TI MCUs for Motor Control Application

Brian Chang

TI MCU FAE

Agenda

- Introduction of TI MCU for Motor Control
- Introduction of the C2000 for Motor Control
- Single-Shunt Phase Current Reconstruction
- Implementation of Single-Shunt Phase Current Reconstruction with C2000
- Practical Measurement Results
- Summary

27 Years at the Forefront of Motor Control

TMS320 C10 & C25

1985
Hard Disk Drive Servo and Automotive Active Suspension go DSP

F24x 16-bit DSP

1997
First SOC for Inverter & Servo Control starts Digital Motor Control Revolution

LF240x 16-bit DSP

2000
Low Cost + Performance broadens reach in variable speed

F28x 1st 32-bit DSP/DSC

2003
Driving performance FOC applications

ASIC Motor Drive

2005
DVD Spin & Tray Control

Delfino™ Floating Point

2007
Continue performance leadership with ease of use

MSP430™
Piccolo™
Concerto™ ARM
Stellaris® ARM
Hercules™ ARM

2009+
Mass appeal with adoption of smart control due to low cost, high performance MCUs

Now Leveraging motor driver expertise in:

- Power MOSFETs
- Audio
- MCU
- Laser printers
- Hard disk drive
- Mixed signal automotive

TI MCU Portfolio: Silicon to Solutions

MSP430™
16-bit ultra-low-power

Up to 25 MHz
Flash: 1 KB to 256 KB
Analog I/O, ADC, LCD, USB

Measurement
sensing
general purpose

$0.25 - $9.00

Stellaris®
32-bit ARM®

Up to 80 MHz
Flash: 64 KB to 256 KB
USB, ENET, MAC+PHY, CAN, ADC, PWM

Motion control
HMI
industrial automation

$1.00 - $8.00

C2000™
32-bit real-time

40 MHz to 300 MHz
Flash, RAM: 16 KB to 512 KB
PWM, ADC, CAN, SPI, I²C

Motor control
digital power, lighting
renewable energy

$1.00 - $15.00

Hercules™
32-bit ARM® safety

80 MHz to 220 MHz
Flash: 256 KB to 3MB
USB, ENET, FlexRay, ADC, CAN, LIN, SPI, I²C

Transportation
medical
industrial safety

$5.00 - $20.00

IDEs compilers & debuggers
Modular code examples
Code Composer Studio® IDE
Stacks & libraries
GUI-based code gen tools
Development network

Smart Sensors
- Sensor hub
- Cap touch
- Ultra-Low Power proximity Sensing

Smart Grid
- Grid infrastructure
- Utility meters
- Smart homes/ buildings

Embedded RF
- RF connectivity
- Active and passive (RFID)
- SoC / 2 chip solutions
- S/W and stack integration

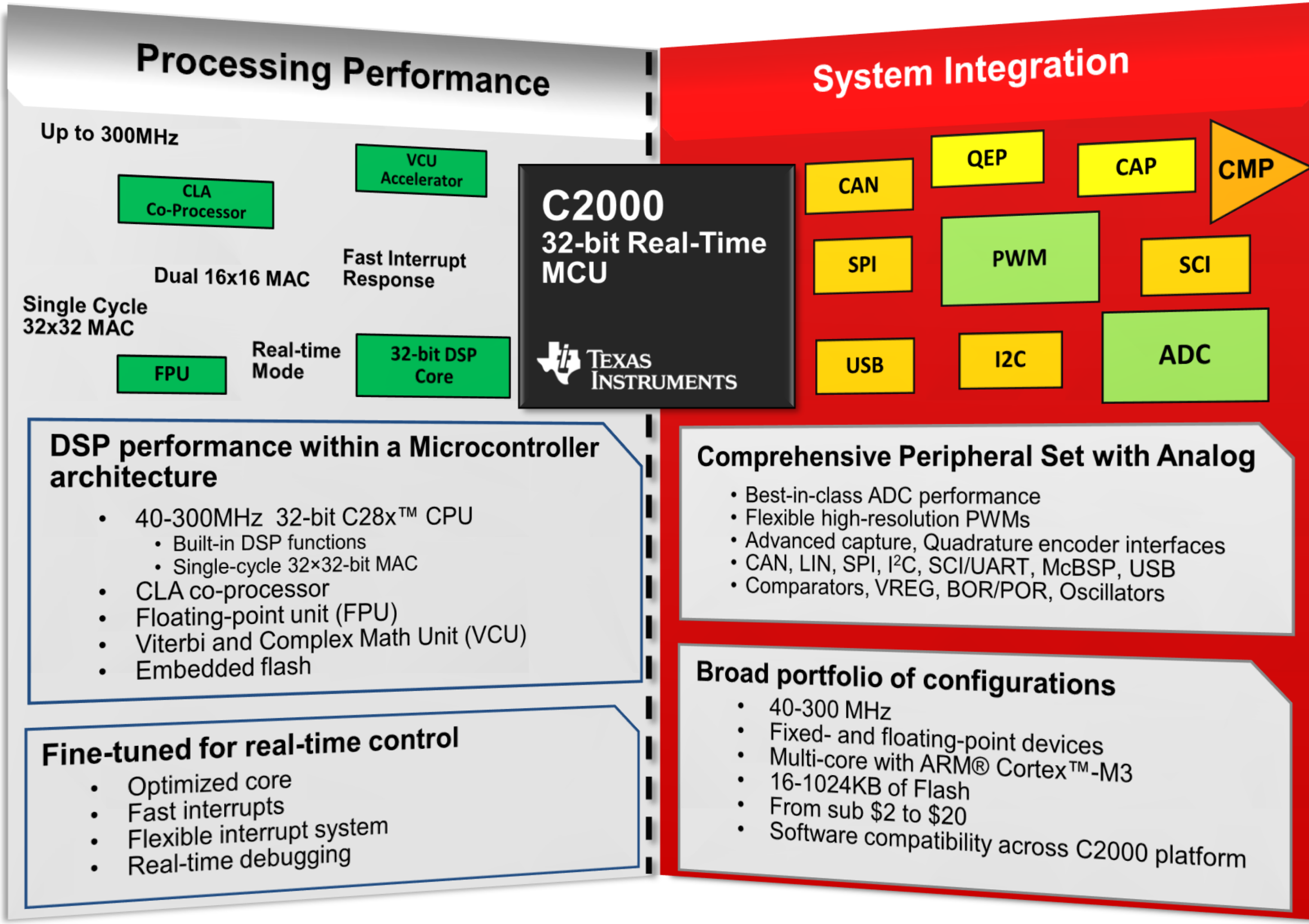
Motor Control
- InstaSPIN solutions
- Premier sensorless three phase
- Automotive
- Safety
- Connectivity

Lighting
- Bulb replacement technology
- Switches, dimmers
- Multi-string LED drivers
- Lighting communication

Safety
- HW Design for safety
- Hercules MSP430, Cortex™-R4 & Cortex-RM, C2000™
- SIL3, ASILD

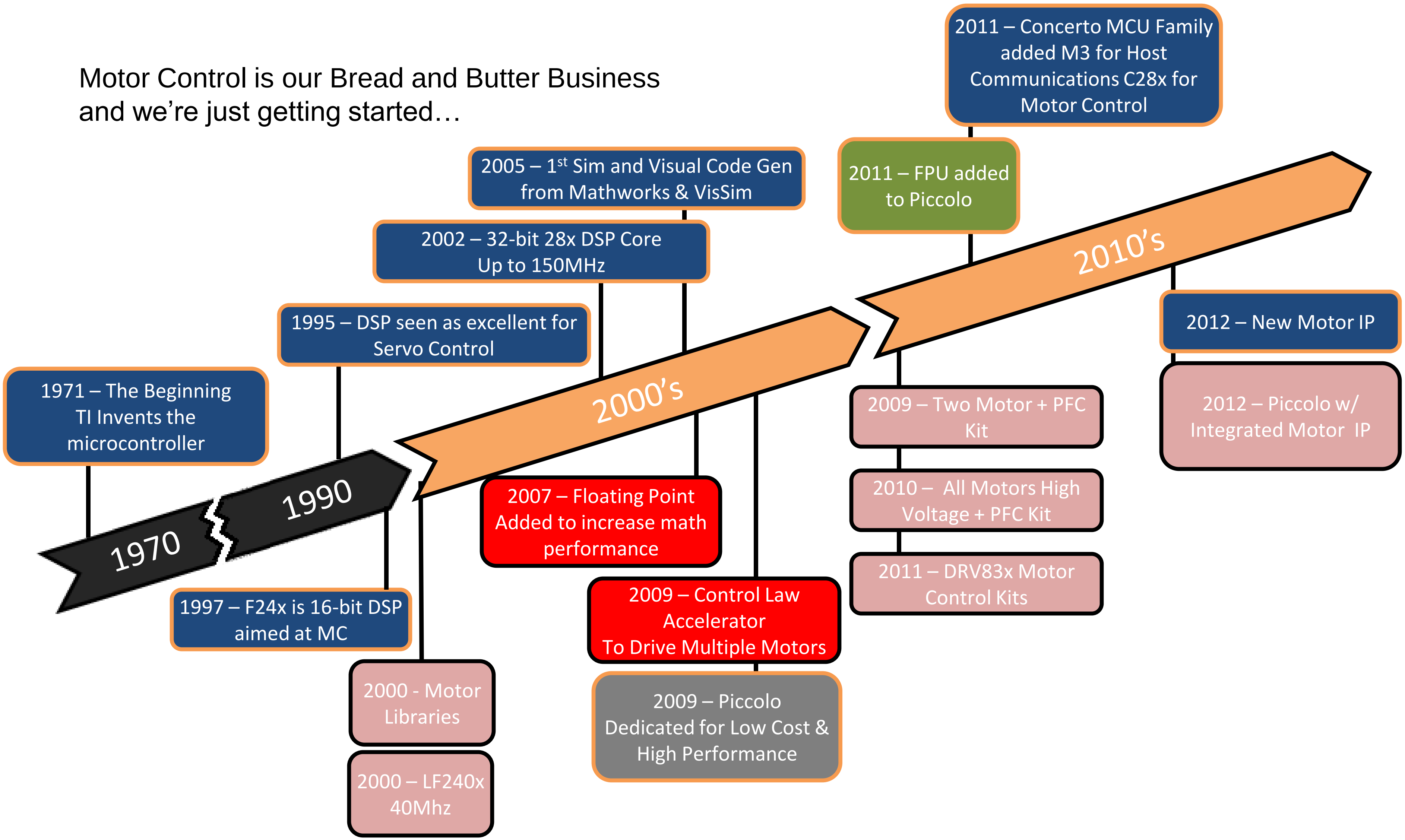
What is C2000™?

The 32-bit real-time microcontroller family

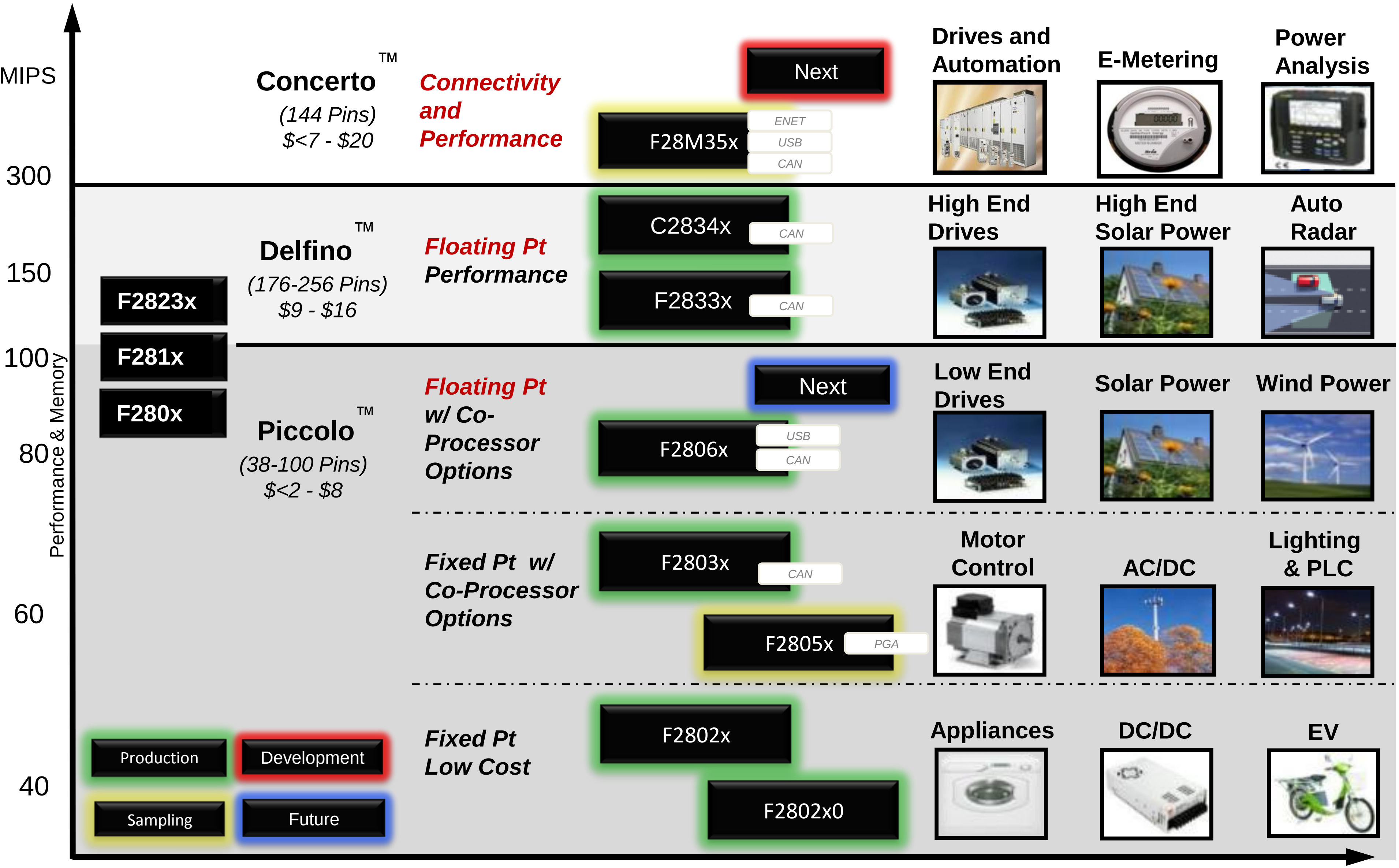


C2000 – Always Differentiating

Motor Control is our Bread and Butter Business
and we're just getting started…



C2000™ 32-bit MCU Roadmap

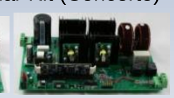


100+ Code Compatible Devices

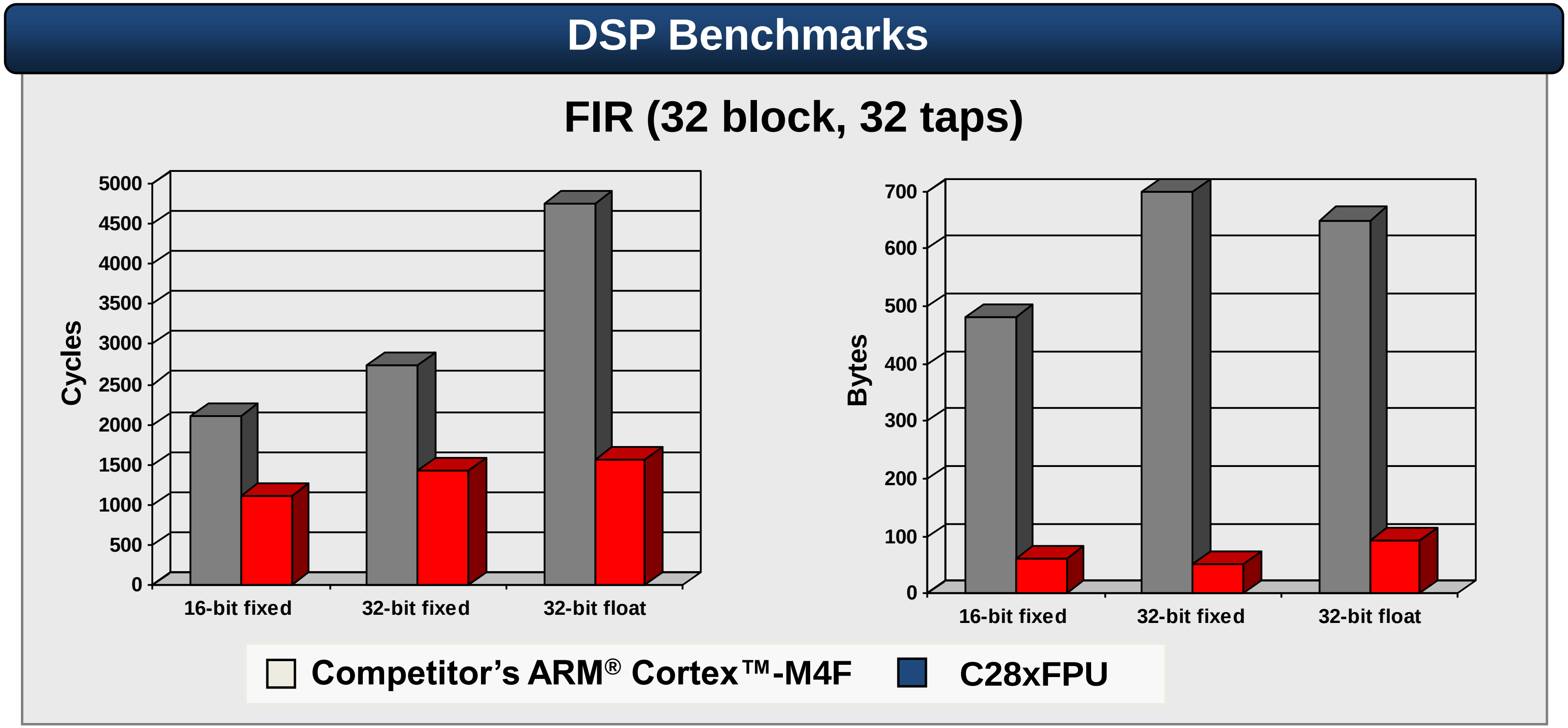
All pricing is to be considered budgetary and subject to change. Pricing is 1KU SRP -40 to 105°C.

