位，且最大包长的数据包已从 FIFO 中卸下，则会自动地将 RXRDY 位和 FULL 位清零。对于包长小于最大包长的数据包，必须对 RXRDY 位手动清零。

#### 双包缓存

如果接收端点的 FIFO 大小不小于该端点的最大包长的 2 倍时，可以使用双缓冲机制缓存两个数据包。当第一个数据包被接收并加载到 FIFO 时，USBXCSSLn 寄存器中的 RXRDY 位置位，同时产生相应的接收端点中断信号，指示有一个数据包可从 FIFO 卸下。

注意：当接收到第一个数据包时，USBXCSSLn 寄存器中的 FULL 位不会被置位。该位只在接收到第二个数据包且已装载入接收 FIFO 时才置位。

当所有的数据包都卸下后，必须将 RXRDY 位清零以允许接收后续的数据包。如果 USBXCSSLn 寄存器中的 AUTOCL 位置位，且最大包长的数据包已从 FIFO 中卸载，则会自动地将 RXRDY 位清零。对于包长小于最大包长的数据包，必须对 RXRDY 位手动清零。如果 RXRDY 清零时 FULL 位置位，USB 控制器先清除 FULL 位，然后再次置位 RXRDY，表明在 FIFO 中含有正在等待卸下的另一个数据包。

注意：如果 USB 接收双包缓存禁用 (USBXDPKTBUFDIS) 寄存器中的 EPn 位置位，相应的端点禁用双包缓存。此位默认为置位，需要将该位清零才可启用双包缓存时。

### 18.3.1.4 调度

传输事务由主机控制器调度决定，设备无法控制事务调度。该 Stellaris USB 控制器可随时建立传输事务。当传输事务完成或由于某些原因被终止时，会产生中断信号。当主机主控制器发起请求，而设备控制器还未准备就绪，设备会返回一个 NAK 忙信号。

### 18.3.1.5 其他操作

USB 控制器自动响应某些 USB 总线的状况或主机控制器的动作。例如，USB 控制器自动暂停某些控制传输或预期外的零长度输出数据包。

#### 暂停 (STALL) 控制传输

USB 控制器在下面情况下会自动发出一个 STALL 挂起握手信号：

1. 在控制传输的输出数据过程中，Host 主机发送的数据比建立过程中设备请求的数据多。当设备卸下最后一个输出数据包，并置位 USB 控制和状态端点 0 低字节 (USBCSSL0) 寄存器中的 DATAEND 位后，主机却发送了输出令牌包（本该发送输入令牌包），此时 USB 控制器将检测此状况。

2. 在控制传输的输入阶段，主机请求的数据多于建立过程中设备请求的数据。CPU 根据主机发出的关于最后的数据包内容的 ACK 信号，将 TXRDY 清零并将 DATAEND 置位后，主机发送输入令牌包（本该发送输出令牌包）时，USB 控制器将检测这种情况。
3. 主机发送多于 USBRXMAXPn 字节数的输出数据令牌包。
4. 输出状态阶段，主机发送多于 1 个零长度数据包。

#### 零长度输出数据包

零长度输出数据包指示控制传输结束。正常操作时，此包只应在整个长度的设备请求发送完成后接收。

但是，如果在整个的设备请求发送完之前，主机发送零长度输出数据包，就代表此次传输提前结束。USB 控制器自动清空所有 FIFO 中为输入令牌的数据阶段准备的数据，同时将 USBCSRL0 寄存器中的 DATAEND 位置位。

#### 设置设备地址

当主机试图枚举设备时，请求设备把地址由 0 改为其他值。通过向 USB 设备功能地址 (USBFADDR) 寄存器写主机请求的值，即可改变设备地址。但是，向 USBFADDR 寄存器写值时需要小心，避免传输还未完成时改变地址。该寄存器只能在 SET\_ADDRESS 命令完成后才能被置位。像所有控制传输一样，只有在设备离开状态阶段时传输才结束。在 SET\_ADDRESS 命令时，设备在主机输入请求时发送零长度数据包来响应主机。一旦设备响应输入请求，USBFADDR 寄存器应尽快写入新值，以避免丢失发送到新地址的其他新命令。

**注意：** 如果设备在输出事务的数据包中接收到 SET\_ADDRESS 命令，立即向寄存器 USBFADDR 中写入新值，设备将在控制传输过程中改变地址。此时主机发送的输入请求发到了旧地址，设备将无法收到该输入请求，不能退出该控制传输的状态阶段。主机无法在输入请求时得到响应，从而造成枚举失败。

### 18.3.1.6 设备模式挂起

当 USB 总线空闲达到 3 ms 时间，USB 控制器自动进入挂起 (SUSPEND) 模式。如果 USB 中断启用 (USBIE) 寄存器中启用挂起中断，此时会产生中断。当 USB 控制器进入挂起模式，USB PHY 也将进入挂起模式。当检测到恢复信号时，USB 控制器退出挂起模式，同时使 USB PHY 退出挂起模式。此时如果启用了恢复中断，将产生中断信号。置位 USB 电源 (USBPOWER) 寄存器中的 RESUME 位同样可以强制 USB 控制器退出挂起模式。当此位置位时，USB 控制器退出挂起模式，同时在总线上发出恢复信号。RESUME 位必须在 10 ms (最大 15 ms) 后清零，以结束恢复信号。

为满足电源功耗需求，USB 控制器可进入深睡眠模式。USB 控制不支持休眠模式，因为这样会导致所有的内部状态丢失。

### 18.3.1.7 帧起始

当 USB 控制器运行在设备模式，它每 1ms 收到一次主机发出的帧起始包 (SOF)。当收到 SOF 包时，包中包含的 11 位帧号写入 USB 帧值 (USBFRAME) 寄存器中，同时发出 SOF 中断信号，由应用程序处理。一旦 USB 控制器开始收到 SOF 包，它将预期每 1ms 收到 1 次。如果 1.00358 ms 后仍未收到 SOF 包，将假定此包丢失，USBFRAME 寄存器也将不更新。当 SOF 包重新成功接收时，USB 控制器继续，并重新同步这些脉冲。

### 18.3.1.8 USB 复位

当 USB 控制器处于设备模式，并在 USB 总线上检测到复位条件时，USB 控制器将自动进行下面的动作：

- 清空 USBFADDR 寄存器。

- 清空 USB 端点索引 (USBEPIDX) 寄存器。
- 清空所有端点 FIFO。
- 将所有控制/状态寄存器清零。
- 启用所有端点中断。
- 产生复位中断。

如果软件驱动USB控制器接收复位中断，所有打开的管道(pipe)关闭，USB控制器等待总线开始设备枚举。

### 18.3.1.9 连接/断开

USB控制器的USB总线连接由软件控制的。USB PHY 可以通过置位和清零 USBPOWER 寄存器中的 SOFTCONN 位，在正常模式和无驱动模式之间切换。当 SOFTCONN 位置位，USB PHY 处于正常模式，USB 总线上的 USB0DP/USB0DM 被启用。此时，USB控制器不响应除USB复位外的任何信号。

当 SOFTCONN 位清零，USB PHY 处于无驱动模式，USB0DP 和 USB0DM 呈高阻态，此时，USB 控制器对于其他 USB 总线上的设备而言，处于断开状态。由于默认设置为无驱动模式，USB 控制器呈现断开状态，直到 SOFTCONN 位被置位。应用软件可以选择何时设置PHY进入正常模式。具有很长的初始化程序的系统可通过这一点确保初始化完成，以及在连接USB总线之前设备已准备好开始枚举。一旦 SOFTCONN 位置位，USB 控制器可以通过将此位清零来断开连接。

注意： 当设备连接到主机时，USB 控制器不产生中断。但是主机终止会话时，将产生中断信号。

### 18.3.2 用作主机时的操作

当该 Stellaris USB 控制器运行在主机模式时，它可用于同其他 USB 设备的点对点通讯，还可用于连接到集线器，与多个设备进行通讯。在 USB 控制器的运行模式由主机模式改变为设备模式或由设备模式改变为主机模式之前，必须通过设置软件复位控制 2 (SRCR2) 寄存器（请参考 440页）中的 USB0 位来复位 USB 控制器。USB控制器支持全速和低速设备，支持点对点通讯和通过集线器操作。它通过自动执行必要的事务传输，允许USB 2.0集线器使用低速设备和全速设备。支持控制传输、批量传输、等时传输和中断传输。此章节描述了当USB控制器用于主机模式时的操作。输入端点、输出端点、进入和退出挂起(SUSPEND)模式、复位都在本节有所描述。

运行在主机模式时，输入事务由端点的接收接口进行控制。所有输入事务使用端点的接收端点寄存器，所有输出事务使用端点的发送端点寄存器。和设备模式一样，当配置端点的FIFO大小时，需要考虑最大数据包大小。

- 批量传输. 批量端点的FIFO可配置为最大包长(最大64字节)，如果使用双包缓存，则需要配置为2倍于最大包长大小(后面的章节将详细描述)。
- 中断. 中断端点的 FIFO 可配置为最大包长（最大 64 字节），如果使用双包缓存，则需要配置为两倍的包长大小。
- 等时传输. 等时端点的FIFO比较灵活，最大支持1023字节。
- 控制. USB设备可能指定一个独立的控制端点与设备通讯。但在大多数情况，USB设备使用USB控制器的端点0作为专用的控制端点与设备的端点0通讯。

#### 18.3.2.1 集合

端点寄存器用于控制USB端点接口，通过这些接口可与连接的设备进行通讯。端点由一个专用控制输入端点、一个专用控制输出端点、7 个可配置的输出端点和 7 个可配置的输入端点组成。



专用的控制接口只能用于与设备的端点0之间的控制传输。他们用于设备枚举或其他使用设备端点0的控制功能。控制端点的输入和输出事务共享USB控制器FIFO内存的前64字节。其余输入和输出接口可配置为与控制端点、批量端点、中断端点或等时端点通讯。

这些 USB 接口可用于将 7 个独立的输出事务和 7 个独立的输入事务同时调度到任何设备的任何端点。输入和输出控制有成对的三组寄存器。通过配置，它们可以与不同类型的端点以及不同设备的不同端点进行通讯。例如，第一对端点可分开控制，输出部分与设备的批量输出端点1通讯，同时输入部分与设备的中断端点2通讯。

无论用于点对点通讯还是通过集线器通讯，在访问设备之前，必须设置相关的 USB 端点 n 接收功能地址 (USBXFUNCADDRn) 寄存器或 USB 端点 n 发送功能地址 (USBTXFUNCADDRn) 寄存器，以记录每个接收和发送端点将被访问的设备地址。

USB控制器支持通过USB集线器连接设备，它通过一个寄存器实现，该寄存器说明集线器地址和每个传输的端口。FIFO的地址和大小可定制，并可指定用于任一USB输入和输出传输。定制包括每个传输一个 FIFO，不同传输共享 FIFO，以及双包缓存 FIFO。

### 18.3.2.2 作为主机时的输入事务

处理输入事务时将采用与 USB 控制器处于设备模式时处理输出事务类似的方式，但传输事务必须通过置位 USBCSRL0 寄存器中的 REQPKT 位开始，向事务调度表明此端点存在一个活动的传输。此时事务调度向目标设备发送一个输入令牌包。当接收到数据包且置于接收 FIFO 中时，USBCSRL0 寄存器中的 RXRDY 位置位，同时产生相应的接收端点中断信号，指示有一个数据包可从 FIFO 卸下。

当数据包卸下时，必须将 RXRDY 位清零。当最大包长的数据包从 FIFO 卸下时，USBXCSRHn 寄存器中的 AUTOCL 位可用于使 RXRDY 自动清零。当 RXRDY 位清零时，USBXCSRHn 寄存器中的 AUTORQ 位将使 REQPKT 位自动置位。AUTOCL 和 AUTORQ 位可用于  $\mu$ DMA 访问，以完成批量传输，而无需主处理器干预。当 RXRDY 位清零时，控制器向设备发送确认信号。当所要传输的数据包数量不确定时，与该端点相关联的 USB 端点 n 块传输中请求包数量 (USBRQPKTCOUNTn) 寄存器应配置为要传输的数据包数量。每次请求前，USB 控制器都将 USBRQPKTCOUNTn 寄存器中的值减 1。当 USBRQPKTCOUNTn 寄存器中的值减到 0 时，将 AUTORQ 位清零以阻止后面试图进行的事务。如果传输的数量未知，则应当将 USBRQPKTCOUNTn 位清零。AUTORQ 保持置位状态，直到接收到一个短包 (小于寄存器 USBRXMAXPn 中的 MAXLOAD 值) 才清零，此种情形只在批量传输的末尾才可出现。

如果设备用 NAK 响应批量或中断传输的输入令牌，USB 主机控制器将重试，直到达到设置的 NAK 限制次数。但是，如果目标设备用 STALL 握手响应，USB 主机控制器将不重试传输，而将 USBCSRL0 寄存器中的 STALLED 位置位。如果目标设备在需要的时间内不响应输入令牌包，或者包存在 CRC 或位填充错误，USB 主机将重试传输。如果重试三次后目标设备仍无响应，USB 主机控制器会将 REQPKT 位清零，并将 USBCSRL0 寄存器中的 ERROR 位置位。

### 18.3.2.3 作为主机时的输出事务

输出事务的处理，采用与设备模式处理输入事务类似的方式。当数据包加载到发送 FIFO 中时，USBTXCSRln 寄存器中的 TXRDY 位必须置位。如果将 USBTXCSRln 寄存器中的 AUTOSSET 位置位，当最大包长的数据包加载到 FIFO 中时，TXRDY 位自动置位。此外，AUTOSSET 位与  $\mu$ DMA 控制器配合使用，可以在不需要软件干预的情况下完成批量传输。

如果目标设备用 NAK 响应输出令牌包，USB 主机控制器将重试，直到达到设置的 NAK 限制次数。但是，如果目标设备用 STALL 握手响应，USB 控制器将不重试传输，而通过将 USBTXCSRln 寄存器中的 STALLED 位置位来中断主处理器。如果目标设备在需要的时间内不响应输出令牌包，或者包存在 CRC 或位填充错误，USB 主机将重试传输。如果重试三次后目标设备仍无响应，USB 控制器将清空 FIFO，并置位 USBTXCSRln 寄存器中的 ERROR 位。

#### 18.3.2.4 事务调度

事务调度由USB主机控制器自动处理。主机控制器允许根据端点事务类型配置端点通讯调度。中断传输可以是每1帧进行一次，也可以每255帧进行一次，可以在1帧到255之间以1帧增量调度。批量端点不允许调度参数，但在设备的端点不响应时，允许NAK超时。等时端点可以在每帧到每  $2^{16}$  帧之间调度 (2 的幂)。

USB控制器维持帧计数。如果目标设备为全速设备，控制器在每帧开始时自动发送SOF包，同时帧计数加1。如果目标设备为低速设备，将在总线上发送 *K* 状态来保持总线活动，防止低速设备进入挂起模式。

在SOF包发送后，USB主机控制器巡检所有配置好的端点，寻找激活的传输事务。激活的传输事务被定义为 REQPKT 位置位的接收端点或 TXRDY 和/或 FIFONE 位置位的发送端点。

如果传输建立在一帧的第一个调度周期，而且端点的间隔计数器减到0，则等时传输和中断传输开始。所以每个端点的中断传输和等时传输每 *n* 帧才发生一次，其中 *n* 是通过 USB 端点 *n* 主机发送间隔 (USBTXINTERVAL<sub>*n*</sub>) 或 USB 端点 *n* 主机接收间隔 (USBRXINTERVAL<sub>*n*</sub>) 寄存器为该端点设置的间隔。

如果在帧中下一个SOF包之前有足够的提供足够的时间完成传输，则激活的批量传输立即开始。如果传输需要重发时(例如，收到NAK或设备未响应)，需要在调度器先检查完其他所有端点是否有激活的传输之后，传输才能重传。这保证了一个发送大量NAK响应的端点不阻塞总线上的其他传输正常进行。控制器同样允许用户设置目标设备发送NAK的端点的超时限制。

#### 18.3.2.5 USB集线器

下面的建立要求只适用于使用USB 2.0集线器的主机。当全速设备或低速设备通过 USB 2.0 集线器连接到 USB 控制器时，集线器地址和端口的详细信息必须记录在相应的 USB 端点 *n* 接收集线器地址 (USBRXHUBADDR<sub>*n*</sub>) 和 USB 端点 *n* 接收集线器端口 (USBRXHUBPORT<sub>*n*</sub>) 或 USB 端点 *n* 发送集线器地址 (USBTXHUBADDR<sub>*n*</sub>) 和 USB 端点 *n* 发送集线器端口 (USBTXHUBPORT<sub>*n*</sub>) 寄存器中。此外，设备的运行速度 (全速或低速) 必须记录在相应的 USB 类型端点 0 (USBTYP0) 寄存器、USB 端点 *n* 主机配置发送类型 (USBTXTYPE<sub>*n*</sub>) 寄存器，或者 USB 端点 *n* 主机配置接收类型 (USBRXTYPE<sub>*n*</sub>) 寄存器中。

对于集线器通讯，这些寄存器的设置记录接入的USB设备的当前相应端点的配置。为了支持更多数量的设备，USB主机控制器允许通过简单的更新记录在这些寄存器中的地址和速度信息来动态改变这些配置。设备功能端点配置的改变必须在相关端点正在进行的传输完成后进行。

#### 18.3.2.6 超时干扰

USB主机控制器直到总线至少空闲最小包间隔的时间，才开始传输。USB控制器不会发起事务传输，除非它能在帧结束前完成。如果USB总线上在帧结束时仍有活动，USB主机将假定连接的目标设备发生故障，同时USB控制器挂起所有传输事务，并产生超时干扰(Babble)中断信号。

#### 18.3.2.7 主机挂起

如果 USBPOWER 寄存器中的 SUSPEND 位置位，USB 主机控制器完成当前的传输事务，然后停止事务调度和帧计数。此时，不再启动事务传输，不再产生SOF包。

要退出挂起模式，可以将 RESUME 位置位并将 SUSPEND 位清零。当 RESUME 位置位时，USB 主机控制器将在总线上产生恢复信号。但 20 ms 之后，必须将 RESUME 位清零，此时，帧计数和事务调度开始。主机支持远程恢复检测。

#### 18.3.2.8 USB复位

如果 USBPOWER 寄存器中的 RESET 位置位，USB 主机控制器将在总线上产生 USB 复位信号。RESET 位需要保持置位至少 20 ms，以确保目标设备的正确复位。CPU清除此位后，USB主机控制器开始帧计数和事务调度。

### 18.3.2.9 连接/断开

通过将 USB 设备控制 (USBDEVCTL) 寄存器中的 SESSION 位置位来启动会话。当检测到设备时，将产生连接中断信号。连接的设备的速度可通过读 USBDEVCTL 寄存器来确定。如果 FSDEV 位置位，连接的设备为全速设备；如果 LSDEV 位置位，连接的设备为低速设备。USB 控制器必须对设备发出一个复位信号，此时 USB 主机开始设备枚举。如果会话过程中设备断开连接，将产生断开中断信号。

### 18.3.3 OTG 模式

为了节省电源，USB OTG 允许当需要时才给 VBUS 上电，当不使用 USB 总线时，则关断 VBUS。VBUS 总是由总线上的 A 设备提供。OTG 控制器通过 PHY 采样 ID 输入来决定哪个是 A 设备哪个是 B 设备。ID 信号拉低时，检测到插入 A 设备 (表示 OTG 控制器作为 A 设备角色)；ID 信号为高时，检测到插入 B 设备 (表示 OTG 控制器作为 B 设备角色)。注意当在 OTG A 和 OTG B 之间切换时，控制器保留所有的寄存器内容。

#### 18.3.3.1 开始会话

当 USB OTG 控制器准备开始会话时，USBDEVCTL 寄存器中的 SESSION 位必须置位。此时 OTG 控制器启用 ID 脚感应。当检测到 A 类型连接时，ID 输入为低；当检测到 B 类型连接时，ID 输入为高。也可将 USBDEVCTL 寄存器中的 DEV 位置位，以表明 USB OTG 控制器用作 A 设备还是 B 设备。该 USB OTG 控制器也提供中断信号，以表明 ID 管脚已完成感应，USBDEVCTL 寄存器中的模式是有效的。通过寄存器 USBIDVIM 可启用此中断，可通过寄存器 USBIDVISC 查看其状态。当 USB 控制器检测到处于线缆的 A 端，必须尽快在 100ms 内启用 VBUS 电源。

如果 USB OTG 控制器是 A 设备，则它进入主机模式 (A 设备总是默认为主机)，打开 VBUS 电源，等待 VBUS 达到有效门限以上 (USBDEVCTL 寄存器中的 VBUS 位域值为 0x3)。此时，OTG 控制器等待外设接入。当检测到外设接入，则产生一个连接中断信号，USBDEVCTL 寄存器中的 FSDEV 或 LSDEV 位置位 (取决于接入的是全速设备还是低速设备)。这时，USB 控制器向接入的设备发送一个复位信号。可以通过将 USBDEVCTL 寄存器中的 SESSION 位清零来结束会话。如果发生超时干扰或 VBUS 掉到会话有效电压以下，OTG 控制器将自动结束会话。

**注意：** 当接入高电流设备，可能会造成 USB OTG 控制器不能保留在主机模式。一些设备吸收大量电流会暂时将 VBUS 拉低到 VBUS 有效电压之下，造成控制器退出主机模式。保持控制器在主机模式，唯一的方法是允许 VBUS 低于会话结束电平。此时，设备引起 VBUS 重复掉到有效电压下又恢复正常。

此外，当设备被告之可以使用它的激活配置启动时，OTG 控制器可能不能保留在主机模式。此时，设备吸取大量的电流，可能造成 VBUS 掉到 VBUS 有效电压以下。

如果 OTG 控制器用作 B 设备，它使用 USB OTG 规范中定义的会话请求协议来请求会话。首先它释放 VBUS，然后当 VBUS 电平低于会话结束门限 (USBDEVCTL 寄存器中的 VBUS 位域值为 0x0)，总线保持单一结束零状态 (single-ended zero, SE0) 大于 2 ms 时间时，USB OTG 控制器将向数据线和 VBUS 线发送脉冲。会话结束时，SESSION 位可通过 USB OTG 控制器或应用软件清零。OTG 控制器使 PHY 切断 D+ 上的上拉电阻，向 A 设备发送会话结束信号。

#### 18.3.3.2 活动性探测

当 OTG 的其他设备希望发起会话时，如果它是 A 设备，将 VBUS 电平提升到会话有效电压之上；如果它是 B 设备，它将向数据线盒 VBUS 线发送脉冲。USB 控制器依据这些动作，决定发起建立会话的是 A 设备还是 B 设备。如果 VBUS 提升到会话有效电压之上，则 USB 控制器用作 B 设备。USB 控制器将 USBDEVCTL 寄存器中的 SESSION 位置位。当探测到总线上有复位信号，将产生复位中断信号，作为会话的起始信号。

USB 控制器的默认模式是用作 B 设备的设备模式。会话结束时，A 设备关断 VBUS 电源。当 VBUS 下降到会话有效电平之下时，USB 控制器将检测到此下降信号，并将 SESSION 位清零来表明会话已经结束，同时引发一个断开中断的信号。如果检测到数据行和 VBUS 脉冲，则 USB 控制器用作 A

设备。该控制器产生会话请求中断信号来指示 B 设备在请求会话。必须将 USBDEVCTL 寄存器中的 SESSION 位置位才可启动会话。

### 18.3.3.3 主机协商

当 USB 控制器是 A 设备，ID 信号为低，当会话发起时它将自动进入主机模式。当 USB 控制器是 B 设备，ID 信号为高，当会话发起时它将自动进入设备状态。但是，可以通过软件方式将 USBDEVCTL 寄存器中的 HOSTREQ 位置位，以使 USB 控制器变为主机。此位可以在通过置位 USBDEVCTL 寄存器中的 SESSION 位以发起会话请求的同时置位，也可以在发起请求之后任意时刻置位。当 USB 控制器下次进入挂起模式时，如果 HOSTREQ 位保持置位，控制器进入主机模式，并通过引发 PHY 断开 D+ 上的上拉电阻开始主机协商（在 USB OTG 补充协议中规定），使 A 设备切换到设备模式，并连接设备模式的上拉电阻。当 USB 控制器检测到此情况，将产生连接中断信号，并将 USBPOWER 寄存器中的 RESET 位置位使 A 设备复位。USB 控制器自动开始复位序列，确保复位在 A 设备连接上拉电阻的 1ms 内开始。主处理器应等待至少 20 ms，然后将 RESET 位清零，开始枚举 A 设备。

当 USB OTG 控制器 B 设备结束使用总线，它将 USBPOWER 寄存器中的 SUSPEND 位置位进入挂起模式。A 设备检测此情况，则结束会话或恢复到主机模式。如果 A 设备是 OTG 控制器，将产生一个连接断开的中断信号。

### 18.3.4 DMA 操作

USB 外设提供了连接到  $\mu$ DMA 控制器的接口。 $\mu$ DMA 控制器提供了三个发送端点和三个接收端点的独立通道。通过 USB DMA 选择 (USBDMASEL) 寄存器，软件选择哪个端点要使用  $\mu$ DMA 通道服务。发送和接收通道分别通过 USBTXCSRn 和 USBRXCSRn 寄存器启用 USB 的  $\mu$ DMA 操作。当  $\mu$ DMA 操作启用，USB 在 FIFO 能传输数据时，在启用的接收或发送通道上发出  $\mu$ DMA 请求。当任一 FIFO 能传输数据时，则该通道发出猝发请求。 $\mu$ DMA 通道必须配置为基本模式， $\mu$ DMA 传输的大小必须限制为 USB FIFO 的整数倍。使用  $\mu$ DMA 的 USB FIFO 的读和写传输都必须这样配置。例如，如果 USB 端点配置 64 字节大小的 FIFO， $\mu$ DMA 通道可以与端点 FIFO 之间进行 64 字节的传输。如果传输的字节数小于 64，必须通过编程的软件 I/O 方式来从 FIFO 复制数据，或复制数据到 FIFO。

如果 USBTXCSRn/ USBRXCSRn 寄存器中的 DMAMOD 位清零， $\mu$ DMA 传输中每个包传输完成后都产生中断信号，但  $\mu$ DMA 会继续传输数据。如果 DMAMOD 位置位，整个  $\mu$ DMA 传输完成后才会产生中断信号。此中断产生到 USB 中断向量。因此，如果 USB 操作使用的中断和  $\mu$ DMA 启用，必须设计 USB 中断处理函数来处理  $\mu$ DMA 完成中断。

使用  $\mu$ DMA 从接收 FIFO 中读数据时需要谨慎，无论 USBRXCSRn 寄存器中的 MAXLOAD 位域的值是多少，每次都从接收 FIFO 中读出数据 4 字节的块。RXRDY 位按下面情况清零：

表 18-2. 余数 (MAXLOAD/4)

值	描述
0	MAXLOAD = 64 字节
1	MAXLOAD = 61 字节
2	MAXLOAD = 62 字节
3	MAXLOAD = 63 字节

表 18-3. 实际读出的字节

值	描述
0	MAXLOAD
1	MAXLOAD+3
2	MAXLOAD+2
3	MAXLOAD+1

表 18-4. 清除 RXRDY 的数据包大小

值	描述
0	MAXLOAD ; MAXLOAD-1 ; MAXLOAD-2 ; MAXLOAD-3
1	MAXLOAD
2	MAXLOAD ; MAXLOAD-1
3	MAXLOAD ; MAXLOAD-1 ; MAXLOAD-2

要启用该端点接收通道的 DMA 操作，应当将 USBRXCSRHn 寄存器中的 DMAEN 位置位。要启用该端点发送通道的 DMA 操作，必须将 USBTXCSRHn 寄存器中的 DMAEN 位置位。

关于如果对  $\mu$ DMA 控制器编程的详细信息，请参考“微型直接存储器访问 ( $\mu$ DMA)”在 534 页。

## 18.4 初始化及配置

要使用 USB 控制器，必须通过 RCGCUSB 寄存器启用其外设时钟（见 313 页）。另外，相应 GPIO 模块的时钟也必须通过系统控制模块中的 RCGCGPIO 寄存器来启用（参见 302 页）。要了解要启用哪一个 GPIO 端口，请参考表 23-4 在 1271 页。配置 GPIOPCTL 寄存器中的 PMCn 域，将 USB 信号指派给相应的管脚（参见 636 页和表 23-5 在 1280 页）。

所有情况下的配置都需要在设置寄存器前，先启用 USB 控制器及其物理层接口 PHY。接下来启用 USB 的 PLL 来给 PHY 提供正确的时钟。为了确保不给总线提供错误的电压，应通过将 USB0EPEN 和 USB0PFLT 管脚配置为由 USB 控制器控制而不显示其默认的 GPIO 行为，在启动时禁止外部电源控制信号 USB0EPEN。

**注意：** 在 OTG 模式，由于 USB0VBUS 和 USB0ID 是 USB 控制器专用的管脚，所以无需配置，可直接连接到 USB 连接器的 VBUS 和 ID 信号。如果 USB 控制器用作专用的主机或设备，USB 通用控制和状态 (USBGPCS) 寄存器中的 DEVMODOTG 和 DEVMOD 位可用于将 USB0VBUS 和 USB0ID 输入连接到内部固定电平，从而释放 PB0 和 PB1 管脚，以用于 GPIO。当用作自供电的设备时，需要监测 VBUS，来确定主机是否断开 VBUS，以及自供电设备是否禁用 D+/D- 上拉电阻。此功能可通过将一个标准 GPIO 连接到 VBUS 实现。

### 18.4.1 管脚配置

当在具有主机功能的系统中使用 USB 控制器的设备部分时，必须禁止向 VBUS 供电，而使用外部主机控制器供电。通常，USB0EPEN 信号用于控制外部稳压器，必须禁止启用外部稳压器，避免两个设备同时驱动 USB 连接器的电源管脚 SB0VBUS。

当 USB 控制器用作主机时，连接到外部电源的两个信号控制向 VBUS 提供电源。主机控制器使用 USB0EPEN 信号来启用或禁止向 USB 连接器的 USB0VBUS 管脚供电。输入管脚 USB0PFLT 用于在 VBUS 上的电源出现异常时提供反馈。可将 USB0PFLT 信号配置为自动禁用 USB0EPEN 信号以禁用电源，也可以通过对中断控制器产生中断信号，以通过软件来处理电源故障状况。USB0EPEN 和 USB0PFLT 的极性以及相关动作全部可以通过 USB 控制器配置。USB 控制器还能为设备的接入和断开提供中断信号，以使主机控制响应这些外部事件。

### 18.4.2 端点配置

在主机或设备模式开始发起通讯之前，必须先配置端点寄存器。在主机模式，端点配置在端点寄存器和设备的端点之间建立连接。在设备模式时，设备枚举之前必须先配置端点。

由于端点 0 具有固定功能和固定 FIFO 大小，所以其配置是受限的。在设备和主机模式时，端点需要很少的设置，但在标准控制传输的建立、数据和状态阶段过程中，需要一个基于软件的状态机。在设备模式中，端点都在设备枚举之前配置完成，只有在主机选择代替配置时才改变。在主机模式中，端点必须配置为控制、批量、中断或等时端点。一旦端点的类型配置完成，必须给每个端点分配 FIFO 内存区域。对于批量传输、控制和中断端点，每个事务最大可传输 64 字节。等时端点则每包最大支持 1023 字节。端点的最大包长必须在发送和接收数据前设置。

配置端点的FIFO时，将全部USB FIFO内存RAM的一部分配置给每个端点。整个 FIFO RAM 为 2 K 字节，前 64 字节为端点 0 保留。端点的 FIFO 的最小大小应与最大的数据包的大小相等。同时也可配置为双包缓存 FIFO，此时每个包传输结束后都将产生中断，并允许填充 FIFO 的另一半。

如果作为设备运行，当设备准备开始通讯时，必须启用软件连接来通知主机已准备好开始枚举。如果作为主机运行，必须禁止设备软件连接，并通过 USB0EPEN 信号给 VBUS 供电。

## 18.5 寄存器映射

表 18-5 在 1038 页列出了寄存器。所有给出的地址都是相对于 0x4005.0000 的 USB 基地址而言的。请注意，在对寄存器编程之前，必须先启用 USB 控制器的时钟（请参考 313 页）。USB 模块时钟启用后，必须等待至少 3 个系统时钟才可访问 USB 模块寄存器。

表 18-5. 通用串行总线 (USB) 控制器 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	USBFADDR	R/W	0x00	USB 设备功能地址寄存器	1045
0x001	USBPOWER	R/W	0x20	USB电源寄存器	1046
0x002	USBTXIS	RO	0x0000	USB发送中断状态寄存器	1049
0x004	USBRXIS	RO	0x0000	USB接收中断状态寄存器	1051
0x006	USBTXIE	R/W	0xFFFF	USB发送中断启用寄存器	1052
0x008	USBRXIE	R/W	0xFFFE	USB接收中断启用寄存器	1053
0x00A	USBIS	RO	0x00	USB通用中断状态寄存器	1054
0x00B	USBIE	R/W	0x06	USB中断启用	1057
0x00C	USBFRAME	RO	0x0000	USB帧值	1060
0x00E	USBEPIDX	R/W	0x00	USB 端点索引寄存器	1061
0x00F	USBTEST	R/W	0x00	USB 测试模式寄存器	1062
0x020	USBFIFO0	R/W	0x0000.0000	USB FIFO 端点 0	1064
0x024	USBFIFO1	R/W	0x0000.0000	USB FIFO 端点 1	1064
0x028	USBFIFO2	R/W	0x0000.0000	USB FIFO 端点 2	1064
0x02C	USBFIFO3	R/W	0x0000.0000	USB FIFO 端点 3	1064
0x030	USBFIFO4	R/W	0x0000.0000	USB FIFO 端点 4	1064
0x034	USBFIFO5	R/W	0x0000.0000	USB FIFO 端点 5	1064
0x038	USBFIFO6	R/W	0x0000.0000	USB FIFO 端点 6	1064
0x03C	USBFIFO7	R/W	0x0000.0000	USB FIFO 端点 7	1064
0x060	USBDEVCTL	R/W	0x80	USB设备控制寄存器	1065
0x062	USBTXFIFOSZ	R/W	0x00	USB 发送动态 FIFO 大小	1067
0x063	USBRXFIFOSZ	R/W	0x00	USB 接收动态 FIFO 大小	1067
0x064	USBTXFIFOADD	R/W	0x0000	USB 发送 FIFO 起始地址	1068
0x066	USBRXFIFOADD	R/W	0x0000	USB 接收 FIFO 起始地址	1068

表 18-5. 通用串行总线 (USB) 控制器 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x07A	USBCONTIM	R/W	0x5C	USB连接时序寄存器	1069
0x07B	USBVPLEN	R/W	0x3C	USB OTG VBUS脉冲时序寄存器	1070
0x07D	USBFSEOF	R/W	0x77	USB 全速模式下最后的传输与帧结束时序寄存器	1071
0x07E	USBLSEOF	R/W	0x0072	USB 低速模式下最后的传输与帧结束时序寄存器	1072
0x080	USBTXFUNCADDR0	R/W	0x00	USB 发送端点 0 功能地址	1073
0x082	USBTXHUBADDR0	R/W	0x00	USB 发送端点 0 集线器地址	1074
0x083	USBTXHUBPORT0	R/W	0x00	USB 发送端点 0 集线器端口	1075
0x088	USBTXFUNCADDR1	R/W	0x00	USB 发送端点 1 功能地址	1073
0x08A	USBTXHUBADDR1	R/W	0x00	USB 发送端点 1 集线器地址	1074
0x08B	USBTXHUBPORT1	R/W	0x00	USB 发送端点 1 集线器端口	1075
0x08C	USBRXFUNCADDR1	R/W	0x00	USB 接收端点 1 功能地址	1076
0x08E	USBRXHUBADDR1	R/W	0x00	USB 接收端点 1 集线器地址	1077
0x08F	USBRXHUBPORT1	R/W	0x00	USB 接收端点 1 集线器端口	1078
0x090	USBTXFUNCADDR2	R/W	0x00	USB 发送端点 2 功能地址	1073
0x092	USBTXHUBADDR2	R/W	0x00	USB 发送端点 2 集线器地址	1074
0x093	USBTXHUBPORT2	R/W	0x00	USB 发送端点 2 集线器端口	1075
0x094	USBRXFUNCADDR2	R/W	0x00	USB 接收端点 2 功能地址	1076
0x096	USBRXHUBADDR2	R/W	0x00	USB 接收端点 2 集线器地址	1077
0x097	USBRXHUBPORT2	R/W	0x00	USB 接收端点 2 集线器端口	1078
0x098	USBTXFUNCADDR3	R/W	0x00	USB 发送端点 3 功能地址	1073
0x09A	USBTXHUBADDR3	R/W	0x00	USB 发送端点 3 集线器地址	1074
0x09B	USBTXHUBPORT3	R/W	0x00	USB 发送端点 3 集线器端口	1075
0x09C	USBRXFUNCADDR3	R/W	0x00	USB 接收端点 3 功能地址	1076
0x09E	USBRXHUBADDR3	R/W	0x00	USB 接收端点 3 集线器地址	1077
0x09F	USBRXHUBPORT3	R/W	0x00	USB 接收端点 3 集线器端口	1078
0x0A0	USBTXFUNCADDR4	R/W	0x00	USB 发送端点 4 功能地址	1073
0x0A2	USBTXHUBADDR4	R/W	0x00	USB 发送端点 4 集线器地址	1074
0x0A3	USBTXHUBPORT4	R/W	0x00	USB 发送端点 4 集线器端口	1075
0x0A4	USBRXFUNCADDR4	R/W	0x00	USB 接收端点 4 功能地址	1076
0x0A6	USBRXHUBADDR4	R/W	0x00	USB 接收端点 4 集线器地址	1077
0x0A7	USBRXHUBPORT4	R/W	0x00	USB 接收端点 4 集线器端口	1078
0x0A8	USBTXFUNCADDR5	R/W	0x00	USB 发送端点 5 功能地址	1073
0x0AA	USBTXHUBADDR5	R/W	0x00	USB 发送端点 5 集线器地址	1074

表 18-5. 通用串行总线 (USB) 控制器 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x0AB	USBTXHUBPORT5	R/W	0x00	USB 发送端点 5 集线器端口	1075
0x0AC	USBRXFUNCADDR5	R/W	0x00	USB 接收端点 5 功能地址	1076
0x0AE	USBRXHUBADDR5	R/W	0x00	USB 接收端点 5 集线器地址	1077
0x0AF	USBRXHUBPORT5	R/W	0x00	USB 接收端点 5 集线器端口	1078
0x0B0	USBTXFUNCADDR6	R/W	0x00	USB 发送端点 6 功能地址	1073
0x0B2	USBTXHUBADDR6	R/W	0x00	USB 发送端点 6 集线器地址	1074
0x0B3	USBTXHUBPORT6	R/W	0x00	USB 发送端点 6 集线器端口	1075
0x0B4	USBRXFUNCADDR6	R/W	0x00	USB 接收端点 6 功能地址	1076
0x0B6	USBRXHUBADDR6	R/W	0x00	USB 接收端点 6 集线器地址	1077
0x0B7	USBRXHUBPORT6	R/W	0x00	USB 接收端点 6 集线器端口	1078
0x0B8	USBTXFUNCADDR7	R/W	0x00	USB 发送端点 7 功能地址	1073
0x0BA	USBTXHUBADDR7	R/W	0x00	USB 发送端点 7 集线器地址	1074
0x0BB	USBTXHUBPORT7	R/W	0x00	USB 发送端点 7 集线器端口	1075
0x0BC	USBRXFUNCADDR7	R/W	0x00	USB 接收端点 7 功能地址	1076
0x0BE	USBRXHUBADDR7	R/W	0x00	USB 接收端点 7 集线器地址	1077
0x0BF	USBRXHUBPORT7	R/W	0x00	USB 接收端点 7 集线器端口	1078
0x102	USBCSRL0	W1C	0x00	USB 端点 0 控制和状态低字节寄存器	1080
0x103	USBCSRH0	W1C	0x00	USB 端点 0 控制和状态高字节寄存器	1084
0x108	USBCOUNT0	RO	0x00	USB 端点 0 接收字节数量寄存器	1086
0x10A	USBTTYPE0	R/W	0x00	USB 端点 0 类型寄存器	1087
0x10B	USBNAKLMT	R/W	0x00	USB NAK 限制寄存器	1088
0x110	USBTXMAXP1	R/W	0x0000	USB 发送端点 1 最大传输数据	1079
0x112	USBTXCSRL1	R/W	0x00	USB 端点 1 发送控制和状态低字节	1089
0x113	USBTXCSRH1	R/W	0x00	USB 发送端点 1 控制和状态高字节	1093
0x114	USBRXMAXP1	R/W	0x0000	USB 接收端点 1 最大传输数据	1097
0x116	USBRXCSRL1	R/W	0x00	USB 接收端点 1 控制和状态低字节	1098
0x117	USBRXCSRH1	R/W	0x00	USB 接收端点 1 控制和状态高字节	1102
0x118	USBRXCOUNT1	RO	0x0000	USB 接收端点 1 字节计数	1106
0x11A	USBTXTYPE1	R/W	0x00	USB 端点 1 主机发送配置类型	1107
0x11B	USBTXINTERVAL1	R/W	0x00	USB 端点 1 主机发送间隔	1109
0x11C	USBRXTYPE1	R/W	0x00	USB 端点 1 主机配置接收类型	1110
0x11D	USBRXINTERVAL1	R/W	0x00	USB 端点 1 主机接收轮询间隔	1112
0x120	USBTXMAXP2	R/W	0x0000	USB 发送端点 2 最大传输数据	1079



表 18-5. 通用串行总线 (USB) 控制器 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x122	USBTXCSSL2	R/W	0x00	USB 端点 2 发送控制和状态低字节	1089
0x123	USBTXCSRH2	R/W	0x00	USB 发送端点 2 控制和状态高字节	1093
0x124	USBRXMAXP2	R/W	0x0000	USB 接收端点 2 最大传输数据	1097
0x126	USBRXCSSL2	R/W	0x00	USB 接收端点 2 控制和状态低字节	1098
0x127	USBRXCSRH2	R/W	0x00	USB 接收端点 2 控制和状态高字节	1102
0x128	USBRXCOUNT2	RO	0x0000	USB 接收端点 2 字节计数	1106
0x12A	USBTXTYPE2	R/W	0x00	USB 端点 2 主机发送配置类型	1107
0x12B	USBTXINTERVAL2	R/W	0x00	USB 端点 2 主机发送间隔	1109
0x12C	USBRXTYPE2	R/W	0x00	USB 端点 2 主机配置接收类型	1110
0x12D	USBRXINTERVAL2	R/W	0x00	USB 端点 2 主机接收轮询间隔	1112
0x130	USBTXMAXP3	R/W	0x0000	USB 发送端点 3 最大传输数据	1079
0x132	USBTXCSSL3	R/W	0x00	USB 端点 3 发送控制和状态低字节	1089
0x133	USBTXCSRH3	R/W	0x00	USB 发送端点 3 控制和状态高字节	1093
0x134	USBRXMAXP3	R/W	0x0000	USB 接收端点 3 最大传输数据	1097
0x136	USBRXCSSL3	R/W	0x00	USB 接收端点 3 控制和状态低字节	1098
0x137	USBRXCSRH3	R/W	0x00	USB 接收端点 3 控制和状态高字节	1102
0x138	USBRXCOUNT3	RO	0x0000	USB 接收端点 3 字节计数	1106
0x13A	USBTXTYPE3	R/W	0x00	USB 端点 3 主机发送配置类型	1107
0x13B	USBTXINTERVAL3	R/W	0x00	USB 端点 3 主机发送间隔	1109
0x13C	USBRXTYPE3	R/W	0x00	USB 端点 3 主机配置接收类型	1110
0x13D	USBRXINTERVAL3	R/W	0x00	USB 端点 3 主机接收轮询间隔	1112
0x140	USBTXMAXP4	R/W	0x0000	USB 发送端点 4 最大传输数据	1079
0x142	USBTXCSSL4	R/W	0x00	USB 端点 4 发送控制和状态低字节	1089
0x143	USBTXCSRH4	R/W	0x00	USB 发送端点 4 控制和状态高字节	1093
0x144	USBRXMAXP4	R/W	0x0000	USB 接收端点 4 最大传输数据	1097
0x146	USBRXCSSL4	R/W	0x00	USB 接收端点 4 控制和状态低字节	1098
0x147	USBRXCSRH4	R/W	0x00	USB 接收端点 4 控制和状态高字节	1102
0x148	USBRXCOUNT4	RO	0x0000	USB 接收端点 4 字节计数	1106
0x14A	USBTXTYPE4	R/W	0x00	USB 端点 4 主机发送配置类型	1107
0x14B	USBTXINTERVAL4	R/W	0x00	USB 端点 4 主机发送间隔	1109
0x14C	USBRXTYPE4	R/W	0x00	USB 端点 4 主机配置接收类型	1110
0x14D	USBRXINTERVAL4	R/W	0x00	USB 端点 4 主机接收轮询间隔	1112
0x150	USBTXMAXP5	R/W	0x0000	USB 发送端点 5 最大传输数据	1079

表 18-5. 通用串行总线 (USB) 控制器 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x152	USBTXCSSL5	R/W	0x00	USB 端点 5 发送控制和状态低字节	1089
0x153	USBTXCSRH5	R/W	0x00	USB 发送端点 5 控制和状态高字节	1093
0x154	USBRXMAXP5	R/W	0x0000	USB 接收端点 5 最大传输数据	1097
0x156	USBRXCSSL5	R/W	0x00	USB 接收端点 5 控制和状态低字节	1098
0x157	USBRXCSRH5	R/W	0x00	USB 接收端点 5 控制和状态高字节	1102
0x158	USBRXCOUNT5	RO	0x0000	USB 接收端点 5 字节计数	1106
0x15A	USBTXTYPE5	R/W	0x00	USB 端点 5 主机发送配置类型	1107
0x15B	USBTXINTERVAL5	R/W	0x00	USB 端点 5 主机发送间隔	1109
0x15C	USBRXTYPE5	R/W	0x00	USB 端点 5 主机配置接收类型	1110
0x15D	USBRXINTERVAL5	R/W	0x00	USB 端点 5 主机接收轮询间隔	1112
0x160	USBTXMAXP6	R/W	0x0000	USB 发送端点 6 最大传输数据	1079
0x162	USBTXCSSL6	R/W	0x00	USB 端点 6 发送控制和状态低字节	1089
0x163	USBTXCSRH6	R/W	0x00	USB 发送端点 6 控制和状态高字节	1093
0x164	USBRXMAXP6	R/W	0x0000	USB 接收端点 6 最大传输数据	1097
0x166	USBRXCSSL6	R/W	0x00	USB 接收端点 6 控制和状态低字节	1098
0x167	USBRXCSRH6	R/W	0x00	USB 接收端点 6 控制和状态高字节	1102
0x168	USBRXCOUNT6	RO	0x0000	USB 接收端点 6 字节计数	1106
0x16A	USBTXTYPE6	R/W	0x00	USB 端点 6 主机发送配置类型	1107
0x16B	USBTXINTERVAL6	R/W	0x00	USB 端点 6 主机发送间隔	1109
0x16C	USBRXTYPE6	R/W	0x00	USB 端点 6 主机配置接收类型	1110
0x16D	USBRXINTERVAL6	R/W	0x00	USB 端点 6 主机接收轮询间隔	1112
0x170	USBTXMAXP7	R/W	0x0000	USB 发送端点 7 最大传输数据	1079
0x172	USBTXCSSL7)	R/W	0x00	USB 端点 7 发送控制和状态低字节	1089
0x173	USBTXCSRH7	R/W	0x00	USB 发送端点 7 控制和状态高字节	1093
0x174	USBRXMAXP7	R/W	0x0000	USB 接收端点 7 最大传输数据	1097
0x176	USBRXCSSL7	R/W	0x00	USB 接收端点 7 控制和状态低字节	1098
0x177	USBRXCSRH7	R/W	0x00	USB 接收端点 7 控制和状态高字节	1102
0x178	USBRXCOUNT7	RO	0x0000	USB 接收端点 7 字节计数	1106
0x17A	USBTXTYPE7	R/W	0x00	USB 端点 7 主机发送配置类型	1107
0x17B	USBTXINTERVAL7	R/W	0x00	USB 端点 7 主机发送间隔	1109
0x17C	USBRXTYPE7	R/W	0x00	USB 端点 7 主机配置接收类型	1110
0x17D	USBRXINTERVAL7	R/W	0x00	USB 端点 7 主机接收轮询间隔	1112
0x304	USBRQPKTCOUNT1	R/W	0x0000	USB 端点 1 块传输中请求包数量	1113

表 18-5. 通用串行总线 (USB) 控制器 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x308	USBRQPKTCOUNT2	R/W	0x0000	USB 端点 2 块传输中请求包数量	1113
0x30C	USBRQPKTCOUNT3	R/W	0x0000	USB 端点 3 块传输中请求包数量	1113
0x310	USBRQPKTCOUNT4	R/W	0x0000	USB 端点 4 块传输中请求包数量	1113
0x314	USBRQPKTCOUNT5	R/W	0x0000	USB 端点 5 块传输中请求包数量	1113
0x318	USBRQPKTCOUNT6	R/W	0x0000	USB 端点 6 块传输中请求包数量	1113
0x31C	USBRQPKTCOUNT7	R/W	0x0000	USB 端点 7 块传输中请求包数量	1113
0x340	USBRXDPKTBUFFDIS	R/W	0x0000	USB 接收双包缓存禁用寄存器	1114
0x342	USBTXDPKTBUFFDIS	R/W	0x0000	USB 发送双包缓存禁用寄存器	1115
0x400	USBEPCC	R/W	0x0000.0000	USB 外部电源控制寄存器	1116
0x404	USBEPCCRIS	RO	0x0000.0000	USB 外部电源控制原始中断状态寄存器	1119
0x408	USBEPCCIM	R/W	0x0000.0000	USB 外部电源控制中断屏蔽寄存器	1120
0x40C	USBEPCCISC	R/W	0x0000.0000	USB 外部电源控制中断状态和清除寄存器	1121
0x410	USBDRRIS	RO	0x0000.0000	USB 设备恢复原始中断状态寄存器	1122
0x414	USBDRIM	R/W	0x0000.0000	USB 设备恢复中断屏蔽寄存器	1123
0x418	USBDRISC	W1C	0x0000.0000	USB 设备恢复中断状态和清除寄存器	1124
0x41C	USBGPCS	R/W	0x0000.0003	USB 通用控制和状态寄存器	1125
0x430	USBVDC	R/W	0x0000.0000	USB VBUS 浮动控制寄存器	1126
0x434	USBVDCRIS	RO	0x0000.0000	USB VBUS 浮动控制原始中断状态寄存器	1127
0x438	USBVDCIM	R/W	0x0000.0000	USB VBUS 浮动控制中断屏蔽寄存器	1128
0x43C	USBVDCISC	R/W	0x0000.0000	USB VBUS 浮动控制中断状态和清除寄存器	1129
0x444	USBIDVRIS	RO	0x0000.0000	USB ID 有效检测原始中断状态寄存器	1130
0x448	USBIDVIM	R/W	0x0000.0000	USB ID 有效检测中断屏蔽寄存器	1131
0x44C	USBIDVISIC	R/W1C	0x0000.0000	USB ID 有效检测中断状态与清除寄存器	1132
0x450	USBDMASEL	R/W	0x0033.2211	USB DMA 选择寄存器	1133
0xFC0	USBPP	RO	0x0000.10D0	USB 外设属性寄存器	1135

## 18.6 寄存器描述

如 DC6 寄存器的 USB0 位域中所指定的，LM4F232H5QD USB 控制器可用作 On-The-Go (OTG) (请参考 428 页)。

**OTG B /  
设备**

此图标指示此寄存器用于 OTGB 或设备模式。有些寄存器同时用于主机和设备模式，依据不同模式可能有不同的位定义。

**OTG A /**  
主机

此图标指示此寄存器用于 OTGA 或主机模式。有些寄存器同时用于主机和设备模式，依据不同模式可能有不同的位定义。复位时，USB 控制器将处于 OTG B 或设备模式，因此这些寄存器复位值将用于设备模式定义。

**OTG**

此图标指示此寄存器用于 OTG 的特定功能，比如 ID 探测和协商。一旦 OTG 协商完成，USB 控制器的寄存器按照 Host 主机模式使用还是按照 Device 设备模式使用，决定于 OTG 协商使 USB 工作在 OTG A ( Host ) 还是 OTG B ( Device )。

**寄存器 1: USB 设备功能地址寄存器 ( USBFADDR ) , 偏移量 0x000**

**OTG B /  
设备**

USBFADDR 是 8 位寄存器，其中 7 位用作此次传输的设备地址部分。

当 USB 控制器用于设备模式时 ( USBDEVCTL 寄存器中的 HOST 位清零 )，将通过 SET\_ADDRESS 命令收到的地址写入该寄存器，用于随后的令牌包功能地址解码。

**重要:** 当向该寄存器写值时所要注意到特殊需求，参看“设置设备地址”一节在 1031 页。

**USB 设备功能地址寄存器 (USBFADDR)**

基址 0x4005.0000  
偏移量 0x000  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										FUNCADDR					
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
6:0	FUNCADDR	R/W	0x00	功能地址 设备功能地址通过 SET_ADDRESS 命令接收。

## 寄存器 2: USB电源寄存器 ( USBPOWER ) , 偏移量 0x001

OTG A /  
主机

USBPOWER 是一个 8 位寄存器 , 用于控制挂起和恢复信号以及 USB 控制器的一些基本操作。

OTG B /  
设备

## OTG A/主机模式

## USB电源寄存器 (USBPOWER)

基址 0x4005.0000

偏移量 0x001

类型 R/W, 复位 0x20

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												RESET	RESUME	SUSPEND	PWRDNPHY
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W1S	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件,保留位的值在读-修改-写操作过程中应该保持不变。
3	RESET	R/W	0	RESET复位信号  值 描述 1 启用总线上的RESET复位信号。 0 终止总线上的RESET复位信号。
2	RESUME	R/W	0	RESUME 恢复信号  值 描述 1 当设备处于挂起 (SUSPEND) 状态,启用总线上的恢复 (RESUME) 信号。 0 终止总线上的恢复 (RESUME) 信号。  此位必须在置位 20 ms 后才能被软件清零。
1	SUSPEND	R/W1S	0	SUSPEND 挂起模式  值 描述 1 启用挂起 (SUSPEND) 模式。 0 无影响。

位/域	名称	类型	复位	描述
0	PWRDNPHY	R/W	0	PHY 掉电

## 值 描述

- 1 使内部USB PHY掉电。
- 0 无影响。

## OTG B/设备模式

## USB电源寄存器 (USBPOWER)

基址 0x4005.0000

偏移量 0x001

类型 R/W, 复位 0x20

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								ISOUP	SOFTCONN	保留		RESET	RESUME	SUSPEND	PWRDNPHY
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	R/W	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	ISOUP	R/W	0	等时更新  值 描述 1 USB 控制器在发送包之前，USBTXCSSLn 寄存器的 TXRDY 位置位时，等待 SOF 令牌包。如果在 SOF 令牌之前收到输入令牌，将发送 0 长度的数据包。 0 无影响。  注意： 此位只对等时传输有效。
6	SOFTCONN	R/W	0	软件连接/断开  值 描述 1 启用USB D+/D-。 0 USB D+/D-处于三态。
5:4	保留	RO	0x2	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	RESET	RO	0	RESET复位信号  值 描述 1 启用总线上的RESET复位信号。 0 终止总线上的RESET复位信号。

位/域	名称	类型	复位	描述
2	RESUME	R/W	0	RESUME 恢复信号  值 描述 1 当设备处于挂起 (SUSPEND) 状态，启用总线上的恢复 (RESUME) 信号。 0 终止总线上的恢复 (RESUME) 信号。  此位必须在置位 10 ms ( 最大 15 ms ) 后才能被软件清除。
1	SUSPEND	RO	0	SUSPEND 挂起模式  值 描述 1 USB 控制器工作在挂起模式。 0 当软件读中断寄存器或置位上面的 RESUME 位时，此位清零。
0	PWRDNPHY	R/W	0	PHY 掉电  值 描述 1 使内部USB PHY掉电。 0 无影响。



### 寄存器 3: USB发送中断状态寄存器 ( USBTXIS ) , 偏移量 0x002

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

OTG A /  
主机

USBTXIS 是一个 16 位的只读寄存器, 用于指示端点 0 和发送端点 1-7 的哪个中断当前是有效的。寄存器中 EPn 位域的含意取决于设备的模式。EP1 到 EP7 位始终指示 USB 控制器正在发送数据; 在主机模式下, 这些位适用于输出端点; 而在设备模式下, 这些位适用于输入端点。EP0 位比较特殊, 在主机和设备模式中, 该位指示控制输入或控制输出端点是否产生了中断。

OTG B /  
设备

**注意:** 与这些端点相关的位如果没有配置, 其返回值一直为 0。注: 对该寄存器的读操作, 会清除所有已激活的中断。

USB发送中断状态寄存器 (USBTXIS)

基址 0x4005.0000  
偏移量 0x002  
类型 RO, 复位 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	EP7	RO	0	发送端点7中断  值 描述 0 无中断。 1 端点 7 的发送中断有效。
6	EP6	RO	0	发送端点6中断 与 EP7 描述相同。
5	EP5	RO	0	发送端点5中断 与 EP7 描述相同。
4	EP4	RO	0	发送端点4中断 与 EP7 描述相同。
3	EP3	RO	0	发送端点3中断 与 EP7 描述相同。
2	EP2	RO	0	发送端点2中断 与 EP7 描述相同。
1	EP1	RO	0	发送端点1中断 与 EP7 描述相同。

位/域	名称	类型	复位	描述
0	EPO	RO	0	发送和接收端点 0 中断 值 描述 0 无中断。 1 端点 0 的发送和接收中断有效。

## 寄存器 4: USB接收中断状态寄存器 ( USBRXIS ) , 偏移量 0x004

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

OTG A /  
主机

USBRXIS 是一个 16 位的只读寄存器, 用于指示接收端点 1-7 的哪个中断当前是有效的。

**注意:** 与这些端点相关的位如果没有配置, 其返回值一直为 0。注: 对该寄存器的读操作, 会清除所有已激活的中断。

OTG B /  
设备

USB接收中断状态寄存器 (USBRXIS)

基址 0x4005.0000

偏移量 0x004

类型 RO, 复位 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								EP7	EP6	EP5	EP4	EP3	EP2	EP1	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	EP7	RO	0	接收端点 7 中断  值 描述 0 无中断。 1 端点 7 的发送中断有效。
6	EP6	RO	0	接收端点 6 中断 与 EP7 描述相同。
5	EP5	RO	0	接收端点 5 中断 与 EP7 描述相同。
4	EP4	RO	0	接收端点 4 中断 与 EP7 描述相同。
3	EP3	RO	0	接收端点 3 中断 与 EP7 描述相同。
2	EP2	RO	0	接收端点 2 中断 与 EP7 描述相同。
1	EP1	RO	0	接收端点 1 中断 与 EP7 描述相同。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 5: USB发送中断启用寄存器 ( USBTXIE ) , 偏移量 0x006

OTG A /  
主机

USBTXIE 是 16 位寄存器，用于为 USBTXIS 寄存器的中断提供中断启用位。如果某位置位，USBTXIS 寄存器中相应的中断位也置 1，则 USB 中断对中断控制器有效。如果某位清零，USBTXIS 寄存器中的中断位仍然置 1，但 USB 中断对中断控制器无效。复位时，所有的中断是启用的。

OTG B /  
设备

USB发送中断启用寄存器 (USBTXIE)

基址 0x4005.0000

偏移量 0x006

类型 R/W, 复位 0xFFFF

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	EP7	R/W	1	发送端点7中断启用  值 描述 1 当 USBTXIS 寄存器中的 EP7 位置位时，向中断控制器发送中断。 0 EP7 发送中断被抑制，不发送到中断控制器。
6	EP6	R/W	1	发送端点6中断启用 与 EP7 描述相同。
5	EP5	R/W	1	发送端点5中断启用 与 EP7 描述相同。
4	EP4	R/W	1	发送端点4中断启用 与 EP7 描述相同。
3	EP3	R/W	1	发送端点3中断启用 与 EP7 描述相同。
2	EP2	R/W	1	发送端点2中断启用 与 EP7 描述相同。
1	EP1	R/W	1	发送端点1中断启用 与 EP7 描述相同。
0	EP0	R/W	1	发送和接收端点 0 中断启用  值 描述 1 当 USBTXIS 寄存器中的 EP0 位置位时，向中断控制器发送中断。 0 EP0 发送和接收中断被抑制，不发送到中断控制器。

## 寄存器 6: USB接收中断启用寄存器 ( USBRXIE ) , 偏移量 0x008

OTG A /  
主机

USBRXIE 是 16 位寄存器, 用于为 USBRXIS 寄存器的中断提供中断启用位。如果某位置位, USBRXIS 寄存器中相应的中断位也置 1, 则 USB 中断对中断控制器有效。如果某位清零, USBRXIS 寄存器中的中断位仍然置 1, 但 USB 中断对中断控制器无效。复位时, 所有的中断是启用的。

OTG B /  
设备

USB接收中断启用寄存器 (USBRXIE)

基址 0x4005.0000

偏移量 0x008

类型 R/W, 复位 0xFFFE

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								EP7	EP6	EP5	EP4	EP3	EP2	EP1	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	EP7	R/W	1	接收端点 7 中断启用  值 描述 1 当 USBRXIS 寄存器中的 EP7 位置位时, 向中断控制器发送中断。 0 EP7 接收中断被抑制, 不发送到中断控制器。
6	EP6	R/W	1	接收端点 6 中断启用 与 EP7 描述相同。
5	EP5	R/W	1	接收端点 5 中断启用 与 EP7 描述相同。
4	EP4	R/W	1	接收端点 4 中断启用 与 EP7 描述相同。
3	EP3	R/W	1	接收端点 3 中断启用 与 EP7 描述相同。
2	EP2	R/W	1	接收端点 2 中断启用 与 EP7 描述相同。
1	EP1	R/W	1	接收端点 1 中断启用 与 EP7 描述相同。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 7: USB通用中断状态寄存器 (USBIS), 偏移量 0x00A

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

OTG A /  
主机

USBIS 是一个 8 位只读寄存器, 用来指示哪个 USB 中断当前是有效的。读此寄存器, 所有的中断将被清除。

OTG B /  
设备

## OTG A/主机模式

## USB通用中断状态寄存器 (USBIS)

基址 0x4005.0000  
偏移量 0x00A  
类型 RO, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								VBUSERR	SESREQ	DISCON	CONN	SOF	BABBLE	RESUME	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	VBUSERR	RO	0	VBUS错误  值 描述 1 会话期间, VBUS掉到有效电压门限以下。 0 无中断。
6	SESREQ	RO	0	会话请求  值 描述 1 检测到会话请求信号。 0 无中断。
5	DISCON	RO	0	会话断开  值 描述 1 检测到设备断开。 0 无中断。

位/域	名称	类型	复位	描述
4	CONN	RO	0	会话连接  值 描述 1 检测到设备连接。 0 无中断。
3	SOF	RO	0	帧起始  值 描述 1 新的帧开始。 0 无中断。
2	BABBLE	RO	0	检测到超时干扰(Babble)  值 描述 1 检测到超时干扰。此中断只有在第一个 SOF 令牌发送后才有效。 0 无中断。
1	RESUME	RO	0	检测到恢复 (RESUME) 信号  值 描述 1 USB 控制器处于挂起模式时，在总线上检测到恢复 (RESUME) 信号。 0 无中断。  此中断再能在USB控制器的系统时钟启用时使用。如果用户禁止时钟编程，应该使用 USBDRRIS、USBDRIM 和 USBDRISC 寄存器。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

## OTG B/设备模式

### USB通用中断状态寄存器 (USBIS)

基址 0x4005.0000  
偏移量 0x00A  
类型 RO, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											DISCON	保留	SOF	RESET	RESUME	SUSPEND
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。

位/域	名称	类型	复位	描述
5	DISCON	RO	0	会话断开  值 描述 1 设备已从主机断开。 0 无中断。
4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	SOF	RO	0	帧起始  值 描述 1 新的帧开始。 0 无中断。
2	RESET	RO	0	检测到复位信号  值 描述 1 在USB总线上检测到复位信号。 0 无中断。
1	RESUME	RO	0	检测到恢复 (RESUME) 信号  值 描述 1 USB 控制器处于挂起模式时，在总线上检测到恢复 (RESUME) 信号。 0 无中断。  此中断再能在USB控制器的系统时钟启用时使用。如果用户禁止时钟编程，应该使用 USBDRRIS、USBDRIM 和 USBDRISC 寄存器。
0	SUSPEND	RO	0	检测到挂起 (SUSPEND) 信号  值 描述 1 在总线上检测到挂起信号。 0 无中断。



## 寄存器 8: USB中断启用 ( USBIE ) , 偏移量 0x00B

OTG A /  
主机

USBIE是8位寄存器,为USBIS寄存器中的每个中断提供中断启用位。设备模式中,复位时,中断1和中断2是启用的。

OTG B /  
设备

## OTG A/主机模式

## USB中断启用 (USBIE)

基址 0x4005.0000

偏移量 0x00B

类型 R/W, 复位 0x06

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								VBUSERR	SESREQ	DISCON	CONN	SOF	BABBLE	RESUME	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

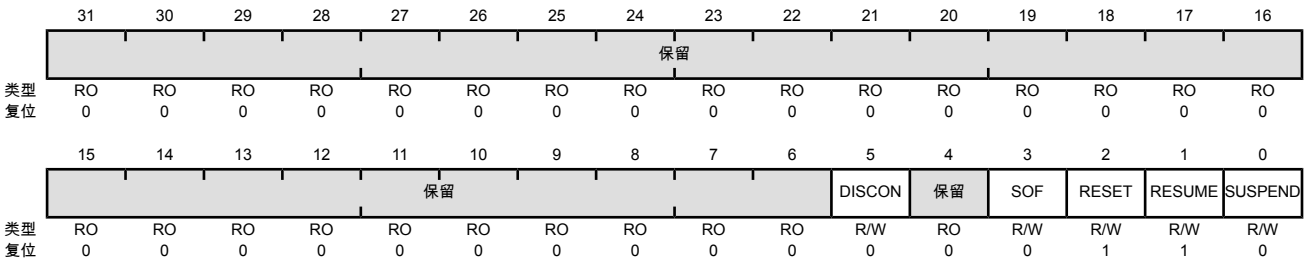
位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件,保留位的值在读-修改-写操作过程中应该保持不变。
7	VBUSERR	R/W	0	启用VBUS错误中断  值 描述 1 当 USBIS 寄存器中的 VBUSERR 位置位时,向中断控制器发送中断。 0 VBUSERR 中断被抑制,不发送到中断控制器。
6	SESREQ	R/W	0	启用会话请求  值 描述 1 当 USBIS 寄存器中的 SESREQ 位置位时,向中断控制器发送中断。 0 SESREQ 中断被抑制,不发送到中断控制器。
5	DISCON	R/W	0	启用断开中断  值 描述 1 当 USBIS 寄存器中的 DISCON 位置位时,向中断控制器发送中断。 0 DISCON 中断被抑制,不发送到中断控制器。

位/域	名称	类型	复位	描述
4	CONN	R/W	0	启用连接中断  值 描述 1 当 USBIS 寄存器中的 CONN 位置位时，向中断控制器发送中断。 0 CONN 中断被抑制，不发送到中断控制器。
3	SOF	R/W	0	启用帧起始中断  值 描述 1 当 USBIS 寄存器中的 SOF 位置位时，向中断控制器发送中断。 0 SOF 中断被抑制，不发送到中断控制器。
2	BABBLE	R/W	1	启用超时干扰中断  值 描述 1 当 USBIS 寄存器中的 BABBLE 位置位时，向中断控制器发送中断。 0 BABBLE 中断被抑制，不发送到中断控制器。
1	RESUME	R/W	1	启用恢复中断  值 描述 1 当 USBIS 寄存器中的 RESUME 位置位时，向中断控制器发送中断。 0 RESUME 中断被抑制，不发送到中断控制器。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

**OTG B/设备模式**

**USB中断启用 (USBIE)**

基址 0x4005.0000  
 偏移量 0x00B  
 类型 R/W, 复位 0x06



位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读取-修改-写入操作过程中应该保持不变。

位/域	名称	类型	复位	描述
5	DISCON	R/W	0	<p>启用断开中断</p> <p>值 描述</p> <p>1 当 USBIS 寄存器中的 DISCON 位置位时，向中断控制器发送中断。</p> <p>0 DISCON 中断被抑制，不发送到中断控制器。</p>
4	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
3	SOF	R/W	0	<p>启用帧起始中断</p> <p>值 描述</p> <p>1 当 USBIS 寄存器中的 SOF 位置位时，向中断控制器发送中断。</p> <p>0 SOF 中断被抑制，不发送到中断控制器。</p>
2	RESET	R/W	1	<p>启用复位中断</p> <p>值 描述</p> <p>1 当 USBIS 寄存器中的 RESET 位置位时，向中断控制器发送中断。</p> <p>0 RESET 中断被抑制，不发送到中断控制器。</p>
1	RESUME	R/W	1	<p>启用恢复中断</p> <p>值 描述</p> <p>1 当 USBIS 寄存器中的 RESUME 位置位时，向中断控制器发送中断。</p> <p>0 RESUME 中断被抑制，不发送到中断控制器。</p>
0	SUSPEND	R/W	0	<p>启用挂起中断</p> <p>值 描述</p> <p>1 当 USBIS 寄存器中的 SUSPEND 位置位时，向中断控制器发送中断。</p> <p>0 SUSPEND 中断被抑制，不发送到中断控制器。</p>

### 寄存器 9: USB 帧值 ( USBFRAME ) ， 偏移量 0x00C

OTG A /  
主机

USBFRAME 是 16 位只读寄存器，用于保存最后接收到的帧编号。

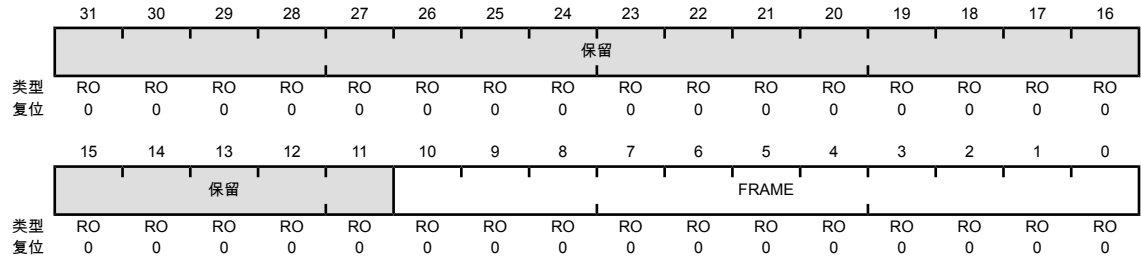
OTG B /  
设备

USB 帧值 (USBFRAME)

基址 0x4005.0000

偏移量 0x00C

类型 RO, 复位 0x0000



位/域	名称	类型	复位	描述
31:11	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
10:0	FRAME	RO	0x000	帧编号

**寄存器 10: USB 端点索引寄存器 ( USBEPIDX ) , 偏移量 0x00E**OTG A /  
主机

端点的缓冲区可以通过设置FIFO大小和起始地址进行访问。USBEPIDX 8 位寄存器 与寄存器 USBTXFIFOSZ、USBRXFIFOSZ、USBTXFIFOADD 以及 USBRXFIFOADD 配合使用。

OTG B /  
设备

USB 端点索引寄存器 (USBEPIDX)

基址 0x4005.0000  
偏移量 0x00E  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												EPIDX			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
3:0	EPIDX	R/W	0x0	端点索引 当读写USB控制器的采用索引方式的寄存器时，此位域设置要访问的端点号。0x0 对应端点 0；0x7 对应端点 7。

## 寄存器 11: USB 测试模式寄存器 ( USBTEST ) , 偏移量 0x00F

OTG A /  
主机

USBTEST 是 8 位寄存器。执行 SET FEATURE:USBTESTMODE 命令后，USB 控制器根据该寄存器的值选择“USB 2.0 规范”中所描述的四种测试模式中的一种模式：USBTESTMODE 命令。此寄存器不用于正常操作。

OTG B /  
设备

注意： 任何时候，这些位中只能有一位被设置。

## OTG A/主机模式

## USB 测试模式寄存器 (USBTEST)

基址 0x4005.0000  
偏移量 0x00F  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								FORCEH	FIFOACC	FORCEFS	保留				
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W1S	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	FORCEH	R/W	0	强制主机模式  值 描述 1 无论 USB 控制器是否连接到外设，当将 SESSION 位置位时，强制 USB 控制器进入主机模式。此时忽略 USB0DP 和 USB0DM 信号的状态。如果不将 SESSION 位清零，即使在与设备断开连接时，该 USB 控制器仍保持主机模式。如果 FORCEH 位保持置位，则 SESSION 位下次置位时，USB 控制器重新进入主机模式。 0 无影响。  处于此模式时，可以通过 USBDEVCTL 寄存器的 DEV 位来读取总线的连接状态。运行速度取决于 FORCEFS 位。
6	FIFOACC	R/W1S	0	FIFO访问  值 描述 1 将端点0发送FIFO中的包传送到端点0的接收FIFO中。 0 无影响。  此位自动清零。

位/域	名称	类型	复位	描述
5	FORCEFS	R/W	0	强制全速模式  值 描述 1 接收到USB复位信号后，强制USB控制器运行在全速模式。 0 USB控制器运行在低速模式。
4:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

## OTG B/设备模式

### USB 测试模式寄存器 (USBTEST)

基址 0x4005.0000

偏移量 0x00F

类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留											FIFOACC	FORCEFS	保留		
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1S	R/W	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
6	FIFOACC	R/W1S	0	FIFO访问  值 描述 1 将端点0发送FIFO中的包传送到端点0的接收FIFO中。 0 无影响。  此位自动清零。
5	FORCEFS	R/W	0	强制全速模式  值 描述 1 接收到USB复位信号后，强制USB控制器运行在全速模式。 0 USB控制器运行在低速模式。
4:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

寄存器 12: USB FIFO 端点 0 ( USBFIFO0 ) , 偏移量 0x020

寄存器 13: USB FIFO 端点 1 ( USBFIFO1 ) , 偏移量 0x024

寄存器 14: USB FIFO 端点 2 ( USBFIFO2 ) , 偏移量 0x028

寄存器 15: USB FIFO 端点 3 ( USBFIFO3 ) , 偏移量 0x02C

寄存器 16: USB FIFO 端点 4 ( USBFIFO4 ) , 偏移量 0x030

寄存器 17: USB FIFO 端点 5 ( USBFIFO5 ) , 偏移量 0x034

寄存器 18: USB FIFO 端点 6 ( USBFIFO6 ) , 偏移量 0x038

寄存器 19: USB FIFO 端点 7 ( USBFIFO7 ) , 偏移量 0x03C

**重要:** 本寄存器为读敏感型寄存器。有关详细信息, 请参阅寄存器描述部分。

OTG A /  
主机

这些 32 位寄存器为 CPU 访问每个端点的 FIFO 提供地址。对这些地址进行写操作, 将向相应端点的发送 FIFO 中写入数据。对这些地址进行读操作, 将从相应端点的接收 FIFO 中读出数据。

OTG B /  
设备

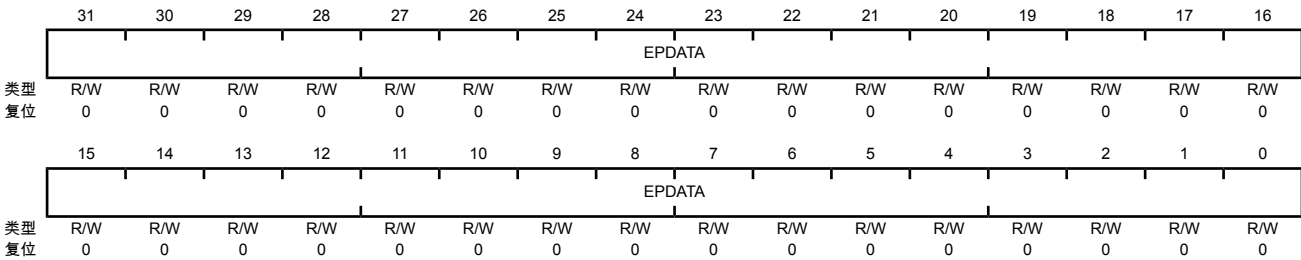
与 FIFO 交换数据的宽度, 可以根据需要设置为 8 位、16 位或 32 位, 任何允许访问的组合提供的数据存取是连续的。与一个包相关联的数据传送必须是相同宽度的, 也就是说这些数据始终是字节对齐、半字对齐或字对齐的。然而, 为了完成奇数字节或奇数字的传送, 最后一次传送可能比先前的数据传送少一些字节。

根据 FIFO 的大小和预期的最大包大小, FIFO 支持单包或双包缓存 ( 参见“单包缓存”一节在 1030 页 )。由于必须在每个包后写标志, 所以不支持多个包的猝发写方式。

端点 1-7 发生 STALL 握手响应或发送错误时, 相关联的 FIFO 被完全清空。

USB FIFO 端点 n ( USBFIFOn )

基址 0x4005.0000  
偏移量 0x020  
类型 R/W, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:0	EPDATA	R/W	0x0000.0000	端点数据 对此寄存器进行写操作, 将向发送FIFO中写入数据, 对此寄存器进行读操作, 将从接收FIFO中读出数据。



## 寄存器 20: USB设备控制寄存器 ( USBDEVCTL ) , 偏移量 0x060

OTG A /  
主机

USBDEVCTL 是用于控制和监测 USB VBUS 线路的 8 位寄存器。如果设备挂起, 无设备时钟可以接收, 则不对VBUS进行采样。此外, 在主机模式中, USBDEVCTL 提供 USB 控制器当前运行模式 ( 主机或设备 ) 的状态信息。如果 USB 控制器处于主机模式, 此寄存器可指示接入的是全速设备还是低速设备。

## USB设备控制寄存器 (USBDEVCTL)

基址 0x4005.0000  
偏移量 0x060  
类型 R/W, 复位 0x80

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DEV	FSDEV	LSDEV	VBUS		主机	HOSTREQ	SESSION
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	DEV	RO	1	设备模式  值 描述 0 USB控制器运行在电缆的OTG A端。 1 USB控制器运行在电缆的OTG B端。  注意: 此值只在会话过程中有效。
6	FSDEV	RO	0	检测到全速设备  值 描述 0 没有检测到全速设备。 1 检测到全速设备。
5	LSDEV	RO	0	检测到低速设备  值 描述 0 没有检测到低速设备。 1 检测到低速设备。

位/域	名称	类型	复位	描述
4:3	VBUS	RO	0x0	<p>VBUS电平</p> <p>值 描述</p> <p>0x0 低于SessionEnd电平 检测到VBUS低于0.5 V。</p> <p>0x1 高于SessionEnd电平，低于AValid电平 检测到VBUS高于0.5 V，低于1.5 V。</p> <p>0x2 高于AValid电平，低于VBUSValid电平 检测到VBUS高于1.5 V，低于4.75 V</p> <p>0x3 高于 VBUSValid 电平 检测到VBUS高于4.75 V</p>
2	主机	RO	0	<p>主机模式</p> <p>值 描述</p> <p>0 USB控制器作为设备。</p> <p>1 USB控制器作为主机。</p> <p>注意: 此值只在会话过程中有效。</p>
1	HOSTREQ	R/W	0	<p>主机请求</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 当进入挂起模式时，发起主机协商。</p> <p>当主机协商完成后，此位自动清零。</p>
0	SESSION	R/W	0	<p>会话开始/结束</p> <p>当作为OTG A Device运行：</p> <p>值 描述</p> <p>0 软件清零，结束会话。</p> <p>1 软件置1，开始会话。</p> <p>当作为OTG B Device运行：</p> <p>值 描述</p> <p>0 USB控制器结束一次会话。如果USB控制器处于挂起模式，软件清零，连接将软断开。</p> <p>1 USB控制器开始一次会话。软件置1，将发起会话请求协议。</p> <p>注意: 当USB控制器由于未定义的行为造成处于未挂起状态时，清除此位。</p>

寄存器 21: USB 发送动态 FIFO 大小 ( USBTXFIFOSZ ) , 偏移量 0x062

寄存器 22: USB 接收动态 FIFO 大小 ( USBRXFIFOSZ ) , 偏移量 0x063

OTG A /  
主机

此8位寄存器允许设置被选发送/接收端点为动态大小。USBEPIDX 用于配置每个发送端点的 FIFO 大小。

USB 动态 FIFO 大小 ( USBnXFIFOSZ )

基址 0x4005.0000  
偏移量 0x062  
类型 R/W, 复位 0x00

OTG B /  
设备

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												DPB		SIZE	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:5	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
4	DPB	R/W	0	双包缓存支持  值 描述 0 只支持单包缓存 1 支持双包缓存
3:0	SIZE	R/W	0x0	最大包大小 最大允许的包大小 如果 DPB = 0，FIFO 的大小与它相等；如果 DPB = 1，FIFO 的大小是它的两倍。  值 数据包长 ( 字节 ) 0x0 8 0x1 16 0x2 32 0x3 64 0x4 128 0x5 256 0x6 512 0x7 1024 0x8 2048 0x9-0xF 保留

**寄存器 23: USB 发送 FIFO 起始地址 ( USBTXFIFOADD ) , 偏移量 0x064**

**寄存器 24: USB 接收 FIFO 起始地址 ( USBRXFIFOADD ) , 偏移量 0x066**

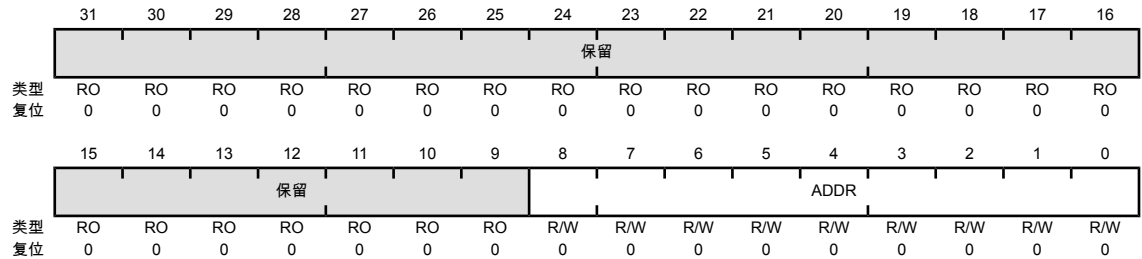
OTG A /  
主机

USBTXFIFOADD 和 USBRXFIFOADD 是 16 位寄存器，用于控制所选的发送和接收端点 FIFO 的起始地址。

OTG B /  
设备

USB 发送 FIFO 起始地址 ( USBnXFIFOADD )

基址 0x4005.0000  
偏移量 0x064  
类型 R/W, 复位 0x0000



位/域	名称	类型	复位	描述
31:9	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
8:0	ADDR	R/W	0x00	发送/接收起始地址 端点FIFO的起始地址
	值			起始地址
	0x0			0
	0x1			8
	0x2			16
	0x3			24
	0x4			32
	0x5			40
	0x6			48
	0x7			56
	0x8			64
	...			...
	0x1FF			4095

## 寄存器 25: USB连接时序寄存器 ( USBCONTIM ) , 偏移量 0x07A

OTG A /  
主机

此 8 位寄存器用于配置连接和协商的延时。

USB连接时序寄存器 (USBCONTIM)

基址 0x4005.0000

偏移量 0x07A

类型 R/W, 复位 0x5C

OTG B /  
设备

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								WTCON				WTID			
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:4	WTCON	R/W	0x5	连接等待 使用此位域可按需求配置等待延时，用于满足用户连接/断开的滤波需求；单位为 533.3 ns。默认为 2.667 $\mu$ s。
3:0	WTID	R/W	0xC	等待 ID 此位域根据需要来配置等待 ID 的延时，等待 ID 值有效时才启用 OTG 的 ID 检测。单位为 4.369 ms。默认为 52.43 ms。

**寄存器 26: USB OTG VBUS脉冲时序寄存器 ( USBVPLEN ) , 偏移量 0x07B****OTG**

此8位寄存器用于配置VBUS脉冲充电的持续时间。

USB OTG VBUS脉冲时序寄存器 (USBVPLEN)

基址 0x4005.0000  
 偏移量 0x07B  
 类型 R/W, 复位 0x3C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								VPLEN							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:0	VPLEN	R/W	0x3C	VBUS 脉宽 此位域用于配置 VBUS 脉冲充电的持续时间,单位 546.1 $\mu$ s。默认为 32.77 ms。

## 寄存器 27: USB 全速模式下最后的传输与帧结束时序寄存器 ( USBFSEOF ) , 偏移量 0x07D

OTG A /  
主机

此8位寄存器用于配置全速模式下允许最后的传输开始与帧结束(EOF)之间的最小时间间隔。

USB 全速模式下最后的传输与帧结束时序寄存器 (USBFSEOF)

基址 0x4005.0000

偏移量 0x07D

类型 R/W, 复位 0x77

OTG B /  
设备

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留							FSEOFG								
类型	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:0	FSEOFG	R/W	0x77	全速模式帧结束EOF间隙 在全速传输过程中，此位域用于设置最后的传输与帧结束(EOF)之间的时间间隔，单位 533.3 ns。默认为 63.46 μs。

## 寄存器 28: USB 低速模式下最后的传输与帧结束时序寄存器 ( USBLSEOF ) , 偏移量 0x07E

OTG A /  
主机

此8位寄存器用于配置低速模式下允许最后的传输开始与帧结束(EOF)之间的最小时间间隔。

OTG B /  
设备

USB 低速模式下最后的传输与帧结束时序寄存器 (USBLSEOF)

基址 0x4005.0000

偏移量 0x07E

类型 R/W, 复位 0x0072

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								LSEOFG							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:0	LSEOFG	R/W	0x72	低速模式帧结束EOF间隙 在低速传输过程中，此位域用于配置最后的传输与帧结束之间的时间间隔，单位1.067 μs，默认为121.6 μs。默认为 121.6 μs。



寄存器 29: USB 发送端点 0 功能地址 ( USBTXFUNCADDR0 ) , 偏移量 0x080

寄存器 30: USB 发送端点 1 功能地址 ( USBTXFUNCADDR1 ) , 偏移量 0x088

寄存器 31: USB 发送端点 2 功能地址 ( USBTXFUNCADDR2 ) , 偏移量 0x090

寄存器 32: USB 发送端点 3 功能地址 ( USBTXFUNCADDR3 ) , 偏移量 0x098

寄存器 33: USB 发送端点 4 功能地址 ( USBTXFUNCADDR4 ) , 偏移量 0x0A0

寄存器 34: USB 发送端点 5 功能地址 ( USBTXFUNCADDR5 ) , 偏移量 0x0A8

寄存器 35: USB 发送端点 6 功能地址 ( USBTXFUNCADDR6 ) , 偏移量 0x0B0

寄存器 36: USB 发送端点 7 功能地址 ( USBTXFUNCADDR7 ) , 偏移量 0x0B8

OTG A /

主机

USBTXFUNCADDRn 是 8 位可读/写寄存器, 用于记录通过相关端点 (EPn) 访问的目标功能地址。USBTXFUNCADDRn 必须根据每个使用的发送端点定义。

