

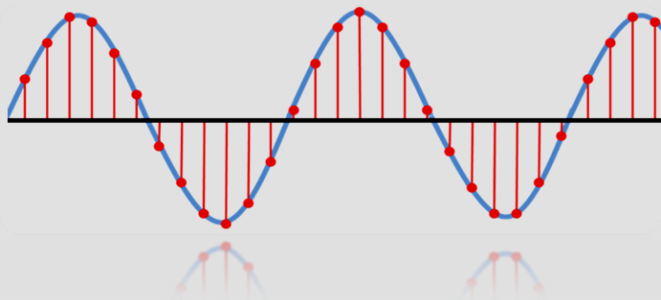
# Stellaris® ARM® Cortex™ M4F 培训



Peripheral Overview



Low Power



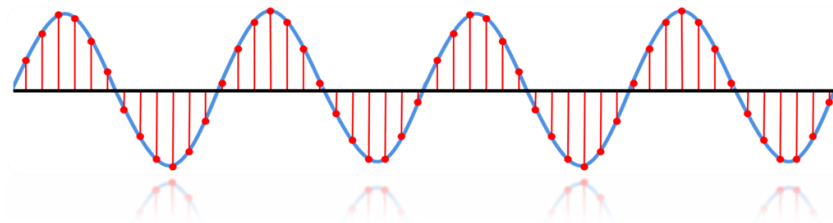
Analog Peripherals



USB

# Stellaris® ARM® Cortex™ M4F 培训

## 模拟外设



**ADC 与模拟比较器**

# 议程 — 模拟外设

- Section 1 (Key Concepts)
  - Part 1: Analog-to-Digital Converter
  - Part 2: Analog Comparators



- Section 2
  - Getting started with ADC using EK-LM4F232
- Section 3
  - Exercises

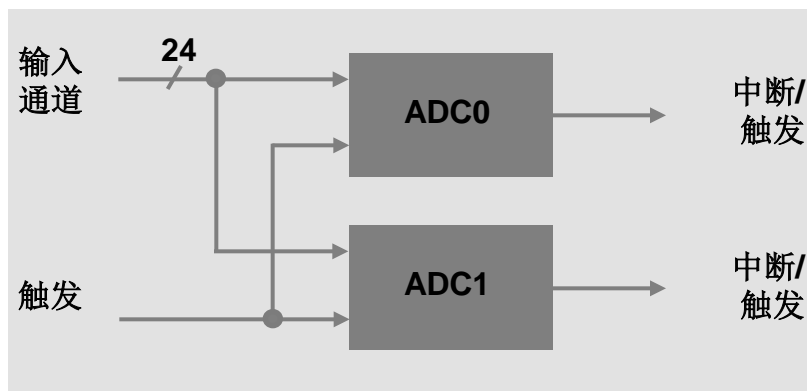


# 第一部分：模数转换器

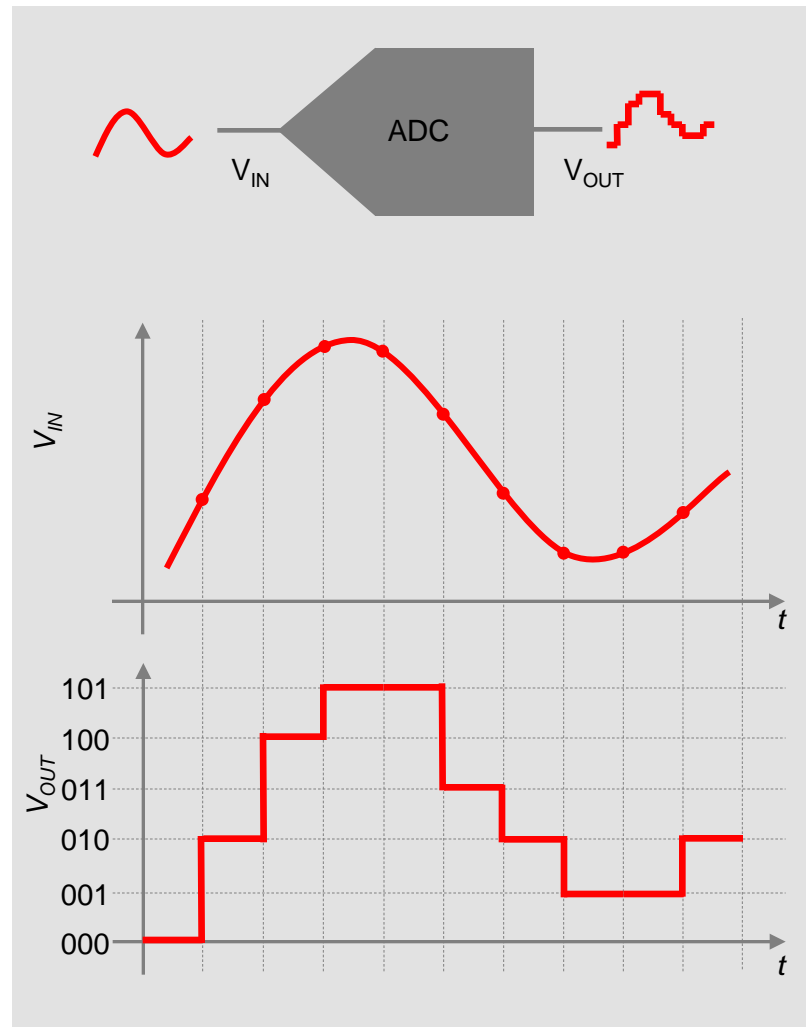
- 模数转换器
- 主要特性与规格
- 方框图与信号说明
- 功能说明
  - ADC 模块时钟
  - 采样序列
  - ADC 参考电压
  - 差分采样
  - 采样相位控制
  - 硬件采样平均电路
  - 内部温度传感器
  - 数字比较器单元

# 模数转换器

- Stellaris LM4F MCU 具有两个 ADC 模块（ADC0 和 ADC1），可用于将连续的模拟电压转换为离散的数字值
- 每个 ADC 模块具有 12 位分辨率
- 每个 ADC 模块独立运作，因而能够：
  - 执行不同的采样序列
  - 对 24 个模拟输入通道中的任一个进行采样
  - 产生不同的中断与触发源



Stellaris LM4F MCU 中的 ADC 可实现方案




模数转换器

# 第一部分：模数转换器

- 模数转换器
- **主要特性与规格**
- 方框图与信号说明
- 功能说明
  - ADC 模块时钟
  - 样本排序器
  - ADC 电压基准
  - 差分采样
  - 样本相位控制
  - 硬件样本平均电路
  - 内部温度传感器
  - 数字比较器单元

# 主要特性



- 12 位精度 ADC
- 24 个共用模拟输入通道 
- 单端与差分输入配置
- 片上内部温度传感器
- 最大采样速率为每秒 100 万次采样 (1MSPS)
- 可选的基准信号 (VDDA、GNDA 或两个外部电压\*) 
- 4 个可编程采样转换排序器
- 灵活的触发控制
  - 控制器/软件
  - 定时器
  - 模拟比较器
  - 脉宽调制 (PWM)
  - 通用输入输出 (GPIO)
- 对多达 64 个采样进行硬件平均处理以改善准确度
- 每个 ADC 带 8 个数字比较器
- 采用  $\mu$ DMA 的高效传输
- 采样时间里的任选相移, 可设置范围从 22.5 ° 至 337.5 °

