Hands-on Workshop: Motor Control: Efficient and Easy Designs

AMF-IND-T0145

Eduardo Viramontes
Applications Engineer

September 2013

Agenda

- Motor control challenges
- Motor control solutions overview
- Freescale Motor control IP
- Motor control enablement
  - MC Libraries
  - Safety libraries
  - FreeMASTER
  - MCAT
  - Roadmap
- Hands on / Demo

Motor control challenges

What can impact MCU selection?

- Motor type
- Used Hardware (Sensors type and connection)
- Speed range / Sensorless operation
- Application dynamic
- Motor parameters
- Application complexity (Other application requirements)

Motor Type

- **Number of PWM channels**
  - DC Motors 1 or 4 channels
  - BLDC motor, PMSM and ACIM: 6 channel
- **Sine wave generation (PMSM, ACIM)**
  - Complementary logic
  - automatic dead time insertion
- **Electronic commutation (BLDC motors, SR motors)**
  - mask, swap, restart PWM features
    These features allow to provide commutation without change of duty cycle
- **Fault Control**

Sensors Type and Connection

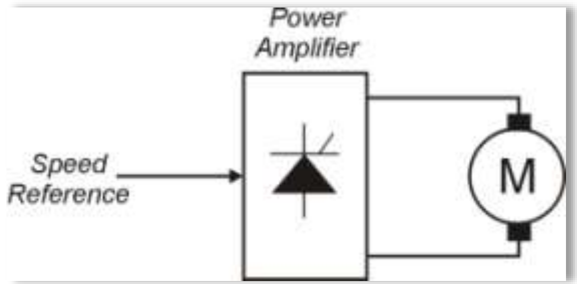
- **Speed/Position Measurement**
  - If quadrature encoder used, decoding of quadrature signals is necessary
- **Current measurement**
  - If there is current loop fast ADC (<2.5μs) is advantage (the less time spent by ADC conversion, the more time for control loop calculation. Typically the current control loop is 50 – 150 μs
  - If shunts used for current sensing the PWM to ADC synchronization necessary

Speed range / Sensorless Operation

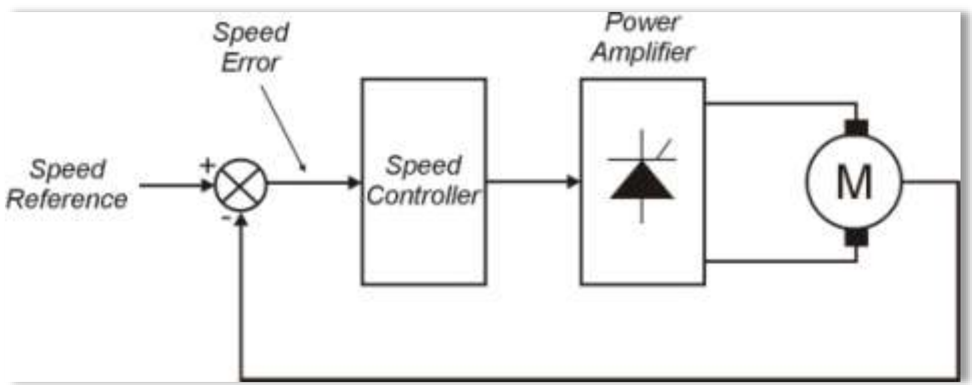
- **Speed Range**
  - High speed especially for electronically commutated motors requires powerful MCU include powerful peripherals (HW support of commutation, fast ADC) since commutation period becomes very short (few μs)
  - Zero or low speed may be issue for sensorless algorithms
- **Sensorless Motor control**
  - DC/BLDC motors
    - Simple algorithms, can run on 8-bit MCU
  - ACIM, PMSM
    - Require powerful MCU core due to motor model calculation

Dynamic performance

- **Open Loop Control System**

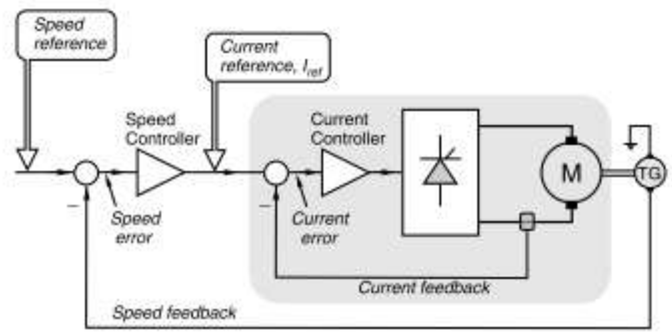


- **Close Loop Control System**



Dynamic performance

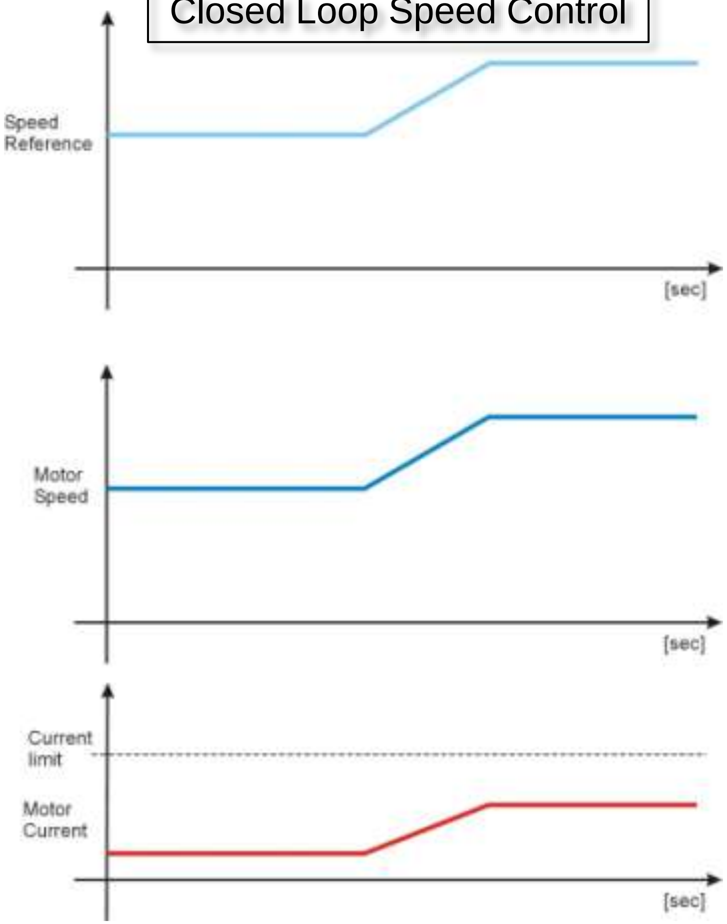
- **Speed Control**
  - Applications requiring the motor to operate with a specified speed (pumps, fans, compressors, etc.)
  - Low dynamic performance
  - The actual motor speed is kept by speed controller to follow reference speed command
- **Speed Control with Inner Current Loop**
  - Majority of variable speed drives
  - High dynamic performance
- **Position Control**
  - Applications with additional position control loop to keep desired position (servos, industrial robots, linear motors)
  - Most complex drives
- **Torque Control**
  - Applications requiring the motor to operate with a specified torque regardless of speed (vehicles, electric power steering, winding machines, etc.)