注意: USBTXFUNCADDR0 可用于端点 0 的接收和发送。

#### USB 发送端点 n 功能地址 ( USBTXFUNCADDRn )

基址 0x4005.0000  
偏移量 0x080  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										ADDR					
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
6:0	ADDR	R/W	0x00	设备地址 目标设备的USB总线地址。

寄存器 37: USB 发送端点 0 集线器地址 ( USBTXHUBADDR0 ), 偏移量 0x082

寄存器 38: USB 发送端点 1 集线器地址 ( USBTXHUBADDR1 ), 偏移量 0x08A

寄存器 39: USB 发送端点 2 集线器地址 ( USBTXHUBADDR2 ), 偏移量 0x092

寄存器 40: USB 发送端点 3 集线器地址 ( USBTXHUBADDR3 ), 偏移量 0x09A

寄存器 41: USB 发送端点 4 集线器地址 ( USBTXHUBADDR4 ), 偏移量 0x0A2

寄存器 42: USB 发送端点 5 集线器地址 ( USBTXHUBADDR5 ), 偏移量 0x0AA

寄存器 43: USB 发送端点 6 集线器地址 ( USBTXHUBADDR6 ), 偏移量 0x0B2

寄存器 44: USB 发送端点 7 集线器地址 ( USBTXHUBADDR7 ), 偏移量 0x0BA

OTG A /  
主机

USBTXHUBADDRn 是一个 8 位可读写寄存器, 与 USBTXHUBPORTn 类似, 只有当 USB 设备通过 USB 2.0 集线器连接到发送端点 EPn 时才必须被写入。此寄存器记录 USB 2.0 集线器的地址, 通过该地址可访问与端点相关联的目标。

注意: USBTXHUBADDR0 可用于端点 0 的接收和发送。

#### USB 发送端点 n 集线器地址 ( USBTXHUBADDRn )

基址 0x4005.0000  
偏移量 0x082  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										ADDR					
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
6:0	ADDR	R/W	0x00	集线器地址 该域用于指定 USB 2.0 集线器的 USB 总线地址。

寄存器 45: USB 发送端点 0 集线器端口 ( USBTXHUBPORT0 ) , 偏移量 0x083

寄存器 46: USB 发送端点 1 集线器端口 ( USBTXHUBPORT1 ) , 偏移量 0x08B

寄存器 47: USB 发送端点 2 集线器端口 ( USBTXHUBPORT2 ) , 偏移量 0x093

寄存器 48: USB 发送端点 3 集线器端口 ( USBTXHUBPORT3 ) , 偏移量 0x09B

寄存器 49: USB 发送端点 4 集线器端口 ( USBTXHUBPORT4 ) , 偏移量 0x0A3

寄存器 50: USB 发送端点 5 集线器端口 ( USBTXHUBPORT5 ) , 偏移量 0x0AB

寄存器 51: USB 发送端点 6 集线器端口 ( USBTXHUBPORT6 ) , 偏移量 0x0B3

寄存器 52: USB 发送端点 7 集线器端口 ( USBTXHUBPORT7 ) , 偏移量 0x0BB

OTG A /  
主机

USBTXHUBPORTn 是一个 8 位可读写寄存器, 与 USBTXHUBADDRn 类似, 只有当全速设备或低速设备通过 USB 2.0 集线器连接到发送端点 EPn 时才必须被写入。此寄存器记录 USB 2.0 集线器的端口, 通过该端口可访问与端点相关联的目标。

注意: USBTXHUBPORT0 可用于端点 0 的接收和发送。

#### USB 发送端点 n 集线器端口 ( USBTXHUBPORTn )

基址 0x4005.0000  
偏移量 0x083  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										端口					
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
6:0	端口	R/W	0x00	集线器端口 此位域指定 USB 集线器端点编号。

寄存器 53: USB 接收端点 1 功能地址 ( USBRXFUNCADDR1 ) , 偏移量 0x08C

寄存器 54: USB 接收端点 2 功能地址 ( USBRXFUNCADDR2 ) , 偏移量 0x094

寄存器 55: USB 接收端点 3 功能地址 ( USBRXFUNCADDR3 ) , 偏移量 0x09C

寄存器 56: USB 接收端点 4 功能地址 ( USBRXFUNCADDR4 ) , 偏移量 0x0A4

寄存器 57: USB 接收端点 5 功能地址 ( USBRXFUNCADDR5 ) , 偏移量 0x0AC

寄存器 58: USB 接收端点 6 功能地址 ( USBRXFUNCADDR6 ) , 偏移量 0x0B4

寄存器 59: USB 接收端点 7 功能地址 ( USBRXFUNCADDR7 ) , 偏移量 0x0BC

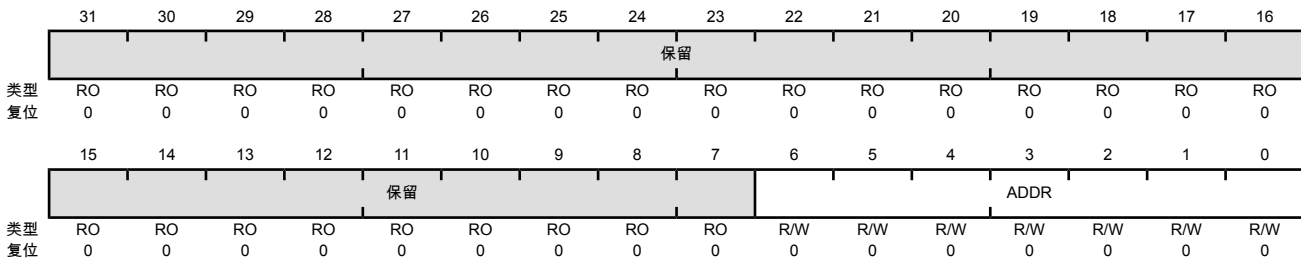
OTG A /  
主机

USBRXFUNCADDRn 是 8 位可读/写寄存器，用于记录通过相关端点 (EPn) 访问的目标功能地址。USBRXFUNCADDRn 必须根据每个使用的接收端点定义。

注意: USBTXFUNCADDR0 可用于端点 0 的接收和发送。

USB 接收端点 n 功能地址 ( USBRXFUNCADDRn )

基址 0x4005.0000  
偏移量 0x08C  
类型 R/W, 复位 0x00



位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
6:0	ADDR	R/W	0x00	设备地址 该域指定目标设备的 USB 总线地址。

寄存器 60: USB 接收端点 1 集线器地址 ( USBRXHUBADDR1 ) , 偏移量 0x08E

寄存器 61: USB 接收端点 2 集线器地址 ( USBRXHUBADDR2 ) , 偏移量 0x096

寄存器 62: USB 接收端点 3 集线器地址 ( USBRXHUBADDR3 ) , 偏移量 0x09E

寄存器 63: USB 接收端点 4 集线器地址 ( USBRXHUBADDR4 ) , 偏移量 0x0A6

寄存器 64: USB 接收端点 5 集线器地址 ( USBRXHUBADDR5 ) , 偏移量 0x0AE

寄存器 65: USB 接收端点 6 集线器地址 ( USBRXHUBADDR6 ) , 偏移量 0x0B6

寄存器 66: USB 接收端点 7 集线器地址 ( USBRXHUBADDR7 ) , 偏移量 0x0BE

OTG A /  
主机

USBRXHUBADDRn 是一个 8 位可读写寄存器, 与 USBRXHUBPORTn 类似, 只有当全速设备或低速设备通过 USB 2.0 集线器连接到接收端点 EPn 时才必须被写入。此寄存器记录 USB 2.0 集线器的地址, 通过该地址可访问与端点相关联的目标。

注意: USBTXHUBADDR0 可用于端点 0 的接收和发送。

#### USB 接收端点 n 集线器地址 ( USBRXHUBADDRn )

基址 0x4005.0000  
偏移量 0x08E  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										ADDR					
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
6:0	ADDR	R/W	0x00	集线器地址 该域用于指定 USB 2.0 集线器的 USB 总线地址。

寄存器 67: USB 接收端点 1 集线器端口 ( USBRXHUBPORT1 ) , 偏移量 0x08F

寄存器 68: USB 接收端点 2 集线器端口 ( USBRXHUBPORT2 ) , 偏移量 0x097

寄存器 69: USB 接收端点 3 集线器端口 ( USBRXHUBPORT3 ) , 偏移量 0x09F

寄存器 70: USB 接收端点 4 集线器端口 ( USBRXHUBPORT4 ) , 偏移量 0x0A7

寄存器 71: USB 接收端点 5 集线器端口 ( USBRXHUBPORT5 ) , 偏移量 0x0AF

寄存器 72: USB 接收端点 6 集线器端口 ( USBRXHUBPORT6 ) , 偏移量 0x0B7

寄存器 73: USB 接收端点 7 集线器端口 ( USBRXHUBPORT7 ) , 偏移量 0x0BF

OTG A /  
主机

USBRXHUBPORTn 是一个 8 位可读写寄存器, 与 USBRXHUBADDRn 类似, 只有当全速设备或低速设备通过 USB 2.0 集线器连接到接收端点 EPn 时才必须被写入。此寄存器记录 USB 2.0 集线器的端口, 通过该端口可访问与端点相关联的目标。

注意: USBTXHUBPORT0 可用于端点 0 的接收和发送。

#### USB 接收端点 n 集线器端口 ( USBRXHUBPORTn )

基址 0x4005.0000  
偏移量 0x08F  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留									端口						
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
6:0	端口	R/W	0x00	集线器端口 此位域指定 USB 集线器端点编号。

寄存器 74: USB 发送端点 1 最大传输数据 ( USBTXMAXP1 ), 偏移量 0x110

寄存器 75: USB 发送端点 2 最大传输数据 ( USBTXMAXP2 ), 偏移量 0x120

寄存器 76: USB 发送端点 3 最大传输数据 ( USBTXMAXP3 ), 偏移量 0x130

寄存器 77: USB 发送端点 4 最大传输数据 ( USBTXMAXP4 ), 偏移量 0x140

寄存器 78: USB 发送端点 5 最大传输数据 ( USBTXMAXP5 ), 偏移量 0x150

寄存器 79: USB 发送端点 6 最大传输数据 ( USBTXMAXP6 ), 偏移量 0x160

寄存器 80: USB 发送端点 7 最大传输数据 ( USBTXMAXP7 ), 偏移量 0x170

OTG A /

主机

USBTXMAXPn 16 位寄存器定义了通过发送端点单次可传输的最大数据量。

位 10:0 定义了单次传输的最大发送数据长度 ( 字节 )。该值最大为 1024 字节, 但必须服从 “USB 规范” 里批量传输、中断传输和全速模式下等时传输的包长限制。

OTG B /

设备

写入寄存器中的值代表的总数据大小不能超过发送端点的 FIFO 大小, 如果支持双包缓存, 不能超过 FIFO 大小的一半。

如果该寄存器在端点发送数据包后发生改变, 在向该寄存器写入新值之后, 发送端点 FIFO 必须完全被清空 ( 使用 USBTXCSRLn 寄存器中的 FLUSH 位 )。

注意: 在  $\mu$ DMA 基本模式中, 为了产生合适的中断, 必须将 USBTXMAXPn 设置为偶数字节。

#### USB 发送端点 n 最大传输数据 ( USBTXMAXPn )

基址 0x4005.0000  
偏移量 0x110  
类型 R/W, 复位 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留					MAXLOAD										
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:11	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
10:0	MAXLOAD	R/W	0x000	最大数据净荷 每次传输数据的最大字节数。

**寄存器 81: USB 端点 0 控制和状态低字节寄存器 ( USBCSR0 ) , 偏移量 0x102**

OTG A /  
主机

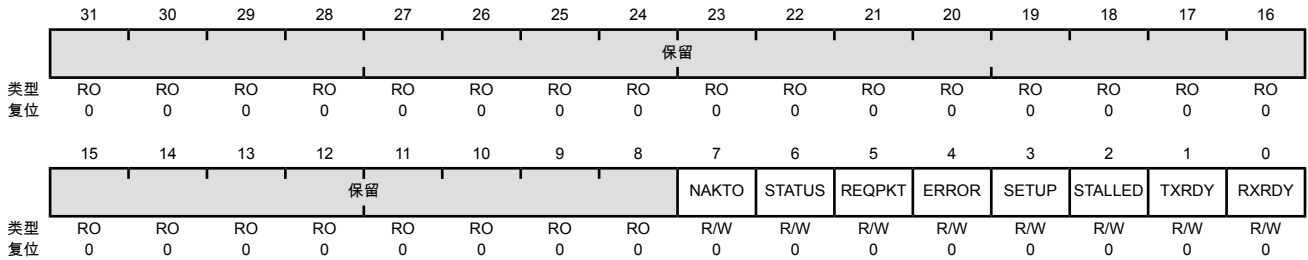
OTG B /  
设备

USBCSR0 是 8 位寄存器，为端点 0 提供控制和状态位。

**OTG A/主机模式**

**USB 端点 0 控制和状态低字节寄存器 (USBCSR0)**

基址 0x4005.0000  
偏移量 0x102  
类型 W1C, 复位 0x00



位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	NAKTO	R/W	0	NAK超时  值 描述 0 无超时 1 表明接收到 NAK 响应的次数超过了 USBNAKLMT 寄存器设置的次数，端点 0 已停止。  需软件清除此位允许端点继续。
6	STATUS	R/W	0	状态包  值 描述 0 无传输 1 发起状态阶段的传输。在置位 TXRDY 或 REQPKT 的同时，必须置位此位。  此位置位可确保寄存器 USBCSRH0 中 DT 位置位，以将 DATA1 包用于状态阶段传输。 当状态阶段传输完成，此位自动清零。

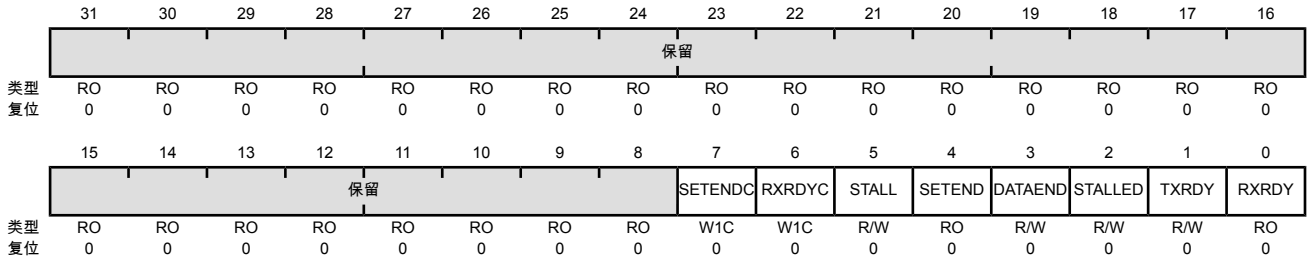


位/域	名称	类型	复位	描述
5	REQPKT	R/W	0	<p>请求包</p> <p>值 描述</p> <p>0 无请求。</p> <p>1 请求输入事务。</p> <p>当 RXRDY 位置位时，此位清零。</p>
4	ERROR	R/W	0	<p>错误</p> <p>值 描述</p> <p>0 无错误。</p> <p>1 执行三次事务，外设都无响应。此时 USBTXIS 寄存器中的 EP0 位也要置位。</p> <p>必须通过软件清除此位。</p>
3	SETUP	R/W	0	<p>建立令牌包</p> <p>值 描述</p> <p>0 发送输出令牌包。</p> <p>1 为此传输事务发送建立令牌包而不是输出令牌包。在置位 TXRDY 的同时，也应将此位置位。</p> <p>此位置位会清除 USBCSRH0 寄存器中的 DT 位，以发送 DATA0 数据包。</p>
2	STALLED	R/W	0	<p>端点挂起(Stalled)</p> <p>值 描述</p> <p>0 没有接收到握手信号</p> <p>1 接收到STALL握手信号。</p> <p>必须通过软件清除此位。</p>
1	TXRDY	R/W	0	<p>发送包准备好</p> <p>值 描述</p> <p>0 未准备好发送包。</p> <p>1 在装载数据包到发送FIFO中后，软件置1此位。此时 USBTXIS 寄存器中的 EP0 位也要置位。</p> <p>如果 TXRDY 位和 SETUP 位同时置位，将发送建立令牌包。如果只有 TXRDY位置位，将发送输出令牌包。</p> <p>当数据包发送完成时，此位自动清零。</p>
0	RXRDY	R/W	0	<p>接收包准备好</p> <p>值 描述</p> <p>0 未接收到包。</p> <p>1 表明在接收 FIFO 中已接收到数据包。此时 USBTXIS 寄存器中的 EP0 位也要置位。</p> <p>在包从接收FIFO读出后，必须软件置1此位，确认数据已经被从FIFO中读出。</p>

OTG B/设备模式

USB 端点 0 控制和状态低字节寄存器 (USBCSRL0)

基址 0x4005.0000  
 偏移量 0x102  
 类型 W1C, 复位 0x00



位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	SETENDC	W1C	0	Setup End Clear SETEND 位写 1 清零。
6	RXRDYC	W1C	0	RXRDY 清零 RXRDY 位写 1 清 0。
5	STALL	R/W	0	发送STALL握手  值 描述 0 无影响。 1 发送STALL握手，终止当前传输。  发送STALL握手完成后，此位自动清零。
4	SETEND	RO	0	建立结束  值 描述 0 控制传输没有结束，或者在 DATAEND 位置位后结束。 1 在 DATAEND 位被置位前，控制传输已结束。此时 USBTXIS 寄存器中的 EP0 位也要置位。  SETENDC 位写 1 清零。

位/域	名称	类型	复位	描述
3	DATAEND	R/W	0	<p>数据结束</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 在下面几种情况将该位置位：</p> <ul style="list-style-type: none"> <li>■ 当最后数据包的 TXRDY 位置位时</li> <li>■ 当最后的数据包卸下后将 RXRDY 位清零时</li> <li>■ 当零长度数据包的 TXRDY 位置位时</li> </ul> <p>此位自动清零。</p>
2	STALLED	R/W	0	<p>端点挂起(Stalled)</p> <p>值 描述</p> <p>0 未发送STALL握手。</p> <p>1 已发送STALL握手。</p> <p>必须通过软件清除此位。</p>
1	TXRDY	R/W	0	<p>发送包准备好</p> <p>值 描述</p> <p>0 未准备好发送包。</p> <p>1 在将输入数据包装载到发送 FIFO 中后，软件对该位置位。此时 USBTXIS 寄存器中的 EP0 位也要置位。</p> <p>当数据包发送完成时，此位自动清零。</p>
0	RXRDY	RO	0	<p>接收包准备好</p> <p>值 描述</p> <p>0 未接收到数据包。</p> <p>1 已接收到数据包。此时 USBTXIS 寄存器中的 EP0 位也要置位。</p> <p>RXRDYC 位写 1 清零。</p>

## 寄存器 82: USB 端点 0 控制和状态高字节寄存器 ( USBCSRH0 ) , 偏移量 0x103

USBSR0H 是 8 位寄存器, 为端点 0 提供控制和状态位。

OTG A /  
主机OTG B /  
设备

## OTG A/主机模式

## USB 端点 0 控制和状态高字节寄存器 (USBCSRH0)

基址 0x4005.0000  
偏移量 0x103  
类型 W1C, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													DTWE	DT	FLUSH	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
2	DTWE	R/W	0	数据切换写启用  值 描述 0 不能对 DT 位进行写操作。 1 将端点 0 数据切换的当前状态启用为可写 ( 请参考 DT )。  一旦写入新值, 此位自动清零。
1	DT	R/W	0	数据切换 此位指示端点0数据切换的当前状态。 如果 DTWE 位置位, 此位可根据数据切换的需求设置写值。如果 DTWE 位为低, 不能对该位进行写操作。写此位时需要小心, 它只应为复位发送端点而被改变。

位/域	名称	类型	复位	描述
0	FLUSH	R/W	0	清空 FIFO  值 描述 0 无影响。 1 清空下一个将从端点0 FIFO中发送/读出的数据包。同时，FIFO指针复位，TXRDY/RXRDY 位清零。  清空完成后，该位自动清零。  <b>重要:</b> 仅在 TXRDY/RXRDY 位置位时，该位才应当置位。否则，可能会导致数据毁坏。

## OTG B/设备模式

### USB 端点 0 控制和状态高字节寄存器 (USBCSRH0)

基址 0x4005.0000  
偏移量 0x103  
类型 W1C, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															FLUSH
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
0	FLUSH	R/W	0	清空 FIFO  值 描述 0 无影响。 1 清空下一个将从端点0 FIFO中发送/读出的数据包。同时，FIFO指针复位，TXRDY/RXRDY 位清零。  清空完成后，该位自动清零。  <b>重要:</b> 仅在 TXRDY/RXRDY 位置位时，该位才应当置位。否则，可能会导致数据毁坏。

### 寄存器 83: USB 端点 0 接收字节数量寄存器 ( USBCOUNT0 ) , 偏移量 0x108

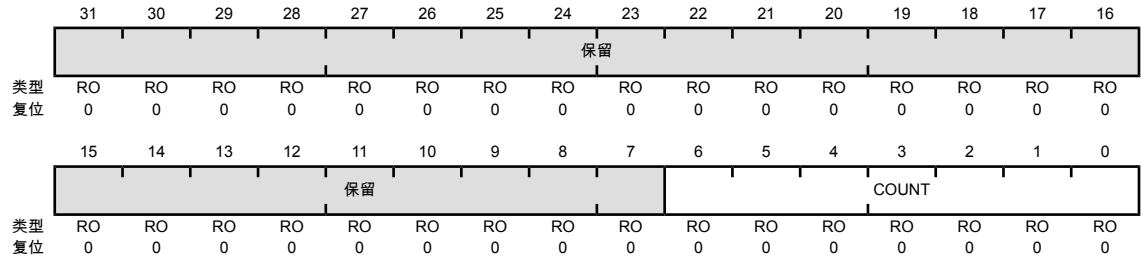
OTG A /  
主机

USBCOUNT0 是 8 位只读寄存器，用来指示端点 0 的 FIFO 中收到的数据的字节数量。此值伴随 FIFO 中内容变化而变化，只有在 RXRDY 位置位时有效。

OTG B /  
设备

USB 端点 0 接收字节数量寄存器 (USBCOUNT0)

基址 0x4005.0000  
偏移量 0x108  
类型 RO, 复位 0x00



位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
6:0	COUNT	RO	0x00	FIFO 计数 COUNT 是只读值，用来指示端点 0 的 FIFO 中收到的数据的字节数量。

## 寄存器 84: USB 端点 0 类型寄存器 ( USBTYPE0 ) , 偏移量 0x10A

OTG A /

主机

此8位寄存器用于存储与端点0通讯的目标设备的运行速度。

USB 端点 0 类型寄存器 (USBTYPE0)

基址 0x4005.0000

偏移量 0x10A

类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								SPEED		保留					
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:6	SPEED	R/W	0x0	运行速度 此位域指定目标设备的运行速度。目标设备被假定为与USB控制器具有相同的连接速度。  值    描述 0x0 - 0x1 保留 0x2    全速 0x3    低电平
5:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

**寄存器 85: USB NAK 限制寄存器 ( USBNAKLMT ) , 偏移量 0x10B**

OTG A /  
主机

USBNAKLMT 是 8 位寄存器, 用于设置在接收到一连串的 NAK 响应时, 端点 0 应设置多少个帧的超时。(其它端点的相同功能设置可通过它们的 USBTXINTERVALn 和 USBRXINTERVALn 寄存器实现。)

所选的帧数为  $2^{(m-1)}$  (其中,  $m$  是该寄存器中设定的值, 其有效数值为 2-16)。如果主机从目标设备接收的 NAK 相应大于此寄存器中代表的限度值, 端点将暂停。

注意: 值为 0 或 1 时将禁止 NAK 超时功能。

**USB NAK 限制寄存器 (USBNAKLMT)**

基址 0x4005.0000  
偏移量 0x10B  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												NAKLMT			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:5	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
4:0	NAKLMT	R/W	0x0	EPO NAK限制 此位域指定接收到 NAK 响应后的帧数。



寄存器 86: USB 端点 1 发送控制和状态低字节 ( USBTXCSRL1 ) , 偏移量 0x112

寄存器 87: USB 端点 2 发送控制和状态低字节 ( USBTXCSRL2 ) , 偏移量 0x122

寄存器 88: USB 端点 3 发送控制和状态低字节 ( USBTXCSRL3 ) , 偏移量 0x132

寄存器 89: USB 端点 4 发送控制和状态低字节 ( USBTXCSRL4 ) , 偏移量 0x142

寄存器 90: USB 端点 5 发送控制和状态低字节 ( USBTXCSRL5 ) , 偏移量 0x152

寄存器 91: USB 端点 6 发送控制和状态低字节 ( USBTXCSRL6 ) , 偏移量 0x162

寄存器 92: USB 端点 7 发送控制和状态低字节 ( USBTXCSRL7 ) , 偏移量 0x172

USBTXCSRLn 是 8 位寄存器, 为通过当前所选发送端点的传输提供控制和状态位。

OTG A /

主机

OTG B /

设备

## OTG A/主机模式

### USB 端点 n 发送控制和状态低字节 ( USBTXCSRLn )

基址 0x4005.0000

偏移量 0x112

类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								NAKTO	CLRDT	STALLED	SETUP	FLUSH	ERROR	FIFONE	TXRDY
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	NAKTO	R/W	0	NAK超时  值 描述 0 无超时 1 只用于批量端点: 接收 NAK 响应的次数超过了 USBTXINTERVALn 寄存器中的 NAKLMT 位域设置的次数, 发送端点暂停。需软件清除此位允许端点继续。
6	CLRDT	R/W	0	清除数据转换(Toggle) 向该位写 1 可将 USBTXCSRHn 寄存器中的 DT 位清零。

位/域	名称	类型	复位	描述
5	STALLED	R/W	0	<p>端点挂起(Stalled)</p> <p>值 描述</p> <p>0 没有接收到 STALL 握手信号。</p> <p>1 表示已接收到 STALL 握手信号。当该位置位时，停止任何正在进行 <math>\mu</math>DMA 请求，将 FIFO 完全清空并将 TXRDY 位清零。</p> <p>必须通过软件清除此位。</p>
4	SETUP	R/W	0	<p>建立令牌包</p> <p>值 描述</p> <p>0 不发送建立令牌包。</p> <p>1 为此传输事务发送建立令牌包而不是输出令牌包。在置位 TXRDY 的同时，也应将此位置位。</p> <p>注意： 此位置位也会将 USBTXCSRHn 寄存器中的 DT 位清零。</p>
3	FLUSH	R/W	0	<p>清空 FIFO</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 清空发送端点FIFO中最后的包。同时，FIFO 指针复位，TXRDY 位清零。此时 USBTXIS 寄存器中的 EPn 位也要置位。</p> <p>此位与 TXRDY 位同时置位，将放弃当前正在装载到 FIFO 的数据包。请注意，如果 FIFO 是双缓存的，FLUSH 位需被置位两次才可完全清空 FIFO。</p> <hr/> <p><b>重要：</b> 仅在 TXRDY 位置位时，该位才应当置位。否则，可能会导致数据毁坏。</p> <hr/>
2	ERROR	R/W	0	<p>错误</p> <p>值 描述</p> <p>0 无错误。</p> <p>1 尝试三次发送包，都未接收到握手包。此时 TXRDY 位清零，寄存器 USBTXIS 中的 EPn 位也置位，FIFO 完全清空。</p> <p>必须通过软件清除此位。</p> <p>注意： 仅对运行在批量或中断模式下端点有效。</p>
1	FIFONE	R/W	0	<p>FIFO不空</p> <p>值 描述</p> <p>0 FIFO空。</p> <p>1 发送FIFO中至少有1个包。</p>

位/域	名称	类型	复位	描述
0	TXRDY	R/W	0	发送包准备好

## 值 描述

0 未准备好发送包。

1 在装载数据包到发送FIFO中后，软件置1此位。

当数据包发送完成时，此位自动清零。此时 USBTXIS 寄存器中的 EPn 位也要置位。向双缓存的 FIFO 中装载第二个数据包前，TXRDY 位也将自动清零。

## OTG B/设备模式

## USB 端点 n 发送控制和状态低字节 ( USBTXCSRLn )

基址 0x4005.0000

偏移量 0x112

类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留										CLRDT	STALLED	STALL	FLUSH	UNDRN	FIFONE	TXRDY
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
6	CLRDT	R/W	0	清除数据转换(Toggle) 向该位写 1 可将 USBTXCSRHn 寄存器中的 DT 位清零。
5	STALLED	R/W	0	端点挂起(Stalled)  值 描述 0 未发送STALL握手。 1 已发送STALL握手。同时，清空 FIFO 并将 TXRDY 位清零。  必须通过软件清除此位。
4	STALL	R/W	0	发送STALL  值 描述 0 无影响。 1 向输入令牌发送STALL握手。  软件将此位清零，终止STALL状况。 注意： 此位对等时传输无效。

位/域	名称	类型	复位	描述
3	FLUSH	R/W	0	<p>清空 FIFO</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 清空发送端点FIFO中最后的包。同时，FIFO 指针复位，TXRDY 位清零。此时 USBTXIS 寄存器中的 EPn 位也要置位。</p> <p>此位与 TXRDY 位同时置位，将放弃当前正在装载到 FIFO 的数据包。请注意，如果 FIFO 是双缓存的，FLUSH 位需被置位两次才可完全清空 FIFO。</p> <hr/> <p><b>重要:</b> 仅在 TXRDY 位置位时，该位才应当置位。否则，可能会导致数据毁坏。</p> <hr/>
2	UNDRN	R/W	0	<p>欠运转(Underrun)</p> <p>值 描述</p> <p>0 无欠运转(underrun)发生。</p> <p>1 在 TXRDY 位还未置位时，收到输入令牌包。</p> <p>必须通过软件清除此位。</p>
1	FIFONE	R/W	0	<p>FIFO不空</p> <p>值 描述</p> <p>0 FIFO空。</p> <p>1 发送FIFO中至少有1个包。</p>
0	TXRDY	R/W	0	<p>发送包准备好</p> <p>值 描述</p> <p>0 未准备好发送包。</p> <p>1 在装载数据包到发送FIFO中后，软件置1此位。</p> <p>当数据包发送完成时，此位自动清零。此时 USBTXIS 寄存器中的 EPn 位也要置位。向双缓存的 FIFO 中装载第二个数据包前，TXRDY 位也将自动清零。</p>

寄存器 93: USB 发送端点 1 控制和状态高字节 ( USBTXCSRH1 ) , 偏移量 0x113

寄存器 94: USB 发送端点 2 控制和状态高字节 ( USBTXCSRH2 ) , 偏移量 0x123

寄存器 95: USB 发送端点 3 控制和状态高字节 ( USBTXCSRH3 ) , 偏移量 0x133

寄存器 96: USB 发送端点 4 控制和状态高字节 ( USBTXCSRH4 ) , 偏移量 0x143

寄存器 97: USB 发送端点 5 控制和状态高字节 ( USBTXCSRH5 ) , 偏移量 0x153

寄存器 98: USB 发送端点 6 控制和状态高字节 ( USBTXCSRH6 ) , 偏移量 0x163

寄存器 99: USB 发送端点 7 控制和状态高字节 ( USBTXCSRH7 ) , 偏移量 0x173

USBTXCSRHn 是 8 位寄存器, 为当选已选发送端点的传输提供额外的控制。

OTG A /

主机

OTG B /

设备

## OTG A/主机模式

### USB 发送端点 n 控制和状态高字节 ( USBTXCSRHn )

基址 0x4005.0000  
偏移量 0x113  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								AUTOSET	保留	MODE	DMAEN	FDT	DMAMOD	DTWE	DT
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写入操作过程中应该保持不变。
7	AUTOSET	R/W	0	自动置位  值 描述 0 TXRDY 位必须手动置位。 1 当最大包长 ( USBTXMAXPn 中的值 ) 中的数据装载到发送 FIFO 时, TXRDY 位可自动置位。如果装载的数据包小于最大包长, TXRDY 位必须手动置位。
6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

位/域	名称	类型	复位	描述
5	MODE	R/W	0	<p>模式</p> <p>值 描述</p> <p>0 设置端点方向为接受。</p> <p>1 设置端点方向为发送。</p> <p>注意: 此位只在发送和接收传输使用相同端点FIFO时起作用。</p>
4	DMAEN	R/W	0	<p>DMA请求启用</p> <p>值 描述</p> <p>0 禁止发送端点的<math>\mu</math>DMA请求。</p> <p>1 启用发送端点的<math>\mu</math>DMA请求。</p> <p>注意: 3个发送和3个接收端点可以连接到<math>\mu</math>DMA模块。如果此位为某一特殊端点置位, 相应地必须要对 USB DMA 选择 (USBDMASEL) 寄存器中的 DMAATX、DMABTX 或 DMAXTX 位域编程。</p>
3	FDT	R/W	0	<p>强制数据切换</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 无论是否收到 ACK 确认, 强制切换端点 DT 位, 清空 FIFO 中的数据包。此位用于中断通信速率反馈的发送端点。此位用于为等时端点的通信速率反馈的中断传输发送端点。</p>
2	DMAMOD	R/W	0	<p>DMA 请求模式</p> <p>值 描述</p> <p>0 <math>\mu</math>DMA 传输中每个包传输完成后都产生中断。</p> <p>1 整个 <math>\mu</math>DMA 传输完成后产生中断。</p> <p>注意: 此位不能在上面的 DMAEN 位清零之前清零, 也不能在同一时钟周期内清零。</p>
1	DTWE	R/W	0	<p>数据切换写启用</p> <p>值 描述</p> <p>0 不能对 DT 位进行写操作。</p> <p>1 将发送端点数据的当前状态启用为可写 (请参考 DT)。</p> <p>一旦写入新值, 此位自动清零。</p>
0	DT	R/W	0	<p>数据切换</p> <p>此位指示端点0数据切换的当前状态。</p> <p>如果 DTWE 位为高电平, 此位可根据数据切换的需求设置写值。如果 DTWE 位为低电平, 写到该位的任何值都将被忽略。写此位时需要小心, 它只应为复位发送端点而被改变。</p>

## OTG B/设备模式

## USB 发送端点 n 控制和状态高字节 ( USBTXCSRHn )

基址 0x4005.0000

偏移量 0x113

类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								AUTOSET	ISO	MODE	DMAEN	FDT	DMAMOD	保留	
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	AUTOSET	R/W	0	自动置位  值 描述 0 TXRDY 位必须手动置位。 1 当最大包长 ( USBTXMAXPn 中的值 ) 中的数据装载到发送 FIFO 时, TXRDY 位可自动置位。如果装载的数据包小于最大包长, TXRDY 位必须手动置位。
6	ISO	R/W	0	等时传输  值 描述 0 启用发送端点的批量或中断传输。 1 启用发送端点的等时传输。
5	MODE	R/W	0	模式  值 描述 0 设置端点方向为接受。 1 设置端点方向为发送。  注意: 此位只在发送和接收传输使用相同端点FIFO时起作用。
4	DMAEN	R/W	0	DMA请求启用  值 描述 0 禁止发送端点的μDMA请求。 1 启用发送端点的μDMA请求。  注意: 3个发送和接收端点可以连接到μDMA模块。如果此位为某一特殊端点置位, 相应地必须要对 USB DMA 选择 (USBDMASEL) 寄存器中的 DMAATX、DMABTX 或 DMACTX 位域编程。

位/域	名称	类型	复位	描述
3	FDT	R/W	0	<p>强制数据切换</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 无论是否收到 ACK 确认，强制切换端点 DT 位，清空 FIFO 中的数据包。此位用于中断通信速率反馈的发送端点。此位用于为等时端点的通信速率反馈的中断传输发送端点。</p>
2	DMAMOD	R/W	0	<p>DMA 请求模式</p> <p>值 描述</p> <p>0 <math>\mu</math>DMA 传输中每个包传输完成后都产生中断。</p> <p>1 整个 <math>\mu</math>DMA 传输完成后产生中断。</p> <p>注意： 此位不能在上面的 DMAEN 位清零之前清零，也不能在同一时钟周期内清零。</p>
1:0	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。</p>



寄存器 100: USB 接收端点 1 最大传输数据 ( USBRXMAXP1 ), 偏移量 0x114

寄存器 101: USB 接收端点 2 最大传输数据 ( USBRXMAXP2 ), 偏移量 0x124

寄存器 102: USB 接收端点 3 最大传输数据 ( USBRXMAXP3 ), 偏移量 0x134

寄存器 103: USB 接收端点 4 最大传输数据 ( USBRXMAXP4 ), 偏移量 0x144

寄存器 104: USB 接收端点 5 最大传输数据 ( USBRXMAXP5 ), 偏移量 0x154

寄存器 105: USB 接收端点 6 最大传输数据 ( USBRXMAXP6 ), 偏移量 0x164

寄存器 106: USB 接收端点 7 最大传输数据 ( USBRXMAXP7 ), 偏移量 0x174

OTG A /

主机

USBRXMAXPn 是 16 位寄存器, 定义了通过所选接收端点单次可传输的最大数据的长度。

位 10:0 定义了单次传输的最大发送数据长度 ( 字节 )。该值最大为 1024 字节, 但必须服从 “USB 规范” 里批量传输、中断传输和全速模式下等时传输的包长限制。

OTG B /

设备

写入寄存器中的值代表的总数据大小不能超过接收端点的 FIFO 大小, 如果支持双包缓存, 不能超过 FIFO 大小的一半。

注意: 在  $\mu$ DMA 基本模式中, 为了产生合适的中断, 必须将 USBRXMAXPn 设置为偶数字节。

#### USB 接收端点 n 最大传输数据 ( USBRXMAXPn )

基址 0x4005.0000  
偏移量 0x114  
类型 R/W, 复位 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留					MAXLOAD										
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:11	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
10:0	MAXLOAD	R/W	0x000	最大数据净荷 每次传输数据的最大字节数。

寄存器 107: USB 接收端点 1 控制和状态低字节 ( USBRXCSRL1 ) , 偏移量 0x116

寄存器 108: USB 接收端点 2 控制和状态低字节 ( USBRXCSRL2 ) , 偏移量 0x126

寄存器 109: USB 接收端点 3 控制和状态低字节 ( USBRXCSRL3 ) , 偏移量 0x136

寄存器 110: USB 接收端点 4 控制和状态低字节 ( USBRXCSRL4 ) , 偏移量 0x146

寄存器 111: USB 接收端点 5 控制和状态低字节 ( USBRXCSRL5 ) , 偏移量 0x156

寄存器 112: USB 接收端点 6 控制和状态低字节 ( USBRXCSRL6 ) , 偏移量 0x166

寄存器 113: USB 接收端点 7 控制和状态低字节 ( USBRXCSRL7 ) , 偏移量 0x176

USBRXCSRLn 是 8 位寄存器, 为通过当前所选接收端点的传输提供控制和状态位。

OTG A /

主机

OTG B /

设备

## OTG A/主机模式

### USB 接收端点 n 控制和状态低字节 ( USBRXCSRLn )

基址 0x4005.0000  
偏移量 0x116  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								CLRDT	STALLED	REQPKT	FLUSH	DATAERR / NAKTO	ERROR	FULL	RXRDY
类型	RO	RO	RO	RO	RO	RO	RO	RO	W1C	R/W	R/W	R/W	R/W	R/W	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	CLRDT	W1C	0	清除数据转换(Toggle) 向该位写 1 可将 USBRXCSRHn 寄存器中的 DT 位清零。
6	STALLED	R/W	0	端点挂起(Stalled)  值 描述 0 没有接收到 STALL 握手信号。 1 接收到STALL握手信号。此时 USBRXIS 寄存器中的 EPn 位也要置位。  必须通过软件清除此位。

位/域	名称	类型	复位	描述
5	REQPKT	R/W	0	<p>请求包</p> <p>值 描述</p> <p>0 无请求。</p> <p>1 请求输入事务。</p> <p>当 RXRDY 位置位时，此位清零。</p>
4	FLUSH	R/W	0	<p>清空 FIFO</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 清空下一个将从端点FIFO中读出的包。同时，FIFO 指针复位，RXRDY 位清零。</p> <p>请注意，如果 FIFO 是双缓存的，FLUSH 位需被置位两次才可完全清空 FIFO。</p> <hr/> <p><b>重要:</b> 仅在 RXRDY 位置位时，该位才应当置位。否则，可能会导致数据毁坏。</p> <hr/>
3	DATAERR / NAKTO	R/W	0	<p>数据错误/ NAK 超时</p> <p>值 描述</p> <p>0 正常工作。</p> <p>1 只用于等时端点：表示 RXRDY 位置位且数据包有 CRC 或位填充错误。当 RXRDY 位清零时，此位清零。</p> <p>只用于批量端点：接收 NAK 响应的次数超过了 USBRXINTERVALn 寄存器中的 NAKLMT 位域设置的次数，接收端点暂停。需软件清除此位允许端点继续。</p>
2	ERROR	R/W	0	<p>错误</p> <p>值 描述</p> <p>0 无错误。</p> <p>1 尝试三次接收包，都未接收到数据包。此时 USBRXIS 寄存器中的 EPn 位也要置位。</p> <p>必须通过软件清除此位。</p> <p>注意： 仅当此接收端点运行在批量或中断模式下时，此位才有效。在等时模式，它总是返回 0。</p>
1	FULL	RO	0	<p>FIFO满</p> <p>值 描述</p> <p>0 接收 FIFO 未滿。</p> <p>1 此时没有更多的数据包可装载到接收 FIFO 中。</p>

位/域	名称	类型	复位	描述
0	RXRDY	R/W	0	接收包准备好

值 描述

0 未接收到数据包。

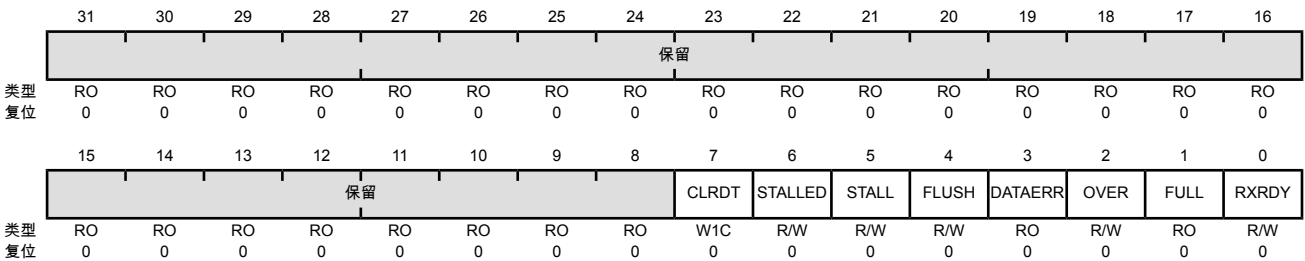
1 已接收到数据包。此时 USBRXIS 寄存器中的 EPn 位也要置位。

如果 USBRXCSRHn 寄存器中的 AUTOCLR 位置位，则当 USBRXMAXPn 字节的数据包从接收 FIFO 卸下时，该位会自动清零。如果 AUTOCLR 位清零或者如果卸下了比最大包长小的数据包，则软件必须在数据包已从接收 FIFO 中卸载时将该位清零。

OTG B/设备模式

USB 接收端点 n 控制和状态低字节 ( USBRXCSRLn )

基址 0x4005.0000  
偏移量 0x116  
类型 R/W, 复位 0x00



位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	CLRDT	W1C	0	清除数据转换(Toggle) 向该位写 1 可将 USBRXCSRHn 寄存器中的 DT 位清零。
6	STALLED	R/W	0	端点挂起(Stalled)  值 描述 0 未发送STALL握手。 1 已发送STALL握手。  必须通过软件清除此位。
5	STALL	R/W	0	发送STALL  值 描述 0 无影响。 1 发送 STALL 握手。  软件必须将该位清零，以终止该 STALL 状况。  注意： 此位对用于等时传输的端点无影响。

位/域	名称	类型	复位	描述
4	FLUSH	R/W	0	<p>清空 FIFO</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 清空从端点接收 FIFO 中读出的下一个数据包。同时，FIFO 指针复位，RXRDY 位清零。</p> <p>CPU 向此位写 1 清空下一个将从端点 FIFO 中读出的数据包。同时，FIFO 指针复位，RXRDY 位清零。请注意，如果 FIFO 是双缓存的，FLUSH 位需被置位两次才可完全清空 FIFO。</p> <hr/> <p><b>重要:</b> 仅在 RXRDY 位置位时，该位才应当置位。否则，可能会导致数据毁坏。</p> <hr/>
3	DATAERR	RO	0	<p>数据错误</p> <p>值 描述</p> <p>0 正常工作。</p> <p>1 表示 RXRDY 位置位且数据包有 CRC 或位填充错误。</p> <p>当 RXRDY 位清零时，此位清零。</p> <p>注意: 仅当该端点以等时模式操作时，此位才有效。在批量模式下，它总是返回 0。</p>
2	OVER	R/W	0	<p>超限</p> <p>值 描述</p> <p>0 无超限错误。</p> <p>1 表示输出包不能装载到接收 FIFO 中。</p> <p>必须通过软件清除此位。</p> <p>注意: 仅当该端点以等时模式操作时，此位才有效。在批量模式下，它总是返回 0。</p>
1	FULL	RO	0	<p>FIFO 满</p> <p>值 描述</p> <p>0 接收 FIFO 未满。</p> <p>1 此时没有更多的数据包可装载到接收 FIFO 中。</p>
0	RXRDY	R/W	0	<p>接收包准备好</p> <p>值 描述</p> <p>0 未接收到数据包。</p> <p>1 已接收到数据包。此时 USBRXIS 寄存器中的 EPn 位也要置位。</p> <p>如果 USBRXCSRn 寄存器中的 AUTOCLR 位置位，则当 USBRXMAXPn 字节的数据包从接收 FIFO 卸下时，该位会自动清零。如果 AUTOCLR 位清零或者如果卸下了比最大包长小的数据包，则软件必须在数据包已从接收 FIFO 中卸载时将该位清零。</p>

- 寄存器 114: USB 接收端点 1 控制和状态高字节 ( USBRXCSRH1 ) , 偏移量 0x117
- 寄存器 115: USB 接收端点 2 控制和状态高字节 ( USBRXCSRH2 ) , 偏移量 0x127
- 寄存器 116: USB 接收端点 3 控制和状态高字节 ( USBRXCSRH3 ) , 偏移量 0x137
- 寄存器 117: USB 接收端点 4 控制和状态高字节 ( USBRXCSRH4 ) , 偏移量 0x147
- 寄存器 118: USB 接收端点 5 控制和状态高字节 ( USBRXCSRH5 ) , 偏移量 0x157
- 寄存器 119: USB 接收端点 6 控制和状态高字节 ( USBRXCSRH6 ) , 偏移量 0x167
- 寄存器 120: USB 接收端点 7 控制和状态高字节 ( USBRXCSRH7 ) , 偏移量 0x177

USBRXCSRHn 是 8 位寄存器, 为通过当前所选接收端点的传输提供额外的控制和状态位。

OTG A /  
主机

OTG B /  
设备

### OTG A/主机模式

#### USB 接收端点 n 控制和状态高字节 ( USBRXCSRHn )

基址 0x4005.0000  
偏移量 0x117  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								AUTOCL	AUTORQ	DMAEN	PIDERR	DMAMOD	DTWE	DT	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	R/W	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	AUTOCL	R/W	0	自动清零

#### 值 描述

0 无影响。

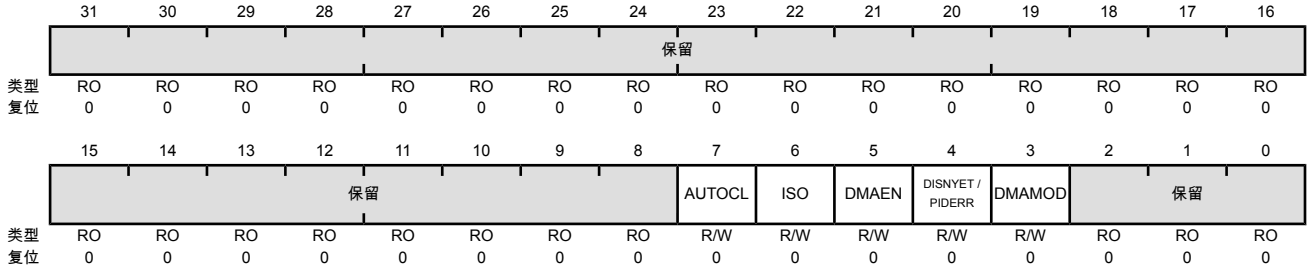
- 1 当 USBRXMAXPn 字节的数据包已从接收 FIFO 卸下时, RXRDY 位可自动清零。当卸下的包长小于最大包长时, 必须手动将 RXRDY 位清零。使用  $\mu$ DMA 从接收 FIFO 中卸载数据时需要谨慎, 无论 USBRXMAXPn 寄存器中的 MAXLOAD 位域的值是多少, 每次都从接收 FIFO 中读出数据 4 字节的块, 参见“DMA操作”在 1036页。

位/域	名称	类型	复位	描述
6	AUTORQ	R/W	0	<p>自动请求</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 当 RXRDY 位清零时，REQPKT 位可自动置位。</p> <p>注意： 当收到一个短包时，此位自动清零。</p>
5	DMAEN	R/W	0	<p>DMA请求启用</p> <p>值 描述</p> <p>0 禁止接收端点的μDMA请求。</p> <p>1 启用接收端点的 μDMA 请求。</p> <p>注意： 3个发送和接收端点可以连接到μDMA模块。如果此位为某一特殊端点置位，相应地必须要对 USB DMA 选择 (USBDMASEL) 寄存器中的 DMAARX、DMABRX 或 DMACRX 位域编程。</p>
4	PIDERR	RO	0	<p>PID错误</p> <p>值 描述</p> <p>0 无错误。</p> <p>1 等时传输的接收包中存在 PID 错误。</p> <p>在批量或中断传输时，此位忽略。</p>
3	DMAMOD	R/W	0	<p>DMA 请求模式</p> <p>值 描述</p> <p>0 μDMA 传输中每个包传输完成后都产生中断。</p> <p>1 整个 μDMA 传输完成后产生中断。</p> <p>注意： 此位不能在上面的 DMAEN 位清零之前清零，也不能在同一时钟周期内清零。</p>
2	DTWE	RO	0	<p>数据切换写启用</p> <p>值 描述</p> <p>0 不能对 DT 位进行写操作。</p> <p>1 将接收端点数据切换的当前状态启用为可写（请参考 DT）。</p> <p>一旦写入新值，此位自动清零。</p>
1	DT	RO	0	<p>数据切换</p> <p>此位指示端点0数据切换的当前状态。</p> <p>如果 DTWE 位为高电平，此位可根据数据切换的需求设置写值。如果 DTWE 位为低电平，写到该位的任何值都将被忽略。写此位时需要小心，它只应为复位接收端点而被改变。</p>
0	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>

OTG B/设备模式

USB 接收端点 n 控制和状态高字节 ( USBRXCSRHn )

基址 0x4005.0000  
 偏移量 0x117  
 类型 R/W, 复位 0x00



位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	AUTOCL	R/W	0	自动清零  值 描述 0 无影响。 1 当 USBRXMAXPn 字节的数据包已从接收 FIFO 卸下时，RXRDY 位可自动清零。当卸下的包长小于最大包长时，必须手动将 RXRDY 位清零。使用 $\mu$ DMA 从接收 FIFO 中卸载数据时需要谨慎，无论 USBRXMAXPn 寄存器中的 MAXLOAD 位域的值是多少，每次都从接收 FIFO 中读出数据 4 字节的块，参见“DMA操作”在 1036页。
6	ISO	R/W	0	等时传输  值 描述 0 启用接收端点的等时传输。 1 启用接收端点的批量/中断传输。
5	DMAEN	R/W	0	DMA请求启用  值 描述 0 禁止接收端点的 $\mu$ DMA 请求。 1 启用接收端点的 $\mu$ DMA 请求。  注意： 3个发送和接收端点可以连接到 $\mu$ DMA 模块。如果此位为某一特殊端点置位，相应地必须要对 USB DMA 选择 (USBDMASEL) 寄存器中的 DMAARX、DMABRX 或 DMACRX 位域编程。
4	DISNYET / PIDERR	R/W	0	禁止 NYET / PID 错误  值 描述 0 无影响。 1 对于批量或中断传输：禁止 NYET 握手的发送。当此位置1，所有成功接收的包都将被确认，即使此时 FIFO 变满。 对于等时传输：表明在接收到的数据包存在 PID 错误。



位/域	名称	类型	复位	描述
3	DMAMOD	R/W	0	DMA 请求模式  值 描述 0 $\mu$ DMA 传输中每个包传输完成后都产生中断。 1 整个 $\mu$ DMA 传输完成后产生中断。  注意: 此位不能在上面的 DMAEN 位清零之前清零, 也不能在同一时钟周期内清零。
2:0	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

寄存器 121: USB 接收端点 1 字节计数 ( USBRXCOUNT1 ) , 偏移量 0x118

寄存器 122: USB 接收端点 2 字节计数 ( USBRXCOUNT2 ) , 偏移量 0x128

寄存器 123: USB 接收端点 3 字节计数 ( USBRXCOUNT3 ) , 偏移量 0x138

寄存器 124: USB 接收端点 4 字节计数 ( USBRXCOUNT4 ) , 偏移量 0x148

寄存器 125: USB 接收端点 5 字节计数 ( USBRXCOUNT5 ) , 偏移量 0x158

寄存器 126: USB 接收端点 6 字节计数 ( USBRXCOUNT6 ) , 偏移量 0x168

寄存器 127: USB 接收端点 7 字节计数 ( USBRXCOUNT7 ) , 偏移量 0x178

OTG A /

主机

注意: 返回的值在 FIFO 被卸下时改变, 此值只在寄存器 USBRXCSRLn 中的 RXRDY 位置位时有效。

USBRXCOUNTn 是 16 位只读寄存器, 用于指示在要从接收 FIFO 读出的、当前行的数据包中保留的数据字节数。如果被发送的是多个批量数据包, 则给出的数字适用于组合的数据包。

OTG B /

设备

USB 接收端点 n 字节计数 ( USBRXCOUNTn )

基址 0x4005.0000

偏移量 0x118

类型 RO, 复位 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留			COUNT												
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:13	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
12:0	COUNT	RO	0x000	接收包字节数量 指示接收包的字节数。

寄存器 128: USB 端点 1 主机发送配置类型 ( USBTXTYPE1 ) , 偏移量 0x11A

寄存器 129: USB 端点 2 主机发送配置类型 ( USBTXTYPE2 ) , 偏移量 0x12A

寄存器 130: USB 端点 3 主机发送配置类型 ( USBTXTYPE3 ) , 偏移量 0x13A

寄存器 131: USB 端点 4 主机发送配置类型 ( USBTXTYPE4 ) , 偏移量 0x14A

寄存器 132: USB 端点 5 主机发送配置类型 ( USBTXTYPE5 ) , 偏移量 0x15A

寄存器 133: USB 端点 6 主机发送配置类型 ( USBTXTYPE6 ) , 偏移量 0x16A

寄存器 134: USB 端点 7 主机发送配置类型 ( USBTXTYPE7 ) , 偏移量 0x17A

OTG A /

主机

USBTXTYPE<sub>n</sub> 是 8 位寄存器，必须写入端点的目标端点号、用于当前所选发送端点的传输协议及其运行速度。

USB 端点 n 主机发送配置类型 ( USBTXTYPE<sub>n</sub> )

基址 0x4005.0000

偏移量 0x11A

类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								SPEED		PROTO		TEP			
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:6	SPEED	R/W	0x0	运行速度 此位域指定目标设备的运行速度：  值 描述 0x0 默认值 目标设备假定与USB控制器具有相同的连接速度。 0x1 保留 0x2 全速 0x3 低电平
5:4	PROTO	R/W	0x0	协议 必须软件配置此位域，用来选择发送端点需要的协议：  值 描述 0x0 控制 0x1 等时传输 0x2 批量传输 0x3 中断

位/域	名称	类型	复位	描述
3:0	TEP	R/W	0x0	目标端点号 此端点号必须软件配置，包含在设备枚举过程中返回USB控制器的描述符中。

寄存器 135: USB 端点 1 主机发送间隔 ( USBTXINTERVAL1 ) , 偏移量 0x11B

寄存器 136: USB 端点 2 主机发送间隔 ( USBTXINTERVAL2 ) , 偏移量 0x12B

寄存器 137: USB 端点 3 主机发送间隔 ( USBTXINTERVAL3 ) , 偏移量 0x13B

寄存器 138: USB 端点 4 主机发送间隔 ( USBTXINTERVAL4 ) , 偏移量 0x14B

寄存器 139: USB 端点 5 主机发送间隔 ( USBTXINTERVAL5 ) , 偏移量 0x15B

寄存器 140: USB 端点 6 主机发送间隔 ( USBTXINTERVAL6 ) , 偏移量 0x16B

寄存器 141: USB 端点 7 主机发送间隔 ( USBTXINTERVAL7 ) , 偏移量 0x17B

OTG A /  
主机

USBTXINTERVAL<sub>n</sub> 是 8 位寄存器, 为当前选中的发送端点的中断传输和等时传输定义了轮询间隔。为批量传输定义了接收到多少帧的 NAK 响应将超时。

USBTXINTERVAL<sub>n</sub> 寄存器的值定义了帧数, 如下表:

传输类型	速度	有效值 (m)	说明
中断	低速或全速	0x00 - 0xFF	轮询间隔是 $m$ 帧。
等时传输	全速	0x01 - 0x10	轮询间隔是 $2^{(m-1)}$ 帧。
批量传输	全速	0x02 - 0x10	NAK 限制是 $2^{(m-1)}$ 帧。值为 0 或 1 时将禁止 NAK 超时功能。

#### USB 端点 n 主机发送间隔 ( USBTXINTERVAL<sub>n</sub> )

基址 0x4005.0000  
偏移量 0x11B  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								TXPOLL / NAKLMT							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7:0	TXPOLL / NAKLMT	R/W	0x00	发送轮询/NAK限制 中断传输/等时传输的巡检间隔; 批量传输的NAK限制。全部有效值见上面的表; 其他值保留。

寄存器 142: USB 端点 1 主机配置接收类型 ( USBRXTYPE1 ) , 偏移量 0x11C

寄存器 143: USB 端点 2 主机配置接收类型 ( USBRXTYPE2 ) , 偏移量 0x12C

寄存器 144: USB 端点 3 主机配置接收类型 ( USBRXTYPE3 ) , 偏移量 0x13C

寄存器 145: USB 端点 4 主机配置接收类型 ( USBRXTYPE4 ) , 偏移量 0x14C

寄存器 146: USB 端点 5 主机配置接收类型 ( USBRXTYPE5 ) , 偏移量 0x15C

寄存器 147: USB 端点 6 主机配置接收类型 ( USBRXTYPE6 ) , 偏移量 0x16C

寄存器 148: USB 端点 7 主机配置接收类型 ( USBRXTYPE7 ) , 偏移量 0x17C

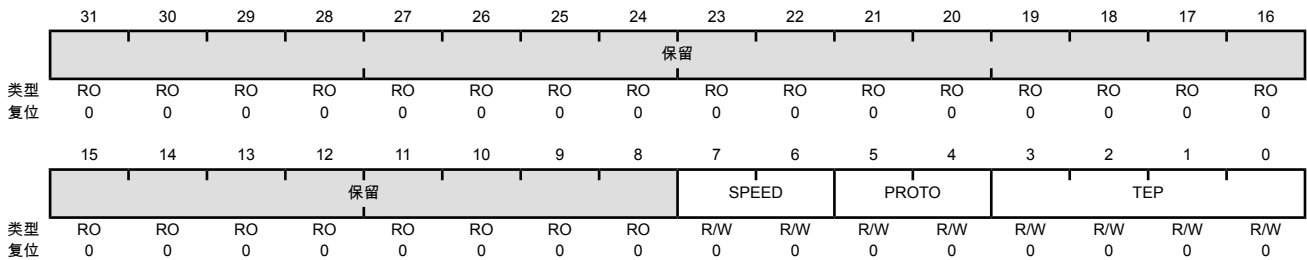
OTG A /  
主机

USBRXTYPE<sub>n</sub> 是 8 位寄存器 , 必须写入端点的目标端点号、用于当前所选接收端点的传输协议及其运行速度。

USB 端点 n 主机配置接收类型 ( USBRXTYPE<sub>n</sub> )

基址 0x4005.0000

偏移量 0x11C  
类型 R/W, 复位 0x00



位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:6	SPEED	R/W	0x0	运行速度 此位域指定目标设备的运行速度：  值 描述 0x0 默认值 目标设备假定与USB控制器具有相同的连接速度。 0x1 保留 0x2 全速 0x3 低电平
5:4	PROTO	R/W	0x0	协议 必须软件配置此位域，用来选择接收端点需要的协议：  值 描述 0x0 控制 0x1 等时传输 0x2 批量传输 0x3 中断

---

位/域	名称	类型	复位	描述
3:0	TEP	R/W	0x0	目标端点号 此端点号必须软件设置，包含在设备枚举过程中返回 USB 控制器的接收端点描述符中。

- 寄存器 149: USB 端点 1 主机接收轮询间隔 ( USBRXINTERVAL1 ) , 偏移量 0x11D
- 寄存器 150: USB 端点 2 主机接收轮询间隔 ( USBRXINTERVAL2 ) , 偏移量 0x12D
- 寄存器 151: USB 端点 3 主机接收轮询间隔 ( USBRXINTERVAL3 ) , 偏移量 0x13D
- 寄存器 152: USB 端点 4 主机接收轮询间隔 ( USBRXINTERVAL4 ) , 偏移量 0x14D
- 寄存器 153: USB 端点 5 主机接收轮询间隔 ( USBRXINTERVAL5 ) , 偏移量 0x15D
- 寄存器 154: USB 端点 6 主机接收轮询间隔 ( USBRXINTERVAL6 ) , 偏移量 0x16D
- 寄存器 155: USB 端点 7 主机接收轮询间隔 ( USBRXINTERVAL7 ) , 偏移量 0x17D

OTG A /  
主机

USBRXINTERVAL<sub>n</sub> 是 8 位寄存器，为当前选中的接收端点的中断传输和等时传输定义了轮询间隔。为批量传输定义了接收到多少帧的 NAK 响应将超时。

USBRXINTERVAL<sub>n</sub> 寄存器的值定义了帧数，如下表：

传输类型	速度	有效值 (m)	说明
中断	低速或全速	0x00 - 0xFF	轮询间隔是 <i>m</i> 帧。
等时传输	全速	0x01 - 0x10	轮询间隔是 $2^{(m-1)}$ 帧。
批量传输	全速	0x02 - 0x10	NAK 限制是 $2^{(m-1)}$ 帧。值为 0 或 1 时将禁止 NAK 超时功能。

USB 端点 n 主机接收轮询间隔 ( USBRXINTERVAL<sub>n</sub> )

基址 0x4005.0000  
偏移量 0x11D  
类型 R/W, 复位 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								TXPOLL / NAKLMT							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7:0	TXPOLL / NAKLMT	R/W	0x00	接收巡检/NAK限制 中断传输/等时传输的巡检间隔；批量传输的NAK限制。全部有效值见上面的表；其他值保留。



寄存器 156: USB 端点 1 块传输中请求包数量 ( USBRQPKTCOUNT1 ) , 偏移量 0x304

寄存器 157: USB 端点 2 块传输中请求包数量 ( USBRQPKTCOUNT2 ) , 偏移量 0x308

寄存器 158: USB 端点 3 块传输中请求包数量 ( USBRQPKTCOUNT3 ) , 偏移量 0x30C

寄存器 159: USB 端点 4 块传输中请求包数量 ( USBRQPKTCOUNT4 ) , 偏移量 0x310

寄存器 160: USB 端点 5 块传输中请求包数量 ( USBRQPKTCOUNT5 ) , 偏移量 0x314

寄存器 161: USB 端点 6 块传输中请求包数量 ( USBRQPKTCOUNT6 ) , 偏移量 0x318

寄存器 162: USB 端点 7 块传输中请求包数量 ( USBRQPKTCOUNT7 ) , 偏移量 0x31C

OTG A /  
主机

用于主机模式时, 此 16 位可读写寄存器将指定块传输中可以向接收端点 n 传输的批量数据包的数量。USB 控制器使用此寄存器记录的值来决定 USBRXCSRn 寄存器中的 AUTORQ 位置位时发起请求包的数量。请参考“作为主机时的输入事务”在 1033 页。

注意: 多个包含并到单批量包中作为一个数据包。

#### USB 端点 n 块传输中请求包数量 ( USBRQPKTCOUNTn )

基址 0x4005.0000  
偏移量 0x304  
类型 R/W, 复位 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COUNT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
15:0	COUNT	R/W	0x0000	块传输包数量 设置块传输中由 MAXLOAD 位域定义的大小的包的数量。 注意: 仅用于 AUTORQ 位置位时的主机模式中。在设备模式或 AUTORQ 位未置位时此位不起作用。

## 寄存器 163: USB 接收双包缓存禁用寄存器 ( USBRXPKTBUFDIS ) , 偏移量 0x340

OTG A /  
主机

USBRXPKTBUFDIS 是 16 位寄存器, 用来指示哪个接收端点禁用双包缓存功能(见“双包缓存”一节在 1030页)。

OTG B /  
设备

USB 接收双包缓存禁用寄存器 (USBRXPKTBUFDIS)

基址 0x4005.0000  
偏移量 0x340  
类型 R/W, 复位 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								EP7	EP6	EP5	EP4	EP3	EP2	EP1	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
7	EP7	R/W	0	禁用 EP7 的接收双包缓存  值 描述 0 禁用双包缓存。 1 启用双包缓存。
6	EP6	R/W	0	禁用 EP6 的接收双包缓存 与 EP7 描述相同。
5	EP5	R/W	0	禁用 EP5 的接收双包缓存 与 EP7 描述相同。
4	EP4	R/W	0	禁用 EP4 的接收双包缓存 与 EP7 描述相同。
3	EP3	R/W	0	禁用 EP3 的接收双包缓存 与 EP7 描述相同。
2	EP2	R/W	0	禁用 EP2 的接收双包缓存 与 EP7 描述相同。
1	EP1	R/W	0	禁用 EP1 的接收双包缓存 与 EP7 描述相同。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 164: USB 发送双包缓存禁用寄存器 ( USBTXDPKTBUFDIS ) , 偏移量 0x342

OTG A /  
主机

USBTXDPKTBUFDIS 是 16 位寄存器，用来指示哪个发送端点禁止双包缓存功能(见“双包缓存”一节在 1029页)。

OTG B /  
设备

USB 发送双包缓存禁用寄存器 (USBTXDPKTBUFDIS)

基址 0x4005.0000

偏移量 0x342

类型 R/W, 复位 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								EP7	EP6	EP5	EP4	EP3	EP2	EP1	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件，保留位的值在读-修改-写操作过程中应该保持不变。
7	EP7	R/W	0	禁用 EP7 的发送双包缓存  值 描述 0 禁用双包缓存。 1 启用双包缓存。
6	EP6	R/W	0	禁用 EP6 的发送双包缓存 与 EP7 描述相同。
5	EP5	R/W	0	禁用 EP5 的发送双包缓存 与 EP7 描述相同。
4	EP4	R/W	0	禁用 EP4 的发送双包缓存 与 EP7 描述相同。
3	EP3	R/W	0	禁用 EP3 的发送双包缓存 与 EP7 描述相同。
2	EP2	R/W	0	禁用 EP2 的发送双包缓存 与 EP7 描述相同。
1	EP1	R/W	0	禁用 EP1 的发送双包缓存 与 EP7 描述相同。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

## 寄存器 165: USB 外部电源控制寄存器 ( USBEPC ) , 偏移量 0x400

OTG A /  
主机

此 32 位寄存器指定双管脚外部电源接口 ( USB0EPEN 和 USB0PFLT ) 的功能。电源异常输入可能产生由硬件配置寄存器控制的自动动作。由于错误状态可能需要比软件提供的响应要快, 因此自动动作是必要的。

OTG B /  
设备

USB 外部电源控制寄存器 (USBEPIC)

基址 0x4005.0000

偏移量 0x400

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留						PFLTACT	保留	PFLTAEN	PFLTSEN	PFLTEN	保留	EPENDE	EPEN		
类型	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	R/W	R/W	R/W	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:10	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
9:8	PFLTACT	R/W	0x0	电压异常动作 此位域指定当检测到 USB 电源异常时, USB0EPEN 信号如何变化。  值 描述 0x0 无变化 USB0EPEN 由 EPEN 和 EPENDE 位联合控制。 0x1 三态 USB0EPEN 无驱动 ( 三态 )。 0x2 低电平 USB0EPEN 驱动为低。 0x3 高电平 USB0EPEN 驱动为高。
7	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
6	PFLTAEN	R/W	0	电源异常动作启用 此位指定 USB 电源故障是否触发 USB0EPEN 信号驱动状态的自动纠正动作。  值 描述 0 禁能 USB0EPEN 由 EPEN 和 EPENDE 位联合控制。 1 启用 USB0EPEN 输出自动改变为 PFLTACT 域指定的状态。

位/域	名称	类型	复位	描述
5	PFLTSEN	R/W	0	<p>电源故障判断</p> <p>此位指定指示错误状态的 USB0PFLT 输入信号的逻辑判断。补充的状态为非活动状态。</p> <p>值 描述</p> <p>0 低异常 如果 USB0PFLT 驱动为低，则电源向内部发送电源故障信号(如果已通过 PFLTEN 位启用电源异常输入)。</p> <p>1 高异常 如果 USB0PFLT 驱动为高，则电源向内部发送电源故障信号(如果已通过 PFLTEN 位启用电源异常输入)。</p>
4	PFLTEN	R/W	0	<p>启用电源异常输入</p> <p>此位指定 USB0PFLT 输入信号是否用于内部逻辑。</p> <p>值 描述</p> <p>0 没有使用 忽略 USB0PFLT 信号。</p> <p>1 被使用 USB0PFLT 信号是内部使用的。</p>
3	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
2	EPENDE	R/W	0	<p>EPEN驱动启用</p> <p>此位指定是否驱动 USB0EPEN 信号。当驱动时，该信号值由 EPEN 位域指定。当不驱动时，忽略 EPEN 域且将 USB0EPEN 信号置于高阻态。</p> <p>值 描述</p> <p>0 不被驱动 USB0EPEN 信号处于高阻态。</p> <p>1 被驱动 USB0EPEN 信号被驱动为由 EPEN 域指定的逻辑值。</p> <p>在复位时，不驱动 USB0EPEN 信号，因为对外部电源启用的判断是未知的。系统设计者可以先通过一个大电阻将电源置于关闭状态，后面再配置和驱动此输出信号来开启电源供电。</p>

位/域	名称	类型	复位	描述
1:0	EPEN	R/W	0x0	<p>外部电源启用配置 此位域说明和控制 USB0EPEN 信号的逻辑值。</p> <p>值 描述</p> <p>0x0 低电平启用电源 如果 EPENDE 位置位，USB0EPEN 信号驱动为低。</p> <p>0x1 高电平启用电源 如果 EPENDE 位置位，USB0EPEN 信号驱动为高。</p> <p>0x2 如果VBUS为低，低电平启用电源 未识别 A 设备时，USB0EPEN 信号驱动为高。</p> <p>0x3 如果VBUS为高，高电平启用电源 识别出 A 设备时，USB0EPEN 信号驱动为高。</p>

## 寄存器 166: USB 外部电源控制原始中断状态寄存器 ( USBEPCRIS ) , 偏移量 0x404

OTG A /  
主机

此32位寄存器说明外部两个电源接口管脚的未屏蔽的原始中断状态。

USB 外部电源控制原始中断状态寄存器 (USBEPCRIS)

基址 0x4005.0000

偏移量 0x404

类型 RO, 复位 0x0000.0000

OTG B /  
设备

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															PF
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	PF	RO	0	USB电源异常中断状态

### 值 描述

- 1 检测到电源异常状态。
- 0 未产生中断。

对 USBEPCRIS 寄存器中的 PF 位写 1 将该位清零。

**寄存器 167: USB 外部电源控制中断屏蔽寄存器 ( USBEPCIM ) , 偏移量 0x408**

此32位寄存器说明外部两个电源接口管脚的中断屏蔽。

OTG A /

主机

USB 外部电源控制中断屏蔽寄存器 (USBEPICM)

基址 0x4005.0000

偏移量 0x408

类型 R/W, 复位 0x0000.0000

OTG B /

设备

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															PF
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	PF	R/W	0	USB电源异常中断屏蔽
				值 描述
				1 将检测到电源异常的原始中断信号送到中断控制器。
				0 检测到电源异常不影响中断状态。



## 寄存器 168: USB 外部电源控制中断状态和清除寄存器 ( USBEPCISC ) , 偏移量 0x40C

OTG A /  
主机

此32位寄存器说明外部两个电源接口管脚屏蔽的原始中断状态。它还提供了一种清除中断状态的方法。

OTG B /  
设备

### USB 外部电源控制中断状态和清除寄存器 (USBEPCISC)

基址 0x4005.0000

偏移量 0x40C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															PF
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	PF	R/W1C	0	USB电源异常中断状态和清除

#### 值 描述

- 1 假设有中断发送到中断控制器，USBEPKRIS 和 USBEPCIM 寄存器中的 PF 位置位。
- 0 未发生中断，或该中断被屏蔽。

对该位写 1，即可将其清零。将此位清零会将 USBEPCIS 寄存器中的 PF 位清零。

**寄存器 169: USB 设备恢复原始中断状态寄存器 ( USBDRRIS ) , 偏移量 0x410**

OTG A /  
主机

USBDRRIS 32 位寄存器是原始中断状态寄存器。读操作时，此寄存器给出屏蔽前的当前原始中断状态值。写操作对本寄存器无效。

OTG B /  
设备

USB 设备恢复原始中断状态寄存器 (USBDRRIS)

基址 0x4005.0000  
偏移量 0x410  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															RESUME
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	RESUME	RO	0	恢复中断状态  值 描述 1 检测到恢复状态。 0 未产生中断。  向 USBDRISC 寄存器中 RESUME 位写 1 将该位清 0。

## 寄存器 170: USB 设备恢复中断屏蔽寄存器 ( USBDRIM ) , 偏移量 0x414

OTG A /  
主机

USBDRIM 32 位寄存器是屏蔽的中断状态寄存器。对本寄存器进行读操作，可获取相应中断的当前屏蔽值。将某位置位会屏蔽相应的中断，该中断就不会发送到中断控制器。将某位清零会清除相应中断的屏蔽，该中断就会发送到中断控制器。

OTG B /  
设备

## USB 设备恢复中断屏蔽寄存器 (USBDRIM)

基址 0x4005.0000

偏移量 0x414

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															RESUME
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	RESUME	R/W	0	恢复中断掩码
				值 描述
				1 检测到恢复的原始中断信号被送到中断控制器。此位只在检测到挂起 ( USBIS 寄存器中的 SUSPEND 位置位 ) 时才应被置位。
				0 检测到恢复不影响中断状态。

**寄存器 171: USB 设备恢复中断状态和清除寄存器 ( USBDRISC ) , 偏移量 0x418**

OTG A /  
主机

32 位只读寄存器 USBDRISC 是中断清零寄存器。当对本寄存器的有效位写1时，即会清除相应的中断。对本寄存器写0无效。

OTG B /  
设备

USB 设备恢复中断状态和清除寄存器 (USBDRISC)

基址 0x4005.0000  
偏移量 0x418  
类型 W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															RESUME
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	RESUME	R/W1C	0	恢复中断状态和清除  <b>值 描述</b> 1 假设有中断发送到中断控制器，USBDRRIS 和 USBDRICIM 寄存器中的 RESUME 位置位。 0 未发生中断，或该中断被屏蔽。  对该位写 1，即可将其清零。将此位清零会将 USBDRCRIS 寄存器中的 RESUME 位清零。

## 寄存器 172: USB 通用控制和状态寄存器 ( USBGPCS ) , 偏移量 0x41C

OTG A /  
主机

USBGPCS 提供内部 ID 信号的状态。

OTG B /  
设备

注意: 在 OTG 模式, 由于 USB0VBUS 和 USB0ID 是 USB 控制器专用的管脚, 所以无需配置, 可直接连接到 USB 连接器的 VBUS 和 ID 信号。如果 USB 控制器用作专用的主机或设备, USB 通用控制和状态 (USBGPCS) 寄存器中的 DEVMODOTG 和 DEVMOD 位可用于将 USB0VBUS 和 USB0ID 输入连接到内部固定电平, 从而释放 PB0 和 PB1 管脚, 以用于 GPIO。当用作自供电的设备时, 需要监测 VBUS, 来确定主机是否断开 VBUS, 以及自供电设备是否禁用 D+/D- 上拉电阻。此功能可通过将一个标准GPIO连接到VBUS实现。

## USB 通用控制和状态寄存器 (USBGPCS)

基址 0x4005.0000  
偏移量 0x41C  
类型 R/W, 复位 0x0000.0003

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留															DEVMODOTG	DEVMOD
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
1	DEVMODOTG	R/W	1	<p>启用设备模式</p> <p>此位启用 DEVMOD 位, 用来控制 OTG 模式中的内部 ID 信号状态。</p> <p>值 描述</p> <p>0 模式由内部ID的状态来确定。</p> <p>1 此位启用 DEVMOD 位, 用来控制内部 ID 信号。</p>
0	DEVMOD	R/W	1	<p>设备模式</p> <p>当DEVMODOTG 位置位时, 此位指定主机模式和 OTG 模式里内部 ID 信号的状态。</p> <p>设备模式中, 此位被忽略 (假定置位)。</p> <p>值 描述</p> <p>0 主机模式</p> <p>1 设备模式</p>

### 寄存器 173: USB VBUS 浮动控制寄存器 ( USBVDC ) , 偏移量 0x430

OTG A /  
主机

此32位寄存器用来控制是否屏蔽当设备接入Host主机控制器时带来的冲击电流对VBUS的影响。冲击电流会造成VBUS电压浮动，从而引起USB控制器的误动作。启用 VBDEN 位，USB 主机控制器允许 VBUS 电压下降到低于 VBUS 有效电压 (4.75 V) 但不低于 AVaid 电压 (2.0 V) 长达 65 微秒，而不产生 VBUSERR 中断信号。如果不启用VBDEN位，VBUS电压上出现任何毛刺时，USB Host 控制器将切断VBUS，重新枚举设备。

#### USB VBUS 浮动控制寄存器 (USBVDC)

基址 0x4005.0000  
偏移量 0x430  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															VBDEN
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	VBDEN	R/W	0	VBUS电压浮动(Droop)启用  值 描述 0 无影响。 1 当 VBUS 电压下降到低于 VBUS 有效电压 (4.75 V) 但不低于 AVaid 电压 (2.0 V) 长达 65 微妙，VBUSVALID 的变化将被屏蔽。此时 VBUS 状态为 VBUSVALID。

## 寄存器 174: USB VBUS 浮动控制原始中断状态寄存器 ( USBVDCRIS ) , 偏移量 0x434

OTG A /  
主机

此32位寄存器说明VBUS电压浮动(Droop)超过65ms限制的原始中断状态。

USB VBUS 浮动控制原始中断状态寄存器 (USBVDCRIS)

基址 0x4005.0000

偏移量 0x434

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															VD
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	VD	RO	0	VBUS浮动(Droop)原始中断状态

### 值 描述

- 1 检测到VBUS电压浮动(droop)持续65ms
- 0 未产生中断。

向 USBVDCISC 寄存器中 VD 位写 1 将该位清零。

## 寄存器 175: USB VBUS 浮动控制中断屏蔽寄存器 ( USBVDCIM ) , 偏移量 0x438

OTG A /  
主机

此32位寄存器说明VBUS浮动(Droop)的中断屏蔽。

USB VBUS 浮动控制中断屏蔽寄存器 (USBVDCIM)

基址 0x4005.0000

偏移量 0x438

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															VD
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品,保留位的值在读取-修改-写入操作过程中应当保持不变。
0	VD	R/W	0	VBUS浮动(Droop)中断屏蔽

## 值 描述

- 1 检测到VBUS浮动(Droop)的原始中断信号发送到中断控制器。
- 0 检测到VBUS浮动(Droop)不影响中断状态。



## 寄存器 176: USB VBUS 浮动控制中断状态和清除寄存器 ( USBVDCISC ) , 偏移量 0x43C

OTG A /  
主机

此32位寄存器说明VBUS浮动(Droop)的中断状态，并提供清除此中断状态的方法。

### USB VBUS 浮动控制中断状态和清除寄存器 (USBVDCISC)

基址 0x4005.0000

偏移量 0x43C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															VD
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	VD	R/W1C	0	VBUS浮动(Droop)中断状态和清除

#### 值 描述

- 1 假设有中断发送到中断控制器，USBVDCRIS 和 USBVDCIM 寄存器中的 VD 位置位。
- 0 未发生中断，或该中断被屏蔽。

对该位写 1，即可将其清零。将此位清零会将 USBVDCRIS 寄存器中的 VD 位清零。

## 寄存器 177: USB ID 有效检测原始中断状态寄存器 ( USBIDVRIS ) , 偏移量 0x444

OTG

此32位寄存器说明ID检测是否有效的原始中断信号的状态。

USB ID 有效检测原始中断状态寄存器 (USBIDVRIS)

基址 0x4005.0000  
 偏移量 0x444  
 类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															ID
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	ID	RO	0	ID有效检测原始中断状态

## 值 描述

- 1 检测到有效ID。
- 0 未产生中断。

向 USBIDVISC 寄存器中的 ID 位写 1 将该位清零。

## 寄存器 178: USB ID 有效检测中断屏蔽寄存器 ( USBIDVIM ) , 偏移量 0x448

OTG

此32位寄存器说明ID有效检测的中断屏蔽。

USB ID 有效检测中断屏蔽寄存器 (USBIDVIM)

基址 0x4005.0000

偏移量 0x448

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															ID
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	ID	R/W	0	ID有效检测中断屏蔽

## 值 描述

- 1 将检测到ID有效的原始中断信号发送到中断控制器。
- 0 检测到ID有效不影响中断状态

## 寄存器 179: USB ID 有效检测中断状态与清除寄存器 ( USBIDVISC ) , 偏移量 0x44C

OTG

此 32 位寄存器指定该 ID 有效检测的屏蔽中断状态。它还提供了一种清除中断状态的方法。

### USB ID 有效检测中断状态与清除寄存器 (USBIDVISC)

基址 0x4005.0000  
偏移量 0x44C  
类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															ID
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
0	ID	R/W1C	0	ID有效检测中断状态与清除

#### 值 描述

- 1 假设有中断发送到中断控制器，则 USBIDVRIS 和 USBIDVIM 寄存器中的 ID 位会被置位。
- 0 未发生中断，或该中断被屏蔽。

对该位写 1，即可将其清零。如果将此位清零，则寄存器 USBIDVRIS 中的 ID 位也会被清零。

## 寄存器 180: USB DMA 选择寄存器 ( USBDMASEL ), 偏移量 0x450

OTG A /  
主机此 32 位寄存器指定哪个端点映射到 6 个分配 USB 的  $\mu$ DMA 通道上, 通道分配的更多信息见表 9-1 在 536 页。OTG B /  
设备

USB DMA 选择寄存器 (USBDMASEL)

基址 0x4005.0000

偏移量 0x450

类型 R/W, 复位 0x0033.2211

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								DMACTX				DMACRX			
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMABTX				DMABRX				DMAATX				DMAARX			
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1

位/域	名称	类型	复位	描述																				
31:24	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。																				
23:20	DMACTX	R/W	0x3	DMA C发送选择 指定在 $\mu$ DMA 通道 5 (主功能) 上第 3 个端点的发送映射。  <table border="1"> <thead> <tr> <th>值</th><th>描述</th></tr> </thead> <tbody> <tr><td>0x0</td><td>保留</td></tr> <tr><td>0x1</td><td>端点 1 发送</td></tr> <tr><td>0x2</td><td>端点 2 发送</td></tr> <tr><td>0x3</td><td>端点 3 发送</td></tr> <tr><td>0x4</td><td>端点 4 发送</td></tr> <tr><td>0x5</td><td>端点 5 发送</td></tr> <tr><td>0x6</td><td>端点 6 发送</td></tr> <tr><td>0x7</td><td>端点 7 发送</td></tr> <tr><td>0x8 - 0xF</td><td>保留</td></tr> </tbody> </table>	值	描述	0x0	保留	0x1	端点 1 发送	0x2	端点 2 发送	0x3	端点 3 发送	0x4	端点 4 发送	0x5	端点 5 发送	0x6	端点 6 发送	0x7	端点 7 发送	0x8 - 0xF	保留
值	描述																							
0x0	保留																							
0x1	端点 1 发送																							
0x2	端点 2 发送																							
0x3	端点 3 发送																							
0x4	端点 4 发送																							
0x5	端点 5 发送																							
0x6	端点 6 发送																							
0x7	端点 7 发送																							
0x8 - 0xF	保留																							
19:16	DMACRX	R/W	0x3	DMA C接收选择 指定在 $\mu$ DMA 通道 4 (主功能) 上第 3 个端点的接收和发送映射。  <table border="1"> <thead> <tr> <th>值</th><th>描述</th></tr> </thead> <tbody> <tr><td>0x0</td><td>保留</td></tr> <tr><td>0x1</td><td>端点 1 接收</td></tr> <tr><td>0x2</td><td>端点 2 接收</td></tr> <tr><td>0x3</td><td>端点 3 接收</td></tr> <tr><td>0x4</td><td>端点 4 接收</td></tr> <tr><td>0x5</td><td>端点 5 接收</td></tr> <tr><td>0x6</td><td>端点 6 接收</td></tr> <tr><td>0x7</td><td>端点 7 接收</td></tr> <tr><td>0x8 - 0xF</td><td>保留</td></tr> </tbody> </table>	值	描述	0x0	保留	0x1	端点 1 接收	0x2	端点 2 接收	0x3	端点 3 接收	0x4	端点 4 接收	0x5	端点 5 接收	0x6	端点 6 接收	0x7	端点 7 接收	0x8 - 0xF	保留
值	描述																							
0x0	保留																							
0x1	端点 1 接收																							
0x2	端点 2 接收																							
0x3	端点 3 接收																							
0x4	端点 4 接收																							
0x5	端点 5 接收																							
0x6	端点 6 接收																							
0x7	端点 7 接收																							
0x8 - 0xF	保留																							

位/域	名称	类型	复位	描述
15:12	DMABTX	R/W	0x2	DMA B发送选择 指定在 $\mu$ DMA 通道 3 (主功能) 上第 2 个端点的发送映射。 位定义与 DMACTX 域的定义相同。
11:8	DMABRX	R/W	0x2	DMA B接收选择 指定在 $\mu$ DMA 通道 2 (主功能) 上第 2 个端点的接收映射。 位域定义与 DMACRX 位域中的定义相同。
7:4	DMAATX	R/W	0x1	DMA A发送选择 指定在 $\mu$ DMA 通道 1 (主功能) 上第 1 个端点的发送映射。 位定义与 DMACTX 域的定义相同。
3:0	DMAARX	R/W	0x1	DMA A接收选择 指定在 $\mu$ DMA 通道 0 (主功能) 上第 1 个端点的接收映射。 位域定义与 DMACRX 位域中的定义相同。