\* 64 引脚封装除外

# 主要规格

规格*	符号	Stellaris LM4F 系列
ADC 通道的数量	n	24
分辨率	N	12 位
ADC 电源电压	$V_{DDA}$	2.9 至 3.6V
	$V_{REFA+}$	2.4V 至 $V_{DDA}$
	$V_{REFA-}$	0 至 0.3 V
采用内部基准电压的单端、全标度模拟输入	$V_{ADCIN}$	0 至 $V_{DDA}$
采用外部基准电压的单端、全标度模拟输入		$V_{REFA-}$ 至 $V_{REFA+}$
采用内部基准电压的差分、全标度模拟输入		$-V_{DDA}$ 至 $V_{DDA}$
采用外部基准电压的差分、全标度模拟输入		$-[V_{REFA+} - V_{REFA-}]$ 至 $[V_{REFA+} - V_{REFA-}]$
转换时间	$T_{ADCCONV}$	1 $\mu$ s
转换速率	$F_{CONV}$	1 MSPS
积分非线性误差	INL	$\pm 1.5$ LSB
差分非线性误差	DNL	$\pm 0.8$ LSB
偏移误差	$E_O$	$\pm 1$ LSB
全标度增益误差	$E_G$	$\pm 2$ LSB
总不可调整误差	$E_T$	$\pm 3.5$ LSB

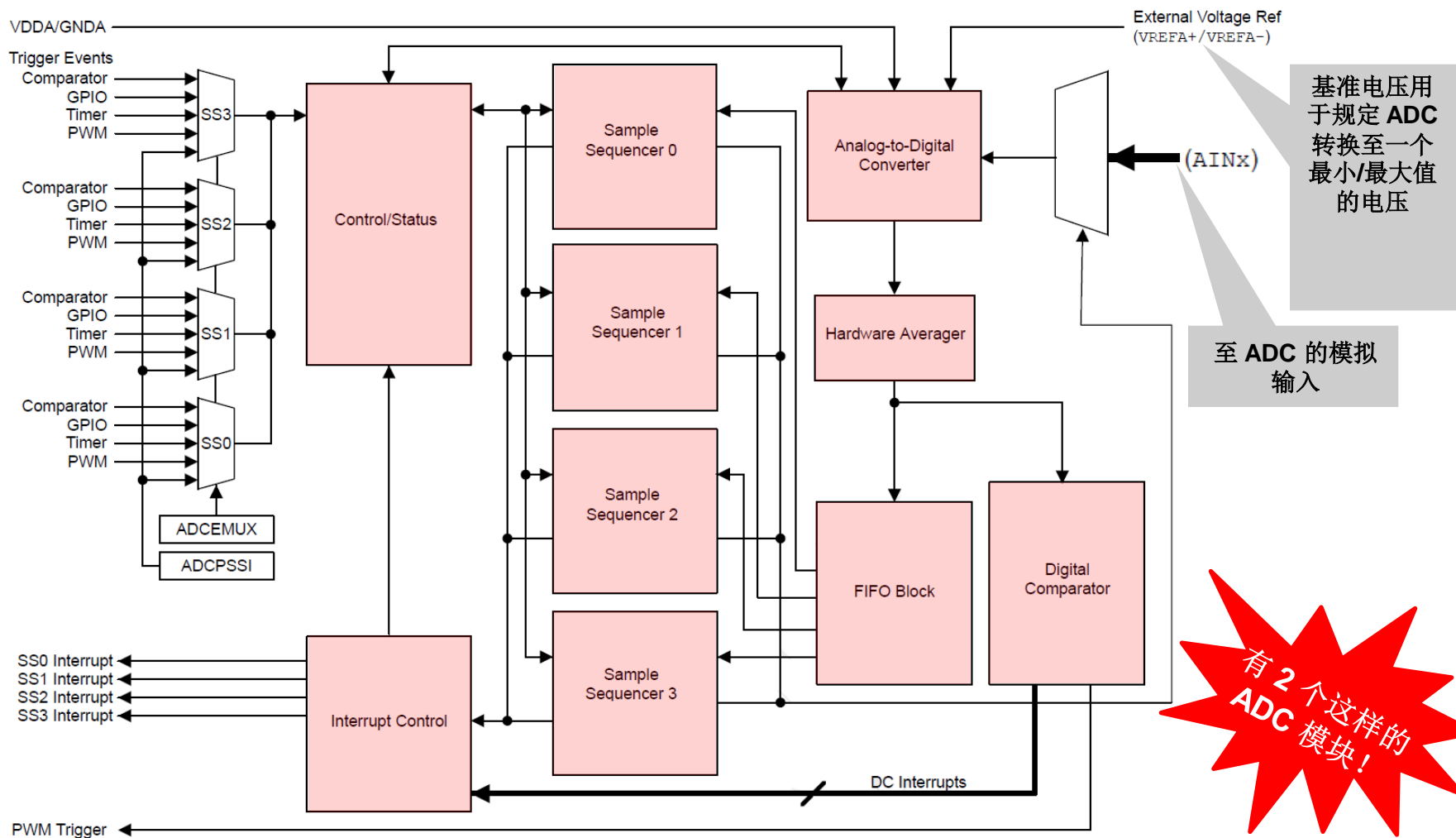
\* 在  $T = -40^{\circ}\text{C}$  至  $85^{\circ}\text{C}$  的温度范围内以及 12 位分辨率和 1MSPS 采样速率条件下。



# 第一部分：模数转换器

- 模数转换器
- 主要特性与规格
- 方框图与信号说明
- 功能说明
  - ADC 模块时钟
  - 样本排序器
  - ADC 电压基准
  - 差分采样
  - 样本相位控制
  - 硬件样本平均电路
  - 内部温度传感器
  - 数字比较器单元

# 方框图与信号说明



ADC 模块的功能方框图

# 第一部分：模数转换器

- 模数转换器
- 主要特性与规格
- 方框图与信号说明
- 功能说明
  - ADC 模块时钟
  - 样本排序器
  - ADC 电压基准
  - 差分采样
  - 样本相位控制
  - 硬件样本平均电路
  - 内部温度传感器
  - 数字比较器单元

# 功能说明

## ADC 模块定时

- 时钟源
  - ADC 控制逻辑电路的运行频率为16 MHz。
  - 根据 ADCCC 寄存器中 CS 位的数值，可以将 PLL 或 PIOSC (16 MHz) 选择作为一个时钟源。
  - 如果 ADC 模块未将 PLL/PIOSC 用作其时钟源，则系统时钟的运行频率至少必须为 16
- 低功耗模式中的操作
  - 假如 PIOSC 被选择作为 ADC 控制逻辑电路的时钟源，则 ADC 可工作于深度睡眠模式。
- 必须先通过设定 RCGCADC 寄存器中的 Rx 位(对于 ADC0, x=0; 对于 ADC1, x=1) 以启用 ADC 时钟，然后才能使用 ADC 模块。