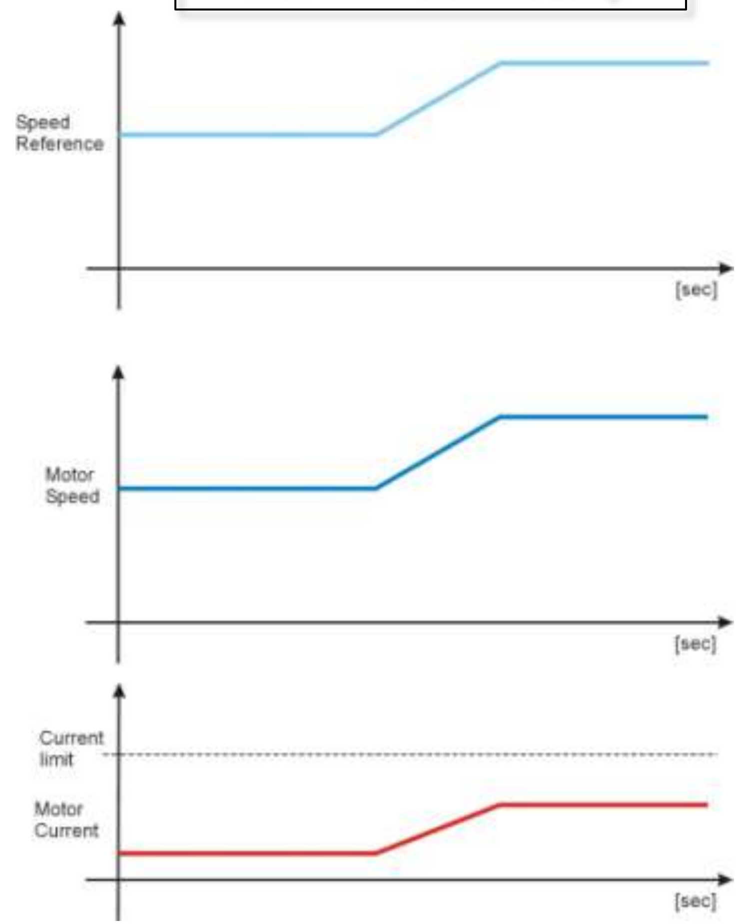
Dynamic performance

- Low Dynamic Applications

Closed Loop Speed Control



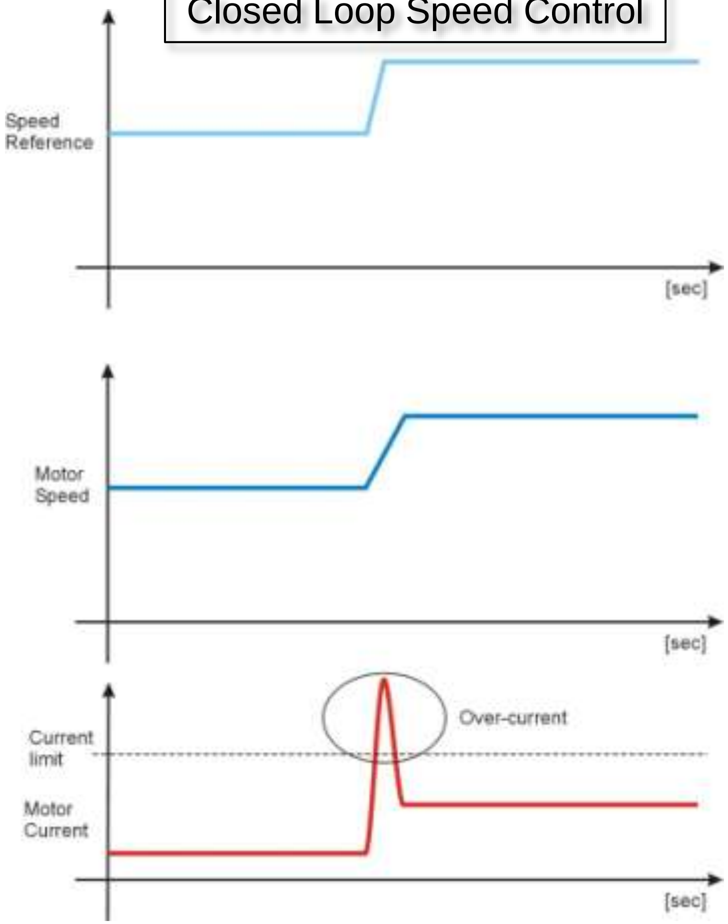
Closed Loop Speed Control
with Inner Current Loop