## 寄存器 181: USB 外设属性寄存器 (USBPP), 偏移量 0xFC0

该 USBPP 寄存器提供关于 USB 模块属性的信息。

### USB 外设属性寄存器 (USBPP)

基址 0x4005.0000

偏移量 0xFC0

类型 RO, 复位 0x0000.10D0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNT				USB				保留	PHY	TYPE					
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:8	ECNT	RO	0x10	端点计数 该域指定提供的端点数量。
7:6	USB	RO	0x3	USB 能力  值 描述 0x0 NA 没有 USB 0x1 DEVICE 设备模式 0x2 主机 设备或主机 0x3 OTG 设备、主机或 OTG
5	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
4	PHY	RO	0x1	PHY 存在  值 描述 1 PHY 与 USB MAC 合为一体。 0 PHY 未与 USB MAC 合为一体。
3:0	TYPE	RO	0x0	控制器类型  值 描述 0x0 第一代 USB 控制器。 0x1 - 0xF 保留

## 19 模拟比较器

模拟比较器是一个外设，它能比较两个模拟电压的大小，并通过自身提供的逻辑输出端将比较结果以信号的形式输出。

注意：不是所有比较器都可以选择驱动输出管脚。详见“信号描述”在 1137页。

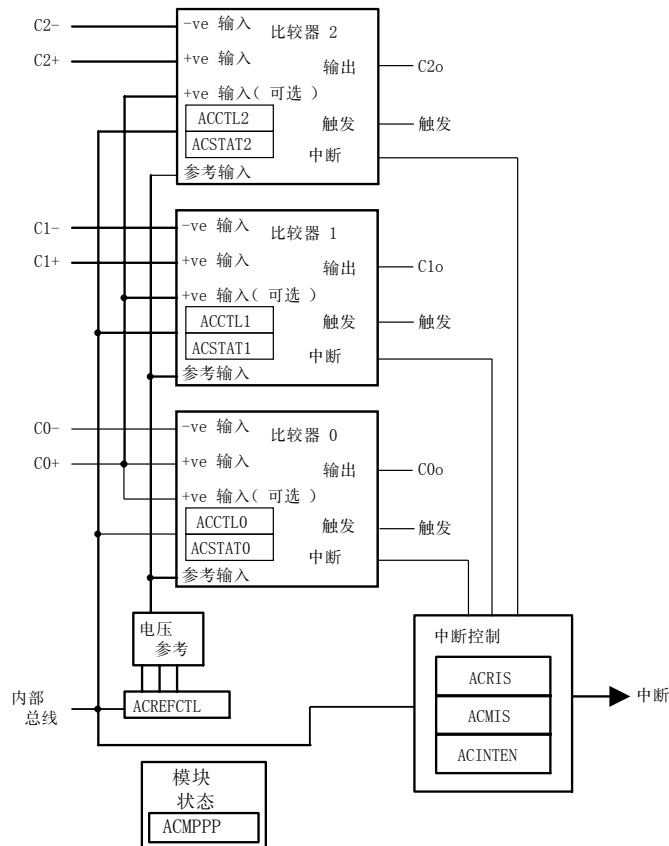
比较器可以向器件管脚提供输出，以替换板上的模拟比较器。比较器也可以通过中断通知应用或触发 ADC 开始捕获采样序列。中断产生逻辑和 ADC 触发是各自独立的。这就意味着，中断可以在上升沿产生，而 ADC 在下降沿触发。

该 Stellaris<sup>®</sup> LM4F232H5QD 微控制器提供三个独立集成的模拟比较器。它们具有如下功能：

- 可以比较外部输入管脚和外部输入管脚或内部可编程的参考电压
- 比较器可将测试电压与下面的其中一种电压相比较
  - 单个外部独立的参考电压
  - 共用的外部参考电压
  - 公用的内部参考电压

### 19.1 结构图

图 19-1. 模拟比较器模块的结构图





## 19.2 信号描述

下表列出了模拟比较器的外部信号及其功能描述。模拟比较器输出信号是某些 GPIO 管脚的复用功能，复位时，它默认为 GPIO 信号。表中列标题名为‘管脚复用/分配’的列列出了可以被用作模拟比较器的管脚。通过将 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618页) 中的 AFSEL 位置位来选择模拟比较器功能。括号中的数字是必须写入 GPIO 端口控制 (GPIOPCTL) 寄存器 (636页) 中的 PMCN 位的编码，以便将管脚配置为模拟比较器功能。通过将 GPIO 数字使能 (GPIODEN) 寄存器中的 DEN 位清零即可配置正极和负极的输入信号。关于如何配置 GPIO 的更多信息，请参考“通用输入/输出端口 (GPIO)”在 597页。

表 19-1. 模拟比较器 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
C0+	34	PC6	I	模拟	模拟比较器 0 正极输入。
C0-	33	PC7	I	模拟	模拟比较器 0 负极输入。
C0o	62 112	PF0 (9) PK4 (8)	O	TTL	模拟比较器 0 输出。
C1+	35	PC5	I	模拟	模拟比较器 1 正极输入。
C1-	36	PC4	I	模拟	模拟比较器 1 负极输入。
C1o	63 111	PF1 (9) PK5 (8)	O	TTL	模拟比较器 1 输出。
C2+	127	PJ4	I	模拟	模拟比较器 2 正极输入。
C2-	128	PJ5	I	模拟	模拟比较器 2 负极输入。
C2o	64 110	PF2 (9) PK6 (8)	O	TTL	模拟比较器 2 输出。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 19.3 功能描述

比较器比较 VIN- 和 VIN+ 的输入，然后输出 VOUT。

$$VIN- < VIN+, VOUT = 1$$

$$VIN- > VIN+, VOUT = 0$$

如图 19-2 在 1138页 所示，。VIN 的输入源是一个外部输入，Cn-，其中 n 指模拟比较器编号。VIN+ 输入源除了外部的输入 Cn+ 之外，还可以是 C0+ 或者是一个内部参考源 V<sub>IREF</sub>。



在低电平模式下， $V_{IREF}$  阈值电压始于 0 V，并以理想恒电压阶跃  $V_{DDA}/22.12$  增加。有关每种模式的理想的  $V_{IREF}$  阶跃电压以及 RNG 和 VREF 域对其有何作用，请参阅表 19-2。

表 19-2. 内部参考电压和 ACREFCTL 域值

ACREFCTL 寄存器		基于 VREF 域值的输出参考电压
EN 位值	RNG 位值	
EN=0	RNG=X	任何 VREF 值均为 0 V (GND)。建议使用 RNG = 1 和 VREF = 0 来降低参考接地的噪声。
EN=1	RNG=0	$V_{IREF}$ 高电平：16 个电压阈值，索引为 VREF = 0x0 ..0xF 理想启动电压 (VREF=0)： $V_{DDA} / 4.2$ 理想阶跃电压： $V_{DDA} / 29.4$ 理想 $V_{IREF}$ 阈值： $V_{IREF}(VREF) = V_{DDA} / 4.2 + VREF * (V_{DDA} / 29.4)$ ，其中 VREF = 0x0 ..0xF 最小和最大 $V_{IREF}$ 阈值请参考表 19-3 在 1139 页。
	RNG=1	$V_{IREF}$ 下限：16 个电压阈值，索引为 VREF = 0x0 ..0xF 理想启动电压 (VREF=0)：0 V 理想阶跃电压： $V_{DDA} / 22.12$ 理想 $V_{IREF}$ 阈值： $V_{IREF}(VREF) = VREF * (V_{DDA} / 22.12)$ ，其中 VREF = 0x0 ..0xF 最小和最大 $V_{IREF}$ 阈值请参考表 19-4 在 1140 页。

请注意表 19-2 中的值是  $V_{IREF}$  阈值的理想值。实际上，每个阈值阶跃的这些值都将在最小值和最大值之间变动，具体取决于进程和温度。每个阶跃的最小值和最大值通过以下的公式计算：

- $V_{IREF}(VREF)$  [最小值] = 理想  $V_{IREF}(VREF) - (\text{理想阶跃电压} - 2 \text{ mV}) / 2$
- $V_{IREF}(VREF)$  [最大值] = 理想  $V_{IREF}(VREF) + (\text{理想阶跃电压} - 2 \text{ mV}) / 2$

高电平和低电平模式下， $V_{DDA} = 3.3\text{V}$  时的最小和最大  $V_{IREF}$  值示例请见表 19-3 和表 19-4。请注意这些示例只适用于  $V_{DDA} = 3.3\text{V}$ ； $V_{DDA}$  发生变化时，数值将按比例增加和减少。

表 19-3. 模拟比较器参考电压特性， $V_{DDA} = 3.3\text{V}$ ，EN=1 且 RNG=0

VREF 值	$V_{IREF}$ 最小值	理想 $V_{IREF}$	$V_{IREF}$ 最大值	单位
0x0	0.731	0.786	0.841	V
0x1	0.843	0.898	0.953	V
0x2	0.955	1.010	1.065	V
0x3	1.067	1.122	1.178	V
0x4	1.180	1.235	1.290	V
0x5	1.292	1.347	1.402	V
0x6	1.404	1.459	1.514	V
0x7	1.516	1.571	1.627	V
0x8	1.629	1.684	1.739	V
0x9	1.741	1.796	1.851	V
0xA	1.853	1.908	1.963	V
0xB	1.965	2.020	2.076	V
0xC	2.078	2.133	2.188	V
0xD	2.190	2.245	2.300	V
0xE	2.302	2.357	2.412	V

表 19-3. 模拟比较器参考电压特性， $V_{DDA} = 3.3V$ ， $EN = 1$  且  $RNG = 0$  (续)

VREF 值	$V_{IREF}$ 最小值	理想 $V_{IREF}$	$V_{IREF}$ 最大值	单位
0xF	2.414	2.469	2.525	V

表 19-4. 模拟比较器参考电压特性， $V_{DDA} = 3.3V$ ， $EN = 1$  且  $RNG = 1$ 

VREF 值	$V_{IREF}$ 最小值	理想 $V_{IREF}$	$V_{IREF}$ 最大值	单位
0x0	0.000	0.000	0.074	V
0x1	0.076	0.149	0.223	V
0x2	0.225	0.298	0.372	V
0x3	0.374	0.448	0.521	V
0x4	0.523	0.597	0.670	V
0x5	0.672	0.746	0.820	V
0x6	0.822	0.895	0.969	V
0x7	0.971	1.044	1.118	V
0x8	1.120	1.193	1.267	V
0x9	1.269	1.343	1.416	V
0xA	1.418	1.492	1.565	V
0xB	1.567	1.641	1.715	V
0xC	1.717	1.790	1.864	V
0xD	1.866	1.939	2.013	V
0xE	2.015	2.089	2.162	V
0xF	2.164	2.238	2.311	V

## 19.4 初始化及配置

下面的例子展示了应如何配置模拟比较器才能从内部寄存器中读回其输出值。

1. 向系统控制模块中的 RCGCACMP 寄存器写入 0x0000.0001 来启用模拟比较器时钟 (参见 316 页)。
2. 通过 RCGCGPIO 寄存器 (参考 302 页) 启用相应 GPIO 模块的时钟。欲了解需要启用哪些 GPIO 端口，请参阅表 23-5 在 1280 页。
3. 在 GPIO 模块中，启用输入信号的相应 GPIO 端口/管脚相关的启用为，将其作为 GPIO 输入。欲了解需要配置哪些 GPIO，请参阅表 23-4 在 1271 页。
4. 配置寄存器 GPIOPCTL 中 PMCn 域，将模拟比较器输出信号分配给相应的管脚 (参见 636 页和表 23-5 在 1280 页)。
5. 向 ACREFCTL 寄存器写入 0x0000.030C，从而将内部电压参考配置为 1.65 V。
6. 向 ACCTLn 寄存器写入 0x0000.040C，将参考选为内部电压参考源，并且不将输出反相。
7. 延迟 10  $\mu$ s。
8. 读取 ACSTATn 寄存器的 OVAL 值，获得比较器的输出值。

改变 C- 上输入信号的电平，观察 OVAL 值的变化。

## 19.5 寄存器映射

表 19-5 在 1141 页 中列出了比较器寄存器。表中所列偏移量是寄存器地址相对于模拟比较器基址 0x4003.C000 的 16 进制增量。注意，在对这些寄存器编程之前必须先启用模拟比较器的时钟（参考 316 页）。模拟比较器模块时钟启用后，必须等待至少 3 个系统时钟才可访问模拟比较器寄存器。

表 19-5. 模拟比较器 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x00	ACMIS	R/W1C	0x0000.0000	模拟比较器屏蔽中断状态寄存器	1142
0x010	ACREFCTL	R/W	0x0000.0000	模拟比较器参考电压控制寄存器	1145
0x020	ACSTAT0	RO	0x0000.0000	模拟比较器状态寄存器 0	1146
0x024	ACCTL0	R/W	0x0000.0000	模拟比较器控制寄存器 0	1147
0x04	ACRIS	RO	0x0000.0000	模拟比较器原始中断状态寄存器	1143
0x040	ACSTAT1	RO	0x0000.0000	模拟比较器状态寄存器 1	1146
0x044	ACCTL1	R/W	0x0000.0000	模拟比较器控制寄存器 1	1147
0x060	ACSTAT2	RO	0x0000.0000	模拟比较器状态寄存器 2	1146
0x064	ACCTL2	R/W	0x0000.0000	模拟比较器控制寄存器 2	1147
0x08	ACINTEN	R/W	0x0000.0000	模拟比较器中断启用寄存器	1144
0xFC0	ACMPPP	RO	0x0007.0007	模拟比较器外设属性寄存器	1149

## 19.6 寄存器描述

本章剩余的部分列出并详细描述了模拟比较器的每个寄存器，是以地址的偏移量递增的顺序排列的。

## 寄存器 1: 模拟比较器屏蔽中断状态寄存器 (ACMIS), 偏移量 0x00

该寄存器汇总了比较器的中断状态 (被屏蔽的)。

### 模拟比较器屏蔽中断状态寄存器 (ACMIS)

基址 0x4003.C000  
偏移量 0x00  
类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												IN2	IN1	IN0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2	IN2	R/W1C	0	比较器 2 屏蔽后的中断状态  值 描述 1 ACRIS 和 ACINTEN 的 IN2 位被置位, 向中断控制器提交了中断。 0 未发生中断, 或该中断被屏蔽。  对该位写入 1, 即可将其清零。设置此位会清除寄存器 ACRIS 中的 IN2 位。
1	IN1	R/W1C	0	比较器 1 屏蔽后的中断状态  值 描述 1 ACRIS 和 ACINTEN 的 IN1 位被置位, 向中断控制器提交了中断。 0 未发生中断, 或该中断被屏蔽。  对该位写入 1, 即可将其清零。设置此位会清除寄存器 ACRIS 中的 IN1 位。
0	IN0	R/W1C	0	比较器 0 屏蔽后的中断状态  值 描述 1 ACRIS 和 ACINTEN 的 IN0 位被置位, 向中断控制器提交了中断。 0 未发生中断, 或该中断被屏蔽。  对该位写入 1, 即可将其清零。设置此位会清除寄存器 ACRIS 中的 IN0 位。

## 寄存器 2: 模拟比较器原始中断状态寄存器 (ACRIS), 偏移量 0x04

该寄存器汇总了比较器的 (原始) 中断状态。必须启用该寄存器中的这些位, 才可以使用 ACINTEN 寄存器来产生中断。

### 模拟比较器原始中断状态寄存器 (ACRIS)

基址 0x4003.C000  
偏移量 0x04  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													IN2	IN1	IN0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2	IN2	RO	0	比较器 2 中断状态  值 描述 1 模拟比较器 2 产生了一个通过 ACCTL2 的 ISEN 位来配置的中断。 0 未产生中断。  向 ACMIS 寄存器中 IN2 位写入 1 来将其清零。
1	IN1	RO	0	比较器 1 中断状态  值 描述 1 模拟比较器 1 产生了一个通过 ACCTL1 的 ISEN 位来配置的中断。 0 未产生中断。  向 ACMIS 寄存器中 IN1 位写入 1 来将其清零。
0	IN0	RO	0	比较器 0 中断状态  值 描述 1 模拟比较器 0 产生了一个通过 ACCTL0 的 ISEN 位来配置的中断。 0 未产生中断。  向 ACMIS 寄存器中 IN0 位写入 1 来将其清零。

### 寄存器 3: 模拟比较器中断启用寄存器 ( ACINTEN ) , 偏移量 0x08

该寄存器为比较器启用中断。

#### 模拟比较器中断启用寄存器 (ACINTEN)

基址 0x4003.C000  
偏移量 0x08  
类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													IN2	IN1	IN0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:3	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
2	IN2	R/W	0	比较器 2 中断启用  值 描述 1 模拟比较器 2 原始中断信号被送到中断控制器。 0 比较器 2 中断不影响中断状态。
1	IN1	R/W	0	比较器 1 中断启用  值 描述 1 模拟比较器 1 原始中断信号被送到中断控制器。 0 比较器 1 中断不影响中断状态。
0	IN0	R/W	0	比较器 0 中断启用  值 描述 1 模拟比较器 0 原始中断信号被送到中断控制器。 0 比较器 0 中断不影响中断状态。



## 寄存器 4: 模拟比较器参考电压控制寄存器 (ACREFCTL), 偏移量 0x010

该寄存器指示阶梯电阻是否已上电, 以及该电阻的范围和抽头。

### 模拟比较器参考电压控制寄存器 (ACREFCTL)

基址 0x4003.C000

偏移量 0x010

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留						EN	RNG	保留				VREF			
类型	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:10	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
9	EN	R/W	0	阶梯电阻启用  值 描述 0 阶梯电阻未上电。 1 阶梯电阻上电。阶梯电阻被连接到 $V_{DDA}$ 。  复位时, 该位将清零, 使得内部参考在未使用时仅消耗最小的功率。
8	RNG	R/W	0	阶梯电阻范围  值 描述 0 内部参考电阻的理想阶跃电压为 $V_{DDA} / 29.4$ 。 1 内部参考电阻的理想阶跃电压为 $V_{DDA} / 22.12$ 。
7:4	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3:0	VREF	R/W	0x0	阶梯电阻参考电压  VREF 位域指示的是通过模拟复用器的阶梯电阻的抽头。每个抽头所对应的电压是可用于比较的内部参考电压。有关输出参考电压的一些例子请见 表 19-2 在 1139页。

**寄存器 5: 模拟比较器状态寄存器 0 ( ACSTAT0 ) , 偏移量 0x020**

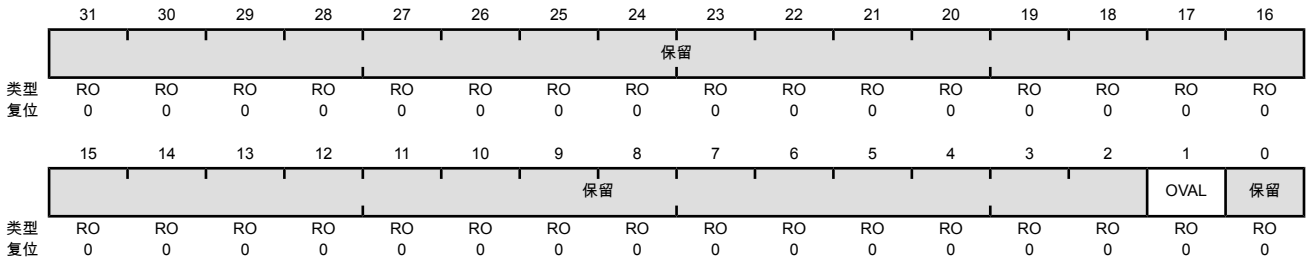
**寄存器 6: 模拟比较器状态寄存器 1 ( ACSTAT1 ) , 偏移量 0x040**

**寄存器 7: 模拟比较器状态寄存器 2 ( ACSTAT2 ) , 偏移量 0x060**

这些寄存器指示比较器的当前输出值。

模拟比较器状态寄存器 n ( ACSTATn )

基址 0x4003.C000  
偏移量 0x020  
类型 RO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
1	OVAL	RO	0	比较器输出值  值 描述 0 VIN- > VIN+ 1 VIN- < VIN+  VIN - 是在 Cn- 管脚上的电压。VIN+ 是在 Cn+ 管脚，C0+ 管脚上的电压；或者由 ACCTL 寄存器中的 ASRCP 位所定义的内部参考电压 (V <sub>IREF</sub> )。
0	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。

寄存器 8: 模拟比较器控制寄存器 0 ( ACCTL0 ) , 偏移量 0x024

寄存器 9: 模拟比较器控制寄存器 1 ( ACCTL1 ) , 偏移量 0x044

寄存器 10: 模拟比较器控制寄存器 2 ( ACCTL2 ) , 偏移量 0x064

这些寄存器用来配置比较器的输入和输出。

#### 模拟比较器控制寄存器 n ( ACCTLn )

基址 0x4003.C000

偏移量 0x024

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留				TOEN	ASRCP			保留	TSLVAL	TSEN		ISLVAL	ISEN		CINV	保留
类型	RO	RO	RO	RO	R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
11	TOEN	R/W	0	触发输出启用  值 描述 0 ADC 事件被阻止，不发送到 ADC。 1 ADC 事件被发送到 ADC。
10:9	ASRCP	R/W	0x0	模拟正电压源 ASRCP 位域指定了到比较器 VIN+ 端口的输入电压源。该位域的编码如下：  值 描述 0x0 Cn+ 管脚的值 0x1 C0+ 管脚的值 0x2 内部参考电压 (V <sub>IREF</sub> ) 0x3 保留
8	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7	TSLVAL	R/W	0	电平触发值  值 描述 0 比较器输出低电平触发 ADC 事件。 1 比较器输出高电平触发 ADC 事件。

位/域	名称	类型	复位	描述
6:5	TSEN	R/W	0x0	<p>触发方式</p> <p>TSEN 域指定了产生 ADC 事件的比较器输出的检测方式。检测方式如下所述：</p> <p>值 描述</p> <p>0x0 电平触发，见 TSLVAL</p> <p>0x1 下降沿</p> <p>0x2 上升沿</p> <p>0x3 上升沿、下降沿均可</p>
4	ISLVAL	R/W	0	<p>电平触发中断值</p> <p>值 描述</p> <p>0 比较器输出低电平触发中断。</p> <p>1 比较器输出高电平触发中断。</p>
3:2	ISEN	R/W	0x0	<p>中断触发方式</p> <p>ISEN 位域指定了产生中断的比较器输出的检测方式。检测方式如下所述：</p> <p>值 描述</p> <p>0x0 电平触发，见 ISLVAL</p> <p>0x1 下降沿</p> <p>0x2 上升沿</p> <p>0x3 上升沿、下降沿均可</p>
1	CINV	R/W	0	<p>比较器输出翻转</p> <p>值 描述</p> <p>0 比较器输出不改变</p> <p>1 比较器输出之前由硬件进行了翻转。</p>
0	保留	RO	0	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>

## 寄存器 11: 模拟比较器外设属性寄存器 (ACMPPP), 偏移量 0xFC0

ACMPPP 寄存器提供关于模拟比较器模块属性的信息。

### 模拟比较器外设属性寄存器 (ACMPPP)

基址 0x4003.C000

偏移量 0xFC0

类型 RO, 复位 0x0007.0007

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留													C2O	C1O	C0O
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													CMP2	CMP1	CMP0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

位/域	名称	类型	复位	描述
31:19	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
18	C2O	RO	0x1	存在比较器输出 2  值 描述 1 存在比较器输出 2。 0 不存在比较器输出 2。
17	C1O	RO	0x1	存在比较器输出 1  值 描述 1 存在比较器输出 1。 0 不存在比较器输出 1。
16	C0O	RO	0x1	存在比较器输出 0  值 描述 1 存在比较器输出 0。 0 不存在比较器输出 0。
15:3	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
2	CMP2	RO	0x1	存在比较器 2  值 描述 1 存在比较器 2。 0 不存在比较器 2。

---

位/域	名称	类型	复位	描述
1	CMP1	RO	0x1	存在比较器 1  值 描述 1 存在比较器 1。 0 不存在比较器 1。
0	CMP0	RO	0x1	存在比较器 0  值 描述 1 存在比较器 0。 0 不存在比较器 0。

## 20 脉宽调制器 (PWM)

脉宽调制 (PWM) 是一项功能强大的技术，它是一种对模拟信号电平进行数字化编码的方法。在脉宽调制中使用高分辨率计数器来产生方波，并且可以通过调整方波的占空比来对模拟信号电平进行编码。PWM 通常用于开关电源和运动控制。

该 Stellaris® 微控制器包含两个 PWM 模块，每个模块有四个 PWM 发生器和一个控制模块，总共 16 个 PWM 输出。控制模块决定了 PWM 信号的极性，以及将哪个信号传递到管脚。

每个 PWM 发生器模块产生两个 PWM 信号，这两个信号基于同一个定时器和频率，也可以是编程产生独立的信号，如插入了死区延时互补信号。PWM 发生器模块的输出信号是 pwmA' 和 pwmB'。在发送到 MnPWM0 和 MnPWM1 或 MnPWM2 和 MnPWM3 等器件管脚之前，这两个输出信号由输出控制模块管理。

每个 Stellaris PWM 模块具有极大地灵活性，它可以产生简单的 PWM 信号，如简易充电泵需要的信号；也可以产生带死区延迟的成对 PWM 信号，如供半-H 桥驱动电路使用的信号。3 个发生器模块也可产生 3 相反相器桥所需的 6 通道栅极控制信号。

每个 PWM 发生模块具有以下特性：

- 4 个故障条件处理输入，能快速实现低延迟关闭，避免马达在被可控之前产生破坏，共计 8 个输入。
- 1 个 16 位的计数器
  - 运行在递减或递增/递减模式
  - 输出频率由一个 16 位的加载值控制
  - 可同步更新加载值
  - 当计数器的值到达零或者加载值的时候生成输出信号
- 2 个 PWM 比较器
  - 比较器值的更新可以同步
  - 在匹配的时候产生输出信号
- PWM 信号发生器
  - 根据计数器和 PWM 比较器的输出信号来产生 PWM 输出信号
  - 可产生两个独立的 PWM 信号
- 死区发生器
  - 产生 2 个带有可编程死区延时的 PWM 信号，适合驱动半 H 桥 ( half-H bridge )
  - 可以被旁路，不修改输入的 PWM 信号
- 可启动一个 ADC 采样序列

控制模块决定了 PWM 信号的极性，以及将哪个信号传递到管脚。PWM 发生器模块的输出在传递到器件管脚之前由输出控制模块管理。PWM 控制模块具有以下特性：

- 每个 PWM 信号都具有 PWM 输出启用
- 每个 PWM 信号都可以选择将输出反相 ( 极性控制 )
- 每个 PWM 信号都可以选择进行故障处理
- PWM 发生器模块的定时器同步
- PWM 发生器模块的定时器/比较器更新同步
- PWM 发生器模块的定时器/比较器更新的扩展 PWM 同步
- PWM 发生器模块中断状态汇总
- 多路故障信号，可编程极性和过滤的扩展故障处理
- PWM 发生器可独立操作或者与其它发生器同步操作

## 20.1 结构图

图 20-1 在 1152 页给出了 Stellaris PWM 模块的结构图，图 20-2 在 1153 页给出了一个更加详细的 Stellaris PWM 信号发生器结构图。该 LM4F232H5QD 控制器包含两个 PWM 模块，每个模块有四个发生器模块，可以产生 8 个独立的 PWM 信号或 4 对插入了死区延时的 PWM 信号对。

图 20-1. PWM 模块结构图

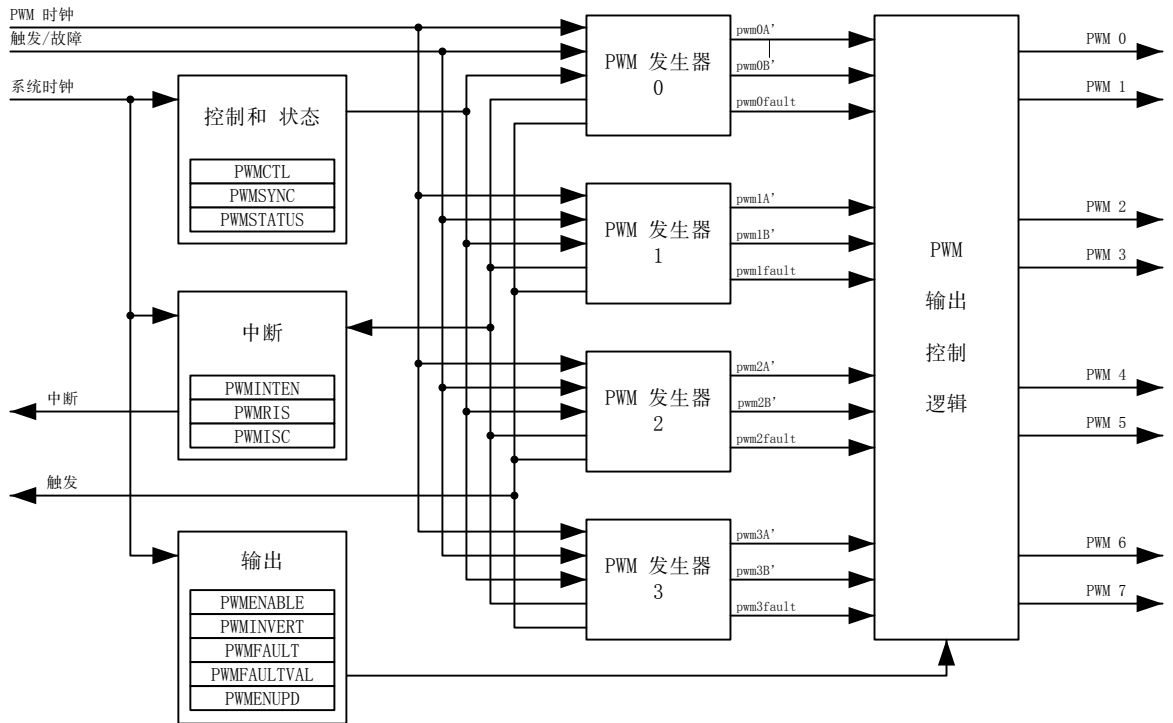
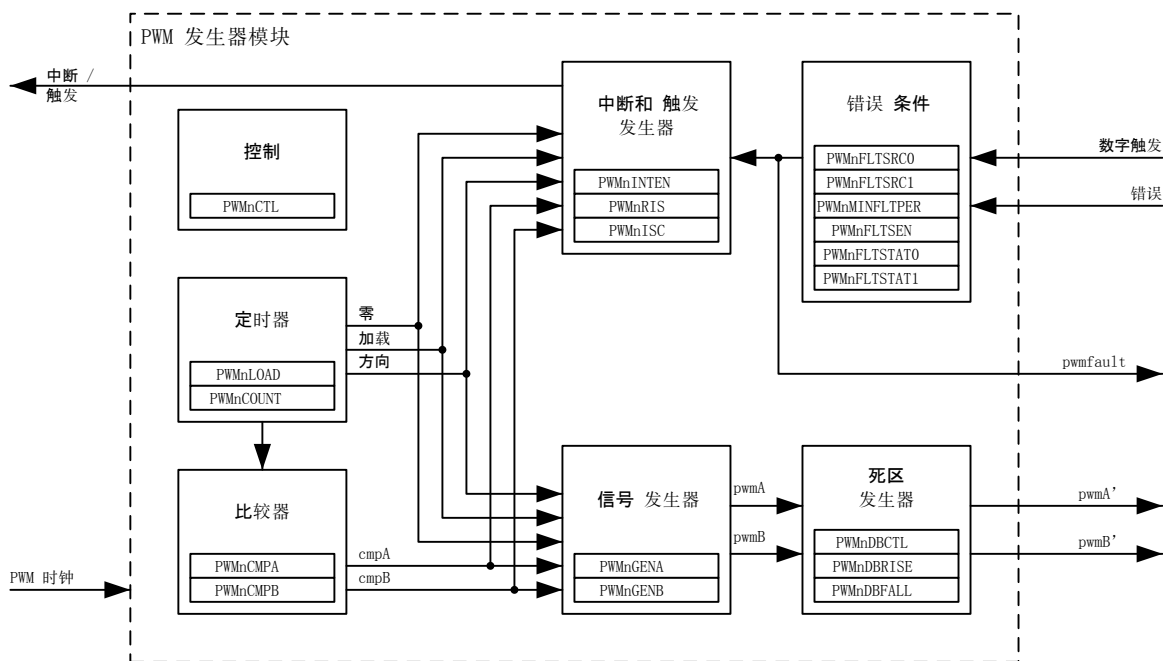




图 20-2. PWM 发生器模块结构图



## 20.2 信号描述

下表列出了 PWM 模块的外部信号，并描述了每个信号的功能。PWM 控制器的信号是某些 GPIO 信号复用功能，在复位后默认为 GPIO 信号。表格中“复用管脚/赋值”一栏下列出了这些 PWM 信号的 GPIO 管脚复用。将 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618页) 中的 AFSEL 位置位，以便选择 PWM 功能。必须将括号中的数字写入 GPIO 端口控制 (GPIOCTL) 寄存器 (636页) 中的 PMCn 域，以便将 PWM 信号分配给指定的 GPIO 端口管脚。有关如何配置 GPIO，请参阅“通用输入/输出端口 (GPIO)”在 597页。

表 20-1. PWM 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
MOFAULT0	3	PD2 (4)	I	TTL	动作控制模块 0 PWM 故障 0。
	32	PH0 (6)			
	64	PF2 (4)			
	112	PK4 (6)			
	143	PD6 (4)			
MOFAULT1	31	PH1 (6)	I	TTL	动作控制模块 0 PWM 故障 1。
	53	PG2 (4)			
	65	PF3 (4)			
	111	PK5 (6)			
	144	PD7 (4)			
MOFAULT2	28	PH2 (6)	I	TTL	动作控制模块 0 PWM 故障 2。
	52	PG3 (4)			
	61	PF4 (4)			
	110	PK6 (6)			
MOFAULT3	27	PH3 (6)	I	TTL	动作控制模块 0 PWM 故障 3。
	60	PF5 (4)			
	109	PK7 (6)			

表 20-1. PWM 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
M0PWM0	32 131	PH0 (4) PP0 (1)	O	TTL	动作控制模块 0 PWM 0。此信号由模块 0 PWM 发生器 0 控制。
M0PWM1	31 132	PH1 (4) PP1 (1)	O	TTL	动作控制模块 0 PWM 1。此信号由模块 0 PWM 发生器 0 控制。
M0PWM2	11 28 136	PP2 (1) PH2 (4) PB4 (4)	O	TTL	动作控制模块 0 PWM 2。此信号由模块 0 PWM 发生器 1 控制。
M0PWM3	27 135	PH3 (4) PB5 (4)	O	TTL	动作控制模块 0 PWM 3。此信号由模块 0 PWM 发生器 1 控制。
M0PWM4	26 51 83 139	PH4 (4) PG4 (4) PM6 (2) PE4 (4)	O	TTL	动作控制模块 0 PWM 4。此信号由模块 0 PWM 发生器 2 控制。
M0PWM5	23 50 82 140	PH5 (4) PG5 (4) PM7 (2) PE5 (4)	O	TTL	动作控制模块 0 PWM 5。此信号由模块 0 PWM 发生器 2 控制。
M0PWM6	1 20 22 36 48	PD0 (4) PN2 (2) PH6 (4) PC4 (4) PG6 (4)	O	TTL	动作控制模块 0 PWM 6。此信号由模块 0 PWM 发生器 3 控制。
M0PWM7	2 21 35 47 119	PD1 (4) PH7 (4) PC5 (4) PG7 (4) PN3 (2)	O	TTL	动作控制模块 0 PWM 7。此信号由模块 0 PWM 发生器 3 控制。
M1FAULT0	16 58 61	PK0 (6) PF7 (5) PF4 (5)	I	TTL	动作控制模块 1 PWM 故障 0。
M1FAULT1	17 55	PK1 (6) PG0 (5)	I	TTL	动作控制模块 1 PWM 故障 1。
M1FAULT2	18 54	PK2 (6) PG1 (5)	I	TTL	动作控制模块 1 PWM 故障 2。
M1FAULT3	19	PK3 (6)	I	TTL	动作控制模块 1 PWM 故障 3。
M1PWM0	1 53	PD0 (5) PG2 (5)	O	TTL	动作控制模块 1 PWM 0。此信号由模块 1 PWM 发生器 0 控制。
M1PWM1	2 52	PD1 (5) PG3 (5)	O	TTL	动作控制模块 1 PWM 1。此信号由模块 1 PWM 发生器 0 控制。
M1PWM2	45 51 139	PA6 (5) PG4 (5) PE4 (5)	O	TTL	动作控制模块 1 PWM 2。此信号由模块 1 PWM 发生器 1 控制。
M1PWM3	46 50 140	PA7 (5) PG5 (5) PE5 (5)	O	TTL	动作控制模块 1 PWM 3。此信号由模块 1 PWM 发生器 1 控制。
M1PWM4	62 71	PF0 (5) PN4 (2)	O	TTL	动作控制模块 1 PWM 4。此信号由模块 1 PWM 发生器 2 控制。
M1PWM5	63 70	PF1 (5) PN5 (2)	O	TTL	动作控制模块 1 PWM 5。此信号由模块 1 PWM 发生器 2 控制。
M1PWM6	64 69	PF2 (5) PN6 (2)	O	TTL	动作控制模块 1 PWM 6。此信号由模块 1 PWM 发生器 3 控制。

表 20-1. PWM 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
M1PWM7	65 68	PF3 (5) PN7 (2)	O	TTL	动作控制模块 1 PWM 7。此信号由模块 1 PWM 发生器 3 控制。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 20.3 功能描述

### 20.3.1 PWM 定时器

每个 PWM 发送器中的定时器都有两种运行模式：递减模式和递增/递减模式。在递减模式中，定时器从装载值计数到零，再返回到装载值，继续递减。在递增/递减模式中，定时器从零开始计数到装载值，然后递减到零，再递增装载值，然后如此反复。一般的，递减模式用来产生左对齐或右对齐的 PWM 信号，递增/递减模式用来产生中间对齐的 PWM 信号。

定时器产生三个信号并被用在 PWM 产生过程中：方向信号（在递减模式中一直是低电平，在递增/递减模式中，则是在高低电平之间切换）；当计时值为零时的一个宽度等于时钟周期的高电平脉冲；当计时值等于装载值时，一个宽度等于时钟周期的高电平脉冲。注：在递减模式中，零脉冲之后紧接着是一个装载脉冲。在本章的图片中，这些信号标记为“dir”、“zero”和“load”。

### 20.3.2 PWM 比较器

每个 PWM 发生器都有两个比较器，用来监控计数器的值；当比较器与计数器的值相等时，比较器输出一个宽度等于时钟周期的高电平脉冲，在本章图片中标记为“cmpA”和“cmpB”。在递增/递减计数模式中，比较器在递增和递减计数时都要进行比较，因此必须通过计数器的方向信号来限定。这些限定脉冲在 PWM 信号的生成过程中使用。如果任一比较器的匹配值大于计时器装载值，则该比较器永远不会输出高脉冲。

图 20-3 在 1156 页显示了计数器处于递减计数模式时的行为以及这些脉冲之间的关系。图 20-4 在 1156 页显示了计数器处于递增/递减计数模式时的行为以及这些脉冲之间的关系。在这些图中使用了如下的定义：

- LOAD 是 PWMnLOAD 里的寄存器的值
- COMPA 是 PWMnCMPA 里的寄存器的值
- COMPB 是 PWMnCMPB 里的寄存器里的值
- 0 就是数值 0
- load 表示当计数器的值等于装载值时的宽度为单时钟周期的内部信号
- zero 表示当计数器的值是零时的宽度为单时钟周期的内部信号
- cmpA 表示宽度为单时钟周期的内部信号，此时计数器的值等于 COMPA
- cmpB 表示宽度为单时钟周期的内部信号，此时计数器的值等于 COMPB
- dir 是表示计数方向的内部信号

图 20-3. PWM 递减计数模式

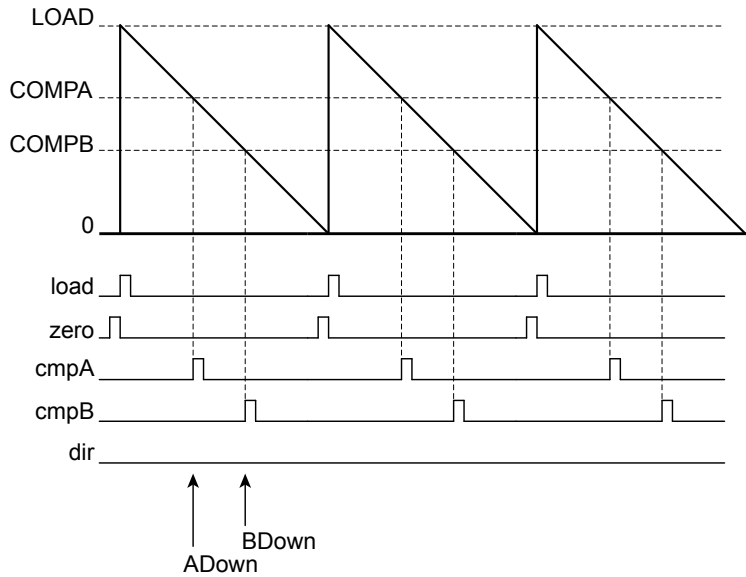
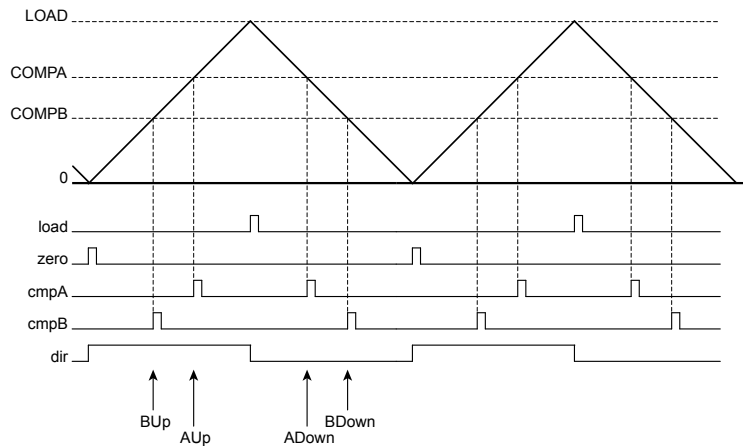


图 20-4. PWM 递增/递减计数模式

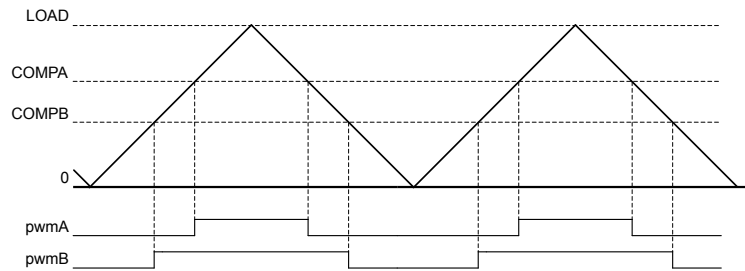


### 20.3.3 PWM 信号发生器

每个 PWM 发生器捕获 load、zero、cmpA 和 cmpB 脉冲（由方向信号来限定），并产生两个内部 PWM 信号：pwmA 和 pwmB。在递减计数模式中，能够影响这些信号的事件有四个：零、装载、匹配 A 递减以及匹配 B 递减。在递增/递减模式中，能够影响这些信号的事件有六个：零、装值、匹配 A 递增、匹配 B 递增、匹配 A 递减以及匹配 B 递减。当匹配 A 或匹配 B 事件与零或者装载事件同时发生时，它们可以被忽略。如果匹配 A 和匹配 B 时间重合，则第一个信号 pwmA 只根据匹配 A 事件生成，第二个信号 pwmB 只根据匹配 B 事件生成。

各个事件在 PWM 输出信号的影响上都是可编程的：可以保留（忽略该事件），可以翻转，可以驱动为低电平、或驱动为高电平。这些动作可用来产生一对不同位置 and 不同占空比的 PWM 信号，这对信号可以重叠或不重叠。图 20-5 在 1157 页显示了递增/递减计数模式产生的一对中心对齐、含不同占空比的重叠 PWM 信号。该图显示了传递到死区发生器之前的 pwmA 和 pwmB 信号。

图 20-5. 递增/递减计数模式产生 PWM 信号的例子



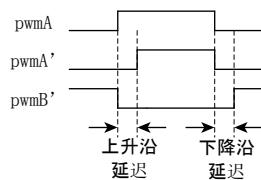
在该实例中，第一个发生器设置为出现匹配 A 递增事件时驱动为高电平，出现匹配 A 递减事件时驱动为低电平，并忽略其它四个事件。第二个发生器设置为出现匹配 B 递增事件时驱动为高电平，出现匹配 B 递减事件时驱动为低电平，并忽略其它四个事件。改变比较器 A 的值可改变 pwmA 信号的占空比，改变比较器 B 的值可改变 pwmB 信号的占空比。

### 20.3.4 死区发生器

每个 PWM 发生器产生的 pwmA 和 pwmB 信号会传递给死区发生器。如果死区发生器禁用，则 PWM 信号只简单地通过该模块得到 pwmA' 和 pwmB'，而不会发生改变。如果死区发生器使能，则丢弃 pwmB 信号，并在 pwmA 信号的基础上产生两个 PWM 信号。第一个 PWM 输出信号 pwmA' 是上升沿延迟了的 pwmA 信号，延迟时间可编程。第二个 PWM 输出信号，pwmB' 是在 pwmA 信号的反相信号，并在 pwmA 信号的下降沿和 pwmB' 信号的上升沿之间增加了可编程的延迟时间。

最终信号是一对高电平有效的信号，并且其中一个信号总是为高电平，但在跳变处的那段可编程延迟时间内都为低电平。这样两个信号便可用来驱动半-H 桥，由于他们带有死区延迟，可以避免过冲电流破坏电力电子器件。图 20-6 在 1157 页显示了死区发生器对 pwmA 信号的影响，以及将被发送到输出控制模块的结果信号 pwmA' 和 pwmB'。

图 20-6. PWM 死区发生器



### 20.3.5 中断/ADC-触发选择器

每个 PWM 发生器还捕获四个（或六个）计数器事件，并使用它们来产生中断或者 ADC 触发事件。用户可以选择这些事件中的任一个或一组作为中断源；只要其中一个所选事件发生就会产生中断。此外，也可以选择相同的事件、不同事件、同组事件、不同组事件作为 ADC 的触发源；只要其中一个所选事件发生就会产生 ADC 脉冲。事件的选择允许在 pwmA 或者 pwmB 信号的特殊位置时发生中断或者 ADC 触发。注：中断和 ADC 触发基于原始事件，不考虑死区发生器在 PWM 信号边沿上产生的延迟。

### 20.3.6 同步方法

每个 PWM 模块提供四个 PWM 发生器，每个发生器提供两个 PWM 输出，应用范围广泛。一般来说，PWM 有两种操作类别：

- 异步. PWM 发生器和它的两个输出信号独自使用，与其他 PWM 发生器独立。

- 同步. PWM 发生器和它的两个输出信号通过共用的统一的时间基准与其它 PWM 发生器保持联系。如果多个 PWM 发生器配置使用相同的计时器装载值，同步可以保证它们有相同的计数值（在同步之前必须配置 PWM 发生器）。通过这个功能，可产生两个或者更多 MnPWMn 信号。而且因为计数器总是具有相同的值，那些信号的边沿之间将存在某种已知关系。该模块中的其他状态提供了一种维持共用时间基准和相互同步的机制。

通过向 PWM 时间基准同步 (PWMSYNC) 寄存器中写入数值，将与发生器相关的 SYNCn 位置位，PWM 发生器中的计数器会被复位为零。可以在一次访问中将必要的 SYNCn 位置位，这样可以同步多个 PWM 发生器。比如，将 PWMSYNC 寄存器中的 SYNC0 和 SYNC1 位置位就可以同时把 PWM 发生器 0 和 PWM 发生器 1 中的计时器复位。

另外，同步也可发生在多个 PWM 发生器之间，通过以下三种方式中的一种：

- 立即. 写入值立即生效，硬件立即动作。
- 本地同步. 写入的值不会影响到逻辑，直到计数器在 PWM 周期结束时减为零。这样写入的的作用延期了，避免出现过短或过长的脉冲。
- 全局同步. 写入的值并不影响逻辑，直到有两个连续的事件发生：(1) 在 PWMnCTL 寄存器中启用全局同步的更新模式。(2) 计数器的值在 PWM 周期的结束到达零。在这种情况下，写入的作用会延迟直到 PWM 周期的结束伴随着所有的更新结束才发生作用。这种模式可以同时多个 PWM 发生器中的多项进行更新，而不会在更新过程中出现意外的情况。所有逻辑在新值运行之前都先在原来的值上运行。装载和比较器匹配值的更新方法可以在各个 PWM 发生器模块中单独配置。当那些模块中的定时器同步时，通常可以在 PWM 发生器模块中使用同步更新机制。尽管该机制正常工作时不要求这个。

下面的寄存器根据 PWMnCTL 寄存器中位和域 (LOADUPD; CMPAUPD; CMPBUPD) 的各种状态提供更新模式：

- 发生器寄存器: PWMnLOAD、PWMnCMPA 和 PWMnCMPB

下面的寄存器默认立即更新模式，提供同步更新的可选功能，而不使更新立即生效：

- 模块级寄存器: PWMENABLE (根据 PWMENUPD 寄存器中的 ENUPDn 位的状态)。
- 发生器寄存器: PWMnGENA、PWMnGENB、PWMnDBCTL、PWMnDBRISE 和 PWMnDBFALL (根据 PWMnCTL 寄存器中更新模式位/域 [GENAUPD、GENBUPD、DBCTLUPD、DBRISEUPD 以及 DBFALLUPD] 的各种状态而定)。

所有其它的寄存器为一个应用提供静态的配置或者与维持同步不相关的动态配置，因此不需要同步更新的功能。

### 20.3.7 故障条件

故障条件时必须向控制器发送信号，停止 PWM 功能，并将 MnPWMn 信号返回到安全状态。有两种情况可能导致故障条件：

- 控制器中止并且在运动控制中所必需的计数也不能工作。
- 检测到一个外部错误或事件。

每个 PWM 发生器可以使用下列的输入来产生故障条件：

- MnFAULTn 管脚有效
- 调试器引发的控制器中止

### ■ ADC 数字比较器的触发

故障条件的产生以每一个 PWM 发生器为基础。每个 PWM 发生器都要配置必要的条件，以便显示出故障条件。用该方法可以开发具有相关控制或者独立控制的应用程序。

提供八个故障输入管脚 (MnFAULTn)。这些输入可能和产生高电平有效或低电平有效的电路一起指示一个错误状态。MnFAULTn 管脚可以通过 PWMnFLTSEN 寄存器单独编程，以获得相应的逻辑意义。

PWMnCTL 寄存器用于提供 PWM 发生器的模式控制（包括故障条件处理）。PWMnCTL 寄存器也可以选择故障条件是跟外部条件持续一样长的时间，还是锁存直到软件清除故障条件。最后，这个寄存器使能一个计数器，用来扩展外部事件的故障条件事件，以确保持续时间达到一个最小值。最小故障周期计数在 PWMnMINFLTPER 寄存器中指定。

PWMnFLTSTAT0 和 PWMnFLTSTAT1 寄存器中提供了具体故障的起因状态。

通过配置 PWMINTEN 寄存器可将 PWM 发生器故障条件提升为控制器中断。

## 20.3.8 输出控制块

在 pwmA' 和 pwmB' 信号被作为 MnPWMn 信号发送到管脚之前，输出控制块负责对它们做最后的调整。通过 PWM 输出使能 (PWNENABLE) 寄存器，即可对送到管脚的一组 PWM 信号进行实质性修改。例如，这个功能可以通过对单个寄存器执行写操作来与无刷直流电机通信（不需要修改单个 PWM 发生器，只需要通过反馈控制回路的修改即可）。另外，对 PWNENABLE 寄存器中的位进行更新时，即可以配置成立即更新，也可以通过 PWM 使能更新 (PWMENUPD) 寄存器配置成本地同步或全局同步到下一个更新。

在故障状态下，PWM 输出信号 MnPWMn 通常要求是安全值，否则可能无法安全地控制外部设备。在故障条件下，PWMFAULT 寄存器决定是继续驱动已生成的信号还是继续按照 PWMFAULTVAL 寄存器的内容进行编码。

用 PWM 输出反相 (PWMINVERT) 寄存器可以将任何 MnPWMn 信号最终反相，把他们从默认的高电平有效变成低电平有效。就算是一个值在 PWMFAULT 寄存器中使能并在 PWMFAULTVAL 寄存器中指定，照样可以反相。换句话说，如果 PWMFAULT、PWMFAULTVAL 和 PWMINVERT 寄存器中的某位已置位，MnPWMn 信号的输出将是 0，并不是 PWMFAULTVAL 寄存器所指定的 1。

## 20.4 初始化及配置

以下是对 PWM 发生器 0 进行初始化的示例，要求频率为 25-kHz，MnPWM0 管脚上的占空比为 25%，MnPWM1 管脚上的占空比为 75%。该例假定系统时钟为 20 MHz。

1. 向系统控制模块的 RCGC0 寄存器写入 0x0010.0000，启用 PWM 时钟（请参考 442 页）。
2. 通过系统时钟模块中的 RCGC2 寄存器启用相应 GPIO 模块的时钟（请参考 448 页）。
3. 在 GPIO 模块中，用 GPIOAFSEL 寄存器来使能对应管脚的复用功能。欲了解需要配置哪些 GPIO，请参阅表 23-4 在 1271 页。
4. 配置 GPIOPCTL 寄存器中的 PMCn 域，以将 PWM 信号分配到适当的管脚（请参考 636 页和表 23-5 在 1280 页）。
5. 配置系统控制模块中的运行模式时钟配置 (RCC) 寄存器，使用 PWM 分频 (USEPWMDIV)，并将分频器 (PWMDIV) 设置为 2 分频 (000)。
6. 将 PWM 发生器配置为递减计数模式，并立即更新参数。
  - 向 PWM0CTL 寄存器写入 0x0000.0000。

- 向 PWM0GENA 寄存器写入 0x0000.008C。
  - 向 PWM0GENB 寄存器写入 0x0000.080C。
7. 设置周期。若要得到 25 KHz 的频率，周期等于 1/25000 ( 40 $\mu$ s )。PWM 时钟源为 10 MHz；由系统时钟 2 分频得到这意味着一个定时器周期要等于 400 个 PWM 时钟周期然后用这个值来设置 PWM0LOAD 寄存器。在递减模式中，将 PWM0LOAD 寄存器的 LOAD 域设置为所需要的周期减 1。
- 向 PWM0LOAD 寄存器写入 0x0000.018F。
8. 将 MnPWM0 管脚的脉冲宽度设置为 25% 占空比。
- 向 PWM0CMPA 寄存器写入 0x0000.012B。
9. 将 MnPWM1 管脚的脉冲宽度设置为 75% 占空比。
- 向 PWM0CMPB 寄存器写入 0x0000.0063；
10. 启动 PWM 发生器 0 中的定时器。
- 向 PWM0CTL 寄存器写入 0x0000.0001；
11. 启用 PWM 输出。
- 向 PWMENABLE 寄存器写入 0x0000.0003；

## 20.5 寄存器映射

表 20-2 在 1160 页列出了 PWM 寄存器。下表中列出的“偏移量”代表该寄存器相对于 PWM 模块基础地址的十六进制地址增量。

- PWM0x4002.8000
- PWM10x4002.9000

请注意，配置这些寄存器之前，必须启用 PWM 模块时钟（请参考 442 页）。PWM 模块时钟启用之后，必须等待三个系统时钟的延迟，然后才能访问 PWM 模块寄存器。

表 20-2. PWM 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	PWMCTL	R/W	0x0000.0000	PWM 主机控制寄存器	1164
0x004	PWMSYNC	R/W	0x0000.0000	PWM 时间基准同步寄存器	1166
0x008	PWMENABLE	R/W	0x0000.0000	PWM 输出使能寄存器	1167
0x00C	PWMINVERT	R/W	0x0000.0000	PWM 输出反相寄存器	1169
0x010	PWMFAULT	R/W	0x0000.0000	PWM 输出故障寄存器	1171
0x014	PWMINTEN	R/W	0x0000.0000	PWM 中断使能寄存器	1173
0x018	PWMRIS	RO	0x0000.0000	PWM 原始中断状态寄存器	1175
0x01C	PWMISC	R/W1C	0x0000.0000	PWM 中断状态和清除寄存器	1177



表 20-2. PWM 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x020	PWMSTATUS	RO	0x0000.0000	PWM 状态寄存器	1179
0x024	PWMFAULTVAL	R/W	0x0000.0000	PWM 故障条件值寄存器	1181
0x028	PWMENUPD	R/W	0x0000.0000	PWM 使能更新寄存器	1183
0x040	PWM0CTL	R/W	0x0000.0000	PWM0 控制寄存器	1186
0x044	PWM0INTEN	R/W	0x0000.0000	PWM0 中断和触发使能寄存器	1190
0x048	PWM0RIS	RO	0x0000.0000	PWM0 原始中断状态寄存器	1193
0x04C	PWM0ISC	R/W1C	0x0000.0000	PWM0 中断状态和清除寄存器	1195
0x050	PWM0LOAD	R/W	0x0000.0000	PWM0 装载寄存器	1197
0x054	PWM0COUNT	RO	0x0000.0000	PWM0 计数寄存器	1198
0x058	PWM0CMPA	R/W	0x0000.0000	PWM0 比较寄存器 A	1199
0x05C	PWM0CMPB	R/W	0x0000.0000	PWM0 比较寄存器 B	1200
0x060	PWM0GENA	R/W	0x0000.0000	PWM0 发生器 A 控制寄存器	1201
0x064	PWM0GENB	R/W	0x0000.0000	PWM0 发生器 B 控制寄存器	1204
0x068	PWM0DBCTL	R/W	0x0000.0000	PWM0 死区控制寄存器	1206
0x06C	PWM0DBRISE	R/W	0x0000.0000	PWM0 死区上升沿延迟寄存器	1207
0x070	PWM0DBFALL	R/W	0x0000.0000	PWM0 死区下降沿延迟寄存器	1208
0x074	PWM0FLTSRC0	R/W	0x0000.0000	PWM0 故障源寄存器 0	1209
0x07C	PWM0MINFLTPER	R/W	0x0000.0000	PWM0 最小故障时间寄存器	1214
0x080	PWM1CTL	R/W	0x0000.0000	PWM1 控制寄存器	1186
0x084	PWM1INTEN	R/W	0x0000.0000	PWM1 中断和触发使能寄存器	1190
0x088	PWM1RIS	RO	0x0000.0000	PWM1 原始中断状态寄存器	1193
0x08C	PWM1ISC	R/W1C	0x0000.0000	PWM1 中断状态和清除寄存器	1195
0x090	PWM1LOAD	R/W	0x0000.0000	PWM1 装载寄存器	1197
0x094	PWM1COUNT	RO	0x0000.0000	PWM1 计数寄存器	1198
0x098	PWM1CMPA	R/W	0x0000.0000	PWM1 比较寄存器 A	1199
0x09C	PWM1CMPB	R/W	0x0000.0000	PWM1 比较寄存器 B	1200
0x0A0	PWM1GENA	R/W	0x0000.0000	PWM1 发生器 A 控制寄存器	1201
0x0A4	PWM1GENB	R/W	0x0000.0000	PWM1 发生器 B 控制寄存器	1204
0x0A8	PWM1DBCTL	R/W	0x0000.0000	PWM1 死区控制寄存器	1206
0x0AC	PWM1DBRISE	R/W	0x0000.0000	PWM1 死区上升沿延迟寄存器	1207
0x0B0	PWM1DBFALL	R/W	0x0000.0000	PWM1 死区下降沿延迟寄存器	1208
0x0B4	PWM1FLTSRC0	R/W	0x0000.0000	PWM1 故障源寄存器 0	1209
0x0B8	PWM1FLTSRC1	R/W	0x0000.0000	PWM1 故障源寄存器 1	1211

表 20-2. PWM 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x0BC	PWM1MINFLTPER	R/W	0x0000.0000	PWM1 最小故障时间寄存器	1214
0x0C0	PWM2CTL	R/W	0x0000.0000	PWM2 控制寄存器	1186
0x0C4	PWM2INTEN	R/W	0x0000.0000	PWM2 中断和触发使能寄存器	1190
0x0C8	PWM2RIS	RO	0x0000.0000	PWM2 原始中断状态寄存器	1193
0x0CC	PWM2ISC	R/W1C	0x0000.0000	PWM2 中断状态和清除寄存器	1195
0x0D0	PWM2LOAD	R/W	0x0000.0000	PWM2 装载寄存器	1197
0x0D4	PWM2COUNT	RO	0x0000.0000	PWM2 计数寄存器	1198
0x0D8	PWM2CMPA	R/W	0x0000.0000	PWM2 比较寄存器 A	1199
0x0DC	PWM2CMPB	R/W	0x0000.0000	PWM2 比较寄存器 B	1200
0x0E0	PWM2GENA	R/W	0x0000.0000	PWM2 发生器 A 控制寄存器	1201
0x0E4	PWM2GENB	R/W	0x0000.0000	PWM2 发生器 B 控制寄存器	1204
0x0E8	PWM2DBCTL	R/W	0x0000.0000	PWM2 死区控制寄存器	1206
0x0EC	PWM2DBRISE	R/W	0x0000.0000	PWM2 死区上升沿延迟寄存器	1207
0x0F0	PWM2DBFALL	R/W	0x0000.0000	PWM2 死区下降沿延迟寄存器	1208
0x0F4	PWM2FLTSRC0	R/W	0x0000.0000	PWM2 故障源寄存器 0	1209
0x0F8	PWM2FLTSRC1	R/W	0x0000.0000	PWM2 故障源寄存器 1	1211
0x0FC	PWM2MINFLTPER	R/W	0x0000.0000	PWM2 最小故障时间寄存器	1214
0x100	PWM3CTL	R/W	0x0000.0000	PWM3 控制寄存器	1186
0x104	PWM3INTEN	R/W	0x0000.0000	PWM3 中断和触发使能寄存器	1190
0x108	PWM3RIS	RO	0x0000.0000	PWM3 原始中断状态寄存器	1193
0x10C	PWM3ISC	R/W1C	0x0000.0000	PWM3 中断状态和清除寄存器	1195
0x110	PWM3LOAD	R/W	0x0000.0000	PWM3 装载寄存器	1197
0x114	PWM3COUNT	RO	0x0000.0000	PWM3 计数寄存器	1198
0x118	PWM3CMPA	R/W	0x0000.0000	PWM3 比较寄存器 A	1199
0x11C	PWM3CMPB	R/W	0x0000.0000	PWM3 比较寄存器 B	1200
0x120	PWM3GENA	R/W	0x0000.0000	PWM3 发生器 A 控制寄存器	1201
0x124	PWM3GENB	R/W	0x0000.0000	PWM3 发生器 B 控制寄存器	1204
0x128	PWM3DBCTL	R/W	0x0000.0000	PWM3 死区控制寄存器	1206
0x12C	PWM3DBRISE	R/W	0x0000.0000	PWM3 死区上升沿延迟寄存器	1207
0x130	PWM3DBFALL	R/W	0x0000.0000	PWM3 死区下降沿延迟寄存器	1208
0x134	PWM3FLTSRC0	R/W	0x0000.0000	PWM3 故障源寄存器 0	1209
0x138	PWM3FLTSRC1	R/W	0x0000.0000	PWM3 故障源寄存器 1	1211
0x13C	PWM3MINFLTPER	R/W	0x0000.0000	PWM3 最小故障时间寄存器	1214

表 20-2. PWM 寄存器映射 (续)

偏移量	名称	类型	复位	描述	见页面
0x800	PWM0FLTSEN	R/W	0x0000.0000	PWM0 故障管脚逻辑感知寄存器	1215
0x804	PWM0FLTSTAT0	-	0x0000.0000	PWM0 故障状态寄存器 0	1216
0x880	PWM1FLTSEN	R/W	0x0000.0000	PWM1 故障管脚逻辑感知寄存器	1215
0x884	PWM1FLTSTAT0	-	0x0000.0000	PWM1 故障状态寄存器 0	1216
0x900	PWM2FLTSEN	R/W	0x0000.0000	PWM2 故障管脚逻辑感知寄存器	1215
0x904	PWM2FLTSTAT0	-	0x0000.0000	PWM2 故障状态寄存器 0	1216
0x908	PWM2FLTSTAT1	-	0x0000.0000	PWM2 故障状态寄存器 1	1218
0x980	PWM3FLTSEN	R/W	0x0000.0000	PWM3 故障管脚逻辑感知寄存器	1215
0x984	PWM3FLTSTAT0	-	0x0000.0000	PWM3 故障状态寄存器 0	1216
0x988	PWM3FLTSTAT1	-	0x0000.0000	PWM3 故障状态寄存器 1	1218
0xFC0	PWMPP	RO	0x0000.0344	PWM 外设属性寄存器	1221
0xFC4	PWMPC	R/W	0x0000.0000	PWM 外设属性寄存器	1223
1x078	PWM0FLTSRC1	R/W	0x0000.0000	PWM0 故障源寄存器 1	1211
1x808	PWM0FLTSTAT1	-	0x0000.0000	PWM0 故障状态寄存器 1	1218
1x888	PWM1FLTSTAT1	-	0x0000.0000	PWM1 故障状态寄存器 1	1218

## 20.6 寄存器描述

下文将按地址偏移量的数字顺序列出 PWM 寄存器,并对它们进行描述。

## 寄存器 1: PWM 主机控制寄存器 ( PWMCTL ) , 偏移量 0x000

该寄存器提供了对 PWM 发生模块的主机控制。

## PWM 主机控制寄存器 (PWMCTL)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x000

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												GLOBALSYNC3	GLOBALSYNC2	GLOBALSYNC1	GLOBALSYNC0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	GLOBALSYNC3	R/W	0	更新 PWM 发生器 3  值 描述 1 在下一对应应的计数器变为 0 时，PWM 发生器 3 中已排队的装载寄存器或比较寄存器的更新将应用。 0 无影响。  更新结束时该位自动清零，不能由软件来清零。
2	GLOBALSYNC2	R/W	0	更新 PWM 发生器 2  值 描述 1 在下一对应应的计数器变为 0 时，PWM 发生器 2 中已排队的装载寄存器或比较寄存器的更新将应用。 0 无影响。  更新结束时该位自动清零，不能由软件来清零。
1	GLOBALSYNC1	R/W	0	更新 PWM 发生器 1  值 描述 1 在下一对应应的计数器变为 0 时，PWM 发生器 1 中已排队的装载寄存器或比较寄存器的更新将应用。 0 无影响。  更新结束时该位自动清零，不能由软件来清零。

位/域	名称	类型	复位	描述
0	GLOBALSYNC0	R/W	0	更新 PWM 发生器 0  值 描述 1 在下次对应的计数器变为 0 时，PWM 发生器 0 中已排队的装载寄存器或比较寄存器的更新将应用。 0 无影响。  更新结束时该位自动清零，不能由软件来清零。

**寄存器 2: PWM 时间基准同步寄存器 ( PWMSYNC ) , 偏移量 0x004**

该寄存器提供一种对 PWM 发生模块中的计数器进行同步的方法。将该寄存器的某个位置 1 会将相应的计数器复位为零，同时将多位置 1 会让多个计数器同时复位。复位之后，寄存器中的位自动清零，读这些位时如果返回零，则表示同步已完成。

**PWM 时间基准同步寄存器 (PWMSYNC)**

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x004

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												SYNC3	Sync2	Sync1	SYNC0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	SYNC3	R/W	0	复位发生器 3 计数器  值 描述 1 对 PWM 发生器 3 的计数器进行复位。 0 无影响。
2	Sync2	R/W	0	复位发生器 2 计数器  值 描述 1 对 PWM 发生器 2 的计数器进行复位。 0 无影响。
1	Sync1	R/W	0	复位发生器 1 计数器  值 描述 1 对 PWM 发生器 1 的计数器进行复位。 0 无影响。
0	SYNC0	R/W	0	复位发生器 0 计数器  值 描述 1 对 PWM 发生器 0 的计数器进行复位。 0 无影响。

### 寄存器 3: PWM 输出使能寄存器 ( PWMENABLE ) , 偏移量 0x008

该寄存器主要控制将哪一个已产生的 PWM 信号 ( pwmA' 和 pwmB' ) 输出到管脚 MnPWMn。如果禁止 PWM 输出, PWM 信号就不会传递给管脚, 但可以继续生成 PWM 信号 ( 例如在时间基准同步时 )。当该寄存器中的某些位置 1 时, 相应的 pwmA' 信号和 pwmB' 信号会被传递到输出级。当该寄存器中的某些位被清零时, 相应的 pwmA' 信号和 pwmB' 信号会被清零, 并传递到输出级。

PWMINVERT 寄存器控制着输出级, 所以如果将该寄存器中相应的位置位, 则 MnPWMn 管脚输出的值将与此寄存器中配置的值反相。该寄存器中的位更新可以设置为立即更新, 也可以用 PWMENUPD 寄存器中的 ENUPDn 域设置成本地或全局同步到下一次同步更新。

#### PWM 输出使能寄存器 (PWMENABLE)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x008

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PWM7EN	PWM6EN	PWM5EN	PWM4En	PWM3En	PWM2En	PWM1En	PWM0EN
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	PWM7EN	R/W	0	MnPWM7 输出启用  值 描述 1 产生的 pwm3B' 信号被传递到 MnPWM7 管脚。 0 MnPWM7 信号值是零。
6	PWM6EN	R/W	0	MnPWM6 输出启用  值 描述 1 产生的 pwm3A' 信号被传递到 MnPWM6 管脚。 0 MnPWM6 信号值是零。
5	PWM5EN	R/W	0	MnPWM5 输出启用  值 描述 1 产生的 pwm2B' 信号被传递到 MnPWM5 管脚。 0 MnPWM5 信号值是零。
4	PWM4En	R/W	0	MnPWM4 输出启用  值 描述 1 产生的 pwm2A' 信号被传递到 MnPWM4 管脚。 0 MnPWM4 信号值是零。

位/域	名称	类型	复位	描述
3	PWM3En	R/W	0	MnPWM3 输出启用  值 描述 1 产生的 pwm1B' 信号被传递到 MnPWM3 管脚。 0 MnPWM3 信号值是零。
2	PWM2En	R/W	0	MnPWM2 输出启用  值 描述 1 产生的 pwm1A' 信号被传递到 MnPWM2 管脚。 0 MnPWM2 信号值是零。
1	PWM1En	R/W	0	MnPWM1 输出启用  值 描述 1 产生的 pwm0B' 信号被传递到 MnPWM1 管脚。 0 MnPWM1 信号值是零。
0	PWM0EN	R/W	0	MnPWM0 输出启用  值 描述 1 产生的 pwm0A' 信号被传递到 MnPWM0 管脚。 0 MnPWM0 信号值是零。



## 寄存器 4: PWM 输出反相寄存器 ( PWMINVERT ) , 偏移量 0x00C

该寄存器用于对器件管脚上的 MnPWMn 信号极性进行主机控制。PWM 发生器生成的 pwmA' 和 pwmB' 信号为高电平有效，通过本寄存器可以将其变为低电平有效。被禁用的 PWM 通道也通过输出反相器被传递（假如这样配置的话），处于不活动状态的信号可以是高电平。另外，在故障状态下，如果 PWMFAULT 寄存器在 MnPWMn 信号处指定了值，则当此寄存器的相应位被置位时，该值也将反相。

### PWM 输出反相寄存器 (PWMINVERT)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x00C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PWM7INV	PWM6INV	PWM5INV	PWM4INV	PWM3INV	PWM2INV	PWM1INV	PWM0INV
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
7	PWM7INV	R/W	0	反相 MnPWM7 信号  值 描述 1 MnPWM7 信号反相。 0 MnPWM7 信号未反相。
6	PWM6INV	R/W	0	反相 MnPWM6 信号  值 描述 1 MnPWM6 信号反相。 0 MnPWM6 信号未反相。
5	PWM5INV	R/W	0	反相 MnPWM5 信号  值 描述 1 MnPWM5 信号反相。 0 MnPWM5 信号未反相。
4	PWM4INV	R/W	0	反相 MnPWM4 信号  值 描述 1 MnPWM4 信号反相。 0 MnPWM4 信号未反相。

---

位/域	名称	类型	复位	描述
3	PWM3INV	R/W	0	反相 MnPWM3 信号  值 描述 1 MnPWM3 信号反相。 0 MnPWM3 信号未反相。
2	PWM2INV	R/W	0	反相 MnPWM2 信号  值 描述 1 MnPWM2 信号反相。 0 MnPWM2 信号未反相。
1	PWM1INV	R/W	0	反相 MnPWM1 信号  值 描述 1 MnPWM1 信号反相。 0 MnPWM1 信号未反相。
0	PWM0INV	R/W	0	反相 MnPWM0 信号  值 描述 1 MnPWM0 信号反相。 0 MnPWM0 信号未反相。

## 寄存器 5: PWM 输出故障寄存器 ( PWMFAULT ) , 偏移量 0x010

该寄存器用来控制出现故障情况时的 MnPWMn 输出行为。故障输入 ( MnFAULTn 管脚以及数字比较器输出 ) 和调试事件均可看作故障条件。在故障条件下, 每个 pwmA' 和 pwmB' 信号可以按原样直接通过, 也可以通过 PWMFAULTVAL 寄存器中相应的位指定数值。对于配置为直通的输出来说, 处理相应 PWM 发生器的调试事件会决定是否继续产生 pwmA' 或者 pwmB' 信号。

若在输出反相器之前对故障条件进行控制, 如果通道被配置成反相, 那么在故障条件下驱动为指定值的 PWM 信号将被反相 ( 即, 该管脚在存在故障情况时驱动为指定值 )。

### PWM 输出故障寄存器 (PWMFAULT)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x010

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								FAULT7	FAULT6	FAULT5	FAULT4	FAULT3	FAULT2	FAULT1	FAULT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	FAULT7	R/W	0	MnPWM7 故障  值 描述 1 MnPWM7 输出信号被驱动为 PWMFAULTVAL 寄存器中 PWM7 位所指定的值。 0 产生的 pwm3B' 信号被传递到 MnPWM7 管脚。
6	FAULT6	R/W	0	MnPWM6 故障  值 描述 1 MnPWM6 输出信号被驱动为 PWMFAULTVAL 寄存器中 PWM6 位所指定的值。 0 产生的 pwm3A' 信号被传递到 MnPWM6 管脚。
5	FAULT5	R/W	0	MnPWM5 故障  值 描述 1 MnPWM5 输出信号被驱动为 PWMFAULTVAL 寄存器中 PWM5 位所指定的值。 0 产生的 pwm2B' 信号被传递到 MnPWM5 管脚。

位/域	名称	类型	复位	描述
4	FAULT4	R/W	0	MnPWM4 故障  值 描述 1 MnPWM4 输出信号被驱动为 PWMFAULTVAL 寄存器中 PWM4 位所指定的值。 0 产生的 pwm2A' 信号被传递到 MnPWM4 管脚。
3	FAULT3	R/W	0	MnPWM3 故障  值 描述 1 MnPWM3 输出信号被驱动为 PWMFAULTVAL 寄存器中 PWM3 位所指定的值。 0 产生的 pwm1B' 信号被传递到 MnPWM3 管脚。
2	FAULT2	R/W	0	MnPWM2 故障  值 描述 1 MnPWM2 输出信号被驱动为 PWMFAULTVAL 寄存器中 PWM2 位所指定的值。 0 产生的 pwm1A' 信号被传递到 MnPWM2 管脚。
1	FAULT1	R/W	0	MnPWM1 故障  值 描述 1 MnPWM1 输出信号被驱动为 PWMFAULTVAL 寄存器中 PWM1 位所指定的值。 0 产生的 pwm0B' 信号被传递到 MnPWM1 管脚。
0	FAULT0	R/W	0	MnPWM0 故障  值 描述 1 MnPWM0 输出信号被驱动为 PWMFAULTVAL 寄存器中 PWM0 位所指定的值。 0 产生的 pwm0A' 信号被传递到 MnPWM0 管脚。

## 寄存器 6: PWM 中断使能寄存器 ( PWMINTEN ) , 偏移量 0x014

该寄存器控制 PWM 模块的全局中断产生功能。能够引起中断的事件包括故障输入和来自 PWM 发生器的各个中断。

注意: 该寄存器的 INTFAULTn 和 INTPWMn 位中的 n 与 PWM 发生器相对应, 而不与 FAULTn 信号相对应。

### PWM 中断使能寄存器 (PWMINTEN)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x014

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留												INTFAULT3	INTFAULT2	INTFAULT1	INTFAULT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												INTPWM3	INTPWM2	INTPWM1	INTPWM0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:20	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
19	INTFAULT3	R/W	0	中断故障 3  值 描述 1 当 PWM 发生器 3 发生故障时, 中断将被发送到中断控制器。 0 PWM 发生器 3 的故障条件被抑制, 不会发送到中断控制器。
18	INTFAULT2	R/W	0	中断故障 2  值 描述 1 当 PWM 发生器 2 发生故障时, 中断将被发送到中断控制器。 0 PWM 发生器 2 的故障条件被抑制, 不会发送到中断控制器。
17	INTFAULT1	R/W	0	中断故障 1  值 描述 1 当 PWM 发生器 1 发生故障时, 中断将被发送到中断控制器。 0 PWM 发生器 1 的故障条件被抑制, 不会发送到中断控制器。
16	INTFAULT0	R/W	0	中断故障 0  值 描述 1 当 PWM 发生器 0 发生故障时, 中断将被发送到中断控制器。 0 PWM 发生器 0 的故障条件被抑制, 不会发送到中断控制器。

位/域	名称	类型	复位	描述
15:4	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	INTPWM3	R/W	0	<p>PWM3 中断使能</p> <p>值 描述</p> <p>1 当 PWM 发生器模块 3 发生中断时，中断将被发送到中断控制器。</p> <p>0 PWM 发生器 3 中断被抑制，不会发送到中断控制器。</p>
2	INTPWM2	R/W	0	<p>PWM2 中断使能</p> <p>值 描述</p> <p>1 当 PWM 发生器模块 2 发生中断时，中断将被发送到中断控制器。</p> <p>0 PWM 发生器 2 中断被抑制，不会发送到中断控制器。</p>
1	INTPWM1	R/W	0	<p>PWM1 中断使能</p> <p>值 描述</p> <p>1 当 PWM 发生器模块 1 发生中断时，中断将被发送到中断控制器。</p> <p>0 PWM 发生器 1 中断被抑制，不会发送到中断控制器。</p>
0	INTPWM0	R/W	0	<p>PWM0 中断使能</p> <p>值 描述</p> <p>1 当 PWM 发生器模块 0 发生中断时，中断将被发送到中断控制器。</p> <p>0 PWM 发生器 0 中断被抑制，不会发送到中断控制器。</p>

## 寄存器 7: PWM 原始中断状态寄存器 (PWMRIS), 偏移量 0x018

该寄存器提供的是已发生的中断源的当前设置, 而不管它们是否被提交给中断控制器。根据 PWMnCTL、PWMnFLTSRC0 和 PWMnFLTSRC1 寄存器指定的故障条件源确认故障中断。故障中断被锁存并且必须通过 PWM 中断状态和清除 (PWMISC) 寄存器来清除。可以通过 PWMSTATUS 寄存器查看 MnFAULTn 信号的实际值。

PWM 发生器中断只简单地反映了 PWM 发生器的状态, 它们通过 PWM 发生器模块 中的中断状态寄存器清除。如果其中一位被置位, 那么表示该事件激活; 如果是 0, 则表示事件处于不活动状态。

### PWM 原始中断状态寄存器 (PWMRIS)

PWM0x4002.8000 (0x4002.8FFF 的结束地址) 基址: 0x4002.8000

PWM10x4002.9000 (0x4002.9FFF 的结束地址) 基址: 0x4002.9000

偏移量 0x018

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留												INTFAULT3	INTFAULT2	INTFAULT1	INTFAULT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												INTPWM3	INTPWM2	INTPWM1	INTPWM0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:20	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
19	INTFAULT3	RO	0	故障中断 PWM 3  值 描述 1 PWM 发生器 3 的故障条件确认。 0 PWM 发生器 3 的故障条件中断未确认。  可以通过将 PWMISC 寄存器中 INTFAULT3 位置位来把该位清零。
18	INTFAULT2	RO	0	故障中断 PWM 2  值 描述 1 PWM 发生器 2 的故障条件确认。 0 PWM 发生器 2 的故障条件中断未确认。  可以通过将 PWMISC 寄存器中 INTFAULT2 位置位来把该位清零。
17	INTFAULT1	RO	0	故障中断 PWM 1  值 描述 1 PWM 发生器 1 的故障条件确认。 0 PWM 发生器 1 的故障条件中断未确认。  可以通过将 PWMISC 寄存器中 INTFAULT1 位置位来把该位清零。

位/域	名称	类型	复位	描述
16	INTFAULT0	RO	0	故障中断 PWM 0  值 描述 1 PWM 发生器 0 的故障条件确认。 0 PWM 发生器 0 的故障条件中断未确认。  可以通过将 PWMISC 寄存器中 INTFAULT0 位置位来把该位清零。
15:4	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	INTPWM3	RO	0	PWM3 中断有效  值 描述 1 PWM 发生器 3 模块中断确认。 0 PWM 发生器 3 模块中断未确认。  PWM3RIS 寄存器显示了该中断的来源。可通过向 PWM3ISC 寄存器的相应位写入 1 来将此位清零。
2	INTPWM2	RO	0	PWM2 中断有效  值 描述 1 PWM 发生器 2 模块中断确认。 0 PWM 发生器 2 模块中断未确认。  PWM2RIS 寄存器显示了该中断的来源。可通过向 PWM2ISC 寄存器的相应位写入 1 来将此位清零。
1	INTPWM1	RO	0	PWM1 中断有效  值 描述 1 PWM 发生器 1 模块中断确认。 0 PWM 发生器 1 模块中断未确认。  PWM1RIS 寄存器显示了该中断的来源。可通过向 PWM1ISC 寄存器的相应位写入 1 来将此位清零。
0	INTPWM0	RO	0	PWM0 中断有效  值 描述 1 PWM 发生器 0 模块中断确认。 0 PWM 发生器 0 模块中断未确认。  PWM0RIS 寄存器显示了该中断的来源。可通过向 PWM0ISC 寄存器的相应位写入 1 来将此位清零。



## 寄存器 8: PWM 中断状态和清除寄存器 ( PWMISC ) , 偏移量 0x01C

该寄存器汇总了单个PWM发生器模块的中断状态如果一个故障中断位被置位, 表示相应的MnFAULTn输入导致了中断的发生。对于故障中断, 向该位写 1 会清除锁存的中断状态。如果某模块中断位被置位, 那么表示相应发生器模块正在发出中断。通过查询每个模块中的 单个中断状态寄存器 PWMnISC, 可以确定中断原因和清除中断。

### PWM 中断状态和清除寄存器 (PWMISC)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x01C

类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留												INTFAULT3	INTFAULT2	INTFAULT1	INTFAULT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												INTPWM3	INTPWM2	INTPWM1	INTPWM0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:20	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
19	INTFAULT3	R/W1C	0	FAULT3 中断有效  值 描述 1 已为 PWM 发生器 3 的故障条件启用的中断已确认或锁存。 0 PWM 发生器 3 的故障条件未确认或未启用。  向该位写入 1 会将该位和 PWMRIS 寄存器的 INTFAULT3 位清零。
18	INTFAULT2	R/W1C	0	FAULT2 中断有效  值 描述 1 已为 PWM 发生器 2 的故障条件启用的中断已确认或锁存。 0 PWM 发生器 2 的故障条件未确认或未启用。  向该位写入 1 会将该位和 PWMRIS 寄存器的 INTFAULT2 位清零。
17	INTFAULT1	R/W1C	0	FAULT1 中断确认  值 描述 1 已为 PWM 发生器 1 的故障条件启用的中断已确认或锁存。 0 PWM 发生器 1 的故障条件未确认或未启用。  向该位写入 1 会将该位和 PWMRIS 寄存器的 INTFAULT1 位清零。

位/域	名称	类型	复位	描述
16	INTFAULT0	R/W1C	0	<p>FAULT0 中断确认</p> <p>值 描述</p> <p>1 已为 PWM 发生器 0 的故障条件启用的中断已确认或锁存。</p> <p>0 PWM 发生器 0 的故障条件未确认或未启用。</p> <p>向该位写入 1 会将该位和 PWMRIS 寄存器的 INTFAULT0 位清零。</p>
15:4	保留	RO	0x000	<p>软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。</p>
3	INTPWM3	RO	0	<p>PWM3 中断状态</p> <p>值 描述</p> <p>1 已为 PWM 发生器 3 的启用的中断已确认。</p> <p>0 PWM 发生器 3 模块的中断未确认或未启用。</p> <p>PWM3RIS 寄存器显示了该中断的来源。可通过向 PWM3ISC 寄存器的相应位写入 1 来将此位清零。</p>
2	INTPWM2	RO	0	<p>PWM2 中断状态</p> <p>值 描述</p> <p>1 已为 PWM 发生器 2 的启用的中断已确认。</p> <p>0 PWM 发生器 2 模块的中断未确认或未启用。</p> <p>PWM2RIS 寄存器显示了该中断的来源。可通过向 PWM2ISC 寄存器的相应位写入 1 来将此位清零。</p>
1	INTPWM1	RO	0	<p>PWM1 中断状态</p> <p>值 描述</p> <p>1 已为 PWM 发生器 1 的启用的中断已确认。</p> <p>0 PWM 发生器 1 模块的中断未确认或未启用。</p> <p>PWM1RIS 寄存器显示了该中断的来源。可通过向 PWM1ISC 寄存器的相应位写入 1 来将此位清零。</p>
0	INTPWM0	RO	0	<p>PWM0 中断状态</p> <p>值 描述</p> <p>1 已为 PWM 发生器 0 的启用的中断已确认。</p> <p>0 PWM 发生器 0 模块的中断未确认或未启用。</p> <p>PWM0RIS 寄存器显示了该中断的来源。可通过向 PWM0ISC 寄存器的相应位写入 1 来将此位清零。</p>

## 寄存器 9: PWM 状态寄存器 ( PWMSTATUS ) , 偏移量 0x020

该寄存器用于显示 PWM 发生器未锁的故障条件。

### PWM 状态寄存器 (PWMSTATUS)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x020

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												FAULT3	FAULT2	FAULT1	FAULT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	FAULT3	RO	0	发生器 3 的故障状态  值 描述 1 PWM 发生器 3 的故障条件确认。 如果 PWM3CTL 寄存器的 FLT SRC 位被清零，输入将成为故障条件源并被确认。 0 PWM 发生器 3 故障条件未确认。
2	FAULT2	RO	0	发生器 2 的故障状态  值 描述 1 PWM 发生器 2 的故障条件确认。 如果 PWM2CTL 寄存器的 FLT SRC 位被清零，输入将成为故障条件源并被确认。 0 PWM 发生器 2 故障条件未确认。
1	FAULT1	RO	0	发生器 1 的故障状态  值 描述 1 PWM 发生器 1 的故障条件确认。 如果 PWM1CTL 寄存器的 FLT SRC 位被清零，输入将成为故障条件源并被确认。 0 PWM 发生器 1 故障条件未确认。