# 采样序列

- Stellaris LM4F 器件上的 ADC 采用一种基于可编程序列的方法进行数据的收集和采样。
- 每个采样序列都是一系列完全可编程的连续（背对背）样本，因而允许 ADC 模块不必重新配置即可从多个输入信号源收集数据。
- 每个 ADC 模块具有 **4 个采样排序器**，用于控制采样与数据捕获。
- 所有的采样排序器都是相同的，唯一的差别在于其能够捕获的样本数量以及 FIFO 的深度。
- 如欲配置一个采样序列，则需要下列信息：
  - 输入信号源
  - 模式（单端或差分）
  - 采样结束时的中断发生
  - 用于序列中最后一个样本的指示符
- **μDMA 操作：**每个 ADC 样本排序器具有一个专用的 μDMA 通道。每个样本排序器能够独立地传输数据。

排序器	样本的数量	FIFO 的深度
SS 3	1	1
SS 2	4	4
SS 1	4	4
SS 0	8	8

ADC 样本排序器

# ADC 基准电压

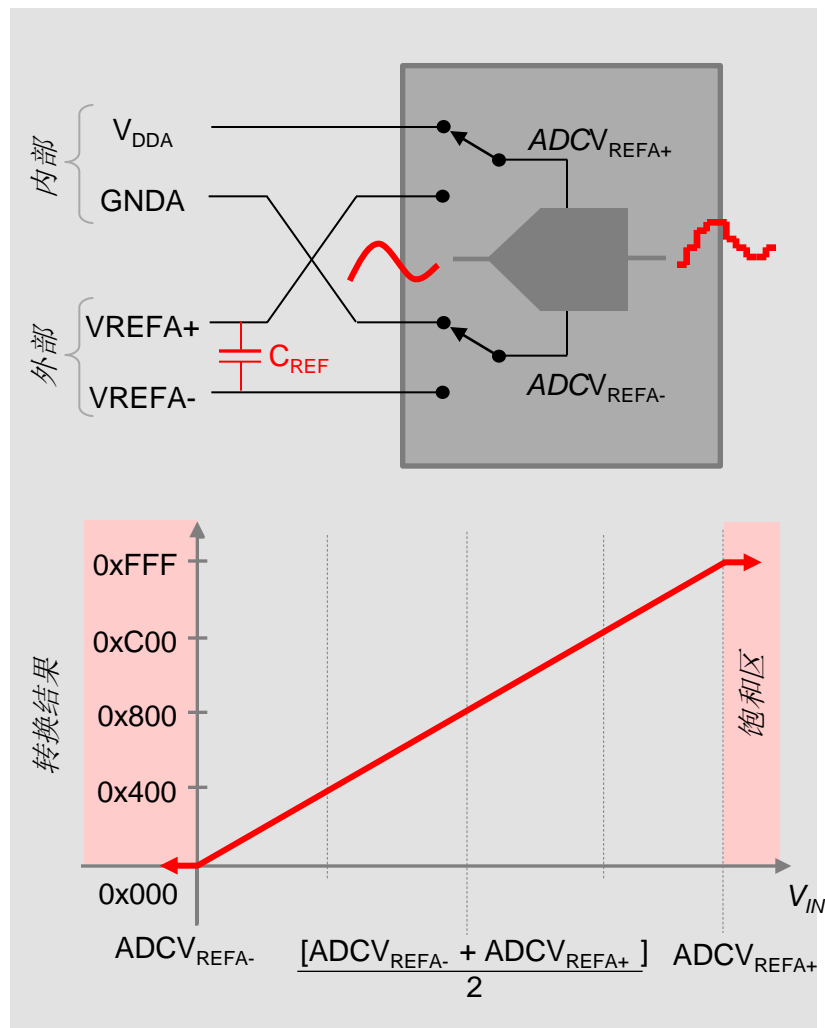
- ADC 采用  $ADCV_{REFA+}$  和  $ADCV_{REFA-}$  基准电压以产生一个转换值。
- ADC 的内部/外部基准电压可以采用 ADCCTL 寄存器中的 VREF 位来选择。
- 分辨率（在单端模式中）：

$$mv/ADC\ Code = \frac{ADCV_{REF+} - ADCV_{REF-}}{4096}$$

- 范围：

$$\begin{aligned} V_{IN} &\in \{ADCV_{REFA-}, ADCV_{REFA+}\} \\ V_{REFA-} &\in \{0, 0.3V\} \\ V_{REFA+} &\in \{2.4V, V_{DDA}\} \\ Output &\in \{0x000, 0xFFF\} \end{aligned}$$

- ADC 在欠压和过压情况下发生饱和（阴影区域）



ADC 电压基准与转换范围

# 差分采样

- ADC 还允许对两个模拟输入通道进行差分采样。
  - 差分输入对可在 ADCSSMUXn 寄存器中进行配置。
  - 差分采样可通过设定 ADCSSCTL0 中的 Dn 位来启用。
- 差分输入对 ‘n’ 负责对连续的偶数和奇数模拟输入通道之间的电压差 ( $\Delta V_{IN}$ ) 进行采样。

$$\Delta V_{IN} = V_{EVEN\_IN\_CH} - V_{ODD\_IN\_CH} = V_{2n} - V_{2n+1}$$

where,  $n \in \{0,11\}$

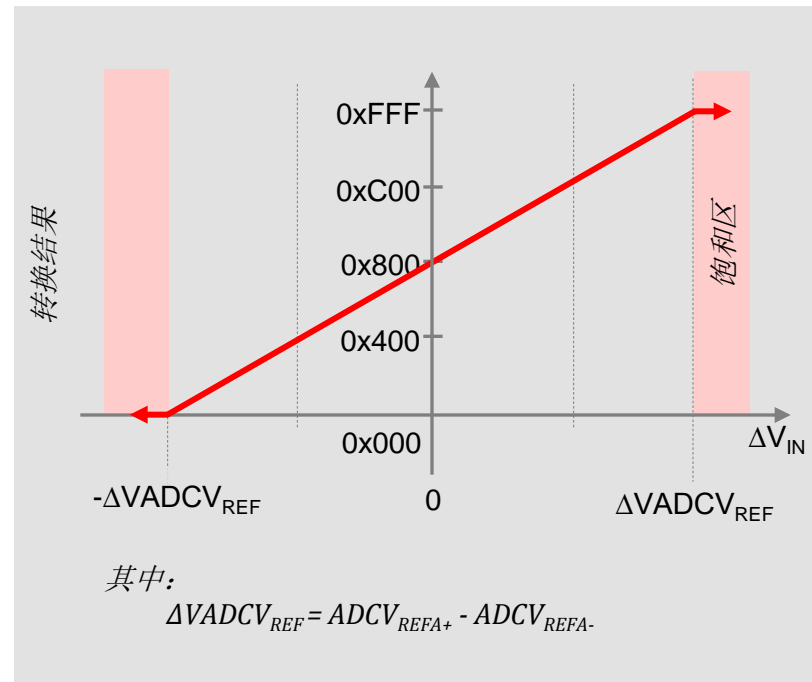
- 对于转换准确度:

$$\Delta V_{IN} \in \{-\Delta V_{ADC V_{REF}}, \Delta V_{ADC V_{REF}}\}$$

- 分辨率:

$$mv/ADC \text{ Code} = 2 \times \frac{[ADCV_{REF+} - ADCV_{REF-}]}{4096}$$

- ADC 转换在欠压和过压情况下饱和 (阴影区域)