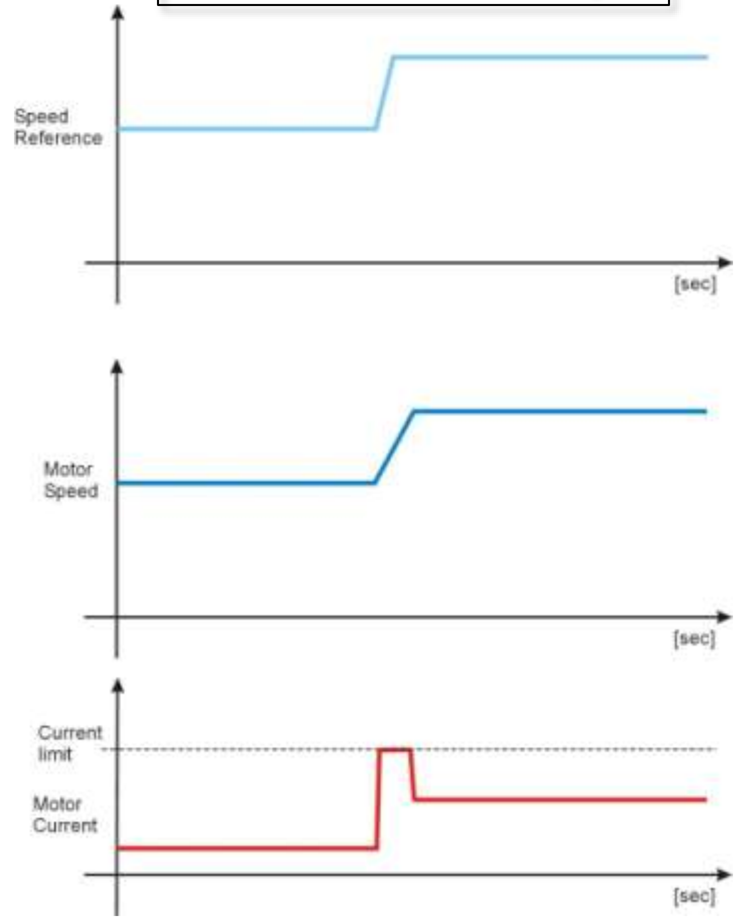
Dynamic performance

- High Dynamic Applications

Closed Loop Speed Control



Closed Loop Speed Control with Inner Current Loop



Dynamic performance

- **Low dynamic performance**
  - Speed control loop only
  - Volt per Hertz (V/Hz) method is suitable for low dynamic drives (ACIM & PMSM)
  - Low performance MCU core required (also 8-bit)
- **High dynamic performance**
  - Inner current loop brings benefit for high dynamic application
  - Inner current loop requires more computation power since the current controller is calculated every PWM period
  - Speed Control with inner current/torque loop (DC/BLDC motors)
    - Current control loop calculated every PWM period
    - 16-bit MCU preferred
  - Field Oriented Control (ACIM and PMSM)
    - FOC loop calculated every or second PWM period
    - Powerful 16-bit MCU core required

Motor Parameters

- **The motor drive has two important time constants:**
- **Electrical motor constant**
  - The electrical constant is defined by RL parameters of stator windings:

$$\tau_e = \frac{L}{R}$$

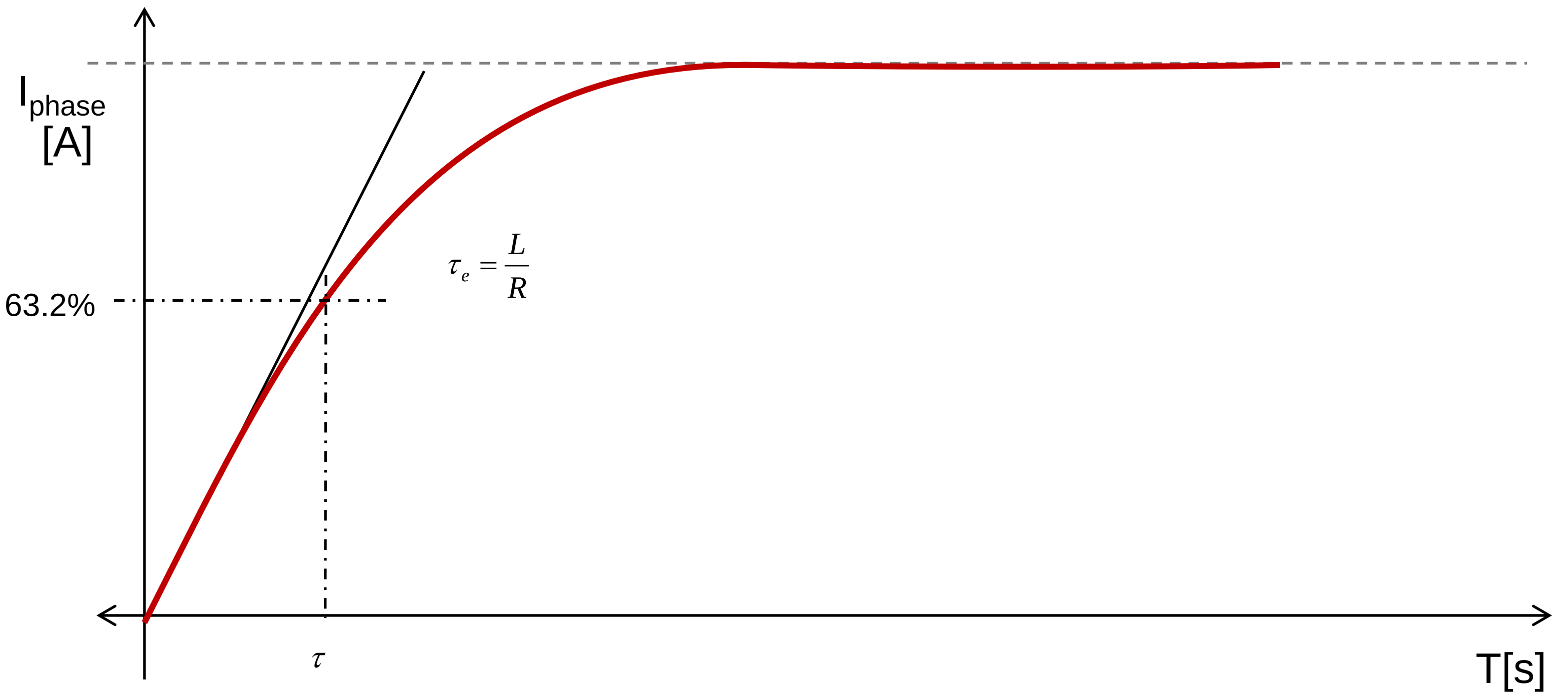
  - The electrical constant impacts the execution/timing of current loop
- **Mechanical motor constant**
  - The mechanical constant is defined by the motor inertia include the load
  - The mechanical constant impacts the execution/timing of speed loop

- **Since the electrical constant is much smaller than mechanical, it has critical impact on MCU performance**

freescale™

Motor Parameters

- **The execution time of control loop should be ideally at least 10-times faster than the time constant of control loop**