位/域	名称	类型	复位	描述
0	FAULT0	RO	0	发生器 0 的故障状态  值 描述 1 PWM 发生器 0 的故障条件确认。 如果 PWM0CTL 寄存器的 FLT SRC 位被清零，输入将成为故障条件源并被确认。 0 PWM 发生器 0 故障条件未确认。

## 寄存器 10: PWM 故障条件值寄存器 ( PWMFAULTVAL ) , 偏移量 0x024

如果通过 PWMFAULT 寄存器的相应位启用了故障条件, 那么该寄存器指定在故障条件下 MnPWMn 信号的输出值。注, 如果 PWMINVERT 寄存器中相应的位被置位, 则输出的是此寄存器中的值取逻辑非时的值。

### PWM 故障条件值寄存器 (PWMFAULTVAL)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x024

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	PWM7	R/W	0	MnPWM7 故障值  值 描述 1 如果 PWMFAULT 寄存器中的 FAULT7 位被置位, 则在故障条件下 MnPWM7 将输出高电平。 0 如果 PWMFAULT 寄存器中的 FAULT7 位被置位, 则在故障条件下 MnPWM7 将输出低电平。
6	PWM6	R/W	0	MnPWM6 故障值  值 描述 1 如果 PWMFAULT 寄存器中的 FAULT6 位被置位, 则在故障条件下 MnPWM6 将输出高电平。 0 如果 PWMFAULT 寄存器中的 FAULT6 位被置位, 则在故障条件下 MnPWM6 将输出低电平。
5	PWM5	R/W	0	MnPWM5 故障值  值 描述 1 如果 PWMFAULT 寄存器中的 FAULT5 位被置位, 则在故障条件下 MnPWM5 将输出高电平。 0 如果 PWMFAULT 寄存器中的 FAULT5 位被置位, 则在故障条件下 MnPWM5 将输出低电平。

位/域	名称	类型	复位	描述
4	PWM4	R/W	0	MnPWM4 故障值  值 描述 1 如果 PWMFAULT 寄存器中的 FAULT4 位被置位，则在故障条件下 MnPWM4 将输出高电平。 0 如果 PWMFAULT 寄存器中的 FAULT4 位被置位，则在故障条件下 MnPWM4 将输出低电平。
3	PWM3	R/W	0	MnPWM3 故障值  值 描述 1 如果 PWMFAULT 寄存器中的 FAULT3 位被置位，则在故障条件下 MnPWM3 将输出高电平。 0 如果 PWMFAULT 寄存器中的 FAULT3 位被置位，则在故障条件下 MnPWM3 将输出低电平。
2	PWM2	R/W	0	MnPWM2 故障值  值 描述 1 如果 PWMFAULT 寄存器中的 FAULT2 位被置位，则在故障条件下 MnPWM2 将输出高电平。 0 如果 PWMFAULT 寄存器中的 FAULT2 位被置位，则在故障条件下 MnPWM2 将输出低电平。
1	PWM1	R/W	0	MnPWM1 故障值  值 描述 1 如果 PWMFAULT 寄存器中的 FAULT1 位被置位，则在故障条件下 MnPWM1 将输出高电平。 0 如果 PWMFAULT 寄存器中的 FAULT1 位被置位，则在故障条件下 MnPWM1 将输出低电平。
0	PWM0	R/W	0	MnPWM0 故障值  值 描述 1 如果 PWMFAULT 寄存器中的 FAULT0 位被置位，则在故障条件下 MnPWM0 将输出高电平。 0 如果 PWMFAULT 寄存器中的 FAULT0 位被置位，则在故障条件下 MnPWM0 将输出低电平。

## 寄存器 11: PWM 使能更新寄存器 ( PWMENUPD ) , 偏移量 0x028

该寄存器指定何时更新 PWMENABLE 寄存器中的 PWMnEN 位。PWMnEN 位启用将传递到微控制器的管脚的 pwmA' 和 pwmB' 输出。更新可以是立即、本地或全局同步到下一个同步更新。

## PWM 使能更新寄存器 (PWMENUPD)

PWM00x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x028

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENUPD7		ENUPD6		ENUPD5		ENUPD4		ENUPD3		ENUPD2		ENUPD1		ENUPD0	
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
15:14	ENUPD7	R/W	0	MnPWM07 启用更新模式  值 描述 0x0 立即更新 PWM 发生器立即使用 PWMENABLE 寄存器中的 PWM7EN 位写入的数据。 0x1 保留 0x2 本地同步 在下一次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM7EN 位写入的数据。 0x3 全局同步 通过 PWM 主机控制 (PWMCTL) 寄存器请求同步更新之后，下一次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM7EN 位写入的数据。
13:12	ENUPD6	R/W	0	MnPWM6 启用更新模式  值 描述 0x0 立即更新 PWM 发生器立即使用 PWMENABLE 寄存器中的 PWM6EN 位写入的数据。 0x1 保留 0x2 本地同步 在下一次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM6EN 位写入的数据。 0x3 全局同步 通过 PWM 主机控制 (PWMCTL) 寄存器请求同步更新之后，下一次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM6EN 位写入的数据。

位/域	名称	类型	复位	描述
11:10	ENUPD5	R/W	0	<p>MnPWM5 启用更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWM 发生器立即使用 PWMENABLE 寄存器中的 PWM5EN 位写入的数据。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下次计数器计数到 0 时, PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM5EN 位写入的数据。</p> <p>0x3 全局同步 通过 PWM 主机控制 (PWMCTL) 寄存器请求同步更新之后, 下次计数器计数到 0 时, PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM5EN 位写入的数据。</p>
9:8	ENUPD4	R/W	0	<p>MnPWM4 启用更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWM 发生器立即使用 PWMENABLE 寄存器中的 PWM4EN 位写入的数据。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下次计数器计数到 0 时, PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM4EN 位写入的数据。</p> <p>0x3 全局同步 通过 PWM 主机控制 (PWMCTL) 寄存器请求同步更新之后, 下次计数器计数到 0 时, PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM4EN 位写入的数据。</p>
7:6	ENUPD3	R/W	0	<p>MnPWM3 启用更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWM 发生器立即使用 PWMENABLE 寄存器中的 PWM3EN 位写入的数据。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下次计数器计数到 0 时, PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM3EN 位写入的数据。</p> <p>0x3 全局同步 通过 PWM 主机控制 (PWMCTL) 寄存器请求同步更新之后, 下次计数器计数到 0 时, PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM3EN 位写入的数据。</p>



位/域	名称	类型	复位	描述
5:4	ENUPD2	R/W	0	<p>MnPWM2 启用更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWM 发生器立即使用 PWMENABLE 寄存器中的 PWM2EN 位写入的数据。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM2EN 位写入的数据。</p> <p>0x3 全局同步 通过 PWM 主机控制 (PWMCTL) 寄存器请求同步更新之后，下次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM2EN 位写入的数据。</p>
3:2	ENUPD1	R/W	0	<p>MnPWM1 启用更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWM 发生器立即使用 PWMENABLE 寄存器中的 PWM1EN 位写入的数据。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM1EN 位写入的数据。</p> <p>0x3 全局同步 通过 PWM 主机控制 (PWMCTL) 寄存器请求同步更新之后，下次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM1EN 位写入的数据。</p>
1:0	ENUPD0	R/W	0	<p>MnPWM0 启用更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWM 发生器立即使用 PWMENABLE 寄存器中的 PWM0EN 位写入的数据。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM0EN 位写入的数据。</p> <p>0x3 全局同步 通过 PWM 主机控制 (PWMCTL) 寄存器请求同步更新之后，下次计数器计数到 0 时，PWM 发生器将使用向 PWMENABLE 寄存器中的 PWM0EN 位写入的数据。</p>

寄存器 12: PWM0 控制寄存器 ( PWM0CTL ) , 偏移量 0x040

寄存器 13: PWM1 控制寄存器 ( PWM1CTL ) , 偏移量 0x080

寄存器 14: PWM2 控制寄存器 ( PWM2CTL ) , 偏移量 0x0C0

寄存器 15: PWM3 控制寄存器 ( PWM3CTL ) , 偏移量 0x100

这些寄存器用来配置 PWM 信号发生模块 ( PWM0CTL 控制 PWM 发生器 0 模块, 依此类推)。寄存器更新模式、调试模式、计数模式和模块使能模式都是通过这些寄存器控制。该模块可以产生两个独立的 PWM 信号 ( 来自同一个计数器 ) , 或一对增加了死区延迟的 PWM 信号。

PWM0 模块产生 MnPWM0 和 MnPWM1 输出, PWM1 模块产生 MnPWM2 和 MnPWM3 输出, PWM2 模块产生 MnPWM4 和 MnPWM5 输出, PWM3 模块产生 MnPWM6 和 MnPWM7 输出。

#### PWMn 控制寄存器 ( PWMnCTL )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x040

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留													LATCH	MINFLTPER	FLTSRC
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBFALLUPD		DBRISEUPD		DBCTLUPD		GENBUPD		GENAUPD		CMPBUPD	CMPAUPD	LOADUPD	DEBUG	MODE	ENABLE
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:19	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
18	LATCH	R/W	0	锁存故障输入  值 描述 0 故障条件未锁存 只要故障源发出故障, 故障条件有效 1 故障条件已锁存 确认故障源就会产生一个故障条件。在 PWMISCINTFAULTn 被置位时, 该故障条件将被锁存。将 INTFAULTn 位清零会清除该故障条件。

位/域	名称	类型	复位	描述
17	MINFLTPER	R/W	0	<p>最小故障周期</p> <p>该位指定 PWM 发生器使能一个单次触发计数器来提供一个最小故障条件周期。</p> <p>定时器在故障条件的上升沿开始计数并将故障条件延迟一段时间。定时器在计数的时候忽略故障的状态。</p> <p>只有当 MINFLTPER 位被置位时，最小故障延迟才有效。如果故障条件在处理延时的过程中 MINFLTPER 位清零了，故障条件延迟将被终止。</p> <p>延迟时间由 PWMnMINFLTPER 寄存器中的 MFP 域值指定。这个的作用是宽故障条件输入的脉冲。</p> <p>该延迟数值由 PWM 时钟周期定义。因为故障输入与 PWM 时钟不同步，所以该时间值 = PWMClock * (MFP 值 + 1) 或者 PWMClock * (MFP 值 + 2)。</p> <p>延迟功能仅在故障源未被锁定的情况下有效。锁定的故障源会一直产生故障条件，直到通过软件清除，而且不能使用扩展功能。这对 FLT SRC 域中指定的故障条件源都有效。</p> <p>值 描述</p> <p>0 故障输入无效不受影响。</p> <p>1 PWMnMINFLTPER 单次触发定时器有效，它会将故障条件的时间扩展一个最小周期。</p>
16	FLTSRC	R/W	0	<p>故障条件源</p> <p>值 描述</p> <p>0 故障条件由 Fault0 输入确定。</p> <p>1 故障条件由 PWMnFLTSRC0 和 PWMnFLTSRC1 寄存器的配置确定。</p>
15:14	DBFALLUPD	R/W	0x0	<p>PWMnDBFALL 更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWMnDBFALL 寄存器的值在写入动作时会立即更新。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下一次计数器为 0 时，寄存器的更新会反映到发生器。</p> <p>0x3 全局同步 PWMCTL 寄存器发出同步更新要求以后，一直到下次计数器为 0 时寄存器才会更新。</p>

位/域	名称	类型	复位	描述
13:12	DBRISEUPD	R/W	0x0	<p>PWMnDBRISE 更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWMnDBRISE 寄存器的值在写入动作时会立即更新。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下一次计数器为 0 时，寄存器的更新会反映到发生器。</p> <p>0x3 全局同步 PWMCTL 寄存器发出同步更新要求以后，一直到下次计数器为 0 时寄存器才会更新。</p>
11:10	DBCTLUPD	R/W	0x0	<p>PWMnDBCTL 更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWMnDBCTL 寄存器的值在写入动作时会立即更新。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下一次计数器为 0 时，寄存器的更新会反映到发生器。</p> <p>0x3 全局同步 PWMCTL 寄存器发出同步更新要求以后，一直到下次计数器为 0 时寄存器才会更新。</p>
9:8	GENBUPD	R/W	0x0	<p>PWMnGENB 更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWMnGENB 寄存器的值在写入动作时会立即更新。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下一次计数器为 0 时，寄存器的更新会反映到发生器。</p> <p>0x3 全局同步 PWMCTL 寄存器发出同步更新要求以后，一直到下次计数器为 0 时寄存器才会更新。</p>
7:6	GENAUPD	R/W	0x0	<p>PWMnGENA 更新模式</p> <p>值 描述</p> <p>0x0 立即更新 PWMnGENA 寄存器的值在写入动作时会立即更新。</p> <p>0x1 保留</p> <p>0x2 本地同步 在下一次计数器为 0 时，寄存器的更新会反映到发生器。</p> <p>0x3 全局同步 PWMCTL 寄存器发出同步更新要求以后，一直到下次计数器为 0 时寄存器才会更新。</p>

位/域	名称	类型	复位	描述
5	CMPBUPD	R/W	0	<p>比较器 B 更新模式</p> <p>值 描述</p> <p>0 本地同步 在下次计数器为 0 时，PWMnCMPB 寄存器的更新会反映到发生器。</p> <p>1 全局同步 PWMCTL 寄存器发出同步更新要求以后，一直到下次计数器为 0 时寄存器才会更新。</p>
4	CMPAUPD	R/W	0	<p>比较器 A 更新模式</p> <p>值 描述</p> <p>0 本地同步 在下次计数器为 0 时，PWMnCMPA 寄存器的更新会反映到发生器。</p> <p>1 全局同步 PWMCTL 寄存器发出同步更新要求以后，一直到下次计数器为 0 时寄存器才会更新。</p>
3	LOADUPD	R/W	0	<p>加载寄存器更新模式</p> <p>值 描述</p> <p>0 本地同步 在下次计数器为 0 时，PWMnLOAD 寄存器的更新会反映到发生器。</p> <p>1 全局同步 PWMCTL 寄存器发出同步更新要求以后，一直到下次计数器为 0 时寄存器才会更新。</p>
2	DEBUG	R/W	0	<p>调试模式</p> <p>值 描述</p> <p>0 当计数到 0 时计数器停止，并在退出调试模式后继续运行。</p> <p>1 计数器在调试模式下会一直运行。</p>
1	MODE	R/W	0	<p>计数器模式</p> <p>值 描述</p> <p>0 计数器计数从装载值减到 0 之后再返回到装载值（递减模式）。</p> <p>1 计数器从 0 递增到装载值再递减到 0，然后反复（递增/递减模式）。</p>
0	ENABLE	R/W	0	<p>PWM 模块启用</p> <p>值 描述</p> <p>0 整个 PWM 发生器模块被禁用且不计数。</p> <p>1 PWM 发生器模块启用，并产生 PWM 信号。</p>

寄存器 16: PWM0 中断和触发使能寄存器 ( PWM0INTEN ) , 偏移量 0x044

寄存器 17: PWM1 中断和触发使能寄存器 ( PWM1INTEN ) , 偏移量 0x084

寄存器 18: PWM2 中断和触发使能寄存器 ( PWM2INTEN ) , 偏移量 0x0C4

寄存器 19: PWM3 中断和触发使能寄存器 ( PWM3INTEN ) , 偏移量 0x104

这些寄存器用于控制 PWM 发生器产生中断和 ADC 触发的能力 ( PWM0INTEN 控制 PWM 发生器 0 模块, 以此类推 )。能够引起中断或 ADC 触发的事件包括:

- 计数器的计数值等于装载寄存器的值
- 计数器的计数值等于零
- 递增计数时, 计数器的计数值等于 PWMnCMPA 寄存器中的值
- 递减计数时, 计数器的计数值等于 PWMnCMPA 寄存器中的值
- 递增计数时, 计数器的计数值等于 PWMnCMPB 寄存器中的值
- 递减计数时, 计数器的计数值等于 PWMnCMPB 寄存器中的值

上述事件的任何组合都可以产生中断或者 ADC 触发, 尽管出现多于一个事件时不能确定实际引起 ADC 触发的为何事件。PWMnRIS 寄存器显示了哪些事件触发了原始中断。

#### PWMn 中断和触发使能寄存器 ( PWMnINTEN )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x044

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		TRCMPBD	TRCMPBU	TRCMPAD	TRCMPAU	TRCNTLOAD	TRCNTZERO	保留		INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:14	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
13	TRCMPBD	R/W	0	计数器触发 = PWMnCMPB 递减  值 描述 1 在递减计数时, 当计数器的值与 PWMnCMPB 寄存器的值相等时, ADC 触发脉冲被输出。 0 无 ADC 触发输出。

位/域	名称	类型	复位	描述
12	TRCMPBU	R/W	0	计数器触发 = PWMnCMPB 递增  值 描述 1 在递增计数时，当计数器的值与 PWMnCMPB 寄存器的值相等时，ADC 触发脉冲被输出。 0 无 ADC 触发输出。
11	TRCMPAD	R/W	0	计数器触发 = PWMnCMPA 递减  值 描述 1 在递减计数时，当计数器的值与 PWMnCMPA 寄存器的值相等时，ADC 触发脉冲被输出。 0 无 ADC 触发输出。
10	TRCMPAU	R/W	0	计数器触发 = PWMnCMPA 递增  值 描述 1 在递增计数时，当计数器的值与 PWMnCMPA 寄存器的值相等时，ADC 触发脉冲被输出。 0 无 ADC 触发输出。
9	TRCNTLOAD	R/W	0	计数器触发 = PWMnLOAD  值 描述 1 当计数器的值与 PWMnLOAD 寄存器的值相等时，ADC 触发脉冲被输出。 0 无 ADC 触发输出。
8	TRCNTZERO	R/W	0	计数器触发 = 0  值 描述 1 当计数器的计数值为 0 时，ADC 触发脉冲被输出。 0 无 ADC 触发输出。
7:6	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
5	INTCMPBD	R/W	0	计数器中断 = PWMnCMPB 递减  值 描述 1 在递减计数时，当计数器的值与 PWMnCMPB 寄存器的值相等时产生一个原始中断。 0 无中断。

位/域	名称	类型	复位	描述
4	INTCMPBU	R/W	0	计数器中断 = PWMnCMPB 递增  值 描述 1 在递增计数时，当计数器的值与 PWMnCMPB 寄存器的值相等时产生一个原始中断。 0 无中断。
3	INTCMPAD	R/W	0	计数器中断 = PWMnCMPA 递减  值 描述 1 在递减计数时，当计数器的值与 PWMnCMPA 寄存器的值相等时产生一个原始中断。 0 无中断。
2	INTCMPAU	R/W	0	计数器中断 = PWMnCMPA 递增  值 描述 1 在递增计数时，当计数器的值与 PWMnCMPA 寄存器的值相等时产生一个原始中断。 0 无中断。
1	INTCNTLOAD	R/W	0	计数器中断 = PWMnLOAD  值 描述 1 当计数器的值与 PWMnLOAD 寄存器值相等时产生一个原始中断。 0 无中断。
0	INTCNTZERO	R/W	0	计数器中断 = 0  值 描述 1 当计数器的值为 0 时产生一个原始中断。 0 无中断。



寄存器 20: PWM0 原始中断状态寄存器 ( PWM0RIS ) , 偏移量 0x048

寄存器 21: PWM1 原始中断状态寄存器 ( PWM1RIS ) , 偏移量 0x088

寄存器 22: PWM2 原始中断状态寄存器 ( PWM2RIS ) , 偏移量 0x0C8

寄存器 23: PWM3 原始中断状态寄存器 ( PWM3RIS ) , 偏移量 0x108

这些寄存器显示了已确认的中断源的当前设置, 不管中断是否被提交到控制器 ( PWM0RIS 控制 PWM 发生器 0 模块, 依此类推 )。寄存器中的某一位为 1 时, 表示相应的事件发生; 为 0 时, 表示相应事件没有发生。将 PWMnISC 寄存器中的相应位写 1 可以把此寄存器中的位清零。

#### PWMn 原始中断状态寄存器 ( PWMnRIS )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x048

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	INTCMPBD	RO	0	比较器 B 递减中断状态  值 描述 1 在递减计数时, 计数器的值与 PWMnCMPB 寄存器的值相等。 0 未产生中断。  可以通过在 PWMnISC 寄存器中的 INTCMPBD 位写入 1 来把该位清零。
4	INTCMPBU	RO	0	比较器 B 递增中断状态  值 描述 1 在递增计数时, 计数器的值与 PWMnCMPB 寄存器的值相等。 0 未产生中断。  可以通过在 PWMnISC 寄存器中的 INTCMPBU 位写入 1 来把该位清零。
3	INTCMPAD	RO	0	比较器 A 递减中断状态  值 描述 1 在递减计数时, 计数器的值与 PWMnCMPA 寄存器的值相等。 0 未产生中断。  可以通过在 PWMnISC 寄存器中的 INTCMPAD 位写入 1 来把该位清零。

位/域	名称	类型	复位	描述
2	INTCMPAU	RO	0	<p>比较器 A 递增中断状态</p> <p>值 描述</p> <p>1 在递增计数时，计数器的值与 PWMnCMPA 寄存器的值相等。</p> <p>0 未产生中断。</p> <p>可以通过在 PWMnISC 寄存器中的 INTCMPAU 位写入 1 来把该位清零。</p>
1	INTCNTLOAD	RO	0	<p>计数器 = 装载值的中断状态</p> <p>值 描述</p> <p>1 表示计数器的值与 PWMnLOAD 寄存器的值相等。</p> <p>0 未产生中断。</p> <p>可以通过在 PWMnISC 寄存器中的 INTCNTLOAD 位写入 1 来把该位清零。</p>
0	INTCNTZERO	RO	0	<p>计数器 = 0 的中断状态</p> <p>值 描述</p> <p>1 表示计数器等于 0。</p> <p>0 未产生中断。</p> <p>可以通过在 PWMnISC 寄存器中的 INTCNTZERO 位写入 1 来把该位清零。</p>

寄存器 24: PWM0 中断状态和清除寄存器 ( PWM0ISC ) , 偏移量 0x04C

寄存器 25: PWM1 中断状态和清除寄存器 ( PWM1ISC ) , 偏移量 0x08C

寄存器 26: PWM2 中断状态和清除寄存器 ( PWM2ISC ) , 偏移量 0x0CC

寄存器 27: PWM3 中断状态和清除寄存器 ( PWM3ISC ) , 偏移量 0x10C

这些寄存器显示了已经发送到中断控制器的中断源的当前设置 ( PWM0ISC 控制 PWM 发生器 0 模块, 以此类推 )。如果事件发生, 而且在 PWMnINTEN 寄存器中使能, 相应的位会被置 1; 如果相应的位被清零, 则表示事件没有发生。它们都是 R/W1C, 即向某个位写 1 将使对应的中断原因清零。

#### PWMn 中断状态和清除寄存器 ( PWMnISC )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x04C

类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:6	保留	RO	0	软件不应该依赖保留位的值。为了兼容未来的器件, 保留位的值在读-修改-写操作过程中应该保持不变。
5	INTCMPBD	R/W1C	0	比较器 B 递减中断  值 描述 1 寄存器 PWMnRIS 和 PWMnINTEN 中的 INTCMPBD 位被置位, 向中断控制器发出中断信号。 0 未发生中断, 或该中断被屏蔽。  对该位写入 1, 即可将其清零。将该位清零也把 PWMnRIS 寄存器中的 INTCMPBD 位清零。
4	INTCMPBU	R/W1C	0	比较器 B 递增中断  值 描述 1 寄存器 PWMnRIS 和 PWMnINTEN 中的 INTCMPBU 位被置位, 向中断控制器发出中断信号。 0 未发生中断, 或该中断被屏蔽。  对该位写入 1, 即可将其清零。将该位清零也会把 PWMnRIS 寄存器中的 INTCMPBU 位清零。

位/域	名称	类型	复位	描述
3	INTCMPAD	R/W1C	0	<p>比较器 A 递减中断</p> <p>值 描述</p> <p>1 寄存器 PWMnRIS 和 PWMnINTEN 中的 INTCMPAD 位被置位，向中断控制器发出中断信号。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>对该位写入 1，即可将其清零。将该位清零也会把 PWMnRIS 寄存器中的 INTCMPAD 位清零。</p>
2	INTCMPAU	R/W1C	0	<p>比较器 A 递增中断</p> <p>值 描述</p> <p>1 寄存器 PWMnRIS 和 PWMnINTEN 中的 INTCMPAU 位被置位，向中断控制器发出中断信号。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>对该位写入 1，即可将其清零。将该位清零也会把 PWMnRIS 寄存器中的 INTCMPAU 位清零。</p>
1	INTCNTLOAD	R/W1C	0	<p>计数器 = 装载值的中断</p> <p>值 描述</p> <p>1 寄存器 PWMnRIS 和 PWMnINTEN 中的 INTCNTLOAD 位被置位，向中断控制器发出中断信号。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>对该位写入 1，即可将其清零。将该位清零也会把 PWMnRIS 寄存器中的 INTCNTLOAD 位清零。</p>
0	INTCNTZERO	R/W1C	0	<p>计数器 = 0 时的中断</p> <p>值 描述</p> <p>1 寄存器 PWMnRIS 和 PWMnINTEN 中的 INTCNTZERO 位被置位，向中断控制器发出中断信号。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>对该位写入 1，即可将其清零。将该位清零也会把 PWMnRIS 寄存器中的 INTCNTZERO 位清零。</p>

寄存器 28: PWM0 装载寄存器 ( PWM0LOAD ) , 偏移量 0x050

寄存器 29: PWM1 装载寄存器 ( PWM1LOAD ) , 偏移量 0x090

寄存器 30: PWM2 装载寄存器 ( PWM2LOAD ) , 偏移量 0x0D0

寄存器 31: PWM3 装载寄存器 ( PWM3LOAD ) , 偏移量 0x110

这些寄存器包含 PWM 计数器的加载值 ( PWM0LOAD 控制 PWM 发生器 0 模块, 以此类推 )。根据 PWMnCTL 寄存器的 MODE 位设定的计数模式, 该值可以在计数器到达零之后加载到计数器, 也可以在计数器递减到零之后, 作为递增计数的极限值。当此值与计数器的值相匹配时, 会输出一个脉冲, 该脉冲可以用来驱动 pwmA 和 pwmB 信号的产生 ( 通过 PWMnGENA/PWMnGENB 寄存器 ) 或者驱动中断或 ADC 触发 ( 通过 PWMnINTEN 寄存器 )。

如果装载值更新模式是本地同步 ( 基于 PWMnCTL 寄存器的 LOADUPD 域 ), 该 16 位的 LOAD 值将在计数器下次到达零时装载。如果是全局同步更新模式, 那么在 PWM 主机控制 (PWMCTL) 寄存器 ( 请参考 1164 页 ) 发出同步更新之后, 下一次计数器到达零时才会使用该数值。如果该寄存器在实际更新之前改写, 那么之前的值不再使用并丢弃。

#### PWMn 装载寄存器 ( PWMnLOAD )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x050

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOAD															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	LOAD	R/W	0x0000	计数器加载值 计数器加载值。

**寄存器 32: PWM0 计数寄存器 ( PWM0COUNT ) , 偏移量 0x054**

**寄存器 33: PWM1 计数寄存器 ( PWM1COUNT ) , 偏移量 0x094**

**寄存器 34: PWM2 计数寄存器 ( PWM2COUNT ) , 偏移量 0x0D4**

**寄存器 35: PWM3 计数寄存器 ( PWM3COUNT ) , 偏移量 0x114**

这些寄存器包含 PWM 计数器的当前值 ( PWM0COUNT 是 PWM 发生器 0 模块的值, 以此类推 )。当该值等于零或者与 PWMnLOAD、PWMnCMPA、或 PWMnCMPB 寄存器中的值相匹配时, 会产生一个脉冲。该脉冲能驱动 PWM 信号的产生, 或者驱动中断或 ADC 触发。

#### PWMn 计数寄存器 ( PWMnCOUNT )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x054

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COUNT															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	COUNT	RO	0x0000	计数器值 计数器的当前值。

寄存器 36: PWM0 比较寄存器 A ( PWM0CMPA ) , 偏移量 0x058

寄存器 37: PWM1 比较寄存器 A ( PWM1CMPA ) , 偏移量 0x098

寄存器 38: PWM2 比较寄存器 A ( PWM2CMPA ) , 偏移量 0x0D8

寄存器 39: PWM3 比较寄存器 A ( PWM3CMPA ) , 偏移量 0x118

这些寄存器包含用于和计数器相比较的值 ( PWM0CMPA 控制 PWM 发生器 0 模块, 以此类推 )。当该值与计数器的值相等时, 将输出一个脉冲。该脉冲能够驱动 pwmA 和 pwmB 信号的产生 ( 通过 PWMnGENA 和 PWMnGENB 寄存器 ), 或驱动中断或 ADC 触发 ( 通过 PWMnINTEN 寄存器 )。如果该寄存器的值大于 PWMnLOAD 寄存器 ( 请参考 1197 页 ) 中的值, 那么不会输出脉冲。

如果比较器 A 的更新模式为本地同步 ( 根据 PWMnCTL 寄存器的 CMPAUPD 位 ), 那么该寄存器的 16 位 COMPA 数值将在下一次计数器到达零时使用。如果是全局同步更新模式, 那么在 PWM 主机控制 (PWMCTL) 寄存器 ( 请参考 1164 页 ) 发出同步更新之后, 下一次计数器到达零时才会使用该数值。在实际更新之前, 重写该寄存器则以前的值不被采用并将丢失。

#### PWMn 比较寄存器 A ( PWMnCMPA )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x058

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COMPA															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	COMPA	R/W	0x00	比较器 A 值 与计数器进行比较的值。

寄存器 40: PWM0 比较寄存器 B ( PWM0CMPB ) , 偏移量 0x05C

寄存器 41: PWM1 比较寄存器 B ( PWM1CMPB ) , 偏移量 0x09C

寄存器 42: PWM2 比较寄存器 B ( PWM2CMPB ) , 偏移量 0x0DC

寄存器 43: PWM3 比较寄存器 B ( PWM3CMPB ) , 偏移量 0x11C

这些寄存器包含用于和计数器相比较的值 ( PWM0CMPB 控制 PWM 发生器 0 模块, 以此类推 )。当该值与计数器的值相等时, 将输出一个脉冲。该脉冲能够驱动 pwmA 和 pwmB 信号的产生 ( 通过 PWMnGENA 和 PWMnGENB 寄存器 ) , 或驱动中断或 ADC 触发 ( 通过 PWMnINTEN 寄存器 )。如果该寄存器的值大于 PWMnLOAD 寄存器的值, 则始终不输出脉冲。

如果比较器 B 的更新模式为本地同步 ( 根据 PWMnCTL 寄存器的 CMPBUPD 位 ) , 那么该寄存器的 16 位 COMPB 数值将在下一次计数器到达零时使用。如果是全局同步更新模式, 那么在 PWM 主机控制 (PWMCTL) 寄存器 ( 请参考 1164 页 ) 发出同步更新之后, 下一次计数器到达零时才会使用该数值。在实际更新之前, 重写该寄存器则以前的值不被采用并将丢失。

#### PWMn 比较寄存器 B ( PWMnCMPB )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x05C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COMPB															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	COMPB	R/W	0x0000	比较器 B 值 与计数器进行比较的值。



**寄存器 44: PWM0 发生器 A 控制寄存器 ( PWM0GENA ) , 偏移量 0x060**

**寄存器 45: PWM1 发生器 A 控制寄存器 ( PWM1GENA ) , 偏移量 0x0A0**

**寄存器 46: PWM2 发生器 A 控制寄存器 ( PWM2GENA ) , 偏移量 0x0E0**

**寄存器 47: PWM3 发生器 A 控制寄存器 ( PWM3GENA ) , 偏移量 0x120**

这些寄存器控制 pwmA 信号的产生 ( 根据计数器中的加载值和零值输出脉冲 ) , 以及比较器中比较 A 和比较 B 脉冲的产生 ( PWM0GENA 控制 PWM 发生器 0 模块, 以此类推 )。当计数器处于递减计数模式时, 只发生四个事件; 当处于递增/递减模式时, 六个事件都发生。这些事件为 PWM 信号产生的位置和占空比提供了极大的灵活性。

PWM0GENA 寄存器控制 pwm0A 信号的产生; ; PWM1GENA 寄存器控制 pwm1A 信号的产生; PWM2GENA 寄存器控制 pwm2A 信号的产生; PWM3GENA 寄存器控制 pwm3A 信号的产生。

如果零或加载事件与比较 A 或比较 B 事件同时发生, 那么发生零或加载的动作, 比较 A 或比较 B 的动作将被忽略。如果比较 A 事件与比较 B 事件同时发生, 那么发生比较 A 的动作, 比较 B 的动作将被忽略。

如果发生器 A 的更新模式是立即更新 ( 根据 PWMnCTL 寄存器中的 GENAUPD 域 ) , ACTCMPBD、ACTCMPBU、ACTCMPAD、ACTCMPAU、ACTLOAD、以及 ACTZERO 的值会被立即使用。如果更新模式是本地同步, 这些值将在下次计数器的值为 0 时被使用。如果是全局同步更新模式, 那么在 PWM 主机控制 (PWMCTL) 寄存器 ( 请参考 1164 页 ) 发出同步更新之后, 下一次计数器到达零时才会使用这些值。在实际更新之前, 重写该寄存器则以前的值不被采用并将丢失。

#### PWMn 发生器 A 控制寄存器 ( PWMnGENA )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x060

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				ACTCMPBD		ACTCMPBU		ACTCMPAD		ACTCMPAU		ACTLOAD		ACTZERO	
类型	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
11:10	ACTCMPBD	R/W	0x0	比较器 B 递减时的动作 在递减计数时, 该域指定计数器的值与比较器 B 相等时采取的动作。  值 描述 0x0 不动作。 0x1 反相 pwmA。 0x2 驱动 pwmA 为低电平。 0x3 驱动 pwmA 为高电平。

位/域	名称	类型	复位	描述
9:8	ACTCMPBU	R/W	0x0	<p>比较器 B 递增时的动作</p> <p>在递增计数时，该域指定计数器的值与比较器 B 相等时采取的动作。只有 PWMnCTL 寄存器的 MODE 位被置位时该动作才会发生。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmA。</p> <p>0x2 驱动 pwmA 为低电平。</p> <p>0x3 驱动 pwmA 为高电平。</p>
7:6	ACTCMPAD	R/W	0x0	<p>比较器 B 递减时的动作</p> <p>在递减计数时，该域指定计数器的值与比较器 A 相等时采取的动作。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmA。</p> <p>0x2 驱动 pwmA 为低电平。</p> <p>0x3 驱动 pwmA 为高电平。</p>
5:4	ACTCMPAU	R/W	0x0	<p>比较器 A 递增时的动作</p> <p>在递增计数时，该域指定计数器的值与比较器 A 相等时采取的动作。只有 PWMnCTL 寄存器的 MODE 位被置位时该动作才会发生。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmA。</p> <p>0x2 驱动 pwmA 为低电平。</p> <p>0x3 驱动 pwmA 为高电平。</p>
3:2	ACTLOAD	R/W	0x0	<p>计数器动作 =LOAD</p> <p>该域指定当计数器的值与 PWMnLOAD 寄存器中的值相等时采取的动作。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmA。</p> <p>0x2 驱动 pwmA 为低电平。</p> <p>0x3 驱动 pwmA 为高电平。</p>

位/域	名称	类型	复位	描述
1:0	ACTZERO	R/W	0x0	计数器动作 = 0 该域指定计数器的值为零时采取的动作。  值 描述 0x0 不动作。 0x1 反相 pwmA。 0x2 驱动 pwmA 为低电平。 0x3 驱动 pwmA 为高电平。

寄存器 48: PWM0 发生器 B 控制寄存器 ( PWM0GENB ) , 偏移量 0x064

寄存器 49: PWM1 发生器 B 控制寄存器 ( PWM1GENB ) , 偏移量 0x0A4

寄存器 50: PWM2 发生器 B 控制寄存器 ( PWM2GENB ) , 偏移量 0x0E4

寄存器 51: PWM3 发生器 B 控制寄存器 ( PWM3GENB ) , 偏移量 0x124

这些寄存器控制 pwmB 信号的产生 ( 根据计数器中的加载值和零值输出脉冲 ) , 以及比较器中比较 A 和比较 B 脉冲的产生 ( PWM0GENB 控制 PWM 发生器 0 模块, 以此类推 )。当计数器处于递减计数模式时, 只发生四个事件; 当处于递增/递减模式时, 六个事件都发生。这些事件为 PWM 信号产生的位置和占空比提供了极大的灵活性。

PWM0GENB 寄存器控制 pwm0B 信号的产生; PWM1GENB 寄存器控制 pwm1B 信号的产生; PWM2GENB 寄存器控制 pwm2B 信号的产生; PWM3GENB 寄存器控制 pwm3B 信号的产生。

如果零或加载事件与比较 A 或比较 B 事件同时发生, 那么发生零或加载的动作, 比较 A 或比较 B 的动作将被忽略。如果比较 A 事件与比较 B 事件同时发生, 则比较 A 动作将被忽略, 而采用比较 B 动作。

如果发生器 B 的更新模式是立即更新 ( 根据 PWMnCTL 寄存器中的 GENBUPD 域 ) , ACTCMPBD、ACTCMPBU、ACTCMPAD、ACTCMPAU、ACTLOAD 以及 ACTZERO 的值会被立即使用。如果更新模式是本地同步, 这些值将在下次计数器的值为 0 时被使用。如果是全局同步更新模式, 那么在 PWM 主机控制 (PWMCTL) 寄存器 ( 请参考 1164 页 ) 发出同步更新之后, 下一次计数器到达零时才会使用这些值。在实际更新之前, 重写该寄存器则以前的值不被采用并将丢失。

PWMn 发生器 B 控制寄存器 ( PWMnGENB ) , 偏移量 0x064

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x064

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				ACTCMPBD		ACTCMPBU		ACTCMPAD		ACTCMPAU		ACTLOAD		ACTZERO	
类型	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
11:10	ACTCMPBD	R/W	0x0	比较器 B 递减时的动作 在递减计数时, 该地段指定计数器的值与比较器 B 相等时采取的动作。  值 描述 0x0 不动作。 0x1 反相 pwmB。 0x2 驱动 pwmB 为低电平。 0x3 驱动 pwmB 为高电平。

位/域	名称	类型	复位	描述
9:8	ACTCMPBU	R/W	0x0	<p>比较器 B 递增时的动作</p> <p>在递增计数时，该域指定计数器的值与比较器 B 相等时采取的动作。只有 PWMnCTL 寄存器的 MODE 位被置位时该动作才会发生。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmB。</p> <p>0x2 驱动 pwmB 为低电平。</p> <p>0x3 驱动 pwmB 为高电平。</p>
7:6	ACTCMPAD	R/W	0x0	<p>比较器 B 递减时的动作</p> <p>在递减计数时，该域指定计数器的值与比较器 A 相等时采取的动作。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmB。</p> <p>0x2 驱动 pwmB 为低电平。</p> <p>0x3 驱动 pwmB 为高电平。</p>
5:4	ACTCMPAU	R/W	0x0	<p>比较器 A 递增时的动作</p> <p>在递增计数时，该域指定计数器的值与比较器 A 相等时采取的动作。只有 PWMnCTL 寄存器的 MODE 位被置位时该动作才会发生。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmB。</p> <p>0x2 驱动 pwmB 为低电平。</p> <p>0x3 驱动 pwmB 为高电平。</p>
3:2	ACTLOAD	R/W	0x0	<p>计数器动作 =LOAD</p> <p>该域指定计数器的值与装载值相等时采取的动作。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmB。</p> <p>0x2 驱动 pwmB 为低电平。</p> <p>0x3 驱动 pwmB 为高电平。</p>
1:0	ACTZERO	R/W	0x0	<p>计数器动作 = 0</p> <p>该域指定计数器的值为零时采取的动作。</p> <p>值 描述</p> <p>0x0 不动作。</p> <p>0x1 反相 pwmB。</p> <p>0x2 驱动 pwmB 为低电平。</p> <p>0x3 驱动 pwmB 为高电平。</p>

寄存器 52: PWM0 死区控制寄存器 ( PWM0DBCTL ) , 偏移量 0x068

寄存器 53: PWM1 死区控制寄存器 ( PWM1DBCTL ) , 偏移量 0x0A8

寄存器 54: PWM2 死区控制寄存器 ( PWM2DBCTL ) , 偏移量 0x0E8

寄存器 55: PWM3 死区控制寄存器 ( PWM3DBCTL ) , 偏移量 0x128

PWMnDBCTL 寄存器控制死区发生器, 而后者在 pwmA 和 pwmB 信号的基础上产生 MnPWMn 信号。当该寄存器被禁用时, pwmA 直接传递至 pwmA' 信号, pwmB 直接传递至 pwmB' 信号。使能死区控制时, pwmB 信号被忽略, pwmA' 信号通过对 pwmA 信号的上升沿延迟 PWMnDBRISE 寄存器 ( 请参考 1207 页 ) 中指定的值来产生, pwmB' 信号通过翻转 pwmA 信号并对 pwmA 信号的下降沿延迟 PWMnDBFALL 寄存器 ( 请参考 1208 页 ) 中指定的值来产生。输出控制模块在 MnPWM0 管脚输出 pwm0A' 信号, 在 MnPWM1 管脚输出 pwm0B' 信号。同样的, MnPWM2 和 MnPWM3 来自 pwm1A' 和 pwm1B' 信号, MnPWM4 和 MnPWM5 来自 pwm2A' 和 pwm2B' 信号, MnPWM6 和 MnPWM7 来自 pwm3A' 和 pwm3B' 信号。

如果死区控制模式是立即模式 ( 取决于 PWMnCTL 寄存器中的 DBCTLUPD 域的编码 ), ENABLE 位的值将被立即采用。如果更新模式是本地同步, 此值将在下次计数器值为 0 时被采用。如果是全局同步更新模式, 那么在 PWM 主机控制 (PWMCTL) 寄存器 ( 请参考 1164 页 ) 发出同步更新之后, 下一次计数器到达零时才会使用该数值。在实际更新之前, 重写该寄存器则以前的值不被采用并将丢失。

#### PWMn 死区控制寄存器 ( PWMnDBCTL )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x068

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															ENABLE
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:1	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
0	ENABLE	R/W	0	死区发生器启用
				值 描述
				1 死区发生器通过在 pwmA' 和 pwmB' 信号中插入死区而修改 pwmA 信号。
				0 pwmA 和 pwmB 信号直接传递给 pwmA' 和 pwmB' 信号, 未经修改。

寄存器 56: PWM0 死区上升沿延迟寄存器 ( PWM0DBRISE ) , 偏移量 0x06C

寄存器 57: PWM1 死区上升沿延迟寄存器 ( PWM1DBRISE ) , 偏移量 0x0AC

寄存器 58: PWM2 死区上升沿延迟寄存器 ( PWM2DBRISE ) , 偏移量 0x0EC

寄存器 59: PWM3 死区上升沿延迟寄存器 ( PWM3DBRISE ) , 偏移量 0x12C

PWMnDBRISE 寄存器包含产生 pwmA'信号时延迟 pwmA 信号上升沿所需要的时钟周期的数量。如果通过 PWMnDBCTL 寄存器将死区发生器禁用, 则此寄存器被忽略。如果此寄存器的值比 pwmA 信号的高电平脉冲的宽度还大, 则上升沿延迟将消耗整个信号的高电平时间, 导致输出没有高电平的时间。要确保 pwmA 的高电平时间总是比上升沿延迟时间大。

如果死区上升沿延迟模式是立即模式 ( PWMnCTL 寄存器中的 DBRISEUPD 域 ) , 那么 12 位的 RISEDELAY 值将被立即采用。如果更新模式是本地同步, 此值将在下次计数器值为 0 时被采用。如果是全局同步更新模式, 那么在 PWM 主机控制 (PWMCTL) 寄存器 ( 请参考 1164 页 ) 发出同步更新之后, 下一次计数器到达零时才会使用该数值。在实际更新之前, 重写该寄存器则以前的值不被采用并将丢失。

#### PWMn 死区上升沿延迟寄存器 ( PWMnDBRISE )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM1x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x06C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				RISEDELAY											
类型	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
11:0	RISEDELAY	R/W	0x000	死区上升延迟 在 pwmA 上升沿之后 pwmA'的上升沿延迟的时钟数量。

寄存器 60: PWM0 死区下降沿延迟寄存器 ( PWM0DBFALL ) , 偏移量 0x070

寄存器 61: PWM1 死区下降沿延迟寄存器 ( PWM1DBFALL ) , 偏移量 0x0B0

寄存器 62: PWM2 死区下降沿延迟寄存器 ( PWM2DBFALL ) , 偏移量 0x0F0

寄存器 63: PWM3 死区下降沿延迟寄存器 ( PWM3DBFALL ) , 偏移量 0x130

PWMnDBFALL 寄存器包含了从 pwmA 信号的下降沿开始延迟 pwmB'信号上升沿所需要的时钟周期的数量。如果通过 PWMnDBCTL 寄存器将死区发生器禁用, 则此寄存器被忽略。如果此寄存器的值比 pwmA 信号上的低电平脉冲的宽度要大, 则下降沿将消耗信号整个低电平的时间, 导致输出没有低电平时间。要确保 pwmA 信号的低电平时间超过下降沿延迟时间。

如果死区上升沿延迟模式是立即应用模式 ( PWMnCTL 寄存器中的 DBFALLUP 域 ) , 那么 12 位的 FALLDELAY 值将被立即采用。如果更新模式是本地同步, 此值将在下次计数器值为 0 时被采用。如果是全局同步更新模式, 那么在 PWM 主机控制 (PWMCTL) 寄存器 ( 请参考 1164 页 ) 发出同步更新之后, 下一次计数器到达零时才会使用该数值。在实际更新之前, 重写该寄存器则以前的值不被采用并将丢失。

#### PWMn 死区下降沿延迟寄存器 ( PWMnDBFALL )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x070

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				FALLDELAY											
类型	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:12	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
11:0	FALLDELAY	R/W	0x000	死区下降沿延迟 从 pwmA 上升沿开始 pwmB'的下降沿延迟的时钟周期数。



寄存器 64: PWM0 故障源寄存器 0 ( PWM0FLTSRC0 ) , 偏移量 0x074

寄存器 65: PWM1 故障源寄存器 0 ( PWM1FLTSRC0 ) , 偏移量 0x0B4

寄存器 66: PWM2 故障源寄存器 0 ( PWM2FLTSRC0 ) , 偏移量 0x0F4

寄存器 67: PWM3 故障源寄存器 0 ( PWM3FLTSRC0 ) , 偏移量 0x134

该寄存器指定某个故障引脚输入来产生一个故障状态。下面寄存器中的每一位都指示相应的故障引脚指示了相应的故障引脚是否处于故障状态。对所有启用的故障管脚进行逻辑或操作，以形成故障条件的 PWMnFLTSRC0 部分。然后，对 PWMnFLTSRC0 故障条件与 PWMnFLTSRC1 故障条件进行或运算，产生 PWM 发生器的故障条件。

如果 PWMnCTL 寄存器 ( 请参考 1186 页 ) 中的 FLTSRC 位被清零，那么只有 Fault0 信号影响产生的故障条件。否则，PWMnFLTSRC0 和 PWMnFLTSRC1 所定义的源也会影响产生的故障条件。

#### PWMn 故障源寄存器 0 ( PWMnFLTSRC0 )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x074

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													FAULT3	FAULT2	FAULT1	FAULT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	FAULT3	R/W	0	Fault3 输入  值 描述 0 Fault3 信号被抑制，无法产生故障条件。 1 Fault3 信号值与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。  注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。
2	FAULT2	R/W	0	Fault2 输入  值 描述 0 Fault2 信号被抑制，无法产生故障条件。 1 Fault2 信号值与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。  注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。

位/域	名称	类型	复位	描述
1	FAULT1	R/W	0	Fault1 输入  值 描述 0 Fault1 信号被抑制，无法产生故障条件。 1 Fault1 信号值与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。  注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。
0	FAULT0	R/W	0	Fault0 输入  值 描述 0 Fault0 信号被抑制，无法产生故障条件。 1 Fault0 信号值与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。  注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。

寄存器 68: PWM0 故障源寄存器 1 ( PWM0FLTSRC1 ) , 偏移量 1x078

寄存器 69: PWM1 故障源寄存器 1 ( PWM1FLTSRC1 ) , 偏移量 0x0B8

寄存器 70: PWM2 故障源寄存器 1 ( PWM2FLTSRC1 ) , 偏移量 0x0F8

寄存器 71: PWM3 故障源寄存器 1 ( PWM3FLTSRC1 ) , 偏移量 0x138

该寄存器指定 ADC 中的哪个数字比较器触发用来产生故障条件。下面寄存器中的每一位显示了相应的数字比较器触发是否包含在故障条件中。对所有启用的数字比较器触发进行或运算, 以形成 PWMnFLTSRC1, 作为故障条件的一部分。然后, 对 PWMnFLTSRC1 故障条件和 PWMnFLTSRC0 故障条件进行或运行, 为 PWM 发生器形成最后的故障条件。

如果 PWMnCTL 寄存器 ( 请参考 1186 页 ) 的 FLTSRC 位被清零, 则只有 PWM Fault0 信号影响产生的故障条件。否则, PWMnFLTSRC0 和 PWMnFLTSRC1 所定义的源也会影响产生的故障条件。

#### PWMn 故障源寄存器 1 ( PWMnFLTSRC1 )

PWM0 0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM1 0x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 1x078

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DCMP7	DCMP6	DCMP5	DCMP4	DCMP3	DCMP2	DCMP1	DCMP0
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	DCMP7	R/W	0	数字比较器 7  值 描述 0 来自数字比较器 7 的触发被抑制且不会, 无法产生故障条件。 1 来自数字比较器 7 的触发与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。  注意: 注: 必须将 PWMnCTL 寄存器的 FLTSRC 位置位, 否则该位无法影响故障条件的产生。
6	DCMP6	R/W	0	数字比较器 6  值 描述 0 来自数字比较器 6 的触发被抑制且不会, 无法产生故障条件。 1 来自数字比较器 6 的触发与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。  注意: 注: 必须将 PWMnCTL 寄存器的 FLTSRC 位置位, 否则该位无法影响故障条件的产生。

位/域	名称	类型	复位	描述
5	DCMP5	R/W	0	<p>数字比较器 5</p> <p>值 描述</p> <p>0 来自数字比较器 5 的触发被抑制且不会，无法产生故障条件。</p> <p>1 来自数字比较器 5 的触发与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。</p> <p>注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。</p>
4	DCMP4	R/W	0	<p>数字比较器 4</p> <p>值 描述</p> <p>0 来自数字比较器 4 的触发被抑制且不会，无法产生故障条件。</p> <p>1 来自数字比较器 4 的触发与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。</p> <p>注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。</p>
3	DCMP3	R/W	0	<p>数字比较器 3</p> <p>值 描述</p> <p>0 来自数字比较器 3 的触发被抑制且不会，无法产生故障条件。</p> <p>1 来自数字比较器 3 的触发与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。</p> <p>注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。</p>
2	DCMP2	R/W	0	<p>数字比较器 2</p> <p>值 描述</p> <p>0 来自数字比较器 2 的触发被抑制且不会，无法产生故障条件。</p> <p>1 来自数字比较器 2 的触发与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。</p> <p>注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。</p>
1	DCMP1	R/W	0	<p>数字比较器 1</p> <p>值 描述</p> <p>0 来自数字比较器 1 的触发被抑制且不会，无法产生故障条件。</p> <p>1 来自数字比较器 1 的触发与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。</p> <p>注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。</p>

位/域	名称	类型	复位	描述
0	DCMP0	R/W	0	数字比较器 0  值 描述 0 来自数字比较器 0 的触发被抑制且不会，无法产生故障条件。 1 来自数字比较器 0 的触发与其它所有能够产生故障条件的输入 ( Faultn 信号和数字比较器 ) 进行逻辑或操作。  注意: 注：必须将 PWMnCTL 寄存器的 FLTSRC 位置位，否则该位无法影响故障条件的产生。

寄存器 72: PWM0 最小故障时间寄存器 ( PWM0MINFLTPER ) , 偏移量 0x07C

寄存器 73: PWM1 最小故障时间寄存器 ( PWM1MINFLTPER ) , 偏移量 0x0BC

寄存器 74: PWM2 最小故障时间寄存器 ( PWM2MINFLTPER ) , 偏移量 0x0FC

寄存器 75: PWM3 最小故障时间寄存器 ( PWM3MINFLTPER ) , 偏移量 0x13C

如果 PWMnCTL 寄存器的 MINFLTPER 位被置位, 此寄存器指定用来扩展故障条件的 16 位延时值。该值被装载到一个 16 位递减计数器, 该计数器的值用来扩展故障条件。计数器的值到零之后的第一个时钟故障条件被释放。故障条件与 PWM 模块是异步的; 延时值是 PWM 模块周期和 ( MFP 域值 + 1 ) 或 ( MFP 域值 + 2 ) 的乘积, 取决于 PWM 时钟下故障条件何时发出。计数器以 PWM 模块时钟频率递减, 没有暂停或者其它条件。

#### PWMn 最小故障时间寄存器 ( PWMnMINFLTPER )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x07C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MFP															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:16	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
15:0	MFP	R/W	0x0000	最小故障时间 通过 PWMnCTLMINFLTPER 寄存器启用延迟之后, 故障条件延迟所需要的 PWM 时钟数量。

寄存器 76: PWM0 故障管脚逻辑感知寄存器 ( PWM0FLTSEN ) , 偏移量 0x800

寄存器 77: PWM1 故障管脚逻辑感知寄存器 ( PWM1FLTSEN ) , 偏移量 0x880

寄存器 78: PWM2 故障管脚逻辑感知寄存器 ( PWM2FLTSEN ) , 偏移量 0x900

寄存器 79: PWM3 故障管脚逻辑感知寄存器 ( PWM3FLTSEN ) , 偏移量 0x980

该寄存器定义了 PWM 故障管脚的逻辑感知。

#### PWMn 故障管脚逻辑感知寄存器 ( PWMnFLTSEN )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x800

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												FAULT3	FAULT2	FAULT1	FAULT0	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	FAULT3	R/W	0	Fault3 感知  值 描述 0 如果 Fault3 信号是高电平则指示一个错误。 1 如果 Fault3 信号是低电平则指示一个错误。
2	FAULT2	R/W	0	Fault2 感知  值 描述 0 如果 Fault2 信号是高电平则指示一个错误。 1 如果 Fault2 信号是低电平则指示一个错误。
1	FAULT1	R/W	0	Fault1 感知  值 描述 0 如果 Fault1 信号是高电平则指示一个错误。 1 如果 Fault1 信号是低电平则指示一个错误。
0	FAULT0	R/W	0	Fault0 感知  值 描述 0 如果 Fault0 信号是高电平则指示一个错误。 1 如果 Fault0 信号是低电平则指示一个错误。

寄存器 80: PWM0 故障状态寄存器 0 ( PWM0FLTSTAT0 ) , 偏移量 0x804

寄存器 81: PWM1 故障状态寄存器 0 ( PWM1FLTSTAT0 ) , 偏移量 0x884

寄存器 82: PWM2 故障状态寄存器 0 ( PWM2FLTSTAT0 ) , 偏移量 0x904

寄存器 83: PWM3 故障状态寄存器 0 ( PWM3FLTSTAT0 ) , 偏移量 0x984

配合 PWMnFLTSTAT1 寄存器, 该寄存器可提供故障条件输入的状态。

如果 PWMnCTL 寄存器的 LATCH 位被清零, 那么 PWMnFLTSTAT0 寄存器中的内容将为只读 (RO) 状态, 且提供 MnFAULTn 输入的当前状态。

如果 PWMnCTL 寄存器的 LATCH 位被置位, 那么 PWMnFLTSTAT0 寄存器中的内容将为可读/写 1 清零状态 (R/W1C), 且提供 MnFAULTn 输入的锁存版。在此模式下, 向置位写入 1 即可将该位清零。MnFAULTn 感知在发生器中调整之后, 输入会被记录。

只有启用故障源扩展功能 ( PWMnCTL 寄存器中的 FLTSRC 位被置位 ) , 才能向该寄存器写入内容。

#### PWMn 故障状态寄存器 0 ( PWMnFLTSTAT0 )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0x804

类型 -, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													FAULT3	FAULT2	FAULT1	FAULT0
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	-	-	-	-	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3	FAULT3	-	0	故障输入 3 如果 PWMnCTL 寄存器的 LATCH 位被清零, 则该位将处于只读状态, 且显示逻辑感知调整后的 MnFAULT3 输入信号的当前状态。 如果 PWMnCTL 寄存器的 LATCH 位被置位, 则该位可读/写 1 清零, 且显示逻辑感知调整后的 MnFAULT3 输入信号副本。 <ul style="list-style-type: none"> <li>如果 FAULT3 置位, 输入在事先就转变到了活动状态。</li> <li>如果 FAULT3 清零, 自从上次被清零后, 输入就没有转变到活动状态。</li> <li>在 FAULT3 位写 1 即可将其清零。</li> </ul>



位/域	名称	类型	复位	描述
2	FAULT2	-	0	<p>故障输入 2</p> <p>如果 PWMnCTL 寄存器的 LATCH 位被清零，则该位将处于只读状态，且显示逻辑感知调整后的 MnFAULT2 输入信号的当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 位被置位，则该位可读/写 1 清零，且显示逻辑感知调整后的 MnFAULT2 输入信号副本。</p> <ul style="list-style-type: none"> <li>■ 如果 FAULT2 置位，则输入将在事先转变到活动状态。</li> <li>■ 如果 FAULT2 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 FAULT2 位写 1 即可将其清零。</li> </ul>
1	FAULT1	-	0	<p>故障输入 1</p> <p>如果 PWMnCTL 寄存器的 LATCH 位被清零，则该位将处于只读状态，且显示逻辑感知调整后的 MnFAULT1 输入信号的当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 位被置位，则该位可读/写 1 清零，且显示逻辑感知调整后的 MnFAULT1 输入信号副本。</p> <ul style="list-style-type: none"> <li>■ 如果 FAULT1 置位，则输入将在事先转变到活动状态。</li> <li>■ 如果 FAULT1 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 FAULT1 位写 1 即可将其清零。</li> </ul>
0	FAULT0	-	0	<p>故障输入 0</p> <p>如果 PWMnCTL 寄存器的 LATCH 位被清零，则该位将处于只读状态，且显示逻辑感知调整后的 输入信号的当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 位被置位，则该位可读/写 1 清零，且显示逻辑感知调整后的 输入信号副本。</p> <ul style="list-style-type: none"> <li>■ 如果 FAULT0 置位，则输入将在事先转变到活动状态。</li> <li>■ 如果 FAULT0 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 FAULT0 位写 1 即可将其清零。</li> </ul>

寄存器 84: PWM0 故障状态寄存器 1 ( PWM0FLTSTAT1 ) , 偏移量 1x808

寄存器 85: PWM1 故障状态寄存器 1 ( PWM1FLTSTAT1 ) , 偏移量 1x888

寄存器 86: PWM2 故障状态寄存器 1 ( PWM2FLTSTAT1 ) , 偏移量 0x908

寄存器 87: PWM3 故障状态寄存器 1 ( PWM3FLTSTAT1 ) , 偏移量 0x988

配合 PWMnFLTSTAT0 寄存器, 该寄存器可提供关于故障条件输入的状态。

如果 PWMnCTL 寄存器的 LATCH 位被清零, 那么 PWMnFLTSTAT0 寄存器中的内容将为只读 (RO) 状态, 且提供数字比较器触发的当前状态。

如果 PWMnCTL 寄存器的 LATCH 位被置位, 那么 PWMnFLTSTAT0 寄存器中的内容将为可读/写 1 清零状态 (R/W1C), 且提供锁存版的数字比较器触发。在此模式下, 向置位位写入 1 即可将该位清零。只有启用故障源扩展功能 ( PWMnCTL 寄存器中的 FLTSRC 位被置位 ), 才能向该寄存器写入内容。

#### PWMn 故障状态寄存器 1 ( PWMnFLTSTAT1 )

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 1x808

类型 -, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DCMP7	DCMP6	DCMP5	DCMP4	DCMP3	DCMP2	DCMP1	DCMP0
类型	RO	RO	RO	RO	RO	RO	RO	RO	-	-	-	-	-	-	-	-
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:8	保留	RO	0x0000.00	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
7	DCMP7	-	0	数字比较器 7 触发 如果 PWMnCTL 寄存器的 LATCH 位清零, 那么该位代表数字比较器 7 触发输入的当前状态。 如果 PWMnCTL 寄存器的 LATCH 置位, 那么该位代表触发的副本。 <ul style="list-style-type: none"> <li>■ 如果 DCMP7 置位, 触发在事先就转变到了活动状态。</li> <li>■ 如果 DCMP7 清零, 自从上次被清零后, 触发就没有转变到活动状态。</li> <li>■ 在 DCMP7 位写 1 即可将其清零。</li> </ul>

位/域	名称	类型	复位	描述
6	DCMP6	-	0	<p>数字比较器 6 触发</p> <p>如果 PWMnCTL 寄存器的 LATCH 位清零，那么该位代表数字比较器 6 触发输入当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 置位，那么该位代表触发的副本。</p> <ul style="list-style-type: none"> <li>■ 如果 DCMP6 置位，触发在事先就转变到了活动状态。</li> <li>■ 如果 DCMP6 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 DCMP6 位写 1 即可将其清零。</li> </ul>
5	DCMP5	-	0	<p>数字比较器 5 触发</p> <p>如果 PWMnCTL 寄存器的 LATCH 位清零，那么该位代表数字比较器 5 触发输入当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 置位，那么该位代表触发的副本。</p> <ul style="list-style-type: none"> <li>■ 如果 DCMP5 置位，触发在事先就转变到了活动状态。</li> <li>■ 如果 DCMP5 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 DCMP5 位写 1 即可将其清零。</li> </ul>
4	DCMP4	-	0	<p>数字比较器 4 触发</p> <p>如果 PWMnCTL 寄存器的 LATCH 位清零，那么该位代表数字比较器 4 触发输入当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 置位，那么该位代表触发的副本。</p> <ul style="list-style-type: none"> <li>■ 如果 DCMP4 置位，触发在事先就转变到了活动状态。</li> <li>■ 如果 DCMP4 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 DCMP4 位写 1 即可将其清零。</li> </ul>
3	DCMP3	-	0	<p>数字比较器 3 触发</p> <p>如果 PWMnCTL 寄存器的 LATCH 位清零，那么该位代表数字比较器 3 触发输入当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 置位，那么该位代表触发的副本。</p> <ul style="list-style-type: none"> <li>■ 如果 DCMP3 置位，触发在事先就转变到了活动状态。</li> <li>■ 如果 DCMP3 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 DCMP3 位写 1 即可将其清零。</li> </ul>

位/域	名称	类型	复位	描述
2	DCMP2	-	0	<p>数字比较器 2 触发</p> <p>如果 PWMnCTL 寄存器的 LATCH 位清零，那么该位代表数字比较器 2 触发输入的当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 置位，那么该位代表触发的副本。</p> <ul style="list-style-type: none"> <li>■ 如果 DCMP2 置位，触发在事先就转变到了活动状态。</li> <li>■ 如果 DCMP2 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 DCMP2 位写 1 即可将其清零。</li> </ul>
1	DCMP1	-	0	<p>数字比较器 1 触发</p> <p>如果 PWMnCTL 寄存器的 LATCH 位清零，那么该位代表数字比较器 1 触发输入的当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 置位，那么该位代表触发的副本。</p> <ul style="list-style-type: none"> <li>■ 如果 DCMP1 置位，触发在事先就转变到了活动状态。</li> <li>■ 如果 DCMP1 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 DCMP1 位写 1 即可将其清零。</li> </ul>
0	DCMP0	-	0	<p>数字比较器 0 触发</p> <p>如果 PWMnCTL 寄存器的 LATCH 位清零，那么该位代表数字比较器 0 触发输入的当前状态。</p> <p>如果 PWMnCTL 寄存器的 LATCH 置位，那么该位代表触发的副本。</p> <ul style="list-style-type: none"> <li>■ 如果 DCMP0 置位，触发在事先就转变到了活动状态。</li> <li>■ 如果 DCMP0 清零，自从上次被清零后，输入就没有转变到活动状态。</li> <li>■ 在 DCMP0 位写 1 即可将其清零。</li> </ul>

## 寄存器 88: PWM 外设属性寄存器 ( PWMPP ), 偏移量 0xFC0

PWMPP 寄存器提供了 PWM 模块的属性信息。

### PWM 外设属性寄存器 (PWMPP)

PWM0x4002.8000 ( 0x4002.8FFF 的结束地址 ) 基址: 0x4002.8000

PWM10x4002.9000 ( 0x4002.9FFF 的结束地址 ) 基址: 0x4002.9000

偏移量 0xFC0

类型 RO, 复位 0x0000.0344

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留				ONE	EFAULT	ESYNC	FCNT				GCNT				
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0

位/域	名称	类型	复位	描述
31:11	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
10	ONE	RO	0x0	单次触发模式  值 描述 1 单次触发模式可用。 0 单次触发模式不可用。
9	EFAULT	RO	0x1	扩展故障  值 描述 1 扩展故障功能可用。 0 扩展故障功能不可用。
8	ESYNC	RO	0x1	扩展同步  值 描述 1 扩展同步可用。 0 扩展同步不可用。
7:4	FCNT	RO	0x4	故障输入  值 描述 0x0 无故障输入。 0x1 1 故障输入 0x2 2 故障输入 0x3 3 故障输入 0x4 4 故障输入 0x5 - 0xF 保留

位/域	名称	类型	复位	描述
3:0	GCNT	RO	0x4	发生器
				值      描述
				0x0      无发生器
				0x1      1 个发生器
				0x2      2 个发生器
				0x3      3 个发生器
				0x4      4 个发生器
				0x5 - 0xF 保留

PWM 输出的数量是 PWM 发生器数量的两倍。

## 寄存器 89: PWM 外设属性寄存器 (PWMPA), 偏移量 0xFC4

PWMPA 寄存器提供了 PWM 模块的属性信息。

## PWM 外设属性寄存器 (PWMPA)

PWMPA0x4002.8000 (0x4002.8FFF 的结束地址) 基址: 0x4002.8000

PWMPA10x4002.9000 (0x4002.9FFF 的结束地址) 基址: 0x4002.9000

偏移量 0xFC4

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留							USEPWMDIV	PWMDIV								
类型	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	

位/域	名称	类型	复位	描述
31:9	保留	RO	0x0000.0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
8	USEPWMDIV	R/W	0	<p>启用 PWM 时钟分频</p> <p>值 描述</p> <p>1 PWM 时钟分频器是 PWM 的时钟源。</p> <p>0 系统时钟是 PWM 的时钟源。</p> <p>该位的功能与传统 RCC 寄存器中的 USEPWMDIV 位相同。</p>
7:0	PWMDIV	R/W	0x7	<p>PWM 时钟分频值</p> <p>该域指定的二进制分频值是系统时钟的分频值，分频以后的时钟作为 PWM 模块的时序参考。该时钟的上升沿与系统时钟同步。</p> <p>值 描述</p> <p>0x0 /2</p> <p>0x1 /4</p> <p>0x2 /8</p> <p>0x3 /16</p> <p>0x4 /32</p> <p>0x5 /64</p> <p>0x6 /64</p> <p>0x7 /64 (默认)</p> <p>0x8 - 0xF 保留</p> <p>该位的功能与传统 RCC 寄存器中的 PWMDIV 位相同。</p>

## 21 正交编码器接口 (QEI)

正交编码器 (又名双通道增量式编码器), 用于将线性位移转换成脉冲信号。通过监控脉冲的数目和两个信号的相对相位, 用户可以跟踪位置、旋转方向和速度。此外还有第三个通道, 称为索引信号, 可用来对位置计数器进行复位, 以确定绝对位置。

该 LM4F232H5QD 微控制器包括两个正交编码器接口 (QEI) 模块。每个 QEI 模块对正交编码器轮产生的代码进行解码, 将它们解释成位置对时间的积分, 并确定旋转的方向。另外, 该接口还能捕获编码器转轮的运行速率。

该 Stellaris<sup>®</sup> LM4F232H5QD 微控制器包含两个 QEI 模块, 可以同时控制两个具有以下特性的电机:

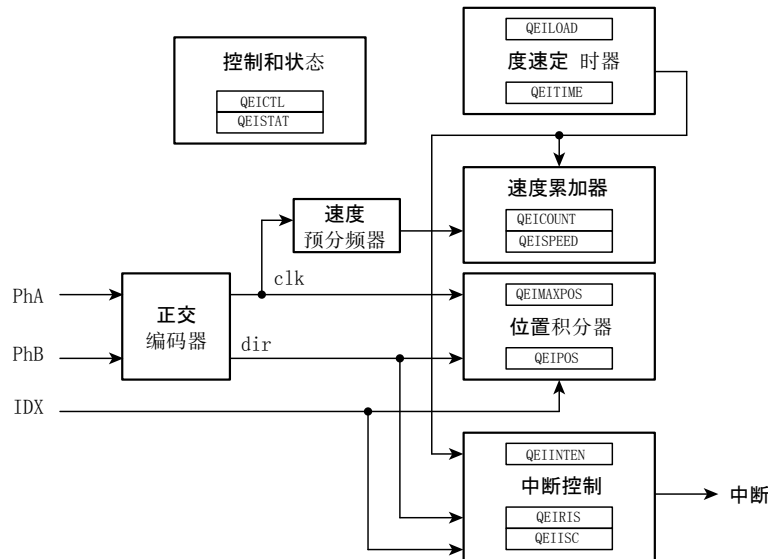
- 使用位置积分器来跟踪编码器的位置
- 输入可编程噪音过滤
- 使用内置定时器来捕获速度
- QEI 输入的频率高达 1/4 处理器频率(例如, 50 MHz 系统可达 12.5 MHz)。
- 以下情况下产生中断:
  - 检测到索引脉冲
  - 速度定时器发生计满返回事件
  - 旋转方向发生改变
  - 检测到正交错误

### 21.1 结构图

图 21-1 在 1225 页显示了 Stellaris QEI 模块的结构图。



图 21-1. QEI 结构图



## 21.2 信号描述

下面的表格列出了 QEI 模块的外部信号，并描述了每个信号的功能。QEI 信号是某些 GPIO 引脚的复用功能，在复位的时候默认是 GPIO 信号。表中“复用管脚/赋值”一栏列出了这些 QEI 信号可能的 GPIO 引脚配置。将 GPIO 备用功能选择 (GPIOAFSEL) 寄存器 (618页) 中的 AFSEL 位置位，以便选择 QEI 功能。必须将括号中的数字写入 GPIO 端口控制 (GPIOCTL) 寄存器 (636页) 的 PMCN 域中，以便把 QEI 信号分配给指定 GPIO 端口管脚。有关如何配置 GPIO，请参阅“通用输入/输出端口 (GPIO)”在 597页。

表 21-1. QEI 信号 (144LQFP)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
IDX0	4	PD3 (6)	I	TTL	QEI 模块 0 索引。
	31	PH1 (5)			
	61	PF4 (6)			
	122	PJ2 (5)			
IDX1	36	PC4 (6)	I	TTL	QEI 模块 1 索引。
	47	PG7 (5)			
	50	PG5 (6)			
PhA0	26	PH4 (5)	I	TTL	QEI 模块 0 相位 A。
	62	PF0 (6)			
	143	PD6 (6)			
PhA1	35	PC5 (6)	I	TTL	QEI 模块 1 相位 A。
	52	PG3 (6)			
	55	PG0 (6)			
PhB0	23	PH5 (5)	I	TTL	QEI 模块 0 相位 B。
	63	PF1 (6)			
	144	PD7 (6)			

表 21-1. QEI 信号 (144LQFP) (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
PhB1	34 51 54	PC6 (6) PG4 (6) PG1 (6)	I	TTL	QEI 模块 1 相位 B。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 21.3 功能描述

QEI 模块对正交编码器轮产生的两位格雷码 (gray code) 进行解码，将它们解释成位置对时间的积分并确定旋转的方向。另外，该接口还能捕获编码器转轮的运行速率。

位置积分器和速度捕获可以单独启用，可是在启用速度捕获之前必须首先启用位置积分器。PhA 和 PhB 这两个相位信号在被 QEI 模块解码前可以进行交换，以改变正向和反向的意义和纠正系统的错误接线。另外，相位信号也可以解释为时钟和方向信号，作为某些编码器的输出。

QEI 模块输入引脚上有数字噪声滤波器，启用它可以避免伪动作。在更新边沿检测器之前，噪声滤波器要求输入信号在一段连续的时钟内必须保持稳定。将 QEI 控制 (QEICTL) 寄存器中的 FILTEN 位置位可以启用该滤波器。输入信号更新的频率可以用 QEICTL 寄存器中的 FILTCNT 位域编程确定。

QEI 模块支持两种信号操作模式：正交相位模式和时钟/方向模式。在正交相位模式中，编码器产生两个相位差为 90 度的时钟信号；它们的边沿关系被用来确定旋转方向。在时钟/方向模式中，编码器产生一个时钟信号和一个方向信号，分别表示步长和旋转方向。这两种模式的选择由 QEICTL 寄存器中的 SIGMODE 位确定（请参考 1230 页）。

将 QEI 模块设置为正交相位模式 (SIGMODE 位清零) 时，位置积分器的捕获模式可设置成在 PhA 信号的上升和下降沿或是在 PhA 和 PhB 的上升和下降沿对位置计数器进行更新。在 PhA 和 PhB 的上升和下降沿上更新位置计数器提供更高精度的数据（更多位置计数），但位置计数器的计数范围却相对变少了。

当 PhA 的边沿超前于 PhB 的边沿时，位置计数器将递增。当 PhA 的边沿滞后于 PhB 的边沿时，位置计数器将递减。当一对上升沿和下降沿出现在其中一个相位上，而在其它相位上没有任何边沿时，这表示旋转方向已经发生了改变。

位置计数器在下列其中一种情况时将自动复位：1、检测到索引脉冲；2、位置计数器的值达到最大值。复位模式由 QEICTL 寄存器的 RESMODE 位确定。

当 RESMODE 位被置位时，位置计数器在检测到索引脉冲时复位。在该模式下，位置计数器的值被限制为 [0:N-1]，其中 N 为编码器轮旋转一圈得到的相位边沿数。QEI 最大位置 (QEIMAXPOS) 寄存器必须设置为 N-1，这样，从位置 0 反向就可以使位置计数器移到 N-1。在该模式中，一旦出现索引脉冲，位置寄存器就包含了编码器相对于索引（或发起）位置的绝对位置。

当 RESMODE 位被清零时，位置计数器被限制为 [0:M]，其中 M 为可编程的最大值。在该模式中，位置计数器将忽略索引脉冲。

速度捕获包含一个可配置的定时器和一个计数寄存器。定时器在给定时间周期内对相位边沿进行计数（使用与位置积分器相同的配置）。控制器通过 QEI 速度 (QEISPEED) 寄存器获得上一个时间周期的边沿计数值，而当前时间周期的边沿计数在 QEI 速度计数器 (QEICOUNT) 寄存器中进行累加。当前时间周期一结束，在该段时间内计得的边沿总数便可以从 QEISPEED 寄存器中获得（覆盖上一个值）。这时 QEICOUNT 被清零，并在一个新的时间周期开始计数。在给定时间周期内所计得的边沿数目与编码器的速度成正比例。

图 21-2 在 1227 页显示了 Stellaris 正交编码器如何将相位输入信号转换为时钟脉冲、方向信号，以及速度预分频器如何操作（在 4 分频模式中）。



QEI 模块能够在出现以下事件时产生控制器中断：相位错误、方向改变、接收到索引脉冲、速度定时器发生计满返回事件。该模块还提供标准屏蔽、原始中断状态、中断状态，以及中断清零功能。

## 21.4 初始化及配置

下面的例子说明了如何配置正交编码器模块来读回绝对位置：

1. 通过系统控制模块中的 RCGCQEI 寄存器启用 QEI 时钟（请参考 318 页）。
2. 通过系统控制模块中的 RCGCGPIO 寄存器启用相应 GPIO 模块的时钟（请参考 302 页）。
3. 在 GPIO 模块中，用 GPIOAFSEL 寄存器来启用对应管脚的复用功能。欲了解需要配置哪些 GPIO，请参阅表 23-4 在 1271 页。
4. 配置 GPIOPCTL 寄存器中的 PMCn 域，以将 QEI 信号分配到适当的管脚（请参考 636 页和表 23-5 在 1280 页）。
5. 将正交编码器配置为捕获两个信号的边沿，并在索引脉冲复位时保存绝对位置的信息。使用 1000 线编码器，每条线有 4 个边沿，因此每转一圈产生 4000 个脉冲；位置计数器从 0 开始计数，所以将最大位置计数值设置为 3999（0xF9F）。
  - 向 QEICTL 寄存器写入 0x0000.0018；
  - 向 QEIMAXPOS 寄存器写入 0x0000.0F9F。
6. 将 QEICTL 寄存器的 0 位置位，以启用正交编码器。
7. 延迟一段时间，直到系统要求提供编码器位置。
8. 读取 QEI 位置 (QEIPOS) 寄存器以获取编码器的位置信息。

## 21.5 寄存器映射

表 21-2 在 1228 页列出了 QEI 寄存器。所有列出的地址都是相对于 QEI 模块基础地址的 16 进制地址增量：

- QEI0：0x4002.C000
- QEI1：0x4002.D000

请注意，配置这些寄存器之前，必须启用 QEI 模块时钟（请参考 318 页）。QEI 模块时钟启用之后必须等待三个系统时钟的延迟，QEI 模块的寄存器才能访问。

表 21-2. QEI 寄存器映射

偏移量	名称	类型	复位	描述	见页面
0x000	QEICTL	R/W	0x0000.0000	QEI 控制寄存器	1230
0x004	QEISTAT	RO	0x0000.0000	QEI 状态寄存器	1233
0x008	QEIPOS	R/W	0x0000.0000	QEI 位置寄存器	1234
0x00C	QEIMAXPOS	R/W	0x0000.0000	QEI 最大位置寄存器	1235
0x010	QEILOAD	R/W	0x0000.0000	QEI 定时器加载寄存器	1236
0x014	QEITIME	RO	0x0000.0000	QEI 定时器寄存器	1237

表 21-2. QEI 寄存器映射 ( 续 )

偏移量	名称	类型	复位	描述	见页面
0x018	QEICOUNT	RO	0x0000.0000	QEI 速度计数寄存器	1238
0x01C	QEISPEED	RO	0x0000.0000	QEI 速度寄存器	1239
0x020	QEIINTEN	R/W	0x0000.0000	QEI 中断启用寄存器	1240
0x024	QEIRIS	RO	0x0000.0000	QEI 原始中断状态寄存器	1241
0x028	QEIISC	R/W1C	0x0000.0000	QEI 中断状态和清除寄存器	1242

## 21.6 寄存器描述

本章的剩余部分按照地址偏移量由小到大的顺序依次详细介绍各 QEI 寄存器。

### 寄存器 1: QEI 控制寄存器 (QEICTL), 偏移量 0x000

该寄存器含有 QEI 模块的配置信息。正交编码器和速度捕获模块可单独启用；在捕获速度时，必须启用正交编码器，但在不需要知道速度的应用中可以禁止捕获速率功能。另外，相位信号的解码、相位交换、位置更新模式、位置复位模式和速率预分频器也都通过该寄存器进行设置。

#### QEI 控制寄存器 (QEICTL)

QEI0 基址: 0x4002.C000  
 QEI1 基址: 0x4002.D000  
 偏移量 0x000  
 类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留												FILT CNT			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留		FILTEN	STALLEN	INVI	INVB	INVA	VELDIV			VELEN	RESMODE	CAPMODE	SIGMODE	SWAP	ENABLE
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:20	保留	RO	0x000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
19:16	FILT CNT	R/W	0x0	输入滤波器预分频数 该域控制输入信号的更新频率 如果该域清零，那么输入信号在两个系统时钟之后开始被采样。如果该域的值 0x1，那么输入信号在三个系统时钟之后开始被采样。以此类推，如果该域的值 0xF，那么输入信号在 17 个系统时钟之后开始被采样。
15:14	保留	RO	0x0	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
13	FILTEN	R/W	0	启用输入滤波器 值 描述 0 QEI 输入信号不会被过滤。 1 对 QEI 输入信号启用数字噪音滤波器。在边沿检测器更新之前，输入信号必须在三个连续的时钟边沿内保持稳定。
12	STALLEN	R/W	0	停止 QEI 值 描述 0 当微控制器被调试器停止时，QEI 模块不会停止。 1 当微控制器被调试器停止时，QEI 模块会停止。
11	INVI	R/W	0	反相索引脉冲 值 描述 0 无影响。 1 将 IDX 输入反相。

位/域	名称	类型	复位	描述
10	INVB	R/W	0	<p>将 PhB 反相</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 将 PhB 输入反相。</p>
9	INVA	R/W	0	<p>将 PhA 反相</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 将 PhA 输入反相。</p>
8:6	VELDIV	R/W	0x0	<p>预分频速度</p> <p>该域定义了输入正交脉冲在传输到 QEICOUNT 累加器之前的预分频器。</p> <p>值 预分频器</p> <p>0x0 ÷1</p> <p>0x1 ÷2</p> <p>0x2 ÷4</p> <p>0x3 ÷8</p> <p>0x4 ÷16</p> <p>0x5 ÷32</p> <p>0x6 ÷64</p> <p>0x7 ÷128</p>
5	VELEN	R/W	0	<p>捕获速度</p> <p>值 描述</p> <p>0 无影响。</p> <p>1 启用正交编码器的速度捕获功能。</p>
4	RESMODE	R/W	0	<p>复位模式</p> <p>值 描述</p> <p>0 根据 QEIMAXPOS 寄存器 MAXPOS 域的定义，位置计数器的值达到最大值时计数器复位。</p> <p>1 当捕获到索引脉冲时位置计数器复位。</p>
3	CAPMODE	R/W	0	<p>捕获模式</p> <p>值 描述</p> <p>0 只有 PhA 的边沿被计数。</p> <p>1 PhA 和 PhB 的边沿都被计数，以提供两倍的位置精度，但是只有一半的范围。</p>

位/域	名称	类型	复位	描述
2	SIGMODE	R/W	0	信号模式  值 描述 0 PhA 和 PhB 信号作为正交相位信号运行。 1 PhA 和 PhB 信号作为时钟和方向信号运行。
1	SWAP	R/W	0	交换信号  值 描述 0 无影响。 1 将信号 PhA 和 PhB 进行交换。
0	ENABLE	R/W	0	启用 QEI  值 描述 0 无影响。 1 启用正交编码器模块。



## 寄存器 2: QEI 状态寄存器 (QEISTAT), 偏移量 0x004

该寄存器提供 QEI 模块操作相关的状态。

### QEI 状态寄存器 (QEISTAT)

QEI0 基址: 0x4002.C000

QEI1 基址: 0x4002.D000

偏移量 0x004

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														DIRECTION	ERROR
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:2	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
1	DIRECTION	RO	0	旋转方向 表示编码器旋转的方向。  值 描述 0 编码器正向旋转。 1 编码器反向旋转。
0	ERROR	RO	0	错误监测  值 描述 0 无错误。 1 在格雷码序列中检测到错误 (即, 两个信号同时改变)。

### 寄存器 3: QEI 位置寄存器 ( QEIPOS ) , 偏移量 0x008

该寄存器包含位置积分器的当前值。该数值根据 QEI 相位的输入状态更新，也可以通过写操作将其设置为一个指定值。

#### QEI 位置寄存器 (QEIPOS)

QEI0 基址: 0x4002.C000

QEI1 基址: 0x4002.D000

偏移量 0x008

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POSITION															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POSITION															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	POSITION	R/W	0x0000.0000	位置积分器的当前值 位置积分器的当前值。

## 寄存器 4: QEI 最大位置寄存器 ( QEIMAXPOS ) , 偏移量 0x00C

该寄存器包含位置积分器的最大值。当正向旋转时，如果位置寄存器的值递增后超过该寄存器的值，则位置寄存器复位为 0。当反向旋转时，如果位置寄存器从 0 开始递减，则位置寄存器复位为该寄存器中的值。

### QEI 最大位置寄存器 (QEIMAXPOS)

QEI0 基址: 0x4002.C000

QEI1 基址: 0x4002.D000

偏移量 0x00C

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MAXPOS															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MAXPOS															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	MAXPOS	R/W	0x0000.0000	位置积分器的最大值 位置积分器的最大值。

### 寄存器 5: QEI 定时器加载寄存器 ( QEILOAD ) , 偏移量 0x010

该寄存器包含速度定时器的加载值。该值在定时器到达 0 之后的时钟周期内装入定时器，因此，它应该比目标周期内的时钟数小 1。比如，如果希望每个定时周期内有 2000 个十进制时钟，则该寄存器中的值应为 1999。

#### QEI 定时器加载寄存器 (QEILOAD)

QEI0 基址: 0x4002.C000

QEI1 基址: 0x4002.D000

偏移量 0x010

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LOAD															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOAD															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	LOAD	R/W	0x0000.0000	速度定时器加载值 速度定时器的加载值。

## 寄存器 6: QEI 定时器寄存器 (QEITIME), 偏移量 0x014

该寄存器包含速度定时器的当前值。当 QEICTL 寄存器中的 VELEN 位清零时, 计数器不会递增。

### QEI 定时器寄存器 (QEITIME)

QEI0 基址: 0x4002.C000

QEI1 基址: 0x4002.D000

偏移量 0x014

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TIME															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIME															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

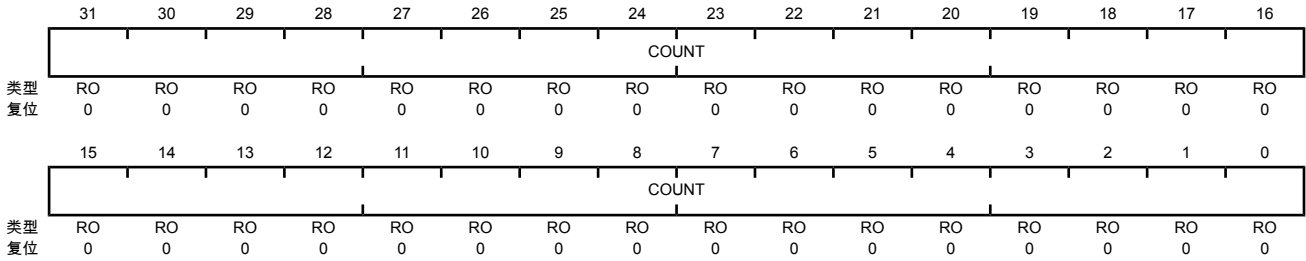
位/域	名称	类型	复位	描述
31:0	TIME	RO	0x0000.0000	速度定时器当前值 速度定时器的当前值。