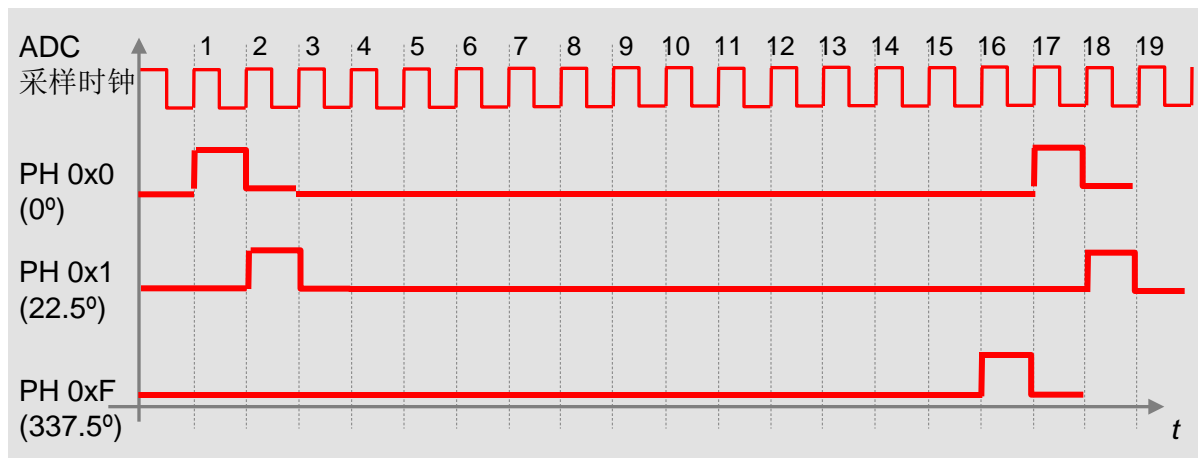


差分模式中的 ADC 转换范围

# 样本相位控制

## • ADC 样本相位

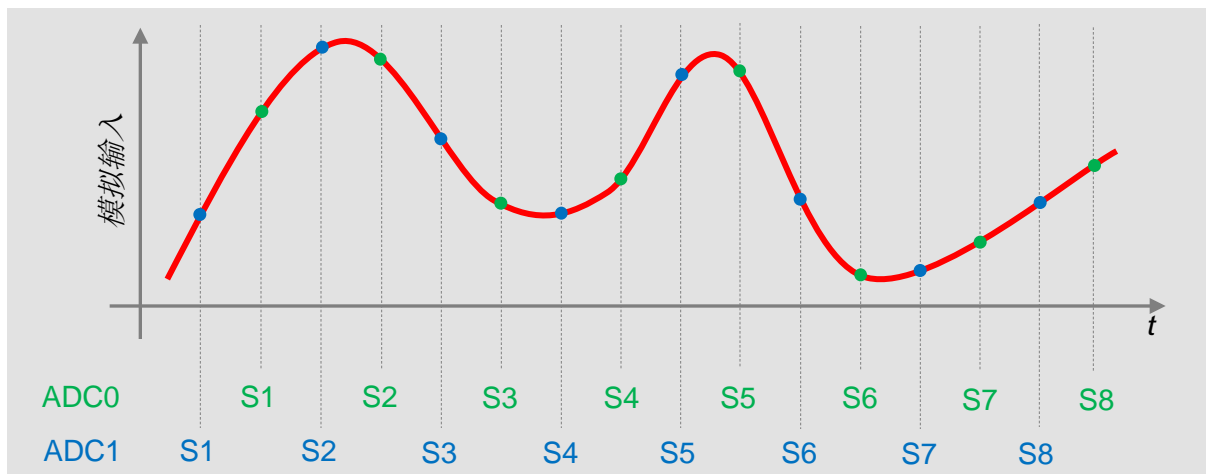
- ADC0 和 ADC1 可采用相同的触发信号源来操作。
- 如果它们是处于相同频率的采样数据，则转换的起点可分 15 个  $22.5^\circ$  的离散增量（从  $0^\circ$  至高达  $337.5^\circ$ ）进行延迟。



ADC 样本相位

## • 时滞采样

- ADC0 和 ADC1 可以彼此异相的方式使用。
- 采样数据可利用软件进行组合。
- 这实际上可将转换带宽倍增至高达 **2MSPS**。

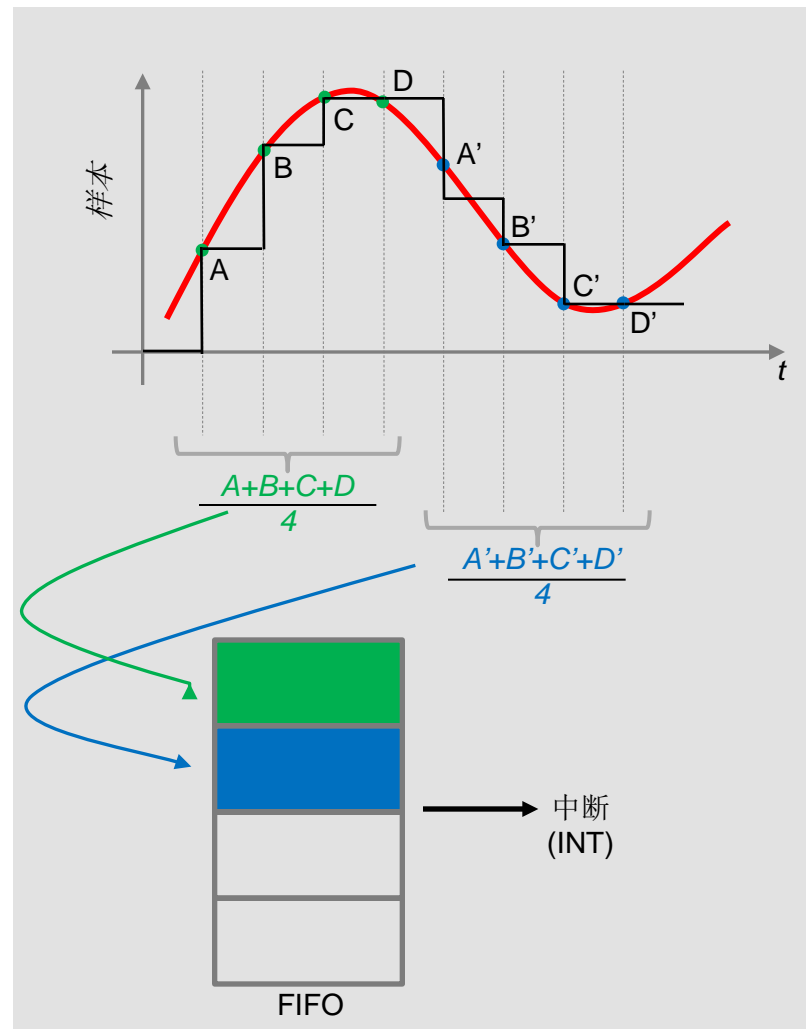


时滞采样



# 硬件样本平均电路

- 硬件样本平均电路可用于生成精度较高的结果。
- 可对多达 **64** 个样本进行累加与平均处理以在排序器 FIFO 中形成单个数据登记项。
- 按照默认设置，硬件样本平均电路是关断的。
  - 所有来自转换器的数据均通过排序器
  - 平均处理受控于 ADCSAC 寄存器
- 对所有的通道进行同等的平均处理，而不考虑其配置（单端或差分）。
- **示例：**如果 ADCSAC 寄存器中的 AVG 值为 **0x02**，则将进行 **4x** 平均处理。假如 ADCSSCTL0 寄存器中的 IE 位被置位，那么当 FIFO 获得第二个数据登记项时将生成一个中断。
- **折衷：**吞吐量随着平均计算中样本数量的增加成比例地降低。