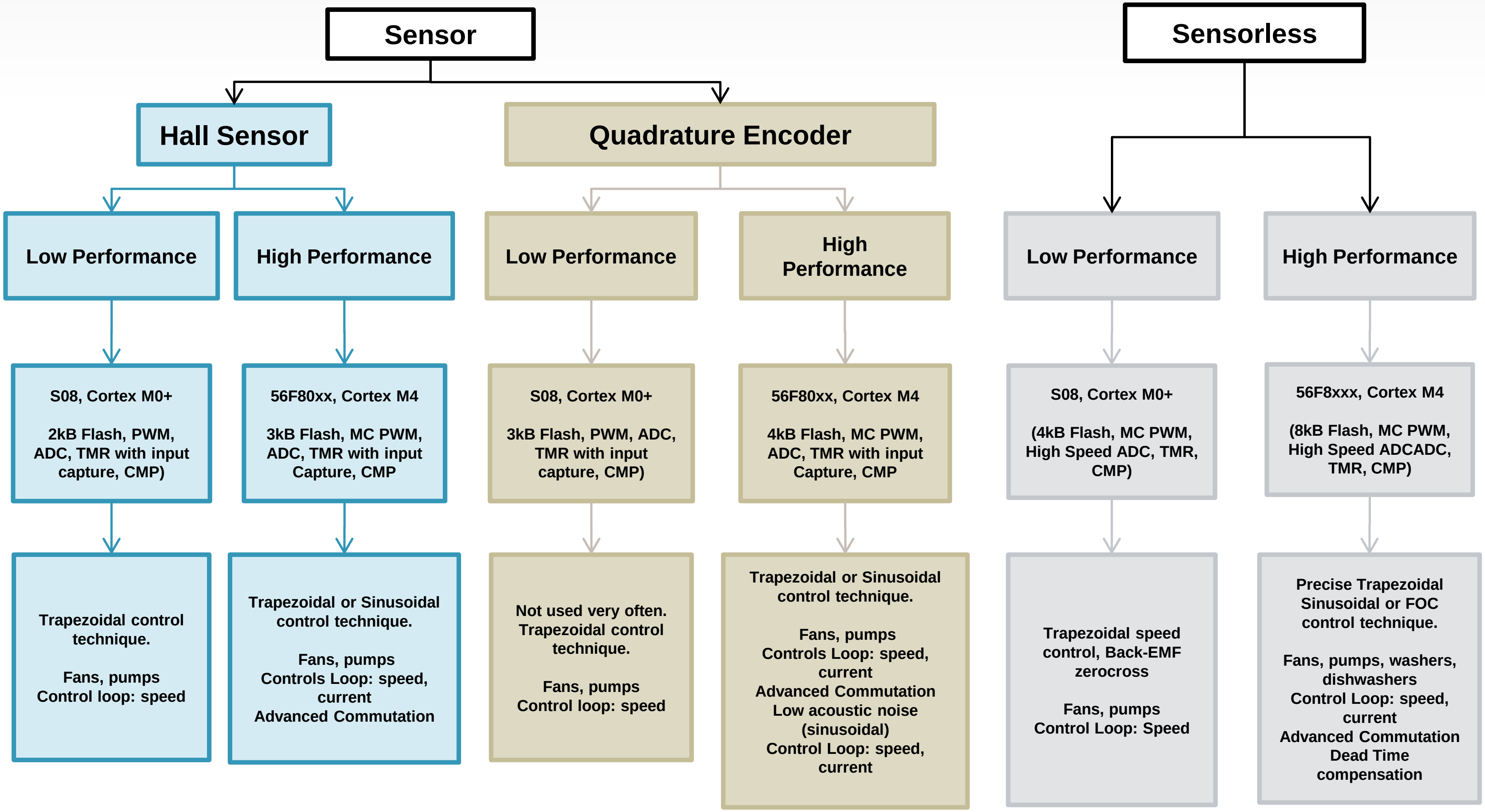
Motor Parameters

- **The execution time of control loop is multiple of PWM period**
- **If $\tau/10$ is significantly longer than the PWM period, the control loop is executed every 2nd , 4th … PWM period -> more time for control loop calculation -> less powerful MCU can be used**
- **If the $\tau/10$ is extremely lower than PWM period it may lead to increase PWM frequency to keep control loop stable**

Application complexity

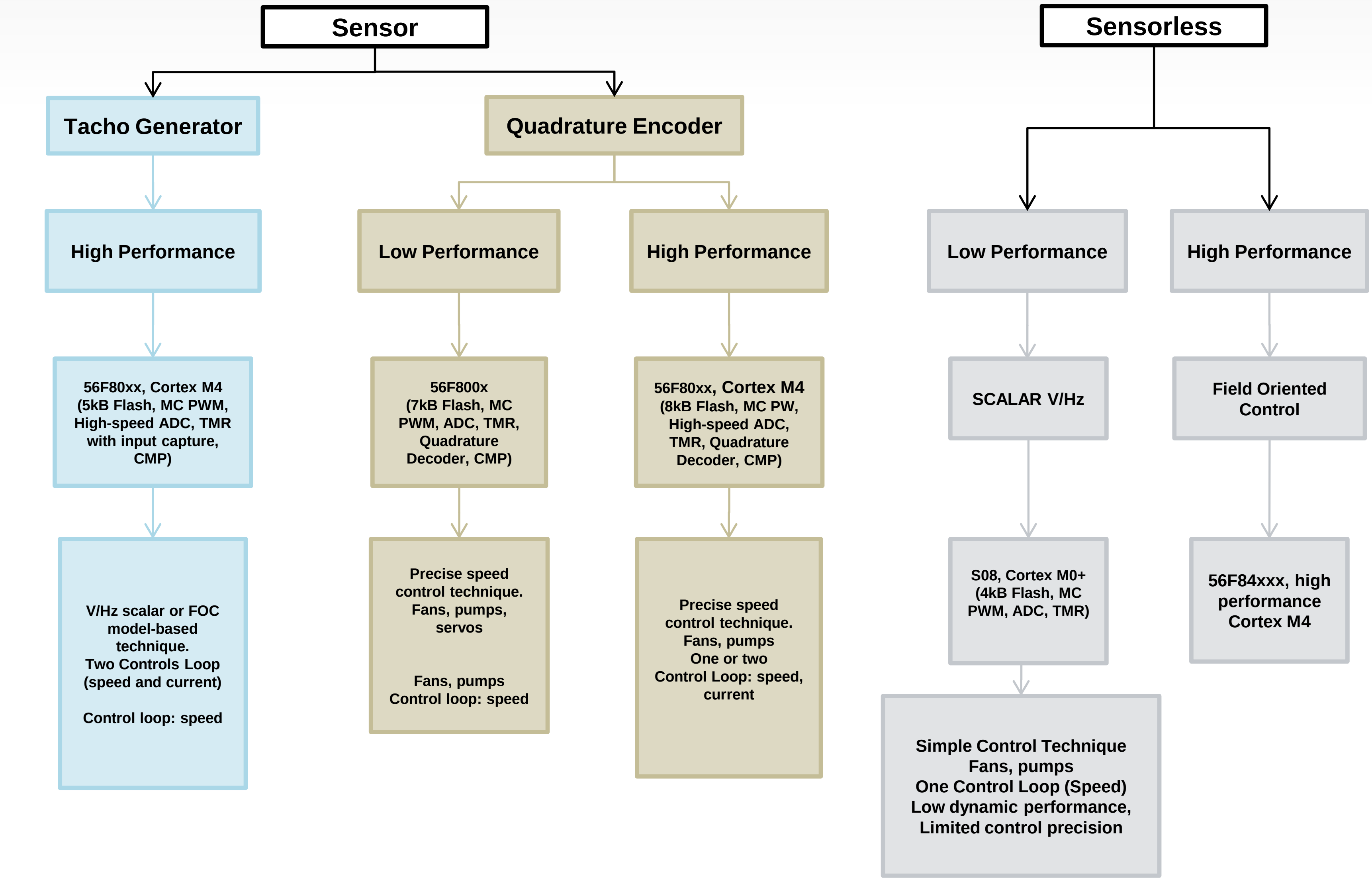
- **There may be other application requirements, which can limit the MCU selection like:**
  - **Communication requirements**
    - Ethernet, CAN, USB, SD card
  - **Graphical interface**
    - LCD, VGA controllers
  - **Application memory requirements**