C2000: EE Focused

Production | Sampling | Development

	Motor	Power	Solar	Auto	Lighting
					
C2000 Devices	F2802x0 -BLDC (Trapezodial) F2802x -PFC + BLDC Sensorless F2803x, F2805x - PFC + 2 PMSM or - 2ACI Sensorless	F2802x0 - PFC F2803x, F2804x - PFC + Advanced Topologies	F2803x, F2804x - Micro Inverter F2806x, F2807x -1PH Inverter w/ PLC F2837x -3PH Inverter /Multi-level F28M35x - 3PH Inverter /Multi-level + Connectivity	F2802x -Front Light LED F2803x, F2806x -DC/DC -HEV Motor -Charging -PLC F2807x, F2837x -Radar -HEV Motor -PLC F28M35x	F2802x0, F2802x -LED Strings F2803x, F2806x -LED Streetlight w/ PLC
EE Kits	2Q' 12 Motor Control Kit (Piccolo F2805x) & Free Motor Library  2H' 12 InstaSPIN Motion (Piccolo F2805x) 2H' 12 Safety Drive / Inv	1Q' 12 PFC-2phs-IL (revised)  2H' 12 HV 3PH mains PFC 	1Q' 12 HV Solar Kit (Piccolo) 1Q' 12 HV Solar Kit (Concerto)   1Q' 12 LV Solar Kit (Piccolo) 1Q' 12 LV Solar Kit (Concerto)  2H' 12 3 phs / 3 level Inv 2H' 12 Microinverter (Piccolo)	2Q' 12 BiDirectional DC/DC  2H' 12 HEV Charging 	1Q' 12 AC Lighting Kit + PLC (Piccolo) 

C2000 Leads in Performance for DSP algo's



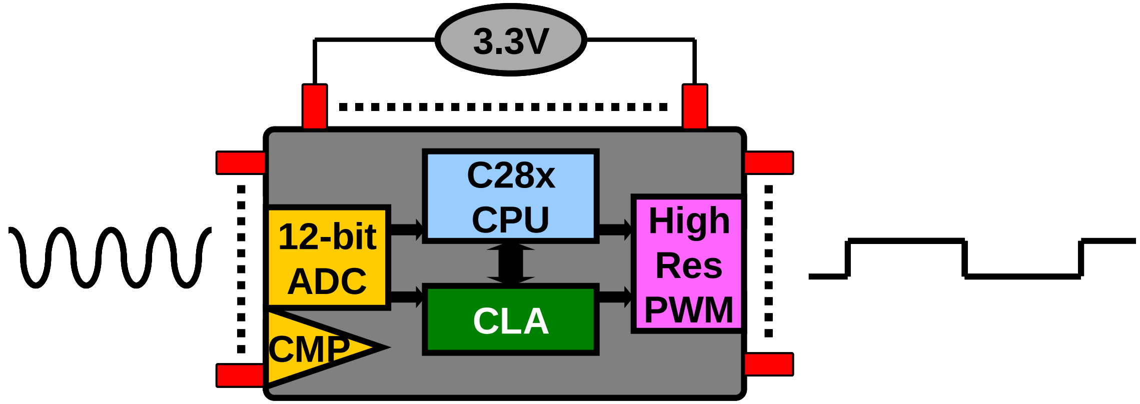
Cycles:

- C2000 is 3x better performance 32-floating pt. FIR
- C2000 is up to 2x better performance 16-bit FIR

Code Density:

- C2000 is >7x better code density 32-floating pt. FIR
- C2000 is 7x better code density 16-floating pt. FIR

What is the Control Law Accelerator (CLA)?



Independent 32-bit floating-point math accelerator

Operates independently of the C28x CPU

- Independent register set, memory bus structure & processing unit
- Low interrupt response time

Direct access to on-chip peripherals

- Execution of algorithms in parallel with the C28x CPU

Fully programmable: IEEE 32-bit floating

- Removes scaling and saturation burden

Reduced Sample-To-Output Delay

Faster System Response & Higher MHz Control Loops

Improved Support For Multi-Channel (Phase/Freq) Loops

→ Digital Power Applications

Improved System Robustness → Automotive, White-goods

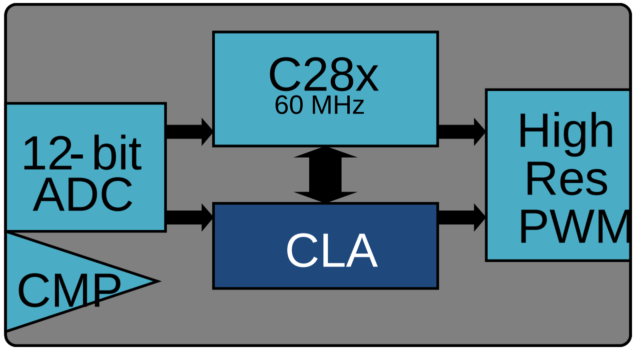
Free-Up C28x CPU For Other Tasks (communication, diagnostics) → General Purpose MCU Applications

Control Law Accelerator

The Control Law Accelerator on Piccolo F2803x devices is an independent math accelerator that can execute algorithms in parallel with the C28x CPU

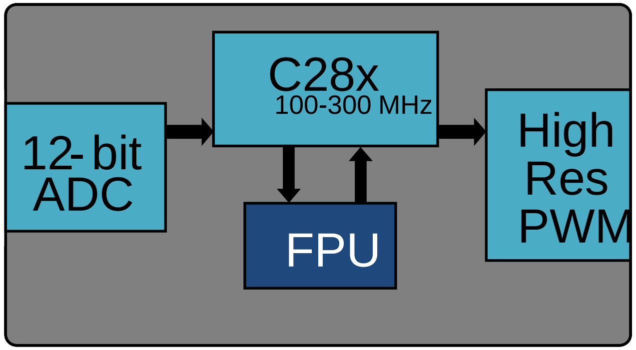
CLA vs. FPU

CLA



- Direct control of Analog and PWMs
- Parallel Processing
- Piccolo 03x: 64-80 pins, $4-$5

FPU



- Higher Clock Frequencies
- C-programmable
- Delfino 33x/34x: 176+pins, $8+

Example Control Loop	C28 (60MHz) Cycles	C28 + CLA (60MHz) Cycles	
2 x PMSM FOC 10 KHz	2400	2400	0
PFC 60 KHz	300	0	250
Total Loop cycles 10 KHz	4200	2400	1500
CPU Load	**70%** **18 MIPS Remain**	**40%** **36 MIPS Remain**	

2x MIPS for rest of system!

CLA enhances control-loop processing with:

- Lowest latency to ADC/PWM
- Better code density
- Higher Cycle Efficiency
- Floating-Point instructions

Piccolo F2805x Series

New

Performance

- 60 MHz 28x CPU
- 60 MHz CLA Floating Point Co-Processor

Embedded Memory

- Up to 128 KB Flash
- Up to 20 KB SRAM
- Boot ROM
- Dual 128-bit Security Key Protected Zones (Flash & ROM)

Flexible Control Peripherals

- 14 enhanced PWM channels (ePWM) with fault mgmt
- 12-bit ADC up to 2.3 MSPS with dual sample and hold
- Up to 4x programmable gain amplifiers (2x, 5x, 10x)
- Up to 7x windowed analog comparators + 10-bit DAC
- 32-bit enhanced input capture module (eCAP)
- Quadrature encoder interface (QEP)

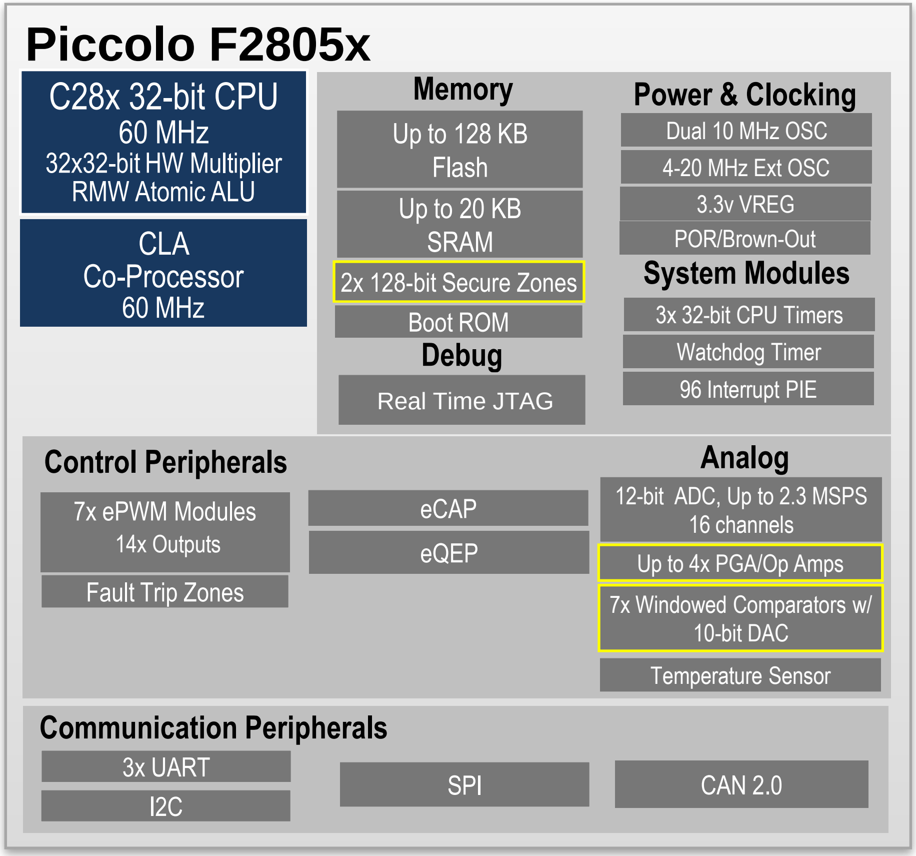
Communication Interfaces

- 3x SCI/UART modules
- SPI module
- I2C bus
- CAN 2.0

Other Features

- Single 3.3V supply with integrated POR/BOR
- Dual on-chip oscillators (10 MHz) with clock fail detect
- 80-pin QFP
- -40 to 105/125^{0} C (Automotive AEC Q-100 Qualified)

Applications: *Motor Control & Drives, White Goods, Digital Power, UPS, Renewable Energy, Power & Protection*



Piccolo F2802x0 Entry Line

Applications: Motor Control (Washing Machines, Pumps, Refrigerators, Compressors, Induction Cooking, A/C)

Performance

- 40-50 MHz 28x CPU

Embedded Memory

- 16-32 KB Flash
- 6-8 KB SRAM
- Boot ROM
- 128-bit Security Key

Flexible Control Peripherals

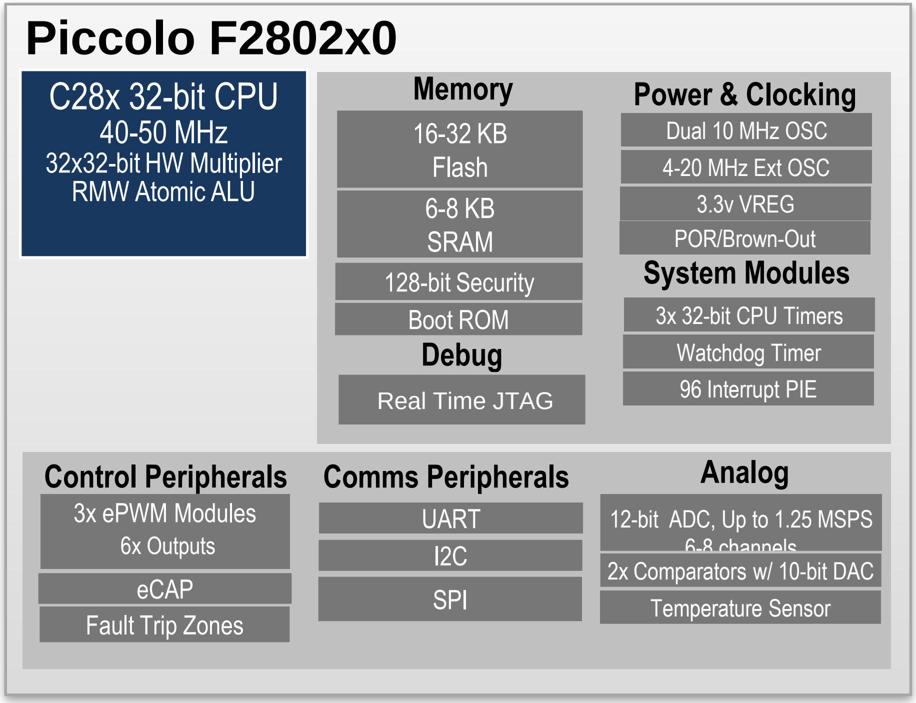
- 6 enhanced PWM channels (ePWM) with fault mgmt
- 12-bit ADC up to 1.25 MSPS with dual sample and hold
- Up to 2x analog comparators + 10-bit DAC with slope compensation
- 32-bit enhanced input capture module (eCAP)

Communication Interfaces

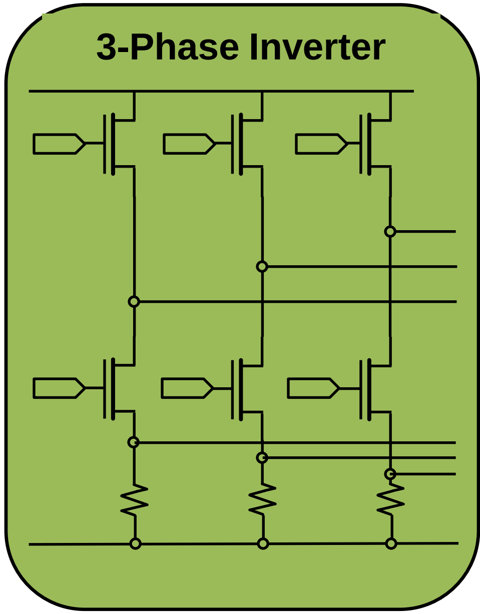
- SCI/UART module
- SPI module
- I2C bus

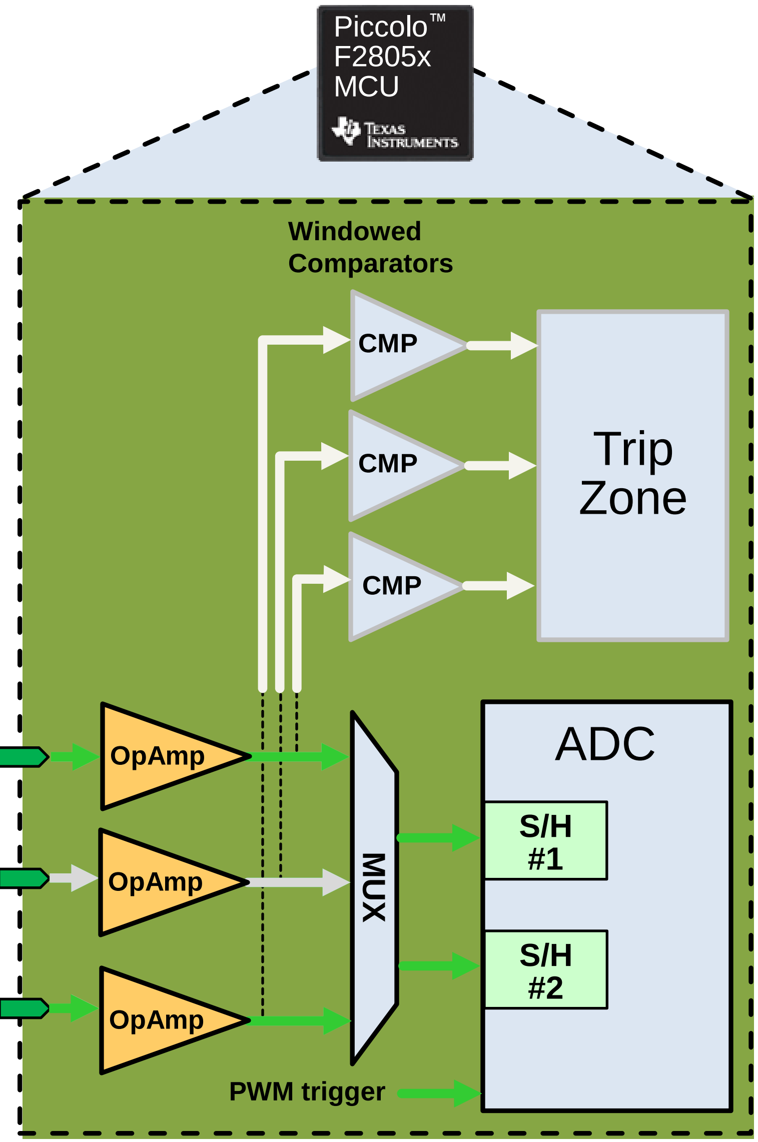
Other Features

- Single 3.3V supply with integrated POR/BOR
- Dual on-chip oscillators (10 MHz) with clock fail detect
- 38-pin TSSOP, 48-pin QFP
- -40 to 105 Temp Range



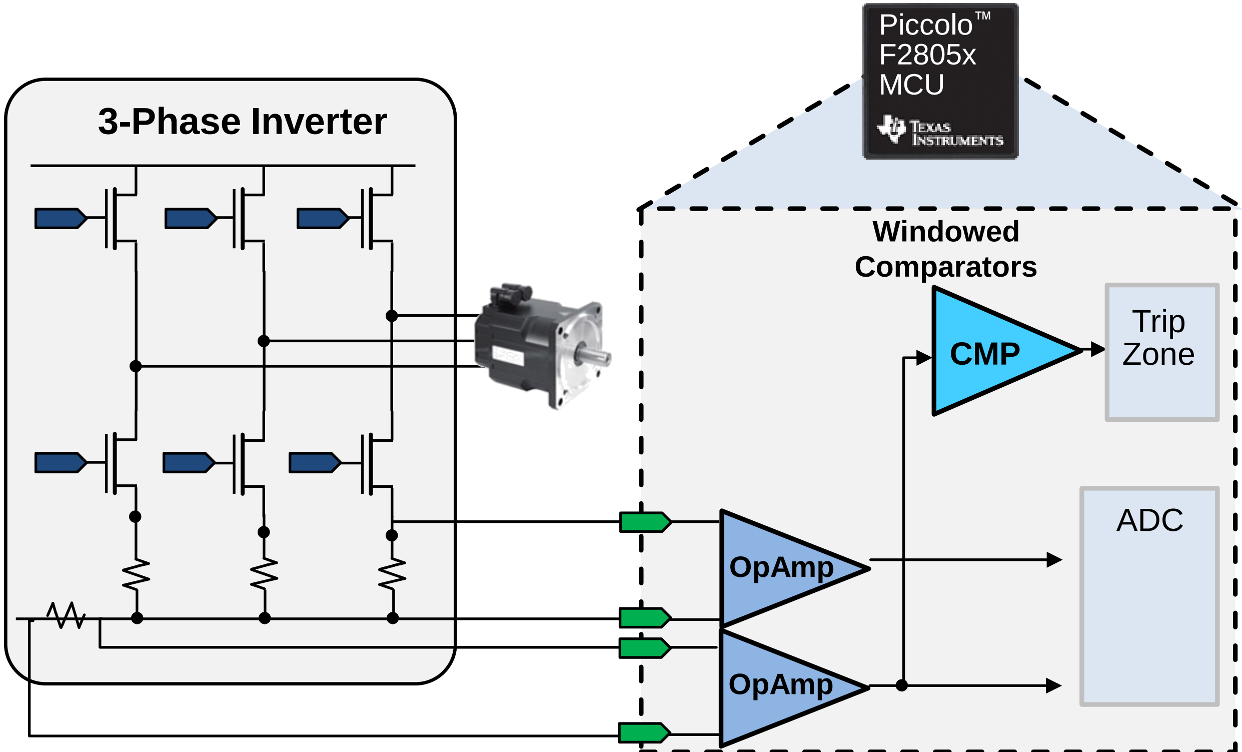
Piccolo 05x ADC designed for motor control