样本平均处理示例

# 内部温度传感器

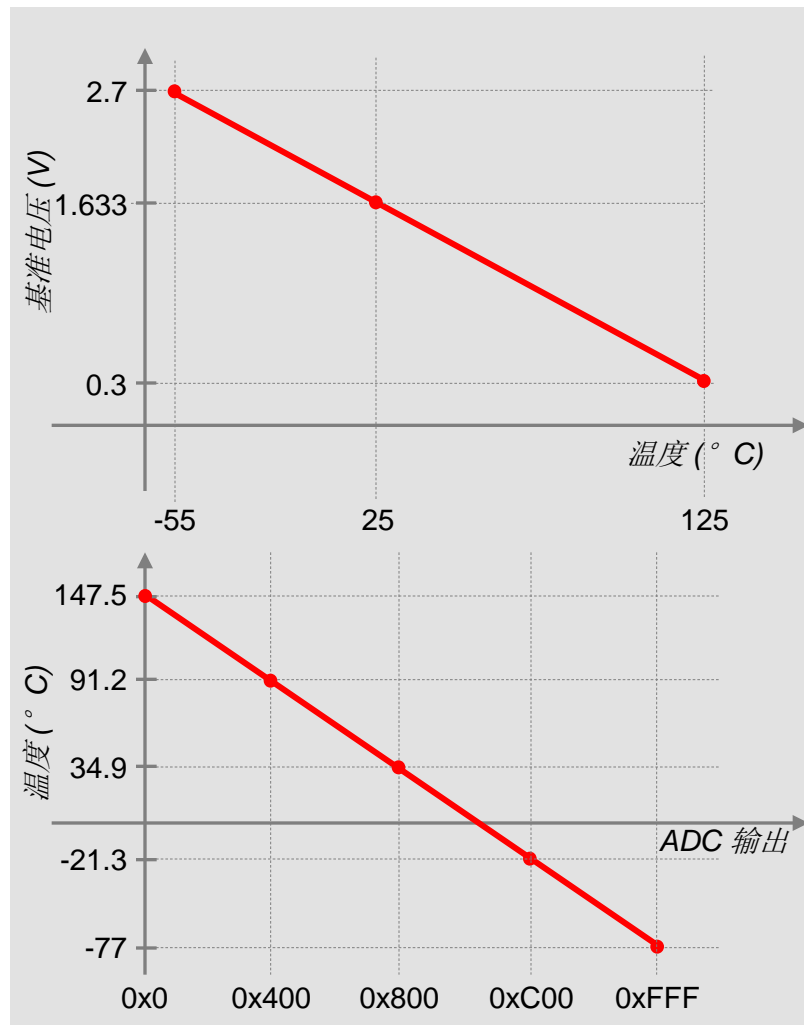
- 内部温度传感器模块包括：
  - 负责给各种模拟外设提供基准电压的带隙基准电路。

$$V_{REF} = 2.7 - \frac{T_{[^{\circ}C]} + 55}{75}$$

- 片上内部温度传感器。

$$T_{[^{\circ}C]} = 147.5 - \frac{225 \times ADC_{OUTPUT}}{4095}$$

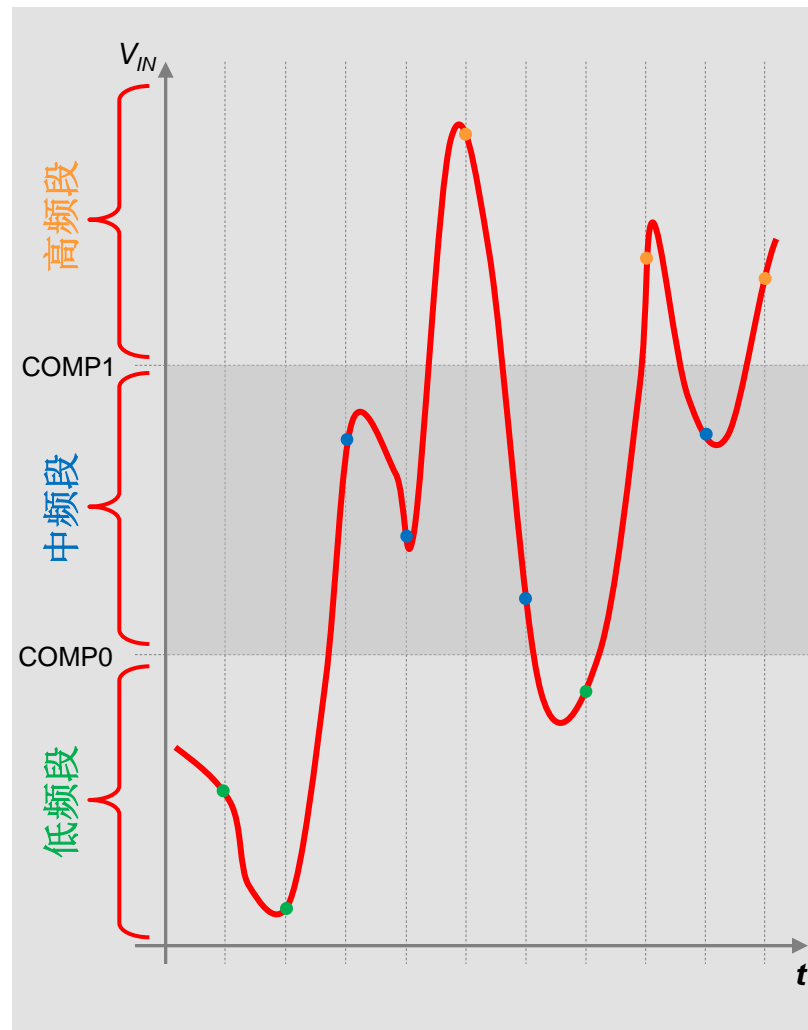
- 内部温度传感器服务于以下主要用途：
  - 检测晶片温度以实现可靠的系统操作。
  - 提供温度测量以校准冬眠模块的 RTC 调整值。
- 温度可通过设定 ADCSSCTLn 寄存器中的 TSn 位进行采样。