BLDC

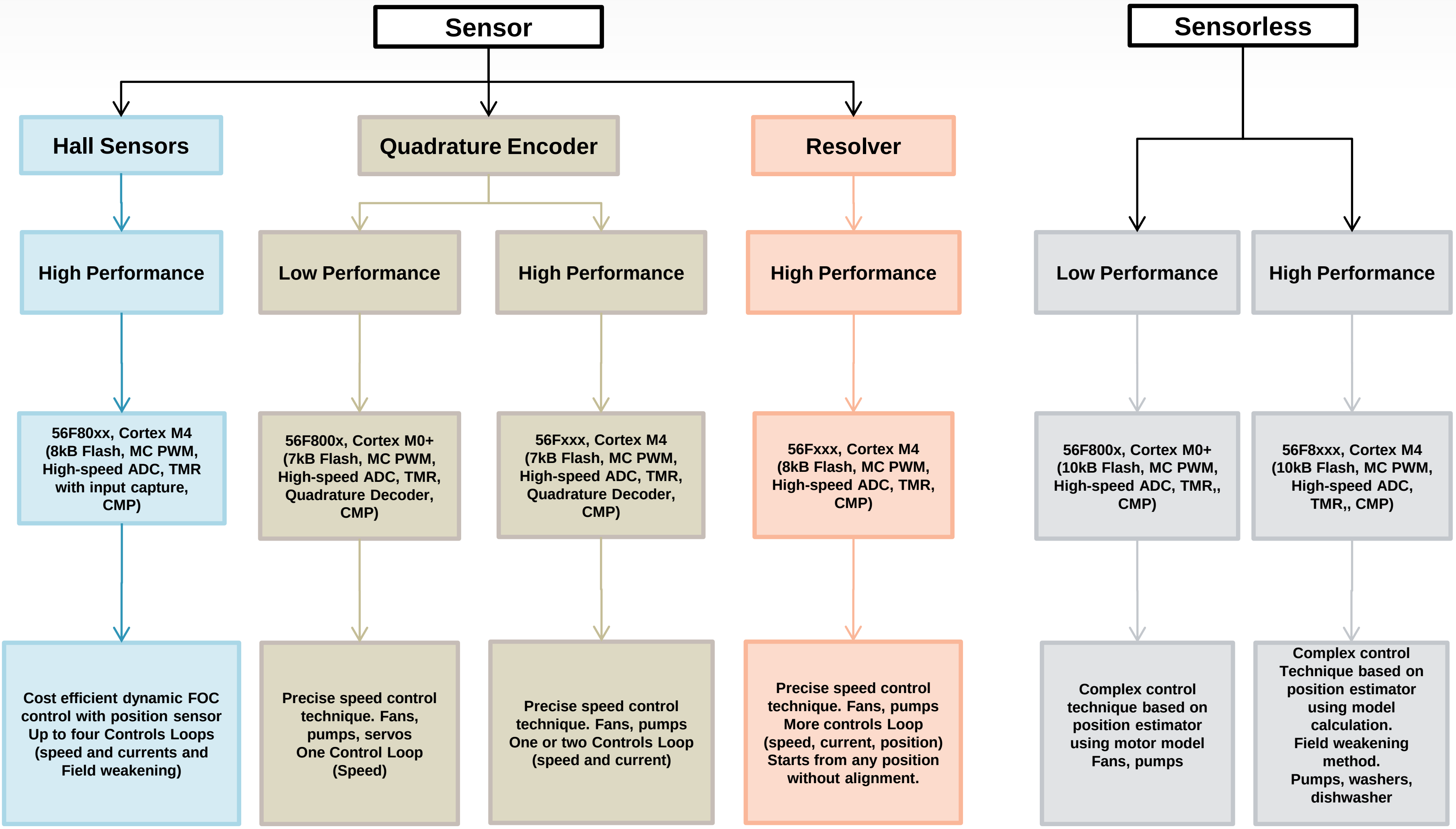


freescale™

ACIM


Sensor
Sensorless
Tacho Generator
Quadrature Encoder
High Performance
Low Performance
High Performance
Low Performance
High Performance
56F80xx, Cortex M4 (5kB Flash, MC PWM, High-speed ADC, TMR with input capture, CMP)
56F800x (7kB Flash, MC PWM, ADC, TMR, Quadrature Decoder, CMP)
56F80xx, Cortex M4 (8kB Flash, MC PW, High-speed ADC, TMR, Quadrature Decoder, CMP)
SCALAR V/Hz
Field Oriented Control
V/Hz scalar or FOC model-based technique. Two Controls Loop (speed and current)
Control loop: speed
Precise speed control technique. Fans, pumps, servos
Fans, pumps Control loop: speed
Precise speed control technique. Fans, pumps One or two Control Loop: speed, current
S08, Cortex M0+ (4kB Flash, MC PWM, ADC, TMR)
56F84xxx, high performance Cortex M4
Simple Control Technique Fans, pumps One Control Loop (Speed) Low dynamic performance, Limited control precision

PMSM



Sensor
Sensorless
Hall Sensors
Quadrature Encoder
Resolver
High Performance
Low Performance
High Performance
High Performance
Low Performance
High Performance
56F80xx, Cortex M4 (8kB Flash, MC PWM, High-speed ADC, TMR with input capture, CMP)
56F800x, Cortex M0+ (7kB Flash, MC PWM, High-speed ADC, TMR, Quadrature Decoder, CMP)
56Fxxx, Cortex M4 (7kB Flash, MC PWM, High-speed ADC, TMR, Quadrature Decoder, CMP)
56Fxxx, Cortex M4 (8kB Flash, MC PWM, High-speed ADC, TMR, CMP)
56F800x, Cortex M0+ (10kB Flash, MC PWM, High-speed ADC, TMR,, CMP)
56F8xxx, Cortex M4 (10kB Flash, MC PWM, High-speed ADC, TMR,, CMP)
Cost efficient dynamic FOC control with position sensor Up to four Controls Loops (speed and currents and Field weakening)
Precise speed control technique. Fans, pumps, servos One Control Loop (Speed)
Precise speed control technique. Fans, pumps One or two Controls Loop (speed and current)
Precise speed control technique. Fans, pumps More controls Loop (speed, current, position) Starts from any position without alignment.
Complex control technique based on position estimator using motor model Fans, pumps
Complex control Technique based on position estimator using model calculation. Field weakening method. Pumps, washers, dishwasher