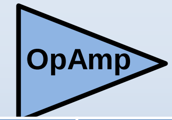




- Synchronous motor phase readings with dual sample & hold ADC
- ADC input mux for greater pin out utilization
- 12-bit, 2.3 MSPS ADC for high speed feedback loops
- Auto-sequencer for auto sampling patterns

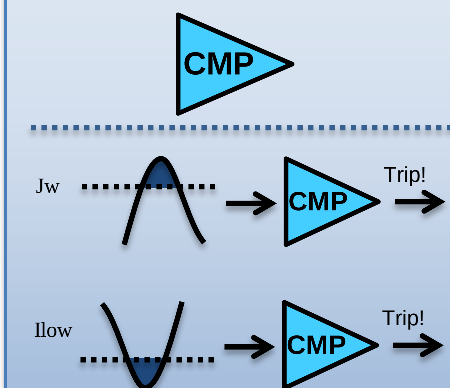
05x Integrated Op-Amps & windowed comparators





	F2805x	OPA376	Kinetis-K60
Gain Settings	2, 5, 10	External Resistors	1, 2, 4, 8, 16, 32, 64
Input Offset	0.2mV	5uV	0.2mV
Gain Accuracy	0.1%	0.001%	7%
Slew Rate	30V/us	2V/us	N/A

Windowed comparators



- **Over-current AND under-current fault protection**

- Individual current feedback for each motor phase
- Integrated fault protection for system robustness and less pin utilization
- High-performance, programmable OpAmps for accurate and on-time system feedback

FOC Introduction

Demands on motor control

- Smooth rotation over entire speed range
- Full torque control at zero speed
- Fast acceleration and deceleration

⇒ **Field Oriented Control (Vector Control)**

Field Oriented Control

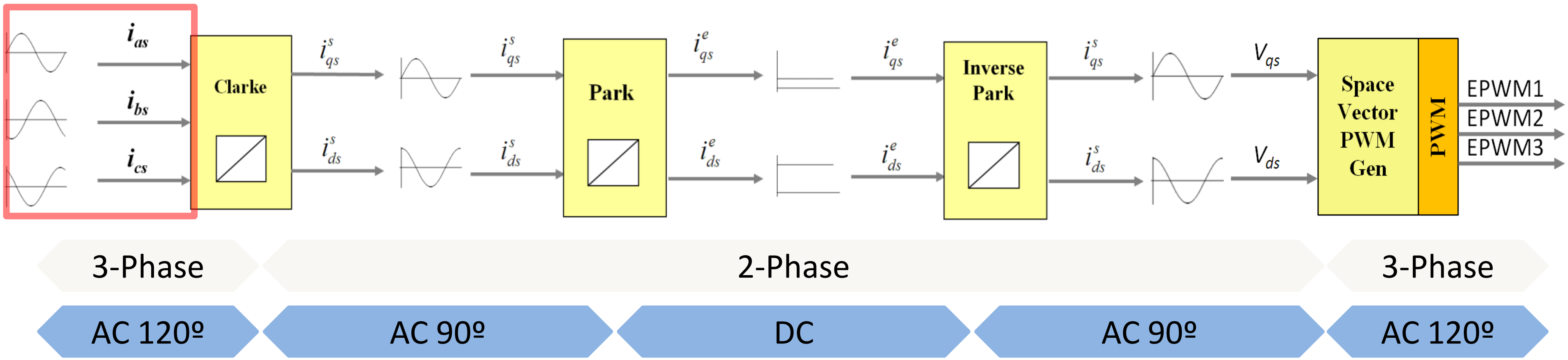
- Decomposition of the three-phase stator currents into
  - magnetic field-generating part ► i^e_{ds}
  - torque-generating part ► i^e_{qs}

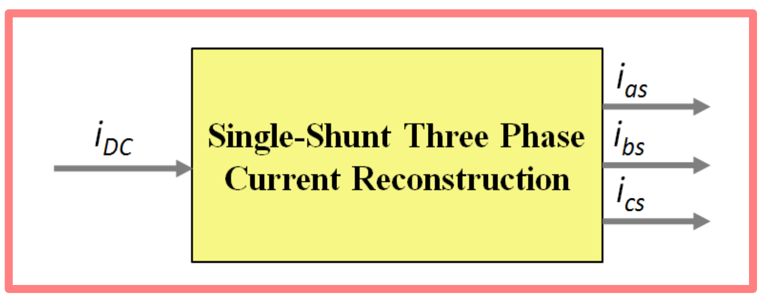
Rotor flux position is estimated using motor quantities

- Measurement of phase voltages
- Measurement of phase currents (three/two/single – shunt)

Introduction

Processing of motor quantities [1]

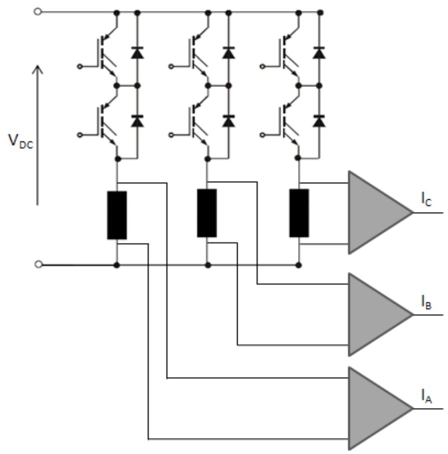




Single-Shunt Phase Current Reconstruction

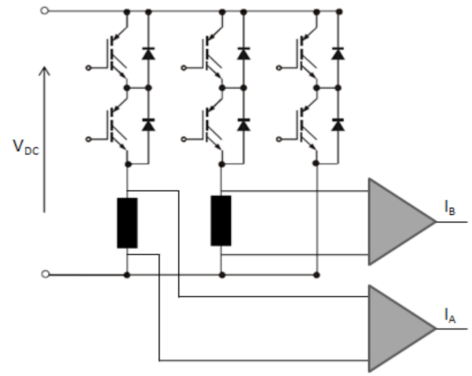
Low side shunt current measurement

Three Shunts



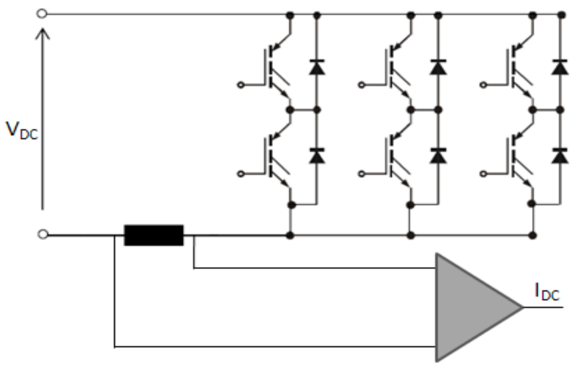
3 phase currents measured

Two Shunts



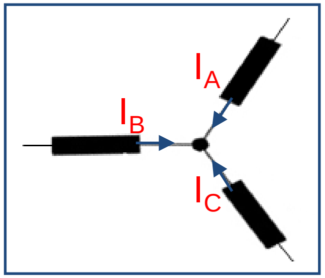
2 phase currents measured
1 phase current calculated

Single-Shunt



DC-Link current measured
3 phase currents calculated

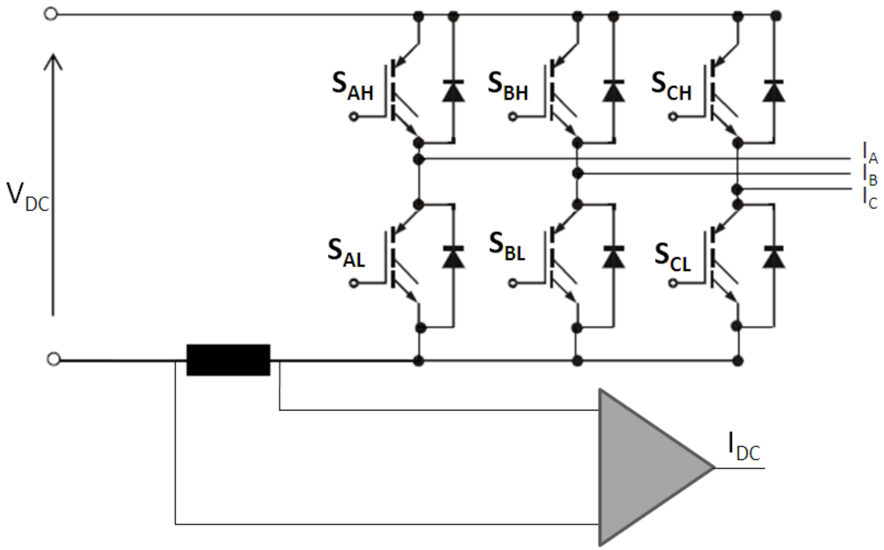
Motor as balanced load with star connection

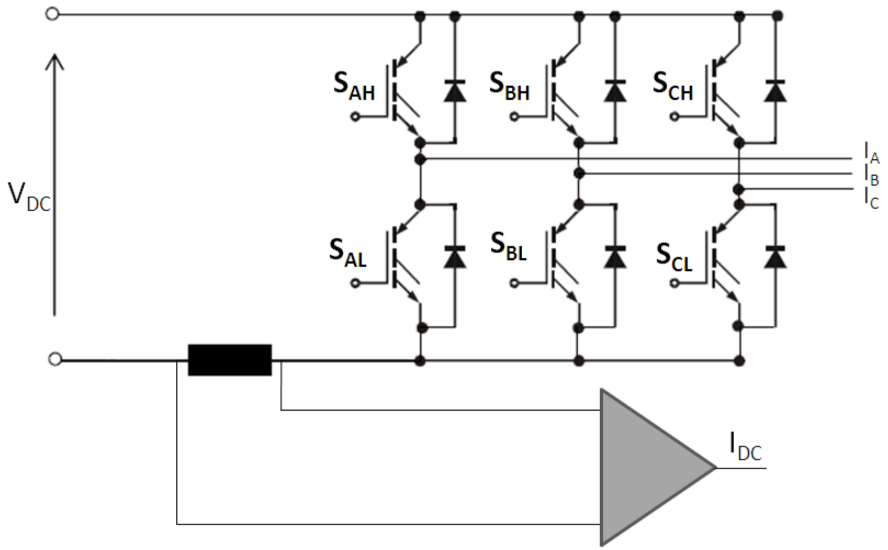


Kirchhoff's Current Law:
$I_A + I_B + I_C = 0$

Single-Shunt Phase Current Reconstruction

Current determination in DC-Link line (two examples)



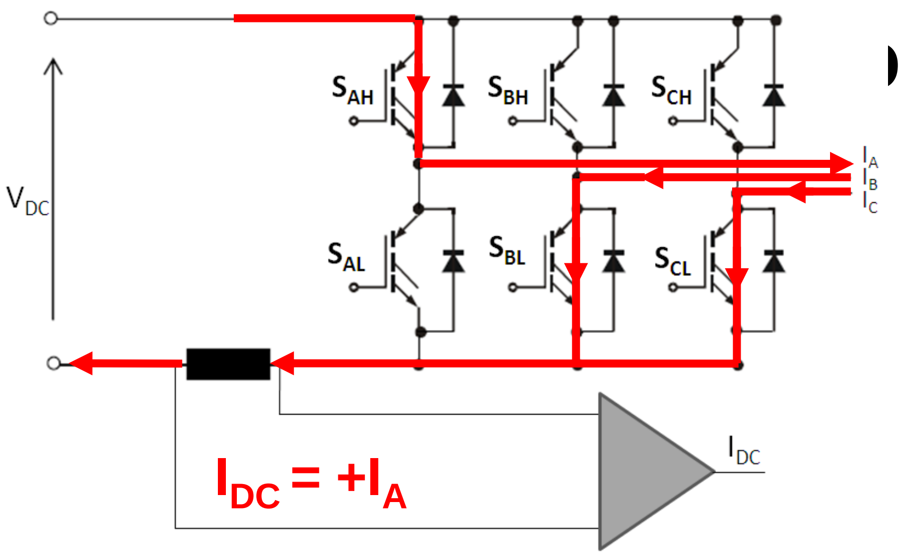


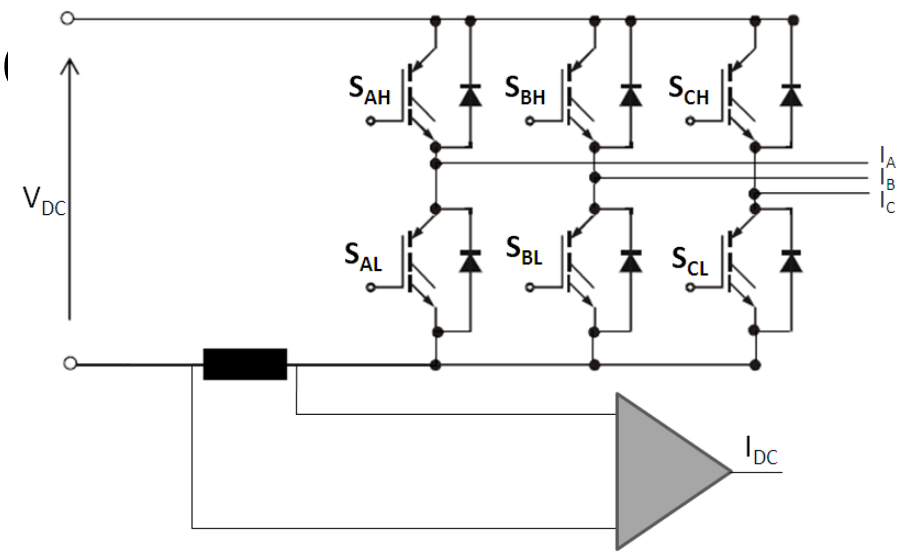
top switches state: 100
bottom switches state: 011

top switches state: 110
bottom switches state: 001

Complementary PWM signals in each leg

Single-Shunt Phase Current Reconstruction



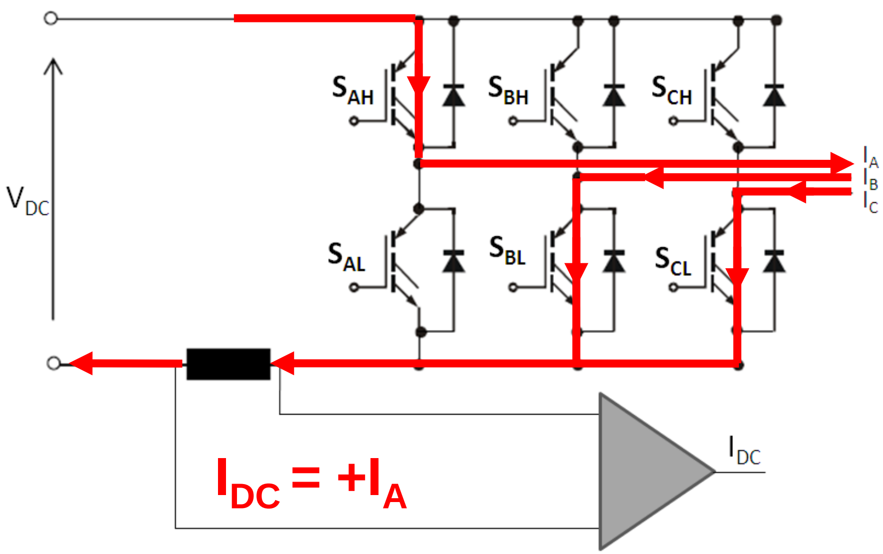


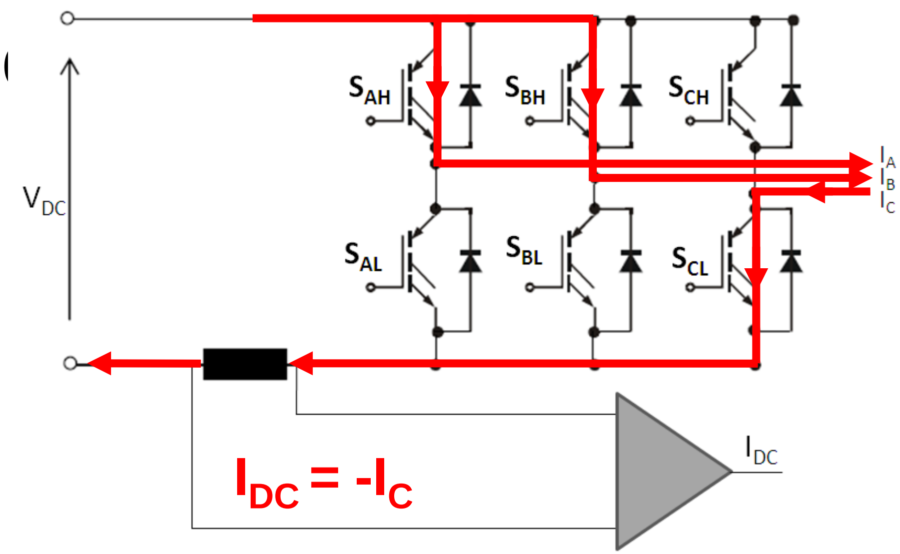
top switches state: 100
bottom switches state: 011

top switches state: 110
bottom switches state: 001

Complementary PWM signals in each leg

Single-Shunt Phase Current Reconstruction





top switches state: 100
bottom switches state: 011

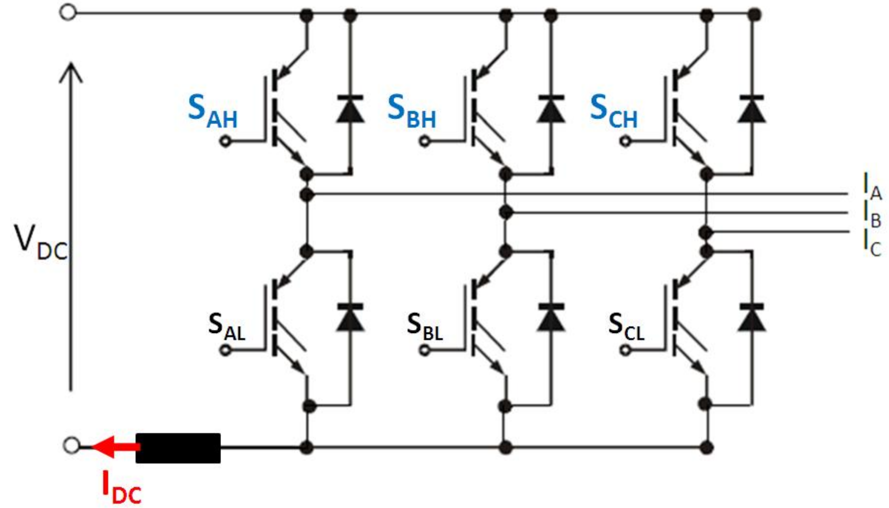
top switches state: 110
bottom switches state: 001