内部温度传感器特性

# 数字比较器单元

- 数字比较器负责比较 ADC 模块的输出与用户可编程限值。根据比较的结果，可相应地生成一个处理器中断信号或针对 PWM 模块的触发信号。
- 每个 ADC 模块包含 **8 个数字比较器**。
- 操作模式：
  - 始终运行模式
  - 一次模式
  - 迟滞模式
  - 迟滞始终运行模式
- 模式可以使用 ADCCTLn 寄存器中的 CIM 或 CTM 位来选择。
- 功能范围：
  - 低频段
  - 中频段
  - 高频段
- 功能范围可以使用 ADCDCCMPn 寄存器中的 COMP0 和 COMP1 位来选择。
- 始终不变的是： $COMP1 \geq COMP0$ 。



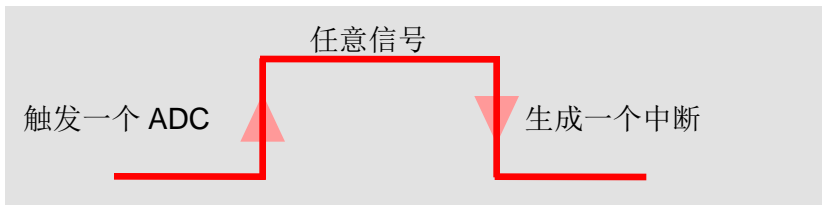
数字比较器功能范围

# 第二部分：模拟比较器

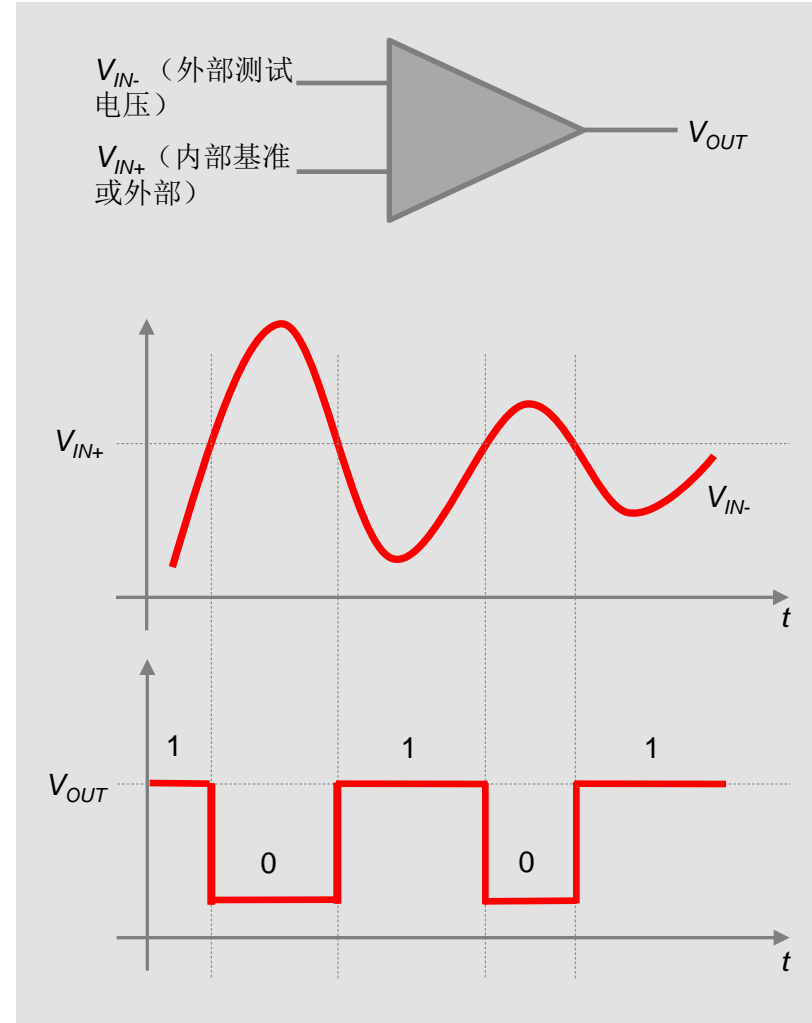
- 主要特性
- 方框图与信号说明
- 功能说明

# 模拟比较器的主要特性

- 模拟比较器用于比较两个模拟电压，并根据比较的结果提供一个逻辑输出。
- 集成在 **Stellaris MCU** 中的 3 个模拟比较器可单独用于：
  - 比较两个模拟信号并替代一个外部/分立模拟比较器以节省板级空间和系统成本。
  - 驱动一个外部引脚
  - 触发一个 **ADC**
  - 采用中断向某个应用电路发送指示信号