Development of the Motor Control Application

- Several challenges wait for developer
  - Understand the physical/mathematical background
  - Transform the control structure from mathematic equations to “C”
  - Study the core and peripherals, configure the peripherals
  - Tune the algorithm parameters (constants of PI-regulators, position estimation observers parameters)
  - Build the application in a reasonable time
- Tools that help speed-up the application development
  - Embedded Software and Motor Control Libraries
    - Tested, documented,
    - Optimized for performance
  - FreeMASTER real-time monitoring software, for application tuning

Motor control solutions

Current Motor Control MCU Offerings

freescale Digital Signal Controllers

Freescale DSC

Dedicated High Performance Motor Control
- Fractional Arithmetic, Parallel Processing, Optimized cost and performance for advanced motor control

Example: Most advanced 3ph Sensorless VOC, High and Low Speed Optimizations

ASP under $1

MC56F84xx Dual motors
MC56F84xx
MC56F823x
MC56F80xx

Vybrid

Real time control enabled with inclusion of ARM Cortex M4
- First available broad-market MPU that integrates ARM Cortex-A5 and Cortex-M4!

Example: Sensored or Sensorless Sinosoidal BLDC/PMSM VOC/FOC

Vybrid < $5

Vybrid Rich Apps in Real Time

Kinetis K Advanced Motor Control

Advanced motor control while multi-tasking on the most popular ecosystem in the world
- MQX RTOS and motor control, Scalability for any application, DSP instructions, Floating Point, ARM ecosystem

Example: Sensored or Sensorless Sinosoidal BLDC/PMSM VOC/FOC

Kinetis X < $4
Kinetis K < $2
Kinetis E < $1.5

Kinetis K 72-120MHz General Purpose

Kinetis K, Kinetis E, Kinetis L Basic Motor Control

General Purpose Motor Control
- Broad portfolio, incredible scalability, exceptional ecosystem

Example: : Suitable for low dynamic sensorless PMSM sinusoidal drives

ASP under $1

Kinetis L Low Power General Purpose
Kinetis K 50MHz General Purpose
Kinetis E 5V drive, robust

S08P– 8-bit

S08 Family – Entry Level Motor Control
- 5V drive, Robust EMC/EMI, Low Cost

Example: Sensored, Sensorless Trapezoidal BLDC

ASP under $1

S08PL
S08PT
S08PA

freescale™

New Motor Control Offerings

freescale
Digital Signal Controllers

ASP under $1

Freescale DSC

Positioning: Dedicated High Performance Motor Control
Key Message: Fractional Arithmetic, Parallel Processing, Optimized cost and performance for advanced motor control
Example: Most advanced 3ph Sensorless VOC, High and Low Speed Optimizations

MC56F827x

MC56F823x

Kinetis < $4

Kinetis K < $2

Kinetis E < $1.5

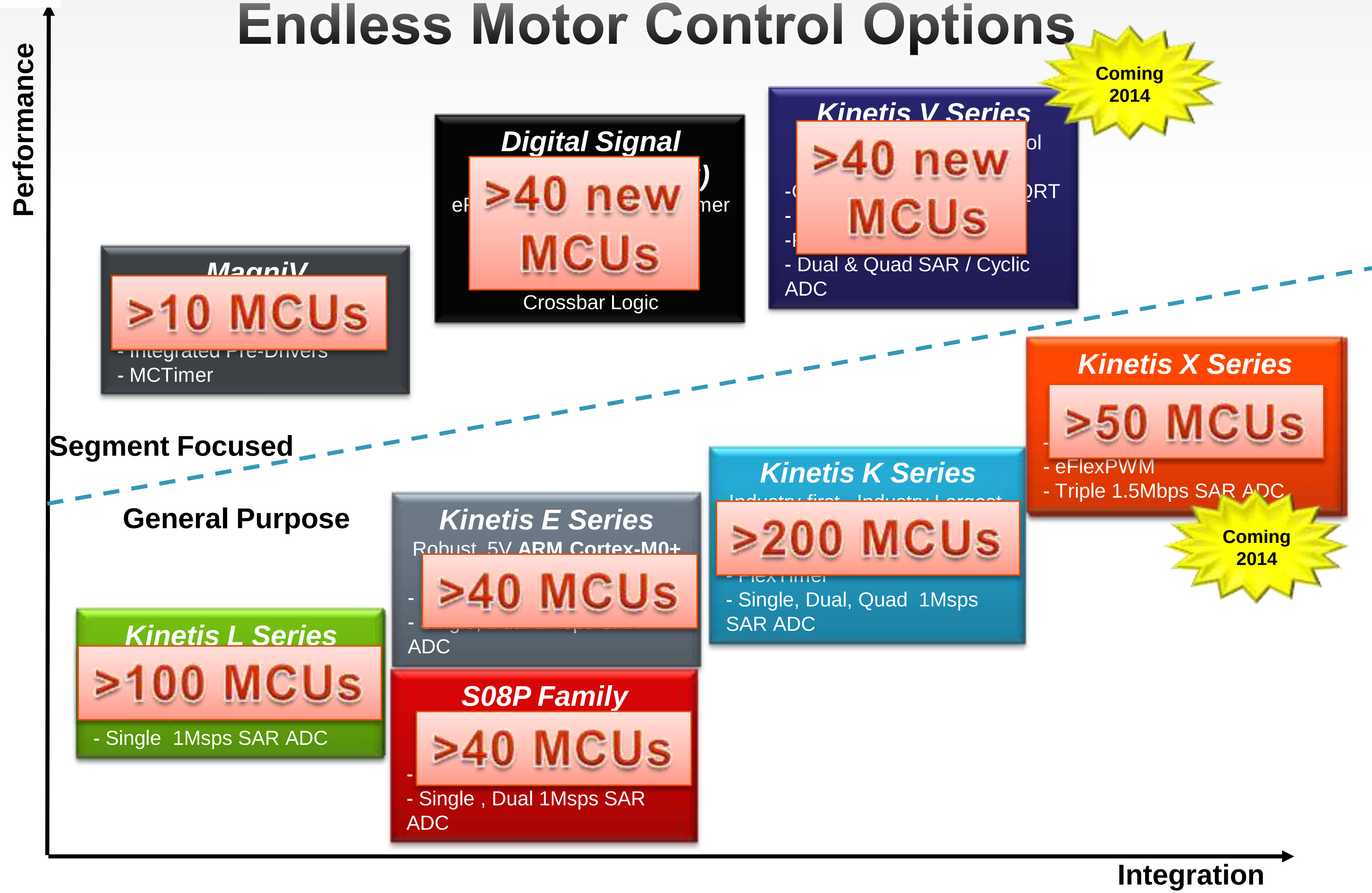
Kinetis V Specialized Motor Control Family