### 寄存器 7: QEI 速度计数寄存器 (QEICOUNT), 偏移量 0x018

该寄存器包含在当前时间周期内, 正在计数的速率脉冲的个数。这是正在运行时的脉冲总数, 因此, 它所对应的时间周期的精度不可知 (即该寄存器的读操作未必与 QEITIME 寄存器返回的时间对应, 因为两次读操作之间有个小的时间间隔, 在这段时间内任一个值都可能发生改变)。QEISPEED 寄存器应该用来确定编码器的实际速度; 它只能用来提供信息。当 QEICTL 寄存器中的 VELEN 位清零时, 计数器不会递增。

#### QEI 速度计数寄存器 (QEICOUNT)

QEI0 基址: 0x4002.C000  
 QEI1 基址: 0x4002.D000  
 偏移量 0x018  
 类型 RO, 复位 0x0000.0000



位/域	名称	类型	复位	描述
31:0	COUNT	RO	0x0000.0000	速度脉冲计数 在速度定时器周期内, 编码器正在运行时的脉冲总数。

**寄存器 8: QEI 速度寄存器 ( QEISPEED ) , 偏移量 0x01C**

该寄存器包含最近测得的正交编码器的速度。它与上一个定时周期内计得的脉冲数相对应。当 QEICTL 的 VELEN 位清零时，该寄存器不会更新。

**QEI 速度寄存器 (QEISPEED)**

QEI0 基址: 0x4002.C000

QEI1 基址: 0x4002.D000

偏移量 0x01C

类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPEED															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPEED															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:0	SPEED	RO	0x0000.0000	速度 测得的正交编码器的速度，以一个周期时间内的脉冲数来计量。

## 寄存器 9: QEI 中断启用寄存器 ( QEIINTEN ) , 偏移量 0x020

该寄存器用于启用 QEI 模块的各个中断。如果将该寄存器中的对应位置位，则向控制器发出中断。

### QEI 中断启用寄存器 (QEIINTEN)

QEI0 基址: 0x4002.C000

QEI1 基址: 0x4002.D000

偏移量 0x020

类型 R/W, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												INTERROR	INTDIR	INTTIMER	INTINDEX	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品，保留位的值在读取-修改-写入操作过程中应当保持不变。
3	INTERROR	R/W	0	相位错误中断启用  值 描述 1 当 QEIRIS 寄存器中的 INTERROR 位置位时，中断被发送到中断控制器。 0 INTERROR 中断被抑制，中断不会发送到中断控制器。
2	INTDIR	R/W	0	方向变化中断启用  值 描述 1 当 QEIRIS 寄存器中的 INTDIR 位置位时，中断将被发送到中断控制器。 0 INTDIR 中断被抑制，中断不会被发送到中断控制器。
1	INTTIMER	R/W	0	定时器到期中断启用  值 描述 1 当 QEIRIS 寄存器中的 INTTIMER 位置位时，中断将被发送到中断控制器。 0 INTTIMER 中断被抑制，中断将不会发送到中断控制器。
0	INTINDEX	R/W	0	索引脉冲监测中断启用  值 描述 1 当 QEIRIS 寄存器中的 INTINDEX 位置位时，中断将被发送到中断控制器。 0 INTINDEX 中断被抑制，中断不会被发送到中断控制器。



## 寄存器 10: QEI 原始中断状态寄存器 (QEIRIS), 偏移量 0x024

该寄存器提供已发出中断的中断源的当前设置, 不管它们是否将中断提交到了控制器 (通过 QEIIINTEN 寄存器进行设置)。寄存器中的某位置位表示已发生相应的锁存事件, 清零表示相应的事件没有发生。

### QEI 原始中断状态寄存器 (QEIRIS)

QEI0 基址: 0x4002.C000  
QEI1 基址: 0x4002.D000  
偏移量 0x024  
类型 RO, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												INTERROR	INTDIR	INTTIMER	INTINDEX
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3	INTERROR	RO	0	相位错误监测  值 描述 1 检测到相位错误。 0 未产生中断。  在 QEIIISC 寄存器中的 INTERROR 位写入 1 即可将该位清零。
2	INTDIR	RO	0	方向变化监测  值 描述 1 旋转方向已经变化。 0 未产生中断。  在 QEIIISC 寄存器中的 INTDIR 位写入 1 即可将该位清零。
1	INTTIMER	RO	0	速度定时器到期  值 描述 1 表示速度定时器已经计满准备返回。 0 未产生中断。  在 QEIIISC 寄存器中的 INTTIMER 位写入 1 即可将该位清零。
0	INTINDEX	RO	0	索引脉冲监测  值 描述 1 表示已出现索引脉冲。 0 未产生中断。  在 QEIIISC 寄存器中的 INTINDEX 位写入 1 即可将该位清零。

### 寄存器 11: QEI 中断状态和清除寄存器 (QEISC), 偏移量 0x028

该寄存器提供已发送到控制器的中断源的当前设置。寄存器中的某位置位, 则表示已发生相应的锁存事件, 而且能够产生中断; 清零则表示相应事件没有发生或者不能产生中断。该寄存器是 R/W1C 类型, 即向某个位写 1 可将该位清零, 并清除对应的中断原因。

#### QEI 中断状态和清除寄存器 (QEISC)

QEI0 基址: 0x4002.C000  
 QEI1 基址: 0x4002.D000  
 偏移量 0x028  
 类型 R/W1C, 复位 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留												INTERROR	INTDIR	INTTIMER	INTINDEX	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/域	名称	类型	复位	描述
31:4	保留	RO	0x0000.000	软件不应该依赖保留位的值。为了兼容未来的产品, 保留位的值在读取-修改-写入操作过程中应当保持不变。
3	INTERROR	R/W1C	0	相位错误中断  值 描述 1 将寄存器 QEIRIS 和 QEINTEN 中的 INTERROR 位置位, 向中断控制器提供中断信号。 0 未发生中断, 或该中断被屏蔽。  对该位写入 1, 即可将其清零。将该位清零也把 QEIRIS 寄存器中的 INTERROR 位清零。
2	INTDIR	R/W1C	0	方向变化中断  值 描述 1 将寄存器 QEIRIS 和 QEINTEN 中的 INTDIR 位置位, 向中断控制器提供中断信号。 0 未发生中断, 或该中断被屏蔽。  对该位写入 1, 即可将其清零。将该位清零也把 QEIRIS 寄存器中的 INTDIR 位清零。
1	INTTIMER	R/W1C	0	速度定时器到期中断  值 描述 1 将寄存器 QEIRIS 和 QEINTEN 中的 INTTIMER 位置位, 向中断控制器提供中断信号。 0 未发生中断, 或该中断被屏蔽。  对该位写入 1, 即可将其清零。将该位清零也把 QEIRIS 寄存器中的 INTTIMER 位清零。

位/域	名称	类型	复位	描述
0	INTINDEX	R/W1C	0	<p>索引脉冲中断</p> <p>值 描述</p> <p>1 将寄存器 QEIRIS 和 QEINTEN 中的 INTINDEX 位置位，向中断控制器提供中断信号。</p> <p>0 未发生中断，或该中断被屏蔽。</p> <p>对该位写入 1，即可将其清零。将该位清零也把 QEIRIS 寄存器中的 INTINDEX 位清零。</p>



## 23 信号表

本章通过表格列举出每个管脚可配置成的信号。除下表列出的管脚外，其余可配置管脚在复位时的默认配置均为GPIO信号。通过GPIOAMSEL寄存器（请参考634页）可选择模拟模式。当需要使用某个GPIO管脚的复用数字功能时，应将GPIOAFSEL寄存器（请参考618页）的相应位置位，并通过GPIOCTL寄存器（请参考636页）中的PMCx位域从该GPIO的多个可选外设功能中选定一个，以提供更多的管脚多路复用选项。

**重要：** 除下表列出的管脚外，其余可配置管脚在复位时的默认配置均为GPIO信号。通过上电复位(POR)或外部复位RST，可以让管脚回到其默认状态。

表 23-1. 默认为复用功能的 GPIO 管脚

GPIO 管脚	默认状态	GPIOAFSEL 位	GPIOCTL 的 PMCx 位域
PA[1:0]	UART0	0	0x1
PA[5:2]	SSIO	0	0x1
PB[3:2]	I <sup>2</sup> C0	0	0x1
PA[3:0]	JTAG/SWD	1	0x3

表 23-2 在 1246 页 显示管脚到信号名称的映射，包括信号的功能特性。在每个管脚之后均完整列出其所有可供备选的模拟/数字功能。

表 23-3 在 1259 页 按信号名称的字母顺序列出信号。假如某个信号在多个管脚上均可配置，则列出该信号可对应的所有管脚。“复用管脚/赋值”一列是信号所对应的 GPIO 管脚以及应写入 GPIOCTL 寄存器 PMCx 位域的编码。

表 23-4 在 1271 页 按功能将信号分组，GPIO 管脚除外。假如某个信号在多个管脚上均可配置，则列出该信号可对应的所有管脚。

表 23-5 在 1280 页 列出了所有 GPIO 管脚及其模拟/数字复用功能。AINx 模拟信号都无法耐受 5 V，管脚需经过隔离电路连接其内部电路。这些信号的配置方式为：将 GPIO 数字启用 (GPIO DEN) 寄存器中相应的 DEN 位清零，并将 GPIO 模拟模式选择 (GPIOAMSEL) 寄存器中相应的 AMSEL 位置位。其余的模拟信号都可耐受 5 V，可以直接和其他电路连接 (C0-、C0+、C1-、C1+、C2-、C2+、USB0VBUS、USB0ID)。这些信号的配置方式为：将 GPIO 数字使能 (GPIO DEN) 寄存器中相应的 DEN 位清零。而在启用数字信号时，应将 GPIO 备用功能选择 (GPIOAFSEL) 以及 GPIO DEN 寄存器的相应位置位，并按照下表中所列出的数字编码对 GPIO 端口控制 (GPIOCTL) 寄存器的 PMCx 位域进行配置。表中的灰色单元格代表相应 GPIO 管脚的默认值。

表 23-6 在 1284 页 按可用于配置的管脚数量升序列出了所有信号。此表格宜用于根据特定功能来规划管脚的配置。应用笔记《AN01274：微控制器的®如何配置微控制器的管脚复用功能》提供了实施管脚复用的概述，解释系统设计人员如何定义管脚配置，并举例说明管脚配置的步骤。

**注意：** 所有数字输入端均为施密特触发。

## 23.1 按管脚编号分类的信号

表 23-2. 按管脚编号分类的信号

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
1	PD0	I/O	TTL	GPIO 端口 D 位 0。
	AIN15	I	模拟	模数转换器输入 15。
	I2C3SCL	I/O	OD	I <sup>2</sup> C 模块 3 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	M0PWM6	O	TTL	动作控制模块 0 PWM 6。此信号由模块 0 PWM 发生器 3 控制。
	M1PWM0	O	TTL	动作控制模块 1 PWM 0。此信号由模块 1 PWM 发生器 0 控制。
	SSI1Clk	I/O	TTL	SSI 模块 1 时钟
	SSI3Clk	I/O	TTL	SSI 模块 3 时钟
	WT2CCP0	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 0。
2	PD1	I/O	TTL	GPIO 端口 D 位 1。
	AIN14	I	模拟	模数转换器输入 14。
	I2C3SDA	I/O	OD	I <sup>2</sup> C 模块 3 数据。
	M0PWM7	O	TTL	动作控制模块 0 PWM 7。此信号由模块 0 PWM 发生器 3 控制。
	M1PWM1	O	TTL	动作控制模块 1 PWM 1。此信号由模块 1 PWM 发生器 0 控制。
	SSI1Fss	I/O	TTL	SSI 模块 1 帧信号
	SSI3Fss	I/O	TTL	SSI 模块 3 帧信号
	WT2CCP1	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 1。
3	PD2	I/O	TTL	GPIO 端口 D 位 2。
	AIN13	I	模拟	模数转换器输入 13。
	M0FAULT0	I	TTL	动作控制模块 0 PWM 故障 0。
	SSI1Rx	I	TTL	SSI 模块 1 接收
	SSI3Rx	I	TTL	SSI 模块 3 接收
	USB0EPEN	O	TTL	可选择性地以主机模式控制外部电源，向 USB 总线供电。
	WT3CCP0	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 0。
4	PD3	I/O	TTL	GPIO 端口 D 位 3。
	AIN12	I	模拟	模数转换器输入 12。
	IDX0	I	TTL	QEI 模块 0 索引。
	SSI1Tx	O	TTL	SSI 模块 1 发送
	SSI3Tx	O	TTL	SSI 模块 3 发送
	USB0PFLT	I	TTL	可选择性地用于主机模式的外部电源，指示外部电源的错误状态。
	WT3CCP1	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 1。
5	VDD	-	电源	I/O 和某些逻辑的电源正极。
6	GND	-	电源	逻辑及 I/O 管脚的接地参考。
7	VDDA	-	电源	模拟电路 (ADC、模拟比较器等) 的电源正端。VDDA 应与 VDD 分开，尽量避免 VDD 中携带的噪声影响模拟功能。不论系统如何实施，VDDA 管脚的电压必须符合表 25-2 在 1292 页中的规格。
8	VREFA+	-	模拟	参考电压用于指定 ADC 转换为最大值时的电压。此管脚与指定最小值的 VREFA- 结合使用。也就是说，若 AINn 输入的信号电压与 VREFA+ 电压相等，则转换结果为 4095。VREFA+ 的电压限定为表 25-25 在 1307 页中指定的范围。

表 23-2. 按管脚编号分类的信号 (续)

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
9	VREFA-	-	模拟	参考电压用于指定 ADC 转换为最小值时的输入电压。此管脚与指定最大值的 VREFA+ 结合使用。也就是说, 若 AINn 输入的信号电压与 VREFA- 电压相等, 则转换结果为 0。而若 AINn 输入的信号电压与 VREFA+ 电压相等, 则转换结果为 4095。VREFA- 的电压限定为表 25-25 在 1307 页中指定的范围。
10	GNDA	-	电源	模拟电路 (ADC、模拟比较器等) 的接地参考。GNDA 应与 GND 分开, 尽量避免 VDD 中携带的噪声影响模拟功能。
11	PP2	I/O	TTL	GPIO 端口 P 位 2。
	M0PWM2	O	TTL	动作控制模块 0 PWM 2。此信号由模块 0 PWM 发生器 1 控制。
	T5CCP0	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 0。
12	PE3	I/O	TTL	GPIO 端口 E 位 3。
	AIN0	I	模拟	模数转换器输入 0。
13	PE2	I/O	TTL	GPIO 端口 E 位 2。
	AIN1	I	模拟	模数转换器输入 1。
14	PE1	I/O	TTL	GPIO 端口 E 位 1。
	AIN2	I	模拟	模数转换器输入 2。
	U7Tx	O	TTL	UART 模块 7 发送信号。
15	PE0	I/O	TTL	GPIO 端口 E 位 0。
	AIN3	I	模拟	模数转换器输入 3。
	U7Rx	I	TTL	UART 模块 7 接收信号。
16	PK0	I/O	TTL	GPIO 端口 K 位 0。
	AIN16	I	模拟	模数转换器输入 16。
	M1FAULT0	I	TTL	动作控制模块 1 PWM 故障 0。
	SSI3Clk	I/O	TTL	SSI 模块 3 时钟
17	PK1	I/O	TTL	GPIO 端口 K 位 1。
	AIN17	I	模拟	模数转换器输入 17。
	M1FAULT1	I	TTL	动作控制模块 1 PWM 故障 1。
	SSI3Fss	I/O	TTL	SSI 模块 3 帧信号
18	PK2	I/O	TTL	GPIO 端口 K 位 2。
	AIN18	I	模拟	模数转换器输入 18。
	M1FAULT2	I	TTL	动作控制模块 1 PWM 故障 2。
	SSI3Rx	I	TTL	SSI 模块 3 接收
19	PK3	I/O	TTL	GPIO 端口 K 位 3。
	AIN19	I	模拟	模数转换器输入 19。
	M1FAULT3	I	TTL	动作控制模块 1 PWM 故障 3。
	SSI3Tx	O	TTL	SSI 模块 3 发送
20	PN2	I/O	TTL	GPIO 端口 N 位 2。
	M0PWM6	O	TTL	动作控制模块 0 PWM 6。此信号由模块 0 PWM 发生器 3 控制。
	WT2CCP0	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 0。
21	PH7	I/O	TTL	GPIO 端口 H 位 7。
	M0PWM7	O	TTL	动作控制模块 0 PWM 7。此信号由模块 0 PWM 发生器 3 控制。
	SSI2Tx	O	TTL	SSI 模块 2 发送
	WT4CCP1	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 1。

表 23-2. 按管脚编号分类的信号 (续)

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
22	PH6	I/O	TTL	GPIO 端口 H 位 6。
	M0PWM6	O	TTL	动作控制模块 0 PWM 6。此信号由模块 0 PWM 发生器 3 控制。
	SSI2Rx	I	TTL	SSI 模块 2 接收
	WT4CCP0	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 0。
23	PH5	I/O	TTL	GPIO 端口 H 位 5。
	M0PWM5	O	TTL	动作控制模块 0 PWM 5。此信号由模块 0 PWM 发生器 2 控制。
	PhB0	I	TTL	QE1 模块 0 相位 B。
	SSI2Fss	I/O	TTL	SSI 模块 2 帧信号
	WT3CCP1	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 1。
24	VDD	-	电源	I/O 和某些逻辑的电源正极。
25	GND	-	电源	逻辑及 I/O 管脚的接地参考。
26	PH4	I/O	TTL	GPIO 端口 H 位 4。
	M0PWM4	O	TTL	动作控制模块 0 PWM 4。此信号由模块 0 PWM 发生器 2 控制。
	PhA0	I	TTL	QE1 模块 0 相位 A。
	SSI2Clk	I/O	TTL	SSI 模块 2 时钟
	WT3CCP0	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 0。
27	PH3	I/O	TTL	GPIO 端口 H 位 3。
	M0FAULT3	I	TTL	动作控制模块 0 PWM 故障 3。
	M0PWM3	O	TTL	动作控制模块 0 PWM 3。此信号由模块 0 PWM 发生器 1 控制。
	SSI3Tx	O	TTL	SSI 模块 3 发送
	WT5CCP1	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 1。
28	PH2	I/O	TTL	GPIO 端口 H 位 2。
	M0FAULT2	I	TTL	动作控制模块 0 PWM 故障 2。
	M0PWM2	O	TTL	动作控制模块 0 PWM 2。此信号由模块 0 PWM 发生器 1 控制。
	SSI3Rx	I	TTL	SSI 模块 3 接收
	WT5CCP0	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 0。
29	VDD	-	电源	I/O 和某些逻辑的电源正极。
30	GND	-	电源	逻辑及 I/O 管脚的接地参考。
31	PH1	I/O	TTL	GPIO 端口 H 位 1。
	IDX0	I	TTL	QE1 模块 0 索引。
	M0FAULT1	I	TTL	动作控制模块 0 PWM 故障 1。
	M0PWM1	O	TTL	动作控制模块 0 PWM 1。此信号由模块 0 PWM 发生器 0 控制。
	SSI3Fss	I/O	TTL	SSI 模块 3 帧信号
	WT2CCP1	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 1。
32	PH0	I/O	TTL	GPIO 端口 H 位 0。
	M0FAULT0	I	TTL	动作控制模块 0 PWM 故障 0。
	M0PWM0	O	TTL	动作控制模块 0 PWM 0。此信号由模块 0 PWM 发生器 0 控制。
	SSI3Clk	I/O	TTL	SSI 模块 3 时钟
	WT2CCP0	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 0。



表 23-2. 按管脚编号分类的信号 (续)

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
33	PC7	I/O	TTL	GPIO 端口 C 位 7。
	C0-	I	模拟	模拟比较器 0 负极输入。
	U3Tx	O	TTL	UART 模块 3 发送信号。
	USB0PFLT	I	TTL	可选择性地用于主机模式的外部电源, 指示外部电源的错误状态。
	WT1CCP1	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 1。
34	PC6	I/O	TTL	GPIO 端口 C 位 6。
	C0+	I	模拟	模拟比较器 0 正极输入。
	PhB1	I	TTL	QE1 模块 1 相位 B。
	U3Rx	I	TTL	UART 模块 3 接收信号。
	USB0EPEN	O	TTL	可选择性地以主机模式控制外部电源, 向 USB 总线供电。
	WT1CCP0	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 0。
35	PC5	I/O	TTL	GPIO 端口 C 位 5。
	C1+	I	模拟	模拟比较器 1 正极输入。
	M0PWM7	O	TTL	动作控制模块 0 PWM 7。此信号由模块 0 PWM 发生器 3 控制。
	PhA1	I	TTL	QE1 模块 1 相位 A。
	U1CTS	I	TTL	UART 模块 1 CTS (Clear to Send, 允许发送) 调制解调器流控输入信号。
	U1Tx	O	TTL	UART 模块 1 发送信号。
	U4Tx	O	TTL	UART 模块 4 发送信号。
	WT0CCP1	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 1。
36	PC4	I/O	TTL	GPIO 端口 C 位 4。
	C1-	I	模拟	模拟比较器 1 负极输入。
	IDX1	I	TTL	QE1 模块 1 索引。
	M0PWM6	O	TTL	动作控制模块 0 PWM 6。此信号由模块 0 PWM 发生器 3 控制。
	U1RTS	O	TTL	UART 模块 1 RTS (Request to Send, 请求发送) 调制解调器流控输出线。
	U1Rx	I	TTL	UART 模块 1 接收信号。
	U4Rx	I	TTL	UART 模块 4 接收信号。
	WT0CCP0	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 0。
37	PA0	I/O	TTL	GPIO 端口 A 位 0。
	CAN1Rx	I	TTL	CAN 模块 1 接收信号。
	U0Rx	I	TTL	UART 模块 0 接收信号。
38	PA1	I/O	TTL	GPIO 端口 A 位 1。
	CAN1Tx	O	TTL	CAN 模块 1 发送信号。
	U0Tx	O	TTL	UART 模块 0 发送信号。
39	PA2	I/O	TTL	GPIO 端口 A 位 2。
	SSI0Clk	I/O	TTL	SSI 模块 0 时钟
40	PA3	I/O	TTL	GPIO 端口 A 位 3。
	SSI0Fss	I/O	TTL	SSI 模块 0 帧信号
41	PA4	I/O	TTL	GPIO 端口 A 位 4。
	SSI0Rx	I	TTL	SSI 模块 0 接收

表 23-2. 按管脚编号分类的信号 (续)

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
42	PA5	I/O	TTL	GPIO 端口 A 位 5。
	SSI0Tx	O	TTL	SSI模块0发送
43	VDD	-	电源	I/O 和某些逻辑的电源正极。
44	GND	-	电源	逻辑及 I/O 管脚的接地参考。
45	PA6	I/O	TTL	GPIO 端口 A 位 6。
	I2C1SCL	I/O	OD	I <sup>2</sup> C 模块 1 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	M1PWM2	O	TTL	动作控制模块 1 PWM 2。此信号由模块 1 PWM 发生器 1 控制。
46	PA7	I/O	TTL	GPIO 端口 A 位 7。
	I2C1SDA	I/O	OD	I <sup>2</sup> C 模块 1 数据。
	M1PWM3	O	TTL	动作控制模块 1 PWM 3。此信号由模块 1 PWM 发生器 1 控制。
47	PG7	I/O	TTL	GPIO 端口 G 位 7。
	I2C5SDA	I/O	OD	I <sup>2</sup> C 模块 5 数据。
	IDX1	I	TTL	QE1 模块 1 索引。
	M0PWM7	O	TTL	动作控制模块 0 PWM 7。此信号由模块 0 PWM 发生器 3 控制。
	WT1CCP1	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 1。
48	PG6	I/O	TTL	GPIO 端口 G 位 6。
	I2C5SCL	I/O	OD	I <sup>2</sup> C 模块 5 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	M0PWM6	O	TTL	动作控制模块 0 PWM 6。此信号由模块 0 PWM 发生器 3 控制。
	WT1CCP0	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 0。
49	VDDC	-	电源	为主要的逻辑部分 (包括处理器内核以及大部分片上外设) 供电的电源正端。该管脚上的电压为 1.2 V, 由片上 LDO 提供。按照表 25-8 在 1297页 中的规定, VDDC 管脚只应相互连接以及外部电容。
50	PG5	I/O	TTL	GPIO 端口 G 位 5。
	I2C1SDA	I/O	OD	I <sup>2</sup> C 模块 1 数据。
	IDX1	I	TTL	QE1 模块 1 索引。
	M0PWM5	O	TTL	动作控制模块 0 PWM 5。此信号由模块 0 PWM 发生器 2 控制。
	M1PWM3	O	TTL	动作控制模块 1 PWM 3。此信号由模块 1 PWM 发生器 1 控制。
	U2Tx	O	TTL	UART 模块 2 发送信号。
	USB0PFLT	I	TTL	可选择性地用于主机模式的外部电源, 指示外部电源的错误状态。
WT0CCP1	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 1。	
51	PG4	I/O	TTL	GPIO 端口 G 位 4。
	I2C1SCL	I/O	OD	I <sup>2</sup> C 模块 1 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	M0PWM4	O	TTL	动作控制模块 0 PWM 4。此信号由模块 0 PWM 发生器 2 控制。
	M1PWM2	O	TTL	动作控制模块 1 PWM 2。此信号由模块 1 PWM 发生器 1 控制。
	PhB1	I	TTL	QE1 模块 1 相位 B。
	U2Rx	I	TTL	UART 模块 2 接收信号。
	USB0EPEN	O	TTL	可选择性地以主机模式控制外部电源, 向 USB 总线供电。
	WT0CCP0	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 0。

表 23-2. 按管脚编号分类的信号 ( 续 )

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
52	PG3	I/O	TTL	GPIO 端口 G 位 3。
	I2C4SDA	I/O	OD	I <sup>2</sup> C 模块 4 数据。
	M0FAULT2	I	TTL	动作控制模块 0 PWM 故障 2。
	M1PWM1	O	TTL	动作控制模块 1 PWM 1。此信号由模块 1 PWM 发生器 0 控制。
	PhA1	I	TTL	QE1 模块 1 相位 A。
	T5CCP1	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 1。
53	PG2	I/O	TTL	GPIO 端口 G 位 2。
	I2C4SCL	I/O	OD	I <sup>2</sup> C 模块 4 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	M0FAULT1	I	TTL	动作控制模块 0 PWM 故障 1。
	M1PWM0	O	TTL	动作控制模块 1 PWM 0。此信号由模块 1 PWM 发生器 0 控制。
	T5CCP0	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 0。
54	PG1	I/O	TTL	GPIO 端口 G 位 1。
	I2C3SDA	I/O	OD	I <sup>2</sup> C 模块 3 数据。
	M1FAULT2	I	TTL	动作控制模块 1 PWM 故障 2。
	PhB1	I	TTL	QE1 模块 1 相位 B。
	T4CCP1	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 1。
55	PG0	I/O	TTL	GPIO 端口 G 位 0。
	I2C3SCL	I/O	OD	I <sup>2</sup> C 模块 3 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	M1FAULT1	I	TTL	动作控制模块 1 PWM 故障 1。
	PhA1	I	TTL	QE1 模块 1 相位 A。
	T4CCP0	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 0。
56	VDD	-	电源	I/O 和某些逻辑的电源正极。
57	GND	-	电源	逻辑及 I/O 管脚的接地参考。
58	PF7	I/O	TTL	GPIO 端口 F 位 7。
	I2C2SDA	I/O	OD	I <sup>2</sup> C 模块 2 数据。
	M1FAULT0	I	TTL	动作控制模块 1 PWM 故障 0。
	T3CCP1	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 1。
59	PF6	I/O	TTL	GPIO 端口 F 位 6。
	I2C2SCL	I/O	OD	I <sup>2</sup> C 模块 2 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	T3CCP0	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 0。
60	PF5	I/O	TTL	GPIO 端口 F 位 5。
	M0FAULT3	I	TTL	动作控制模块 0 PWM 故障 3。
	T2CCP1	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 1。
	USB0PFLT	I	TTL	可选择性地用于主机模式的外部电源, 指示外部电源的错误状态。

表 23-2. 按管脚编号分类的信号 ( 续 )

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
61	PF4	I/O	TTL	GPIO 端口 F 位 4。
	IDX0	I	TTL	QEI 模块 0 索引。
	M0FAULT2	I	TTL	动作控制模块 0 PWM 故障 2。
	M1FAULT0	I	TTL	动作控制模块 1 PWM 故障 0。
	T2CCP0	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 0。
	TRD3	O	TTL	跟踪数据信号 3。
	U1DTR	O	TTL	UART 模块 1 DTR ( Data Terminal Ready, 数据终端就绪 ) 调制解调器状态输入信号。
	USB0EPEN	O	TTL	可选择性地以主机模式控制外部电源, 向 USB 总线供电。
62	PF0	I/O	TTL	GPIO 端口 F 位 0。
	C0o	O	TTL	模拟比较器 0 输出。
	CAN0Rx	I	TTL	CAN 模块 0 接收信号。
	M1PWM4	O	TTL	动作控制模块 1 PWM 4。此信号由模块 1 PWM 发生器 2 控制。
	NMI	I	TTL	不可屏蔽的中断。
	PhA0	I	TTL	QEI 模块 0 相位 A。
	SSI1Rx	I	TTL	SSI 模块 1 接收
	T0CCP0	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 0。
	TRD2	O	TTL	跟踪数据信号 2。
U1RTS	O	TTL	UART 模块 1 RTS ( Request to Send, 请求发送 ) 调制解调器流控输出线。	
63	PF1	I/O	TTL	GPIO 端口 F 位 1。
	C1o	O	TTL	模拟比较器 1 输出。
	M1PWM5	O	TTL	动作控制模块 1 PWM 5。此信号由模块 1 PWM 发生器 2 控制。
	PhB0	I	TTL	QEI 模块 0 相位 B。
	SSI1Tx	O	TTL	SSI 模块 1 发送
	T0CCP1	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 1。
	TRD1	O	TTL	跟踪数据信号 1。
	U1CTS	I	TTL	UART 模块 1 CTS ( Clear to Send, 允许发送 ) 调制解调器流控输入信号。
64	PF2	I/O	TTL	GPIO 端口 F 位 2。
	C2o	O	TTL	模拟比较器 2 输出。
	M0FAULT0	I	TTL	动作控制模块 0 PWM 故障 0。
	M1PWM6	O	TTL	动作控制模块 1 PWM 6。此信号由模块 1 PWM 发生器 3 控制。
	SSI1Clk	I/O	TTL	SSI 模块 1 时钟
	T1CCP0	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 0。
	TRD0	O	TTL	跟踪数据信号 0。
	U1DCD	I	TTL	UART 模块 1 DCD ( Data Carrier Detect, 数据载波检测 ) 调制解调器状态输入信号。

表 23-2. 按管脚编号分类的信号 (续)

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
65	PF3	I/O	TTL	GPIO 端口 F 位 3。
	CAN0Tx	O	TTL	CAN模块0发送信号。
	M0FAULT1	I	TTL	动作控制模块 0 PWM 故障 1。
	M1PWM7	O	TTL	动作控制模块 1 PWM 7。此信号由模块 1 PWM 发生器 3 控制。
	SSI1Fss	I/O	TTL	SSI 模块 1 帧信号
	T1CCP1	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 1。
	TRCLK	O	TTL	跟踪时钟信号。
	U1DSR	I	TTL	UART 模块 1 DSR ( Data Set Ready , 数据设备就绪 ) 调制解调器输出控制线。
66	VDD	-	电源	I/O 和某些逻辑的电源正极。
67	GND	-	电源	逻辑及 I/O 管脚的接地参考。
68	PN7	I/O	TTL	GPIO 端口 N 位 7。
	M1PWM7	O	TTL	动作控制模块 1 PWM 7。此信号由模块 1 PWM 发生器 3 控制。
	WT4CCP1	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 1。
69	PN6	I/O	TTL	GPIO 端口 N 位 6。
	M1PWM6	O	TTL	动作控制模块 1 PWM 6。此信号由模块 1 PWM 发生器 3 控制。
	WT4CCP0	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 0。
70	PN5	I/O	TTL	GPIO 端口 N 位 5。
	M1PWM5	O	TTL	动作控制模块 1 PWM 5。此信号由模块 1 PWM 发生器 2 控制。
	WT3CCP1	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 1。
71	PN4	I/O	TTL	GPIO 端口 N 位 4。
	M1PWM4	O	TTL	动作控制模块 1 PWM 4。此信号由模块 1 PWM 发生器 2 控制。
	WT3CCP0	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 0。
72	WAKE	I	TTL	当有效时外部输入将处理器从休眠模式中唤醒。
73	HTB	O	TTL	该输出指示处理器是否处于休眠模式。
74	XOSC0	I	模拟	休眠模块晶体振荡器输入或外部时钟参考输入。请注意休眠模块 RTC 采用 32.768-kHz 晶体或 32.768-kHz 振荡器。
75	GNDX	-	电源	休眠振荡器的接地。使用晶振时钟源时, 该管脚应仅连接到晶振负载电容, 以提高振荡器对系统噪声的免疫能力。使用外部振荡器时, 该管脚应连接到 GND。
76	XOSC1	O	模拟	休眠模块晶体振荡器输出。使用单端时钟源时不连接。
77	VBAT	-	电源	休眠模块的电源供应源它通常连接到电池的正极端并用作备用电池/休眠模块电源供应器的电源。
78	VDD	-	电源	I/O 和某些逻辑的电源正极。
79	GND	-	电源	逻辑及 I/O 管脚的接地参考。
80	PN1	I/O	TTL	GPIO 端口 N 位 1。
	CAN0Tx	O	TTL	CAN模块0发送信号。
81	PN0	I/O	TTL	GPIO 端口 N 位 0。
	CAN0Rx	I	TTL	CAN模块0接收信号。
82	PM7	I/O	TTL	GPIO 端口 M 位 7。
	M0PWM5	O	TTL	动作控制模块 0 PWM 5。此信号由模块 0 PWM 发生器 2 控制。
	WT0CCP1	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 1。

表 23-2. 按管脚编号分类的信号 ( 续 )

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
83	PM6	I/O	TTL	GPIO 端口 M 位 6。
	M0PWM4	O	TTL	动作控制模块 0 PWM 4。此信号由模块 0 PWM 发生器 2 控制。
	WT0CCP0	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 0。
84	PM5	I/O	TTL	GPIO 端口 M 位 5。
85	PM4	I/O	TTL	GPIO 端口 M 位 4。
86	PM3	I/O	TTL	GPIO 端口 M 位 3。
	T5CCP1	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 1。
	WT5CCP1	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 1。
87	PM2	I/O	TTL	GPIO 端口 M 位 2。
	T5CCP0	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 0。
	WT5CCP0	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 0。
88	PM1	I/O	TTL	GPIO 端口 M 位 1。
	T4CCP1	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 1。
	WT4CCP1	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 1。
89	PM0	I/O	TTL	GPIO 端口 M 位 0。
	T4CCP0	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 0。
	WT4CCP0	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 0。
90	$\overline{RST}$	I	TTL	系统复位输入。
91	GND	-	电源	逻辑及 I/O 管脚的接地参考。
92	OSC0	I	模拟	晶体主振荡器输入或外部时钟参考输入。
93	OSC1	O	模拟	晶体主振荡器输出。使用单端时钟源时不连接。
94	VDD	-	电源	I/O 和某些逻辑的电源正极。
95	PL7	I/O	TTL	GPIO 端口 L 位 7。
	T3CCP1	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 1。
	USB0DM	I/O	模拟	USB0 的双向差分数据管脚 ( USB 规范中的 D- )。
	WT3CCP1	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 1。
96	PL6	I/O	TTL	GPIO 端口 L 位 6。
	T3CCP0	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 0。
	USB0DP	I/O	模拟	USB0 的双向差分数据管脚 ( USB 规范中的 D+ )。
	WT3CCP0	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 0。
97	PB0	I/O	TTL	GPIO 端口 B 位 0。该管脚不能承受 5V 最高电压。
	T2CCP0	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 0。
	U1Rx	I	TTL	UART 模块 1 接收信号。
	USB0ID	I	模拟	此信号用于检测 USB ID 信号的状态。此时 USB PHY 将在内部启用一个上拉电阻，通过外部元件 ( USB 连接器 ) 检测 USB 控制器的初始状态 ( 即电缆的 A 侧设置下拉电阻，B 侧设置上拉电阻 )。
98	PB1	I/O	TTL	GPIO 端口 B 位 1。该管脚不能承受 5V 最高电压。
	T2CCP1	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 1。
	U1Tx	O	TTL	UART 模块 1 发送信号。
	USB0VBUS	I/O	模拟	此信号用于会话请求协议。USB PHY 可通过此信号检测 VBUS 的电平，并在 VBUS 脉冲期间短时下拉。

表 23-2. 按管脚编号分类的信号 (续)

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
99	PB2	I/O	TTL	GPIO 端口 B 位 2。
	I2C0SCL	I/O	OD	I <sup>2</sup> C 模块 0 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	T3CCP0	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 0。
100	PB3	I/O	TTL	GPIO 端口 B 位 3。
	I2C0SDA	I/O	OD	I <sup>2</sup> C 模块 0 数据。
	T3CCP1	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 1。
101	VDD	-	电源	I/O 和某些逻辑的电源正极。
102	GND	-	电源	逻辑及 I/O 管脚的接地参考。
103	PL5	I/O	TTL	GPIO 端口 L 位 5。
	T2CCP1	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 1。
	WT2CCP1	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 1。
104	PL4	I/O	TTL	GPIO 端口 L 位 4。
	T2CCP0	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 0。
	WT2CCP0	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 0。
105	PL3	I/O	TTL	GPIO 端口 L 位 3。
	T1CCP1	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 1。
	WT1CCP1	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 1。
106	PL2	I/O	TTL	GPIO 端口 L 位 2。
	T1CCP0	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 0。
	WT1CCP0	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 0。
107	PL1	I/O	TTL	GPIO 端口 L 位 1。
	T0CCP1	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 1。
	WT0CCP1	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 1。
108	PL0	I/O	TTL	GPIO 端口 L 位 0。
	T0CCP0	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 0。
	WT0CCP0	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 0。
109	PK7	I/O	TTL	GPIO 端口 K 位 7。
	M0FAULT3	I	TTL	动作控制模块 0 PWM 故障 3。
	WT1CCP1	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 1。
110	PK6	I/O	TTL	GPIO 端口 K 位 6。
	C2o	O	TTL	模拟比较器 2 输出。
	M0FAULT2	I	TTL	动作控制模块 0 PWM 故障 2。
	WT1CCP0	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 0。
111	PK5	I/O	TTL	GPIO 端口 K 位 5。
	C1o	O	TTL	模拟比较器 1 输出。
	M0FAULT1	I	TTL	动作控制模块 0 PWM 故障 1。
	U7Tx	O	TTL	UART 模块 7 发送信号。

表 23-2. 按管脚编号分类的信号 ( 续 )

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
112	PK4	I/O	TTL	GPIO 端口 K 位 4。
	C0o	O	TTL	模拟比较器 0 输出。
	M0FAULT0	I	TTL	动作控制模块 0 PWM 故障 0。
	RTCCLK	O	TTL	休眠模块缓冲版的 32.768-kHz 时钟。此部分处于休眠模式时，该信号并非输出信号。
	U7Rx	I	TTL	UART 模块 7 接收信号。
113	VDD	-	电源	I/O 和某些逻辑的电源正极。
114	GND	-	电源	逻辑及 I/O 管脚的接地参考。
115	PC3	I/O	TTL	GPIO 端口 C 位 3。
	SWO	O	TTL	JTAG TDO 及 SWO 信号。
	T5CCP1	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 1。
	TDO	O	TTL	JTAG TDO 及 SWO 信号。
116	PC2	I/O	TTL	GPIO 端口 C 位 2。
	T5CCP0	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 0。
	TDI	I	TTL	JTAG TDI 信号。
117	PC1	I/O	TTL	GPIO 端口 C 位 1。
	SWDIO	I/O	TTL	JTAG TMS 及 SWDIO 信号。
	T4CCP1	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 1。
	TMS	I	TTL	JTAG TMS 及 SWDIO 信号。
118	PC0	I/O	TTL	GPIO 端口 C 位 0。
	SWCLK	I	TTL	JTAG/SWD CLK 信号。
	T4CCP0	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 0。
	TCK	I	TTL	JTAG/SWD CLK 信号。
119	PN3	I/O	TTL	GPIO 端口 N 位 3。
	M0PWM7	O	TTL	动作控制模块 0 PWM 7。此信号由模块 0 PWM 发生器 3 控制。
	WT2CCP1	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 1。
120	PJ0	I/O	TTL	GPIO 端口 J 位 0。
	T1CCP0	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 0。
	U4Rx	I	TTL	UART 模块 4 接收信号。
121	PJ1	I/O	TTL	GPIO 端口 J 位 1。
	T1CCP1	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 1。
	U4Tx	O	TTL	UART 模块 4 发送信号。
122	PJ2	I/O	TTL	GPIO 端口 J 位 2。
	IDX0	I	TTL	QE1 模块 0 索引。
	T2CCP0	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 0。
	U5Rx	I	TTL	UART 模块 5 接收信号。
123	PJ3	I/O	TTL	GPIO 端口 J 位 3。
	T2CCP1	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 1。
	U5Tx	O	TTL	UART 模块 5 发送信号。
124	VDD	-	电源	I/O 和某些逻辑的电源正极。
125	GND	-	电源	逻辑及 I/O 管脚的接地参考。



表 23-2. 按管脚编号分类的信号 (续)

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
126	VDDC	-	电源	为主要的逻辑部分 (包括处理器内核以及大部分片上外设) 供电的电源正端。该管脚上的电压为 1.2 V, 由片上 LDO 提供。按照表 25-8 在 1297 页中的规定, VDDC 管脚只应相互连接以及外部电容。
127	PJ4	I/O	TTL	GPIO 端口 J 位 4。
	C2+	I	模拟	模拟比较器 2 正极输入。
	T3CCP0	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 0。
	U6Rx	I	TTL	UART 模块 6 接收信号。
128	PJ5	I/O	TTL	GPIO 端口 J 位 5。
	C2-	I	模拟	模拟比较器 2 负极输入。
	T3CCP1	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 1。
	U6Tx	O	TTL	UART 模块 6 发送信号。
129	PJ6	I/O	TTL	GPIO 端口 J 位 6。
130	PJ7	I/O	TTL	GPIO 端口 J 位 7。
131	PP0	I/O	TTL	GPIO 端口 P 位 0。
	AIN23	I	模拟	模数转换器输入 23。
	M0PWM0	O	TTL	动作控制模块 0 PWM 0。此信号由模块 0 PWM 发生器 0 控制。
	T4CCP0	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 0。
132	PP1	I/O	TTL	GPIO 端口 P 位 1。
	AIN22	I	模拟	模数转换器输入 22。
	M0PWM1	O	TTL	动作控制模块 0 PWM 1。此信号由模块 0 PWM 发生器 0 控制。
	T4CCP1	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 1。
133	PE6	I/O	TTL	GPIO 端口 E 位 6。
	AIN21	I	模拟	模数转换器输入 21。
	CAN1Rx	I	TTL	CAN 模块 1 接收信号。
134	PE7	I/O	TTL	GPIO 端口 E 位 7。
	AIN20	I	模拟	模数转换器输入 20。
	CAN1Tx	O	TTL	CAN 模块 1 发送信号。
	U1RI	I	TTL	UART 模块 1 RI (Ring Indicator, 振铃指示) 调制解调器状态输入信号。
135	PB5	I/O	TTL	GPIO 端口 B 位 5。
	AIN11	I	模拟	模数转换器输入 11。
	CAN0Tx	O	TTL	CAN 模块 0 发送信号。
	M0PWM3	O	TTL	动作控制模块 0 PWM 3。此信号由模块 0 PWM 发生器 1 控制。
	SSI2Fss	I/O	TTL	SSI 模块 2 帧信号
	T1CCP1	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 1。
136	PB4	I/O	TTL	GPIO 端口 B 位 4。
	AIN10	I	模拟	模数转换器输入 10。
	CAN0Rx	I	TTL	CAN 模块 0 接收信号。
	M0PWM2	O	TTL	动作控制模块 0 PWM 2。此信号由模块 0 PWM 发生器 1 控制。
	SSI2Clk	I/O	TTL	SSI 模块 2 时钟
	T1CCP0	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 0。
137	VDD	-	电源	I/O 和某些逻辑的电源正极。
138	GND	-	电源	逻辑及 I/O 管脚的接地参考。

表 23-2. 按管脚编号分类的信号 ( 续 )

管脚编号	管脚名称	管脚类型	缓冲区类型 <sup>a</sup>	描述
139	PE4	I/O	TTL	GPIO 端口 E 位 4。
	AIN9	I	模拟	模数转换器输入 9。
	CAN0Rx	I	TTL	CAN 模块 0 接收信号。
	I2C2SCL	I/O	OD	I <sup>2</sup> C 模块 2 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	M0PWM4	O	TTL	动作控制模块 0 PWM 4。此信号由模块 0 PWM 发生器 2 控制。
	M1PWM2	O	TTL	动作控制模块 1 PWM 2。此信号由模块 1 PWM 发生器 1 控制。
	U5Rx	I	TTL	UART 模块 5 接收信号。
140	PE5	I/O	TTL	GPIO 端口 E 位 5。
	AIN8	I	模拟	模数转换器输入 8。
	CAN0Tx	O	TTL	CAN 模块 0 发送信号。
	I2C2SDA	I/O	OD	I <sup>2</sup> C 模块 2 数据。
	M0PWM5	O	TTL	动作控制模块 0 PWM 5。此信号由模块 0 PWM 发生器 2 控制。
	M1PWM3	O	TTL	动作控制模块 1 PWM 3。此信号由模块 1 PWM 发生器 1 控制。
	U5Tx	O	TTL	UART 模块 5 发送信号。
141	PD4	I/O	TTL	GPIO 端口 D 位 4。
	AIN7	I	模拟	模数转换器输入 7。
	U6Rx	I	TTL	UART 模块 6 接收信号。
	WT4CCP0	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 0。
142	PD5	I/O	TTL	GPIO 端口 D 位 5。
	AIN6	I	模拟	模数转换器输入 6。
	U6Tx	O	TTL	UART 模块 6 发送信号。
	WT4CCP1	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 1。
143	PD6	I/O	TTL	GPIO 端口 D 位 6。
	AIN5	I	模拟	模数转换器输入 5。
	M0FAULT0	I	TTL	动作控制模块 0 PWM 故障 0。
	PhA0	I	TTL	QEI 模块 0 相位 A。
	U2Rx	I	TTL	UART 模块 2 接收信号。
	WT5CCP0	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 0。
144	PD7	I/O	TTL	GPIO 端口 D 位 7。
	AIN4	I	模拟	模数转换器输入 4。
	M0FAULT1	I	TTL	动作控制模块 0 PWM 故障 1。
	NMI	I	TTL	不可屏蔽的中断。
	PhB0	I	TTL	QEI 模块 0 相位 B。
	U2Tx	O	TTL	UART 模块 2 发送信号。
	WT5CCP1	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 1。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 23.2 按信号名称分类的信号

表 23-3. 按信号名称分类的信号

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
AIN0	12	PE3	I	模拟	模数转换器输入 0。
AIN1	13	PE2	I	模拟	模数转换器输入 1。
AIN2	14	PE1	I	模拟	模数转换器输入 2。
AIN3	15	PE0	I	模拟	模数转换器输入 3。
AIN4	144	PD7	I	模拟	模数转换器输入 4。
AIN5	143	PD6	I	模拟	模数转换器输入 5。
AIN6	142	PD5	I	模拟	模数转换器输入 6。
AIN7	141	PD4	I	模拟	模数转换器输入 7。
AIN8	140	PE5	I	模拟	模数转换器输入 8。
AIN9	139	PE4	I	模拟	模数转换器输入 9。
AIN10	136	PB4	I	模拟	模数转换器输入 10。
AIN11	135	PB5	I	模拟	模数转换器输入 11。
AIN12	4	PD3	I	模拟	模数转换器输入 12。
AIN13	3	PD2	I	模拟	模数转换器输入 13。
AIN14	2	PD1	I	模拟	模数转换器输入 14。
AIN15	1	PD0	I	模拟	模数转换器输入 15。
AIN16	16	PK0	I	模拟	模数转换器输入 16。
AIN17	17	PK1	I	模拟	模数转换器输入 17。
AIN18	18	PK2	I	模拟	模数转换器输入 18。
AIN19	19	PK3	I	模拟	模数转换器输入 19。
AIN20	134	PE7	I	模拟	模数转换器输入 20。
AIN21	133	PE6	I	模拟	模数转换器输入 21。
AIN22	132	PP1	I	模拟	模数转换器输入 22。
AIN23	131	PP0	I	模拟	模数转换器输入 23。
C0+	34	PC6	I	模拟	模拟比较器 0 正极输入。
C0-	33	PC7	I	模拟	模拟比较器 0 负极输入。
C0o	62 112	PF0 (9) PK4 (8)	O	TTL	模拟比较器 0 输出。
C1+	35	PC5	I	模拟	模拟比较器 1 正极输入。
C1-	36	PC4	I	模拟	模拟比较器 1 负极输入。
C1o	63 111	PF1 (9) PK5 (8)	O	TTL	模拟比较器 1 输出。
C2+	127	PJ4	I	模拟	模拟比较器 2 正极输入。
C2-	128	PJ5	I	模拟	模拟比较器 2 负极输入。
C2o	64 110	PF2 (9) PK6 (8)	O	TTL	模拟比较器 2 输出。
CAN0Rx	62 81 136 139	PF0 (3) PN0 (1) PB4 (8) PE4 (8)	I	TTL	CAN模块0接收信号。

表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
CAN0Tx	65 80 135 140	PF3 (3) PN1 (1) PB5 (8) PE5 (8)	O	TTL	CAN模块0发送信号。
CAN1Rx	37 133	PA0 (8) PE6 (8)	I	TTL	CAN模块1接收信号。
CAN1Tx	38 134	PA1 (8) PE7 (8)	O	TTL	CAN模块1发送信号。
GND	6 25 30 44 57 67 79 91 102 114 125 138	固定	-	电源	逻辑及 I/O 管脚的接地参考。
GNDA	10	固定	-	电源	模拟电路 (ADC、模拟比较器等) 的接地参考。GNDA 应与 GND 分开, 尽量避免 VDD 中携带的噪声影响模拟功能。
GNDX	75	固定	-	电源	休眠震荡器的接地。使用晶振时钟源时, 该管脚应仅连接到晶振负载电容, 以提高震荡器对系统噪声的免疫能力。使用外部震荡器时, 该管脚应连接到 GND。
HIB	73	固定	O	TTL	该输出指示处理器是否处于休眠模式。
I2C0SCL	99	PB2 (3)	I/O	OD	I <sup>2</sup> C 模块 0 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C0SDA	100	PB3 (3)	I/O	OD	I <sup>2</sup> C 模块 0 数据。
I2C1SCL	45 51	PA6 (3) PG4 (3)	I/O	OD	I <sup>2</sup> C 模块 1 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C1SDA	46 50	PA7 (3) PG5 (3)	I/O	OD	I <sup>2</sup> C 模块 1 数据。
I2C2SCL	59 139	PF6 (3) PE4 (3)	I/O	OD	I <sup>2</sup> C 模块 2 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C2SDA	58 140	PF7 (3) PE5 (3)	I/O	OD	I <sup>2</sup> C 模块 2 数据。
I2C3SCL	1 55	PD0 (3) PG0 (3)	I/O	OD	I <sup>2</sup> C 模块 3 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C3SDA	2 54	PD1 (3) PG1 (3)	I/O	OD	I <sup>2</sup> C 模块 3 数据。
I2C4SCL	53	PG2 (3)	I/O	OD	I <sup>2</sup> C 模块 4 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C4SDA	52	PG3 (3)	I/O	OD	I <sup>2</sup> C 模块 4 数据。
I2C5SCL	48	PG6 (3)	I/O	OD	I <sup>2</sup> C 模块 5 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
I2C5SDA	47	PG7 (3)	I/O	OD	I <sup>2</sup> C 模块 5 数据。

表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
IDX0	4 31 61 122	PD3 (6) PH1 (5) PF4 (6) PJ2 (5)	I	TTL	QE1 模块 0 索引。
IDX1	36 47 50	PC4 (6) PG7 (5) PG5 (6)	I	TTL	QE1 模块 1 索引。
M0FAULT0	3 32 64 112 143	PD2 (4) PH0 (6) PF2 (4) PK4 (6) PD6 (4)	I	TTL	动作控制模块 0 PWM 故障 0。
M0FAULT1	31 53 65 111 144	PH1 (6) PG2 (4) PF3 (4) PK5 (6) PD7 (4)	I	TTL	动作控制模块 0 PWM 故障 1。
M0FAULT2	28 52 61 110	PH2 (6) PG3 (4) PF4 (4) PK6 (6)	I	TTL	动作控制模块 0 PWM 故障 2。
M0FAULT3	27 60 109	PH3 (6) PF5 (4) PK7 (6)	I	TTL	动作控制模块 0 PWM 故障 3。
M0PWM0	32 131	PH0 (4) PP0 (1)	O	TTL	动作控制模块 0 PWM 0。此信号由模块 0 PWM 发生器 0 控制。
M0PWM1	31 132	PH1 (4) PP1 (1)	O	TTL	动作控制模块 0 PWM 1。此信号由模块 0 PWM 发生器 0 控制。
M0PWM2	11 28 136	PP2 (1) PH2 (4) PB4 (4)	O	TTL	动作控制模块 0 PWM 2。此信号由模块 0 PWM 发生器 1 控制。
M0PWM3	27 135	PH3 (4) PB5 (4)	O	TTL	动作控制模块 0 PWM 3。此信号由模块 0 PWM 发生器 1 控制。
M0PWM4	26 51 83 139	PH4 (4) PG4 (4) PM6 (2) PE4 (4)	O	TTL	动作控制模块 0 PWM 4。此信号由模块 0 PWM 发生器 2 控制。
M0PWM5	23 50 82 140	PH5 (4) PG5 (4) PM7 (2) PE5 (4)	O	TTL	动作控制模块 0 PWM 5。此信号由模块 0 PWM 发生器 2 控制。
M0PWM6	1 20 22 36 48	PD0 (4) PN2 (2) PH6 (4) PC4 (4) PG6 (4)	O	TTL	动作控制模块 0 PWM 6。此信号由模块 0 PWM 发生器 3 控制。
M0PWM7	2 21 35 47 119	PD1 (4) PH7 (4) PC5 (4) PG7 (4) PN3 (2)	O	TTL	动作控制模块 0 PWM 7。此信号由模块 0 PWM 发生器 3 控制。

表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
M1FAULT0	16 58 61	PK0 (6) PF7 (5) PF4 (5)	I	TTL	动作控制模块 1 PWM 故障 0。
M1FAULT1	17 55	PK1 (6) PG0 (5)	I	TTL	动作控制模块 1 PWM 故障 1。
M1FAULT2	18 54	PK2 (6) PG1 (5)	I	TTL	动作控制模块 1 PWM 故障 2。
M1FAULT3	19	PK3 (6)	I	TTL	动作控制模块 1 PWM 故障 3。
M1PWM0	1 53	PD0 (5) PG2 (5)	O	TTL	动作控制模块 1 PWM 0。此信号由模块 1 PWM 发生器 0 控制。
M1PWM1	2 52	PD1 (5) PG3 (5)	O	TTL	动作控制模块 1 PWM 1。此信号由模块 1 PWM 发生器 0 控制。
M1PWM2	45 51 139	PA6 (5) PG4 (5) PE4 (5)	O	TTL	动作控制模块 1 PWM 2。此信号由模块 1 PWM 发生器 1 控制。
M1PWM3	46 50 140	PA7 (5) PG5 (5) PE5 (5)	O	TTL	动作控制模块 1 PWM 3。此信号由模块 1 PWM 发生器 1 控制。
M1PWM4	62 71	PF0 (5) PN4 (2)	O	TTL	动作控制模块 1 PWM 4。此信号由模块 1 PWM 发生器 2 控制。
M1PWM5	63 70	PF1 (5) PN5 (2)	O	TTL	动作控制模块 1 PWM 5。此信号由模块 1 PWM 发生器 2 控制。
M1PWM6	64 69	PF2 (5) PN6 (2)	O	TTL	动作控制模块 1 PWM 6。此信号由模块 1 PWM 发生器 3 控制。
M1PWM7	65 68	PF3 (5) PN7 (2)	O	TTL	动作控制模块 1 PWM 7。此信号由模块 1 PWM 发生器 3 控制。
NMI	62 144	PF0 (8) PD7 (8)	I	TTL	不可屏蔽的中断。
OSC0	92	固定	I	模拟	晶体主振荡器输入或外部时钟参考输入。
OSC1	93	固定	O	模拟	晶体主振荡器输出。使用单端时钟源时不连接。
PA0	37	-	I/O	TTL	GPIO 端口 A 位 0。
PA1	38	-	I/O	TTL	GPIO 端口 A 位 1。
PA2	39	-	I/O	TTL	GPIO 端口 A 位 2。
PA3	40	-	I/O	TTL	GPIO 端口 A 位 3。
PA4	41	-	I/O	TTL	GPIO 端口 A 位 4。
PA5	42	-	I/O	TTL	GPIO 端口 A 位 5。
PA6	45	-	I/O	TTL	GPIO 端口 A 位 6。
PA7	46	-	I/O	TTL	GPIO 端口 A 位 7。
PB0	97	-	I/O	TTL	GPIO 端口 B 位 0。该管脚不能承受 5V 最高电压。
PB1	98	-	I/O	TTL	GPIO 端口 B 位 1。该管脚不能承受 5V 最高电压。
PB2	99	-	I/O	TTL	GPIO 端口 B 位 2。
PB3	100	-	I/O	TTL	GPIO 端口 B 位 3。
PB4	136	-	I/O	TTL	GPIO 端口 B 位 4。
PB5	135	-	I/O	TTL	GPIO 端口 B 位 5。
PC0	118	-	I/O	TTL	GPIO 端口 C 位 0。
PC1	117	-	I/O	TTL	GPIO 端口 C 位 1。

表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
PC2	116	-	I/O	TTL	GPIO 端口 C 位 2。
PC3	115	-	I/O	TTL	GPIO 端口 C 位 3。
PC4	36	-	I/O	TTL	GPIO 端口 C 位 4。
PC5	35	-	I/O	TTL	GPIO 端口 C 位 5。
PC6	34	-	I/O	TTL	GPIO 端口 C 位 6。
PC7	33	-	I/O	TTL	GPIO 端口 C 位 7。
PD0	1	-	I/O	TTL	GPIO 端口 D 位 0。
PD1	2	-	I/O	TTL	GPIO 端口 D 位 1。
PD2	3	-	I/O	TTL	GPIO 端口 D 位 2。
PD3	4	-	I/O	TTL	GPIO 端口 D 位 3。
PD4	141	-	I/O	TTL	GPIO 端口 D 位 4。
PD5	142	-	I/O	TTL	GPIO 端口 D 位 5。
PD6	143	-	I/O	TTL	GPIO 端口 D 位 6。
PD7	144	-	I/O	TTL	GPIO 端口 D 位 7。
PE0	15	-	I/O	TTL	GPIO 端口 E 位 0。
PE1	14	-	I/O	TTL	GPIO 端口 E 位 1。
PE2	13	-	I/O	TTL	GPIO 端口 E 位 2。
PE3	12	-	I/O	TTL	GPIO 端口 E 位 3。
PE4	139	-	I/O	TTL	GPIO 端口 E 位 4。
PE5	140	-	I/O	TTL	GPIO 端口 E 位 5。
PE6	133	-	I/O	TTL	GPIO 端口 E 位 6。
PE7	134	-	I/O	TTL	GPIO 端口 E 位 7。
PF0	62	-	I/O	TTL	GPIO 端口 F 位 0。
PF1	63	-	I/O	TTL	GPIO 端口 F 位 1。
PF2	64	-	I/O	TTL	GPIO 端口 F 位 2。
PF3	65	-	I/O	TTL	GPIO 端口 F 位 3。
PF4	61	-	I/O	TTL	GPIO 端口 F 位 4。
PF5	60	-	I/O	TTL	GPIO 端口 F 位 5。
PF6	59	-	I/O	TTL	GPIO 端口 F 位 6。
PF7	58	-	I/O	TTL	GPIO 端口 F 位 7。
PG0	55	-	I/O	TTL	GPIO 端口 G 位 0。
PG1	54	-	I/O	TTL	GPIO 端口 G 位 1。
PG2	53	-	I/O	TTL	GPIO 端口 G 位 2。
PG3	52	-	I/O	TTL	GPIO 端口 G 位 3。
PG4	51	-	I/O	TTL	GPIO 端口 G 位 4。
PG5	50	-	I/O	TTL	GPIO 端口 G 位 5。
PG6	48	-	I/O	TTL	GPIO 端口 G 位 6。
PG7	47	-	I/O	TTL	GPIO 端口 G 位 7。
PH0	32	-	I/O	TTL	GPIO 端口 H 位 0。
PH1	31	-	I/O	TTL	GPIO 端口 H 位 1。
PH2	28	-	I/O	TTL	GPIO 端口 H 位 2。

表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
PH3	27	-	I/O	TTL	GPIO 端口 H 位 3。
PH4	26	-	I/O	TTL	GPIO 端口 H 位 4。
PH5	23	-	I/O	TTL	GPIO 端口 H 位 5。
PH6	22	-	I/O	TTL	GPIO 端口 H 位 6。
PH7	21	-	I/O	TTL	GPIO 端口 H 位 7。
PhA0	26 62 143	PH4 (5) PF0 (6) PD6 (6)	I	TTL	QE1 模块 0 相位 A。
PhA1	35 52 55	PC5 (6) PG3 (6) PG0 (6)	I	TTL	QE1 模块 1 相位 A。
PhB0	23 63 144	PH5 (5) PF1 (6) PD7 (6)	I	TTL	QE1 模块 0 相位 B。
PhB1	34 51 54	PC6 (6) PG4 (6) PG1 (6)	I	TTL	QE1 模块 1 相位 B。
PJ0	120	-	I/O	TTL	GPIO 端口 J 位 0。
PJ1	121	-	I/O	TTL	GPIO 端口 J 位 1。
PJ2	122	-	I/O	TTL	GPIO 端口 J 位 2。
PJ3	123	-	I/O	TTL	GPIO 端口 J 位 3。
PJ4	127	-	I/O	TTL	GPIO 端口 J 位 4。
PJ5	128	-	I/O	TTL	GPIO 端口 J 位 5。
PJ6	129	-	I/O	TTL	GPIO 端口 J 位 6。
PJ7	130	-	I/O	TTL	GPIO 端口 J 位 7。
PK0	16	-	I/O	TTL	GPIO 端口 K 位 0。
PK1	17	-	I/O	TTL	GPIO 端口 K 位 1。
PK2	18	-	I/O	TTL	GPIO 端口 K 位 2。
PK3	19	-	I/O	TTL	GPIO 端口 K 位 3。
PK4	112	-	I/O	TTL	GPIO 端口 K 位 4。
PK5	111	-	I/O	TTL	GPIO 端口 K 位 5。
PK6	110	-	I/O	TTL	GPIO 端口 K 位 6。
PK7	109	-	I/O	TTL	GPIO 端口 K 位 7。
PL0	108	-	I/O	TTL	GPIO 端口 L 位 0。
PL1	107	-	I/O	TTL	GPIO 端口 L 位 1。
PL2	106	-	I/O	TTL	GPIO 端口 L 位 2。
PL3	105	-	I/O	TTL	GPIO 端口 L 位 3。
PL4	104	-	I/O	TTL	GPIO 端口 L 位 4。
PL5	103	-	I/O	TTL	GPIO 端口 L 位 5。
PL6	96	-	I/O	TTL	GPIO 端口 L 位 6。
PL7	95	-	I/O	TTL	GPIO 端口 L 位 7。
PM0	89	-	I/O	TTL	GPIO 端口 M 位 0。
PM1	88	-	I/O	TTL	GPIO 端口 M 位 1。



表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
PM2	87	-	I/O	TTL	GPIO 端口 M 位 2。
PM3	86	-	I/O	TTL	GPIO 端口 M 位 3。
PM4	85	-	I/O	TTL	GPIO 端口 M 位 4。
PM5	84	-	I/O	TTL	GPIO 端口 M 位 5。
PM6	83	-	I/O	TTL	GPIO 端口 M 位 6。
PM7	82	-	I/O	TTL	GPIO 端口 M 位 7。
PN0	81	-	I/O	TTL	GPIO 端口 N 位 0。
PN1	80	-	I/O	TTL	GPIO 端口 N 位 1。
PN2	20	-	I/O	TTL	GPIO 端口 N 位 2。
PN3	119	-	I/O	TTL	GPIO 端口 N 位 3。
PN4	71	-	I/O	TTL	GPIO 端口 N 位 4。
PN5	70	-	I/O	TTL	GPIO 端口 N 位 5。
PN6	69	-	I/O	TTL	GPIO 端口 N 位 6。
PN7	68	-	I/O	TTL	GPIO 端口 N 位 7。
PP0	131	-	I/O	TTL	GPIO 端口 P 位 0。
PP1	132	-	I/O	TTL	GPIO 端口 P 位 1。
PP2	11	-	I/O	TTL	GPIO 端口 P 位 2。
$\overline{RST}$	90	固定	I	TTL	系统复位输入。
RTCCLK	112	PK4 (7)	O	TTL	休眠模块缓冲版的 32.768-kHz 时钟。此部分处于休眠模式时, 该信号并非输出信号。
SSI0Ck	39	PA2 (2)	I/O	TTL	SSI 模块 0 时钟
SSI0Fss	40	PA3 (2)	I/O	TTL	SSI 模块 0 帧信号
SSI0Rx	41	PA4 (2)	I	TTL	SSI 模块 0 接收
SSI0Tx	42	PA5 (2)	O	TTL	SSI 模块 0 发送
SSI1Ck	1 64	PD0 (2) PF2 (2)	I/O	TTL	SSI 模块 1 时钟
SSI1Fss	2 65	PD1 (2) PF3 (2)	I/O	TTL	SSI 模块 1 帧信号
SSI1Rx	3 62	PD2 (2) PF0 (2)	I	TTL	SSI 模块 1 接收
SSI1Tx	4 63	PD3 (2) PF1 (2)	O	TTL	SSI 模块 1 发送
SSI2Ck	26 136	PH4 (2) PB4 (2)	I/O	TTL	SSI 模块 2 时钟
SSI2Fss	23 135	PH5 (2) PB5 (2)	I/O	TTL	SSI 模块 2 帧信号
SSI2Rx	22	PH6 (2)	I	TTL	SSI 模块 2 接收
SSI2Tx	21	PH7 (2)	O	TTL	SSI 模块 2 发送
SSI3Ck	1 16 32	PD0 (1) PK0 (2) PH0 (2)	I/O	TTL	SSI 模块 3 时钟
SSI3Fss	2 17 31	PD1 (1) PK1 (2) PH1 (2)	I/O	TTL	SSI 模块 3 帧信号

表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
SSI3Rx	3 18 28	PD2 (1) PK2 (2) PH2 (2)	I	TTL	SSI模块3接收
SSI3Tx	4 19 27	PD3 (1) PK3 (2) PH3 (2)	O	TTL	SSI模块3发送
SWCLK	118	PC0 (1)	I	TTL	JTAG/SWD CLK 信号。
SWDIO	117	PC1 (1)	I/O	TTL	JTAG TMS 及 SWDIO 信号。
SWO	115	PC3 (1)	O	TTL	JTAG TDO 及 SWO 信号。
T0CCP0	62 108	PF0 (7) PL0 (7)	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 0。
T0CCP1	63 107	PF1 (7) PL1 (7)	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 1。
T1CCP0	64 106 120 136	PF2 (7) PL2 (7) PJ0 (7) PB4 (7)	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 0。
T1CCP1	65 105 121 135	PF3 (7) PL3 (7) PJ1 (7) PB5 (7)	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 1。
T2CCP0	61 97 104 122	PF4 (7) PB0 (7) PL4 (7) PJ2 (7)	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 0。
T2CCP1	60 98 103 123	PF5 (7) PB1 (7) PL5 (7) PJ3 (7)	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 1。
T3CCP0	59 96 99 127	PF6 (7) PL6 (7) PB2 (7) PJ4 (7)	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 0。
T3CCP1	58 95 100 128	PF7 (7) PL7 (7) PB3 (7) PJ5 (7)	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 1。
T4CCP0	55 89 118 131	PG0 (7) PM0 (7) PC0 (7) PP0 (7)	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 0。
T4CCP1	54 88 117 132	PG1 (7) PM1 (7) PC1 (7) PP1 (7)	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 1。
T5CCP0	11 53 87 116	PP2 (7) PG2 (7) PM2 (7) PC2 (7)	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 0。

表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
T5CCP1	52 86 115	PG3 (7) PM3 (7) PC3 (7)	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 1。
TCK	118	PC0 (1)	I	TTL	JTAG/SWD CLK 信号。
TDI	116	PC2 (1)	I	TTL	JTAG TDI 信号。
TDO	115	PC3 (1)	O	TTL	JTAG TDO 及 SWO 信号。
TMS	117	PC1 (1)	I	TTL	JTAG TMS 及 SWDIO 信号。
TRCLK	65	PF3 (14)	O	TTL	跟踪时钟信号。
TRD0	64	PF2 (14)	O	TTL	跟踪数据信号 0。
TRD1	63	PF1 (14)	O	TTL	跟踪数据信号 1。
TRD2	62	PF0 (14)	O	TTL	跟踪数据信号 2。
TRD3	61	PF4 (14)	O	TTL	跟踪数据信号 3。
U0Rx	37	PA0 (1)	I	TTL	UART 模块 0 接收信号。
U0Tx	38	PA1 (1)	O	TTL	UART 模块 0 发送信号。
U1CTS	35 63	PC5 (8) PF1 (1)	I	TTL	UART 模块 1 CTS ( Clear to Send , 允许发送 ) 调制解调器流控输入信号。
U1DCD	64	PF2 (1)	I	TTL	UART 模块 1 DCD ( Data Carrier Detect , 数据载波检测 ) 调制解调器状态输入信号。
U1DSR	65	PF3 (1)	I	TTL	UART 模块 1 DSR ( Data Set Ready , 数据设备就绪 ) 调制解调器输出控制线。
U1DTR	61	PF4 (1)	O	TTL	UART 模块 1 DTR ( Data Terminal Ready , 数据终端就绪 ) 调制解调器状态输入信号。
U1RI	134	PE7 (1)	I	TTL	UART 模块 1 RI ( Ring Indicator , 振铃指示 ) 调制解调器状态输入信号。
U1RTS	36 62	PC4 (8) PF0 (1)	O	TTL	UART 模块 1 RTS ( Request to Send , 请求发送 ) 调制解调器流控输出线。
U1Rx	36 97	PC4 (2) PB0 (1)	I	TTL	UART 模块 1 接收信号。
U1Tx	35 98	PC5 (2) PB1 (1)	O	TTL	UART 模块 1 发送信号。
U2Rx	51 143	PG4 (1) PD6 (1)	I	TTL	UART 模块 2 接收信号。
U2Tx	50 144	PG5 (1) PD7 (1)	O	TTL	UART 模块 2 发送信号。
U3Rx	34	PC6 (1)	I	TTL	UART 模块 3 接收信号。
U3Tx	33	PC7 (1)	O	TTL	UART 模块 3 发送信号。
U4Rx	36 120	PC4 (1) PJ0 (1)	I	TTL	UART 模块 4 接收信号。
U4Tx	35 121	PC5 (1) PJ1 (1)	O	TTL	UART 模块 4 发送信号。
U5Rx	122 139	PJ2 (1) PE4 (1)	I	TTL	UART 模块 5 接收信号。
U5Tx	123 140	PJ3 (1) PE5 (1)	O	TTL	UART 模块 5 发送信号。
U6Rx	127 141	PJ4 (1) PD4 (1)	I	TTL	UART 模块 6 接收信号。

表 23-3. 按信号名称分类的信号 ( 续 )

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
U6Tx	128 142	PJ5 (1) PD5 (1)	O	TTL	UART 模块 6 发送信号。
U7Rx	15 112	PE0 (1) PK4 (1)	I	TTL	UART 模块 7 接收信号。
U7Tx	14 111	PE1 (1) PK5 (1)	O	TTL	UART 模块 7 发送信号。
USB0DM	95	PL7	I/O	模拟	USB0 的双向差分数据管脚 ( USB 规范中的 D- )。
USB0DP	96	PL6	I/O	模拟	USB0 的双向差分数据管脚 ( USB 规范中的 D+ )。
USB0EPEN	3 34 51 61	PD2 (8) PC6 (8) PG4 (8) PF4 (8)	O	TTL	可选择性地以主机模式控制外部电源, 向 USB 总线供电。
USB0ID	97	PB0	I	模拟	此信号用于检测 USB ID 信号的状态。此时 USB PHY 将在内部启用一个上拉电阻, 通过外部元件 ( USB 连接器 ) 检测 USB 控制器的初始状态 ( 即电缆的 A 侧设置下拉电阻, B 侧设置上拉电阻 )。
USB0PFLT	4 33 50 60	PD3 (8) PC7 (8) PG5 (8) PF5 (8)	I	TTL	可选择性地用于主机模式的外部电源, 指示外部电源的错误状态。
USB0VBUS	98	PB1	I/O	模拟	此信号用于会话请求协议。USB PHY 可通过此信号检测 VBUS 的电平, 并在 VBUS 脉冲期间短时下拉。
VBAT	77	固定	-	电源	休眠模块的电源供应源它通常连接到电池的正极端并用作为备用电池/休眠模块电源供应器的电源。
VDD	5 24 29 43 56 66 78 94 101 113 124 137	固定	-	电源	I/O 和某些逻辑的电源正极。
VDDA	7	固定	-	电源	模拟电路 ( ADC、模拟比较器等 ) 的电源正端。VDDA 应与 VDD 分开, 尽量避免 VDD 中携带的噪声影响模拟功能。不论系统如何实施, VDDA 管脚的电压必须符合表 25-2 在 1292 页中的规格。
VDDC	49 126	固定	-	电源	为主要的逻辑部分 ( 包括处理器内核以及大部分片上外设 ) 供电的电源正端。该管脚上的电压为 1.2 V, 由片上 LDO 提供。按照表 25-8 在 1297 页中的规定, VDDC 管脚只应相互连接 以及外部电容。
VREFA+	8	固定	-	模拟	参考电压用于指定 ADC 转换为最大值时的电压。此管脚与指定最小值的 VREFA- 结合使用。也就是说, 若 AINn 输入的信号电压与 VREFA+ 电压相等, 则转换结果为 4095。VREFA+ 的电压限定为表 25-25 在 1307 页中指定的范围。

表 23-3. 按信号名称分类的信号 (续)

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
VREFA-	9	固定	-	模拟	参考电压用于指定 ADC 转换为最小值时的输入电压。此管脚与指定最大值的 VREFA+ 结合使用。也就是说, 若 AINn 输入的信号电压与 VREFA- 电压相等, 则转换结果为 0。而若 AINn 输入的信号电压与 VREFA+ 电压相等, 则转换结果为 4095。VREFA- 的电压限定为表 25-25 在 1307 页中指定的范围。
$\overline{\text{WAKE}}$	72	固定	I	TTL	当有效时外部输入将处理器从休眠模式中唤醒。
WT0CCP0	36 51 83 108	PC4 (7) PG4 (7) PM6 (7) PL0 (8)	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 0。
WT0CCP1	35 50 82 107	PC5 (7) PG5 (7) PM7 (7) PL1 (8)	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 1。
WT1CCP0	34 48 106 110	PC6 (7) PG6 (7) PL2 (8) PK6 (7)	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 0。
WT1CCP1	33 47 105 109	PC7 (7) PG7 (7) PL3 (8) PK7 (7)	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 1。
WT2CCP0	1 20 32 104	PD0 (7) PN2 (7) PH0 (7) PL4 (8)	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 0。
WT2CCP1	2 31 103 119	PD1 (7) PH1 (7) PL5 (8) PN3 (7)	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 1。
WT3CCP0	3 26 71 96	PD2 (7) PH4 (7) PN4 (7) PL6 (8)	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 0。
WT3CCP1	4 23 70 95	PD3 (7) PH5 (7) PN5 (7) PL7 (8)	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 1。
WT4CCP0	22 69 89 141	PH6 (7) PN6 (7) PM0 (8) PD4 (7)	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 0。
WT4CCP1	21 68 88 142	PH7 (7) PN7 (7) PM1 (8) PD5 (7)	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 1。
WT5CCP0	28 87 143	PH2 (7) PM2 (8) PD6 (7)	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 0。

表 23-3. 按信号名称分类的信号 ( 续 )

管脚名称	管脚编号	管脚复用/管脚赋值	管脚类型	缓冲区类型 <sup>a</sup>	描述
WT5CCP1	27 86 144	PH3 (7) PM3 (8) PD7 (7)	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 1。
XOSC0	74	固定	I	模拟	休眠模块晶体振荡器输入或外部时钟参考输入。请注意休眠模块 RTC 采用 32.768-kHz 晶体或 32.768-kHz 振荡器。
XOSC1	76	固定	O	模拟	休眠模块晶体振荡器输出。使用单端时钟源时不连接。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 23.3 按功能分类的信号 ( GPIO 除外 )

表 23-4. 按功能分类的信号 ( GPIO 除外 )

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
ADC	AIN0	12	I	模拟	模数转换器输入 0。
	AIN1	13	I	模拟	模数转换器输入 1。
	AIN2	14	I	模拟	模数转换器输入 2。
	AIN3	15	I	模拟	模数转换器输入 3。
	AIN4	144	I	模拟	模数转换器输入 4。
	AIN5	143	I	模拟	模数转换器输入 5。
	AIN6	142	I	模拟	模数转换器输入 6。
	AIN7	141	I	模拟	模数转换器输入 7。
	AIN8	140	I	模拟	模数转换器输入 8。
	AIN9	139	I	模拟	模数转换器输入 9。
	AIN10	136	I	模拟	模数转换器输入 10。
	AIN11	135	I	模拟	模数转换器输入 11。
	AIN12	4	I	模拟	模数转换器输入 12。
	AIN13	3	I	模拟	模数转换器输入 13。
	AIN14	2	I	模拟	模数转换器输入 14。
	AIN15	1	I	模拟	模数转换器输入 15。
	AIN16	16	I	模拟	模数转换器输入 16。
	AIN17	17	I	模拟	模数转换器输入 17。
	AIN18	18	I	模拟	模数转换器输入 18。
	AIN19	19	I	模拟	模数转换器输入 19。
	AIN20	134	I	模拟	模数转换器输入 20。
	AIN21	133	I	模拟	模数转换器输入 21。
	AIN22	132	I	模拟	模数转换器输入 22。
	AIN23	131	I	模拟	模数转换器输入 23。
	VREFA+	8	-	模拟	参考电压用于指定 ADC 转换为最大值时的电压。此管脚与指定最小值的 VREFA- 结合使用。也就是说，若 AINn 输入的信号电压与 VREFA+ 电压相等，则转换结果为 4095。VREFA+ 的电压限定为表 25-25 在 1307 页中指定的范围。
VREFA-	9	-	模拟	参考电压用于指定 ADC 转换为最小值时的输入电压。此管脚与指定最大值的 VREFA+ 结合使用。也就是说，若 AINn 输入的信号电压与 VREFA- 电压相等，则转换结果为 0。而若 AINn 输入的信号电压与 VREFA+ 电压相等，则转换结果为 4095。VREFA- 的电压限定为表 25-25 在 1307 页中指定的范围。	

表 23-4. 按功能分类的信号 ( GPIO 除外 ) ( 续 )

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
I2C	I2C0SCL	99	I/O	OD	I <sup>2</sup> C 模块 0 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	I2C0SDA	100	I/O	OD	I <sup>2</sup> C 模块 0 数据。
	I2C1SCL	45 51	I/O	OD	I <sup>2</sup> C 模块 1 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	I2C1SDA	46 50	I/O	OD	I <sup>2</sup> C 模块 1 数据。
	I2C2SCL	59 139	I/O	OD	I <sup>2</sup> C 模块 2 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	I2C2SDA	58 140	I/O	OD	I <sup>2</sup> C 模块 2 数据。
	I2C3SCL	1 55	I/O	OD	I <sup>2</sup> C 模块 3 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	I2C3SDA	2 54	I/O	OD	I <sup>2</sup> C 模块 3 数据。
	I2C4SCL	53	I/O	OD	I <sup>2</sup> C 模块 4 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	I2C4SDA	52	I/O	OD	I <sup>2</sup> C 模块 4 数据。
	I2C5SCL	48	I/O	OD	I <sup>2</sup> C 模块 5 时钟。请注意该信号具有有源上拉。不应将相应的端口管脚配置为开漏。
	I2C5SDA	47	I/O	OD	I <sup>2</sup> C 模块 5 数据。
JTAG/SWD/SWO	SWCLK	118	I	TTL	JTAG/SWD CLK 信号。
	SWDIO	117	I/O	TTL	JTAG TMS 及 SWDIO 信号。
	SWO	115	O	TTL	JTAG TDO 及 SWO 信号。
	TCK	118	I	TTL	JTAG/SWD CLK 信号。
	TDI	116	I	TTL	JTAG TDI 信号。
	TDO	115	O	TTL	JTAG TDO 及 SWO 信号。
	TMS	117	I	TTL	JTAG TMS 及 SWDIO 信号。



表 23-4. 按功能分类的信号 (GPIO 除外) (续)

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
PWM	M0FAULT0	3 32 64 112 143	I	TTL	动作控制模块 0 PWM 故障 0。
	M0FAULT1	31 53 65 111 144	I	TTL	动作控制模块 0 PWM 故障 1。
	M0FAULT2	28 52 61 110	I	TTL	动作控制模块 0 PWM 故障 2。
	M0FAULT3	27 60 109	I	TTL	动作控制模块 0 PWM 故障 3。
	M0PWM0	32 131	O	TTL	动作控制模块 0 PWM 0。此信号由模块 0 PWM 发生器 0 控制。
	M0PWM1	31 132	O	TTL	动作控制模块 0 PWM 1。此信号由模块 0 PWM 发生器 0 控制。
	M0PWM2	11 28 136	O	TTL	动作控制模块 0 PWM 2。此信号由模块 0 PWM 发生器 1 控制。
	M0PWM3	27 135	O	TTL	动作控制模块 0 PWM 3。此信号由模块 0 PWM 发生器 1 控制。
	M0PWM4	26 51 83 139	O	TTL	动作控制模块 0 PWM 4。此信号由模块 0 PWM 发生器 2 控制。
	M0PWM5	23 50 82 140	O	TTL	动作控制模块 0 PWM 5。此信号由模块 0 PWM 发生器 2 控制。
	M0PWM6	1 20 22 36 48	O	TTL	动作控制模块 0 PWM 6。此信号由模块 0 PWM 发生器 3 控制。
	M0PWM7	2 21 35 47 119	O	TTL	动作控制模块 0 PWM 7。此信号由模块 0 PWM 发生器 3 控制。
	M1FAULT0	16 58 61	I	TTL	动作控制模块 1 PWM 故障 0。
	M1FAULT1	17 55	I	TTL	动作控制模块 1 PWM 故障 1。
	M1FAULT2	18 54	I	TTL	动作控制模块 1 PWM 故障 2。
	M1FAULT3	19	I	TTL	动作控制模块 1 PWM 故障 3。

表 23-4. 按功能分类的信号 ( GPIO 除外 ) ( 续 )

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
	M1PWM0	1 53	O	TTL	动作控制模块 1 PWM 0。此信号由模块 1 PWM 发生器 0 控制。
	M1PWM1	2 52	O	TTL	动作控制模块 1 PWM 1。此信号由模块 1 PWM 发生器 0 控制。
	M1PWM2	45 51 139	O	TTL	动作控制模块 1 PWM 2。此信号由模块 1 PWM 发生器 1 控制。
	M1PWM3	46 50 140	O	TTL	动作控制模块 1 PWM 3。此信号由模块 1 PWM 发生器 1 控制。
	M1PWM4	62 71	O	TTL	动作控制模块 1 PWM 4。此信号由模块 1 PWM 发生器 2 控制。
	M1PWM5	63 70	O	TTL	动作控制模块 1 PWM 5。此信号由模块 1 PWM 发生器 2 控制。
	M1PWM6	64 69	O	TTL	动作控制模块 1 PWM 6。此信号由模块 1 PWM 发生器 3 控制。
	M1PWM7	65 68	O	TTL	动作控制模块 1 PWM 7。此信号由模块 1 PWM 发生器 3 控制。
QEI	IDX0	4 31 61 122	I	TTL	QEI 模块 0 索引。
	IDX1	36 47 50	I	TTL	QEI 模块 1 索引。
	PhA0	26 62 143	I	TTL	QEI 模块 0 相位 A。
	PhA1	35 52 55	I	TTL	QEI 模块 1 相位 A。
	PhB0	23 63 144	I	TTL	QEI 模块 0 相位 B。
	PhB1	34 51 54	I	TTL	QEI 模块 1 相位 B。

表 23-4. 按功能分类的信号 (GPIO 除外) (续)

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
SSI	SSI0Clk	39	I/O	TTL	SSI模块0时钟
	SSI0Fss	40	I/O	TTL	SSI 模块 0 帧信号
	SSI0Rx	41	I	TTL	SSI模块0接收
	SSI0Tx	42	O	TTL	SSI模块0发送
	SSI1Clk	1 64	I/O	TTL	SSI模块1时钟
	SSI1Fss	2 65	I/O	TTL	SSI 模块 1 帧信号
	SSI1Rx	3 62	I	TTL	SSI模块1接收
	SSI1Tx	4 63	O	TTL	SSI模块1发送
	SSI2Clk	26 136	I/O	TTL	SSI模块2时钟
	SSI2Fss	23 135	I/O	TTL	SSI 模块 2 帧信号
	SSI2Rx	22	I	TTL	SSI模块2接收
	SSI2Tx	21	O	TTL	SSI模块2发送
	SSI3Clk	1 16 32	I/O	TTL	SSI模块3时钟
	SSI3Fss	2 17 31	I/O	TTL	SSI 模块 3 帧信号
	SSI3Rx	3 18 28	I	TTL	SSI模块3接收
	SSI3Tx	4 19 27	O	TTL	SSI模块3发送

表 23-4. 按功能分类的信号 (GPIO 除外) (续)