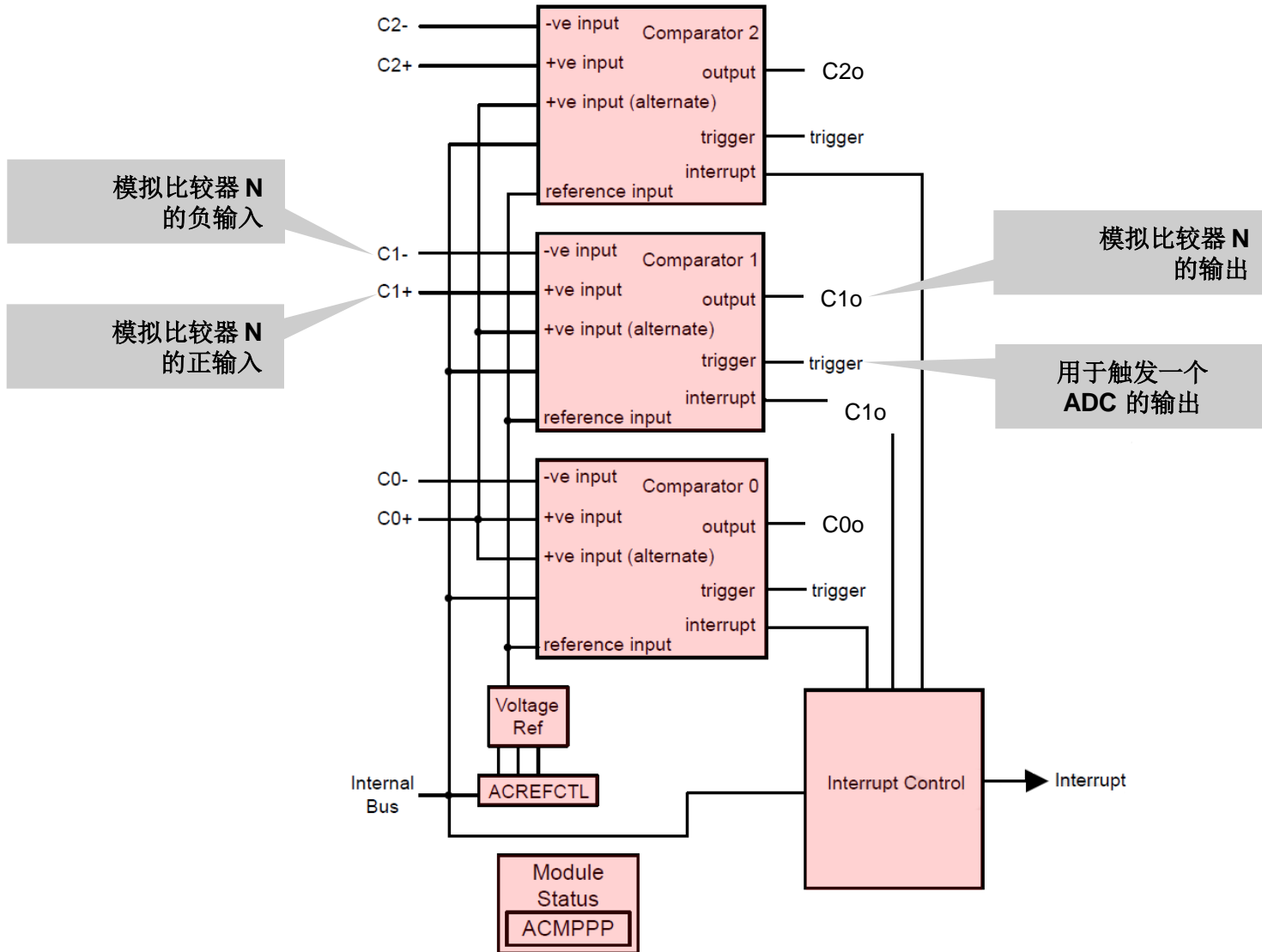


独立地使用模拟比较器



模拟比较器：输入和输出

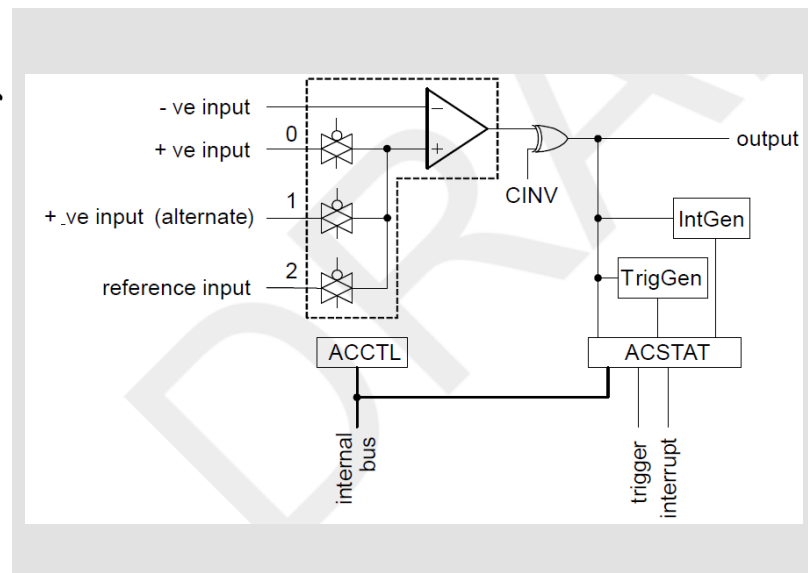
# 方框图与信号说明



模拟比较器模块的功能方框图

# 功能说明

- 如果  $V_{IN^-} < V_{IN^+} \Rightarrow V_{OUT} = 1$
- 如果  $V_{IN^-} > V_{IN^+} \Rightarrow V_{OUT} = 0$
- $V_{IN^-}$  的信号源是一个外部输入 (C0-).
- $V_{IN^+}$  的信号源可以是下面的任一个，就是说：可以将一个测试电压（即： $V_{IN^-}$ ）与下列电压中的任一个进行比较：
  - 一个专用的外部基准电压
  - 单个共用的外部基准电压 (C0+)
  - 一个共用的内部基准电压，( $V_{IREF}$ )
- 比较器通过 ACCTL 和 ACSTAT 寄存器进行配置。
- 内部电压基准通过 ACREFCTL 寄存器进行配置。
- 中断通过 ACMIS、ACRIS 和 ACINTEN 寄存器进行配置。