Positioning: Advanced motor control while multi-tasking on the most popular ecosystem in the world
Key Message: MQX RTOS and motor control, Scalability for any application, DSP instructions, Floating Point, ARM ecosystem
Example: Sensored or Sensorless Sinosoidal BLDC/PMSM VOC/FOC

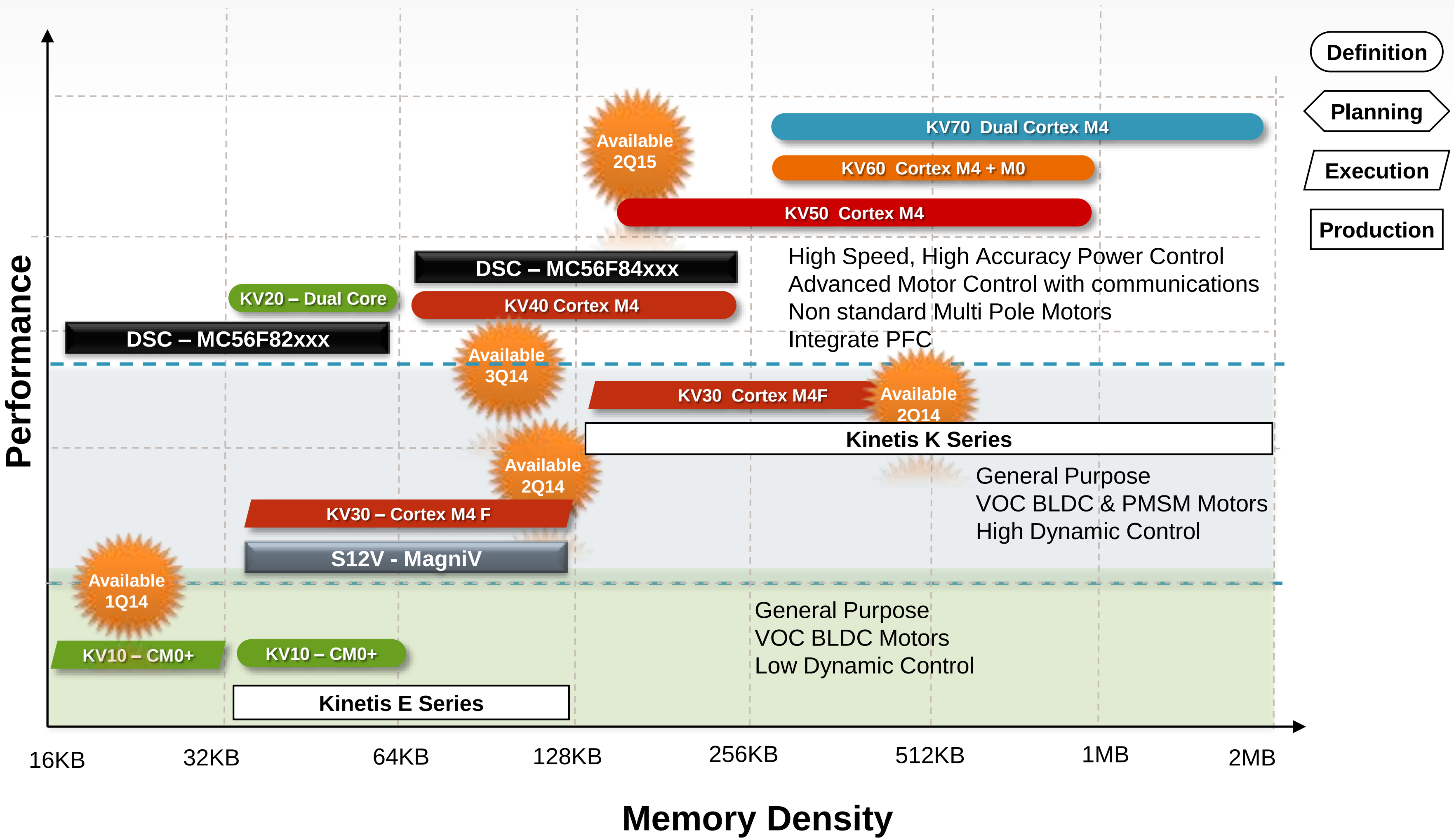
Kinetis V
75 MHz
Cortex M0+

Kinetis V
100 - 120MHz
Cortex M4

Endless Motor Control Options



Motor Control Roadmap



MC56F827xx (64kB Flash, 50/100MHz)

Key Features:

Core

• 56800EX @ 50/100MHz supporting fractional arithmetic with 4 accumulators, 8 cycle pipeline, separate program and data memory maps for parallel moves, single cycle math instructions, nested looping, and superfast interrupts that far outpace any competitive core on the market.

System

• Inter-module crossbar directly connecting any input and/or output with flexibility for additional logic functions (AND/OR/XOR/NOR)

• DMA controller for reduced core intervention when shifting data from peripherals