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
UART	U0Rx	37	I	TTL	UART 模块 0 接收信号。
	U0Tx	38	O	TTL	UART 模块 0 发送信号。
	U1CTS	35 63	I	TTL	UART 模块 1 CTS (Clear to Send, 允许发送) 调制解调器流控输入信号。
	U1DCD	64	I	TTL	UART 模块 1 DCD (Data Carrier Detect, 数据载波检测) 调制解调器状态输入信号。
	U1DSR	65	I	TTL	UART 模块 1 DSR (Data Set Ready, 数据设备就绪) 调制解调器输出控制线。
	U1DTR	61	O	TTL	UART 模块 1 DTR (Data Terminal Ready, 数据终端就绪) 调制解调器状态输入信号。
	U1RI	134	I	TTL	UART 模块 1 RI (Ring Indicator, 振铃指示) 调制解调器状态输入信号。
	U1RTS	36 62	O	TTL	UART 模块 1 RTS (Request to Send, 请求发送) 调制解调器流控输出线。
	U1Rx	36 97	I	TTL	UART 模块 1 接收信号。
	U1Tx	35 98	O	TTL	UART 模块 1 发送信号。
	U2Rx	51 143	I	TTL	UART 模块 2 接收信号。
	U2Tx	50 144	O	TTL	UART 模块 2 发送信号。
	U3Rx	34	I	TTL	UART 模块 3 接收信号。
	U3Tx	33	O	TTL	UART 模块 3 发送信号。
	U4Rx	36 120	I	TTL	UART 模块 4 接收信号。
	U4Tx	35 121	O	TTL	UART 模块 4 发送信号。
	U5Rx	122 139	I	TTL	UART 模块 5 接收信号。
	U5Tx	123 140	O	TTL	UART 模块 5 发送信号。
	U6Rx	127 141	I	TTL	UART 模块 6 接收信号。
	U6Tx	128 142	O	TTL	UART 模块 6 发送信号。
	U7Rx	15 112	I	TTL	UART 模块 7 接收信号。
U7Tx	14 111	O	TTL	UART 模块 7 发送信号。	

表 23-4. 按功能分类的信号 (GPIO 除外) (续)

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
USB	USB0DM	95	I/O	模拟	USB0 的双向差分数据管脚 (USB 规范中的 D-)。
	USB0DP	96	I/O	模拟	USB0 的双向差分数据管脚 (USB 规范中的 D+)。
	USB0EPEN	3 34 51 61	O	TTL	可选择性地以主机模式控制外部电源, 向 USB 总线供电。
	USB0ID	97	I	模拟	此信号用于检测 USB ID 信号的状态。此时 USB PHY 将在内部启用一个上拉电阻, 通过外部元件 (USB 连接器) 检测 USB 控制器的初始状态 (即电缆的 A 侧设置下拉电阻, B 侧设置上拉电阻)。
	USB0PFLT	4 33 50 60	I	TTL	可选择性地用于主机模式的外部电源, 指示外部电源的错误状态。
	USB0VBUS	98	I/O	模拟	此信号用于会话请求协议。USB PHY 可通过此信号检测 VBUS 的电平, 并在 VBUS 脉冲期间短时下拉。
休眠	GNDX	75	-	电源	休眠振荡器的接地。使用晶振时钟源时, 该管脚应仅连接到晶振负载电容, 以提高振荡器对系统噪声的免疫能力。使用外部振荡器时, 该管脚应连接到 GND。
	HIB	73	O	TTL	该输出指示处理器是否处于休眠模式。
	RTCCLK	112	O	TTL	休眠模块缓冲版的 32.768-kHz 时钟。此部分处于休眠模式时, 该信号并非输出信号。
	VBAT	77	-	电源	休眠模块的电源供应源它通常连接到电池的正极端并用作备用电池/休眠模块电源供应器的电源。
	WAKE	72	I	TTL	当有效时外部输入将处理器从休眠模式中唤醒。
	XOSC0	74	I	模拟	休眠模块晶体振荡器输入或外部时钟参考输入。请注意休眠模块 RTC 采用 32.768-kHz 晶体或 32.768-kHz 振荡器。
	XOSC1	76	O	模拟	休眠模块晶体振荡器输出。使用单端时钟源时不连接。
内核	TRCLK	65	O	TTL	跟踪时钟信号。
	TRD0	64	O	TTL	跟踪数据信号 0。
	TRD1	63	O	TTL	跟踪数据信号 1。
	TRD2	62	O	TTL	跟踪数据信号 2。
	TRD3	61	O	TTL	跟踪数据信号 3。
控制器局域网	CAN0Rx	62 81 136 139	I	TTL	CAN 模块 0 接收信号。
	CAN0Tx	65 80 135 140	O	TTL	CAN 模块 0 发送信号。
	CAN1Rx	37 133	I	TTL	CAN 模块 1 接收信号。
	CAN1Tx	38 134	O	TTL	CAN 模块 1 发送信号。

表 23-4. 按功能分类的信号 (GPIO 除外) (续)

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
模拟比较器	C0+	34	I	模拟	模拟比较器 0 正极输入。
	C0-	33	I	模拟	模拟比较器 0 负极输入。
	C0o	62 112	O	TTL	模拟比较器 0 输出。
	C1+	35	I	模拟	模拟比较器 1 正极输入。
	C1-	36	I	模拟	模拟比较器 1 负极输入。
	C1o	63 111	O	TTL	模拟比较器 1 输出。
	C2+	127	I	模拟	模拟比较器 2 正极输入。
	C2-	128	I	模拟	模拟比较器 2 负极输入。
	C2o	64 110	O	TTL	模拟比较器 2 输出。
电源	GND	6 25 30 44 57 67 79 91 102 114 125 138	-	电源	逻辑及 I/O 管脚的接地参考。
	GNDA	10	-	电源	模拟电路 (ADC、模拟比较器等) 的接地参考。GNDA 应与 GND 分开, 尽量避免 VDD 中携带的噪声影响模拟功能。
	VDD	5 24 29 43 56 66 78 94 101 113 124 137	-	电源	I/O 和某些逻辑的电源正极。
	VDDA	7	-	电源	模拟电路 (ADC、模拟比较器等) 的电源正端。VDDA 应与 VDD 分开, 尽量避免 VDD 中携带的噪声影响模拟功能。不论系统如何实施, VDDA 管脚的电压必须符合表 25-2 在 1292 页中的规格。
	VDDC	49 126	-	电源	为主要的逻辑部分 (包括处理器内核以及大部分片上外设) 供电的电源正端。该管脚上的电压为 1.2 V, 由片上 LDO 提供。按照表 25-8 在 1297 页中的规定, VDDC 管脚只应相互连接 以及外部电容。
系统控制及时钟	NMI	62 144	I	TTL	不可屏蔽的中断。
	OSC0	92	I	模拟	晶体主振荡器输入或外部时钟参考输入。
	OSC1	93	O	模拟	晶体主振荡器输出。使用单端时钟源时不连接。
	RST	90	I	TTL	系统复位输入。

表 23-4. 按功能分类的信号 (GPIO 除外) (续)

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
通用定时器	T0CCP0	62 108	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 0。
	T0CCP1	63 107	I/O	TTL	16/32 位 Timer 0 捕获/比较/PWM 1。
	T1CCP0	64 106 120 136	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 0。
	T1CCP1	65 105 121 135	I/O	TTL	16/32 位 Timer 1 捕获/比较/PWM 1。
	T2CCP0	61 97 104 122	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 0。
	T2CCP1	60 98 103 123	I/O	TTL	16/32 位 Timer 2 捕获/比较/PWM 1。
	T3CCP0	59 96 99 127	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 0。
	T3CCP1	58 95 100 128	I/O	TTL	16/32 位 Timer 3 捕获/比较/PWM 1。
	T4CCP0	55 89 118 131	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 0。
	T4CCP1	54 88 117 132	I/O	TTL	16/32 位 Timer 4 捕获/比较/PWM 1。
	T5CCP0	11 53 87 116	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 0。
	T5CCP1	52 86 115	I/O	TTL	16/32 位 Timer 5 捕获/比较/PWM 1。
	WT0CCP0	36 51 83 108	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 0。
	WT0CCP1	35 50 82 107	I/O	TTL	32/64 位宽 Timer 0 捕获/比较/PWM 1。
	WT1CCP0		I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 0。

表 23-4. 按功能分类的信号 ( GPIO 除外 ) ( 续 )

功能	管脚名称	管脚编号	管脚类型	缓冲区类型 <sup>a</sup>	描述
		34 48 106 110			
	WT1CCP1	33 47 105 109	I/O	TTL	32/64 位宽 Timer 1 捕获/比较/PWM 1。
	WT2CCP0	1 20 32 104	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 0。
	WT2CCP1	2 31 103 119	I/O	TTL	32/64 位宽 Timer 2 捕获/比较/PWM 1。
	WT3CCP0	3 26 71 96	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 0。
	WT3CCP1	4 23 70 95	I/O	TTL	32/64 位宽 Timer 3 捕获/比较/PWM 1。
	WT4CCP0	22 69 89 141	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 0。
	WT4CCP1	21 68 88 142	I/O	TTL	32/64 位宽 Timer 4 捕获/比较/PWM 1。
	WT5CCP0	28 87 143	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 0。
	WT5CCP1	27 86 144	I/O	TTL	32/64 位宽 Timer 5 捕获/比较/PWM 1。

a. TTL 表示管脚的电压水平与 TTL 一致。

## 23.4 GPIO 管脚和复用功能

表 23-5. GPIO 管脚和复用功能

IO	管脚	模拟功能	数字功能 ( GPIOCTL PMCx 位域编码 ) <sup>a</sup>											
			1	2	3	4	5	6	7	8	9	14	15	
PA0	37	-	U0Rx	-	-	-	-	-	-	-	CAN1Rx	-	-	-
PA1	38	-	U0Tx	-	-	-	-	-	-	-	CAN1Tx	-	-	-
PA2	39	-	-	SSI0Clk	-	-	-	-	-	-	-	-	-	-
PA3	40	-	-	SSI0Fss	-	-	-	-	-	-	-	-	-	-
PA4	41	-	-	SSI0Rx	-	-	-	-	-	-	-	-	-	-



表 23-5. GPIO 管脚和复用功能 (续)

IO	管脚	模拟功能	数字功能 (GPIOCTL PMCx 位域编码) <sup>a</sup>											
			1	2	3	4	5	6	7	8	9	14	15	
PA5	42	-	-	SSI0Tx	-	-	-	-	-	-	-	-	-	-
PA6	45	-	-	-	I2C1SCL	-	M1PWM2	-	-	-	-	-	-	-
PA7	46	-	-	-	I2C1SDA	-	M1PWM3	-	-	-	-	-	-	-
PB0	97	USB0ID	U1Rx	-	-	-	-	-	-	T2CCP0	-	-	-	-
PB1	98	USB0BUS	U1Tx	-	-	-	-	-	-	T2CCP1	-	-	-	-
PB2	99	-	-	-	I2C0SCL	-	-	-	-	T3CCP0	-	-	-	-
PB3	100	-	-	-	I2C0SDA	-	-	-	-	T3CCP1	-	-	-	-
PB4	136	AIN10	-	SSI2Ck	-	M0PWM2	-	-	-	T1CCP0	CAN0Rx	-	-	-
PB5	135	AIN11	-	SSI2Fss	-	M0PWM3	-	-	-	T1CCP1	CAN0Tx	-	-	-
PC0	118	-	TCK SWCLK	-	-	-	-	-	-	T4CCP0	-	-	-	-
PC1	117	-	TMS SWDIO	-	-	-	-	-	-	T4CCP1	-	-	-	-
PC2	116	-	TDI	-	-	-	-	-	-	T5CCP0	-	-	-	-
PC3	115	-	TDO SWO	-	-	-	-	-	-	T5CCP1	-	-	-	-
PC4	36	C1-	U4Rx	U1Rx	-	M0PWM6	-	IDX1	WT0CCP0	U1RTS	-	-	-	-
PC5	35	C1+	U4Tx	U1Tx	-	M0PWM7	-	PhA1	WT0CCP1	U1CTS	-	-	-	-
PC6	34	C0+	U3Rx	-	-	-	-	PhB1	WT1CCP0	USB0EPEN	-	-	-	-
PC7	33	C0-	U3Tx	-	-	-	-	-	WT1CCP1	USB0PFLT	-	-	-	-
PD0	1	AIN15	SSI3Ck	SSI1Ck	I2C3SCL	M0PWM6	M1PWM0	-	WT2CCP0	-	-	-	-	-
PD1	2	AIN14	SSI3Fss	SSI1Fss	I2C3SDA	M0PWM7	M1PWM1	-	WT2CCP1	-	-	-	-	-
PD2	3	AIN13	SSI3Rx	SSI1Rx	-	M0FAULT0	-	-	WT3CCP0	USB0EPEN	-	-	-	-
PD3	4	AIN12	SSI3Tx	SSI1Tx	-	-	-	IDX0	WT3CCP1	USB0PFLT	-	-	-	-
PD4	141	AIN7	U6Rx	-	-	-	-	-	WT4CCP0	-	-	-	-	-
PD5	142	AIN6	U6Tx	-	-	-	-	-	WT4CCP1	-	-	-	-	-
PD6	143	AIN5	U2Rx	-	-	M0FAULT0	-	PhA0	WT5CCP0	-	-	-	-	-
PD7	144	AIN4	U2Tx	-	-	M0FAULT1	-	PhB0	WT5CCP1	NMI	-	-	-	-
PE0	15	AIN3	U7Rx	-	-	-	-	-	-	-	-	-	-	-
PE1	14	AIN2	U7Tx	-	-	-	-	-	-	-	-	-	-	-
PE2	13	AIN1	-	-	-	-	-	-	-	-	-	-	-	-
PE3	12	AIN0	-	-	-	-	-	-	-	-	-	-	-	-
PE4	139	AIN9	U5Rx	-	I2C2SCL	M0PWM4	M1PWM2	-	-	CAN0Rx	-	-	-	-
PE5	140	AIN8	U5Tx	-	I2C2SDA	M0PWM5	M1PWM3	-	-	CAN0Tx	-	-	-	-
PE6	133	AIN21	-	-	-	-	-	-	-	CAN1Rx	-	-	-	-
PE7	134	AIN20	U1RI	-	-	-	-	-	-	CAN1Tx	-	-	-	-
PF0	62	-	U1RTS	SSI1Rx	CAN0Rx	-	M1PWM4	PhA0	T0CCP0	NMI	C0o	TRD2	-	-
PF1	63	-	U1CTS	SSI1Tx	-	-	M1PWM5	PhB0	T0CCP1	-	C1o	TRD1	-	-
PF2	64	-	U1DCD	SSI1Ck	-	M0FAULT0	M1PWM6	-	T1CCP0	-	C2o	TRD0	-	-
PF3	65	-	U1DSR	SSI1Fss	CAN0Tx	M0FAULT1	M1PWM7	-	T1CCP1	-	-	TRCLK	-	-
PF4	61	-	U1DTR	-	-	M0FAULT2	M1FAULT0	IDX0	T2CCP0	USB0EPEN	-	TRD3	-	-
PF5	60	-	-	-	-	M0FAULT3	-	-	T2CCP1	USB0PFLT	-	-	-	-

表 23-5. GPIO 管脚和复用功能 (续)

IO	管脚	模拟功能	数字功能 (GPIOCTL PMCx 位域编码) <sup>a</sup>										
			1	2	3	4	5	6	7	8	9	14	15
PF6	59	-	-	-	I2C2SCL	-	-	-	T3CCP0	-	-	-	-
PF7	58	-	-	-	I2C2SDA	-	M1FAULT0	-	T3CCP1	-	-	-	-
PG0	55	-	-	-	I2C3SCL	-	M1FAULT1	PhA1	T4CCP0	-	-	-	-
PG1	54	-	-	-	I2C3SDA	-	M1FAULT2	PhB1	T4CCP1	-	-	-	-
PG2	53	-	-	-	I2C4SCL	M0FAULT1	M1PWM0	-	T5CCP0	-	-	-	-
PG3	52	-	-	-	I2C4SDA	M0FAULT2	M1PWM1	PhA1	T5CCP1	-	-	-	-
PG4	51	-	U2Rx	-	I2C1SCL	M0PWM4	M1PWM2	PhB1	WT0CCP0	USB0EPEN	-	-	-
PG5	50	-	U2Tx	-	I2C1SDA	M0PWM5	M1PWM3	IDX1	WT0CCP1	USB0PFLT	-	-	-
PG6	48	-	-	-	I2C5SCL	M0PWM6	-	-	WT1CCP0	-	-	-	-
PG7	47	-	-	-	I2C5SDA	M0PWM7	IDX1	-	WT1CCP1	-	-	-	-
PH0	32	-	-	SSI3Cik	-	M0PWM0	-	M0FAULT0	WT2CCP0	-	-	-	-
PH1	31	-	-	SSI3Fss	-	M0PWM1	IDX0	M0FAULT1	WT2CCP1	-	-	-	-
PH2	28	-	-	SSI3Rx	-	M0PWM2	-	M0FAULT2	WT5CCP0	-	-	-	-
PH3	27	-	-	SSI3Tx	-	M0PWM3	-	M0FAULT3	WT5CCP1	-	-	-	-
PH4	26	-	-	SSI2Cik	-	M0PWM4	PhA0	-	WT3CCP0	-	-	-	-
PH5	23	-	-	SSI2Fss	-	M0PWM5	PhB0	-	WT3CCP1	-	-	-	-
PH6	22	-	-	SSI2Rx	-	M0PWM6	-	-	WT4CCP0	-	-	-	-
PH7	21	-	-	SSI2Tx	-	M0PWM7	-	-	WT4CCP1	-	-	-	-
PJ0	120	-	U4Rx	-	-	-	-	-	T1CCP0	-	-	-	-
PJ1	121	-	U4Tx	-	-	-	-	-	T1CCP1	-	-	-	-
PJ2	122	-	U5Rx	-	-	-	IDX0	-	T2CCP0	-	-	-	-
PJ3	123	-	U5Tx	-	-	-	-	-	T2CCP1	-	-	-	-
PJ4	127	C2+	U6Rx	-	-	-	-	-	T3CCP0	-	-	-	-
PJ5	128	C2-	U6Tx	-	-	-	-	-	T3CCP1	-	-	-	-
PJ6	129	-	-	-	-	-	-	-	-	-	-	-	-
PJ7	130	-	-	-	-	-	-	-	-	-	-	-	-
PK0	16	AIN16	-	SSI3Cik	-	-	-	M1FAULT0	-	-	-	-	-
PK1	17	AIN17	-	SSI3Fss	-	-	-	M1FAULT1	-	-	-	-	-
PK2	18	AIN18	-	SSI3Rx	-	-	-	M1FAULT2	-	-	-	-	-
PK3	19	AIN19	-	SSI3Tx	-	-	-	M1FAULT3	-	-	-	-	-
PK4	112	-	U7Rx	-	-	-	-	M0FAULT0	RTCCLK	C0o	-	-	-
PK5	111	-	U7Tx	-	-	-	-	M0FAULT1	-	C1o	-	-	-
PK6	110	-	-	-	-	-	-	M0FAULT2	WT1CCP0	C2o	-	-	-
PK7	109	-	-	-	-	-	-	M0FAULT3	WT1CCP1	-	-	-	-
PL0	108	-	-	-	-	-	-	-	T0CCP0	WT0CCP0	-	-	-
PL1	107	-	-	-	-	-	-	-	T0CCP1	WT0CCP1	-	-	-
PL2	106	-	-	-	-	-	-	-	T1CCP0	WT1CCP0	-	-	-
PL3	105	-	-	-	-	-	-	-	T1CCP1	WT1CCP1	-	-	-
PL4	104	-	-	-	-	-	-	-	T2CCP0	WT2CCP0	-	-	-
PL5	103	-	-	-	-	-	-	-	T2CCP1	WT2CCP1	-	-	-
PL6	96	USB0DP	-	-	-	-	-	-	T3CCP0	WT3CCP0	-	-	-

表 23-5. GPIO 管脚和复用功能 (续)

IO	管脚	模拟功能	数字功能 ( GPIOCTL PMCx 位域编码 ) <sup>a</sup>											
			1	2	3	4	5	6	7	8	9	14	15	
PL7	95	USB0DM	-	-	-	-	-	-	-	T3CCP1	WT3CCP1	-	-	-
PM0	89	-	-	-	-	-	-	-	-	T4CCP0	WT4CCP0	-	-	-
PM1	88	-	-	-	-	-	-	-	-	T4CCP1	WT4CCP1	-	-	-
PM2	87	-	-	-	-	-	-	-	-	T5CCP0	WT5CCP0	-	-	-
PM3	86	-	-	-	-	-	-	-	-	T5CCP1	WT5CCP1	-	-	-
PM4	85	-	-	-	-	-	-	-	-	-	-	-	-	-
PM5	84	-	-	-	-	-	-	-	-	-	-	-	-	-
PM6	83	-	-	MOPWM4	-	-	-	-	-	WT0CCP0	-	-	-	-
PM7	82	-	-	MOPWM5	-	-	-	-	-	WT0CCP1	-	-	-	-
PN0	81	-	CAN0Rx	-	-	-	-	-	-	-	-	-	-	-
PN1	80	-	CAN0Tx	-	-	-	-	-	-	-	-	-	-	-
PN2	20	-	-	MOPWM6	-	-	-	-	-	WT2CCP0	-	-	-	-
PN3	119	-	-	MOPWM7	-	-	-	-	-	WT2CCP1	-	-	-	-
PN4	71	-	-	M1PWM4	-	-	-	-	-	WT3CCP0	-	-	-	-
PN5	70	-	-	M1PWM5	-	-	-	-	-	WT3CCP1	-	-	-	-
PN6	69	-	-	M1PWM6	-	-	-	-	-	WT4CCP0	-	-	-	-
PN7	68	-	-	M1PWM7	-	-	-	-	-	WT4CCP1	-	-	-	-
PP0	131	AIN23	MOPWM0	-	-	-	-	-	-	T4CCP0	-	-	-	-
PP1	132	AIN22	MOPWM1	-	-	-	-	-	-	T4CCP1	-	-	-	-
PP2	11	-	MOPWM2	-	-	-	-	-	-	T5CCP0	-	-	-	-

a. 带灰色阴影的数字信号是相应 GPIO 管脚的上电默认值。本器件不使用编码 10-13。

## 23.5 复用功能的可能的管脚赋值

表 23-6. 复用功能的可能的管脚赋值

# 种可能的赋值	复用功能	GPIO 功能
1	AIN0	PE3
	AIN1	PE2
	AIN10	PB4
	AIN11	PB5
	AIN12	PD3
	AIN13	PD2
	AIN14	PD1
	AIN15	PD0
	AIN16	PK0
	AIN17	PK1
	AIN18	PK2
	AIN19	PK3
	AIN2	PE1
	AIN20	PE7
	AIN21	PE6
	AIN22	PP1
	AIN23	PP0
	AIN3	PE0
	AIN4	PD7
	AIN5	PD6
	AIN6	PD5
	AIN7	PD4
	AIN8	PE5
	AIN9	PE4
	C0+	PC6
	C0-	PC7
	C1+	PC5
	C1-	PC4
	C2+	PJ4
	C2-	PJ5
	I2C0SCL	PB2
	I2C0SDA	PB3
	I2C4SCL	PG2
	I2C4SDA	PG3
I2C5SCL	PG6	
I2C5SDA	PG7	
M1FAULT3	PK3	
RTCCLK	PK4	
SSI0Cik	PA2	
SSI0Fss	PA3	

表 23-6. 复用功能的可能的管脚赋值 (续)

# 种可能的赋值	复用功能	GPIO 功能
	SSI0Rx	PA4
	SSI0Tx	PA5
	SSI2Rx	PH6
	SSI2Tx	PH7
	SWCLK	PC0
	SWDIO	PC1
	SWO	PC3
	TCK	PC0
	TDI	PC2
	TDO	PC3
	TMS	PC1
	TRCLK	PF3
	TRD0	PF2
	TRD1	PF1
	TRD2	PF0
	TRD3	PF4
	U0Rx	PA0
	U0Tx	PA1
	U1DCD	PF2
	U1DSR	PF3
	U1DTR	PF4
	U1RI	PE7
	U3Rx	PC6
	U3Tx	PC7
	USB0DM	PL7
	USB0DP	PL6
	USB0ID	PB0
	USB0VBUS	PB1

表 23-6. 复用功能的可能的管脚赋值 (续)

# 种可能的赋值	复用功能	GPIO 功能
2	C0o	PF0 PK4
	C1o	PF1 PK5
	C2o	PF2 PK6
	CAN1Rx	PA0 PE6
	CAN1Tx	PA1 PE7
	I2C1SCL	PA6 PG4
	I2C1SDA	PA7 PG5
	I2C2SCL	PE4 PF6
	I2C2SDA	PE5 PF7
	I2C3SCL	PD0 PG0
	I2C3SDA	PD1 PG1
	M0PWM0	PH0 PP0
	M0PWM1	PH1 PP1
	M0PWM3	PB5 PH3
	M1FAULT1	PG0 PK1
	M1FAULT2	PG1 PK2
	M1PWM0	PD0 PG2
	M1PWM1	PD1 PG3
	M1PWM4	PF0 PN4
	M1PWM5	PF1 PN5
	M1PWM6	PF2 PN6
	M1PWM7	PF3 PN7
	NMI	PD7 PF0
	SSI1Clk	PD0 PF2
	SSI1Fss	PD1 PF3
	SSI1Rx	PD2 PF0
	SSI1Tx	PD3 PF1
	SSI2Clk	PB4 PH4
	SSI2Fss	PB5 PH5
	T0CCP0	PF0 PL0
	T0CCP1	PF1 PL1
	U1CTS	PC5 PF1
	U1RTS	PC4 PF0
	U1Rx	PB0 PC4
	U1Tx	PB1 PC5
	U2Rx	PD6 PG4
	U2Tx	PD7 PG5
	U4Rx	PC4 PJ0
	U4Tx	PC5 PJ1
	U5Rx	PE4 PJ2
	U5Tx	PE5 PJ3
	U6Rx	PD4 PJ4

表 23-6. 复用功能的可能的管脚赋值 (续)

# 种可能的赋值	复用功能	GPIO 功能
	U6Tx	PD5 PJ5
	U7Rx	PE0 PK4
	U7Tx	PE1 PK5
3	IDX1	PC4 PG5 PG7
	M0FAULT3	PF5 PH3 PK7
	M0PWM2	PB4 PH2 PP2
	M1FAULT0	PF4 PF7 PK0
	M1PWM2	PA6 PE4 PG4
	M1PWM3	PA7 PE5 PG5
	PhA0	PD6 PF0 PH4
	PhA1	PC5 PG0 PG3
	PhB0	PD7 PF1 PH5
	PhB1	PC6 PG1 PG4
	SSI3Clk	PD0 PH0 PK0
	SSI3Fss	PD1 PH1 PK1
	SSI3Rx	PD2 PH2 PK2
	SSI3Tx	PD3 PH3 PK3
	T5CCP1	PC3 PG3 PM3
	WT5CCP0	PD6 PH2 PM2
	WT5CCP1	PD7 PH3 PM3

表 23-6. 复用功能的可能的管脚赋值 (续)

# 种可能的赋值	复用功能	GPIO 功能
4	CAN0Rx	PB4 PE4 PF0 PN0
	CAN0Tx	PB5 PE5 PF3 PN1
	IDX0	PD3 PF4 PH1 PJ2
	M0FAULT2	PF4 PG3 PH2 PK6
	M0PWM4	PE4 PG4 PH4 PM6
	M0PWM5	PE5 PG5 PH5 PM7
	T1CCP0	PB4 PF2 PJ0 PL2
	T1CCP1	PB5 PF3 PJ1 PL3
	T2CCP0	PB0 PF4 PJ2 PL4
	T2CCP1	PB1 PF5 PJ3 PL5
	T3CCP0	PB2 PF6 PJ4 PL6
	T3CCP1	PB3 PF7 PJ5 PL7
	T4CCP0	PC0 PG0 PM0 PP0
	T4CCP1	PC1 PG1 PM1 PP1
	T5CCP0	PC2 PG2 PM2 PP2
	USB0EPEN	PC6 PD2 PF4 PG4
	USB0PFLT	PC7 PD3 PF5 PG5
	WT0CCP0	PC4 PG4 PL0 PM6
	WT0CCP1	PC5 PG5 PL1 PM7
	WT1CCP0	PC6 PG6 PK6 PL2
	WT1CCP1	PC7 PG7 PK7 PL3
	WT2CCP0	PD0 PH0 PL4 PN2
	WT2CCP1	PD1 PH1 PL5 PN3
	WT3CCP0	PD2 PH4 PL6 PN4
WT3CCP1	PD3 PH5 PL7 PN5	
WT4CCP0	PD4 PH6 PM0 PN6	
WT4CCP1	PD5 PH7 PM1 PN7	
5	M0FAULT0	PD2 PD6 PF2 PH0 PK4
	M0FAULT1	PD7 PF3 PG2 PH1 PK5
	M0PWM6	PC4 PD0 PG6 PH6 PN2
	M0PWM7	PC5 PD1 PG7 PH7 PN3

## 23.6 未用信号的连接

对于 144 管脚 LQFP 封装中的设备，如果在特定的系统中并未用到某些信号，可按照表 23-7 在 1288 页对相应的功能信号进行处理。表中列出了两种选项：一般的处理方法以及推荐的处理方法，其中按照推荐的方法处理有助于降低功耗并改善 EMC 性能。假如系统中并未使用某个功能模块，并且已将其输入端接地，那么必须避免启用该模块的时钟（即将 RCGCx 寄存器中相应的位置位）。

表 23-7. 未用信号的连接 (144 管脚 LQFP)

功能	信号名称	管脚序号	一般的处理方法	推荐的处理方法
GPIO	所有未用的 GPIO	-	NC	GND



表 23-7. 未用信号的连接 ( 144 管脚 LQFP ) ( 续 )

功能	信号名称	管脚序号	一般的处理方法	推荐的处理方法
休眠	HIB	73	NC	NC
	VBAT	77	NC	VDD
	$\overline{\text{WAKE}}$	72	NC	GND
	XOSC0	74	NC	GND
	XOSC1	76	NC	NC
	GNDX	75	GND	GND
无连接	NC	-	NC	NC
系统控制	OSC0	92	NC	GND
	OSC1	93	NC	NC
	$\overline{\text{RST}}$	90	上拉，如图 5-1 在 205 页所示。	经电容接地。电容及接地位置应尽量靠近管脚
USB	USB0DM	95	NC	GND
	USB0DP	96	NC	GND

## 24 工作特性

表 24-1. 温度特性

特性	符号	值	单位
工作温度范围	$T_A$	-40 ~ +85	°C
贮存温度范围	$T_S$	-65 ~ +150	°C

表 24-2. 热学特性

特性	符号	值	单位
热阻 (结-环境) <sup>a</sup>	$\Theta_{JA}$	50 <sup>b</sup>	°C/W
结温度, -40 ~ +125 <sup>c</sup>	$T_J$	$T_A + (P \cdot \Theta_{JA})$	°C

a. 结点到环境的热敏电阻  $\Theta_{JA}$  由封装模拟器提供。

b. 初始值。

c. 功耗由温度决定。

表 24-3. ESD 最大额定速率<sup>a</sup>

参数名称	最小值	额定值	最大值	单位
$V_{ESDHBM}$	-	-	2.0	kV
$V_{ESDCDM}$	-	-	500	V

a. 所有 Stellaris<sup>®</sup> 器件均按照 JEDEC 标准进行 ESD 测试。

## 25 电气特性

### 25.1 Maximum Ratings

The maximum ratings are the limits to which the device can be subjected without permanently damaging the device. Device reliability may be adversely affected by exposure to absolute-maximum ratings for extended periods.

注意： The device is not guaranteed to operate properly at the maximum ratings.

表 25-1. Maximum Ratings

Parameter	Parameter Name <sup>a</sup>	Value		Unit
		Min	Max	
V <sub>DD</sub>	V <sub>DD</sub> supply voltage	0	4	V
V <sub>DDA</sub>	V <sub>DDA</sub> supply voltage	0	4	V
V <sub>BAT</sub>	V <sub>BAT</sub> battery supply voltage	0	4	V
V <sub>IN_GPIO</sub>	Input voltage on GPIOs, regardless of whether the microcontroller is powered <sup>bc</sup>	-0.3	5.5	V
	Input voltage for PB0 and PB1 when configured as GPIO	-0.3	V <sub>DD</sub> + 0.3	V
I <sub>GPIO MAX</sub>	Maximum current per output pin	-	25	mA
I <sub>NON</sub>	Maximum current into or out of a non-power, non-GPIO, non-XOSCn pin when the microcontroller is unpowered <sup>d</sup>	-	pending <sup>e</sup>	mA

a. Voltages are measured with respect to GND.

b. Applies to static and dynamic signals including overshoot.

c. Refer to 图 25-1 在 1292页 for a representation of the ESD protection on GPIOs.

d. Refer to 图 25-2 在 1292页 for a representation of the ESD protection on these pins.

e. Pending characterization.

**重要：** This device contains circuitry to protect the inputs against damage due to high-static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (see “未用信号的连接” 在 1288页).

图 25-1 在 1292页 shows a diagram of the ESD protection on GPIOs and the XOSC0 and XOSC1 pins.

图 25-1. ESD Protection on GPIOs and XOSCn Pins

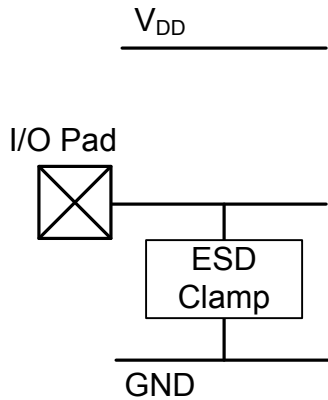
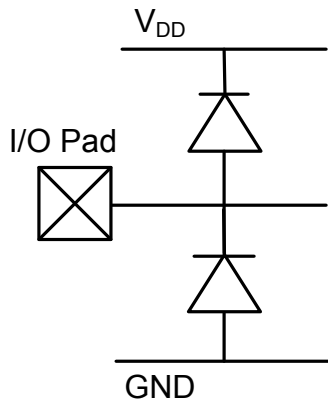


图 25-2 在 1292页 shows a diagram of the ESD protection on non-power, non-GPIO, and non-XOSCn pins.

图 25-2. ESD Protection on Non-Power, Non-GPIO, and Non-XOSCn Pins



## 25.2 Recommended Operating Conditions

For special high-current applications, the GPIO output buffers may be used with the following restrictions. With the GPIO pins configured as 8-mA output drivers, a total of four GPIO outputs may be used to sink current loads up to 18 mA each. At 18-mA sink current loading, the  $V_{OL}$  value is specified as 1.2 V. The high-current GPIO package pins must be selected such that there are only a maximum of two per side of the physical package with the total number of high-current GPIO outputs not exceeding four for the entire package.

表 25-2. Recommended DC Operating Conditions

Parameter	Parameter Name	Min	Nom	Max	Unit
$V_{DD}$	$V_{DD}$ supply voltage	2.97	3.3	3.63	V
$V_{DDA}$	$V_{DDA}$ supply voltage	2.97	3.3	3.63	V
$V_{DDC}$	$V_{DDC}$ supply voltage	1.08	1.2	1.32	V
$V_{IH}$	GPIO high-level input voltage	$0.65 * V_{DD}$	-	$V_{DD}$	V
$V_{IL}$	GPIO low-level input voltage	0	-	$0.35 * V_{DD}$	V

表 25-2. Recommended DC Operating Conditions (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
$V_{HYS}$	GPIO Input Hysteresis	0.2	-	-	V
$V_{OH}$	High-level output voltage	2.4	-	-	V
$V_{OL}$	Low-level output voltage	-	-	0.4	V
$I_{OH}$	High-level source current, $V_{OH}=2.4\text{ V}^a$				
	2-mA Drive	-2.0	-	-	mA
	4-mA Drive	-4.0	-	-	mA
	8-mA Drive	-8.0	-	-	mA
$I_{OL}$	Low-level sink current, $V_{OL}=0.4\text{ V}^a$				
	2-mA Drive	2.0	-	-	mA
	4-mA Drive	4.0	-	-	mA
	8-mA Drive	8.0	-	-	mA
	8-mA Drive, $V_{OL}=1.2\text{ V}$	18.0	-	-	mA

a.  $I_O$  specifications reflect the maximum current where the corresponding output voltage meets the  $V_{OH}/V_{OL}$  thresholds.  $I_O$  current can exceed these limits (subject to absolute maximum ratings).

表 25-3. GPIO Current Restrictions<sup>a</sup>

Parameter	Parameter Name	Min	Nom	Max	Unit
$I_{MAXL}$	Cumulative maximum GPIO current per side, left <sup>b</sup>	-	-	110	mA
$I_{MAXB}$	Cumulative maximum GPIO current per side, bottom <sup>b</sup>	-	-	105	mA
$I_{MAXR}$	Cumulative maximum GPIO current per side, right <sup>b</sup>	-	-	115	mA
$I_{MAXT}$	Cumulative maximum GPIO current per side, top <sup>b</sup>	-	-	135	mA

a. Based on design simulations, not tested in production.

b. Sum of sink and source current for GPIOs as shown in 表 25-4 在 1293页.

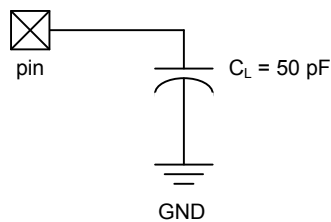
表 25-4. GPIO Package Side Assignments

Side	GPIOs
Left	PC[4-7], PD[0-3], PE[0-3], PH[0-7], PK[0-3], PN2, PP2
Bottom	PA[0-7], PF[0-7], PG[0-7], PN[4-7]
Right	PB[0-3], PL[0-7], PM[0-7], PN[0-1]
Top	PB[4-5], PC[0-3], PD[4-7], PE[4-7], PJ[0-7], PK[4-7], PN3, PP[0-1]

## 25.3 Load Conditions

Unless otherwise specified, the following conditions are true for all timing measurements.

图 25-3. Load Conditions



## 25.4 JTAG and Boundary Scan

表 25-5. JTAG Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
J1	$F_{TCK}$	TCK operational clock frequency <sup>a</sup>	0	-	10	MHz
J2	$T_{TCK}$	TCK operational clock period	100	-	-	ns
J3	$T_{TCK\_LOW}$	TCK clock Low time	-	$t_{TCK}/2$	-	ns
J4	$T_{TCK\_HIGH}$	TCK clock High time	-	$t_{TCK}/2$	-	ns
J5	$T_{TCK\_R}$	TCK rise time	0	-	10	ns
J6	$T_{TCK\_F}$	TCK fall time	0	-	10	ns
J7	$T_{TMS\_SU}$	TMS setup time to TCK rise	8	-	-	ns
J8	$T_{TMS\_HLD}$	TMS hold time from TCK rise	4	-	-	ns
J9	$T_{TDI\_SU}$	TDI setup time to TCK rise	18	-	-	ns
J10	$T_{TDI\_HLD}$	TDI hold time from TCK rise	4	-	-	ns
J11	$T_{TDO\_ZDV}$	TCK fall to Data Valid from High-Z, 2-mA drive	-	13	35	ns
		TCK fall to Data Valid from High-Z, 4-mA drive		9	26	ns
		TCK fall to Data Valid from High-Z, 8-mA drive		8	26	ns
		TCK fall to Data Valid from High-Z, 8-mA drive with slew rate control		10	29	ns
J12	$T_{TDO\_DV}$	TCK fall to Data Valid from Data Valid, 2-mA drive	-	14	20	ns
		TCK fall to Data Valid from Data Valid, 4-mA drive		10	26	ns
		TCK fall to Data Valid from Data Valid, 8-mA drive		8	21	ns
		TCK fall to Data Valid from Data Valid, 8-mA drive with slew rate control		10	26	ns
J13	$T_{TDO\_DVZ}$	TCK fall to High-Z from Data Valid, 2-mA drive	-	7	16	ns
		TCK fall to High-Z from Data Valid, 4-mA drive		7	16	ns
		TCK fall to High-Z from Data Valid, 8-mA drive		7	16	ns
		TCK fall to High-Z from Data Valid, 8-mA drive with slew rate control		8	19	ns

a. A ratio of at least 8:1 must be kept between the system clock and TCK.

图 25-4. JTAG Test Clock Input Timing

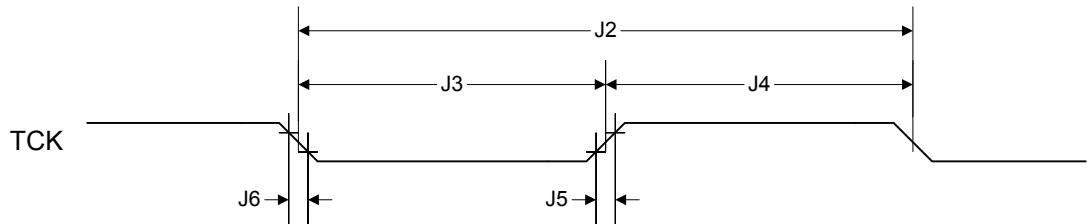
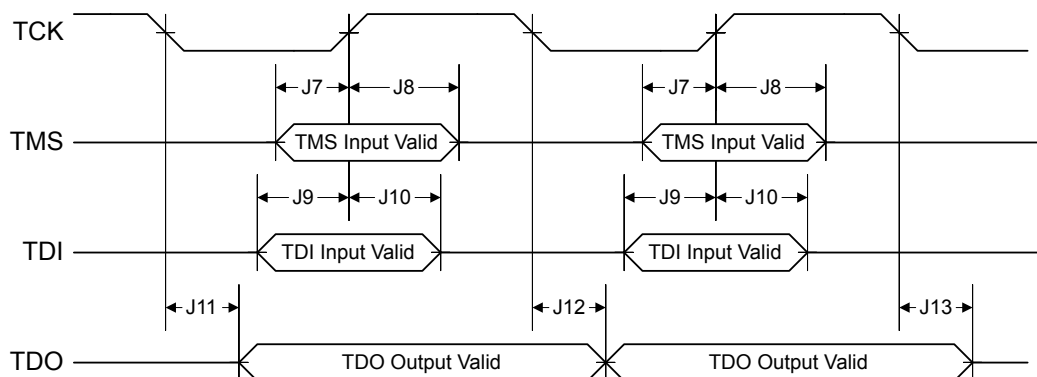


图 25-5. JTAG Test Access Port (TAP) Timing



## 25.5 Power and Brown-Out

表 25-6. Power Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
P1	$T_{VDDRISE}$	Supply voltage ( $V_{DD}$ ) rise time	-	-	$\infty$	ms
P2	$V_{TH}$	Power-On Reset threshold (rising edge)	2.0	2.31	2.6	V
		Power-On Reset threshold (falling edge)	2.0	2.27	2.6	V
P3	$V_{BTH}$	Brown-Out Reset threshold (rising edge)	2.8	2.92	3.1	V
		Brown-Out Reset threshold (falling edge)	2.8	2.90	3.1	V
P4	$T_{POR}$	Power-On Reset timeout	-	-	500	$\mu$ s
P5	$T_{BOR}$	Brown-Out Reset timeout	1	-	<sup>a</sup>	$\mu$ s

a. After the minimum timeout, BOR stays active as long as the power supply is below  $V_{BTH}$ . The voltage level is checked every clock cycle after the initial countdown completes. BOR is deasserted on the next clock cycle after the BOR condition goes away.

图 25-6. Power-On and Brown-Out Reset and Voltage Parameters

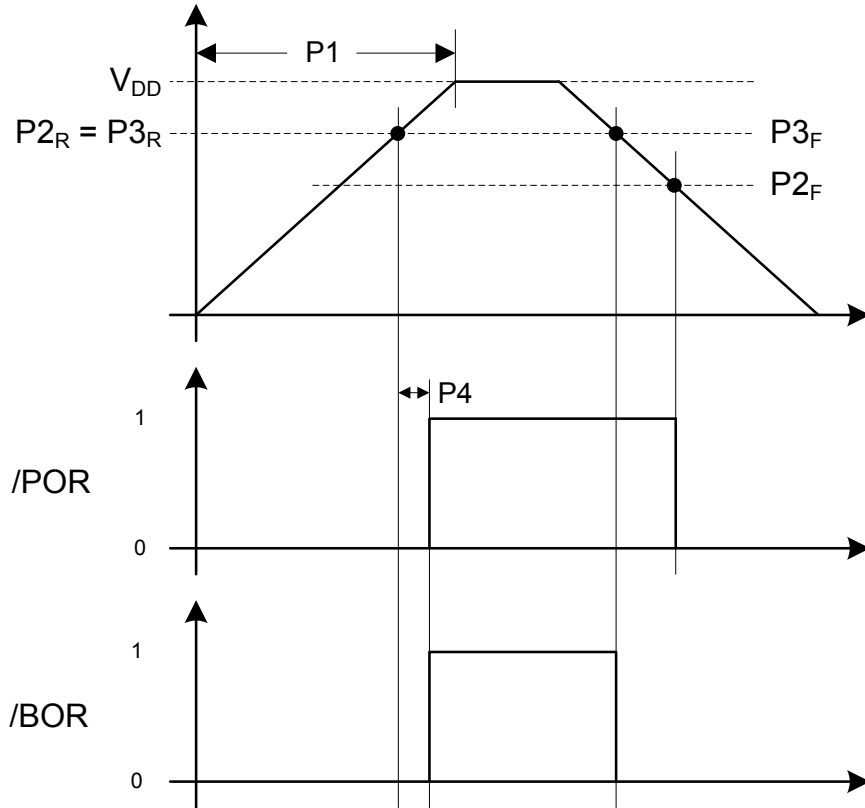
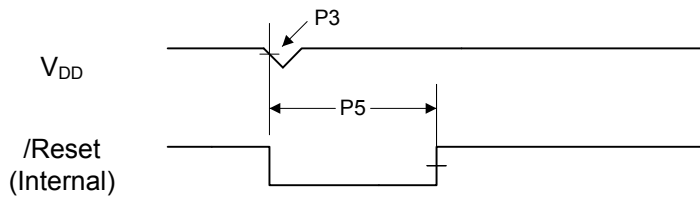


图 25-7. Brown-Out Reset Timing



## 25.6 Reset

表 25-7. Reset Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
R1	$T_{IRHWR}$	Internal reset timeout after hardware reset (RST pin) <sup>a</sup>	20	-	- <sup>b</sup>	$\mu\text{s}$
R2	$T_{IRSWR}$	Internal reset timeout after software-initiated system reset	-	1	-	$\mu\text{s}$
R3	$T_{IRWDR}$	Internal reset timeout after watchdog reset	-	1	-	$\mu\text{s}$
R4	$T_{IRMFR}$	Internal reset timeout after MOSC failure reset	-	1	-	$\mu\text{s}$



表 25-7. Reset Characteristics (续)

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
R5	$T_{MIN}$	Minimum $\overline{RST}$ pulse width	100	-	-	ns

- a. When in Deep-Sleep mode with the IOSC as the clock source, the internal reset timeout after a hardware reset is  $30 \mu s$  \* the Deep-sleep clock divider (DSDIVORIDE).
- b. After the minimum timeout, internal reset stays active as long as the  $\overline{RST}$  pin is asserted, and is released when the  $\overline{RST}$  pin is deasserted.

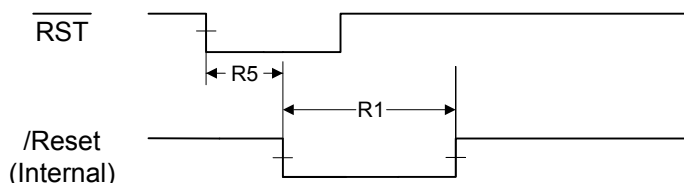
图 25-8. External Reset Timing ( $\overline{RST}$ )

图 25-9. Software Reset Timing

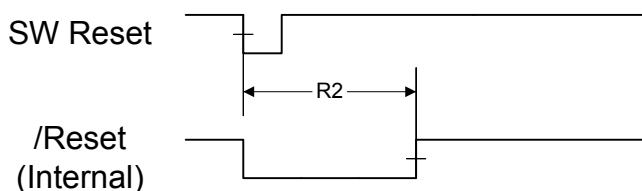


图 25-10. Watchdog Reset Timing

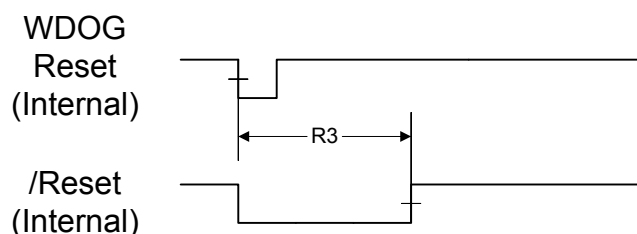
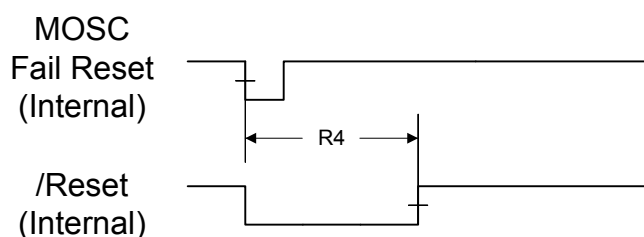


图 25-11. MOSC Failure Reset Timing



## 25.7 On-Chip Low Drop-Out (LDO) Regulator

表 25-8. LDO Regulator Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
$C_{LDO}$	External filter capacitor size for internal power supply <sup>a</sup>	2.5	-	4.0	$\mu F$

表 25-8. LDO Regulator Characteristics (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
ESR	Filter capacitor equivalent series resistance	10	-	100	mΩ
ESL	Filter capacitor equivalent series inductance	-	-	0.5	nH
V <sub>LDO</sub>	LDO output voltage	1.08	1.2	1.32	V
I <sub>INRUSH</sub>	Inrush current	50	-	250	mA

a. The capacitor should be connected as close as possible to pin 126.

## 25.8 Clocks

The following sections provide specifications on the various clock sources and mode.

### 25.8.1 PLL Specifications

The following tables provide specifications for using the PLL.

表 25-9. Phase Locked Loop (PLL) Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
F <sub>REF_XTAL</sub>	Crystal reference	5 <sup>a</sup>	-	25	MHz
F <sub>REF_EXT</sub>	External clock reference <sup>a</sup>	5 <sup>a</sup>	-	25	MHz
F <sub>PLL</sub>	PLL frequency <sup>b</sup>	-	400	-	MHz
T <sub>READY</sub>	PLL lock time, enabling the PLL	-	-	512 * (N+1) <sup>c</sup>	reference clocks <sup>d</sup>
	PLL lock time, changing the XTAL field in the RCC/RCC2 register or changing the OSCSRC between MOSC and PIOSC	-	-	128 * (N+1) <sup>c</sup>	reference clocks <sup>d</sup>

a. If the PLL is not used, the minimum input frequency can be 4 MHz.

b. PLL frequency is automatically calculated by the hardware based on the XTAL field of the RCC register.

c. N is the value in the N field in the PLLFREQ1 register.

d. A reference clock is the clock period of the crystal being used, which can be MOSC or PIOSC. For example, a 16-MHz crystal connected to MOSC yields a reference clock of 62.5 ns.

表 25-10 在 1298页 shows the actual frequency of the PLL based on the crystal frequency used (defined by the XTAL field in the RCC register).

表 25-10. Actual PLL Frequency

XTAL	Crystal Frequency (MHz)	MINT	MFRAC	Q	N	PLL Multiplier	PLL Frequency (MHz)	Error
0x09	5.0	0x50	0x0	0x0	0x0	80	400	-
0x0A	5.12	0x9C	0x100	0x0	0x1	156.25	400	-
0x0B	6.0	0xC8	0x0	0x0	0x2	200	400	-
0x0C	6.144	0xC3	0x140	0x0	0x2	195.3125	400	-
0x0D	7.3728	0xA2	0x30A	0x0	0x2	162.7598	399.9984	0.0004%
0x0E	8.0	0x32	0x0	0x0	0x0	50	400	-
0x0F	8.192	0xC3	0x140	0x0	0x3	195.3125	400	-
0x10	10.0	0x50	0x0	0x0	0x1	80	400	-
0x11	12.0	0xC8	0x0	0x0	0x5	200	400	-
0x12	12.288	0xC3	0x140	0x0	0x5	195.3125	400	-

表 25-10. Actual PLL Frequency (续)

XTAL	Crystal Frequency (MHz)	MINT	MFRAC	Q	N	PLL Multiplier	PLL Frequency (MHz)	Error
0x13	13.56	0xB0	0x3F6	0x0	0x5	176.9902	399.9979	0.0005%
0x14	14.318	0xC3	0x238	0x0	0x6	195.5547	399.9982	0.0005%
0x15	16.0	0x32	0x0	0x0	0x1	50	400	-
0x16	16.384	0xC3	0x140	0x0	0x7	195.3125	400	-
0x17	18	0xC8	0x0	0x0	0x8	200	400	-
0x18	20	0x50	0x0	0x0	0x3	80	400	-
0x19	24	0x32	0x0	0x0	0x2	50	400	-
0x1A	25	0x50	0x0	0x0	0x4	80	400	-

## 25.8.2 PIOSC Specifications

表 25-11. PIOSC Clock Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
F <sub>PIOSC25</sub>	Internal 16-MHz precision oscillator frequency variance, factory calibrated at 25 °C	-	±0.25%	±1%	-
F <sub>PIOSCT</sub>	Internal 16-MHz precision oscillator frequency variance, factory calibrated at 25 °C, across specified temperature range	-	-	±3%	-
F <sub>PIOSCUCAL</sub>	Internal 16-MHz precision oscillator frequency variance, user calibrated at a chosen temperature	-	±0.25%	±1%	-

## 25.8.3 Internal 30-kHz Oscillator Specifications

表 25-12. 30-kHz Clock Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
F <sub>IOSC30KHZ</sub>	Internal 30-KHz oscillator frequency	10	30	90	KHz

## 25.8.4 Hibernation Clock Source Specifications

表 25-13. HIB Oscillator Input Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
F <sub>HIBOSC</sub>	Parallel resonance frequency	-	32.768	-	KHz
C <sub>1</sub> , C <sub>2</sub>	External load capacitance on XOSC0, XOSC1 pins <sup>a</sup>	12	-	24	pF
C <sub>PKG</sub>	Device package stray shunt capacitance <sup>a</sup>	-	0.5	-	pF
C <sub>PCB</sub>	PCB stray shunt capacitance <sup>a</sup>	-	0.5	-	pF
C <sub>0</sub>	Crystal shunt capacitance <sup>a</sup>	-	3	-	pF
C <sub>SHUNT</sub>	Total shunt capacitance <sup>a</sup>	-	-	4	pF
ESR	Crystal effective series resistance, OSCDRV = 0 <sup>b</sup>	-	-	50	kΩ
	Crystal effective series resistance, OSCDRV = 1 <sup>b</sup>	-	-	75	kΩ
DL	Oscillator output drive level	-	-	0.25	μW
T <sub>START</sub>	Oscillator startup time, when using a crystal <sup>c</sup>	-	600	1500 <sup>d</sup>	ms

表 25-13. HIB Oscillator Input Characteristics (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
$V_{IH}^e$	CMOS input high level, when using an external oscillator with Supply > 3.3 V	2.64	-	-	V
	CMOS input high level, when using an external oscillator with $1.8\text{ V} \leq \text{Supply} \leq 3.3\text{ V}$	0.8 * Supply	-	-	V
$V_{IL}^e$	CMOS input low level, when using an external oscillator with $1.8\text{ V} \leq \text{Supply} \leq 3.63\text{ V}$	-	-	0.2 * Supply	V
$V_{HYS}^e$	CMOS input buffer hysteresis, when using an external oscillator with $1.8\text{ V} \leq \text{Supply} \leq 3.63\text{ V}$	360	960	1390	mV
DC <sub>HIBOSC_EXT</sub>	External clock reference duty cycle	30	-	70	%

- a. See information below table.
- b. Crystal ESR specified by crystal manufacturer.
- c. Oscillator startup time is specified from the time the oscillator is enabled to when it reaches a stable point of oscillation such that the internal clock is valid.
- d. Only valid for 3.3 V +/-10% supply conditions. Measured with OSCDRV bit set (high drive strength enabled, 24 pF).
- e. Specification is relative to the larger of  $V_{DD}$  or  $V_{BAT}$ .

The load capacitors added on the board,  $C_1$  and  $C_2$ , should be chosen such that the following equation is satisfied (see 表 25-13 在 1299页 for typical values).

- $C_L$  = load capacitance specified by crystal manufacturer
- $C_L = (C_1 * C_2) / (C_1 + C_2) + C_{PKG} + C_{PCB}$
- $C_{SHUNT} = C_{PKG} + C_{PCB} + C_0$  (total shunt capacitance seen across XOSC0, XOSC1)
- $C_{PKG}$ ,  $C_{PCB}$  as measured across the XOSC0, XOSC1 pins excluding the crystal
- Clear the OSCDRV bit in the Hibernation Control (HIBCTL) register for  $C_{1,2} \leq 18\text{ pF}$ ; set the OSCDRV bit for  $C_{1,2} > 18\text{ pF}$ .
- $C_0$  = Shunt capacitance of crystal specified by the crystal manufacturer

### 25.8.5 Main Oscillator Specifications

表 25-14. Main Oscillator Input Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
$F_{MOSC}$	Parallel resonance frequency	4 <sup>a</sup>	-	25	MHz
$C_1, C_2$	External load capacitance on OSC0, OSC1 pins <sup>b</sup>	12	-	24	pF
$C_{PKG}$	Device package stray shunt capacitance <sup>b</sup>	-	0.5	-	pF
$C_{PCB}$	PCB stray shunt capacitance <sup>b</sup>	-	0.5	-	pF
$C_0$	Crystal shunt capacitance <sup>bc</sup>	-	4	-	pF
$C_{SHUNT}$	Total shunt capacitance <sup>b</sup>	-	-	4	pF

表 25-14. Main Oscillator Input Characteristics (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
ESR	Crystal effective series resistance, 4 MHz <sup>dc</sup>	-	-	300	Ω
	Crystal effective series resistance, 6 MHz <sup>dc</sup>	-	-	200	Ω
	Crystal effective series resistance, 8 MHz <sup>dc</sup>	-	-	130	Ω
	Crystal effective series resistance, 12 MHz <sup>dc</sup>	-	-	120	Ω
	Crystal effective series resistance, 16 MHz <sup>dc</sup>	-	-	100	Ω
	Crystal effective series resistance, 25 MHz <sup>dc</sup>	-	-	50	Ω
DL	Oscillator output drive level <sup>e</sup>	-	OSC <sub>PWR</sub>	-	mW
T <sub>START</sub>	Oscillator startup time, when using a crystal <sup>f</sup>	-	-	18	ms
V <sub>IH</sub>	CMOS input high level, when using an external oscillator	0.65 * V <sub>DD</sub>	-	V <sub>DD</sub>	V
V <sub>IL</sub>	CMOS input low level, when using an external oscillator	GND	-	0.35 * V <sub>DD</sub>	V
V <sub>HYS</sub>	CMOS input buffer hysteresis, when using an external oscillator	150	-	-	mV
DC <sub>HIBOSC_EXT</sub>	External clock reference duty cycle	45	-	55	%

a. 5 MHz is the minimum when using the PLL.

b. See information below table.

c. Crystal vendors can be contacted to confirm these specifications are met for a specific crystal part number if the vendors generic crystal datasheet show limits outside of these specifications.

d. Crystal ESR specified by crystal manufacturer.

e.  $OSC_{PWR} = (2 * \pi * F_P * C_L * 2.5)^2 * ESR / 2$ . An estimation of the typical power delivered to the crystal is based on the  $C_L$ ,  $F_P$  and ESR parameters of the crystal in the circuit as calculated by the  $OSC_{PWR}$  equation. Ensure that the value calculated for  $OSC_{PWR}$  does not exceed the crystal's drive-level maximum.

f. Oscillator startup time is specified from the time the oscillator is enabled to when it reaches a stable point of oscillation such that the internal clock is valid.

The load capacitors added on the board,  $C_1$  and  $C_2$ , should be chosen such that the following equation is satisfied (see 表 25-14 在 1300页 for typical values and 表 25-15 在 1302页 for detailed crystal parameter information).

- $C_L$  = load capacitance specified by crystal manufacturer
- $C_L = (C_1 * C_2) / (C_1 + C_2) + C_{SHUNT}$
- $C_{SHUNT} = C_0 + C_{PKG} + C_{PCB}$  (total shunt capacitance seen across OSC0, OSC1 crystal inputs)
- $C_{PKG}$ ,  $C_{PCB}$  = the mutual caps as measured across the OSC0, OSC1 pins excluding the crystal.
- $C_0$  = Shunt capacitance of crystal specified by the crystal manufacturer

表 25-15. Crystal Parameters

MFG	MFG Part#	Holder	PKG Size (mm x mm)	Freq (MHz)	Crystal Spec	Crystal Parameters						Recommended Component Values			WC DI(uW)
						Typical Values			Max Values			CL1(pF)	CL2(pF)	Rs(K Ohm)	
						C0(pF)	C1(fF)	L1(mH)	ESR (Ohm)	Max DI (uW)	CI(pf)				
NDK	NX8045GB-4.000M-STD-CJL-5	NX8045GB	8 x 4.5	4	STD-CJL-5	1.00	2.70	598.10	300	500	8	12	12	0	132
NDK	NX8045GB-5.000M-STD-CSF-4	NX8045GB	8 x 4.5	5	STD-CJL-4	1.00	2.80	356.50	250	500	8	12	12	0	164
NDK	NX8045GB-6.000M-STD-CSF-4	NX8045GB	8 x 4.5	6	STD-CJL-4	1.30	4.10	173.20	250	500	8	12	12	0	214
NDK	NX8045GB-8.000M-STD-CSF-6	NX8045GB	8 x 4.5	8	STD-CSF-6	1.00	2.80	139.30	200	500	8	12	12	0	277
NDK	NX3225GA-12.000MHZ-STD-CRG-2	NX3225GA	3.2 x2.5	12	STD-CRG-2	0.70	2.20	81.00	100	200	8	12	12	2.5	147
NDK	NX5032GA-12.000MHZ-LN-CD-1	NX5032GA	5 x3.2	12	LN-CD-1	0.93	3.12	56.40	120	500	8	12	12	0	362
NDK	NX3225GA-16.000MHZ-STD-CRG-2	NX3225GA	3.2 x2.5	16	STD-CRG-2	1.00	2.90	33.90	80	200	8	12	12	2	188
NDK	NX5032GA-16.000MHZ-LN-CD-1	NX5032GA	5 x3.2	16	LN-CD-1	1.02	3.82	25.90	120	500	8	10	12	0	437
NDK	NX3225GA-25.000MHZ-STD-CRG-2	NX3225GA	3.2 x2.5	25	STD-CRG-2	1.10	4.70	8.70	50	200	8	12	12	2	181
auris	Q-25.000M-HC3225/4-F-30-30-E-12-TR	HC3225/4	3.2 x2.5	25	HC3225	1.58	5.01	8.34	50	500	12	16	16	1	331

表 25-16. Supported MOSC Crystal Frequencies<sup>a</sup>

Value	Crystal Frequency (MHz) Not Using the PLL	Crystal Frequency (MHz) Using the PLL
0x00-0x5	reserved	
0x06	4 MHz	reserved
0x07	4.096 MHz	reserved
0x08	4.9152 MHz	reserved
0x09	5 MHz (USB)	

表 25-16. Supported MOSC Crystal Frequencies (续)

Value	Crystal Frequency (MHz) Not Using the PLL	Crystal Frequency (MHz) Using the PLL
0x0A		5.12 MHz
0x0B		6 MHz (USB)
0x0C		6.144 MHz
0x0D		7.3728 MHz
0x0E		8 MHz (USB)
0x0F		8.192 MHz
0x10		10.0 MHz (USB)
0x11		12.0 MHz (USB)
0x12		12.288 MHz
0x13		13.56 MHz
0x14		14.31818 MHz
0x15		16.0 MHz (reset value)(USB)
0x16		16.384 MHz
0x17		18.0 MHz (USB)
0x18		20.0 MHz (USB)
0x19		24.0 MHz (USB)
0x1A		25.0 MHz (USB)

a. Frequencies that may be used with the USB interface are indicated in the table.

## 25.8.6 System Clock Specification with ADC Operation

表 25-17. System Clock Characteristics with ADC Operation

Parameter	Parameter Name	Min	Nom	Max	Unit
F <sub>sysadc</sub>	System clock frequency when the ADC module is operating (when PLL is bypassed). <sup>a</sup>	15.9952	16	16.0048	MHz

a. Clock frequency (plus jitter) must be stable inside specified range. ADC can be clocked from the PLL, directly from an external clock source, or from the PIOSC, as long as frequency absolute precision is inside specified range.

## 25.8.7 System Clock Specification with USB Operation

表 25-18. System Clock Characteristics with USB Operation

Parameter	Parameter Name	Min	Nom	Max	Unit
F <sub>sysusb</sub>	System clock frequency when the USB module is operating (note that MOSC must be the clock source, either with or without using the PLL)	20	-	-	MHz

## 25.9 Sleep Modes

表 25-19. Sleep Modes AC Characteristics<sup>a</sup>

Parameter No	Parameter	Parameter Name	Min	Nom	Max	Unit
D1	T <sub>WAKE_S</sub>	Time to wake from interrupt in sleep mode <sup>b</sup>	-	-	2	system clocks
	T <sub>WAKE_DS</sub>	Time to wake from interrupt in deep-sleep mode, using PIOSC for both Run mode and Deep-sleep mode <sup>b c</sup>	-	1.25	-	μs
		Time to wake from interrupt in deep-sleep mode, using PIOSC for Run mode and IOSC for Deep-sleep mode mode <sup>b c</sup>	-	350	-	μs
D2	T <sub>WAKE_PLL_DS</sub>	Time to wake from interrupt in deep-sleep mode when using the PLL <sup>b</sup>	-	-	T <sub>READY</sub>	ms

a. Values in this table assume the IOSC is the clock source during sleep or deep-sleep mode.

b. Specified from registering the interrupt to first instruction.

c. If the main oscillator is used for run mode, add the main oscillator startup time, T<sub>START</sub>.

## 25.10 Hibernation Module

The Hibernation module requires special system implementation considerations because it is intended to power down all other sections of its host device, refer to “休眠模块” 在 475页.

表 25-20. Hibernation Module Battery Characteristics

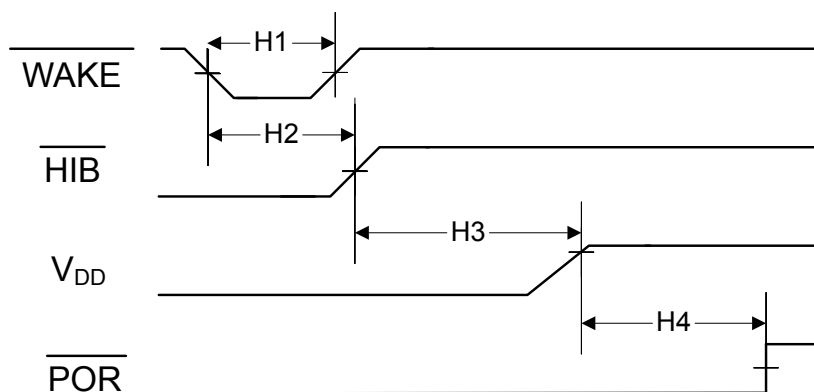
Parameter	Parameter Name	Min	Nominal	Max	Unit
V <sub>BAT</sub>	Battery supply voltage	1.8	3.0	3.6	V
V <sub>LOWBAT</sub>	Low battery detect voltage, VBATSEL=0x0	1.8	1.9	2.0	V
	Low battery detect voltage, VBATSEL=0x1	2.0	2.1	2.2	V
	Low battery detect voltage, VBATSEL=0x2	2.2	2.3	2.4	V
	Low battery detect voltage, VBATSEL=0x3	2.4	2.5	2.6	V

表 25-21. Hibernation Module AC Characteristics

Parameter No	Parameter	Parameter Name	Min	Nom	Max	Unit
H1	T <sub>WAKE</sub>	WAKE assertion time	100	-	-	ns
H2	T <sub>WAKE_TO_HIB</sub>	WAKE assert to HIB desassert (wake up time)	-	-	1	hibernation clock period
H3	T <sub>VDD_RAMP</sub>	V <sub>DD</sub> ramp to 3.0 V	-	Depends on characteristics of power supply	-	μs
H4	T <sub>VDD_CODE</sub>	V <sub>DD</sub> at 3.0 V to internal POR deassert; first instruction executes	-	-	500	μs
H5	DC <sub>RTCCLK</sub>	Duty cycle for RTCCLK output signal, when using a 32.768-kHz crystal	40	-	60	%
		Duty cycle for RTCCLK output signal, when using a 32.768-kHz single-ended clock source	30	-	70	%



图 25-12. Hibernation Module Timing



## 25.11 Flash Memory

表 25-22. Flash Memory Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
PE <sub>CYC</sub>	Number of guaranteed program/erase cycles before failure <sup>a</sup>	100,000	-	-	cycles
T <sub>RET</sub>	Data retention, -40°C to +85°C	10	-	-	years
T <sub>PROG64</sub>	Program time for double-word-aligned 64 bits of data <sup>b</sup>	30	50	300	μs
T <sub>ERASE</sub>	Page erase time, <1k cycles endurance	2	25	-	ms
	Page erase time, 100k cycles endurance	-	-	900	ms
T <sub>ME</sub>	Mass erase time, <1k cycles endurance	70	200	-	ms
	Mass erase time, 100k cycles endurance	-	-	900	ms

a. A program/erase cycle is defined as switching the bits from 1-> 0 -> 1.

b. If programming fewer than 64 bits of data, the programming time is the same. For example, if only 32 bits of data need to be programmed, the other 32 bits are masked off.

表 25-23. EEPROM Characteristics<sup>a</sup>

Parameter	Parameter Name	Min	Nom	Max	Unit
EPE <sub>CYC</sub>	Number of guaranteed mass program/erase cycles of a single word before failure <sup>b</sup>	500,000	-	-	cycles
ET <sub>RET</sub>	Data retention, -40°C to +85°C	10	-	-	years

表 25-23. EEPROM Characteristics (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
ET <sub>PROG</sub>	Program time for 32 bits of data - space available	-	110	600	μs
	Program time for 32 bits of data - requires a copy to the copy buffer, copy buffer has space and less than 10% of EEPROM endurance used	-	30	-	ms
	Program time for 32 bits of data - requires a copy to the copy buffer, copy buffer has space and greater than 90% of EEPROM endurance used	-	-	900	ms
	Program time for 32 bits of data - requires a copy to the copy buffer, copy buffer requires an erase and less than 10% of EEPROM endurance used	-	60	-	ms
	Program time for 32 bits of data - requires a copy to the copy buffer, copy buffer requires an erase and greater than 90% of EEPROM endurance used	-	-	1800	ms
ET <sub>READ</sub>	Read access time	-	4	-	system clocks
ET <sub>ME</sub>	Mass erase time, <1k cycles endurance	70	200	-	ms
	Mass erase time, 100k cycles endurance	-	-	900	ms

a. Because the EEPROM operates as a background task and does not prevent the CPU from executing from Flash memory, the operation will complete within the maximum time specified provided the EEPROM operation is not stalled by a Flash memory program or erase operation.

b. A program/erase cycle is defined as switching the bits from 1-> 0 -> 1.

## 25.12 Input/Output Characteristics

注意: All GPIOs are 5-V tolerant, except PB0 and PB1. See “信号描述” 在 597页 for more information on GPIO configuration.

表 25-24. GPIO Module Characteristics<sup>ab</sup>

Parameter	Parameter Name	Min	Nom	Max	Unit
R <sub>GPIOPU</sub>	GPIO internal pull-up resistor	13	20	30	kΩ
R <sub>GPIOPD</sub>	GPIO internal pull-down resistor	13	20	35	kΩ
I <sub>LKG+</sub>	GPIO input leakage current, 0 V ≤ V <sub>IN</sub> ≤ V <sub>DD</sub> GPIO pins <sup>c</sup>	-	-	1.0	μA
	GPIO input leakage current, 0 V < V <sub>IN</sub> ≤ V <sub>DD</sub> , GPIO pins configured as ADC or analog comparator inputs	-	-	2.0	μA
	GPIO input leakage current, V <sub>DD</sub> < V <sub>IN</sub> ≤ 4.0 V <sup>cd</sup>	-	-	700	μA
	GPIO input leakage current, 4.0 V < V <sub>IN</sub> ≤ 5.5 V <sup>ce</sup>	-	-	100	μA
I <sub>LKG-</sub>	GPIO input leakage current, -0.3 V ≤ V <sub>IN</sub> < 0 V <sup>c</sup>	-	-	10	μA
	GPIO input leakage current, V <sub>IN</sub> < -0.3 V <sup>c</sup>	-	-	- <sup>f</sup>	μA
I <sub>INJ+</sub>	DC injection current, V <sub>DD</sub> < V <sub>IN</sub> ≤ 5.5 V <sup>g</sup>	-	-	I <sub>LKG+</sub>	μA
I <sub>INJ-</sub>	DC injection current, V <sub>IN</sub> ≤ 0 V	-	-	0.5	mA
T <sub>GPIOR</sub>	GPIO rise time, 2-mA drive <sup>h</sup>	-	14.2	16.1	ns
	GPIO rise time, 4-mA drive <sup>h</sup>	-	11.9	15.5	ns
	GPIO rise time, 8-mA drive <sup>h</sup>	-	8.1	11.2	ns
	GPIO rise time, 8-mA drive with slew rate control <sup>h</sup>	-	9.5	11.8	ns

表 25-24. GPIO Module Characteristics (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
T <sub>GPIOF</sub>	GPIO fall time, 2-mA drive <sup>i</sup>	-	25.2	29.4	ns
	GPIO fall time, 4-mA drive <sup>i</sup>		13.3	16.8	ns
	GPIO fall time, 8-mA drive <sup>i</sup>		8.6	11.2	ns
	GPIO fall time, 8-mA drive with slew rate control <sup>i</sup>		11.3	12.9	ns

a. V<sub>DD</sub> must be within the range specified in 表 25-2 在 1292页.

b. Leakage and Injection current characteristics specified in this table also apply to XOSC0 and XOSC1 inputs.

c. The leakage current is measured with V<sub>IN</sub> applied to the corresponding pin(s). The leakage of digital port pins is measured individually. The port pin is configured as an input and the pullup/pulldown resistor is disabled.

d. To protect internal circuitry from over-voltage, the GPIOs have an internal voltage clamp that limits internal swings to V<sub>DD</sub> without affecting swing at the I/O pad. This internal clamp starts turning on while V<sub>DD</sub> < V<sub>IN</sub> < 4.0 V and causes a somewhat larger (but bounded) current draw. To save power, static input voltages between V<sub>DD</sub> and 4 V should be avoided.

e. Leakage current above maximum voltage (V<sub>IN</sub> = 5.5V) is not guaranteed, this condition is not allowed and can result in permanent damage to the device.

f. In this case, I<sub>LKG-</sub> is unbounded and must be limited to I<sub>INJ-</sub> using an external resistor.

g. Current injection is internally bounded for GPIOs, and maximum current into the pin is given by ILKG+ for V<sub>DD</sub> < V<sub>IN</sub> < 5.5 V.

h. Time measured from 20% to 80% of V<sub>DD</sub>.

i. Time measured from 80% to 20% of V<sub>DD</sub>.

## 25.13 Analog-to-Digital Converter (ADC)

表 25-25. ADC Electrical Characteristics<sup>a</sup>

Parameter	Parameter Name	Min	Nom	Max	Unit
POWER SUPPLY REQUIREMENTS					
V <sub>DDA</sub>	ADC supply voltage	2.97	3.3	3.63	V
G <sub>NDA</sub>	ADC ground voltage	-	0	-	V
V <sub>DDA</sub> / G <sub>NDA</sub> VOLTAGE REFERENCE					
C <sub>REF</sub>	Voltage reference decoupling capacitance	-	1.0 // 0.01 <sup>b</sup>	-	μF
EXTERNAL VOLTAGE REFERENCE INPUT					
V <sub>REF+</sub>	Positive external voltage reference for ADC, when VREF field in the ADCCTL register is not 0x0 <sup>c</sup>	2.4	V <sub>DDA</sub>	V <sub>DDA</sub>	V
V <sub>REF-</sub>	Negative external voltage reference for ADC, when VREF field in the ADCCTL register is not 0x0 <sup>c</sup>	G <sub>NDA</sub>	G <sub>NDA</sub>	0.3	V
I <sub>VREF</sub>	Current on VREF+ input, using external V <sub>REF+</sub> = 3.3 V	-	330.5	440	μA
I <sub>LVREF</sub>	DC leakage current on VREF+ input when external VREF disabled	-	-	2.0	μA
C <sub>REF</sub>	External reference decoupling capacitance <sup>c</sup>	-	1.0 // 0.01 <sup>d</sup>	-	μF
ANALOG INPUT					

表 25-25. ADC Electrical Characteristics (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
$V_{ADCIN}$	Single-ended, full-scale analog input voltage, internal reference <sup>ef</sup>	0	-	$V_{DDA}$	V
	Differential, full-scale analog input voltage, internal reference <sup>eg</sup>	$-V_{DDA}$	-	$V_{VDDA}$	V
	Single-ended, full-scale analog input voltage, external reference <sup>cf</sup>	$V_{REFA-}$	-	$V_{REFA+}$	V
	Differential, full-scale analog input voltage, external reference <sup>ch</sup>	$-(V_{REFA+} - V_{REFA-})$	-	$V_{REFA+} - V_{REFA-}$	V
$V_{INCM}$	Input common mode voltage, differential mode <sup>l</sup>	-	-	$(V_{REFP} + V_{REFN}) / 2$ $\pm 25$	mV
$I_L$	ADC input leakage current <sup>l</sup>	-	-	2.0	$\mu$ A
$R_{ADC}$	ADC equivalent input resistance <sup>l</sup>	-	-	2.5	k $\Omega$
$C_{ADC}$	ADC equivalent input capacitance <sup>l</sup>	-	-	10	pF
$R_S$	Analog source resistance <sup>l</sup>	-	-	500	$\Omega$
SAMPLING DYNAMICS					
$F_{ADC}$	ADC conversion clock frequency <sup>k</sup>	-	16	-	MHz
$F_{CONV}$	ADC conversion rate	1			MSPS
$T_S$	ADC sample time	-	250	-	ns
$T_C$	ADC conversion time	1			$\mu$ s
$T_{LT}$	Latency from trigger to start of conversion	-	2	-	ADC clocks
SYSTEM PERFORMANCE when using external reference <sup>lm</sup>					
N	Resolution	12			bits
INL	Integral nonlinearity error, over full input range	-	$\pm 1.5$	$\pm 3.0$	LSB
DNL	Differential nonlinearity error, over full input range	-	$\pm 0.8$	$\pm 1.0$	LSB
$E_O$	Offset error	-	$\pm 1.0$	$\pm 3.0$	LSB
$E_G$	Gain error <sup>n</sup>	-	$\pm 2.0$	$\pm 3.0$	LSB
$E_T$	Total unadjusted error, over full input range <sup>o</sup>	-	$\pm 2.5$	$\pm 4.0$	LSB
SYSTEM PERFORMANCE when using internal reference					
N	Resolution	12			bits
INL	Integral nonlinearity error, over full input range	-	$\pm 1.5$	$\pm 3.0$	LSB
DNL	Differential nonlinearity error, over full input range	-	$\pm 0.8$	$\pm 1.0$	LSB
$E_O$	Offset error	-	$\pm 5.0$	$\pm 15.0$	LSB
$E_G$	Gain error <sup>n</sup>	-	$\pm 10.0$	$\pm 30.0$	LSB
$E_T$	Total unadjusted error, over full input range <sup>o</sup>	-	$\pm 10.0$	$\pm 30.0$	LSB
TEMPERATURE SENSOR					
$V_{TSENS}$	Temperature sensor voltage, junction temperature 25 °C	-	1.633	-	V
$S_{TSENS}$	Temperature sensor slope, ambient temperature -40 °C to 85 °C	-	-13.3	-	mV/°C

表 25-25. ADC Electrical Characteristics (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
$E_{TSENS}$	Temperature sensor accuracy, ambient temperature $-40^{\circ}\text{C}$ to $85^{\circ}\text{C}^{\text{p}}$	-	-	$\pm 5$	$^{\circ}\text{C}$

a. At ambient temperature =  $-40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ ,  $V_{REF+} = 3.3\text{V}$ ,  $F_{ADC} = 16\text{MHz}$  unless otherwise noted.

b. Two capacitors in parallel.

c. Assumes external filtering network between  $V_{REF+}$  and  $V_{REF-}$  as shown in 图 25-13 在 1309页. External reference noise level must be under 12bit ( $-74\text{dB}$ ) of Full Scale input, over input bandwidth, measured at  $V_{REF+} - V_{REF-}$ .

d. Two capacitors in parallel.

e. Internal reference is connected directly between  $V_{DDA}$  and  $GND_A$  ( $V_{REFi} = V_{DDA} - GND_A$ ). In this mode,  $E_O$ ,  $E_G$ ,  $E_T$ , and dynamic specifications are adversely affected due to internal voltage drop and noise on  $V_{DDA}$  and  $GND_A$ . Internal reference voltage is selected when  $V_{REF}$  field in the  $ADCCTL$  register is  $0x0$ .

f.  $V_{ADCIN} = V_{INP} - V_{INN}$

g. With signal common mode as  $V_{DDA}/2$ .

h. With signal common mode as  $(V_{REF+} + V_{REF-})/2$ .

i. This parameter is defined as the average of the differential inputs.

j. As shown in 图 25-14 在 1310页,  $R_{ADC}$  is the total equivalent resistance in the input line all the way up to the sampling node at the input of the ADC.

k. See "System Clock Specification with ADC Operation" 在 1303页 for full ADC clock frequency specification.

l. Low noise environment is assumed in order to obtain values close to spec. Board must have good ground isolation between analog and digital grounds, a clean reference voltage is assumed, and input signal must be bandlimited to Nyquist bandwidth. No anti-aliasing filter is provided internally.

m. ADC static measurements taken by averaging over several samples. At least 20-sample averaging is assumed to obtain expected typical or maximum spec values.

n. Gain error is measured at max code after compensating for offset. Gain error is equivalent to "Full Scale Error." It can be given in % of slope error, or in LSB, as done here.

o. Total Unadjusted Error is the maximum error at any one code versus the ideal ADC curve. It includes all other errors (offset error, gain error and INL) at any given ADC code.

p. Note that this parameter does not include ADC error.

图 25-13. ADC External Reference Filtering

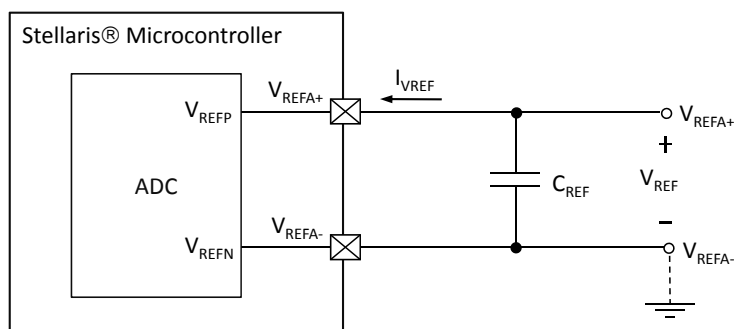
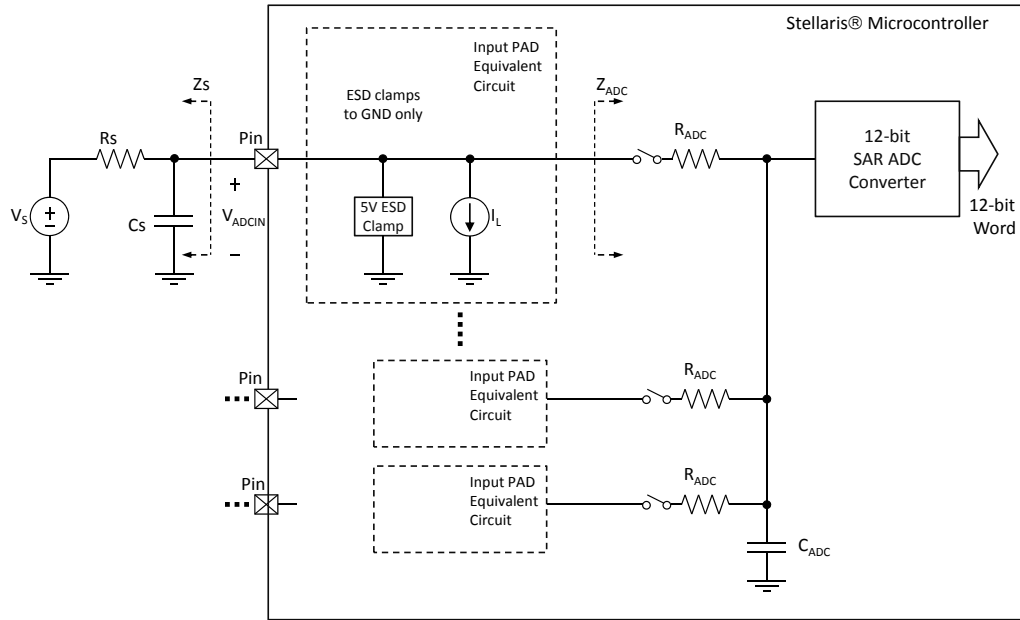


图 25-14. ADC Input Equivalency Diagram



## 25.14 Synchronous Serial Interface (SSI)

表 25-26. SSI Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
S1	$T_{CLK\_PER}$	SSIClk cycle time <sup>a</sup>	40	-	-	ns
S2	$T_{CLK\_HIGH}$	SSIClk high time	-	0.5	-	t clk_per
S3	$T_{CLK\_LOW}$	SSIClk low time	-	0.5	-	t clk_per
S4	$T_{CLKR}$	SSIClk rise time <sup>b</sup>	-	8.1	11.2	ns
S5	$T_{CLKF}$	SSIClk fall time <sup>b</sup>	-	8.6	11.2	ns
S6	$T_{DMD}$	Data from master valid delay time	0	-	1	system clocks
S7	$T_{DMS}$	Data from master setup time	1	-	-	system clocks
S8	$T_{DMH}$	Data from master hold time	2	-	-	system clocks
S9	$T_{DSS}$	Data from slave setup time	1	-	-	system clocks
S10	$T_{DSH}$	Data from slave hold time	2	-	-	system clocks

a. In master mode, the system clock must be at least twice as fast as the SSIClk; in slave mode, the system clock must be at least 6 times faster than the SSIClk.

b. Note that the delays shown are using 8-mA drive strength.