Complementary PWM signals in each leg

Single-Shunt Phase Current Reconstruction

DC-Link current table

S_{AH}	S_{BH}	S_{CH}	I_{DC}
C	NC	NC	$+I_A$
NC	C	NC	$+I_B$
NC	NC	C	$+I_C$
NC	C	C	$-I_A$
C	NC	C	$-I_B$
C	C	NC	$-I_C$

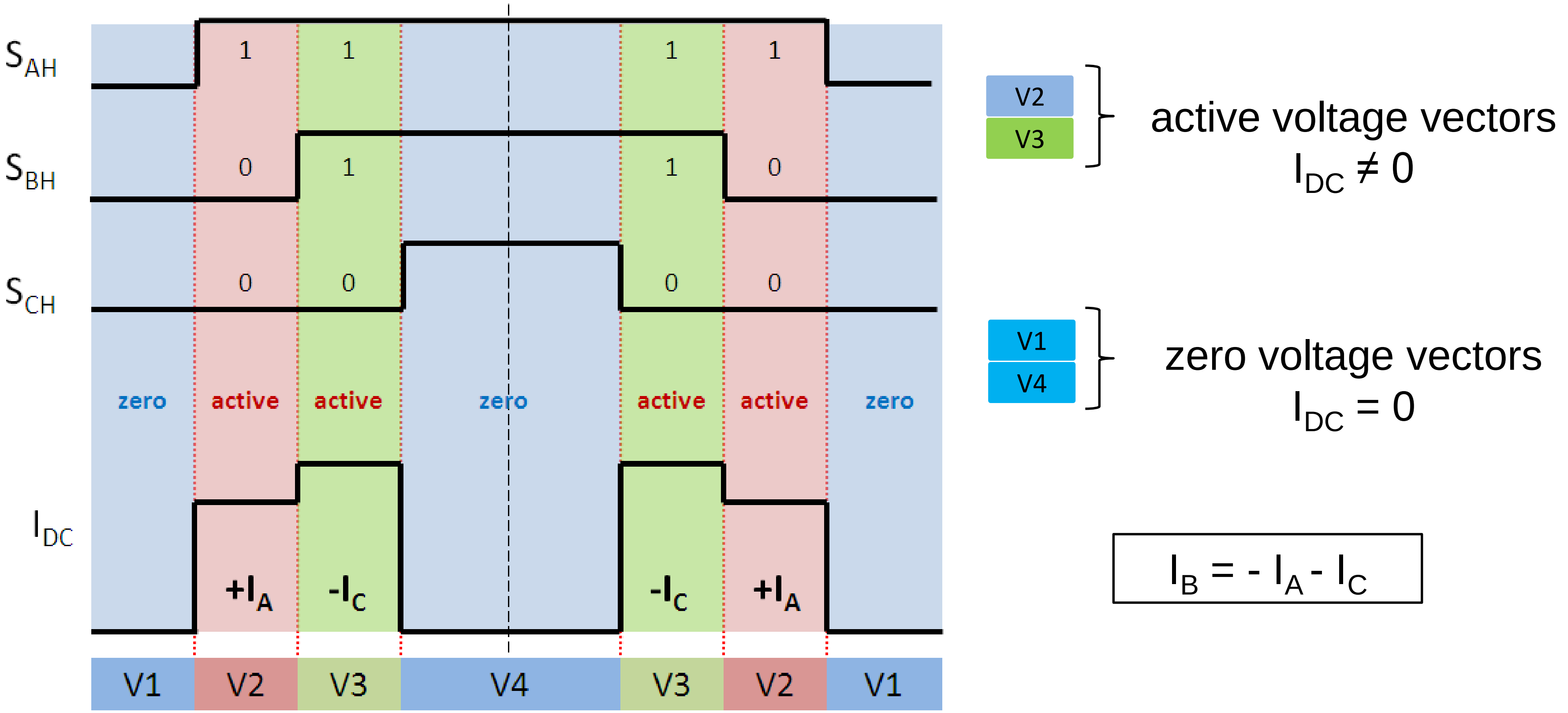


C: **C**onducting Transistor (1)

NC: **N**on-**C**onducting Transistor (0)

Single-Shunt Phase Current Reconstruction

Relation between PWM signals and DC-Link current

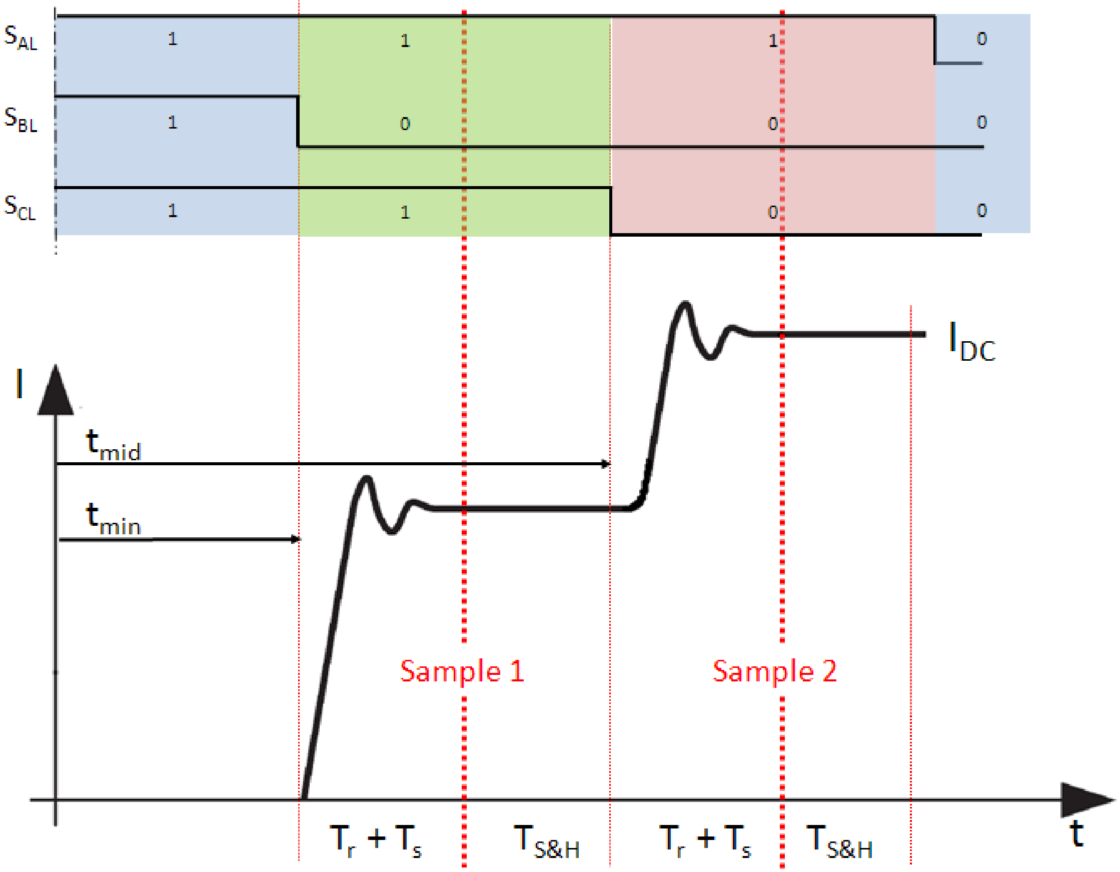


$I_B = - I_A - I_C$

DC-Link current measurements must be done during time interval of active voltage vectors
At least one sample per V2 and V3 => two different currents measured: $+I_A$, $-I_C$

Single-Shunt Phase Current Reconstruction

Determine sampling points for DC-Link current measurements



1. Sample point:

$$t_s = t_{min} + T_r + T_s$$

2. Sample point:

$$t_s = t_{mid} + T_r + T_s$$

with:

T_r = Rise Time
T_s = Settling Time
$T_{S\&H}$ = Sample & Hold Time

Minimum active voltage duration:

$$t_{av\_dur} = T_r + T_s + T_{S\&H}$$

Active voltage duration must be large enough for valid current measurements

Single-Shunt Phase Current Reconstruction

Challenges (explained in detail in next section)

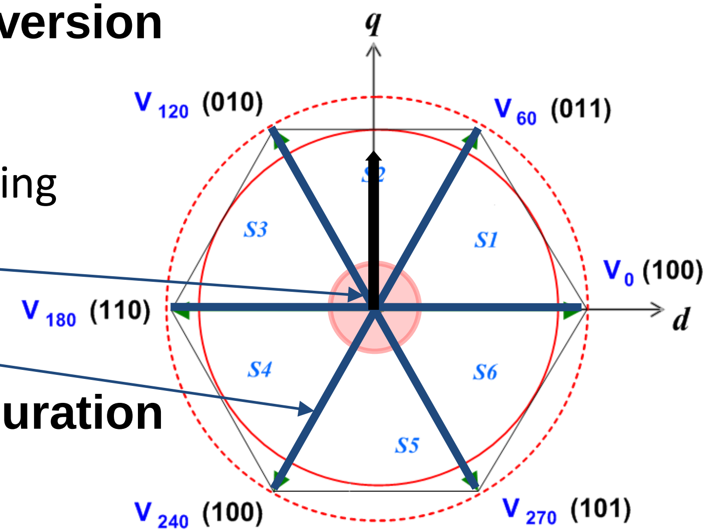
- Sampling point determination (already shown)

⇨ **Synchronization of PWM and ADC conversion**

- Make valid DC-Link current measurements during
  - Low modulation indices operation
  - Space vector sector change

⇨ **Maintain large enough active voltage duration**

1. **PWM Duty Cycle Compensation**
2. **PWM Phase Shift Compensation**

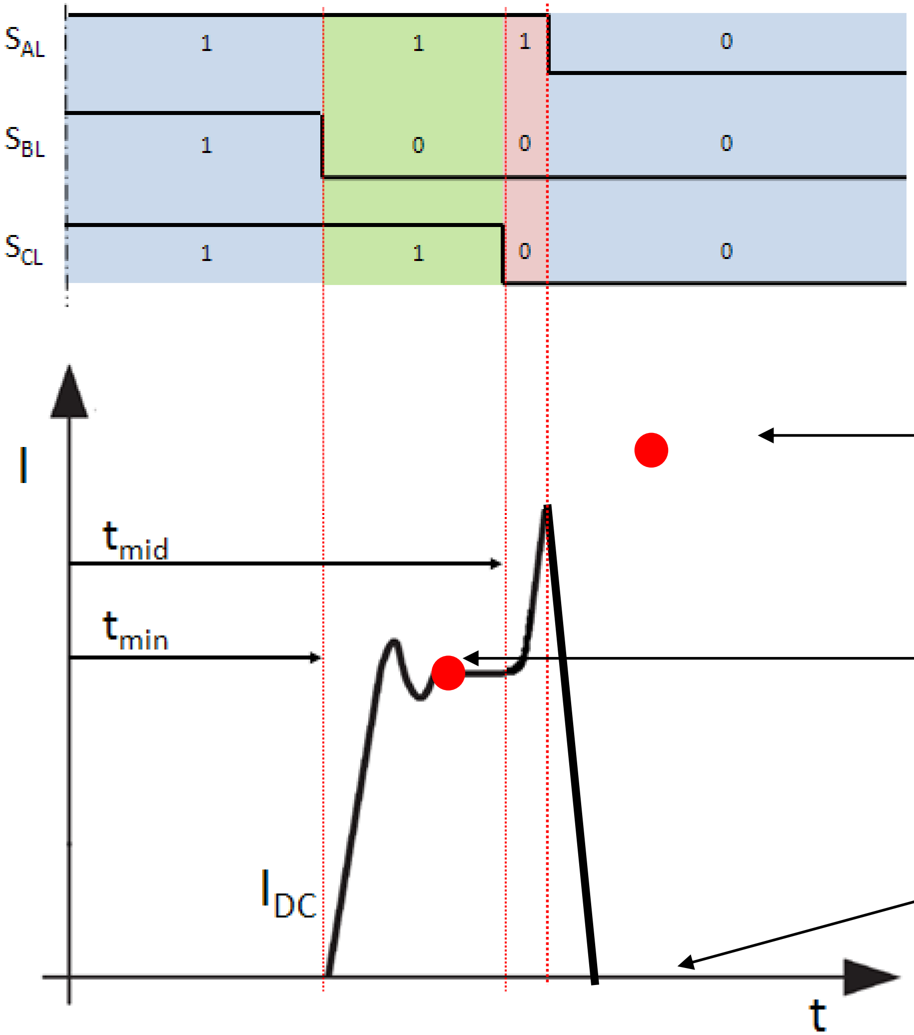


- Compensation of ripple in reconstructed phase currents

⇨ **Phase current ripple compensation**

Single-Shunt Phase Current Reconstruction

Determine sampling points for DC-Link current measurements



Deadband between complementery transistors in inverter stage makes the active voltage durations even smaller

I_{DC} if active voltage duration is large enough

valid current measurement; active voltage duration large enough

invalid current measurement; active voltage duration too small

Implementation of Current Reconstruction

Basic implementation without PWM compensation

- Large enough active voltage duration is not maintained
  - Current measurements during too small active voltage durations are not valid
- Wide variances in reconstructed phase currents as result

Worst current reconstruction performance

Advanced implementation with PWM compensation

- Large enough active voltage duration is maintained in every cycle

1. PWM Duty Cycle Compensation

Better current reconstruction performance

2. PWM Phase Shift Compensation

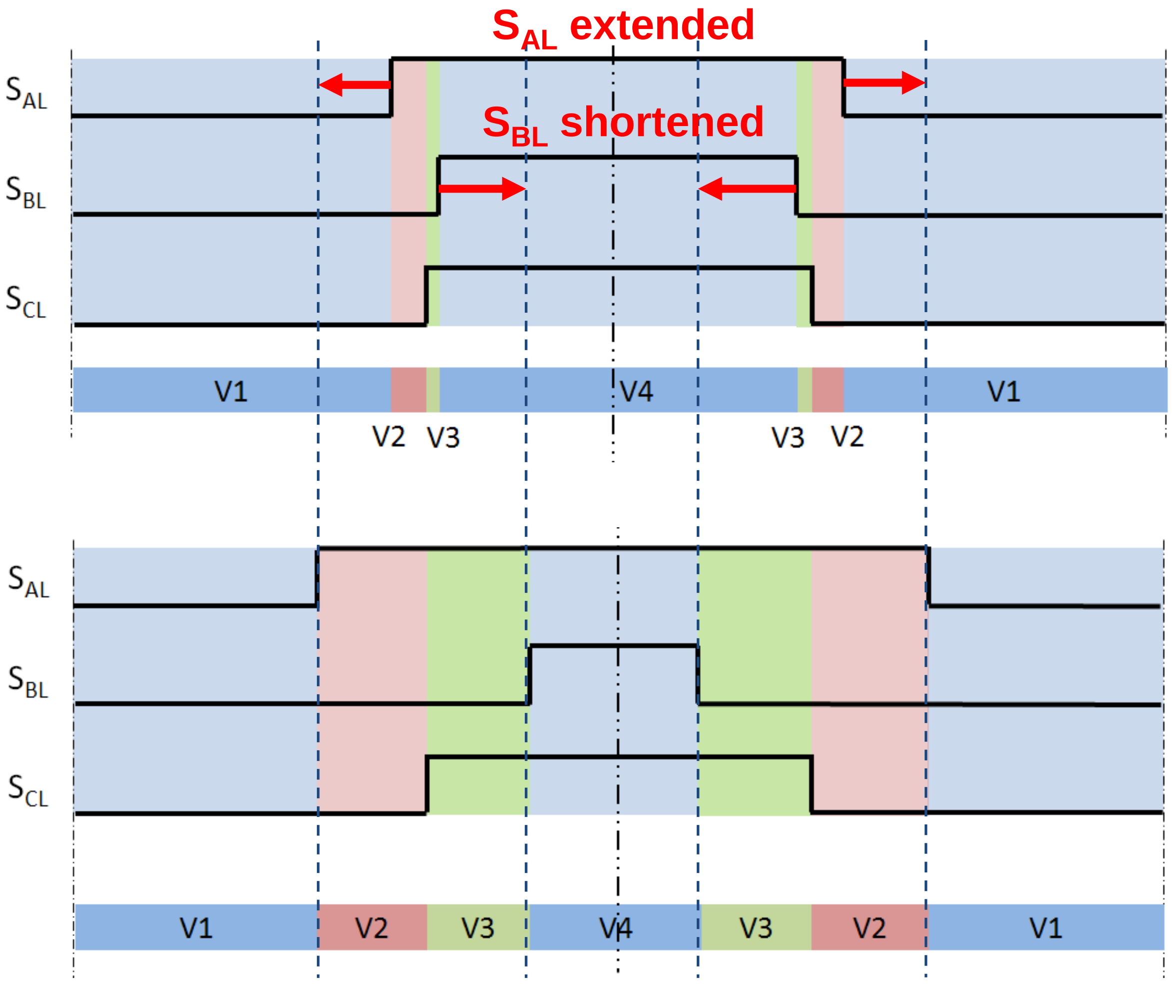
Best current reconstruction performance

Phase current ripple compensation

- Can be applied to basic or advanced implementation type
  => improves current reconstruction performance

Implementation of Current Reconstruction

PWM Duty Cycle Compensation (simple)[2]



PWM with **maximum duty cycle** is **extended**.

PWM with **minimum duty cycle** is **shortened**.

PWM with **midrange duty cycle** remains **unchanged**.

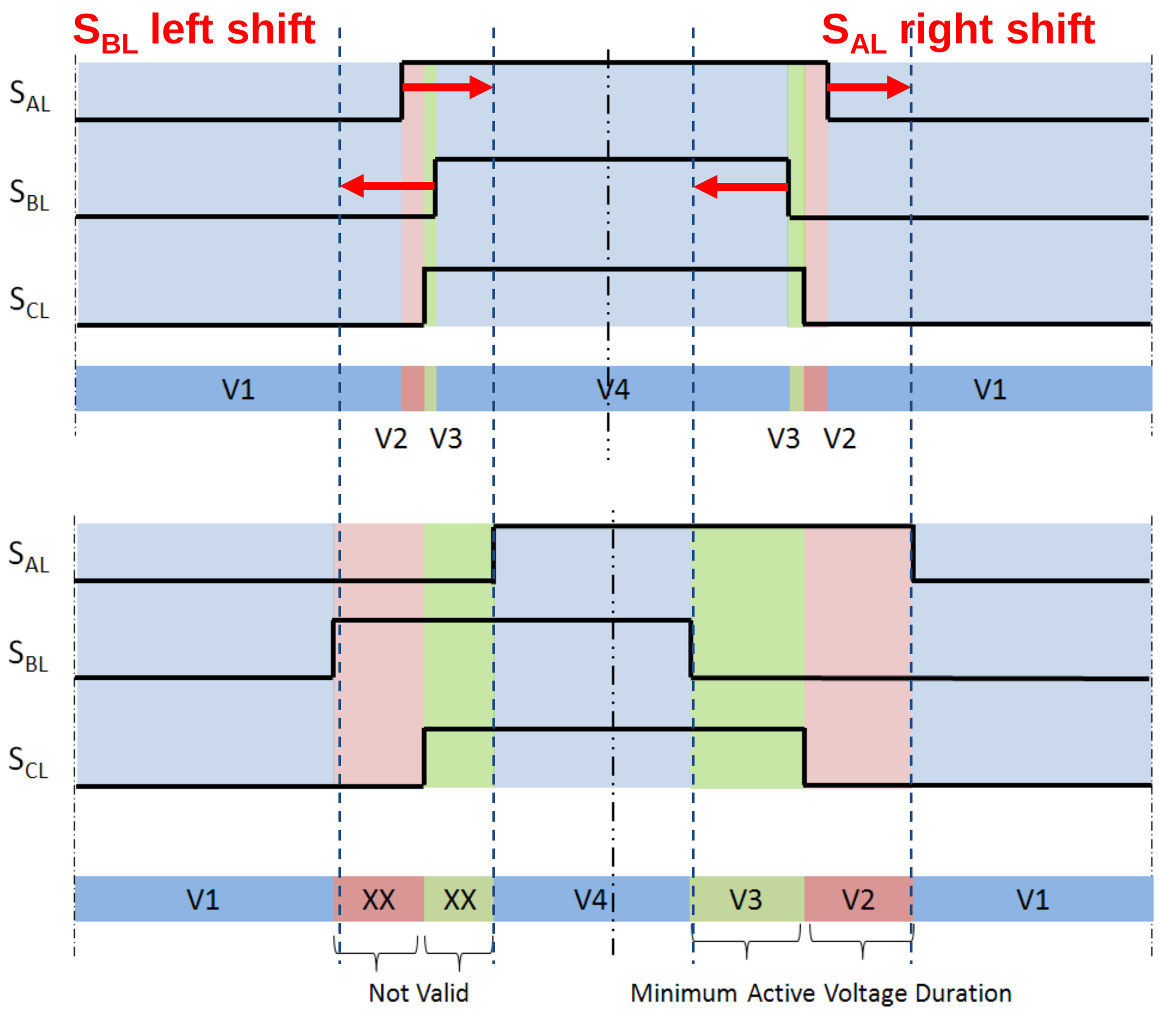
V2 and V3 become large enough active voltage durations.