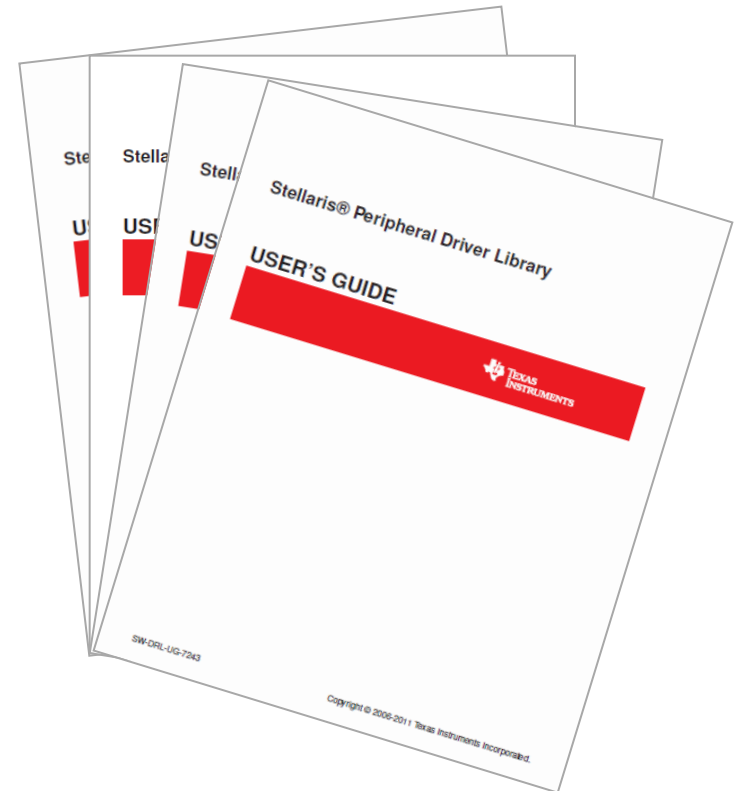
比较器单元的结构

# 易用性与驱动程序库 API

- 在驱动程序库 (DriverLib) 中提供了 API，助力跨越式启动开发工作。

- void **ADCComparatorConfigure** (unsigned long ulBase, unsigned long ulComp, unsigned long ulConfig)
- void **ADCComparatorIntClear** (unsigned long ulBase, unsigned long ulStatus)
- void **ADCComparatorIntDisable** (unsigned long ulBase, unsigned long ulSequenceNum)
- void **ADCComparatorIntEnable** (unsigned long ulBase, unsigned long ulSequenceNum)
- unsigned long **ADCComparatorIntStatus** (unsigned long ulBase)
- void **ADCComparatorRegionSet** (unsigned long ulBase, unsigned long ulComp, unsigned long ulLowRef, unsigned long ulHighRef)
- void **ADCComparatorReset** (unsigned long ulBase, unsigned long ulComp, tBoolean bTrigger, tBoolean bInterrupt)
- void **ADCHardwareOversampleConfigure** (unsigned long ulBase, unsigned long ulFactor)
- void **ADCIntClear** (unsigned long ulBase, unsigned long ulSequenceNum)
- void **ADCIntDisable** (unsigned long ulBase, unsigned long ulSequenceNum)
- void **ADCIntEnable** (unsigned long ulBase, unsigned long ulSequenceNum)
- void **ADCIntRegister** (unsigned long ulBase, unsigned long ulSequenceNum, void (\*pfnHandler)(void))
- unsigned long **ADCIntStatus** (unsigned long ulBase, unsigned long ulSequenceNum, tBoolean bMasked)
- void **ADCIntUnregister** (unsigned long ulBase, unsigned long ulSequenceNum)
- unsigned long **ADCPhaseDelayGet** (unsigned long ulBase)
- void **ADCPhaseDelaySet** (unsigned long ulBase, unsigned long ulPhase)
- void **ADCProcessorTrigger** (unsigned long ulBase, unsigned long ulSequenceNum)
- unsigned long **ADCReferenceGet** (unsigned long ulBase)
- void **ADCReferenceSet** (unsigned long ulBase, unsigned long ulRef)
- void **ADCSequenceConfigure** (unsigned long ulBase, unsigned long ulSequenceNum, unsigned long ulTrigger, unsigned long ulPriority)
- long **ADCSequenceDataGet** (unsigned long ulBase, unsigned long ulSequenceNum, unsigned long \*pulBuffer)

.. 其他还有很多。



下载用户驱动程序库手册，[www.ti.com/stellaris](http://www.ti.com/stellaris)



# 想了解更多信息吗？

- 文档

- 器件的数据表

- [www.ti.com/stellaris](http://www.ti.com/stellaris)

- 驱动程序库用户手册

- 勘误表等



- 软件

- 代码范例

- 驱动程序库 API 等

- [www.ti.com/stellarisware](http://www.ti.com/stellarisware)



- 联系 Stellaris 专家！

- 培训与技术支持

- [e2e.ti.com](http://e2e.ti.com)

- 市场营销信息

- [stellaris-marketing@ti.com](mailto:stellaris-marketing@ti.com)



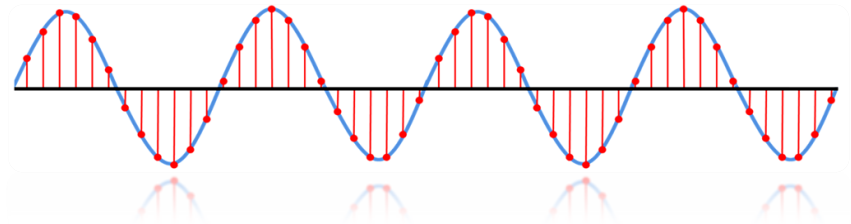
# 动手实验班练习

请继续阅读第 2 节和第 3 节，了解动手实验课程方面的信息。

第 2 节：采用 EK-LM4F232 启动 ADC 的开发

第 3 节：实习

# 备用幻灯片



# ADC 与 $\mu$ DMA 操作

备用

- $\mu$ DMA 可用于完成从 ADC 样本排序器的高效数据传输，而无需 CPU 的干预或进行重新配置。
- 每个 ADC 样本排序器具有一个专用的  $\mu$ DMA 通道。
  - 因此，每个样本排序器都能独立地传输数据。
- ADC 模块从每个样本排序器向关联的  $\mu$ DMA 通道提供一个突发请求信号以进行直接存储器存取。
  - 当 ADCSSCTLn 寄存器中的 IE 位被置位时，将启动一个突发传输请求。
  - 不支持单一传输请求。
- $\mu$ DMA 传输的仲裁规模必须是一个二次幂。

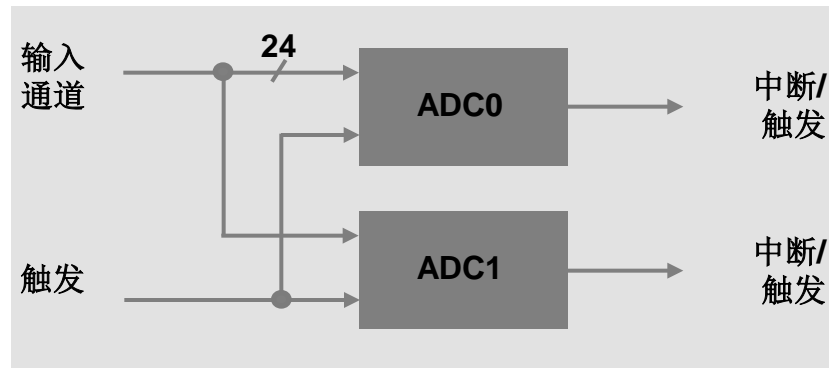
ADCx SSy	$\mu$ DMA 通道
ADC0 SS0	14
ADC0 SS1	15
ADC0 SS2	16
ADC0 SS3	17
ADC1 SS0	24
ADC1 SS1	25
ADC1 SS2	26
ADC1 SS3	27

ADC 通道分配

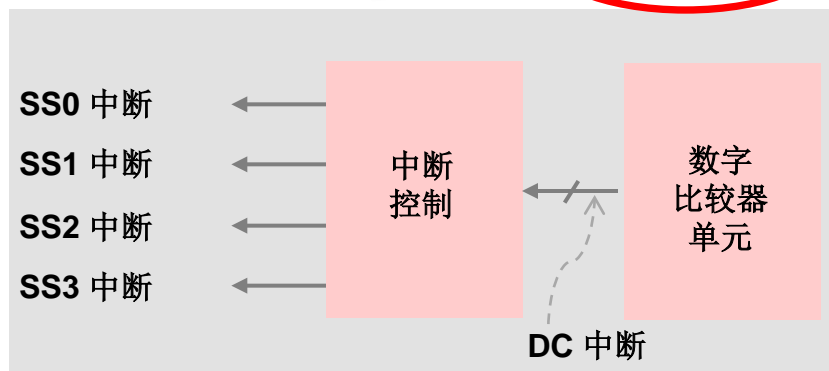
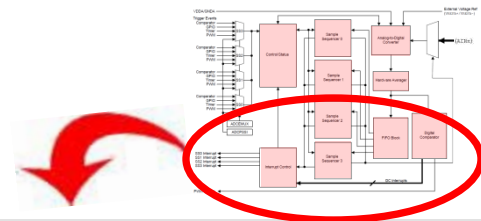
# 中断生成

备用

- 中断可以在下列情况下生成：
  - 样本排序器完成了一个信号的采样；
  - 硬件样本平均电路完成了样本的平均处理；
  - 基于操作模式在数字比较器的某个输入端上承受了一个特定的电压。
- 启用中断：
  - 中断信号受控于 ADCIM 寄存器中的 MASK 位配置。
  - DC 中断功能通过设定 ASDDCCTLn 寄存器中的 CIE 位来启用。
- 监视中断状态
  - 原始中断信号可从 ADCRIS 寄存器读取。
  - 在 ADCIM 寄存器中启动的中断信号可从 ASCISC 寄存器读取。
- 清除中断
  - 样本排序器中断：通过将“1”写入 ADCISC 寄存器中的 Inx 位来清除。
  - 数字比较器中断：通过将“1”写入 ADCDCISC 寄存器来清除。



每个 ADC 的独立中断发生



每个 ADC 的各个 SS 的独立中断