图 25-15. SSI Timing for TI Frame Format (FRF=01), Single Transfer Timing Measurement

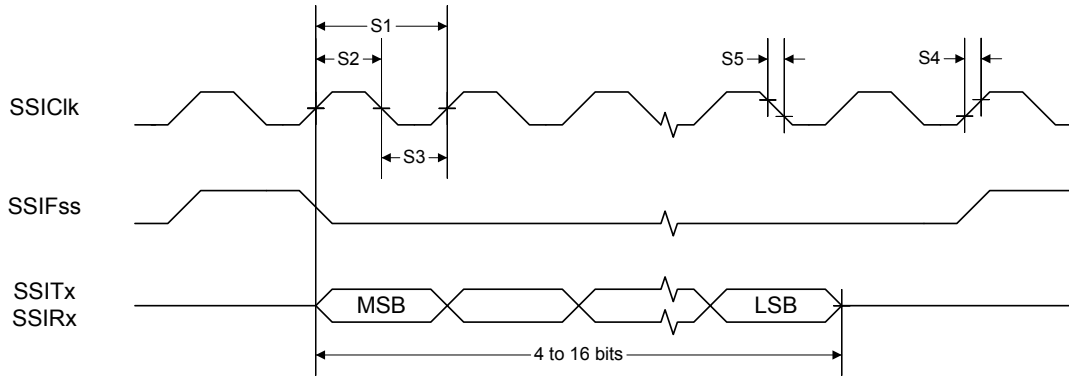


图 25-16. SSI Timing for MICROWIRE Frame Format (FRF=10), Single Transfer

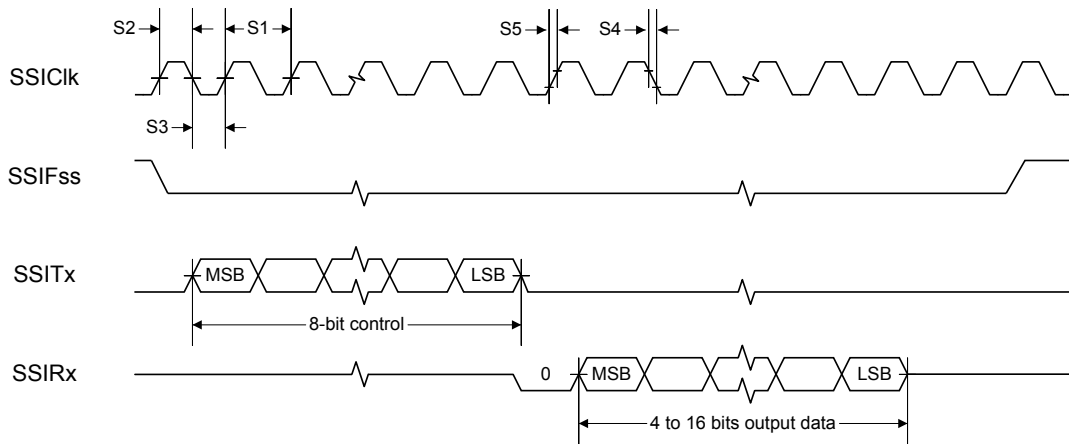
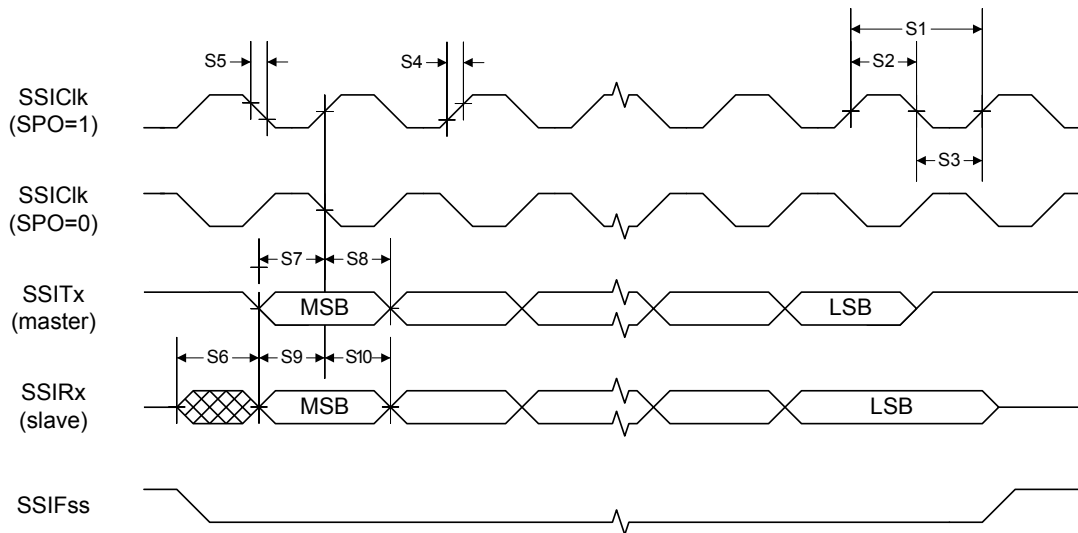


图 25-17. SSI Timing for SPI Frame Format (FRF=00), with SPH=1



## 25.15 Inter-Integrated Circuit (I<sup>2</sup>C) Interface

表 25-27. I<sup>2</sup>C Characteristics

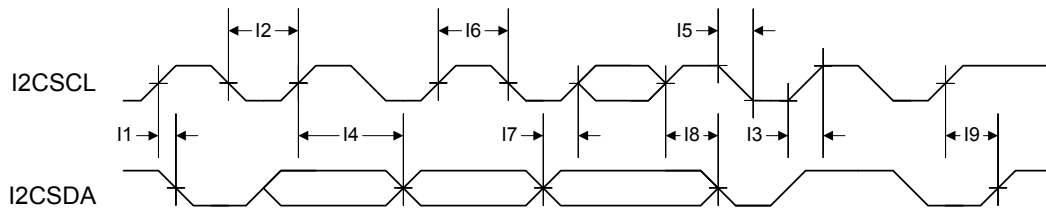
Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
11 <sup>a</sup>	T <sub>SCH</sub>	Start condition hold time	36	-	-	system clocks
12 <sup>a</sup>	T <sub>LP</sub>	Clock Low period	36	-	-	system clocks
13 <sup>b</sup>	T <sub>SRT</sub>	I2CSCL/I2CSDA rise time (V <sub>IL</sub> =0.5 V to V <sub>IH</sub> =2.4 V)	-	-	(see note b)	ns
14 <sup>a</sup>	T <sub>DH</sub>	Data hold time	2	-	-	system clocks
15 <sup>c</sup>	T <sub>SFT</sub>	I2CSCL/I2CSDA fall time (V <sub>IH</sub> =2.4 V to V <sub>IL</sub> =0.5 V)	-	9	10	ns
16 <sup>a</sup>	T <sub>HT</sub>	Clock High time	24	-	-	system clocks
17 <sup>a</sup>	T <sub>DS</sub>	Data setup time	18	-	-	system clocks
18 <sup>a</sup>	T <sub>SCSR</sub>	Start condition setup time (for repeated start condition only)	36	-	-	system clocks
19 <sup>a</sup>	T <sub>SCS</sub>	Stop condition setup time	24	-	-	system clocks

a. Values depend on the value programmed into the TPR bit in the I<sup>2</sup>C Master Timer Period (I2CMTPR) register; a TPR programmed for the maximum I2CSCL frequency (TPR=0x2) results in a minimum output timing as shown in the table above. The I<sup>2</sup>C interface is designed to scale the actual data transition time to move it to the middle of the I2CSCL Low period. The actual position is affected by the value programmed into the TPR; however, the numbers given in the above values are minimum values.

b. Because I2CSCL and I2CSDA operate as open-drain-type signals, which the controller can only actively drive Low, the time I2CSCL or I2CSDA takes to reach a high level depends on external signal capacitance and pull-up resistor values.

c. Specified at a nominal 50 pF load.

图 25-18. I<sup>2</sup>C Timing



## 25.16 Universal Serial Bus (USB) Controller

The Stellaris<sup>®</sup> USB controller electrical specifications are compliant with the “Universal Serial Bus Specification Rev. 2.0” (full-speed and low-speed support) and the “On-The-Go Supplement to the USB 2.0 Specification Rev. 1.0”. Some components of the USB system are integrated within the LM4F232H5QD microcontroller and specific to the Stellaris microcontroller design.

## 25.17 Analog Comparator

表 25-28. Analog Comparator Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
V <sub>INP</sub> , V <sub>INN</sub>	Input voltage range	GNDA	-	V <sub>DDA</sub>	V
V <sub>CM</sub>	Input common mode voltage range	GNDA	-	V <sub>DDA</sub>	V



表 25-28. Analog Comparator Characteristics (续)

Parameter	Parameter Name	Min	Nom	Max	Unit
V <sub>OS</sub>	Input offset voltage	-	±10	±25 <sup>a</sup>	mV
I <sub>INP</sub> , I <sub>INN</sub>	Input leakage current over full voltage range	-	-	2.0	µA
C <sub>MRR</sub>	Common mode rejection ratio	-	50	-	dB
T <sub>RT</sub>	Response time	-	-	1.0 <sup>b</sup>	µs
T <sub>MC</sub>	Comparator mode change to Output Valid	-	-	10	µs

a. Measured at VREF=100 mV.

b. Measured at external VREF=100 mV, input signal switching from 75 mV to 125 mV.

表 25-29. Analog Comparator Voltage Reference Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
R <sub>HR</sub>	Resolution in high range	-	V <sub>DDA</sub> /29.4	-	V
R <sub>LR</sub>	Resolution in low range	-	V <sub>DDA</sub> /22.12	-	V
A <sub>HR</sub>	Absolute accuracy high range	-	-	±R <sub>HR</sub> /2	V
A <sub>LR</sub>	Absolute accuracy low range	-	-	±R <sub>LR</sub> /2	V

表 25-30. Analog Comparator Voltage Reference Characteristics, V<sub>DDA</sub> = 3.3V, EN= 1, and RNG = 0

VREF Value	V <sub>IREF</sub> Min	Ideal V <sub>IREF</sub>	V <sub>IREF</sub> Max	Unit
0x0	0.731	0.786	0.841	V
0x1	0.843	0.898	0.953	V
0x2	0.955	1.010	1.065	V
0x3	1.067	1.122	1.178	V
0x4	1.180	1.235	1.290	V
0x5	1.292	1.347	1.402	V
0x6	1.404	1.459	1.514	V
0x7	1.516	1.571	1.627	V
0x8	1.629	1.684	1.739	V
0x9	1.741	1.796	1.851	V
0xA	1.853	1.908	1.963	V
0xB	1.965	2.020	2.076	V
0xC	2.078	2.133	2.188	V
0xD	2.190	2.245	2.300	V
0xE	2.302	2.357	2.412	V
0xF	2.414	2.469	2.525	V

表 25-31. Analog Comparator Voltage Reference Characteristics, V<sub>DDA</sub> = 3.3V, EN= 1, and RNG = 1

VREF Value	V <sub>IREF</sub> Min	Ideal V <sub>IREF</sub>	V <sub>IREF</sub> Max	Unit
0x0	0.000	0.000	0.074	V
0x1	0.076	0.149	0.223	V
0x2	0.225	0.298	0.372	V
0x3	0.374	0.448	0.521	V

表 25-31. Analog Comparator Voltage Reference Characteristics,  $V_{DDA} = 3.3V$ ,  $EN = 1$ , and  $RNG = 1$  (续)

VREF Value	$V_{IREF}$ Min	Ideal $V_{IREF}$	$V_{IREF}$ Max	Unit
0x4	0.523	0.597	0.670	V
0x5	0.672	0.746	0.820	V
0x6	0.822	0.895	0.969	V
0x7	0.971	1.044	1.118	V
0x8	1.120	1.193	1.267	V
0x9	1.269	1.343	1.416	V
0xA	1.418	1.492	1.565	V
0xB	1.567	1.641	1.715	V
0xC	1.717	1.790	1.864	V
0xD	1.866	1.939	2.013	V
0xE	2.015	2.089	2.162	V
0xF	2.164	2.238	2.311	V

## 25.18 Current Consumption

### 25.18.1 Preliminary Current Consumption

The following table provides preliminary figures for current consumption while ongoing characterization is completed.

表 25-32. Preliminary Current Consumption

Parameter	Parameter Name	Conditions	Nom	Max	Unit
$I_{DD\_RUN}$	Run mode 1 (Flash loop)	$V_{DD} = 3.3 V$ $V_{DDA} = 3.3 V$ Peripherals = All ON System Clock = 80 MHz (with PLL) Temp = 25°C	50	-	mA
	Run mode 1 (SRAM loop)	$V_{DD} = 3.3 V$ $V_{DDA} = 3.3 V$ Peripherals = All ON System Clock = 80 MHz (with PLL) Temp = 25°C	40	-	mA
	Run mode 2 (Flash loop)	$V_{DD} = 3.3 V$ $V_{DDA} = 3.3 V$ Peripherals = All OFF System Clock = 80 MHz (with PLL) Temp = 25°C	30	-	mA
	Run mode 2 (SRAM loop)	$V_{DD} = 3.3 V$ $V_{DDA} = 3.3 V$ Peripherals = All OFF System Clock = 80 MHz (with PLL) Temp = 25°C	20	-	mA

表 25-32. Preliminary Current Consumption (续)

Parameter	Parameter Name	Conditions	Nom	Max	Unit
$I_{DDA\_RUN}^a$	Run mode	$V_{DD} = 3.6\text{ V}$ $V_{DDA} = 3.6\text{ V}$ Peripherals = All ON System Clock = 80 MHz (with PLL) Temp = 25°C	-	4.5	mA
$I_{DD\_SLEEP}$	Sleep mode	$V_{DD} = 3.3\text{ V}$ $V_{DDA} = 3.3\text{ V}$ Peripherals = All OFF System Clock = 80 MHz (with PLL) Temp = 25°C	12	-	mA
		$V_{DD} = 3.3\text{ V}$ $V_{DDA} = 3.3\text{ V}$ Peripherals = All OFF System Clock = 16 MHz (with PIOSC) <sup>b</sup> Temp = 25°C	4.5	-	mA
		$V_{DD} = 3.3\text{ V}$ $V_{DDA} = 3.3\text{ V}$ Peripherals = All OFF System Clock = 1 MHz (with PIOSC/16) <sup>b</sup> Temp = 25°C	3.8	-	mA
$I_{DD\_DEEPSLEEP}$	Deep-sleep mode	$V_{DD} = 3.3\text{ V}$ $V_{DDA} = 3.3\text{ V}$ Peripherals = All OFF System Clock = IOS30KHZ Temp = 25°C	1.05	-	mA
$I_{HIB\_NORTC}$	Hibernate mode (external wake, RTC disabled)	$V_{BAT} = 3.0\text{ V}$ $V_{DD} = 0\text{ V}$ $V_{DDA} = 0\text{ V}$ System Clock = OFF Hibernate Module = 32.768 kHz	1.6	-	μA
$I_{HIB\_RTC}$	Hibernate mode (RTC enabled)	$V_{BAT} = 3.0\text{ V}$ $V_{DD} = 0\text{ V}$ $V_{DDA} = 0\text{ V}$ System Clock = OFF Hibernate Module = 32.768 kHz	1.7	-	μA
$I_{HIB\_VDD3ON}$	Hibernate mode (VDD3ON mode)	$V_{BAT} = 3.0\text{ V}$ $V_{DD} = 3.3\text{ V}$ $V_{DDA} = 3.3\text{ V}$ System Clock = OFF Hibernate Module = 32.768 kHz	5.0	-	μA

a. The value for  $I_{DDA\_RUN}$  is included in the above values for  $I_{DD\_RUN}$ .

b. Note that if the MOSC is the source of the Run-mode system clock and is powered down in Sleep mode, wake time is increased by  $T_{MOSC\_SETTLE}$ .

# A 寄存器快速参考

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>Cortex-M4F 处理器</b>															
<b>R0, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R1, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R2, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R3, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R4, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R5, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R6, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R7, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R8, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R9, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R10, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R11, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>R12, 类型 R/W, 复位 - (见第 76 页)</b>															
DATA															
DATA															
<b>SP, 类型 R/W, 复位 - (见第 77 页)</b>															
SP															
SP															
<b>LR, 类型 R/W, 复位 0xFFFF.FFFF (见第 78 页)</b>															
LINK															
LINK															
<b>PC, 类型 R/W, 复位 - (见第 79 页)</b>															
PC															
PC															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
<b>PSR</b> , 类型 R/W, 复位 0x0100.0000 ( 见第 80 页 )																	
N	Z	C	V	Q	ICI / IT		THUMB					GE					
ICI / IT								ISRNUM									
<b>PRIMASK</b> , 类型 R/W, 复位 0x0000.0000 ( 见第 84 页 )																	
															PRIMASK		
<b>FAULTMASK</b> , 类型 R/W, 复位 0x0000.0000 ( 见第 85 页 )																	
															FAULTMASK		
<b>BASEPRI</b> , 类型 R/W, 复位 0x0000.0000 ( 见第 86 页 )																	
															BASEPRI		
<b>CONTROL</b> , 类型 R/W, 复位 0x0000.0000 ( 见第 87 页 )																	
												FPCA	ASP	TMPL			
<b>FPSC</b> , 类型 R/W, 复位 - ( 见第 89 页 )																	
N	Z	C	V	AHP		DN	FZ	RMODE				IXC	UFC	OFC	DZC	IOC	
								IDC					IXC	UFC	OFC	DZC	IOC
<b>Cortex-M4 外设</b> 系统定时器 ( SysTick ) 寄存器 基址 0xE000.E000																	
<b>STCTRL</b> , 类型 R/W, 偏移量 0x010, 复位 0x0000.0004																	
												CLK_SRC	INTEN	COUNT	ENABLE		
<b>STRELOAD</b> , 类型 R/W, 偏移量 0x014, 复位 -																	
															RELOAD		
															RELOAD		
<b>STCURRENT</b> , 类型 R/W, 偏移量 0x018, 复位 -																	
															CURRENT		
															CURRENT		
<b>Cortex-M4 外设</b> 嵌套向量中断控制器 ( NVIC ) 寄存器 基址 0xE000.E000																	
<b>EN0</b> , 类型 R/W, 偏移量 0x100, 复位 0x0000.0000																	
															INT		
															INT		
<b>EN1</b> , 类型 R/W, 偏移量 0x104, 复位 0x0000.0000																	
															INT		
															INT		
<b>EN2</b> , 类型 R/W, 偏移量 0x108, 复位 0x0000.0000																	
															INT		
															INT		
<b>EN3</b> , 类型 R/W, 偏移量 0x10C, 复位 0x0000.0000																	
															INT		
															INT		
<b>EN4</b> , 类型 R/W, 偏移量 0x110, 复位 0x0000.0000																	
															INT		
															INT		
<b>DIS0</b> , 类型 R/W, 偏移量 0x180, 复位 0x0000.0000																	
															INT		
															INT		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>DIS1, 类型 R/W, 偏移量 0x184, 复位 0x0000.0000</b>															
INT															
INT															
<b>DIS2, 类型 R/W, 偏移量 0x188, 复位 0x0000.0000</b>															
INT															
INT															
<b>DIS3, 类型 R/W, 偏移量 0x18C, 复位 0x0000.0000</b>															
INT															
INT															
<b>DIS4, 类型 R/W, 偏移量 0x190, 复位 0x0000.0000</b>															
INT															
<b>PEND0, 类型 R/W, 偏移量 0x200, 复位 0x0000.0000</b>															
INT															
INT															
<b>PEND1, 类型 R/W, 偏移量 0x204, 复位 0x0000.0000</b>															
INT															
INT															
<b>PEND2, 类型 R/W, 偏移量 0x208, 复位 0x0000.0000</b>															
INT															
INT															
<b>PEND3, 类型 R/W, 偏移量 0x20C, 复位 0x0000.0000</b>															
INT															
INT															
<b>PEND4, 类型 R/W, 偏移量 0x210, 复位 0x0000.0000</b>															
INT															
<b>UNPEND0, 类型 R/W, 偏移量 0x280, 复位 0x0000.0000</b>															
INT															
INT															
<b>UNPEND1, 类型 R/W, 偏移量 0x284, 复位 0x0000.0000</b>															
INT															
INT															
<b>UNPEND2, 类型 R/W, 偏移量 0x288, 复位 0x0000.0000</b>															
INT															
INT															
<b>UNPEND3, 类型 R/W, 偏移量 0x28C, 复位 0x0000.0000</b>															
INT															
INT															
<b>UNPEND4, 类型 R/W, 偏移量 0x290, 复位 0x0000.0000</b>															
INT															
<b>ACTIVE0, 类型 RO, 偏移量 0x300, 复位 0x0000.0000</b>															
INT															
INT															
<b>ACTIVE1, 类型 RO, 偏移量 0x304, 复位 0x0000.0000</b>															
INT															
INT															
<b>ACTIVE2, 类型 RO, 偏移量 0x308, 复位 0x0000.0000</b>															
INT															
INT															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>ACTIVE3, 类型 RO, 偏移量 0x30C, 复位 0x0000.0000</b>															
INT															
INT															
<b>ACTIVE4, 类型 RO, 偏移量 0x310, 复位 0x0000.0000</b>															
INT															
<b>PRI0, 类型 R/W, 偏移量 0x400, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI1, 类型 R/W, 偏移量 0x404, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI2, 类型 R/W, 偏移量 0x408, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI3, 类型 R/W, 偏移量 0x40C, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI4, 类型 R/W, 偏移量 0x410, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI5, 类型 R/W, 偏移量 0x414, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI6, 类型 R/W, 偏移量 0x418, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI7, 类型 R/W, 偏移量 0x41C, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI8, 类型 R/W, 偏移量 0x420, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI9, 类型 R/W, 偏移量 0x424, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI10, 类型 R/W, 偏移量 0x428, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI11, 类型 R/W, 偏移量 0x42C, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI12, 类型 R/W, 偏移量 0x430, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI13, 类型 R/W, 偏移量 0x434, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI14, 类型 R/W, 偏移量 0x438, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>PRI15, 类型 R/W, 偏移量 0x43C, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI16, 类型 R/W, 偏移量 0x440, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI17, 类型 R/W, 偏移量 0x444, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI18, 类型 R/W, 偏移量 0x448, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI19, 类型 R/W, 偏移量 0x44C, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI20, 类型 R/W, 偏移量 0x450, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI21, 类型 R/W, 偏移量 0x454, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI22, 类型 R/W, 偏移量 0x458, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI23, 类型 R/W, 偏移量 0x45C, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI24, 类型 R/W, 偏移量 0x460, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI25, 类型 R/W, 偏移量 0x464, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI26, 类型 R/W, 偏移量 0x468, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI27, 类型 R/W, 偏移量 0x46C, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI28, 类型 R/W, 偏移量 0x470, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI29, 类型 R/W, 偏移量 0x474, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI30, 类型 R/W, 偏移量 0x478, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							
<b>PRI31, 类型 R/W, 偏移量 0x47C, 复位 0x0000.0000</b>															
INTD								INTC							
INTB								INTA							



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
<b>PRI32, 类型 R/W, 偏移量 0x480, 复位 0x0000.0000</b>																											
INTD								INTC																			
INTB								INTA																			
<b>PRI33, 类型 R/W, 偏移量 0x484, 复位 0x0000.0000</b>																											
INTD								INTC																			
INTB								INTA																			
<b>PRI34, 类型 R/W, 偏移量 0x488, 复位 0x0000.0000</b>																											
INTD								INTC																			
INTB								INTA																			
<b>SWTRIG, 类型 WO, 偏移量 0xF00, 复位 0x0000.0000</b>																											
												INTID															
<b>Cortex-M4 外设</b>																											
<b>系统控制模块 (SCB) 寄存器</b>																											
<b>基址 0xE000.E000</b>																											
<b>ACTLR, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000</b>																											
				DISOFP				DISFPCA								DISFOLD				DISWBUF		DISMCYC					
<b>CPUID, 类型 RO, 偏移量 0xD00, 复位 0x410F.C241</b>																											
IMP				PARTNO				VAR				CON															
												REV															
<b>INTCTRL, 类型 R/W, 偏移量 0xD04, 复位 0x0000.0000</b>																											
NMISSET				PENDSV		UNPENDSV		PENDSTSET		PENDSTCLR		ISRPRE		ISRPEND		VECPEND											
VECPEND				RETBASE								VECACT															
<b>VTABLE, 类型 R/W, 偏移量 0xD08, 复位 0x0000.0000</b>																											
BASE				OFFSET																							
OFFSET																											
<b>APINT, 类型 R/W, 偏移量 0xD0C, 复位 0xFA05.0000</b>																											
VECTKEY																											
ENDIANESS				PRIGROUP								SYSRESREQ				VECTOLRACT		VECTRESET									
<b>SYSCTRL, 类型 R/W, 偏移量 0xD10, 复位 0x0000.0000</b>																											
												SEVONPEND		SLEEPDEEP		SLEEPEXIT											
<b>CFGCTRL, 类型 R/W, 偏移量 0xD14, 复位 0x0000.0200</b>																											
				STKALIGN				BFHFMIGN				DIV0				UNALIGNED		MAINPEND		BASETHR							
<b>SYSPR1, 类型 R/W, 偏移量 0xD18, 复位 0x0000.0000</b>																											
BUS								USAGE				MEM															
<b>SYSPR2, 类型 R/W, 偏移量 0xD1C, 复位 0x0000.0000</b>																											
SVC																											
<b>SYSPR3, 类型 R/W, 偏移量 0xD20, 复位 0x0000.0000</b>																											
TICK								PENDSV				DEBUG															
<b>SYSHNDCTRL, 类型 R/W, 偏移量 0xD24, 复位 0x0000.0000</b>																											
SVC		BUSP		MEMP		USAGEP		TICK		PNDSV		MON		SVCA		USGA		USAGE		BUS		MEM					
																		BUSA		MEMA							
<b>FAULTSTAT, 类型 R/W, 偏移量 0xD28, 复位 0x0000.0000</b>																											
BFARV		BLSPERR		BSTKE		BUSTKE		IMPRE		PRECISE		IBUS		MMARV		MLSPERR		MSTKE		NOCP		INVPC		INVSTAT		UNDEF	
																				DERR		IERR					

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
<b>HFAULTSTAT</b> , 类型 R/W1C, 偏移量 0xD2C, 复位 0x0000.0000																	
DBG		FORCED														VECT	
<b>MMADDR</b> , 类型 R/W, 偏移量 0xD34, 复位 -																	
ADDR																	
ADDR																	
<b>FAULTADDR</b> , 类型 R/W, 偏移量 0xD38, 复位 -																	
ADDR																	
ADDR																	
<b>Cortex-M4 外设</b>																	
存储器保护单元 (MPU) 寄存器																	
基址 0xE000.E000																	
<b>MPUTYPE</b> , 类型 RO, 偏移量 0xD90, 复位 0x0000.0800																	
DREGION								IREGION									
SEPARATE																	
<b>MPUCTRL</b> , 类型 R/W, 偏移量 0xD94, 复位 0x0000.0000																	
												PRIVDEFEN		HFNMIENA		ENABLE	
<b>MPUNUMBER</b> , 类型 R/W, 偏移量 0xD98, 复位 0x0000.0000																	
NUMBER																	
<b>MPUBASE</b> , 类型 R/W, 偏移量 0xD9C, 复位 0x0000.0000																	
ADDR																	
ADDR												VALID		REGION			
<b>MPUBASE1</b> , 类型 R/W, 偏移量 0xDA4, 复位 0x0000.0000																	
ADDR																	
ADDR												VALID		REGION			
<b>MPUBASE2</b> , 类型 R/W, 偏移量 0xDAC, 复位 0x0000.0000																	
ADDR																	
ADDR												VALID		REGION			
<b>MPUBASE3</b> , 类型 R/W, 偏移量 0xDB4, 复位 0x0000.0000																	
ADDR																	
ADDR												VALID		REGION			
<b>MPUATTR</b> , 类型 R/W, 偏移量 0xDA0, 复位 0x0000.0000																	
XN				AP				TEX				S		C		B	
SRD								SIZE								ENABLE	
<b>MPUATTR1</b> , 类型 R/W, 偏移量 0xDA8, 复位 0x0000.0000																	
XN				AP				TEX				S		C		B	
SRD								SIZE								ENABLE	
<b>MPUATTR2</b> , 类型 R/W, 偏移量 0xDB0, 复位 0x0000.0000																	
XN				AP				TEX				S		C		B	
SRD								SIZE								ENABLE	
<b>MPUATTR3</b> , 类型 R/W, 偏移量 0xDB8, 复位 0x0000.0000																	
XN				AP				TEX				S		C		B	
SRD								SIZE								ENABLE	
<b>Cortex-M4 外设</b>																	
浮点单元 (FPU) 寄存器																	
基址 0xE000.E000																	
<b>CPAC</b> , 类型 R/W, 偏移量 0xD88, 复位 0x0000.0000																	
								CP11				CP10					

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
<b>FPCC, 类型 R/W, 偏移量 0xF34, 复位 0xC000.0000</b>																											
ASPEN		LSPEN						MONRDY		BFRDY		MMRDY		HFRDY		线程	USER	LSPACT									
<b>FPCA, 类型 R/W, 偏移量 0xF38, 复位 -</b>																											
地址																											
地址																											
<b>FPDSC, 类型 R/W, 偏移量 0xF3C, 复位 0x0000.0000</b>																											
				AHP		DN		FZ		RMODE																	
<b>系统控制</b>																											
<b>系统控制寄存器</b>																											
基址 0x400F.E000																											
<b>DID0, 类型 RO, 偏移量 0x000, 复位 -</b>																											
VER								CLASS																			
MAJOR								MINOR																			
<b>DID1, 类型 RO, 偏移量 0x004, 复位 -</b>																											
VER				FAM				PARTNO																			
PINCOUNT								TEMP				PKG		ROHS		QUAL											
<b>PBORCTL, 类型 R/W, 偏移量 0x030, 复位 0x0000.0000</b>																											
														BORIOR													
<b>RIS, 类型 RO, 偏移量 0x050, 复位 0x0000.0000</b>																											
				MOSCPUPRIS				USBPLLRIS		PLLRIS						MOFRIS		BORRIS									
<b>IMC, 类型 R/W, 偏移量 0x054, 复位 0x0000.0000</b>																											
				MOSCPUPIM				USBPLLLIM		PLLLIM						MOFIM		BORIM									
<b>MISC, 类型 R/W1C, 偏移量 0x058, 复位 0x0000.0000</b>																											
				MOSCPUPMS				USBPLLLMIS		PLLLMIS						MOFMIS		BORMIS									
<b>RESC, 类型 R/W, 偏移量 0x05C, 复位 -</b>																											
														WDT1		SW		WDT0		BOR		POR		EXT		MOSCFAIL	
<b>RCC, 类型 R/W, 偏移量 0x060, 复位 0x078E.3D51</b>																											
PWRDN				ACG		SYSDIV				USESYSIV		USEPWMDIV		PWMDIV													
				BYPASS		XTAL				OSCSRC				IOSCDIS				MOSCDIS									
<b>GPIOHBCTL, 类型 R/W, 偏移量 0x06C, 复位 0x0000.7E00</b>																											
PORTP		PORTN		PORTM		PORTL		PORTK		PORTJ		PORTH		PORTG		PORTF		PORTE		PORTD		PORTC		PORTB		PORTA	
<b>RCC2, 类型 R/W, 偏移量 0x070, 复位 0x07C0.6810</b>																											
USERCC2		DIV400		SYSDIV2				SYSDIV2LSB																			
USBPWRDN		PWRDN2		BYPASS2				OSCSRC2																			
<b>MOSCCTL, 类型 R/W, 偏移量 0x07C, 复位 0x0000.0000</b>																											
														NOXTAL		MOSCIM		CVAL									
<b>DSLCLKCFG, 类型 R/W, 偏移量 0x144, 复位 0x0780.0000</b>																											
DSDIVORIDE								DSOSCSRC																			
<b>SYSPROP, 类型 RO, 偏移量 0x14C, 复位 0x0000.1D31</b>																											
														FPU													

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
<b>PIOSCCAL, 类型 R/W, 偏移量 0x150, 复位 0x0000.0000</b>																																
UTEN				CAL				UPDATE				UT																				
<b>PIOSCSTAT, 类型 RO, 偏移量 0x154, 复位 0x0000.0040</b>																																
RESULT												DT																				
												CT																				
<b>PLLFREQ0, 类型 RO, 偏移量 0x160, 复位 0x0000.0032</b>																																
MFRAC								MINT				MFRAC																				
<b>PLLFREQ1, 类型 RO, 偏移量 0x164, 复位 0x0000.0001</b>																																
Q								N																								
<b>PLLSTAT, 类型 RO, 偏移量 0x168, 复位 0x0000.0000</b>																																
															LOCK																	
<b>PPWD, 类型 RO, 偏移量 0x300, 复位 0x0000.0003</b>																																
														P1	P0																	
<b>PPTIMER, 类型 RO, 偏移量 0x304, 复位 0x0000.003F</b>																																
										P5	P4	P3	P2	P1	P0																	
<b>PPGPIO, 类型 RO, 偏移量 0x308, 复位 0x0000.3FFF</b>																																
P14				P13			P12		P11		P10		P9		P8		P7		P6		P5		P4		P3		P2		P1		P0	
<b>PPDMA, 类型 RO, 偏移量 0x30C, 复位 0x0000.0001</b>																																
															P0																	
<b>PPHIB, 类型 RO, 偏移量 0x314, 复位 0x0000.0001</b>																																
															P0																	
<b>PPUART, 类型 RO, 偏移量 0x318, 复位 0x0000.00FF</b>																																
										P7	P6	P5	P4	P3	P2	P1	P0															
<b>PPSSI, 类型 RO, 偏移量 0x31C, 复位 0x0000.000F</b>																																
												P3	P2	P1	P0																	
<b>PPI2C, 类型 RO, 偏移量 0x320, 复位 0x0000.003F</b>																																
										P5	P4	P3	P2	P1	P0																	
<b>PPUSB, 类型 RO, 偏移量 0x328, 复位 0x0000.0001</b>																																
															P0																	
<b>PPCAN, 类型 RO, 偏移量 0x334, 复位 0x0000.0003</b>																																
														P1	P0																	
<b>PPADC, 类型 RO, 偏移量 0x338, 复位 0x0000.0003</b>																																
														P1	P0																	
<b>PPACMP, 类型 RO, 偏移量 0x33C, 复位 0x0000.0001</b>																																
															P0																	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>PPPWM, 类型 RO, 偏移量 0x340, 复位 0x0000.0003</b>																
														P1	P0	
<b>PPQEI, 类型 RO, 偏移量 0x344, 复位 0x0000.0003</b>																
														P1	P0	
<b>PPEEPROM, 类型 RO, 偏移量 0x358, 复位 0x0000.0001</b>																
															P0	
<b>PPWTIMER, 类型 RO, 偏移量 0x35C, 复位 0x0000.003F</b>																
										P5	P4	P3	P2	P1	P0	
<b>SRWD, 类型 R/W, 偏移量 0x500, 复位 0x0000.0000</b>																
														R1	R0	
<b>SRTIMER, 类型 R/W, 偏移量 0x504, 复位 0x0000.0000</b>																
										R5	R4	R3	R2	R1	R0	
<b>SRGPIO, 类型 R/W, 偏移量 0x508, 复位 0x0000.0000</b>																
			R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
<b>SRDMA, 类型 R/W, 偏移量 0x50C, 复位 0x0000.0000</b>																
																R0
<b>SRHIB, 类型 R/W, 偏移量 0x514, 复位 0x0000.0000</b>																
																R0
<b>SRUART, 类型 R/W, 偏移量 0x518, 复位 0x0000.0000</b>																
									R7	R6	R5	R4	R3	R2	R1	R0
<b>SRSSI, 类型 R/W, 偏移量 0x51C, 复位 0x0000.0000</b>																
													R3	R2	R1	R0
<b>SRI2C, 类型 R/W, 偏移量 0x520, 复位 0x0000.0000</b>																
										R5	R4	R3	R2	R1	R0	
<b>SRUSB, 类型 R/W, 偏移量 0x528, 复位 0x0000.0000</b>																
																R0
<b>SRCAN, 类型 R/W, 偏移量 0x534, 复位 0x0000.0000</b>																
															R1	R0
<b>SRADC, 类型 R/W, 偏移量 0x538, 复位 0x0000.0000</b>																
															R1	R0
<b>SRACMP, 类型 R/W, 偏移量 0x53C, 复位 0x0000.0000</b>																
																R0
<b>SRPWM, 类型 R/W, 偏移量 0x540, 复位 0x0000.0000</b>																
															R1	R0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>SRQEI, 类型 R/W, 偏移量 0x544, 复位 0x0000.0000</b>																
														R1	R0	
<b>SRWTIMER, 类型 R/W, 偏移量 0x55C, 复位 0x0000.0000</b>																
										R5	R4	R3	R2	R1	R0	
<b>RCGCWD, 类型 R/W, 偏移量 0x600, 复位 0x0000.0000</b>																
														R1	R0	
<b>RCGCTIMER, 类型 R/W, 偏移量 0x604, 复位 0x0000.0000</b>																
										R5	R4	R3	R2	R1	R0	
<b>RCGCGPIO, 类型 R/W, 偏移量 0x608, 复位 0x0000.0000</b>																
			R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
<b>RCGCDMA, 类型 R/W, 偏移量 0x60C, 复位 0x0000.0000</b>																
															R0	
<b>RCGCHIB, 类型 R/W, 偏移量 0x614, 复位 0x0000.0001</b>																
															R0	
<b>RCGCUART, 类型 R/W, 偏移量 0x618, 复位 0x0000.0000</b>																
								R7	R6	R5	R4	R3	R2	R1	R0	
<b>RCGCSSI, 类型 R/W, 偏移量 0x61C, 复位 0x0000.0000</b>																
												R3	R2	R1	R0	
<b>RCGC12C, 类型 R/W, 偏移量 0x620, 复位 0x0000.0000</b>																
										R5	R4	R3	R2	R1	R0	
<b>RCGCUSB, 类型 R/W, 偏移量 0x628, 复位 0x0000.0000</b>																
															R0	
<b>RCGCCAN, 类型 R/W, 偏移量 0x634, 复位 0x0000.0000</b>																
														R1	R0	
<b>RCGCADC, 类型 R/W, 偏移量 0x638, 复位 0x0000.0000</b>																
														R1	R0	
<b>RCGCACMP, 类型 R/W, 偏移量 0x63C, 复位 0x0000.0000</b>																
															R0	
<b>RCGCPWM, 类型 R/W, 偏移量 0x640, 复位 0x0000.0000</b>																
														R1	R0	
<b>RCGCQEI, 类型 R/W, 偏移量 0x644, 复位 0x0000.0000</b>																
														R1	R0	
<b>RCGCWTIMER, 类型 R/W, 偏移量 0x65C, 复位 0x0000.0000</b>																
										R5	R4	R3	R2	R1	R0	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>SCGCWD, 类型 R/W, 偏移量 0x700, 复位 0x0000.0000</b>																
														S1	S0	
<b>SCGCTIMER, 类型 R/W, 偏移量 0x704, 复位 0x0000.0000</b>																
										S5	S4	S3	S2	S1	S0	
<b>SCGCGPIO, 类型 R/W, 偏移量 0x708, 复位 0x0000.0000</b>																
			S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
<b>SCGCDMA, 类型 R/W, 偏移量 0x70C, 复位 0x0000.0000</b>																
																S0
<b>SCGCHIB, 类型 R/W, 偏移量 0x714, 复位 0x0000.0001</b>																
																S0
<b>SCGCUART, 类型 R/W, 偏移量 0x718, 复位 0x0000.0000</b>																
								S7	S6	S5	S4	S3	S2	S1	S0	
<b>SCGCSSI, 类型 R/W, 偏移量 0x71C, 复位 0x0000.0000</b>																
												S3	S2	S1	S0	
<b>SCGC12C, 类型 R/W, 偏移量 0x720, 复位 0x0000.0000</b>																
										S5	S4	S3	S2	S1	S0	
<b>SCGCUSB, 类型 R/W, 偏移量 0x728, 复位 0x0000.0000</b>																
																S0
<b>SCGCCAN, 类型 R/W, 偏移量 0x734, 复位 0x0000.0000</b>																
														S1	S0	
<b>SCGCADC, 类型 R/W, 偏移量 0x738, 复位 0x0000.0000</b>																
														S1	S0	
<b>SCGCACMP, 类型 R/W, 偏移量 0x73C, 复位 0x0000.0000</b>																
																S0
<b>SCGCPWM, 类型 R/W, 偏移量 0x740, 复位 0x0000.0000</b>																
														S1	S0	
<b>SCGCQE1, 类型 R/W, 偏移量 0x744, 复位 0x0000.0000</b>																
														S1	S0	
<b>SCGCWTIMER, 类型 R/W, 偏移量 0x75C, 复位 0x0000.0000</b>																
										S5	S4	S3	S2	S1	S0	
<b>DCGCWD, 类型 R/W, 偏移量 0x800, 复位 0x0000.0000</b>																
														D1	D0	
<b>DCGCTIMER, 类型 R/W, 偏移量 0x804, 复位 0x0000.0000</b>																
										D5	D4	D3	D2	D1	D0	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>DCGCGPIO, 类型 R/W, 偏移量 0x808, 复位 0x0000.0000</b>																
			D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
<b>DCGCDMA, 类型 R/W, 偏移量 0x80C, 复位 0x0000.0000</b>																
																D0
<b>DCGCHIB, 类型 R/W, 偏移量 0x814, 复位 0x0000.0001</b>																
																D0
<b>DCGCUART, 类型 R/W, 偏移量 0x818, 复位 0x0000.0000</b>																
								D7	D6	D5	D4	D3	D2	D1	D0	
<b>DCGCSSI, 类型 R/W, 偏移量 0x81C, 复位 0x0000.0000</b>																
												D3	D2	D1	D0	
<b>DCGC12C, 类型 R/W, 偏移量 0x820, 复位 0x0000.0000</b>																
										D5	D4	D3	D2	D1	D0	
<b>DCGCUSB, 类型 R/W, 偏移量 0x828, 复位 0x0000.0000</b>																
																D0
<b>DCGCCAN, 类型 R/W, 偏移量 0x834, 复位 0x0000.0000</b>																
															D1	D0
<b>DCGCADC, 类型 R/W, 偏移量 0x838, 复位 0x0000.0000</b>																
															D1	D0
<b>DCGCACMP, 类型 R/W, 偏移量 0x83C, 复位 0x0000.0000</b>																
																D0
<b>DCGCPWM, 类型 R/W, 偏移量 0x840, 复位 0x0000.0000</b>																
															D1	D0
<b>DCGCQEI, 类型 R/W, 偏移量 0x844, 复位 0x0000.0000</b>																
															D1	D0
<b>DCGCWTIMER, 类型 R/W, 偏移量 0x85C, 复位 0x0000.0000</b>																
										D5	D4	D3	D2	D1	D0	
<b>PCWD, 类型 R/W, 偏移量 0x900, 复位 0x0000.0003</b>																
															P1	P0
<b>PCTIMER, 类型 R/W, 偏移量 0x904, 复位 0x0000.003F</b>																
										P5	P4	P3	P2	P1	P0	
<b>PCGPIO, 类型 R/W, 偏移量 0x908, 复位 0x0000.7FFF</b>																
			P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
<b>PCDMA, 类型 R/W, 偏移量 0x90C, 复位 0x0000.0001</b>																
																P0



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>PCCHIB, 类型 R/W, 偏移量 0x914, 复位 0x0000.0001</b>																
															P0	
<b>PCUART, 类型 R/W, 偏移量 0x918, 复位 0x0000.00FF</b>																
								P7	P6	P5	P4	P3	P2	P1	P0	
<b>PCSSI, 类型 R/W, 偏移量 0x91C, 复位 0x0000.000F</b>																
												P3	P2	P1	P0	
<b>PCI2C, 类型 R/W, 偏移量 0x920, 复位 0x0000.003F</b>																
										P5	P4	P3	P2	P1	P0	
<b>PCUSB, 类型 R/W, 偏移量 0x928, 复位 0x0000.0001</b>																
															P0	
<b>PCCAN, 类型 R/W, 偏移量 0x934, 复位 0x0000.0003</b>																
														P1	P0	
<b>PCADC, 类型 R/W, 偏移量 0x938, 复位 0x0000.0003</b>																
														P1	P0	
<b>PCACMP, 类型 R/W, 偏移量 0x93C, 复位 0x0000.0001</b>																
															P0	
<b>PCPWM, 类型 R/W, 偏移量 0x940, 复位 0x0000.0003</b>																
														P1	P0	
<b>PCQEI, 类型 R/W, 偏移量 0x944, 复位 0x0000.0003</b>																
														P1	P0	
<b>PCWTIMER, 类型 R/W, 偏移量 0x95C, 复位 0x0000.0000</b>																
										P5	P4	P3	P2	P1	P0	
<b>PRWD, 类型 R/W, 偏移量 0xA00, 复位 0x0000.0000</b>																
														R1	R0	
<b>PRTIMER, 类型 R/W, 偏移量 0xA04, 复位 0x0000.0000</b>																
										R5	R4	R3	R2	R1	R0	
<b>PRGPIO, 类型 R/W, 偏移量 0xA08, 复位 0x0000.0000</b>																
			R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
<b>PRDMA, 类型 R/W, 偏移量 0xA0C, 复位 0x0000.0000</b>																
																R0
<b>PRHIB, 类型 R/W, 偏移量 0xA14, 复位 0x0000.0001</b>																
																R0
<b>PRUART, 类型 R/W, 偏移量 0xA18, 复位 0x0000.0000</b>																
								R7	R6	R5	R4	R3	R2	R1	R0	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
<b>PRSSI, 类型 R/W, 偏移量 0xA1C, 复位 0x0000.0000</b>																				
												R3	R2	R1	R0					
<b>PRI2C, 类型 R/W, 偏移量 0xA20, 复位 0x0000.0000</b>																				
										R5	R4	R3	R2	R1	R0					
<b>PRUSB, 类型 R/W, 偏移量 0xA28, 复位 0x0000.0000</b>																				
															R0					
<b>PRCAN, 类型 R/W, 偏移量 0xA34, 复位 0x0000.0000</b>																				
														R1	R0					
<b>PRADC, 类型 R/W, 偏移量 0xA38, 复位 0x0000.0000</b>																				
														R1	R0					
<b>PRACMP, 类型 R/W, 偏移量 0xA3C, 复位 0x0000.0000</b>																				
															R0					
<b>PRPWM, 类型 R/W, 偏移量 0xA40, 复位 0x0000.0000</b>																				
														R1	R0					
<b>PRQEI, 类型 R/W, 偏移量 0xA44, 复位 0x0000.0000</b>																				
														R1	R0					
<b>PRWTIMER, 类型 R/W, 偏移量 0xA5C, 复位 0x0000.0000</b>																				
										R5	R4	R3	R2	R1	R0					
<b>系统控制</b>																				
<b>系统控制传统寄存器</b>																				
基址 0x400F.E000																				
<b>DC0, 类型 RO, 偏移量 0x008, 复位 0x007F.007F</b>																				
SRAMSZ																				
FLASHSZ																				
<b>DC1, 类型 RO, 偏移量 0x010, 复位 0x1333.2FFF</b>																				
				WDT1					CAN1	CAN0					PWM1	PWM0			ADC1	ADC0
MINSYSDIV				MAXADC1SPD				MAXADC0SPD				MPU	HIB	TEMPSNS	PLL	WDT0	SWO	SWD	JTAG	
<b>DC2, 类型 RO, 偏移量 0x014, 复位 0x070F.F337</b>																				
EPI0		I2S0		COMP2				COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0			
I2C1HS	I2C1	I2C0HS	I2C0					QE1	QE0					SSI1	SSI0	UART2	UART1	UART0		
<b>DC3, 类型 RO, 偏移量 0x018, 复位 0xBFFF.FFFF</b>																				
32KHZ	CCP5		CCP4	CCP3	CCP2	CCP1	CCP0	ADC0AIN7	ADC0AIN6	ADC0AIN5	ADC0AIN4	ADC0AIN3	ADC0AIN2	ADC0AIN1	ADC0AIN0					
PWMFAULT	C20	C2PLUS	C2MINUS	C10	C1PLUS	C1MINUS	C00	C0PLUS	C0MINUS	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0					
<b>DC4, 类型 RO, 偏移量 0x01C, 复位 0x0004.F1FF</b>																				
EPHY0		EMAC0								E1588					PICAL					
CCP7	CCP6	UDMA	ROM					GPIOJ	GPIOH	GPIOG	GPIOF	GPIOE	GIPOD	GPIOC	GPIOB	GPIOA				
<b>DC5, 类型 RO, 偏移量 0x020, 复位 0x0F30.00FF</b>																				
								PWMFAULT3	PWMFAULT2	PWMFAULT1	PWMFAULT0					PWMEFLT	PWMSYNC			
								PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0					
<b>DC6, 类型 RO, 偏移量 0x024, 复位 0x0000.0013</b>																				
												USB0PHY					USB0			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
<b>DC7, 类型 RO, 偏移量 0x028, 复位 0xFFFF.FFFF</b>																	
	DMACH30	DMACH29	DMACH28	DMACH27	DMACH26	DMACH25	DMACH24	DMACH23	DMACH22	DMACH21	DMACH20	DMACH19	DMACH18	DMACH17	DMACH16		
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8	DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0		
<b>DC8, 类型 RO, 偏移量 0x02C, 复位 0xFFFF.FFFF</b>																	
ADC1AIN15	ADC1AIN14	ADC1AIN13	ADC1AIN12	ADC1AIN11	ADC1AIN10	ADC1AIN9	ADC1AIN8	ADC1AIN7	ADC1AIN6	ADC1AIN5	ADC1AIN4	ADC1AIN3	ADC1AIN2	ADC1AIN1	ADC1AIN0		
ADC0AIN15	ADC0AIN14	ADC0AIN13	ADC0AIN12	ADC0AIN11	ADC0AIN10	ADC0AIN9	ADC0AIN8	ADC0AIN7	ADC0AIN6	ADC0AIN5	ADC0AIN4	ADC0AIN3	ADC0AIN2	ADC0AIN1	ADC0AIN0		
<b>SRCR0, 类型 RO, 偏移量 0x040, 复位 0x0000.0000</b>																	
			WDT1				CAN1	CAN0				PWM0			ADC1	ADC0	
										HIB			WDT0				
<b>SRCR1, 类型 RO, 偏移量 0x044, 复位 0x0000.0000</b>																	
					COMP2	COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0		
	I2C1		I2C0			QE1	QE0				SSI1	SSI0		UART2	UART1	UART0	
<b>SRCR2, 类型 RO, 偏移量 0x048, 复位 0x0000.0000</b>																	
																USB0	
		UDMA						GPIOD	GPIOC	GPIOB	GPIOA						
<b>RCGC0, 类型 RO, 偏移量 0x100, 复位 0x0000.0040</b>																	
			WDT1				CAN1	CAN0				PWM0			ADC1	ADC0	
						MAXADC1SPD	MAXADC0SPD			HIB			WDT0				
<b>RCGC1, 类型 RO, 偏移量 0x104, 复位 0x0000.0000</b>																	
					COMP2	COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0		
	I2C1		I2C0			QE1	QE0				SSI1	SSI0		UART2	UART1	UART0	
<b>RCGC2, 类型 RO, 偏移量 0x108, 复位 0x0000.0000</b>																	
								GPIOD	GPIOC	GPIOB	GPIOA					USB0	
		UDMA															
<b>SCGC0, 类型 RO, 偏移量 0x110, 复位 0x0000.0040</b>																	
			WDT1				CAN1	CAN0				PWM0			ADC1	ADC0	
										HIB			WDT0				
<b>SCGC1, 类型 RO, 偏移量 0x114, 复位 0x0000.0000</b>																	
					COMP2	COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0		
	I2C1		I2C0			QE1	QE0				SSI1	SSI0		UART2	UART1	UART0	
<b>SCGC2, 类型 RO, 偏移量 0x118, 复位 0x0000.0000</b>																	
								GPIOD	GPIOC	GPIOB	GPIOA					USB0	
		UDMA															
<b>DCGC0, 类型 RO, 偏移量 0x120, 复位 0x0000.0040</b>																	
			WDT1				CAN1	CAN0				PWM0			ADC1	ADC0	
										HIB			WDT0				
<b>DCGC1, 类型 RO, 偏移量 0x124, 复位 0x0000.0000</b>																	
					COMP2	COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0		
	I2C1		I2C0			QE1	QE0				SSI1	SSI0		UART2	UART1	UART0	
<b>DCGC2, 类型 RO, 偏移量 0x128, 复位 0x0000.0000</b>																	
								GPIOD	GPIOC	GPIOB	GPIOA					USB0	
		UDMA															
<b>DC9, 类型 RO, 偏移量 0x190, 复位 0x00FF.00FF</b>																	
								ADC1DC7	ADC1DC6	ADC1DC5	ADC1DC4	ADC1DC3	ADC1DC2	ADC1DC1	ADC1DC0		
								ADC0DC7	ADC0DC6	ADC0DC5	ADC0DC4	ADC0DC3	ADC0DC2	ADC0DC1	ADC0DC0		
<b>NVMSTAT, 类型 RO, 偏移量 0x1A0, 复位 0x0000.0001</b>																	
																FWB	
<b>系统异常模块</b>																	
基址 0x400F.9000																	
<b>SYSEXCRIS, 类型 RO, 偏移量 0x000, 复位 0x0000.0000 ( 见第 468 页 )</b>																	
												FPIXCRIS	FPOFCRIS	FPUFCRIS	FPIOCRIS	FPDZCRIS	FPIDCRIS

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
SYSEXCIM, 类型 R/W, 偏移量 0x004, 复位 0x0000.0000 (见第 470 页)																							
												FPIXCIM	FPOFCIM	FPUFCIM	FPIOCIM	FPDZCIM	FPIDCIM						
SYSEXCMS, 类型 RO, 偏移量 0x008, 复位 0x0000.0000 (见第 472 页)																							
												FPIXCMS	FPOFCMS	FPUFCMS	FPIOCMS	FPDZCMS	FPIDCMS						
SYSEXCIC, 类型 W1C, 偏移量 0x00C, 复位 0x0000.0000 (见第 474 页)																							
												FPIXCIC	FPOFCIC	FPUFCIC	FPIOCIC	FPDZCIC	FPIDCIC						
休眠模块 基址 0x400F.C000																							
HIBRTCC, 类型 RO, 偏移量 0x000, 复位 0x0000.0000 (见第 486 页)																							
												RTCC											
												RTCC											
HIBRTCM0, 类型 R/W, 偏移量 0x004, 复位 0xFFFF.FFFF (见第 487 页)																							
												RTCM0											
												RTCM0											
HIBRTCLD, 类型 R/W, 偏移量 0x00C, 复位 0x0000.0000 (见第 488 页)																							
												RTCLD											
												RTCLD											
HIBCTL, 类型 R/W, 偏移量 0x010, 复位 0x8000.2000 (见第 489 页)																							
WRC		VBATSEL		BATCHK		BATWKEN		VDD3ON		VABORT		CLK32EN		PINWEN		RTCWEN		OSCHYS		OSCDRV		OSCBYP	
												HIBREQ		RTCEN									
HIBIM, 类型 R/W, 偏移量 0x014, 复位 0x0000.0000 (见第 493 页)																							
												WC		EXTW		LOWBAT		RTCALTO					
HIBRIS, 类型 RO, 偏移量 0x018, 复位 0x0000.0000 (见第 495 页)																							
												WC		EXTW		LOWBAT		RTCALTO					
HIBMIS, 类型 RO, 偏移量 0x01C, 复位 0x0000.0000 (见第 497 页)																							
												WC		EXTW		LOWBAT		RTCALTO					
HIBIC, 类型 R/W1C, 偏移量 0x020, 复位 0x0000.0000 (见第 499 页)																							
												WC		EXTW		LOWBAT		RTCALTO					
HIBRTCT, 类型 R/W, 偏移量 0x024, 复位 0x0000.7FFF (见第 500 页)																							
												TRIM											
HIBRTCSS, 类型 R/W, 偏移量 0x028, 复位 0x0000.0000 (见第 501 页)																							
												RTCSSM											
												RTCSSC											
HIBDATA, 类型 R/W, 偏移量 0x030-0x06F, 复位 - (见第 502 页)																							
												RTD											
												RTD											
内部存储器 Flash 存储器寄存器 (Flash 控制偏移量) 基址 0x400F.D000																							
FMA, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000																							
												OFFSET											
												OFFSET											

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
<b>FMD, 类型 R/W, 偏移量 0x004, 复位 0x0000.0000</b>																		
DATA																		
DATA																		
<b>FMC, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000</b>																		
WRKEY																		
												COMT	MERASE	ERASE	WRITE			
<b>FCRIS, 类型 RO, 偏移量 0x00C, 复位 0x0000.0000</b>																		
												ERRIS	INVDRIS	VOLTRIS				
												ERIS	PRIS	ARIS				
<b>FCIM, 类型 R/W, 偏移量 0x010, 复位 0x0000.0000</b>																		
												PROGMASK	ERMASK	INVDMASK	VOLTMASK			
												EMASK	PMASK	AMASK				
<b>FCMISC, 类型 R/W1C, 偏移量 0x014, 复位 0x0000.0000</b>																		
												PROGMISC	ERMISC	INVDMISC	VOLTMISC			
												EMISC	PMISC	AMISC				
<b>FMC2, 类型 R/W, 偏移量 0x020, 复位 0x0000.0000</b>																		
WRKEY																		
															WRBUF			
<b>FWBVAL, 类型 R/W, 偏移量 0x030, 复位 0x0000.0000</b>																		
FWB[n]																		
FWB[n]																		
<b>FWBn, 类型 R/W, 偏移量 0x100 - 0x17C, 复位 0x0000.0000</b>																		
DATA																		
DATA																		
<b>FSIZE, 类型 RO, 偏移量 0xFC0, 复位 0x0000.007F</b>																		
SIZE																		
<b>SSIZE, 类型 RO, 偏移量 0xFC4, 复位 0x0000.007F</b>																		
SIZE																		
<b>ROMSWMAP, 类型 RO, 偏移量 0xFCC, 复位 0x0000.0000</b>																		
															SAFERTOS			
内部存储器																		
存储器寄存器 (系统控制偏移量)																		
基址 0x400F.E000																		
<b>RMCTL, 类型 R/W1C, 偏移量 0x0F0, 复位 -</b>																		
															BA			
<b>FMPRE0, 类型 R/W, 偏移量 0x130 和 0x200, 复位 0xFFFF.FFFF</b>																		
READ_ENABLE																		
READ_ENABLE																		
<b>FMPRE1, 类型 R/W, 偏移量 0x204, 复位 0xFFFF.FFFF</b>																		
READ_ENABLE																		
READ_ENABLE																		
<b>FMPRE2, 类型 R/W, 偏移量 0x208, 复位 0xFFFF.FFFF</b>																		
READ_ENABLE																		
READ_ENABLE																		
<b>FMPRE3, 类型 R/W, 偏移量 0x20C, 复位 0xFFFF.FFFF</b>																		
READ_ENABLE																		
READ_ENABLE																		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
<b>FMPPE0, 类型 R/W, 偏移量 0x134 和 0x400, 复位 0xFFFF.FFFF</b>																			
PROG_ENABLE																			
PROG_ENABLE																			
<b>FMPPE1, 类型 R/W, 偏移量 0x404, 复位 0xFFFF.FFFF</b>																			
PROG_ENABLE																			
PROG_ENABLE																			
<b>FMPPE2, 类型 R/W, 偏移量 0x408, 复位 0xFFFF.FFFF</b>																			
PROG_ENABLE																			
PROG_ENABLE																			
<b>FMPPE3, 类型 R/W, 偏移量 0x40C, 复位 0xFFFF.FFFF</b>																			
PROG_ENABLE																			
PROG_ENABLE																			
<b>BOOTCFG, 类型 RO, 偏移量 0x1D0, 复位 0xFFFF.FFFE</b>																			
NW																			
PORT				PIN				POL	EN						DBG1	DBG0			
<b>USER_REG0, 类型 R/W, 偏移量 0x1E0, 复位 0xFFFF.FFFF</b>																			
DATA																			
DATA																			
<b>USER_REG1, 类型 R/W, 偏移量 0x1E4, 复位 0xFFFF.FFFF</b>																			
DATA																			
DATA																			
<b>USER_REG2, 类型 R/W, 偏移量 0x1E8, 复位 0xFFFF.FFFF</b>																			
DATA																			
DATA																			
<b>USER_REG3, 类型 R/W, 偏移量 0x1EC, 复位 0xFFFF.FFFF</b>																			
DATA																			
DATA																			
<b>微型直接存储器访问 (μDMA)</b>																			
<b>μDMA 通道控制结构体 ( 从通道控制表基地址的偏移量 )</b>																			
基址 n/a																			
<b>DMASRCNDP, 类型 R/W, 偏移量 0x000, 复位 -</b>																			
ADDR																			
ADDR																			
<b>DMADSTNDP, 类型 R/W, 偏移量 0x004, 复位 -</b>																			
ADDR																			
ADDR																			
<b>DMACHCTL, 类型 R/W, 偏移量 0x008, 复位 -</b>																			
DSTINC				DSTSIZE				SRCINC				SRCSIZE				ARBSIZE			
ARBSIZE								XFERSIZE				NXTUSEBURST				XFERMODE			
<b>微型直接存储器访问 (μDMA)</b>																			
<b>μDMA 寄存器 ( 从 μDMA 基地址的偏移量 )</b>																			
基址 0x400F.F000																			
<b>DMASTAT, 类型 RO, 偏移量 0x000, 复位 0x001F.0000</b>																			
DMACHANS																			
STATE																			
MASTEN																			
<b>DMACFG, 类型 WO, 偏移量 0x004, 复位 -</b>																			
MASTEN																			
<b>DMACTLBASE, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000</b>																			
ADDR																			
ADDR																			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
<b>DMAALTBASE</b> , 类型 RO, 偏移量 0x00C, 复位 0x0000.0200																			
								ADDR											
								ADDR											
<b>DMAWAITSTAT</b> , 类型 RO, 偏移量 0x010, 复位 0xFFFF.FFC0																			
								WAITREQ[n]											
								WAITREQ[n]											
<b>DMASWREQ</b> , 类型 WO, 偏移量 0x014, 复位 -																			
								SWREQ[n]											
								SWREQ[n]											
<b>DMAUSEBURSTSET</b> , 类型 R/W, 偏移量 0x018, 复位 0x0000.0000																			
								SET[n]											
								SET[n]											
<b>DMAUSEBURSTCLR</b> , 类型 WO, 偏移量 0x01C, 复位 -																			
								CLR[n]											
								CLR[n]											
<b>DMAREQMASKSET</b> , 类型 R/W, 偏移量 0x020, 复位 0x0000.0000																			
								SET[n]											
								SET[n]											
<b>DMAREQMASKCLR</b> , 类型 WO, 偏移量 0x024, 复位 -																			
								CLR[n]											
								CLR[n]											
<b>DMAENASET</b> , 类型 R/W, 偏移量 0x028, 复位 0x0000.0000																			
								SET[n]											
								SET[n]											
<b>DMAENACL</b> , 类型 WO, 偏移量 0x02C, 复位 -																			
								CLR[n]											
								CLR[n]											
<b>DMAALTSET</b> , 类型 R/W, 偏移量 0x030, 复位 0x0000.0000																			
								SET[n]											
								SET[n]											
<b>DMAALTCLR</b> , 类型 WO, 偏移量 0x034, 复位 -																			
								CLR[n]											
								CLR[n]											
<b>DMAPRIOSET</b> , 类型 R/W, 偏移量 0x038, 复位 0x0000.0000																			
								SET[n]											
								SET[n]											
<b>DMAPRIOCLR</b> , 类型 WO, 偏移量 0x03C, 复位 -																			
								CLR[n]											
								CLR[n]											
<b>DMAERRCLR</b> , 类型 R/W, 偏移量 0x04C, 复位 0x0000.0000																			
															ERRCLR				
<b>DMACHASGN</b> , 类型 R/W, 偏移量 0x500, 复位 0x0000.0000																			
								CHASGN[n]											
								CHASGN[n]											
<b>DMACHIS</b> , 类型 R/W1C, 偏移量 0x504, 复位 0x0000.0000																			
								CHIS[n]											
								CHIS[n]											
<b>DMACHMAP0</b> , 类型 R/W, 偏移量 0x510, 复位 0x0000.0000																			
				CH7SEL				CH6SEL				CH5SEL				CH4SEL			
				CH3SEL				CH2SEL				CH1SEL				CH0SEL			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>DMACHMAP1, 类型 R/W, 偏移量 0x514, 复位 0x0000.0000</b>															
CH15SEL				CH14SEL				CH13SEL				CH12SEL			
CH11SEL				CH10SEL				CH9SEL				CH8SEL			
<b>DMACHMAP2, 类型 R/W, 偏移量 0x518, 复位 0x0000.0000</b>															
CH23SEL				CH22SEL				CH21SEL				CH20SEL			
CH19SEL				CH18SEL				CH17SEL				CH16SEL			
<b>DMACHMAP3, 类型 R/W, 偏移量 0x51C, 复位 0x0000.0000</b>															
CH31SEL				CH30SEL				CH29SEL				CH28SEL			
CH27SEL				CH26SEL				CH25SEL				CH24SEL			
<b>DMAPeriphID0, 类型 RO, 偏移量 0xFE0, 复位 0x0000.0030</b>															
												PID0			
<b>DMAPeriphID1, 类型 RO, 偏移量 0xFE4, 复位 0x0000.00B2</b>															
												PID1			
<b>DMAPeriphID2, 类型 RO, 偏移量 0xFE8, 复位 0x0000.00B8</b>															
												PID2			
<b>DMAPeriphID3, 类型 RO, 偏移量 0xFEC, 复位 0x0000.0000</b>															
												PID3			
<b>DMAPeriphID4, 类型 RO, 偏移量 0xFD0, 复位 0x0000.0004</b>															
												PID4			
<b>DMAPCellID0, 类型 RO, 偏移量 0xFF0, 复位 0x0000.000D</b>															
												CID0			
<b>DMAPCellID1, 类型 RO, 偏移量 0xFF4, 复位 0x0000.00F0</b>															
												CID1			
<b>DMAPCellID2, 类型 RO, 偏移量 0xFF8, 复位 0x0000.0005</b>															
												CID2			
<b>DMAPCellID3, 类型 RO, 偏移量 0xFFC, 复位 0x0000.00B1</b>															
												CID3			



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>通用输入/输出端口 (GPIO)</b>															
GPIO 端口 A (APB) 基址: 0x4000.4000															
GPIO 端口 A (AHB) 基址: 0x4005.8000															
GPIO 端口 B (APB) 基址: 0x4000.5000															
GPIO 端口 B (AHB) 基址: 0x4005.9000															
GPIO 端口 C (APB) 基址: 0x4000.6000															
GPIO 端口 C (AHB) 基址: 0x4005.A000															
GPIO 端口 D (APB) 基址: 0x4000.7000															
GPIO 端口 D (AHB) 基址: 0x4005.B000															
GPIO 端口 E (APB) 基址: 0x4002.4000															
GPIO 端口 E (AHB) 基址: 0x4005.C000															
GPIO 端口 F (APB) 基址: 0x4002.5000															
GPIO 端口 F (AHB) 基址: 0x4005.D000															
GPIO 端口 G (APB) 基址: 0x4002.6000															
GPIO 端口 G (AHB) 基址: 0x4005.E000															
GPIO 端口 H (APB) 基址: 0x4002.7000															
GPIO 端口 H (AHB) 基址: 0x4005.F000															
GPIO 端口 J (APB) 基址: 0x4003.D000															
GPIO 端口 J (AHB) 基址: 0x4006.0000															
GPIO 端口 K (AHB) 基址: 0x4006.1000															
GPIO 端口 L (AHB) 基址: 0x4006.2000															
GPIO 端口 M (AHB) 基址: 0x4006.3000															
GPIO 端口 N (AHB) 基址: 0x4006.4000															
GPIO 端口 P (AHB) 基址: 0x4006.5000															
GPIODATA, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000 (见第 609 页)															
DATA															
GPIODIR, 类型 R/W, 偏移量 0x400, 复位 0x0000.0000 (见第 610 页)															
DIR															
GPIOIS, 类型 R/W, 偏移量 0x404, 复位 0x0000.0000 (见第 611 页)															
IS															
GPIOIBE, 类型 R/W, 偏移量 0x408, 复位 0x0000.0000 (见第 612 页)															
IBE															
GPIOIEV, 类型 R/W, 偏移量 0x40C, 复位 0x0000.0000 (见第 613 页)															
IEV															
GPIOIM, 类型 R/W, 偏移量 0x410, 复位 0x0000.0000 (见第 614 页)															
IME															
GPIOIRIS, 类型 RO, 偏移量 0x414, 复位 0x0000.0000 (见第 615 页)															
RIS															
GPIOMIS, 类型 RO, 偏移量 0x418, 复位 0x0000.0000 (见第 616 页)															
MIS															
GPIOICR, 类型 W1C, 偏移量 0x41C, 复位 0x0000.0000 (见第 617 页)															
IC															
GPIOAFSEL, 类型 R/W, 偏移量 0x420, 复位 - (见第 618 页)															
AFSEL															
GPIODR2R, 类型 R/W, 偏移量 0x500, 复位 0x0000.00FF (见第 620 页)															
DRV2															
GPIODR4R, 类型 R/W, 偏移量 0x504, 复位 0x0000.0000 (见第 621 页)															
DRV4															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIODR8R, 类型 R/W, 偏移量 0x508, 复位 0x0000.0000 ( 见第 622 页 )															
DRV8															
GPIOODR, 类型 R/W, 偏移量 0x50C, 复位 0x0000.0000 ( 见第 623 页 )															
ODE															
GPIOPUR, 类型 R/W, 偏移量 0x510, 复位 - ( 见第 624 页 )															
PUE															
GPIOPDR, 类型 R/W, 偏移量 0x514, 复位 0x0000.0000 ( 见第 626 页 )															
PDE															
GPIOSLR, 类型 R/W, 偏移量 0x518, 复位 0x0000.0000 ( 见第 628 页 )															
SRL															
GPIODEN, 类型 R/W, 偏移量 0x51C, 复位 - ( 见第 629 页 )															
DEN															
GPIOLOCK, 类型 R/W, 偏移量 0x520, 复位 0x0000.0001 ( 见第 631 页 )															
LOCK															
LOCK															
GPIOCR, 类型 -, 偏移量 0x524, 复位 - ( 见第 632 页 )															
CR															
GPIOAMSEL, 类型 R/W, 偏移量 0x528, 复位 0x0000.0000 ( 见第 634 页 )															
GPIOAMSEL															
GPIOPCTL, 类型 R/W, 偏移量 0x52C, 复位 - ( 见第 636 页 )															
PMC7				PMC6				PMC5				PMC4			
PMC3				PMC2				PMC1				PMC0			
GPIOADCCTL, 类型 R/W, 偏移量 0x530, 复位 0x0000.0000 ( 见第 638 页 )															
ADCEN															
GPIODMACTL, 类型 R/W, 偏移量 0x534, 复位 0x0000.0000 ( 见第 639 页 )															
DMAEN															
GPIOI, 类型 R/W, 偏移量 0x538, 复位 0x0000.0000 ( 见第 640 页 )															
SUM															
GPIOPeriphID4, 类型 RO, 偏移量 0xFD0, 复位 0x0000.0000 ( 见第 641 页 )															
PID4															
GPIOPeriphID5, 类型 RO, 偏移量 0xFD4, 复位 0x0000.0000 ( 见第 642 页 )															
PID5															
GPIOPeriphID6, 类型 RO, 偏移量 0xFD8, 复位 0x0000.0000 ( 见第 643 页 )															
PID6															
GPIOPeriphID7, 类型 RO, 偏移量 0xFDC, 复位 0x0000.0000 ( 见第 644 页 )															
PID7															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIOPeriphID0, 类型 RO, 偏移量 0xFE0, 复位 0x0000.0061 (见第 645 页)															
PID0															
GPIOPeriphID1, 类型 RO, 偏移量 0xFE4, 复位 0x0000.0000 (见第 646 页)															
PID1															
GPIOPeriphID2, 类型 RO, 偏移量 0xFE8, 复位 0x0000.0018 (见第 647 页)															
PID2															
GPIOPeriphID3, 类型 RO, 偏移量 0xFEC, 复位 0x0000.0001 (见第 648 页)															
PID3															
GPIOCellID0, 类型 RO, 偏移量 0xFF0, 复位 0x0000.000D (见第 649 页)															
CID0															
GPIOCellID1, 类型 RO, 偏移量 0xFF4, 复位 0x0000.00F0 (见第 650 页)															
CID1															
GPIOCellID2, 类型 RO, 偏移量 0xFF8, 复位 0x0000.0005 (见第 651 页)															
CID2															
GPIOCellID3, 类型 RO, 偏移量 0xFFC, 复位 0x0000.00B1 (见第 652 页)															
CID3															
通用定时器															
16/32 位Timer 0 基址: 0x4003.0000															
16/32 位Timer 1 基址: 0x4003.1000															
16/32 位Timer 2 基址: 0x4003.2000															
16/32 位Timer 3 基址: 0x4003.3000															
16/32 位Timer 4 基址: 0x4003.4000															
16/32 位Timer 5 基址: 0x4003.5000															
32/64 位宽 Timer 0 基址: 0x4003.6000															
32/64 位宽 Timer 1 基址: 0x4003.7000															
32/64 位宽 Timer 2 基址: 0x4004.C000															
32/64 位宽 Timer 3 基址: 0x4004.D000															
32/64 位宽 Timer 4 基址: 0x4004.E000															
32/64 位宽 Timer 5 基址: 0x4004.F000															
GPTMCFG, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000 (见第 675 页)															
GPTMCFG															
GPTMTAMR, 类型 R/W, 偏移量 0x004, 复位 0x0000.0000 (见第 676 页)															
TAPLO TAMRSU TAPWMIETAILD TASNAPS TAWOT TAMIE TACDIR TAAMS TACMR TAMR															
GPTMTBMR, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000 (见第 679 页)															
TBPLO TBMRSU TBPWMIETBILD TBSNAPS TBWOT TBMIE TBCDIR TBAMS TBCMR TBMR															
GPTMCTL, 类型 R/W, 偏移量 0x00C, 复位 0x0000.0000 (见第 682 页)															
TBPWML TBOTE TBEVENT TBSTALL TBEN TAPWML TAOTE RTCEN TAEVENT TASTALL TAEN															
GPTMSYNC, 类型 R/W, 偏移量 0x010, 复位 0x0000.0000 (见第 685 页)															
SYNCWT1 SYNCWT0 SYNCT5 SYNCT4 SYNCT3 SYNCT2 SYNCT1 SYNCT0															
GPTMIMR, 类型 R/W, 偏移量 0x018, 复位 0x0000.0000 (见第 688 页)															
TBMIM CBEIM CBMIM TBOIM TAMIM RTCIM CAEIM CAMIM TATOIM WUEIM															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
GPTMRIS, 类型 RO, 偏移量 0x01C, 复位 0x0000.0000 (见第 691 页)																							
												WUERIS											
				TBMRIS	CBERIS	CBMRIS	TBTORIS					TAMRIS	RTCRIIS	CAERIS	CAMRIS	TATORIS							
GPTMMIS, 类型 RO, 偏移量 0x020, 复位 0x0000.0000 (见第 694 页)																							
												WUEMIS											
				TBMMIS	CBEMIS	CBMMIS	TBTOMIS					TAMMIS	RTCMIS	CAEMIS	CAMMIS	TATOMIS							
GPTMICR, 类型 W1C, 偏移量 0x024, 复位 0x0000.0000 (见第 697 页)																							
												WUECINT											
				TBMCINT	CBECINT	CBMCINT	TBTCINT					TAMCINT	RTCCINT	CAECINT	CAMCINT	TATOCINT							
GPTMTAILR, 类型 R/W, 偏移量 0x028, 复位 0xFFFF.FFFF (见第 699 页)																							
												TAILR											
												TAILR											
GPTMTBILR, 类型 R/W, 偏移量 0x02C, 复位 - (见第 700 页)																							
												TBILR											
												TBILR											
GPTMTAMATCHR, 类型 R/W, 偏移量 0x030, 复位 0xFFFF.FFFF (见第 701 页)																							
												TAMR											
												TAMR											
GPTMTBMATCHR, 类型 R/W, 偏移量 0x034, 复位 - (见第 702 页)																							
												TBMR											
												TBMR											
GPTMTAPR, 类型 R/W, 偏移量 0x038, 复位 0x0000.0000 (见第 703 页)																							
								TAPSRH								TAPSR							
GPTMTBPR, 类型 R/W, 偏移量 0x03C, 复位 0x0000.0000 (见第 704 页)																							
								TBPSRH								TBPSR							
GPTMTAPMR, 类型 R/W, 偏移量 0x040, 复位 0x0000.0000 (见第 705 页)																							
								TAPSMRH								TAPSMR							
GPTMTBPMR, 类型 R/W, 偏移量 0x044, 复位 0x0000.0000 (见第 706 页)																							
								TBPSMRH								TBPSMR							
GPTMTAPR, 类型 RO, 偏移量 0x048, 复位 0xFFFF.FFFF (见第 707 页)																							
												TAR											
												TAR											
GPTMTBPR, 类型 RO, 偏移量 0x04C, 复位 - (见第 708 页)																							
												TBR											
												TBR											
GPTMTAV, 类型 RW, 偏移量 0x050, 复位 0xFFFF.FFFF (见第 709 页)																							
												TAV											
												TAV											
GPTMTBV, 类型 RW, 偏移量 0x054, 复位 - (见第 710 页)																							
												TBV											
												TBV											
GPTMRTCPD, 类型 RO, 偏移量 0x058, 复位 0x0000.7FFF (见第 711 页)																							
												RTCPD											
GPTMTAPS, 类型 RO, 偏移量 0x05C, 复位 0x0000.0000 (见第 712 页)																							
												PSS											

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPTMTBPS, 类型 RO, 偏移量 0x060, 复位 0x0000.0000 (见第 713 页)															
PSS															
GPTMTAPV, 类型 RO, 偏移量 0x064, 复位 0x0000.0000 (见第 714 页)															
PSV															
GPTMTBPV, 类型 RO, 偏移量 0x068, 复位 0x0000.0000 (见第 715 页)															
PSV															
GPTMPP, 类型 RO, 偏移量 0xFC0, 复位 0x0000.0000 (见第 716 页)															
SIZE															
<b>看门狗定时器 (WDT)</b>															
WDT0 基址: 0x4000.0000															
WDT1 基址: 0x4000.1000															
WDTLOAD, 类型 R/W, 偏移量 0x000, 复位 0xFFFF.FFFF (见第 721 页)															
WDTLOAD															
WDTLOAD															
WDTVALUE, 类型 RO, 偏移量 0x004, 复位 0xFFFF.FFFF (见第 722 页)															
WDTVALUE															
WDTVALUE															
WDTCTL, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000 (WDT0) 和 0x8000.0000 (WDT1) (见第 723 页)															
WRC															
													INTTYPE	RESEN	INTEN
WDTICR, 类型 WO, 偏移量 0x00C, 复位 - (见第 725 页)															
WDTINTCLR															
WDTINTCLR															
WDTRIS, 类型 RO, 偏移量 0x010, 复位 0x0000.0000 (见第 726 页)															
WDTRIS															
WDTMIS, 类型 RO, 偏移量 0x014, 复位 0x0000.0000 (见第 727 页)															
WDTMIS															
WDTTEST, 类型 R/W, 偏移量 0x418, 复位 0x0000.0000 (见第 728 页)															
STALL															
WDTLOCK, 类型 R/W, 偏移量 0xC00, 复位 0x0000.0000 (见第 729 页)															
WDTLOCK															
WDTLOCK															
WDTPeriphID4, 类型 RO, 偏移量 0xFD0, 复位 0x0000.0000 (见第 730 页)															
PID4															
WDTPeriphID5, 类型 RO, 偏移量 0xFD4, 复位 0x0000.0000 (见第 731 页)															
PID5															
WDTPeriphID6, 类型 RO, 偏移量 0xFD8, 复位 0x0000.0000 (见第 732 页)															
PID6															
WDTPeriphID7, 类型 RO, 偏移量 0xFDC, 复位 0x0000.0000 (见第 733 页)															
PID7															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTPeriphID0, 类型 RO, 偏移量 0xFE0, 复位 0x0000.0005 (见第 734 页)															
												PID0			
WDTPeriphID1, 类型 RO, 偏移量 0xFE4, 复位 0x0000.0018 (见第 735 页)															
												PID1			
WDTPeriphID2, 类型 RO, 偏移量 0xFE8, 复位 0x0000.0018 (见第 736 页)															
												PID2			
WDTPeriphID3, 类型 RO, 偏移量 0xFEC, 复位 0x0000.0001 (见第 737 页)															
												PID3			
WDTPCellID0, 类型 RO, 偏移量 0xFF0, 复位 0x0000.000D (见第 738 页)															
												CID0			
WDTPCellID1, 类型 RO, 偏移量 0xFF4, 复位 0x0000.00F0 (见第 739 页)															
												CID1			
WDTPCellID2, 类型 RO, 偏移量 0xFF8, 复位 0x0000.0006 (见第 740 页)															
												CID2			
WDTPCellID3, 类型 RO, 偏移量 0xFFC, 复位 0x0000.00B1 (见第 741 页)															
												CID3			
<b>模数转换器 (ADC)</b>															
ADC0 基址: 0x4003.8000															
ADC1 基址: 0x4003.9000															
ADCACTSS, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000 (见第 763 页)															
												ASEN3	ASEN2	ASEN1	ASEN0
ADCRIS, 类型 RO, 偏移量 0x004, 复位 0x0000.0000 (见第 764 页)															
												INR3	INR2	INR1	INR0
ADCIM, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000 (见第 766 页)															
												DCONSS3	DCONSS2	DCONSS1	DCONSS0
												MASK3	MASK2	MASK1	MASK0
ADCISC, 类型 R/W1C, 偏移量 0x00C, 复位 0x0000.0000 (见第 768 页)															
												DCINSS3	DCINSS2	DCINSS1	DCINSS0
												IN3	IN2	IN1	IN0
ADCOSTAT, 类型 R/W1C, 偏移量 0x010, 复位 0x0000.0000 (见第 771 页)															
												OV3	OV2	OV1	OV0
ADCEMUX, 类型 R/W, 偏移量 0x014, 复位 0x0000.0000 (见第 773 页)															
EM3				EM2				EM1				EM0			
ADCUSTAT, 类型 R/W1C, 偏移量 0x018, 复位 0x0000.0000 (见第 778 页)															
												UV3	UV2	UV1	UV0
ADCTSSEL, 类型 R/W, 偏移量 0x01C, 复位 0x0000.0000 (见第 779 页)															
				PS3				PS2							
				PS1				PS0							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
ADCSSPRI, 类型 R/W, 偏移量 0x020, 复位 0x0000.3210 (见第 781 页)																															
SS3				SS2				SS1				SS0																			
ADCSPC, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000 (见第 782 页)																															
PHASE																															
ADCPSSI, 类型 R/W, 偏移量 0x028, 复位 - (见第 783 页)																															
GSYNC				SYNCWAIT				SS3				SS2				SS1				SS0											
ADCSSAC, 类型 R/W, 偏移量 0x030, 复位 0x0000.0000 (见第 785 页)																															
AVG																															
ADCDCISC, 类型 R/W1C, 偏移量 0x034, 复位 0x0000.0000 (见第 786 页)																															
DCINT7				DCINT6				DCINT5				DCINT4				DCINT3				DCINT2				DCINT1				DCINT0			
ADCCTL, 类型 R/W, 偏移量 0x038, 复位 0x0000.0010 (见第 788 页)																															
VREF																															
ADCSSMUX0, 类型 R/W, 偏移量 0x040, 复位 0x0000.0000 (见第 789 页)																															
MUX7				MUX6				MUX5				MUX4																			
MUX3				MUX2				MUX1				MUX0																			
ADCSSCTL0, 类型 R/W, 偏移量 0x044, 复位 0x0000.0000 (见第 791 页)																															
TS7		IE7		END7		D7		TS6		IE6		END6		D6		TS5		IE5		END5		D5		TS4		IE4		END4		D4	
TS3		IE3		END3		D3		TS2		IE2		END2		D2		TS1		IE1		END1		D1		TS0		IE0		END0		D0	
ADCSSFIFO0, 类型 RO, 偏移量 0x048, 复位 - (见第 794 页)																															
DATA																															
ADCSSFIFO1, 类型 RO, 偏移量 0x068, 复位 - (见第 794 页)																															
DATA																															
ADCSSFIFO2, 类型 RO, 偏移量 0x088, 复位 - (见第 794 页)																															
DATA																															
ADCSSFIFO3, 类型 RO, 偏移量 0x0A8, 复位 - (见第 794 页)																															
DATA																															
ADCSSFSTAT0, 类型 RO, 偏移量 0x04C, 复位 0x0000.0100 (见第 795 页)																															
FULL				EMPTY				HPTR				TPTR																			
ADCSSFSTAT1, 类型 RO, 偏移量 0x06C, 复位 0x0000.0100 (见第 795 页)																															
FULL				EMPTY				HPTR				TPTR																			
ADCSSFSTAT2, 类型 RO, 偏移量 0x08C, 复位 0x0000.0100 (见第 795 页)																															
FULL				EMPTY				HPTR				TPTR																			
(ADCSSFSTAT3, 类型 RO, 偏移量 0x0AC, 复位 0x0000.0100 (见第 795 页)																															
FULL				EMPTY				HPTR				TPTR																			
ADCSSOP0, 类型 R/W, 偏移量 0x050, 复位 0x0000.0000 (见第 797 页)																															
S7DCOP				S6DCOP				S5DCOP				S4DCOP																			
S3DCOP				S2DCOP				S1DCOP				S0DCOP																			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ADCSSDC0, 类型 R/W, 偏移量 0x054, 复位 0x0000.0000 (见第 799 页)																			
S7DCSEL				S6DCSEL				S5DCSEL				S4DCSEL							
S3DCSEL				S2DCSEL				S1DCSEL				S0DCSEL							
ADCSEMUX0, 类型 R/W, 偏移量 0x058, 复位 0x0000.0000 (见第 801 页)																			
EMUX7				EMUX6				EMUX5				EMUX4							
EMUX3				EMUX2				EMUX1				EMUX0							
ADCSSMUX1, 类型 R/W, 偏移量 0x060, 复位 0x0000.0000 (见第 803 页)																			
MUX3				MUX2				MUX1				MUX0							
ADCSSMUX2, 类型 R/W, 偏移量 0x080, 复位 0x0000.0000 (见第 803 页)																			
MUX3				MUX2				MUX1				MUX0							
ADCSSCTL1, 类型 R/W, 偏移量 0x064, 复位 0x0000.0000 (见第 804 页)																			
TS3		IE3		END3		D3		TS2		IE2		END2		D2					
TS1		IE1		END1		D1		TS0		IE0		END0		D0					
ADCSSCTL2, 类型 R/W, 偏移量 0x084, 复位 0x0000.0000 (见第 804 页)																			
TS3		IE3		END3		D3		TS2		IE2		END2		D2					
TS1		IE1		END1		D1		TS0		IE0		END0		D0					
ADCSSOP1, 类型 R/W, 偏移量 0x070, 复位 0x0000.0000 (见第 806 页)																			
S3DCOP				S2DCOP				S1DCOP				S0DCOP							
ADCSSOP2, 类型 R/W, 偏移量 0x090, 复位 0x0000.0000 (见第 806 页)																			
S3DCOP				S2DCOP				S1DCOP				S0DCOP							
ADCSSDC1, 类型 R/W, 偏移量 0x074, 复位 0x0000.0000 (见第 807 页)																			
S3DCSEL				S2DCSEL				S1DCSEL				S0DCSEL							
ADCSSDC2, 类型 R/W, 偏移量 0x094, 复位 0x0000.0000 (见第 807 页)																			
S3DCSEL				S2DCSEL				S1DCSEL				S0DCSEL							
ADCSEMUX1, 类型 R/W, 偏移量 0x078, 复位 0x0000.0000 (见第 808 页)																			
EMUX3				EMUX2				EMUX1				EMUX0							
ADCSEMUX2, 类型 R/W, 偏移量 0x098, 复位 0x0000.0000 (见第 808 页)																			
EMUX3				EMUX2				EMUX1				EMUX0							
ADCSSMUX3, 类型 R/W, 偏移量 0x0A0, 复位 0x0000.0000 (见第 810 页)																			
												MUX0							
ADCSSCTL3, 类型 R/W, 偏移量 0x0A4, 复位 0x0000.0000 (见第 811 页)																			
												TS0		IE0		END0		D0	
ADCSSOP3, 类型 R/W, 偏移量 0x0B0, 复位 0x0000.0000 (见第 812 页)																			
												S0DCOP							
ADCSSDC3, 类型 R/W, 偏移量 0x0B4, 复位 0x0000.0000 (见第 813 页)																			
												S0DCSEL							
ADCSSMUX3, 类型 R/W, 偏移量 0x0B8, 复位 0x0000.0000 (见第 814 页)																			
												EMUX0							