<u>Advantage:</u>
Only one PWM CMP value necessary

<u>Disadvantage</u>:
Distortion of phase voltages.
S_{AL} larger than original
S_{BL} smaller than original

Implementation of Current Reconstruction

PWM Phase Shift Compensation (more complex)



PWM with **maximum duty cycle** is **right shifted**.

PWM with **minimum duty cycle** is **left shifted**.

PWM with **midrange duty cycle** remains **unchanged**.

V2 and V3 in second half of PWM period become large enough active voltage durations.

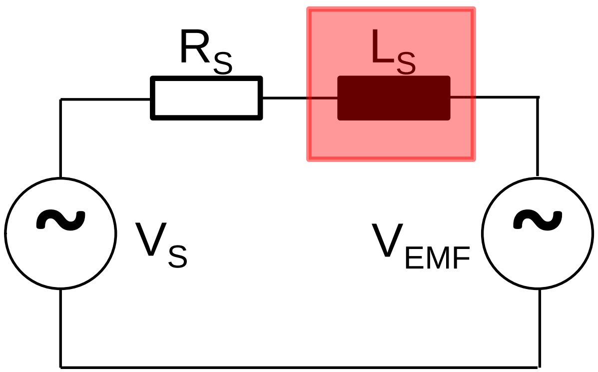
<u>Advantage:</u>
No distortion of phase voltages.
PWM duty cycles remain unchanged.

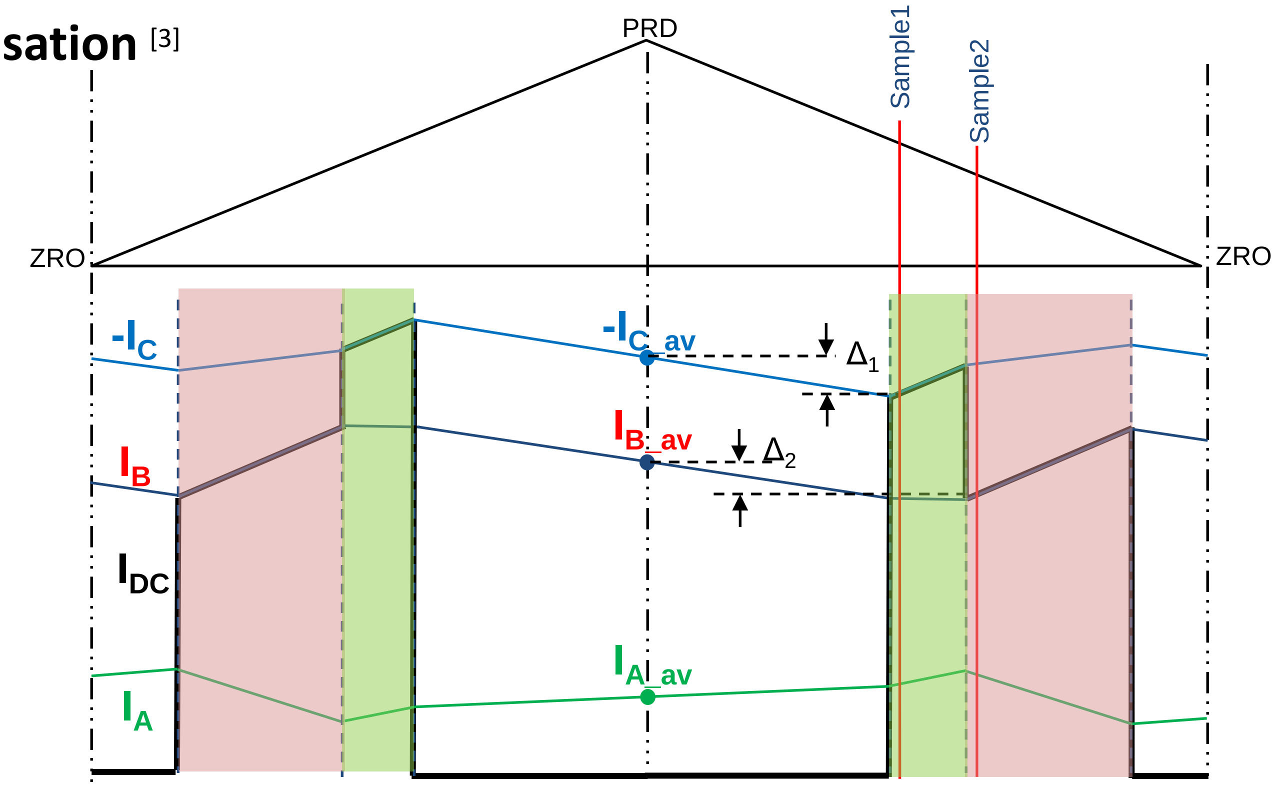
<u>Disadvantage:</u>
Two PWM CMP values used.

Implementation of Current Reconstruction

Current Ripple Compensation [3]

PMSM motor model:



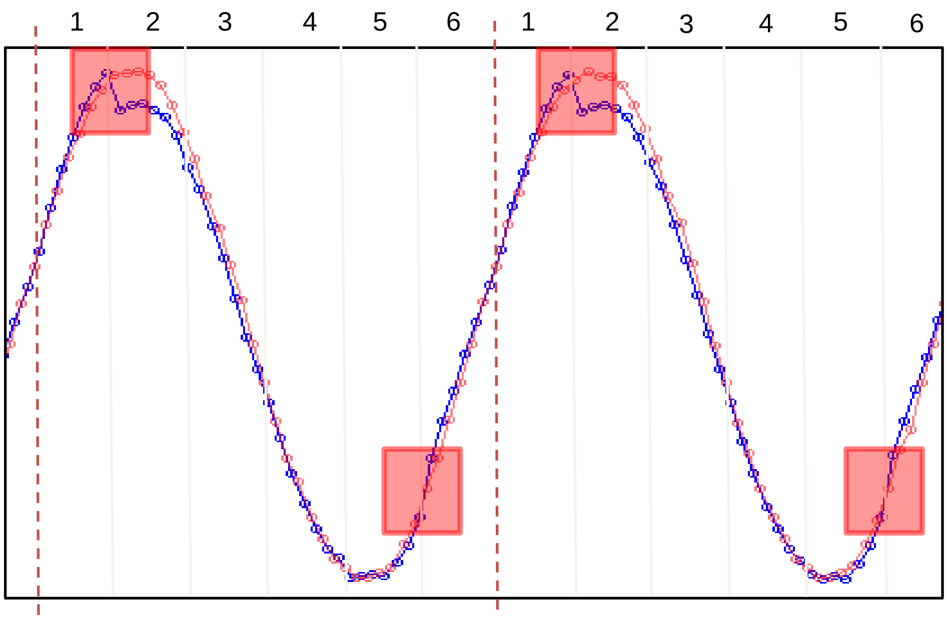


Measurement error of Sample1 and Sample2

Sample1: $-I_C = -I_{C\_av} - \Delta_1$
Sample2: $I_B = I_{B\_av} - \Delta_2$

Implementation of Current Reconstruction

Current Ripple Compensation



I_{A_rec} = reconstructed current I_A

I_{A_av} = directly measured current I_A

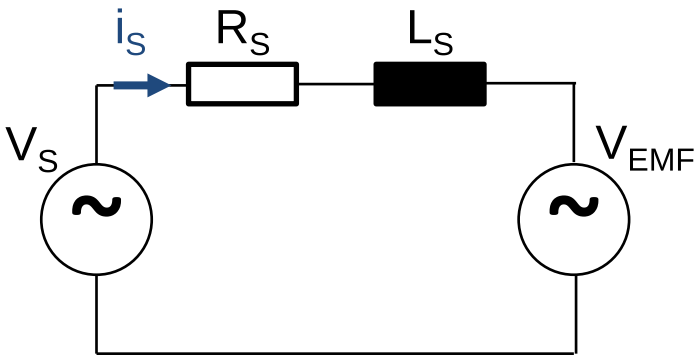
Sector dependent offfset-like current waveform error

Sector	Sector Part			Transition Offset
	Beginning	Middle	End	
1	$I_{A\_av} - \Delta$	I_{A_av}	$I_{A\_av} + \Delta$	-3Δ
2	$I_{A\_av} - 2\Delta$	$I_{A\_av} - \Delta$	I_{A_av}	Δ
3	$I_{A\_av} + \Delta$	$I_{A\_av} + \Delta$	$I_{A\_av} + \Delta$	0
4	$I_{A\_av} + \Delta$	$I_{A\_av} + \Delta$	$I_{A\_av} + \Delta$	$-\Delta$
5	I_{A_av}	$I_{A\_av} - \Delta$	$I_{A\_av} - 2\Delta$	3Δ
6	$I_{A\_av} + \Delta$	I_{A_av}	$I_{A\_av} - \Delta$	0

Implementation of Current Reconstruction

Current Ripple Compensation

PMSM motor model:



$$V_S = R_S \cdot i_S + L_S \cdot \frac{di_S}{dt} + V_{EMF}$$

$$\frac{di_S}{dt} = -\frac{R_S}{L_S} \cdot i_S + \frac{1}{L_S}(V_S - V_{EMF})$$

with:

V_S = measured motor quantity (DC-bus voltage) – phase voltages
V_{EMF} = estimated in SMO (sliding mode observer) module
i_S = measured motor quantity (DC-link current) – phase currents
R_S = motor stator resistance
L_S = motor stator inductance

Implementation of Current Reconstruction

Current Ripple Compensation

PMSM motor model in stationary reference frame:

$$\frac{dI_\alpha}{dt} = -\frac{Rs}{Ls} \cdot I_\alpha + \frac{1}{Ls} \cdot (V_\alpha - E_\alpha)$$

$$\frac{dI_\beta}{dt} = -\frac{Rs}{Ls} \cdot I_\beta + \frac{1}{Ls} \cdot (V_\beta - E_\beta)$$

Current change of rate in original phase domain:

$$\begin{bmatrix} dI_{a0,1}/dt \\ dI_{b0,1}/dt \\ dI_{c0,1}/dt \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{bmatrix} \cdot \begin{bmatrix} dI_{\alpha0,1}/dt \\ dI_{\beta0,1}/dt \end{bmatrix}$$

Offset correction values:

$$\begin{bmatrix} \Delta I_{a1} \\ \Delta I_{b1} \\ \Delta I_{c1} \end{bmatrix} = \begin{bmatrix} dI_{a0}/dt \\ dI_{b0}/dt \\ dI_{c0}/dt \end{bmatrix} \cdot t_{min}$$

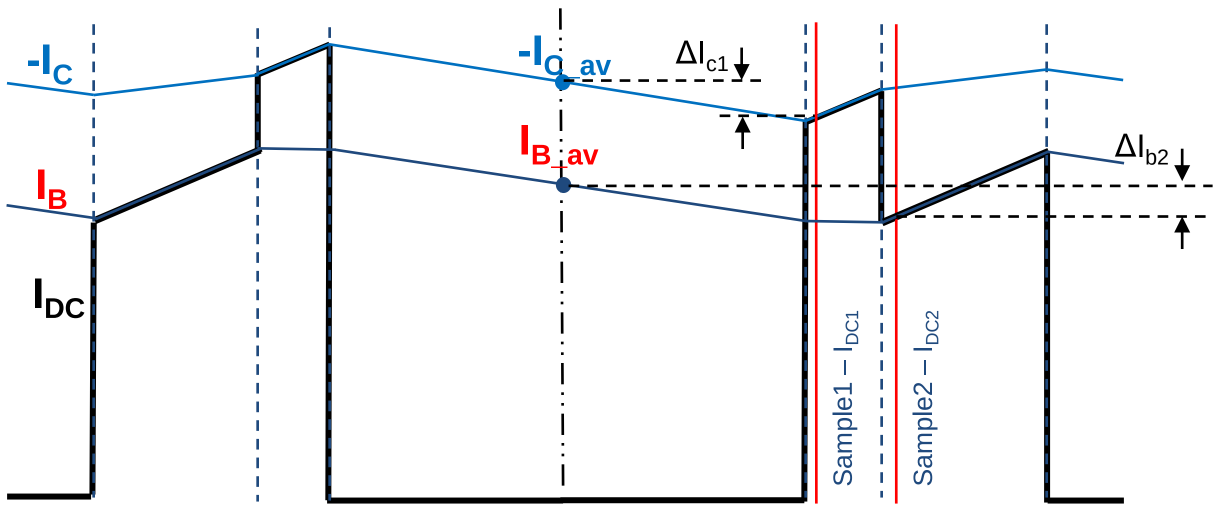
$$\begin{bmatrix} \Delta I_{a2} \\ \Delta I_{b2} \\ \Delta I_{c2} \end{bmatrix} = \begin{bmatrix} \Delta I_{a1} \\ \Delta I_{b1} \\ \Delta I_{c1} \end{bmatrix} + \begin{bmatrix} dI_{a1}/dt \\ dI_{b1}/dt \\ dI_{c1}/dt \end{bmatrix} \cdot (t_{mid} - t_{min})$$

Implementation of Current Reconstruction

Current Ripple Compensation

Sector	I_A	I_B	I_C
1	$I_{DC2}-\Delta I_{a2}$	$-(I_A+I_C)$	$-I_{DC1}-\Delta I_{c1}$
2	$-(I_B+I_C)$	$I_{DC2}-\Delta I_{b2}$	$-I_{DC1}-\Delta I_{c1}$
3	$-I_{DC1}-\Delta I_{a1}$	$I_{DC2}-\Delta I_{b2}$	$-(I_A+I_B)$
4	$-I_{DC1}-\Delta I_{a1}$	$-(I_A+I_C)$	$I_{DC2}-\Delta I_{c2}$
5	$-(I_B+I_C)$	$-I_{DC1}-\Delta I_{b1}$	$I_{DC2}-\Delta I_{c2}$
6	$I_{DC2}-\Delta I_{a2}$	$-I_{DC1}-\Delta I_{b1}$	$-(I_A+I_B)$

Example:

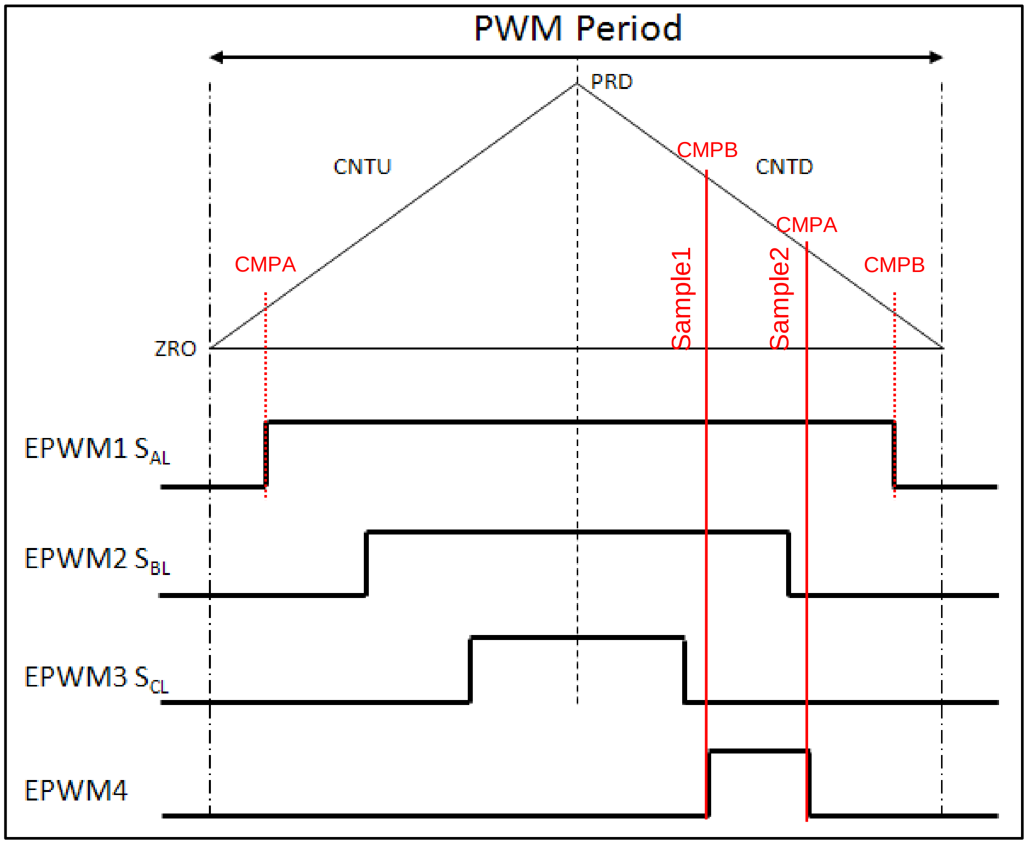


Software Implementation

Implemented solution makes use of the following resources

- **5 A/D channels** (1x dummy, 2x phase currents, 1x DC voltage, 1x DC current)
- **4 EPWMs** (3x phase voltages, 1x ADC synchronisation for DC-Link current sampling)
- **1 interrupt on EPWM CNT=ZRO**

  (processing complete FOC control algorithm with DC-Link current reconstruction)



‚MainISR' called on EPWM CNT=ZRO
Equal time base for all PWMs

EPWM1-4 in up-down count mode
PWM signal set on CMPA match
PWM signal clear on CMPB match