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDCR1C, 类型 WO, 偏移量 0xD00, 复位 0x0000.0000 (见第 815 页)															
								DCTRIG7	DCTRIG6	DCTRIG5	DCTRIG4	DCTRIG3	DCTRIG2	DCTRIG1	DCTRIG0
								DCINT7	DCINT6	DCINT5	DCINT4	DCINT3	DCINT2	DCINT1	DCINT0
ADDCCTL0, 类型 R/W, 偏移量 0xE00, 复位 0x0000.0000 (见第 819 页)															
				CTE				CTC				CTM			
								CIE				CIC			
												CIM			
ADDCCTL1, 类型 R/W, 偏移量 0xE04, 复位 0x0000.0000 (见第 819 页)															
				CTE				CTC				CTM			
								CIE				CIC			
												CIM			
ADDCCTL2, 类型 R/W, 偏移量 0xE08, 复位 0x0000.0000 (见第 819 页)															
				CTE				CTC				CTM			
								CIE				CIC			
												CIM			
ADDCCTL3, 类型 R/W, 偏移量 0xE0C, 复位 0x0000.0000 (见第 819 页)															
				CTE				CTC				CTM			
								CIE				CIC			
												CIM			
ADDCCTL4, 类型 R/W, 偏移量 0xE10, 复位 0x0000.0000 (见第 819 页)															
				CTE				CTC				CTM			
								CIE				CIC			
												CIM			
ADDCCTL5, 类型 R/W, 偏移量 0xE14, 复位 0x0000.0000 (见第 819 页)															
				CTE				CTC				CTM			
								CIE				CIC			
												CIM			
ADDCCTL6, 类型 R/W, 偏移量 0xE18, 复位 0x0000.0000 (见第 819 页)															
				CTE				CTC				CTM			
								CIE				CIC			
												CIM			
ADDCCTL7, 类型 R/W, 偏移量 0xE1C, 复位 0x0000.0000 (见第 819 页)															
				CTE				CTC				CTM			
								CIE				CIC			
												CIM			
ADDCCMP0, 类型 R/W, 偏移量 0xE40, 复位 0x0000.0000 (见第 822 页)															
								COMP1							
								COMP0							
ADDCCMP1, 类型 R/W, 偏移量 0xE44, 复位 0x0000.0000 (见第 822 页)															
								COMP1							
								COMP0							
ADDCCMP2, 类型 R/W, 偏移量 0xE48, 复位 0x0000.0000 (见第 822 页)															
								COMP1							
								COMP0							
ADDCCMP3, 类型 R/W, 偏移量 0xE4C, 复位 0x0000.0000 (见第 822 页)															
								COMP1							
								COMP0							
ADDCCMP4, 类型 R/W, 偏移量 0xE50, 复位 0x0000.0000 (见第 822 页)															
								COMP1							
								COMP0							
ADDCCMP5, 类型 R/W, 偏移量 0xE54, 复位 0x0000.0000 (见第 822 页)															
								COMP1							
								COMP0							
ADDCCMP6, 类型 R/W, 偏移量 0xE58, 复位 0x0000.0000 (见第 822 页)															
								COMP1							
								COMP0							
ADDCCMP7, 类型 R/W, 偏移量 0xE5C, 复位 0x0000.0000 (见第 822 页)															
								COMP1							
								COMP0							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																																												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																												
ADCPP, 类型 RO, 偏移量 0xFC0, 复位 0x00B0.2187 ( 见第 823 页 )																																																											
DC								TS				RSL				类型																																											
								CH				MSR																																															
ADPCPC, 类型 R/W, 偏移量 0xFC4, 复位 0x0000.0007 ( 见第 825 页 )																																																											
SR																																																											
ADCCC, 类型 R/W, 偏移量 0xFC8, 复位 0x0000.0000 ( 见第 826 页 )																																																											
CS																																																											
通用异步收发器 (UART)																																																											
UART0:0x4000.C000 ( 0x4000.CFFF的结束地址 ) 基址: 0x4000.C000																																																											
UART1:0x4000.D000 ( 0x4000.DFFF的结束地址 ) 基址: 0x4000.D000																																																											
UART2:0x4000.E000 ( 0x4000.EFFF的结束地址 ) 基址: 0x4000.E000																																																											
UART3 基址: 0x4000.F000																																																											
UART4 基址: 0x4001.0000																																																											
UART5 基址: 0x4001.1000																																																											
UART6 基址: 0x4001.2000																																																											
UART7 基址: 0x4001.3000																																																											
UARTDR, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000 ( 见第 840 页 )																																																											
								OE				BE				PE				FE				DATA																																			
UARTSR/UARTECR, 类型 RO, 偏移量 0x004, 复位 0x0000.0000 ( 只读状态寄存器 ) ( 见第 842 页 )																																																											
OE																BE				PE				FE																																			
UARTSR/UARTECR, 类型 WO, 偏移量 0x004, 复位 0x0000.0000 ( 只写错误清除寄存器 ) ( 见第 842 页 )																																																											
DATA																																																											
UARTFR, 类型 RO, 偏移量 0x018, 复位 0x0000.0090 ( 见第 845 页 )																																																											
								RI				TXFE				RXFF				TXFF				RXFE				BUSY				DCD				DSR				CTS																			
UARTILPR, 类型 R/W, 偏移量 0x020, 复位 0x0000.0000 ( 见第 847 页 )																																																											
ILPDVSR																																																											
UARTIBRD, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000 ( 见第 848 页 )																																																											
DIVINT																																																											
UARTFBRD, 类型 R/W, 偏移量 0x028, 复位 0x0000.0000 ( 见第 849 页 )																																																											
DIVFRAC																																																											
UARTLCRH, 类型 R/W, 偏移量 0x02C, 复位 0x0000.0000 ( 见第 850 页 )																																																											
								SPS				WLEN				FEN				STP2				EPS				PEN				BRK																											
UARTCTL, 类型 R/W, 偏移量 0x030, 复位 0x0000.0300 ( 见第 852 页 )																																																											
CTSEN				RTSEN				RTS				DTR				RXE				TXE				LBE				LIN				HSE				EOT				SMART				SIRLP				SIREN				UARTEN							
UARTIFLS, 类型 R/W, 偏移量 0x034, 复位 0x0000.0012 ( 见第 856 页 )																																																											
								RXIFLSEL								TXIFLSEL																																											
UARTIM, 类型 R/W, 偏移量 0x038, 复位 0x0000.0000 ( 见第 858 页 )																																																											
LME5IM				LME1IM				LMSBIM				9BITIM				OEIM				BEIM				PEIM				FEIM				RTIM				TXIM				RXIM				DSRIM				DCDIM				CTSIM				RIIM			
UARTRIS, 类型 RO, 偏移量 0x03C, 复位 0x0000.000F ( 见第 861 页 )																																																											
LME5RIS				LME1RIS				LMSBRIS				9BITRIS				OERIS				BERIS				PERIS				FERIS				RTRIS				TXRIS				RXRIS				DSRRIS				DCDRIS				CTSRIS				RIRIS			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UARTMIS, 类型 RO, 偏移量 0x040, 复位 0x0000.0000 ( 见第 865 页 )															
LME5MIS	LME1MIS	LMSBMIS	9BITMIS		OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS	DSRMIS	DCDMIS	CTSMIS	RIMIS
UARTICR, 类型 W1C, 偏移量 0x044, 复位 0x0000.0000 ( 见第 869 页 )															
LME5IC	LME1IC	LMSBIC	9BITIC		OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	DSRMIC	DCDMIC	CTSMIC	RIMIC
UARTDMACTL, 类型 R/W, 偏移量 0x048, 复位 0x0000.0000 ( 见第 871 页 )															
													DMAERR	TXDMAE	RXDMAE
UARTLCTL, 类型 R/W, 偏移量 0x090, 复位 0x0000.0000 ( 见第 872 页 )															
										BLER					MASTER
UARTLSS, 类型 RO, 偏移量 0x094, 复位 0x0000.0000 ( 见第 873 页 )															
															TSS
UARTLTIM, 类型 RO, 偏移量 0x098, 复位 0x0000.0000 ( 见第 874 页 )															
															TIMER
UART9BITADDR, 类型 R/W, 偏移量 0x0A4, 复位 0x0000.0000 ( 见第 875 页 )															
9BITEN															ADDR
UART9BITAMASK, 类型 R/W, 偏移量 0x0A8, 复位 0x0000.00FF ( 见第 876 页 )															
															MASK
UARTPP, 类型 RO, 偏移量 0xFC0, 复位 0x0000.0003 ( 见第 877 页 )															
														NB	SC
UARTCC, 类型 R/W, 偏移量 0xFC8, 复位 0x0000.0000 ( 见第 878 页 )															
															CS
UARTPeriphID4, 类型 RO, 偏移量 0xFD0, 复位 0x0000.0000 ( 见第 879 页 )															
															PID4
UARTPeriphID5, 类型 RO, 偏移量 0xFD4, 复位 0x0000.0000 ( 见第 880 页 )															
															PID5
UARTPeriphID6, 类型 RO, 偏移量 0xFD8, 复位 0x0000.0000 ( 见第 881 页 )															
															PID6
UARTPeriphID7, 类型 RO, 偏移量 0xFDC, 复位 0x0000.0000 ( 见第 882 页 )															
															PID7
UARTPeriphID0, 类型 RO, 偏移量 0xFE0, 复位 0x0000.0060 ( 见第 883 页 )															
															PID0
UARTPeriphID1, 类型 RO, 偏移量 0xFE4, 复位 0x0000.0000 ( 见第 884 页 )															
															PID1
UARTPeriphID2, 类型 RO, 偏移量 0xFE8, 复位 0x0000.0018 ( 见第 885 页 )															
															PID2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
UARTPeriphID3, 类型 RO, 偏移量 0xFEC, 复位 0x0000.0001 ( 见第 886 页 )																							
												PID3											
UARTPCellIID0, 类型 RO, 偏移量 0xFF0, 复位 0x0000.000D ( 见第 887 页 )																							
												CID0											
UARTPCellIID1, 类型 RO, 偏移量 0xFF4, 复位 0x0000.00F0 ( 见第 888 页 )																							
												CID1											
UARTPCellIID2, 类型 RO, 偏移量 0xFF8, 复位 0x0000.0005 ( 见第 889 页 )																							
												CID2											
UARTPCellIID3, 类型 RO, 偏移量 0xFFC, 复位 0x0000.00B1 ( 见第 890 页 )																							
												CID3											
同步串行接口 (SSI) SSI0 基址: 0x4000.8000 SSI1 基址: 0x4000.9000 SSI2 基址: 0x4000.A000 SSI3 基址: 0x4000.B000																							
SSICR0, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000 ( 见第 905 页 )																							
SCR				SPH				SPO				FRF				DSS							
SSICR1, 类型 R/W, 偏移量 0x004, 复位 0x0000.0000 ( 见第 907 页 )																							
												SLBY6		EOT		SOD		MS		SSE		LBM	
SSIDR, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000 ( 见第 909 页 )																							
DATA																							
SSISR, 类型 RO, 偏移量 0x00C, 复位 0x0000.0003 ( 见第 910 页 )																							
												BSY		RFF		RNE		TNF		TFE			
SSICPSR, 类型 R/W, 偏移量 0x010, 复位 0x0000.0000 ( 见第 912 页 )																							
CPSDVSR																							
SSIIM, 类型 R/W, 偏移量 0x014, 复位 0x0000.0000 ( 见第 913 页 )																							
												TXIM		RXIM		RTIM		RORIM					
SSIRIS, 类型 RO, 偏移量 0x018, 复位 0x0000.0008 ( 见第 914 页 )																							
												TXRIS		RXRIS		RTRIS		RORRIS					
SSIMIS, 类型 RO, 偏移量 0x01C, 复位 0x0000.0000 ( 见第 916 页 )																							
												TXMIS		RXMIS		RTMIS		RORMIS					
SSIICR, 类型 W1C, 偏移量 0x020, 复位 0x0000.0000 ( 见第 918 页 )																							
														RTIC									
SSIDMACTL, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000 ( 见第 919 页 )																							
														TXDMAE									
SSICC, 类型 R/W, 偏移量 0xFC8, 复位 0x0000.0000 ( 见第 920 页 )																							
CS																							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSIPeriphID4, 类型 RO, 偏移量 0xFD0, 复位 0x0000.0000 (见第 921 页)															
												PID4			
SSIPeriphID5, 类型 RO, 偏移量 0xFD4, 复位 0x0000.0000 (见第 922 页)															
												PID5			
SSIPeriphID6, 类型 RO, 偏移量 0xFD8, 复位 0x0000.0000 (见第 923 页)															
												PID6			
SSIPeriphID7, 类型 RO, 偏移量 0xFDC, 复位 0x0000.0000 (见第 924 页)															
												PID7			
SSIPeriphID0, 类型 RO, 偏移量 0xFE0, 复位 0x0000.0022 (见第 925 页)															
												PID0			
SSIPeriphID1, 类型 RO, 偏移量 0xFE4, 复位 0x0000.0000 (见第 926 页)															
												PID1			
SSIPeriphID2, 类型 RO, 偏移量 0xFE8, 复位 0x0000.0018 (见第 927 页)															
												PID2			
SSIPeriphID3, 类型 RO, 偏移量 0xFEC, 复位 0x0000.0001 (见第 928 页)															
												PID3			
SSIPCellID0, 类型 RO, 偏移量 0xFF0, 复位 0x0000.000D (见第 929 页)															
												CID0			
SSIPCellID1, 类型 RO, 偏移量 0xFF4, 复位 0x0000.00F0 (见第 930 页)															
												CID1			
SSIPCellID2, 类型 RO, 偏移量 0xFF8, 复位 0x0000.0005 (见第 931 页)															
												CID2			
SSIPCellID3, 类型 RO, 偏移量 0xFFC, 复位 0x0000.00B1 (见第 932 页)															
												CID3			
内部集成电路 (I <sup>2</sup> C) 接口															
I <sup>2</sup> C 主机															
I2C 0 基址: 0x4002.0000															
I2C 1 基址: 0x4002.1000															
I2C 2 基址: 0x4002.2000															
I2C 3 基址: 0x4002.3000															
I2C 4 基址: 0x400C.0000															
I2C 5 基址: 0x400C.1000															
I2CMSA, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000															
												SA			
I2CMCS, 类型 RO, 偏移量 0x004, 复位 0x0000.0020 (只读状态寄存器)															
								CLKTO	BUSBSY	IDLE	ARBLST	DATAACK	ADRACK	ERROR	BUSY
I2CMCS, 类型 WO, 偏移量 0x004, 复位 0x0000.0020 (只写控制寄存器)															
										HS	ACK	STOP	START	RUN	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>I2CMDR, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000</b>															
DATA															
<b>I2CMTPR, 类型 R/W, 偏移量 0x00C, 复位 0x0000.0001</b>															
HS TPR															
<b>I2CMIMR, 类型 R/W, 偏移量 0x010, 复位 0x0000.0000</b>															
CLKIM IM															
<b>I2CMRIS, 类型 RO, 偏移量 0x014, 复位 0x0000.0000</b>															
CLKRIS RIS															
<b>I2CMMIS, 类型 RO, 偏移量 0x018, 复位 0x0000.0000</b>															
CLKMIS MIS															
<b>I2CMICR, 类型 WO, 偏移量 0x01C, 复位 0x0000.0000</b>															
CLKIC IC															
<b>I2CMCR, 类型 R/W, 偏移量 0x020, 复位 0x0000.0000</b>															
SFE MFE LPBK															
<b>I2CMCLKOCNT, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000</b>															
CNTL															
<b>I2CMBMON, 类型 RO, 偏移量 0x02C, 复位 0x0000.0003</b>															
SDA SCL															
<b>内部集成电路 (I<sup>2</sup>C) 接口</b> <b>I<sup>2</sup>C 从机</b> I2C 0 基址: 0x4002.0000 I2C 1 基址: 0x4002.1000 I2C 2 基址: 0x4002.2000 I2C 3 基址: 0x4002.3000 I2C 4 基址: 0x400C.0000 I2C 5 基址: 0x400C.1000															
<b>I2CSOAR, 类型 R/W, 偏移量 0x800, 复位 0x0000.0000</b>															
OAR															
<b>I2CSCSR, 类型 RO, 偏移量 0x804, 复位 0x0000.0000 ( 只读状态寄存器 )</b>															
OAR2SEL FBR TREQ RREQ															
<b>I2CSCSR, 类型 WO, 偏移量 0x804, 复位 0x0000.0000 ( 只写控制寄存器 )</b>															
DA															
<b>I2CSDR, 类型 R/W, 偏移量 0x808, 复位 0x0000.0000</b>															
DATA															
<b>I2CSIMR, 类型 R/W, 偏移量 0x80C, 复位 0x0000.0000</b>															
STOPIM STARTIM DATAIM															
<b>I2CSRIS, 类型 RO, 偏移量 0x810, 复位 0x0000.0000</b>															
STOPRIS STARTRIS DATARIS															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>I2CSMIS, 类型 RO, 偏移量 0x814, 复位 0x0000.0000</b>																
												STOPMIS	STARTMIS	DATAMIS		
<b>I2CSICR, 类型 WO, 偏移量 0x818, 复位 0x0000.0000</b>																
												STOPIC	STARTIC	DATAIC		
<b>I2CSOAR2, 类型 R/W, 偏移量 0x81C, 复位 0x0000.0000</b>																
								OAR2EN	OAR2							
<b>I2CSACKCTL, 类型 R/W, 偏移量 0x820, 复位 0x0000.0000</b>																
												ACKOVAL	ACKOEN			
<b>内部集成电路 (I<sup>2</sup>C) 接口</b>																
<b>I<sup>2</sup>C 状态和控制</b>																
I2C 0 基址: 0x4002.0000																
I2C 1 基址: 0x4002.1000																
I2C 2 基址: 0x4002.2000																
I2C 3 基址: 0x4002.3000																
I2C 4 基址: 0x400C.0000																
I2C 5 基址: 0x400C.1000																
<b>I2CPP, 类型 RO, 偏移量 0xFC0, 复位 0x0000.0001</b>																
																HS
<b>I2CPC, 类型 RO, 偏移量 0xFC4, 复位 0x0000.0001</b>																
																HS
<b>控制器局域网 (CAN) 模块</b>																
CAN0 基址: 0x4004.0000																
CAN1 基址: 0x4004.1000																
<b>CANCTL, 类型 R/W, 偏移量 0x000, 复位 0x0000.0001 (见第 998 页)</b>																
								TEST	CCE	DAR			EIE	SIE	IE	INIT
<b>CANSTS, 类型 R/W, 偏移量 0x004, 复位 0x0000.0000 (见第 1000 页)</b>																
								BOFF	EWARN	EPASS	RxOK	TxOK	LEC			
<b>CANERR, 类型 RO, 偏移量 0x008, 复位 0x0000.0000 (见第 1003 页)</b>																
RP				REC				TEC								
<b>CANBIT, 类型 R/W, 偏移量 0x00C, 复位 0x0000.2301 (见第 1004 页)</b>																
TSEG2				TSEG1				SJW				BRP				
<b>CANINT, 类型 RO, 偏移量 0x010, 复位 0x0000.0000 (见第 1005 页)</b>																
												INTID				
<b>CANTST, 类型 R/W, 偏移量 0x014, 复位 0x0000.0000 (见第 1006 页)</b>																
								RX	TX		LBACK	SILENT	BASIC			
<b>CANBRPE, 类型 R/W, 偏移量 0x018, 复位 0x0000.0000 (见第 1008 页)</b>																
												BRPE				
<b>CANIF1CRQ, 类型 R/W, 偏移量 0x020, 复位 0x0000.0001 (见第 1009 页)</b>																
BUSY												MNUM				

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
CANIF2CRQ, 类型 R/W, 偏移量 0x080, 复位 0x0000.0001 ( 见第 1009 页 )																	
BUSY				MNUM													
CANIF1CMSK, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000 ( 见第 1010 页 )																	
								WRNRD	MASK	ARB	CONTROL	CLRINTPND	NEWDAT / TXRST	DATAA	DATAB		
CANIF2CMSK, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000 ( 见第 1010 页 )																	
								WRNRD	MASK	ARB	CONTROL	CLRINTPND	NEWDAT / TXRST	DATAA	DATAB		
CANIF1MSK1, 类型 R/W, 偏移量 0x028, 复位 0x0000.FFFF ( 见第 1013 页 )																	
MSK																	
CANIF2MSK1, 类型 R/W, 偏移量 0x088, 复位 0x0000.FFFF ( 见第 1013 页 )																	
MSK																	
CANIF1MSK2, 类型 R/W, 偏移量 0x02C, 复位 0x0000.FFFF ( 见第 1014 页 )																	
MXTD		MDIR		MSK													
CANIF2MSK2, 类型 R/W, 偏移量 0x08C, 复位 0x0000.FFFF ( 见第 1014 页 )																	
MXTD		MDIR		MSK													
CANIF1ARB1, 类型 R/W, 偏移量 0x030, 复位 0x0000.0000 ( 见第 1015 页 )																	
ID																	
CANIF2ARB1, 类型 R/W, 偏移量 0x090, 复位 0x0000.0000 ( 见第 1015 页 )																	
ID																	
CANIF1ARB2, 类型 R/W, 偏移量 0x034, 复位 0x0000.0000 ( 见第 1016 页 )																	
MSGVAL		XTD		DIR		ID											
CANIF2ARB2, 类型 R/W, 偏移量 0x094, 复位 0x0000.0000 ( 见第 1016 页 )																	
MSGVAL		XTD		DIR		ID											
CANIF1MCTL, 类型 R/W, 偏移量 0x038, 复位 0x0000.0000 ( 见第 1018 页 )																	
NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB								DLC	
CANIF2MCTL, 类型 R/W, 偏移量 0x098, 复位 0x0000.0000 ( 见第 1018 页 )																	
NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB								DLC	
CANIF1DA1, 类型 R/W, 偏移量 0x03C, 复位 0x0000.0000 ( 见第 1021 页 )																	
DATA																	
CANIF1DA2, 类型 R/W, 偏移量 0x040, 复位 0x0000.0000 ( 见第 1021 页 )																	
DATA																	
CANIF1DB1, 类型 R/W, 偏移量 0x044, 复位 0x0000.0000 ( 见第 1021 页 )																	
DATA																	
CANIF1DB2, 类型 R/W, 偏移量 0x048, 复位 0x0000.0000 ( 见第 1021 页 )																	
DATA																	



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CANIF2DA1, 类型 R/W, 偏移量 0x09C, 复位 0x0000.0000 (见第 1021 页)															
DATA															
CANIF2DA2, 类型 R/W, 偏移量 0x0A0, 复位 0x0000.0000 (见第 1021 页)															
DATA															
CANIF2DB1, 类型 R/W, 偏移量 0x0A4, 复位 0x0000.0000 (见第 1021 页)															
DATA															
CANIF2DB2, 类型 R/W, 偏移量 0x0A8, 复位 0x0000.0000 (见第 1021 页)															
DATA															
CANTXRQ1, 类型 RO, 偏移量 0x100, 复位 0x0000.0000 (见第 1022 页)															
TXRQST															
CANTXRQ2, 类型 RO, 偏移量 0x104, 复位 0x0000.0000 (见第 1022 页)															
TXRQST															
CANNWDA1, 类型 RO, 偏移量 0x120, 复位 0x0000.0000 (见第 1023 页)															
NEWDAT															
CANNWDA2, 类型 RO, 偏移量 0x124, 复位 0x0000.0000 (见第 1023 页)															
NEWDAT															
CANMSG1INT, 类型 RO, 偏移量 0x140, 复位 0x0000.0000 (见第 1024 页)															
INTPND															
CANMSG2INT, 类型 RO, 偏移量 0x144, 复位 0x0000.0000 (见第 1024 页)															
INTPND															
CANMSG1VAL, 类型 RO, 偏移量 0x160, 复位 0x0000.0000 (见第 1025 页)															
MSGVAL															
CANMSG2VAL, 类型 RO, 偏移量 0x164, 复位 0x0000.0000 (见第 1025 页)															
MSGVAL															
<b>通用串行总线 (USB) 控制器</b> 基址 0x4005.0000															
USBFADDR, 类型 R/W, 偏移量 0x000, 复位 0x00 (见第 1045 页)															
FUNCADDR															
USBPOWER, 类型 R/W, 偏移量 0x001, 复位 0x20 (OTG A/主机模式) (见第 1046 页)															
RESET RESUME SUSPEND PWRDNPHY															
USBPOWER, 类型 R/W, 偏移量 0x001, 复位 0x20 (OTG B/设备模式) (见第 1046 页)															
ISOUP SOFTCONN RESET RESUME SUSPEND PWRDNPHY															
USBTXIS, 类型 RO, 偏移量 0x002, 复位 0x0000 (见第 1049 页)															
EP7 EP6 EP5 EP4 EP3 EP2 EP1 EP0															
USBRXIS, 类型 RO, 偏移量 0x004, 复位 0x0000 (见第 1051 页)															
EP7 EP6 EP5 EP4 EP3 EP2 EP1															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBTXIE, 类型 R/W, 偏移量 0x006, 复位 0xFFFF ( 见第 1052 页 )															
								EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
USBRXIE, 类型 R/W, 偏移量 0x008, 复位 0xFFFF ( 见第 1053 页 )															
								EP7	EP6	EP5	EP4	EP3	EP2	EP1	
USBIS, 类型 RO, 偏移量 0x00A, 复位 0x00 ( OTG A/主机模式 ) ( 见第 1054 页 )															
								VBUSERR	SESREQ	DISCON	CONN	SOF	BABBLE	RESUME	
USBIS, 类型 RO, 偏移量 0x00A, 复位 0x00 ( OTG B/设备模式 ) ( 见第 1054 页 )															
										DISCON		SOF	RESET	RESUME	SUSPEND
USBIE, 类型 R/W, 偏移量 0x00B, 复位 0x06 ( OTG A/主机模式 ) ( 见第 1057 页 )															
								VBUSERR	SESREQ	DISCON	CONN	SOF	BABBLE	RESUME	
USBIE, 类型 R/W, 偏移量 0x00B, 复位 0x06 ( OTG B/设备模式 ) ( 见第 1057 页 )															
										DISCON		SOF	RESET	RESUME	SUSPEND
USBFRAME, 类型 RO, 偏移量 0x00C, 复位 0x0000 ( 见第 1060 页 )															
								FRAME							
USBEPIDX, 类型 R/W, 偏移量 0x00E, 复位 0x00 ( 见第 1061 页 )															
								EPIDX							
USBTEST, 类型 R/W, 偏移量 0x00F, 复位 0x00 ( OTG A/主机模式 ) ( 见第 1062 页 )															
								FORCEH	FIFOACC	FORCEFS					
USBTEST, 类型 R/W, 偏移量 0x00F, 复位 0x00 ( OTG B/设备模式 ) ( 见第 1062 页 )															
									FIFOACC	FORCEFS					
USBFIFO0, 类型 R/W, 偏移量 0x020, 复位 0x0000.0000 ( 见第 1064 页 )															
								EPDATA							
								EPDATA							
USBFIFO1, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000 ( 见第 1064 页 )															
								EPDATA							
								EPDATA							
USBFIFO2, 类型 R/W, 偏移量 0x028, 复位 0x0000.0000 ( 见第 1064 页 )															
								EPDATA							
								EPDATA							
USBFIFO3, 类型 R/W, 偏移量 0x02C, 复位 0x0000.0000 ( 见第 1064 页 )															
								EPDATA							
								EPDATA							
USBFIFO4, 类型 R/W, 偏移量 0x030, 复位 0x0000.0000 ( 见第 1064 页 )															
								EPDATA							
								EPDATA							
USBFIFO5, 类型 R/W, 偏移量 0x034, 复位 0x0000.0000 ( 见第 1064 页 )															
								EPDATA							
								EPDATA							
USBFIFO6, 类型 R/W, 偏移量 0x038, 复位 0x0000.0000 ( 见第 1064 页 )															
								EPDATA							
								EPDATA							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBFIFO7, 类型 R/W, 偏移量 0x03C, 复位 0x0000.0000 (见第 1064 页)															
EPDATA															
EPDATA															
USBDEVCTL, 类型 R/W, 偏移量 0x060, 复位 0x80 (见第 1065 页)															
DEV FSDEV LSDEV VBUS 主机 HOSTREQ SESSION															
USBTXFIFOSZ, 类型 R/W, 偏移量 0x062, 复位 0x00 (见第 1067 页)															
DPB SIZE															
USBRXFIFOSZ, 类型 R/W, 偏移量 0x063, 复位 0x00 (见第 1067 页)															
DPB SIZE															
USBTXFIFOADD, 类型 R/W, 偏移量 0x064, 复位 0x0000 (见第 1068 页)															
ADDR															
USBRXFIFOADD, 类型 R/W, 偏移量 0x066, 复位 0x0000 (见第 1068 页)															
ADDR															
USBCONTIM, 类型 R/W, 偏移量 0x07A, 复位 0x5C (见第 1069 页)															
WTCON WTID															
USBVPLEN, 类型 R/W, 偏移量 0x07B, 复位 0x3C (见第 1070 页)															
VPLEN															
USBFSEOF, 类型 R/W, 偏移量 0x07D, 复位 0x77 (见第 1071 页)															
FSEOFG															
USBLSEOF, 类型 R/W, 偏移量 0x07E, 复位 0x0072 (见第 1072 页)															
LSEOFG															
USBTXFUNCADDR0, 类型 R/W, 偏移量 0x080, 复位 0x00 (见第 1073 页)															
ADDR															
USBTXFUNCADDR1, 类型 R/W, 偏移量 0x088, 复位 0x00 (见第 1073 页)															
ADDR															
USBTXFUNCADDR2, 类型 R/W, 偏移量 0x090, 复位 0x00 (见第 1073 页)															
ADDR															
USBTXFUNCADDR3, 类型 R/W, 偏移量 0x098, 复位 0x00 (见第 1073 页)															
ADDR															
USBTXFUNCADDR4, 类型 R/W, 偏移量 0x0A0, 复位 0x00 (见第 1073 页)															
ADDR															
USBTXFUNCADDR5, 类型 R/W, 偏移量 0x0A8, 复位 0x00 (见第 1073 页)															
ADDR															
USBTXFUNCADDR6, 类型 R/W, 偏移量 0x0B0, 复位 0x00 (见第 1073 页)															
ADDR															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBTXFUNCADDR7, 类型 R/W, 偏移量 0x0B8, 复位 0x00 (见第 1073 页)															
												ADDR			
USBTXHUBADDR0, 类型 R/W, 偏移量 0x082, 复位 0x00 (见第 1074 页)															
												ADDR			
USBTXHUBADDR1, 类型 R/W, 偏移量 0x08A, 复位 0x00 (见第 1074 页)															
												ADDR			
USBTXHUBADDR2, 类型 R/W, 偏移量 0x092, 复位 0x00 (见第 1074 页)															
												ADDR			
USBTXHUBADDR3, 类型 R/W, 偏移量 0x09A, 复位 0x00 (见第 1074 页)															
												ADDR			
USBTXHUBADDR4, 类型 R/W, 偏移量 0x0A2, 复位 0x00 (见第 1074 页)															
												ADDR			
USBTXHUBADDR5, 类型 R/W, 偏移量 0x0AA, 复位 0x00 (见第 1074 页)															
												ADDR			
USBTXHUBADDR6, 类型 R/W, 偏移量 0x0B2, 复位 0x00 (见第 1074 页)															
												ADDR			
USBTXHUBADDR7, 类型 R/W, 偏移量 0x0BA, 复位 0x00 (见第 1074 页)															
												ADDR			
USBTXHUBPORT0, 类型 R/W, 偏移量 0x083, 复位 0x00 (见第 1075 页)															
												端口			
USBTXHUBPORT1, 类型 R/W, 偏移量 0x08B, 复位 0x00 (见第 1075 页)															
												端口			
USBTXHUBPORT2, 类型 R/W, 偏移量 0x093, 复位 0x00 (见第 1075 页)															
												端口			
USBTXHUBPORT3, 类型 R/W, 偏移量 0x09B, 复位 0x00 (见第 1075 页)															
												端口			
USBTXHUBPORT4, 类型 R/W, 偏移量 0x0A3, 复位 0x00 (见第 1075 页)															
												端口			
USBTXHUBPORT5, 类型 R/W, 偏移量 0x0AB, 复位 0x00 (见第 1075 页)															
												端口			
USBTXHUBPORT6, 类型 R/W, 偏移量 0x0B3, 复位 0x00 (见第 1075 页)															
												端口			
USBTXHUBPORT7, 类型 R/W, 偏移量 0x0BB, 复位 0x00 (见第 1075 页)															
												端口			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBRXFUNCADDR1, 类型 R/W, 偏移量 0x08C, 复位 0x00 (见第 1076 页)															
												ADDR			
USBRXFUNCADDR2, 类型 R/W, 偏移量 0x094, 复位 0x00 (见第 1076 页)															
												ADDR			
USBRXFUNCADDR3, 类型 R/W, 偏移量 0x09C, 复位 0x00 (见第 1076 页)															
												ADDR			
USBRXFUNCADDR4, 类型 R/W, 偏移量 0x0A4, 复位 0x00 (见第 1076 页)															
												ADDR			
USBRXFUNCADDR5, 类型 R/W, 偏移量 0x0AC, 复位 0x00 (见第 1076 页)															
												ADDR			
USBRXFUNCADDR6, 类型 R/W, 偏移量 0x0B4, 复位 0x00 (见第 1076 页)															
												ADDR			
USBRXFUNCADDR7, 类型 R/W, 偏移量 0x0BC, 复位 0x00 (见第 1076 页)															
												ADDR			
USBRXHUBADDR1, 类型 R/W, 偏移量 0x08E, 复位 0x00 (见第 1077 页)															
												ADDR			
USBRXHUBADDR2, 类型 R/W, 偏移量 0x096, 复位 0x00 (见第 1077 页)															
												ADDR			
USBRXHUBADDR3, 类型 R/W, 偏移量 0x09E, 复位 0x00 (见第 1077 页)															
												ADDR			
USBRXHUBADDR4, 类型 R/W, 偏移量 0x0A6, 复位 0x00 (见第 1077 页)															
												ADDR			
USBRXHUBADDR5, 类型 R/W, 偏移量 0x0AE, 复位 0x00 (见第 1077 页)															
												ADDR			
USBRXHUBADDR6, 类型 R/W, 偏移量 0x0B6, 复位 0x00 (见第 1077 页)															
												ADDR			
USBRXHUBADDR7, 类型 R/W, 偏移量 0x0BE, 复位 0x00 (见第 1077 页)															
												ADDR			
USBRXHUBPORT1, 类型 R/W, 偏移量 0x08F, 复位 0x00 (见第 1078 页)															
												端口			
USBRXHUBPORT2, 类型 R/W, 偏移量 0x097, 复位 0x00 (见第 1078 页)															
												端口			
USBRXHUBPORT3, 类型 R/W, 偏移量 0x09F, 复位 0x00 (见第 1078 页)															
												端口			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBRXHUBPORT4, 类型 R/W, 偏移量 0x0A7, 复位 0x00 (见第 1078 页)															
												端口			
USBRXHUBPORT5, 类型 R/W, 偏移量 0x0AF, 复位 0x00 (见第 1078 页)															
												端口			
USBRXHUBPORT6, 类型 R/W, 偏移量 0x0B7, 复位 0x00 (见第 1078 页)															
												端口			
USBRXHUBPORT7, 类型 R/W, 偏移量 0x0BF, 复位 0x00 (见第 1078 页)															
												端口			
USBTXMAXP1, 类型 R/W, 偏移量 0x110, 复位 0x0000 (见第 1079 页)															
												MAXLOAD			
USBTXMAXP2, 类型 R/W, 偏移量 0x120, 复位 0x0000 (见第 1079 页)															
												MAXLOAD			
USBTXMAXP3, 类型 R/W, 偏移量 0x130, 复位 0x0000 (见第 1079 页)															
												MAXLOAD			
USBTXMAXP4, 类型 R/W, 偏移量 0x140, 复位 0x0000 (见第 1079 页)															
												MAXLOAD			
USBTXMAXP5, 类型 R/W, 偏移量 0x150, 复位 0x0000 (见第 1079 页)															
												MAXLOAD			
USBTXMAXP6, 类型 R/W, 偏移量 0x160, 复位 0x0000 (见第 1079 页)															
												MAXLOAD			
USBTXMAXP7, 类型 R/W, 偏移量 0x170, 复位 0x0000 (见第 1079 页)															
												MAXLOAD			
USBCSRL0, 类型 W1C, 偏移量 0x102, 复位 0x00 (OTG A/主机模式) (见第 1080 页)															
								NAKTO	STATUS	REQPKT	ERROR	SETUP	STALLED	TXRDY	RXRDY
USBCSRL0, 类型 W1C, 偏移量 0x102, 复位 0x00 (OTG B/设备模式) (见第 1080 页)															
								SETENDC	RXRDYC	STALL	SETEND	DATAEND	STALLED	TXRDY	RXRDY
USBCSRH0, 类型 W1C, 偏移量 0x103, 复位 0x00 (OTG A/主机模式) (见第 1084 页)															
												DTWE	DT	FLUSH	
USBCSRH0, 类型 W1C, 偏移量 0x103, 复位 0x00 (OTG B/设备模式) (见第 1084 页)															
												FLUSH			
USBCOUNT0, 类型 RO, 偏移量 0x108, 复位 0x00 (见第 1086 页)															
												COUNT			
USBTYPE0, 类型 R/W, 偏移量 0x10A, 复位 0x00 (见第 1087 页)															
												SPEED			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBNAKLMT, 类型 R/W, 偏移量 0x10B, 复位 0x00 (见第 1088 页)															
NAKLMT															
USBTXCSRL1, 类型 R/W, 偏移量 0x112, 复位 0x00 (OTG A/主机模式) (见第 1089 页)															
NAKTO CLRDT STALLED SETUP FLUSH ERROR FIFONE TXRDY															
USBTXCSRL2, 类型 R/W, 偏移量 0x122, 复位 0x00 (OTG A/主机模式) (见第 1089 页)															
NAKTO CLRDT STALLED SETUP FLUSH ERROR FIFONE TXRDY															
USBTXCSRL3, 类型 R/W, 偏移量 0x132, 复位 0x00 (OTG A/主机模式) (见第 1089 页)															
NAKTO CLRDT STALLED SETUP FLUSH ERROR FIFONE TXRDY															
USBTXCSRL4, 类型 R/W, 偏移量 0x142, 复位 0x00 (OTG A/主机模式) (见第 1089 页)															
NAKTO CLRDT STALLED SETUP FLUSH ERROR FIFONE TXRDY															
USBTXCSRL5, 类型 R/W, 偏移量 0x152, 复位 0x00 (OTG A/主机模式) (见第 1089 页)															
NAKTO CLRDT STALLED SETUP FLUSH ERROR FIFONE TXRDY															
USBTXCSRL6, 类型 R/W, 偏移量 0x162, 复位 0x00 (OTG A/主机模式) (见第 1089 页)															
NAKTO CLRDT STALLED SETUP FLUSH ERROR FIFONE TXRDY															
USBTXCSRL7, 类型 R/W, 偏移量 0x172, 复位 0x00 (OTG A/主机模式) (见第 1089 页)															
NAKTO CLRDT STALLED SETUP FLUSH ERROR FIFONE TXRDY															
USBTXCSR1, 类型 R/W, 偏移量 0x112, 复位 0x00 (OTG B/设备模式) (见第 1089 页)															
CLRDT STALLED STALL FLUSH UNDRN FIFONE TXRDY															
USBTXCSR2, 类型 R/W, 偏移量 0x122, 复位 0x00 (OTG B/设备模式) (见第 1089 页)															
CLRDT STALLED STALL FLUSH UNDRN FIFONE TXRDY															
USBTXCSR3, 类型 R/W, 偏移量 0x132, 复位 0x00 (OTG B/设备模式) (见第 1089 页)															
CLRDT STALLED STALL FLUSH UNDRN FIFONE TXRDY															
USBTXCSR4, 类型 R/W, 偏移量 0x142, 复位 0x00 (OTG B/设备模式) (见第 1089 页)															
CLRDT STALLED STALL FLUSH UNDRN FIFONE TXRDY															
USBTXCSR5, 类型 R/W, 偏移量 0x152, 复位 0x00 (OTG B/设备模式) (见第 1089 页)															
CLRDT STALLED STALL FLUSH UNDRN FIFONE TXRDY															
USBTXCSR6, 类型 R/W, 偏移量 0x162, 复位 0x00 (OTG B/设备模式) (见第 1089 页)															
CLRDT STALLED STALL FLUSH UNDRN FIFONE TXRDY															
USBTXCSR7, 类型 R/W, 偏移量 0x172, 复位 0x00 (OTG B/设备模式) (见第 1089 页)															
CLRDT STALLED STALL FLUSH UNDRN FIFONE TXRDY															
USBTXCSRH1, 类型 R/W, 偏移量 0x113, 复位 0x00 (OTG A/主机模式) (见第 1093 页)															
AUTOSSET MODE DMAEN FDT DMAMOD DTWE DT															
USBTXCSRH2, 类型 R/W, 偏移量 0x123, 复位 0x00 (OTG A/主机模式) (见第 1093 页)															
AUTOSSET MODE DMAEN FDT DMAMOD DTWE DT															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBTXCSRH3, 类型 R/W, 偏移量 0x133, 复位 0x00 (OTG A/主机模式) (见第 1093 页)															
								AUTOSET		MODE	DMAEN	FDT	DMAMOD	DTWE	DT
USBTXCSRH4, 类型 R/W, 偏移量 0x143, 复位 0x00 (OTG A/主机模式) (见第 1093 页)															
								AUTOSET		MODE	DMAEN	FDT	DMAMOD	DTWE	DT
USBTXCSRH5, 类型 R/W, 偏移量 0x153, 复位 0x00 (OTG A/主机模式) (见第 1093 页)															
								AUTOSET		MODE	DMAEN	FDT	DMAMOD	DTWE	DT
USBTXCSRH6, 类型 R/W, 偏移量 0x163, 复位 0x00 (OTG A/主机模式) (见第 1093 页)															
								AUTOSET		MODE	DMAEN	FDT	DMAMOD	DTWE	DT
USBTXCSRH7, 类型 R/W, 偏移量 0x173, 复位 0x00 (OTG A/主机模式) (见第 1093 页)															
								AUTOSET		MODE	DMAEN	FDT	DMAMOD	DTWE	DT
USBTXCSRH1, 类型 R/W, 偏移量 0x113, 复位 0x00 (OTG B/设备模式) (见第 1093 页)															
								AUTOSET	ISO	MODE	DMAEN	FDT	DMAMOD		
USBTXCSRH2, 类型 R/W, 偏移量 0x123, 复位 0x00 (OTG B/设备模式) (见第 1093 页)															
								AUTOSET	ISO	MODE	DMAEN	FDT	DMAMOD		
USBTXCSRH3, 类型 R/W, 偏移量 0x133, 复位 0x00 (OTG B/设备模式) (见第 1093 页)															
								AUTOSET	ISO	MODE	DMAEN	FDT	DMAMOD		
USBTXCSRH4, 类型 R/W, 偏移量 0x143, 复位 0x00 (OTG B/设备模式) (见第 1093 页)															
								AUTOSET	ISO	MODE	DMAEN	FDT	DMAMOD		
USBTXCSRH5, 类型 R/W, 偏移量 0x153, 复位 0x00 (OTG B/设备模式) (见第 1093 页)															
								AUTOSET	ISO	MODE	DMAEN	FDT	DMAMOD		
USBTXCSRH6, 类型 R/W, 偏移量 0x163, 复位 0x00 (OTG B/设备模式) (见第 1093 页)															
								AUTOSET	ISO	MODE	DMAEN	FDT	DMAMOD		
USBTXCSRH7, 类型 R/W, 偏移量 0x173, 复位 0x00 (OTG B/设备模式) (见第 1093 页)															
								AUTOSET	ISO	MODE	DMAEN	FDT	DMAMOD		
USBRXMAXP1, 类型 R/W, 偏移量 0x114, 复位 0x0000 (见第 1097 页)															
USBRXMAXP2, 类型 R/W, 偏移量 0x124, 复位 0x0000 (见第 1097 页)															
USBRXMAXP3, 类型 R/W, 偏移量 0x134, 复位 0x0000 (见第 1097 页)															
USBRXMAXP4, 类型 R/W, 偏移量 0x144, 复位 0x0000 (见第 1097 页)															
USBRXMAXP5, 类型 R/W, 偏移量 0x154, 复位 0x0000 (见第 1097 页)															



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBXMAXP6, 类型 R/W, 偏移量 0x164, 复位 0x0000 (见第 1097 页)															
MAXLOAD															
USBXMAXP7, 类型 R/W, 偏移量 0x174, 复位 0x0000 (见第 1097 页)															
MAXLOAD															
USBXCSRL1, 类型 R/W, 偏移量 0x116, 复位 0x00 (OTG A/主机模式) (见第 1098 页)															
CLRDT STALLED REQPKT FLUSH DATAERR / NAKTO ERROR FULL RXRDY															
USBXCSRL2, 类型 R/W, 偏移量 0x126, 复位 0x00 (OTG A/主机模式) (见第 1098 页)															
CLRDT STALLED REQPKT FLUSH DATAERR / NAKTO ERROR FULL RXRDY															
USBXCSRL3, 类型 R/W, 偏移量 0x136, 复位 0x00 (OTG A/主机模式) (见第 1098 页)															
CLRDT STALLED REQPKT FLUSH DATAERR / NAKTO ERROR FULL RXRDY															
USBXCSRL4, 类型 R/W, 偏移量 0x146, 复位 0x00 (OTG A/主机模式) (见第 1098 页)															
CLRDT STALLED REQPKT FLUSH DATAERR / NAKTO ERROR FULL RXRDY															
USBXCSRL5, 类型 R/W, 偏移量 0x156, 复位 0x00 (OTG A/主机模式) (见第 1098 页)															
CLRDT STALLED REQPKT FLUSH DATAERR / NAKTO ERROR FULL RXRDY															
USBXCSRL6, 类型 R/W, 偏移量 0x166, 复位 0x00 (OTG A/主机模式) (见第 1098 页)															
CLRDT STALLED REQPKT FLUSH DATAERR / NAKTO ERROR FULL RXRDY															
USBXCSRL7, 类型 R/W, 偏移量 0x176, 复位 0x00 (OTG A/主机模式) (见第 1098 页)															
CLRDT STALLED REQPKT FLUSH DATAERR / NAKTO ERROR FULL RXRDY															
USBXCSRL1, 类型 R/W, 偏移量 0x116, 复位 0x00 (OTG B/设备模式) (见第 1098 页)															
CLRDT STALLED STALL FLUSH DATAERR OVER FULL RXRDY															
USBXCSRL2, 类型 R/W, 偏移量 0x126, 复位 0x00 (OTG B/设备模式) (见第 1098 页)															
CLRDT STALLED STALL FLUSH DATAERR OVER FULL RXRDY															
USBXCSRL3, 类型 R/W, 偏移量 0x136, 复位 0x00 (OTG B/设备模式) (见第 1098 页)															
CLRDT STALLED STALL FLUSH DATAERR OVER FULL RXRDY															
USBXCSRL4, 类型 R/W, 偏移量 0x146, 复位 0x00 (OTG B/设备模式) (见第 1098 页)															
CLRDT STALLED STALL FLUSH DATAERR OVER FULL RXRDY															
USBXCSRL5, 类型 R/W, 偏移量 0x156, 复位 0x00 (OTG B/设备模式) (见第 1098 页)															
CLRDT STALLED STALL FLUSH DATAERR OVER FULL RXRDY															
USBXCSRL6, 类型 R/W, 偏移量 0x166, 复位 0x00 (OTG B/设备模式) (见第 1098 页)															
CLRDT STALLED STALL FLUSH DATAERR OVER FULL RXRDY															
USBXCSRL7, 类型 R/W, 偏移量 0x176, 复位 0x00 (OTG B/设备模式) (见第 1098 页)															
CLRDT STALLED STALL FLUSH DATAERR OVER FULL RXRDY															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBXCSRH1, 类型 R/W, 偏移量 0x117, 复位 0x00 (OTG A/主机模式) (见第 1102 页)															
								AUTOCL	AUTORQ	DMAEN	PIDERR	DMAMOD	DTWE	DT	
USBXCSRH2, 类型 R/W, 偏移量 0x127, 复位 0x00 (OTG A/主机模式) (见第 1102 页)															
								AUTOCL	AUTORQ	DMAEN	PIDERR	DMAMOD	DTWE	DT	
USBXCSRH3, 类型 R/W, 偏移量 0x137, 复位 0x00 (OTG A/主机模式) (见第 1102 页)															
								AUTOCL	AUTORQ	DMAEN	PIDERR	DMAMOD	DTWE	DT	
USBXCSRH4, 类型 R/W, 偏移量 0x147, 复位 0x00 (OTG A/主机模式) (见第 1102 页)															
								AUTOCL	AUTORQ	DMAEN	PIDERR	DMAMOD	DTWE	DT	
USBXCSRH5, 类型 R/W, 偏移量 0x157, 复位 0x00 (OTG A/主机模式) (见第 1102 页)															
								AUTOCL	AUTORQ	DMAEN	PIDERR	DMAMOD	DTWE	DT	
USBXCSRH6, 类型 R/W, 偏移量 0x167, 复位 0x00 (OTG A/主机模式) (见第 1102 页)															
								AUTOCL	AUTORQ	DMAEN	PIDERR	DMAMOD	DTWE	DT	
USBXCSRH7, 类型 R/W, 偏移量 0x177, 复位 0x00 (OTG A/主机模式) (见第 1102 页)															
								AUTOCL	AUTORQ	DMAEN	PIDERR	DMAMOD	DTWE	DT	
USBXCSRH1, 类型 R/W, 偏移量 0x117, 复位 0x00 (OTG B/设备模式) (见第 1102 页)															
								AUTOCL	ISO	DMAEN	DISNYET / PIDERR	DMAMOD			
USBXCSRH2, 类型 R/W, 偏移量 0x127, 复位 0x00 (OTG B/设备模式) (见第 1102 页)															
								AUTOCL	ISO	DMAEN	DISNYET / PIDERR	DMAMOD			
USBXCSRH3, 类型 R/W, 偏移量 0x137, 复位 0x00 (OTG B/设备模式) (见第 1102 页)															
								AUTOCL	ISO	DMAEN	DISNYET / PIDERR	DMAMOD			
USBXCSRH4, 类型 R/W, 偏移量 0x147, 复位 0x00 (OTG B/设备模式) (见第 1102 页)															
								AUTOCL	ISO	DMAEN	DISNYET / PIDERR	DMAMOD			
USBXCSRH5, 类型 R/W, 偏移量 0x157, 复位 0x00 (OTG B/设备模式) (见第 1102 页)															
								AUTOCL	ISO	DMAEN	DISNYET / PIDERR	DMAMOD			
USBXCSRH6, 类型 R/W, 偏移量 0x167, 复位 0x00 (OTG B/设备模式) (见第 1102 页)															
								AUTOCL	ISO	DMAEN	DISNYET / PIDERR	DMAMOD			
USBXCSRH7, 类型 R/W, 偏移量 0x177, 复位 0x00 (OTG B/设备模式) (见第 1102 页)															
								AUTOCL	ISO	DMAEN	DISNYET / PIDERR	DMAMOD			
USBXCOUNT1, 类型 RO, 偏移量 0x118, 复位 0x0000 (见第 1106 页)															
COUNT															
USBXCOUNT2, 类型 RO, 偏移量 0x128, 复位 0x0000 (见第 1106 页)															
COUNT															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBXCOUNT3, 类型 RO, 偏移量 0x138, 复位 0x0000 (见第 1106 页)															
COUNT															
USBXCOUNT4, 类型 RO, 偏移量 0x148, 复位 0x0000 (见第 1106 页)															
COUNT															
USBXCOUNT5, 类型 RO, 偏移量 0x158, 复位 0x0000 (见第 1106 页)															
COUNT															
USBXCOUNT6, 类型 RO, 偏移量 0x168, 复位 0x0000 (见第 1106 页)															
COUNT															
USBXCOUNT7, 类型 RO, 偏移量 0x178, 复位 0x0000 (见第 1106 页)															
COUNT															
USBTXTYPE1, 类型 R/W, 偏移量 0x11A, 复位 0x00 (见第 1107 页)															
SPEED      PROTO      TEP															
USBTXTYPE2, 类型 R/W, 偏移量 0x12A, 复位 0x00 (见第 1107 页)															
SPEED      PROTO      TEP															
USBTXTYPE3, 类型 R/W, 偏移量 0x13A, 复位 0x00 (见第 1107 页)															
SPEED      PROTO      TEP															
USBTXTYPE4, 类型 R/W, 偏移量 0x14A, 复位 0x00 (见第 1107 页)															
SPEED      PROTO      TEP															
USBTXTYPE5, 类型 R/W, 偏移量 0x15A, 复位 0x00 (见第 1107 页)															
SPEED      PROTO      TEP															
USBTXTYPE6, 类型 R/W, 偏移量 0x16A, 复位 0x00 (见第 1107 页)															
SPEED      PROTO      TEP															
USBTXTYPE7, 类型 R/W, 偏移量 0x17A, 复位 0x00 (见第 1107 页)															
SPEED      PROTO      TEP															
USBTXINTERVAL1, 类型 R/W, 偏移量 0x11B, 复位 0x00 (见第 1109 页)															
TXPOLL / NAKLMT															
USBTXINTERVAL2, 类型 R/W, 偏移量 0x12B, 复位 0x00 (见第 1109 页)															
TXPOLL / NAKLMT															
USBTXINTERVAL3, 类型 R/W, 偏移量 0x13B, 复位 0x00 (见第 1109 页)															
TXPOLL / NAKLMT															
USBTXINTERVAL4, 类型 R/W, 偏移量 0x14B, 复位 0x00 (见第 1109 页)															
TXPOLL / NAKLMT															
USBTXINTERVAL5, 类型 R/W, 偏移量 0x15B, 复位 0x00 (见第 1109 页)															
TXPOLL / NAKLMT															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBTXINTERVAL6, 类型 R/W, 偏移量 0x16B, 复位 0x00 (见第 1109 页)															
TXPOLL / NAKLMT															
USBTXINTERVAL7, 类型 R/W, 偏移量 0x17B, 复位 0x00 (见第 1109 页)															
TXPOLL / NAKLMT															
USBRXTYPE1, 类型 R/W, 偏移量 0x11C, 复位 0x00 (见第 1110 页)															
SPEED                      PROTO                      TEP															
USBRXTYPE2, 类型 R/W, 偏移量 0x12C, 复位 0x00 (见第 1110 页)															
SPEED                      PROTO                      TEP															
USBRXTYPE3, 类型 R/W, 偏移量 0x13C, 复位 0x00 (见第 1110 页)															
SPEED                      PROTO                      TEP															
USBRXTYPE4, 类型 R/W, 偏移量 0x14C, 复位 0x00 (见第 1110 页)															
SPEED                      PROTO                      TEP															
USBRXTYPE5, 类型 R/W, 偏移量 0x15C, 复位 0x00 (见第 1110 页)															
SPEED                      PROTO                      TEP															
USBRXTYPE6, 类型 R/W, 偏移量 0x16C, 复位 0x00 (见第 1110 页)															
SPEED                      PROTO                      TEP															
USBRXTYPE7, 类型 R/W, 偏移量 0x17C, 复位 0x00 (见第 1110 页)															
SPEED                      PROTO                      TEP															
USBRXINTERVAL1, 类型 R/W, 偏移量 0x11D, 复位 0x00 (见第 1112 页)															
TXPOLL / NAKLMT															
USBRXINTERVAL2, 类型 R/W, 偏移量 0x12D, 复位 0x00 (见第 1112 页)															
TXPOLL / NAKLMT															
USBRXINTERVAL3, 类型 R/W, 偏移量 0x13D, 复位 0x00 (见第 1112 页)															
TXPOLL / NAKLMT															
USBRXINTERVAL4, 类型 R/W, 偏移量 0x14D, 复位 0x00 (见第 1112 页)															
TXPOLL / NAKLMT															
USBRXINTERVAL5, 类型 R/W, 偏移量 0x15D, 复位 0x00 (见第 1112 页)															
TXPOLL / NAKLMT															
USBRXINTERVAL6, 类型 R/W, 偏移量 0x16D, 复位 0x00 (见第 1112 页)															
TXPOLL / NAKLMT															
USBRXINTERVAL7, 类型 R/W, 偏移量 0x17D, 复位 0x00 (见第 1112 页)															
TXPOLL / NAKLMT															
USBRQPKTCOUNT1, 类型 R/W, 偏移量 0x304, 复位 0x0000 (见第 1113 页)															
COUNT															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
USBQPKTCOUNT2, 类型 R/W, 偏移量 0x308, 复位 0x0000 (见第 1113 页)																
COUNT																
USBQPKTCOUNT3, 类型 R/W, 偏移量 0x30C, 复位 0x0000 (见第 1113 页)																
COUNT																
USBQPKTCOUNT4, 类型 R/W, 偏移量 0x310, 复位 0x0000 (见第 1113 页)																
COUNT																
USBQPKTCOUNT5, 类型 R/W, 偏移量 0x314, 复位 0x0000 (见第 1113 页)																
COUNT																
USBQPKTCOUNT6, 类型 R/W, 偏移量 0x318, 复位 0x0000 (见第 1113 页)																
COUNT																
USBQPKTCOUNT7, 类型 R/W, 偏移量 0x31C, 复位 0x0000 (见第 1113 页)																
COUNT																
USBRDPKTBUFDIS, 类型 R/W, 偏移量 0x340, 复位 0x0000 (见第 1114 页)																
								EP7	EP6	EP5	EP4	EP3	EP2	EP1		
USBTDPKTBUFDIS, 类型 R/W, 偏移量 0x342, 复位 0x0000 (见第 1115 页)																
								EP7	EP6	EP5	EP4	EP3	EP2	EP1		
USBEP, 类型 R/W, 偏移量 0x400, 复位 0x0000.0000 (见第 1116 页)																
				PFLTACT				PFLTAEN PFLTSEN PFLTEN				EPENDE		EPEN		
USBEP, 类型 RO, 偏移量 0x404, 复位 0x0000.0000 (见第 1119 页)																
PF																
USBEP, 类型 R/W, 偏移量 0x408, 复位 0x0000.0000 (见第 1120 页)																
PF																
USBEP, 类型 R/W, 偏移量 0x40C, 复位 0x0000.0000 (见第 1121 页)																
PF																
USBRDRIS, 类型 RO, 偏移量 0x410, 复位 0x0000.0000 (见第 1122 页)																
RESUME																
USBRDRIM, 类型 R/W, 偏移量 0x414, 复位 0x0000.0000 (见第 1123 页)																
RESUME																
USBRDRISC, 类型 W1C, 偏移量 0x418, 复位 0x0000.0000 (见第 1124 页)																
RESUME																
USBGPCS, 类型 R/W, 偏移量 0x41C, 复位 0x0000.0003 (见第 1125 页)																
														DEVMOD0TG	DEVMOD	
USBVDC, 类型 R/W, 偏移量 0x430, 复位 0x0000.0000 (见第 1126 页)																
VB DEN																

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBVDCRIS, 类型 RO, 偏移量 0x434, 复位 0x0000.0000 (见第 1127 页)															
															VD
USBVDCIM, 类型 R/W, 偏移量 0x438, 复位 0x0000.0000 (见第 1128 页)															
															VD
USBVDCISC, 类型 R/W, 偏移量 0x43C, 复位 0x0000.0000 (见第 1129 页)															
															VD
USBIDVRIS, 类型 RO, 偏移量 0x444, 复位 0x0000.0000 (见第 1130 页)															
															ID
USBIDVIM, 类型 R/W, 偏移量 0x448, 复位 0x0000.0000 (见第 1131 页)															
															ID
USBIDVISC, 类型 R/W1C, 偏移量 0x44C, 复位 0x0000.0000 (见第 1132 页)															
															ID
USBDMASEL, 类型 R/W, 偏移量 0x450, 复位 0x0033.2211 (见第 1133 页)															
USBPP, 类型 RO, 偏移量 0xFC0, 复位 0x0000.10D0 (见第 1135 页)															
模拟比较器 基址 0x4003.C000															
ACMIS, 类型 R/W1C, 偏移量 0x00, 复位 0x0000.0000 (见第 1142 页)															
ACRIS, 类型 RO, 偏移量 0x04, 复位 0x0000.0000 (见第 1143 页)															
ACINTEN, 类型 R/W, 偏移量 0x08, 复位 0x0000.0000 (见第 1144 页)															
ACREFCTL, 类型 R/W, 偏移量 0x010, 复位 0x0000.0000 (见第 1145 页)															
ACSTAT0, 类型 RO, 偏移量 0x020, 复位 0x0000.0000 (见第 1146 页)															
ACSTAT1, 类型 RO, 偏移量 0x040, 复位 0x0000.0000 (见第 1146 页)															
ACSTAT2, 类型 RO, 偏移量 0x060, 复位 0x0000.0000 (见第 1146 页)															
ACCTL0, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000 (见第 1147 页)															
ACCTL1, 类型 R/W, 偏移量 0x044, 复位 0x0000.0000 (见第 1147 页)															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																												
ACCTL2, 类型 R/W, 偏移量 0x064, 复位 0x0000.0000 (见第 1147 页)																																											
				TOEN		ASRCP		TSLVAL		TSEN		ISLVAL		ISEN		CINV																											
ACMPPP, 类型 RO, 偏移量 0xFC0, 复位 0x0007.0007 (见第 1149 页)																																											
												C20		C10		C00																											
												CMP2		CMP1		CMP0																											
脉宽调制器 (PWM)																																											
PWM0x4002.8000 (0x4002.8FFF 的结束地址) 基址: 0x4002.8000																																											
PWM10x4002.9000 (0x4002.9FFF 的结束地址) 基址: 0x4002.9000																																											
PWMCTL, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000 (见第 1164 页)																																											
												GLOBALSYNC3		GLOBALSYNC2		GLOBALSYNC1		GLOBALSYNC0																									
PWMSYNC, 类型 R/W, 偏移量 0x004, 复位 0x0000.0000 (见第 1166 页)																																											
												SYNC3		Sync2		Sync1		SYNC0																									
PWMENABLE, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000 (见第 1167 页)																																											
								PWM7EN		PWM6EN		PWM5EN		PWM4EN		PWM3EN		PWM2EN		PWM1EN		PWM0EN																					
PWMINVERT, 类型 R/W, 偏移量 0x00C, 复位 0x0000.0000 (见第 1169 页)																																											
								PWM7INV		PWM6INV		PWM5INV		PWM4INV		PWM3INV		PWM2INV		PWM1INV		PWM0INV																					
PWMFAULT, 类型 R/W, 偏移量 0x010, 复位 0x0000.0000 (见第 1171 页)																																											
								FAULT7		FAULT6		FAULT5		FAULT4		FAULT3		FAULT2		FAULT1		FAULT0																					
PWMINTEN, 类型 R/W, 偏移量 0x014, 复位 0x0000.0000 (见第 1173 页)																																											
												INTFAULT3		INTFAULT2		INTFAULT1		INTFAULT0																									
												INTPWM3		INTPWM2		INTPWM1		INTPWM0																									
PWMRIS, 类型 RO, 偏移量 0x018, 复位 0x0000.0000 (见第 1175 页)																																											
												INTFAULT3		INTFAULT2		INTFAULT1		INTFAULT0																									
												INTPWM3		INTPWM2		INTPWM1		INTPWM0																									
PWMISC, 类型 R/W1C, 偏移量 0x01C, 复位 0x0000.0000 (见第 1177 页)																																											
												INTFAULT3		INTFAULT2		INTFAULT1		INTFAULT0																									
												INTPWM3		INTPWM2		INTPWM1		INTPWM0																									
PWMSTATUS, 类型 RO, 偏移量 0x020, 复位 0x0000.0000 (见第 1179 页)																																											
												FAULT3		FAULT2		FAULT1		FAULT0																									
PWMFAULTVAL, 类型 R/W, 偏移量 0x024, 复位 0x0000.0000 (见第 1181 页)																																											
								PWM7		PWM6		PWM5		PWM4		PWM3		PWM2		PWM1		PWM0																					
PWMENUPD, 类型 R/W, 偏移量 0x028, 复位 0x0000.0000 (见第 1183 页)																																											
ENUPD7				ENUPD6				ENUPD5				ENUPD4				ENUPD3				ENUPD2				ENUPD1				ENUPD0															
PWM0CTL, 类型 R/W, 偏移量 0x040, 复位 0x0000.0000 (见第 1186 页)																																											
																				LATCH				MINFLTPER				FLTSRC															
DBFALLUPD				DBRISEUPD				DBCTLUPD				GENBUPD				GENAUPD				CMPBUPD				CMPAUPD				LOADUPD				DEBUG				MODE				ENABLE			
PWM1CTL, 类型 R/W, 偏移量 0x080, 复位 0x0000.0000 (见第 1186 页)																																											
																								LATCH				MINFLTPER				FLTSRC											
DBFALLUPD				DBRISEUPD				DBCTLUPD				GENBUPD				GENAUPD				CMPBUPD				CMPAUPD				LOADUPD				DEBUG				MODE				ENABLE			
PWM2CTL, 类型 R/W, 偏移量 0x0C0, 复位 0x0000.0000 (见第 1186 页)																																											
																								LATCH				MINFLTPER				FLTSRC											
DBFALLUPD				DBRISEUPD				DBCTLUPD				GENBUPD				GENAUPD				CMPBUPD				CMPAUPD				LOADUPD				DEBUG				MODE				ENABLE			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
PWM3CTL, 类型 R/W, 偏移量 0x100, 复位 0x0000.0000 (见第 1186 页)																					
												LATCH	MINFLTPER	FLTSRC							
DBFALLUPD				DBRISEUPD				DBCTLUPD				GENBUPD		GENAUPD		CMPBUPD	CMPAUPD	LOADUPD	DEBUG	MODE	ENABLE
PWM0INTEN, 类型 R/W, 偏移量 0x044, 复位 0x0000.0000 (见第 1190 页)																					
				TRCMPBD	TRCMPBU	TRCMPAD	TRCMPAU	TRCNTLOAD	TRCNTZERO					INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO		
PWM1INTEN, 类型 R/W, 偏移量 0x084, 复位 0x0000.0000 (见第 1190 页)																					
				TRCMPBD	TRCMPBU	TRCMPAD	TRCMPAU	TRCNTLOAD	TRCNTZERO					INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO		
PWM2INTEN, 类型 R/W, 偏移量 0x0C4, 复位 0x0000.0000 (见第 1190 页)																					
				TRCMPBD	TRCMPBU	TRCMPAD	TRCMPAU	TRCNTLOAD	TRCNTZERO					INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO		
PWM3INTEN, 类型 R/W, 偏移量 0x104, 复位 0x0000.0000 (见第 1190 页)																					
				TRCMPBD	TRCMPBU	TRCMPAD	TRCMPAU	TRCNTLOAD	TRCNTZERO					INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO		
PWM0RIS, 类型 RO, 偏移量 0x048, 复位 0x0000.0000 (见第 1193 页)																					
												INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO				
PWM1RIS, 类型 RO, 偏移量 0x088, 复位 0x0000.0000 (见第 1193 页)																					
												INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO				
PWM2RIS, 类型 RO, 偏移量 0x0C8, 复位 0x0000.0000 (见第 1193 页)																					
												INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO				
PWM3RIS, 类型 RO, 偏移量 0x108, 复位 0x0000.0000 (见第 1193 页)																					
												INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO				
PWM0ISC, 类型 R/W1C, 偏移量 0x04C, 复位 0x0000.0000 (见第 1195 页)																					
												INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO				
PWM1ISC, 类型 R/W1C, 偏移量 0x08C, 复位 0x0000.0000 (见第 1195 页)																					
												INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO				
PWM2ISC, 类型 R/W1C, 偏移量 0x0CC, 复位 0x0000.0000 (见第 1195 页)																					
												INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO				
PWM3ISC, 类型 R/W1C, 偏移量 0x10C, 复位 0x0000.0000 (见第 1195 页)																					
												INTCMPBD	INTCMPBU	INTCMPAD	INTCMPAU	INTCNTLOAD	INTCNTZERO				
PWM0LOAD, 类型 R/W, 偏移量 0x050, 复位 0x0000.0000 (见第 1197 页)																					
LOAD																					
PWM1LOAD, 类型 R/W, 偏移量 0x090, 复位 0x0000.0000 (见第 1197 页)																					
LOAD																					
PWM2LOAD, 类型 R/W, 偏移量 0x0D0, 复位 0x0000.0000 (见第 1197 页)																					
LOAD																					
PWM3LOAD, 类型 R/W, 偏移量 0x110, 复位 0x0000.0000 (见第 1197 页)																					
LOAD																					



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
PWM0COUNT, 类型 RO, 偏移量 0x054, 复位 0x0000.0000 (见第 1198 页)																											
COUNT																											
PWM1COUNT, 类型 RO, 偏移量 0x094, 复位 0x0000.0000 (见第 1198 页)																											
COUNT																											
PWM2COUNT, 类型 RO, 偏移量 0x0D4, 复位 0x0000.0000 (见第 1198 页)																											
COUNT																											
PWM3COUNT, 类型 RO, 偏移量 0x114, 复位 0x0000.0000 (见第 1198 页)																											
COUNT																											
PWM0CMPA, 类型 R/W, 偏移量 0x058, 复位 0x0000.0000 (见第 1199 页)																											
COMPA																											
PWM1CMPA, 类型 R/W, 偏移量 0x098, 复位 0x0000.0000 (见第 1199 页)																											
COMPA																											
PWM2CMPA, 类型 R/W, 偏移量 0x0D8, 复位 0x0000.0000 (见第 1199 页)																											
COMPA																											
PWM3CMPA, 类型 R/W, 偏移量 0x118, 复位 0x0000.0000 (见第 1199 页)																											
COMPA																											
PWM0CMPB, 类型 R/W, 偏移量 0x05C, 复位 0x0000.0000 (见第 1200 页)																											
COMPB																											
PWM1CMPB, 类型 R/W, 偏移量 0x09C, 复位 0x0000.0000 (见第 1200 页)																											
COMPB																											
PWM2CMPB, 类型 R/W, 偏移量 0x0DC, 复位 0x0000.0000 (见第 1200 页)																											
COMPB																											
PWM3CMPB, 类型 R/W, 偏移量 0x11C, 复位 0x0000.0000 (见第 1200 页)																											
COMPB																											
PWM0GENA, 类型 R/W, 偏移量 0x060, 复位 0x0000.0000 (见第 1201 页)																											
				ACTCMPBD				ACTCMPBU				ACTCMPAD				ACTCMPAU				ACTLOAD				ACTZERO			
PWM1GENA, 类型 R/W, 偏移量 0x0A0, 复位 0x0000.0000 (见第 1201 页)																											
				ACTCMPBD				ACTCMPBU				ACTCMPAD				ACTCMPAU				ACTLOAD				ACTZERO			
PWM2GENA, 类型 R/W, 偏移量 0x0E0, 复位 0x0000.0000 (见第 1201 页)																											
				ACTCMPBD				ACTCMPBU				ACTCMPAD				ACTCMPAU				ACTLOAD				ACTZERO			
PWM3GENA, 类型 R/W, 偏移量 0x120, 复位 0x0000.0000 (见第 1201 页)																											
				ACTCMPBD				ACTCMPBU				ACTCMPAD				ACTCMPAU				ACTLOAD				ACTZERO			
PWM0GENB, 类型 R/W, 偏移量 0x064, 复位 0x0000.0000 (见第 1204 页)																											
				ACTCMPBD				ACTCMPBU				ACTCMPAD				ACTCMPAU				ACTLOAD				ACTZERO			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
PWM1GENB, 类型 R/W, 偏移量 0x0A4, 复位 0x0000.0000 ( 见第 1204 页 )																											
				ACTCMPBD				ACTCMPBU				ACTCMPAD				ACTCMPAU				ACTLOAD				ACTZERO			
PWM2GENB, 类型 R/W, 偏移量 0x0E4, 复位 0x0000.0000 ( 见第 1204 页 )																											
				ACTCMPBD				ACTCMPBU				ACTCMPAD				ACTCMPAU				ACTLOAD				ACTZERO			
PWM3GENB, 类型 R/W, 偏移量 0x124, 复位 0x0000.0000 ( 见第 1204 页 )																											
				ACTCMPBD				ACTCMPBU				ACTCMPAD				ACTCMPAU				ACTLOAD				ACTZERO			
PWM0DBCTL, 类型 R/W, 偏移量 0x068, 复位 0x0000.0000 ( 见第 1206 页 )																											
														ENABLE													
PWM1DBCTL, 类型 R/W, 偏移量 0x0A8, 复位 0x0000.0000 ( 见第 1206 页 )																											
														ENABLE													
PWM2DBCTL, 类型 R/W, 偏移量 0x0E8, 复位 0x0000.0000 ( 见第 1206 页 )																											
														ENABLE													
PWM3DBCTL, 类型 R/W, 偏移量 0x128, 复位 0x0000.0000 ( 见第 1206 页 )																											
														ENABLE													
PWM0DBRISE, 类型 R/W, 偏移量 0x06C, 复位 0x0000.0000 ( 见第 1207 页 )																											
														RISEDELAY													
PWM1DBRISE, 类型 R/W, 偏移量 0x0AC, 复位 0x0000.0000 ( 见第 1207 页 )																											
														RISEDELAY													
PWM2DBRISE, 类型 R/W, 偏移量 0x0EC, 复位 0x0000.0000 ( 见第 1207 页 )																											
														RISEDELAY													
PWM3DBRISE, 类型 R/W, 偏移量 0x12C, 复位 0x0000.0000 ( 见第 1207 页 )																											
														RISEDELAY													
PWM0DBFALL, 类型 R/W, 偏移量 0x070, 复位 0x0000.0000 ( 见第 1208 页 )																											
														FALLDELAY													
PWM1DBFALL, 类型 R/W, 偏移量 0x0B0, 复位 0x0000.0000 ( 见第 1208 页 )																											
														FALLDELAY													
PWM2DBFALL, 类型 R/W, 偏移量 0x0F0, 复位 0x0000.0000 ( 见第 1208 页 )																											
														FALLDELAY													
PWM3DBFALL, 类型 R/W, 偏移量 0x130, 复位 0x0000.0000 ( 见第 1208 页 )																											
														FALLDELAY													
PWM0FLTSRC0, 类型 R/W, 偏移量 0x074, 复位 0x0000.0000 ( 见第 1209 页 )																											
												FAULT3		FAULT2		FAULT1		FAULT0									
PWM1FLTSRC0, 类型 R/W, 偏移量 0x0B4, 复位 0x0000.0000 ( 见第 1209 页 )																											
												FAULT3		FAULT2		FAULT1		FAULT0									



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
PWM3FLTSTAT0, 类型 -, 偏移量 0x984, 复位 0x0000.0000 (见第 1216 页)																			
												FAULT3	FAULT2	FAULT1	FAULT0				
PWM0FLTSTAT1, 类型 -, 偏移量 1x808, 复位 0x0000.0000 (见第 1218 页)																			
												DCMP7	DCMP6	DCMP5	DCMP4	DCMP3	DCMP2	DCMP1	DCMP0
PWM1FLTSTAT1, 类型 -, 偏移量 1x888, 复位 0x0000.0000 (见第 1218 页)																			
												DCMP7	DCMP6	DCMP5	DCMP4	DCMP3	DCMP2	DCMP1	DCMP0
PWM2FLTSTAT1, 类型 -, 偏移量 0x908, 复位 0x0000.0000 (见第 1218 页)																			
												DCMP7	DCMP6	DCMP5	DCMP4	DCMP3	DCMP2	DCMP1	DCMP0
PWM3FLTSTAT1, 类型 -, 偏移量 0x988, 复位 0x0000.0000 (见第 1218 页)																			
												DCMP7	DCMP6	DCMP5	DCMP4	DCMP3	DCMP2	DCMP1	DCMP0
PWMPP, 类型 RO, 偏移量 0xFC0, 复位 0x0000.0344 (见第 1221 页)																			
				ONE	EFAULT	ESYNC	FCNT				GCNT								
PWMP, 类型 R/W, 偏移量 0xFC4, 复位 0x0000.0000 (见第 1223 页)																			
												USERWMDIV		PWMDIV					
<b>正交编码器接口 (QEI)</b>																			
QEIO 基址: 0x4002.C000																			
QEI1 基址: 0x4002.D000																			
QEICTL, 类型 R/W, 偏移量 0x000, 复位 0x0000.0000 (见第 1230 页)																			
												FILTCNT							
		FILTEN	STALLEN	INVI	INVB	INVA	VELDIV		VELEN	RESMODE	CAPMODE	SIGMODE	SWAP	ENABLE					
QEISTAT, 类型 RO, 偏移量 0x004, 复位 0x0000.0000 (见第 1233 页)																			
														DIRECTION	ERROR				
QEIPPOS, 类型 R/W, 偏移量 0x008, 复位 0x0000.0000 (见第 1234 页)																			
POSITION																			
POSITION																			
QEIMAXPOS, 类型 R/W, 偏移量 0x00C, 复位 0x0000.0000 (见第 1235 页)																			
MAXPOS																			
MAXPOS																			
QEILOAD, 类型 R/W, 偏移量 0x010, 复位 0x0000.0000 (见第 1236 页)																			
LOAD																			
LOAD																			
QEITIME, 类型 RO, 偏移量 0x014, 复位 0x0000.0000 (见第 1237 页)																			
TIME																			
TIME																			
QEICOUNT, 类型 RO, 偏移量 0x018, 复位 0x0000.0000 (见第 1238 页)																			
COUNT																			
COUNT																			
QEISPEED, 类型 RO, 偏移量 0x01C, 复位 0x0000.0000 (见第 1239 页)																			
SPEED																			
SPEED																			
QEINTEN, 类型 R/W, 偏移量 0x020, 复位 0x0000.0000 (见第 1240 页)																			
												INTERROR	INTDIR	INTTIMER	INTINDEX				

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QEIRIS, 类型 RO, 偏移量 0x024, 复位 0x0000.0000 ( 见第 1241 页 )															
												INTERROR	INTDIR	INTTIMER	INTINDEX
QEISC, 类型 R/W1C, 偏移量 0x028, 复位 0x0000.0000 ( 见第 1242 页 )															
												INTERROR	INTDIR	INTTIMER	INTINDEX

## B 订购和联系信息

### B.1 订购信息

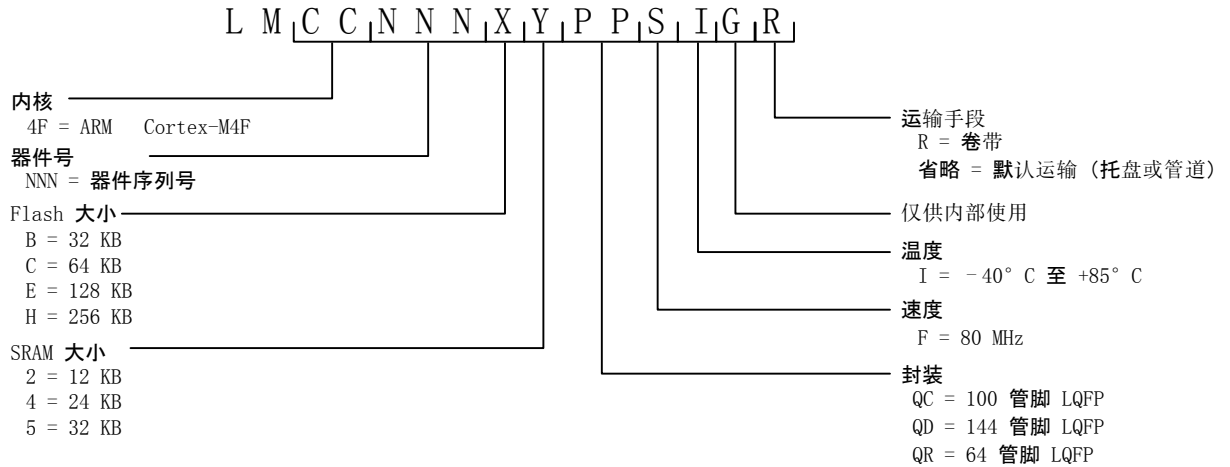


表 B-1. 器件订购信息

可订购的器件型号	描述
LM4F232H5QDFIG	Stellaris® LM4F232H5QD 微控制器工业温度范围 144 管脚 LQFP
LM4F232H5QDFIGR	Stellaris LM4F232H5QD 微控制器工业温度范围 144 管脚 LQFP 卷带

### B.2 器件标记

该 Stellaris 微控制器带有标识号标记。此代码包含以下信息：

- 第一行和第二行表示器件号，例如 LM4F232H5QDFIGA0。器件号中的第二个字母表示产品状态。M 表示该器件完全合格并已发布。X，例如 LX4F232H5QDFIGA0，表示该器件仍处于试验阶段，需要提供责任豁免声明。
- 第三行包含内部跟踪编号。

### B.3 套件

该 Stellaris 系列为工程师提供了各种硬件和软件工具，以便快速开始进行产品研发。

- 参考设计套件通过提供可以运行的硬件和完整的文档来加速产品开发，这些完整的文档包含了硬件设计文件
- 评估套件使用户可在购买之前评估 Stellaris 微控制器，经济实惠且效率极高
- 开发套件为你提供所有需要开发的工具和原型嵌入式应用

有关最新工具，请访问 [www.ti.com/stellaris](http://www.ti.com/stellaris) 网站或者咨询分销商。

## B.4 支持信息

关于 Stellaris 产品的支持信息，请就近联系 TI 全球产品信息中心：  
<http://www-k.ext.ti.com/sc/technical-support/product-information-centers.htm>。

# C 封装信息

## C.1 144 管脚 LQFP 封装

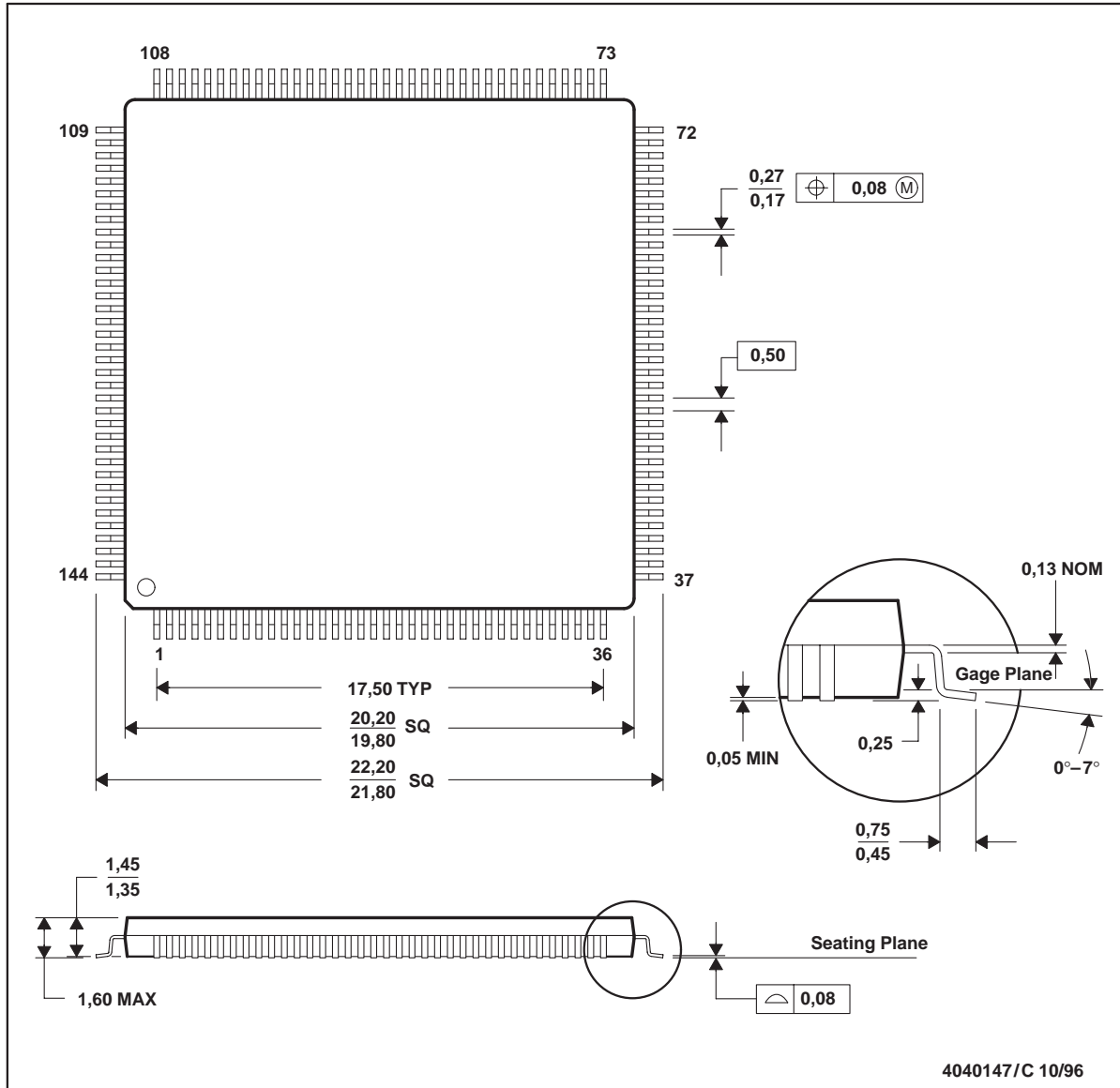
### C.1.1 封装尺寸

图 C-1. Stellaris LM4F232H5QD 144 管脚 LQFP 封装尺寸

MTQF017A – OCTOBER 1994 – REVISED DECEMBER 1996

PGE (S-PQFP-G144)

PLASTIC QUAD FLATPACK



- NOTES: A. All linear dimensions are in millimeters.  
 B. This drawing is subject to change without notice.  
 C. Falls within JEDEC MS-026