EPWM4 SOCB on CMPB match
$\Rightarrow$ Sample1

EPWM4 SOCA on CMPA match
$\Rightarrow$ Sample2

Software

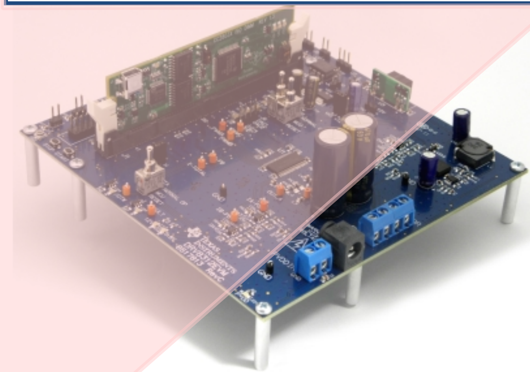
Use of already existing project within Control Suite v2.2.1

– PM_Sensorless project

- TI
  - controlSUITE
    - development_kits
      - DRV8312-C2-KIT
        - ~Docs
        - Project_PM_Sensorless

Motor: DRV8312-C2-KIT Three-Phase Brushless Motor Control

DRV8312 Three-Phase Brushless Motor Control Kit (v1.1)
Buy Now



Main features

- Piccolo F28035 controlCARD with onboard isolated USB JTAG emulation
- DRV8312 3-phase Motor Driver Evaluation Board, 52.5V 6.5A (continuous, per phase)
- 24V DC desktop power supply
- 1x BLDC motor
- GUI, Detailed example software and documentation
- Complete hardware schematics, gerber files, and BOM

Documentation
- How To Run Guide
- Hardware Guide
- GUI Quick Start Guide
- Sensorless FOC of PMSM
- Sensorless Trapezoidal Control of BLDC Motors
- Trapezoidal Control of BLDC Motors Using Hall Effect Sensors

Example projects
- BLDC Sensored
- BLDC Sensorless
- PM Sensorless

Online contents
- DRV8312-C2-KIT Tool Folder
- DRV8312 Product Folder
- TI Integrated Motor Drivers
- C2000 Motor Control Primer
- All C2000 kits and tools
- C2000 Motor Control Site
- Interactive MCU Selection Tool

Folders
- Graphical User Interface (GUI)
- Hardware Development Package
- Digital Motor Control Library
- XDS100 drivers
- Kit Folder

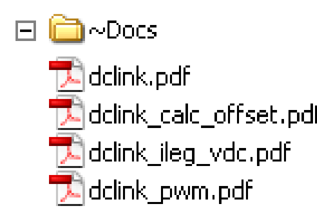
Software

New modules written as macros

- DC-Link current reconstruction f2803xdclink_PM.h
- ADC configuration for DC-Link f2803xdclink_ileg_vdc_PM.h
- PWM configuration for DC-Link f2803xdclink_pwm_PM.h
- Current offset calculation f2803xdclink_calc_offset_PM.h

Additional files

- Linker comm NK_FLASH.cmd
- Module doc

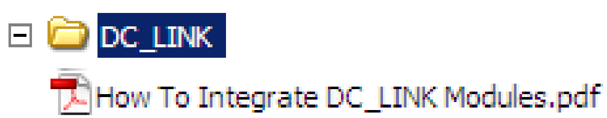


Software

Module integration in existing PM_Sensorless project

1. Import and include module header files and linker command file
2. Declare and define module objects
3. Insert and replace original PWM and ADC initialization macros
4. Insert and replace module function calls

A detailed step-by-step description for DC-Link modules integration can be found in the DC-Link module folder.



Practical Measurement Results

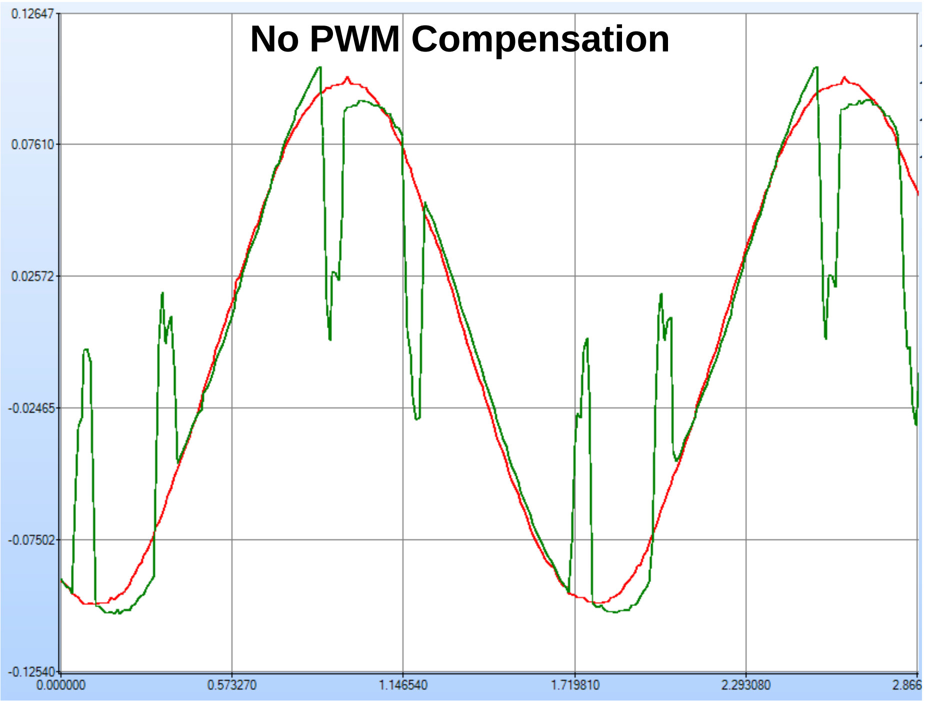
Used hardware

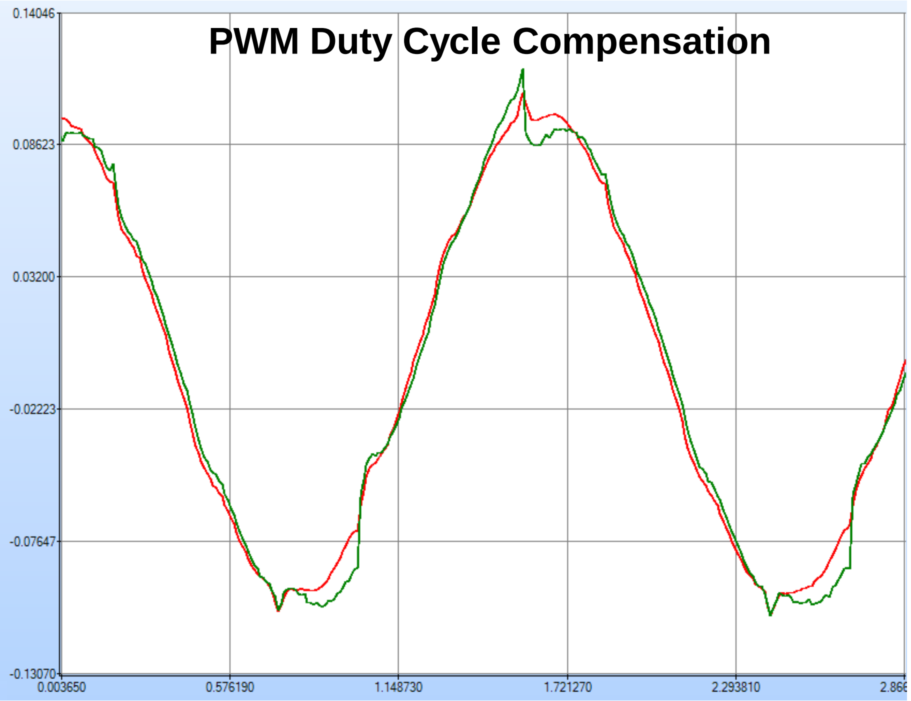
- DRV8312-C2-KIT (F28035) [4]
- Anaheim Automation BLY172S-24V-4000

Practical Measurement Results

No PWM Compensation vs. PWM Duty Cycle Compensation

--- I_A (directly measured in phase A leg of stage inverter)
--- I_A (reconstructed from DC-Link current measurement)





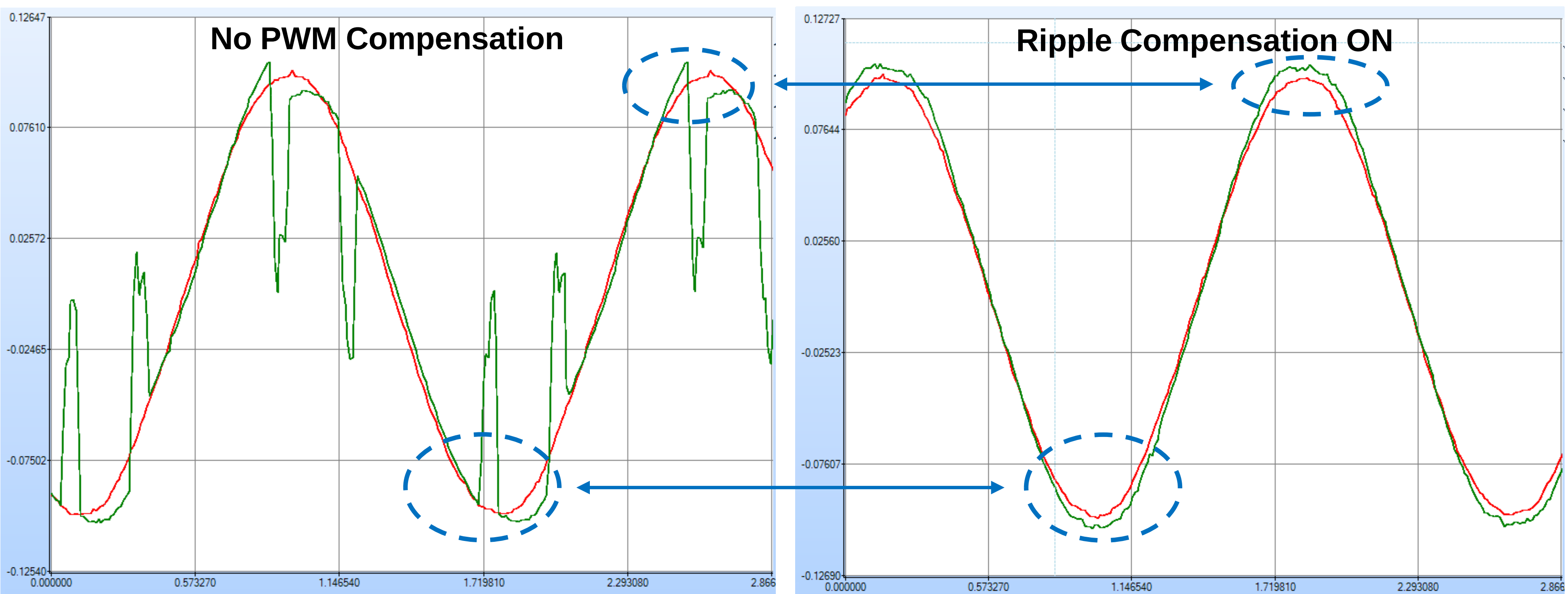
<u>Parameters</u>: SpeedRef = 0.3; Iq=0.1; no load

Practical Measurement Results

PWM Duty Cycle Compensation vs. PWM Phase Shift Compensation

Current Ripple Compensation

--- I_A (directly measured in phase A leg of stage inverter)
--- I_A (reconstructed from DC-Link current measurement)
--- difference in result when using Current Ripple Compensation



Parameters: SpeedRef = 0.3; Iq=0.1; no load

Summary

Following subjects were achieved

- TI MCUs for Motor Control Application
  - MSP430(Low power), Stellaris(32-bit ARM), C2000(DSP) ,Hercules(Safety)
- TI DSP based C2000 series for Motor Control Application
- Current reconstruction algorithm with 6 different options
  - No PWM compensation with/without current ripple compensation
  - PWM Duty Cycle compensation with/without current ripple compensation
  - PWM Phase Shift compensation with/without current ripple compensation
- Software modules written as macros for Piccolo
- Successfully tested on DRV8312-C2-KIT (F28035) with sensorless FOC of PMSM

Thank you

References

[1] Digital Control Systems (DCS) Group (2001). Digital Motor Control Software Library. *SPRU485A*.

[2] Texas Instruments Europe (1998). Three Phase Current Measurements Using a Single Line Resistor on TMS320F240. *BPRA077*.

[3] Darko P. Marceti´c and Evgenije M. Adži´c (2009). Improved Three-Phase Current Reconstruction for Induction Motor Drives With DC-Link Shunt. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 57, NO. 7.*

[4] Texas Instruments. Three Phase BLDC Motor Kit with DRV8312 and Piccolo MCU. *http://focus.ti.com/docs/toolsw/folders/print/drv8312-c2-kit.html* (07/26/2011).

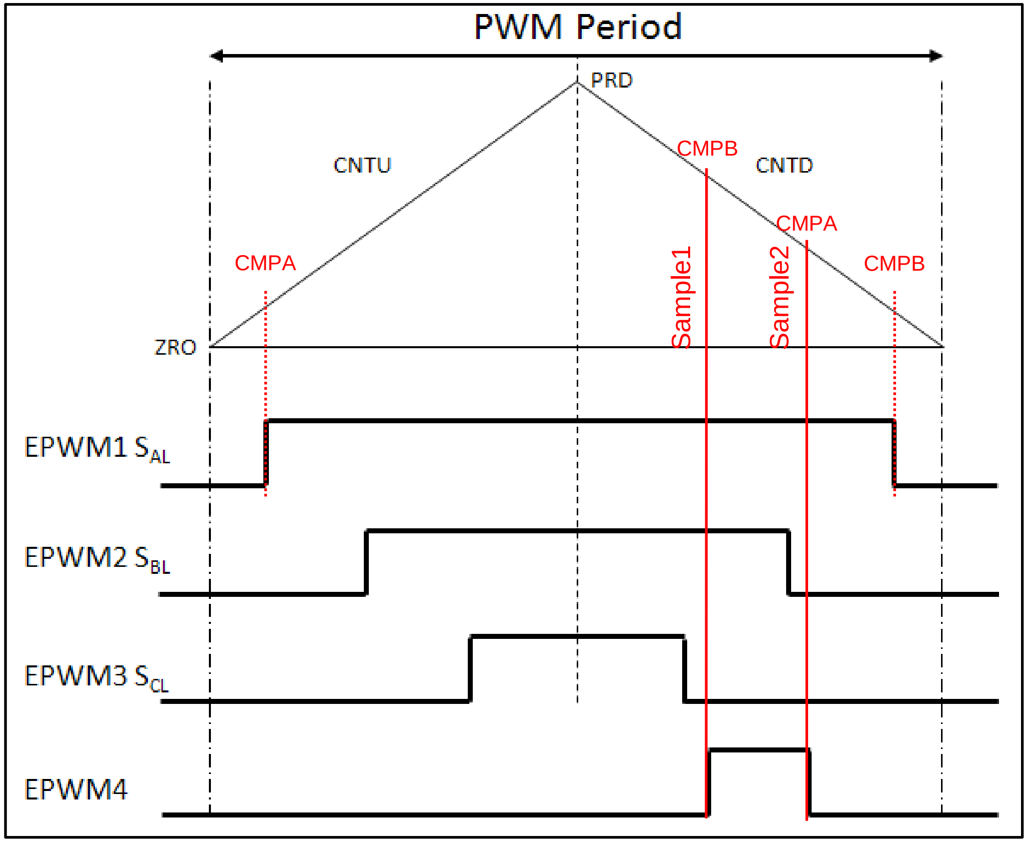
[5] Texas Instruments. ControlSUITE for C2000. *http://processors.wiki.ti.com/index.php/ControlSUITE_for_C2000* (07/26/2011).

BACK UP

Software Implementation

Implemented solution makes use of the following resources

- **5 A/D channels** (1x dummy, 2x phase currents, 1x DC voltage, 1x DC current)
- **4 EPWMs** (3x phase voltages, 1x ADC synchronisation for DC-Link current sampling)
- **1 interrupt on EPWM CNT=ZRO**
  (processing complete FOC control algorithm with DC-Link current reconstruction)



‚MainISR' called on EPWM CNT=ZRO
Equal time base for all PWMs

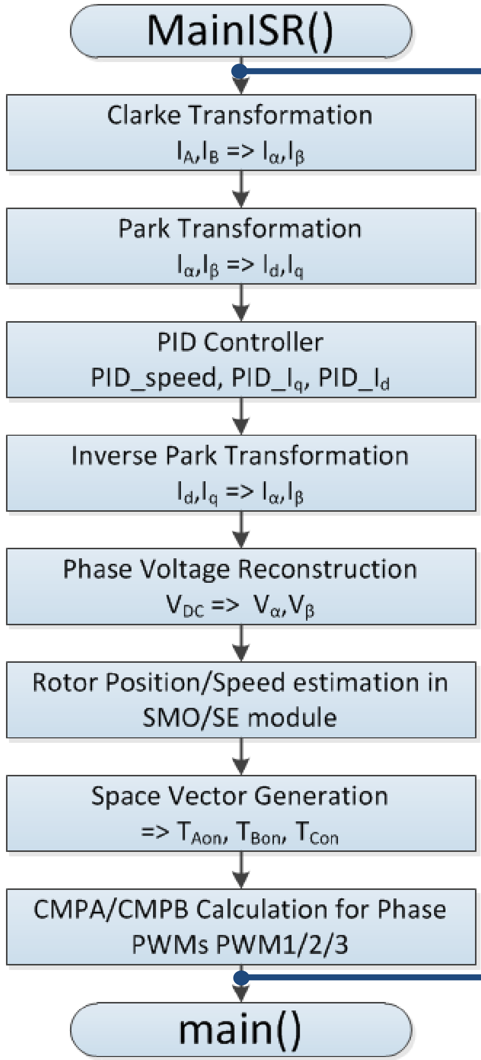
EPWM1-4 in up-down count mode
PWM signal set on CMPA match
PWM signal clear on CMPB match

EPWM4 SOCB on CMPB match
⇒ Sample1

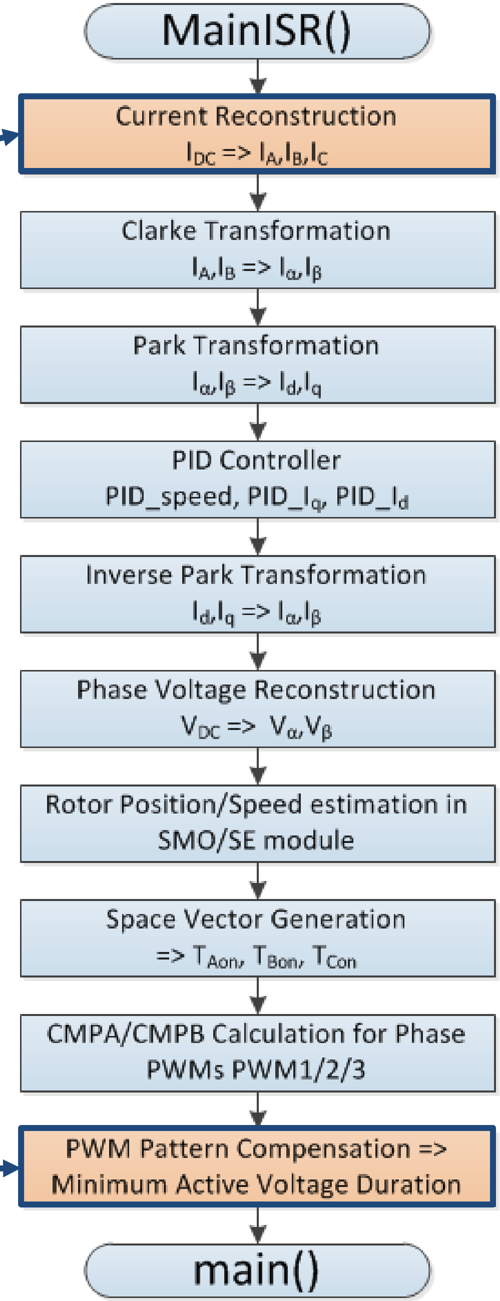
EPWM4 SOCA on CMPA match
⇒ Sample2

Software

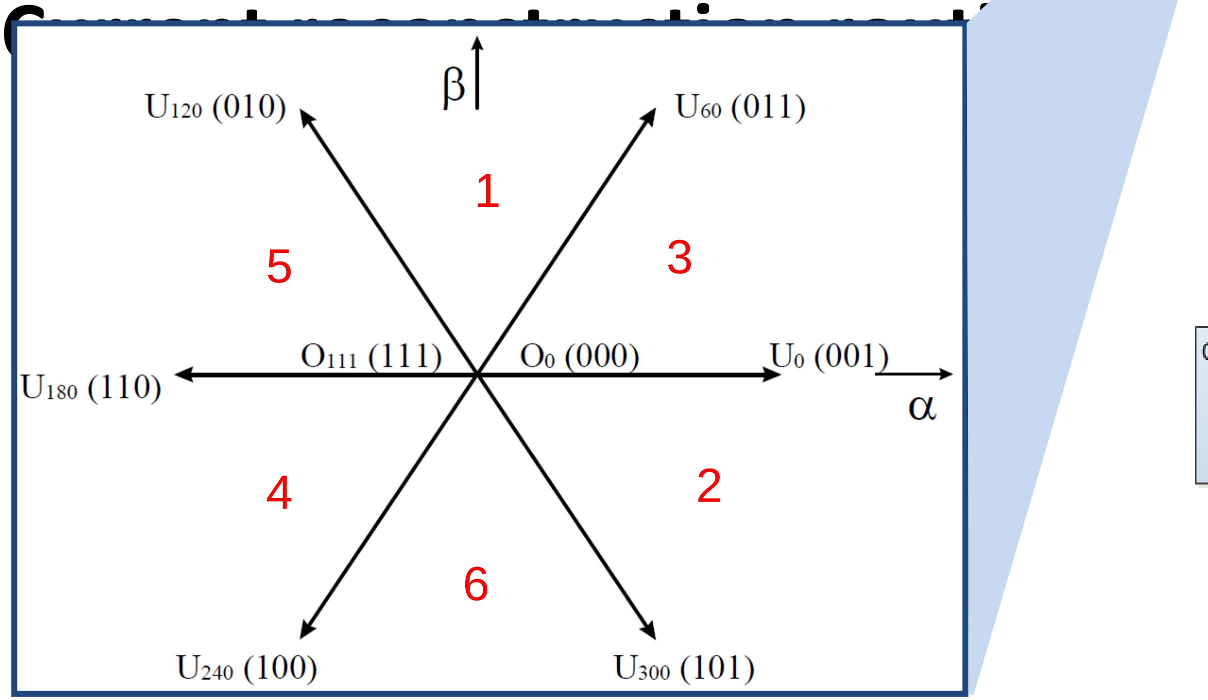
Program flow 'MainISR'



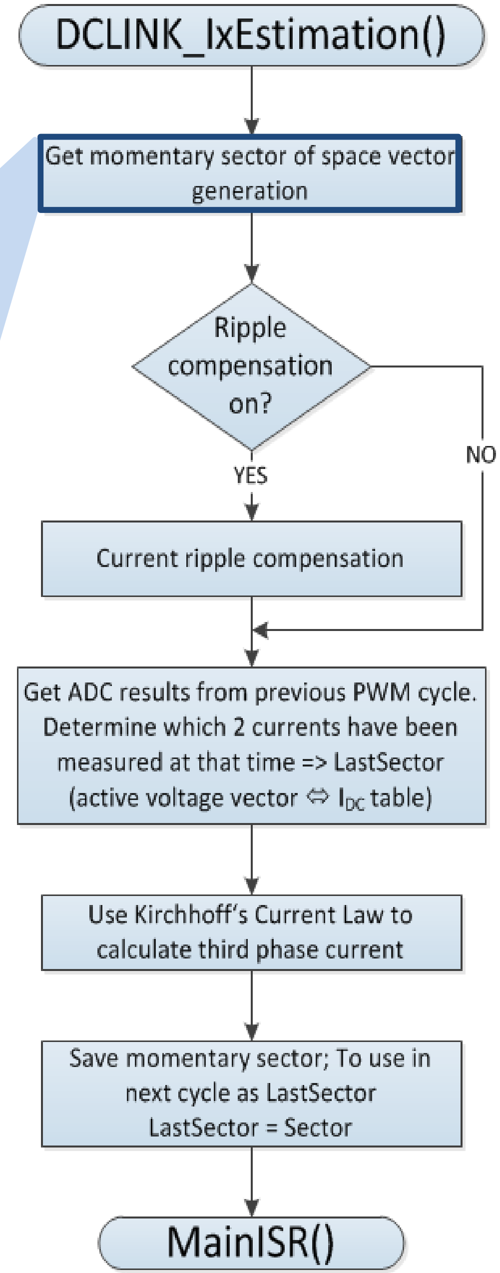
Changes to original ‚MainISR()' code



Software



β
α
U120 (010)
U60 (011)
U180 (110)
O111 (111)
O0 (000)
U0 (001)
U240 (100)
U300 (101)
1
2
3
4
5
6

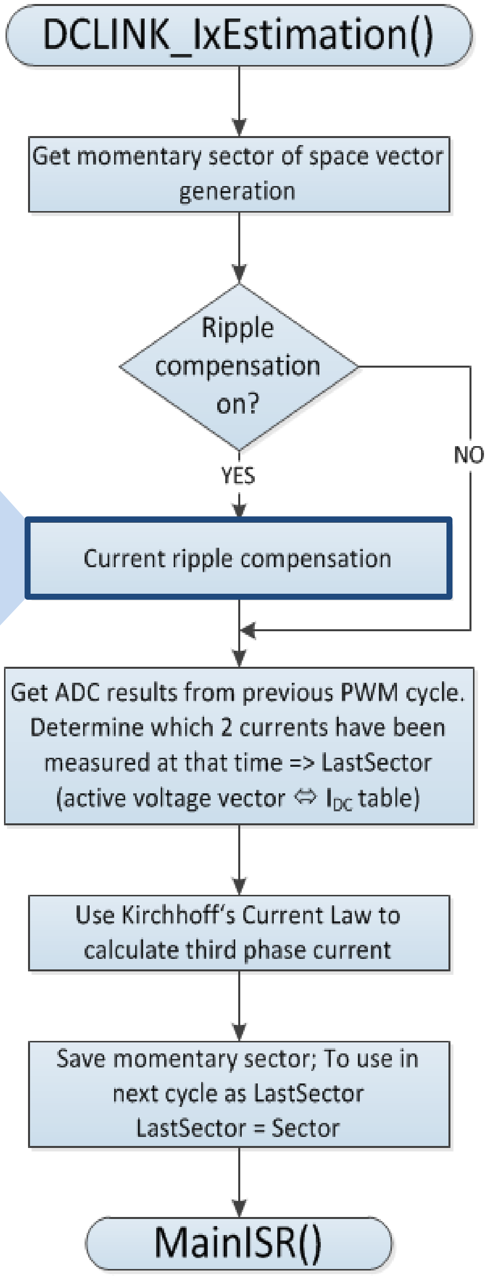


DCLINK_IxEstimation()
Get momentary sector of space vector generation
Ripple compensation on?
YES
NO
Current ripple compensation
Get ADC results from previous PWM cycle. Determine which 2 currents have been measured at that time => LastSector (active voltage vector ⇔ IDC table)
Use Kirchhoff's Current Law to calculate third phase current
Save momentary sector; To use in next cycle as LastSector LastSector = Sector
MainISR()

Software

Current reconstruction routine

Sector	I_A	I_B	I_C
1	$I_{DC2}-\Delta I_{a2}$	$-(I_A+I_C)$	$-I_{DC1}-\Delta I_{c1}$
2	$-(I_B+I_C)$	$I_{DC2}-\Delta I_{b2}$	$-I_{DC1}-\Delta I_{c1}$
3	$-I_{DC1}-\Delta I_{a1}$	$I_{DC2}-\Delta I_{b2}$	$-(I_A+I_B)$
4	$-I_{DC1}-\Delta I_{a1}$	$-(I_A+I_C)$	$I_{DC2}-\Delta I_{c2}$
5	$-(I_B+I_C)$	$-I_{DC1}-\Delta I_{b1}$	$I_{DC2}-\Delta I_{c2}$
6	$I_{DC2}-\Delta I_{a2}$	$-I_{DC1}-\Delta I_{b1}$	$-(I_A+I_B)$

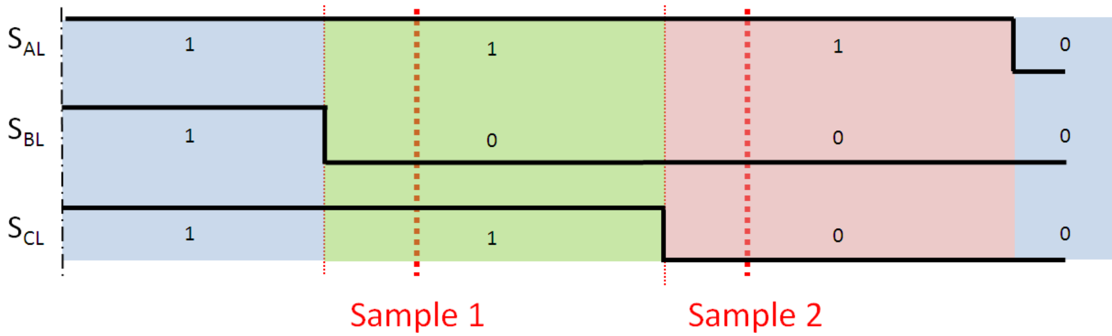


Software

Current reconstruction routine

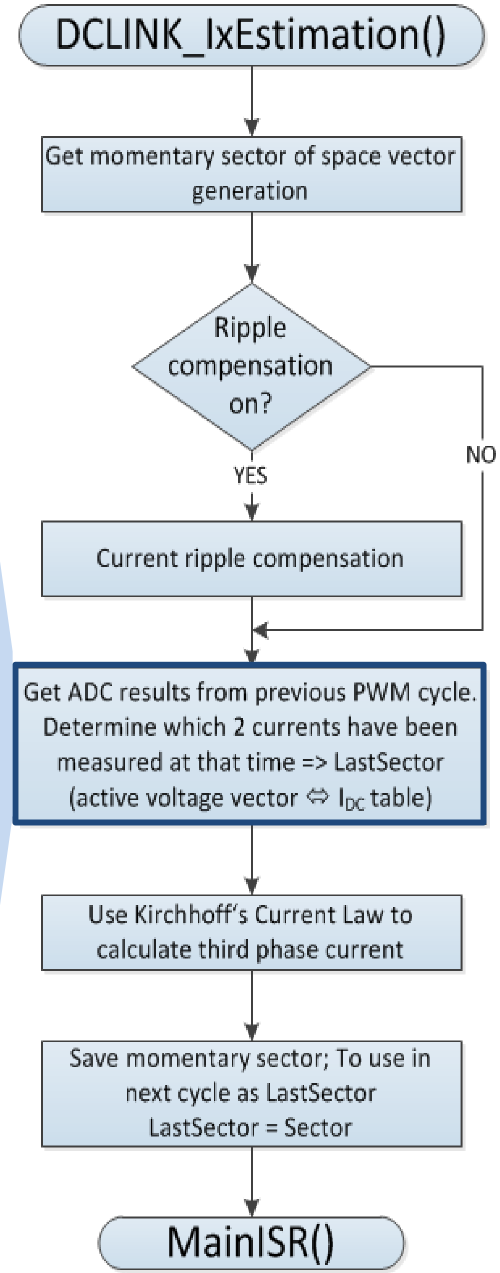
ADCRESULT 10-14:
Sample 1

ADCRESULT 4-9:
Sample 2



SECTOR	S_{AL}	S_{BL}	S_{CL}	I_{DC}
2 / 3	NC	C	C	$+I_A$
1 / 5	C	NC	C	$+I_B$
4 / 6	C	C	NC	$+I_C$
4 / 5	C	NC	NC	$-I_A$
2 / 6	NC	C	NC	$-I_B$
1 / 3	NC	NC	C	$-I_C$

C: **C**onducting Transistor (1)
NC: **N**on-**C**onducting Transistor (0)



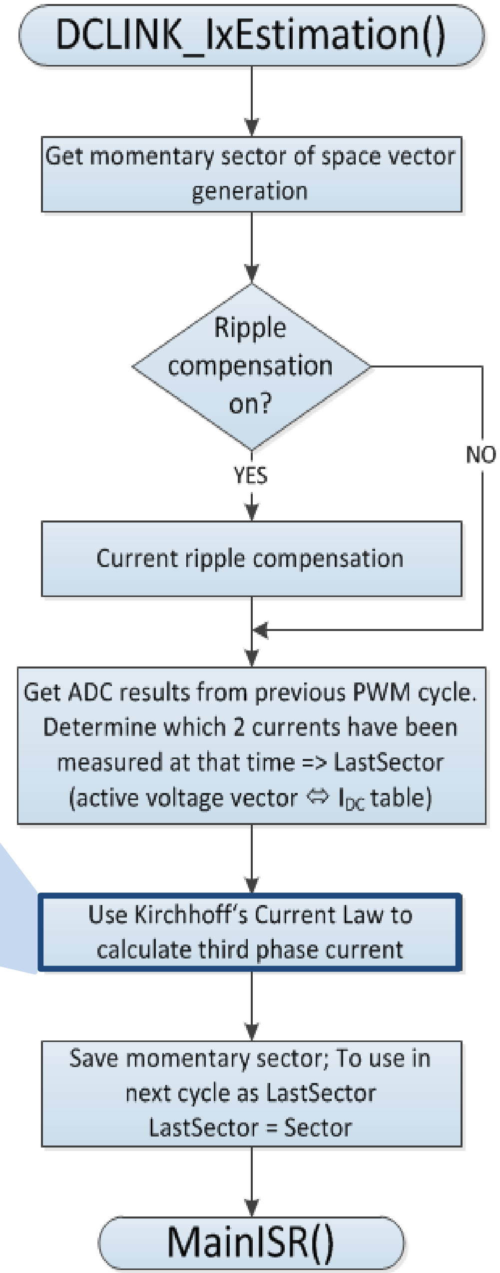
Software

Current reconstruction routine

$$I_A + I_B + I_C = 0$$

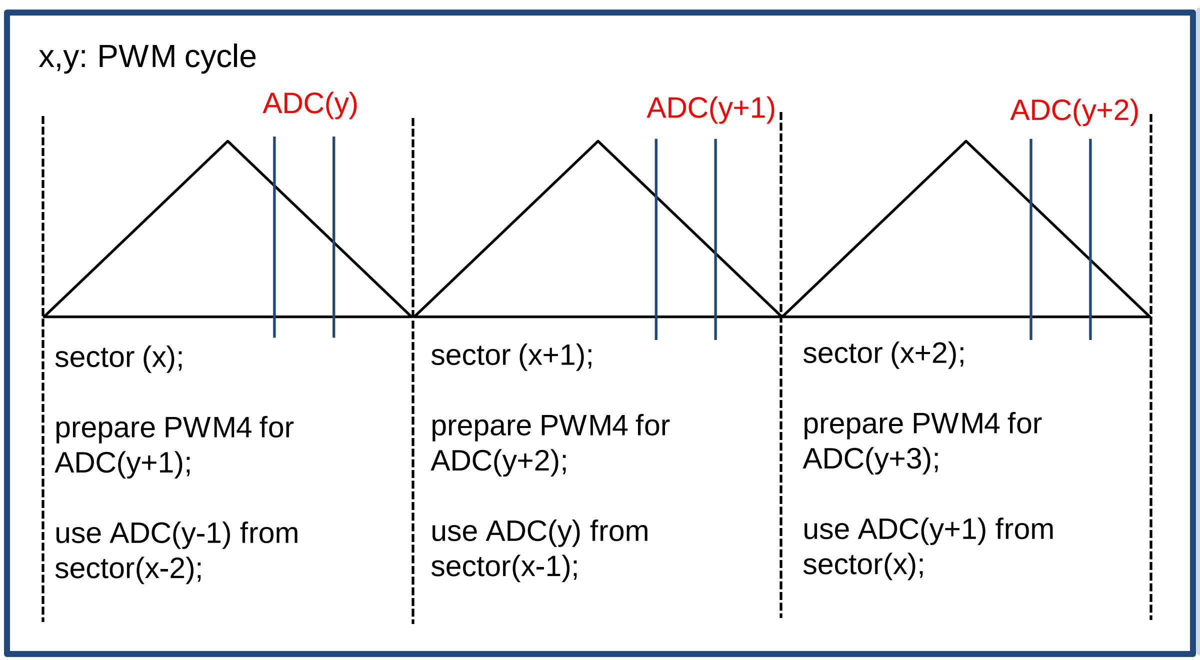
$$I_C = - I_A - I_B$$

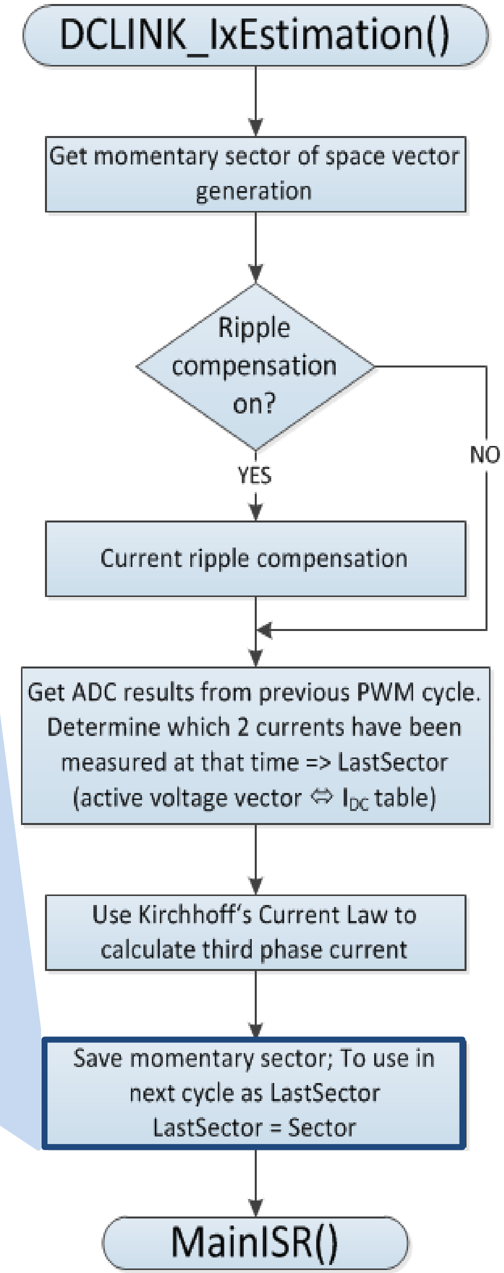
- (ADCRESULT10-14) (ADCRESULT4-9)



Software

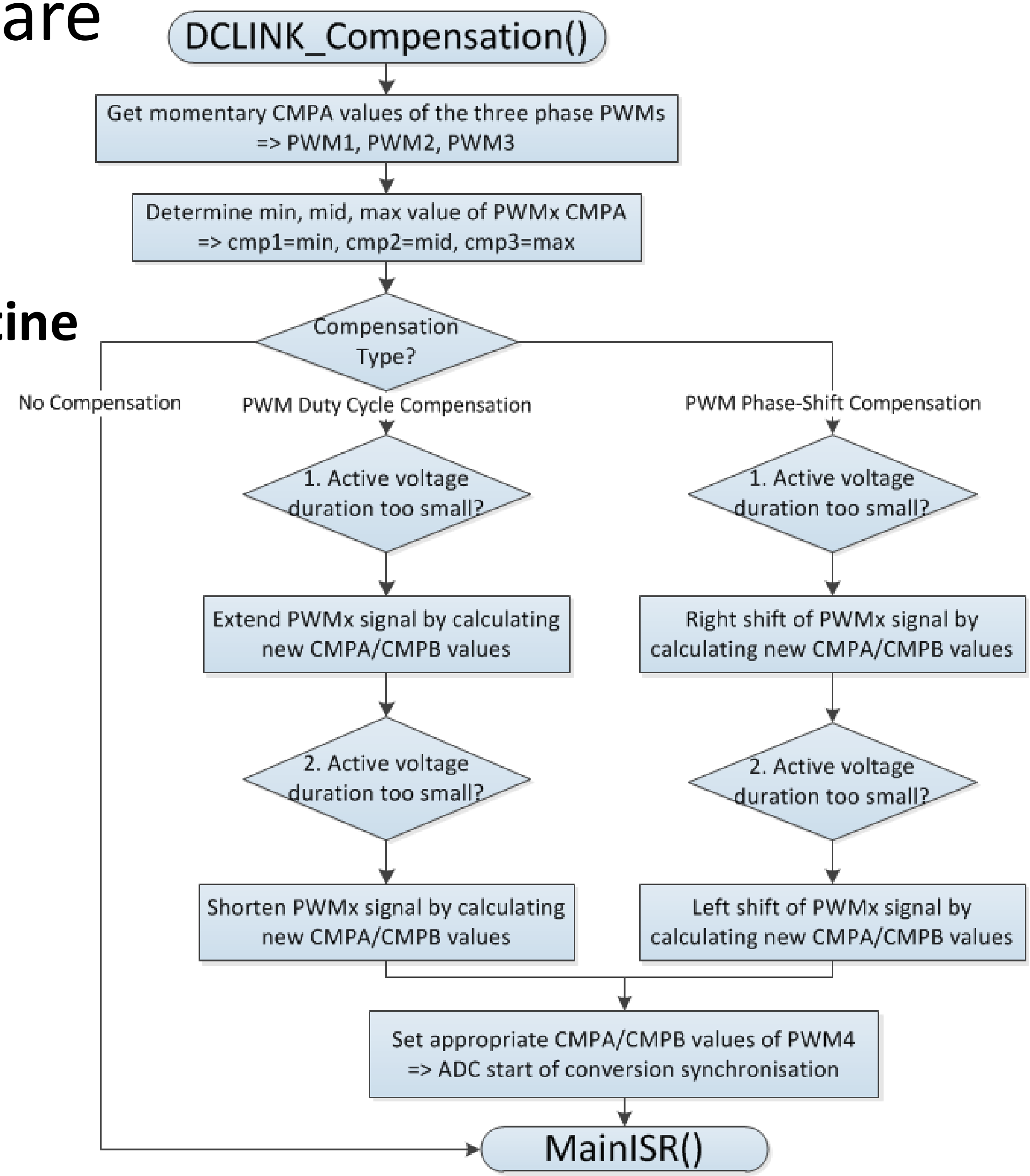
Current reconstruction routine





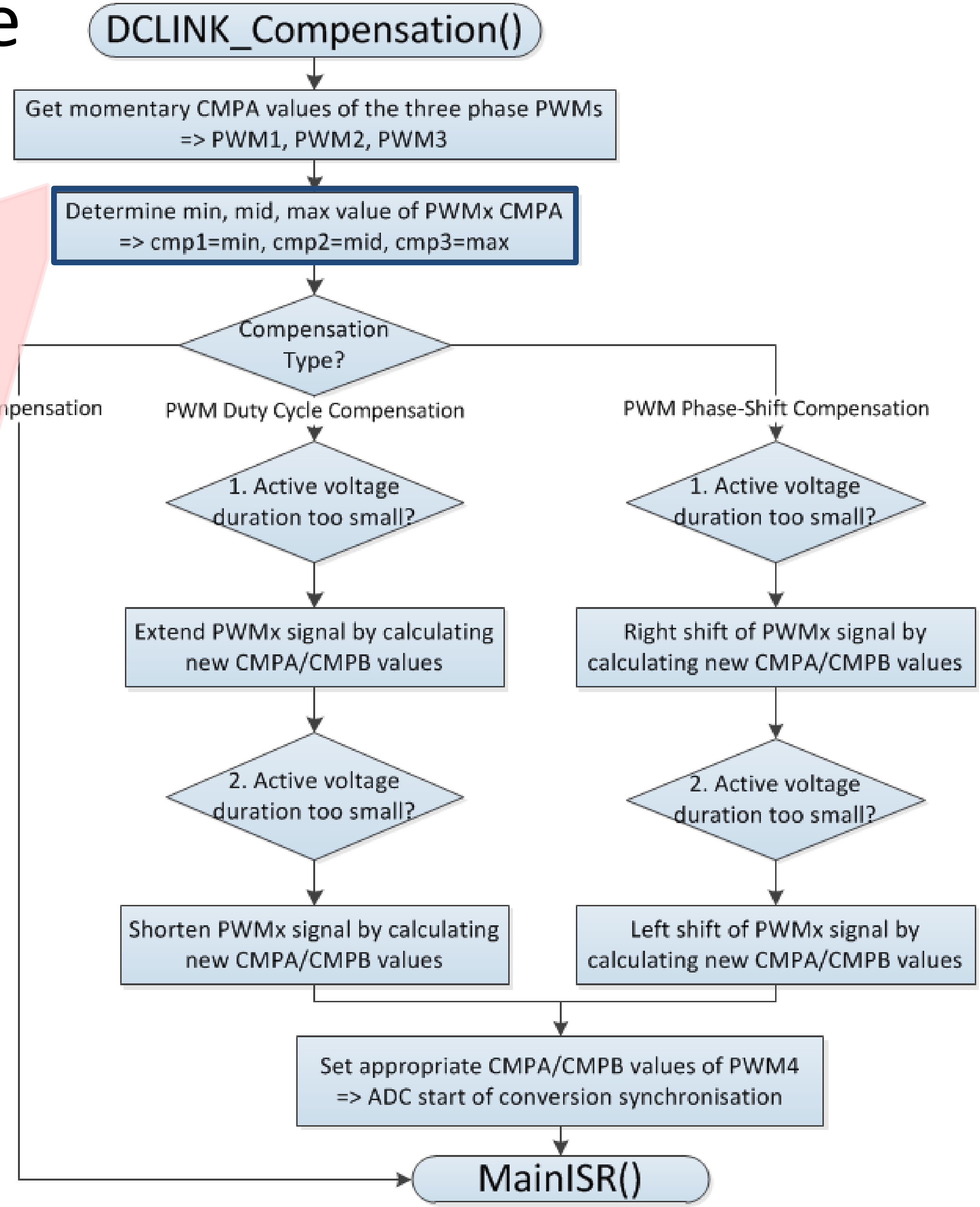
Software

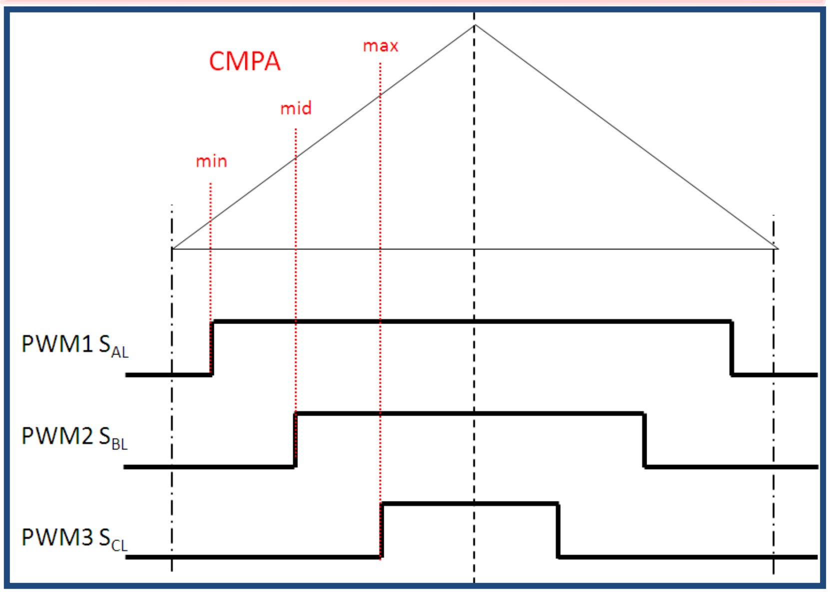
PWM compensation routine



Software

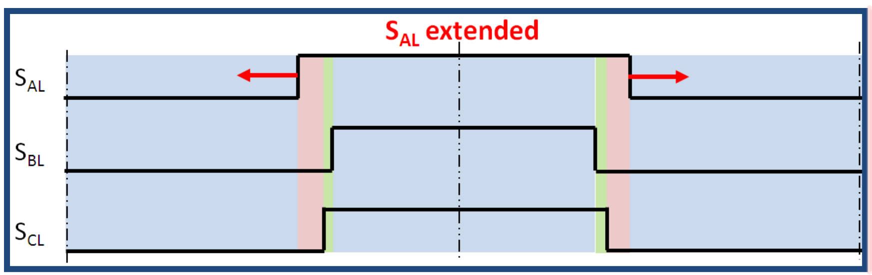
PWM compensation routine

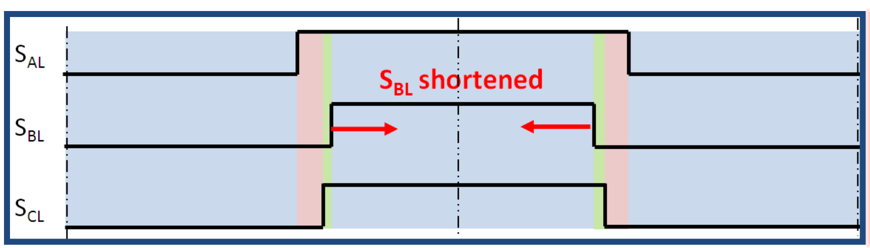


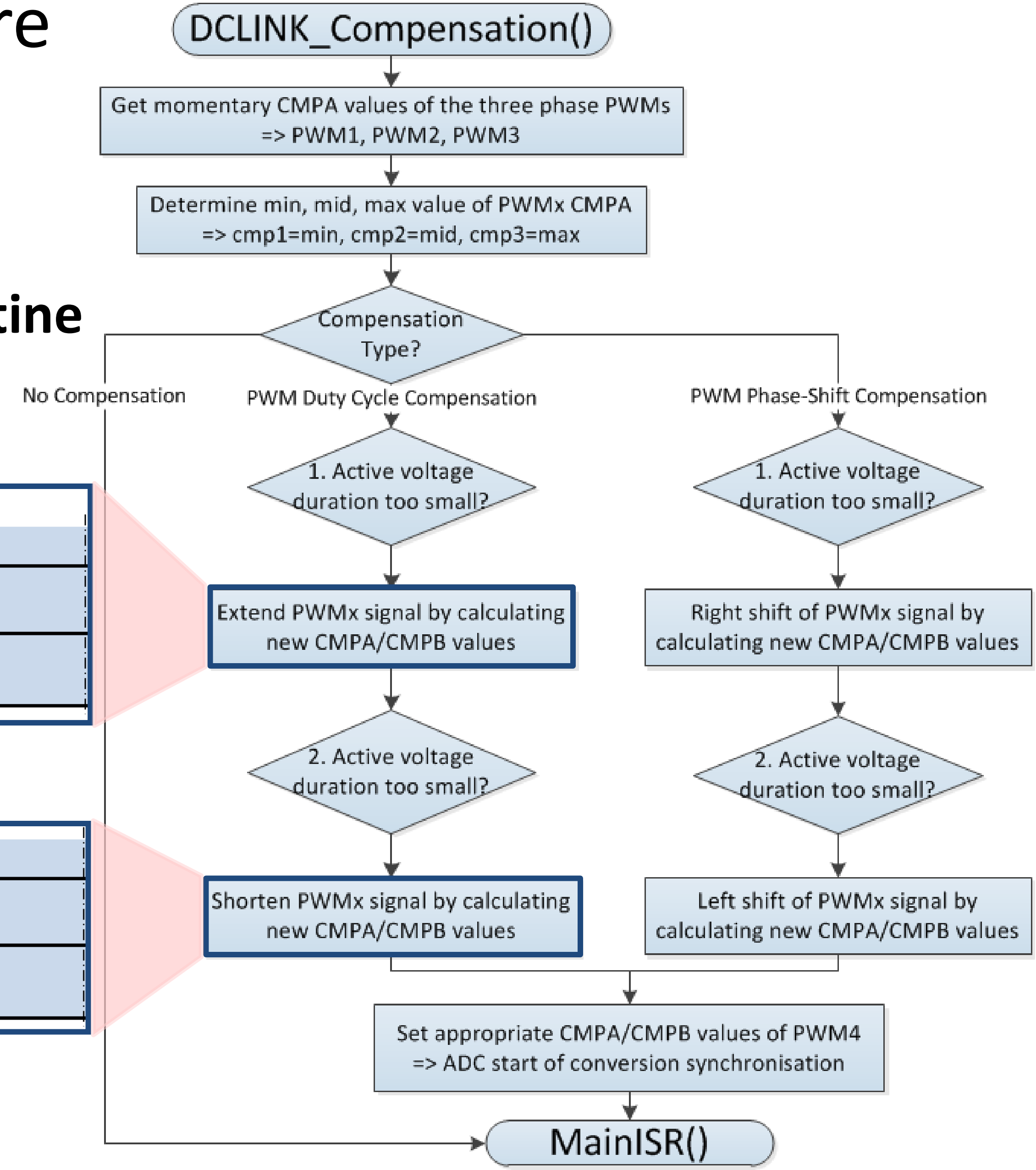


Software

PWM compensation routine

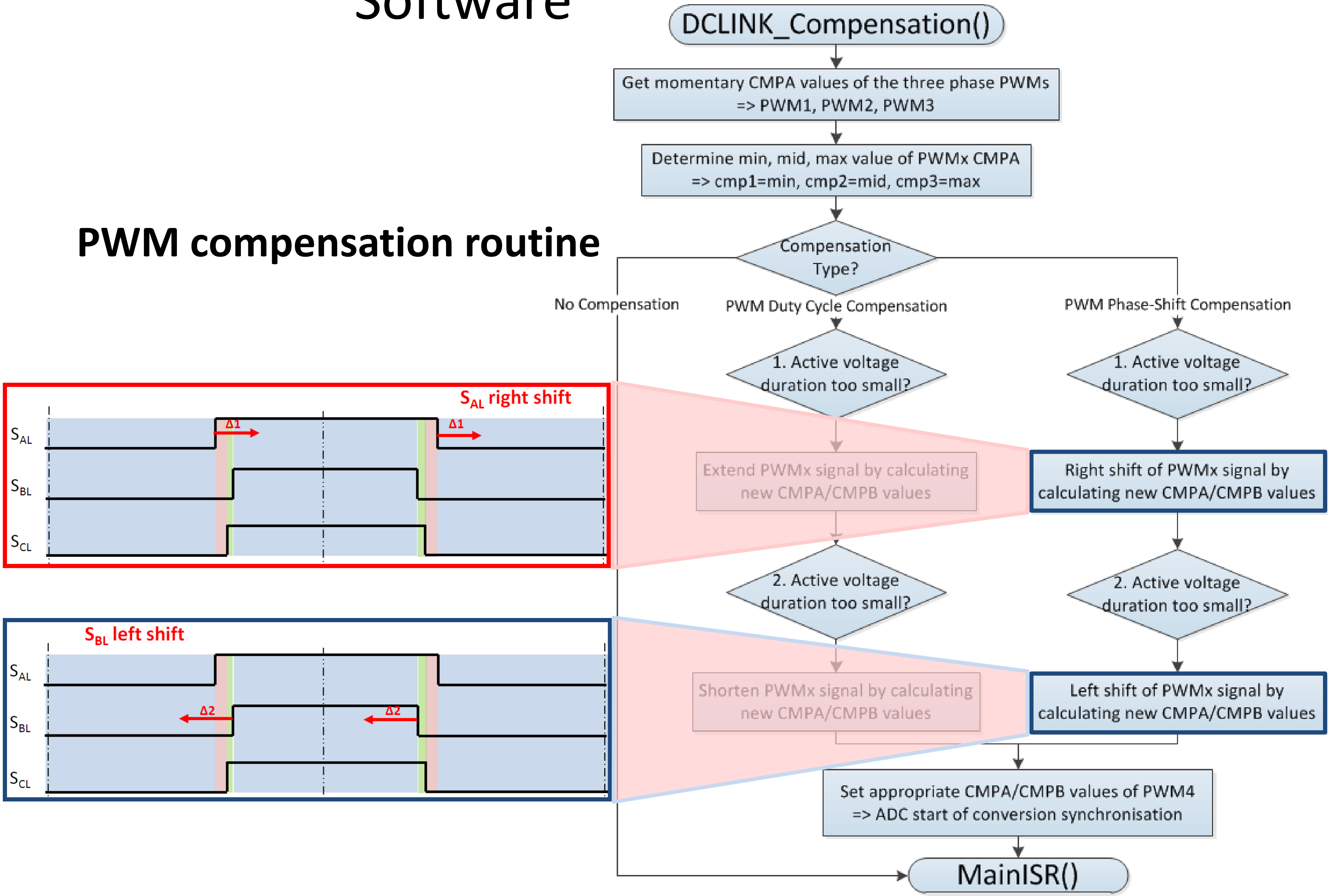






Software

PWM compensation routine



Software

PWM compensation routine