• Memory resource protection unit to ease safety certification

Timers

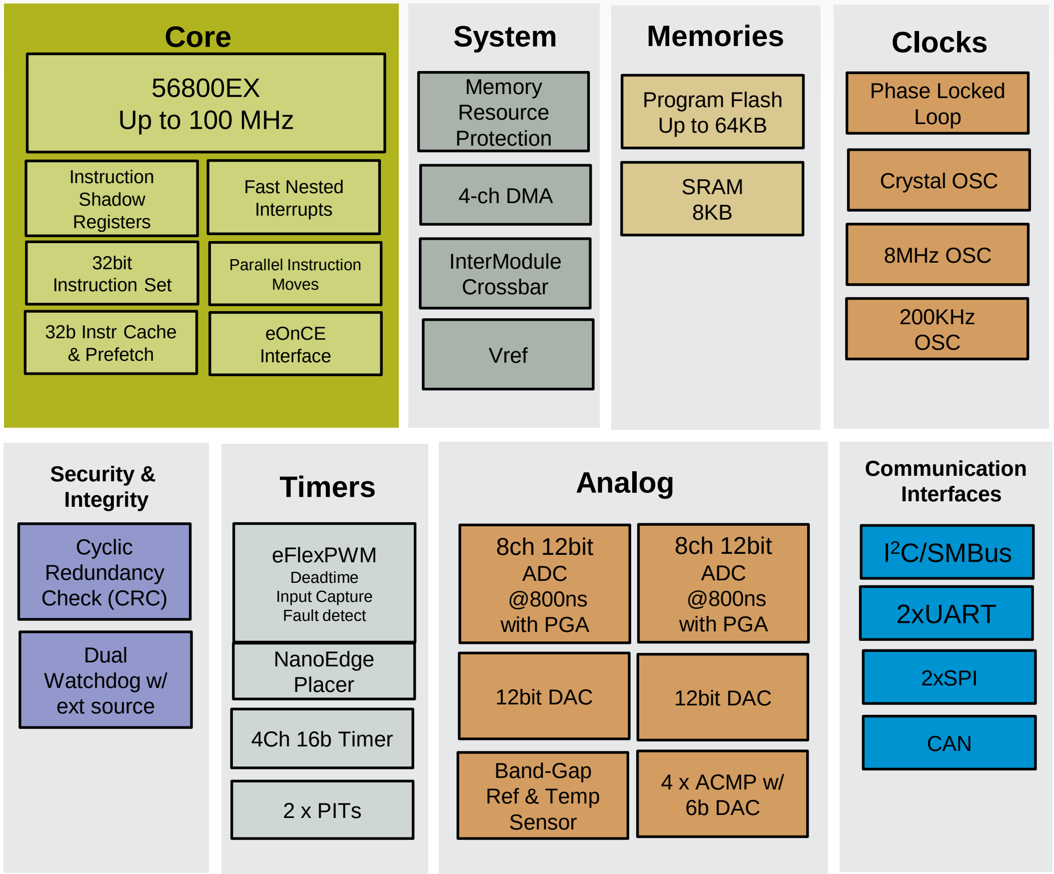
• eFlexPWM – Freescale's most advance timer for Digtial Power Conversion, up to 8ch and 312 pico-sec resolution, 4 independent time bases, with half cycle reloads for increased flexibility, automatic complimentary mode for ease of use and best in class performance

Analog

- 2x12-bit high-speed ADCs each with 800ns conversion rates
- 4 analog comparators with integrated 6-bit DACs that can enable emergency shutdown of the PWMs
- Integrated PGAs to increase the accuracy of ADC conversions on small voltages and currents

Power Consumption:

- Best in class Power Consumption – 50% better than nearest competitor



Others: 5-volt tolerant I/O for cost-effective board design

Packages: 32QFN (5x5), 32LQFP, 48LQFP, 64LQFP

Temperature: -40 to +105C across all packages, with -40 to +125C option on 64LQFP

freescale™

MC56F823xx (32kB Flash, 50MHz)

Key Features:

Core

• 56800EX @ 50MHz supporting fractional arithmetic with 4 accumulators, 8 cycle pipeline, separate program and data memory maps for parallel moves, single cycle math instructions, nested looping, and superfast interrupts that far outpace any competitive core on the market.

System

• Inter-module crossbar directly connecting any input and/or output with flexibility for additional logic functions (AND/OR/XOR/NOR)
• DMA controller for reduced core intervention when shifting data from peripherals
• Memory resource protection unit to ease safety certification

Timers

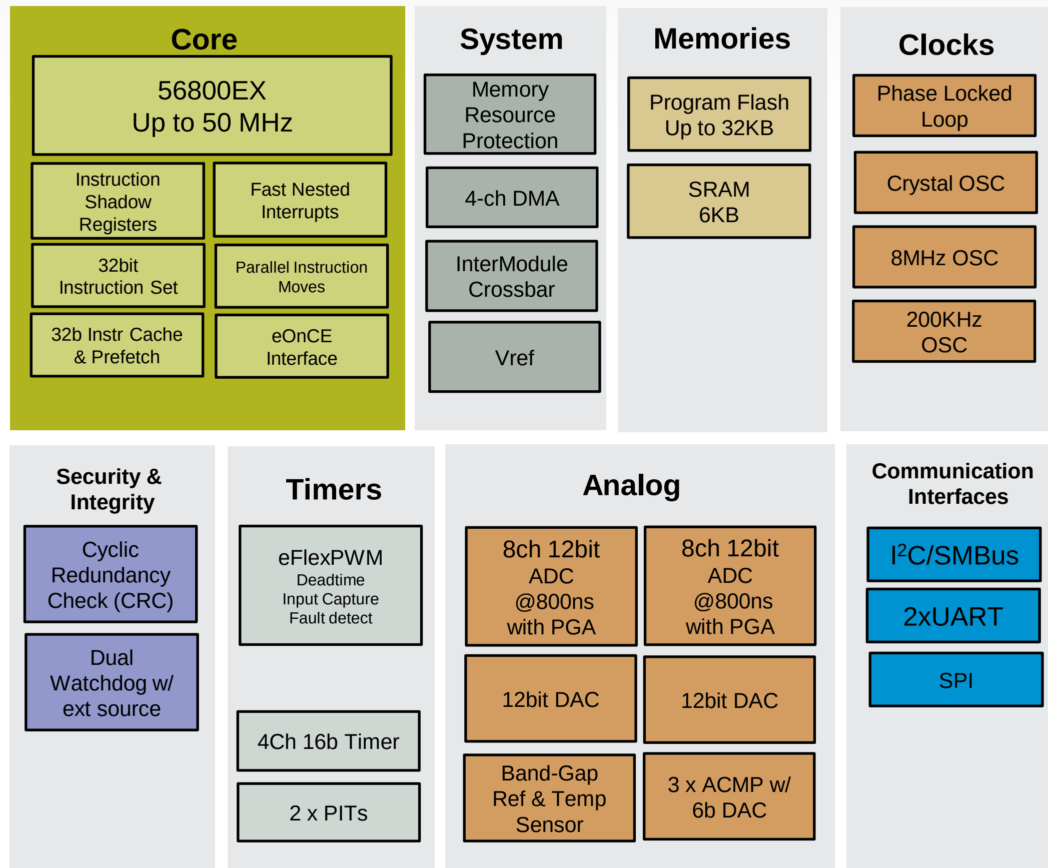
• eFlexPWM – Freescale's most advance timer for Digtial Power Conversion, up to 8ch and 312 pico-sec resolution, 4 independent time bases, with half cycle reloads for increased flexibility, automatic complimentary mode for ease of use and best in class performance

Analog

- 2x12-bit high-speed ADCs each with 800ns conversion rates
- 4 analog comparators with integrated 6-bit DACs that can enable emergency shutdown of the PWMs
- Integrated PGAs to increase the accuracy of ADC conversions on small voltages and currents

Power Consumption:

- Best in class Power Consumption – 50% better than nearest competitor



Others: 5-volt tolerant I/O for cost-effective board design

Packages: 32QFN (5x5), 32LQFP, 48LQFP

Temperature: -40 to +105C across all packages

freescale™

Freescale DSC

MC56F84xxx (256kB Flash, 100MHz)

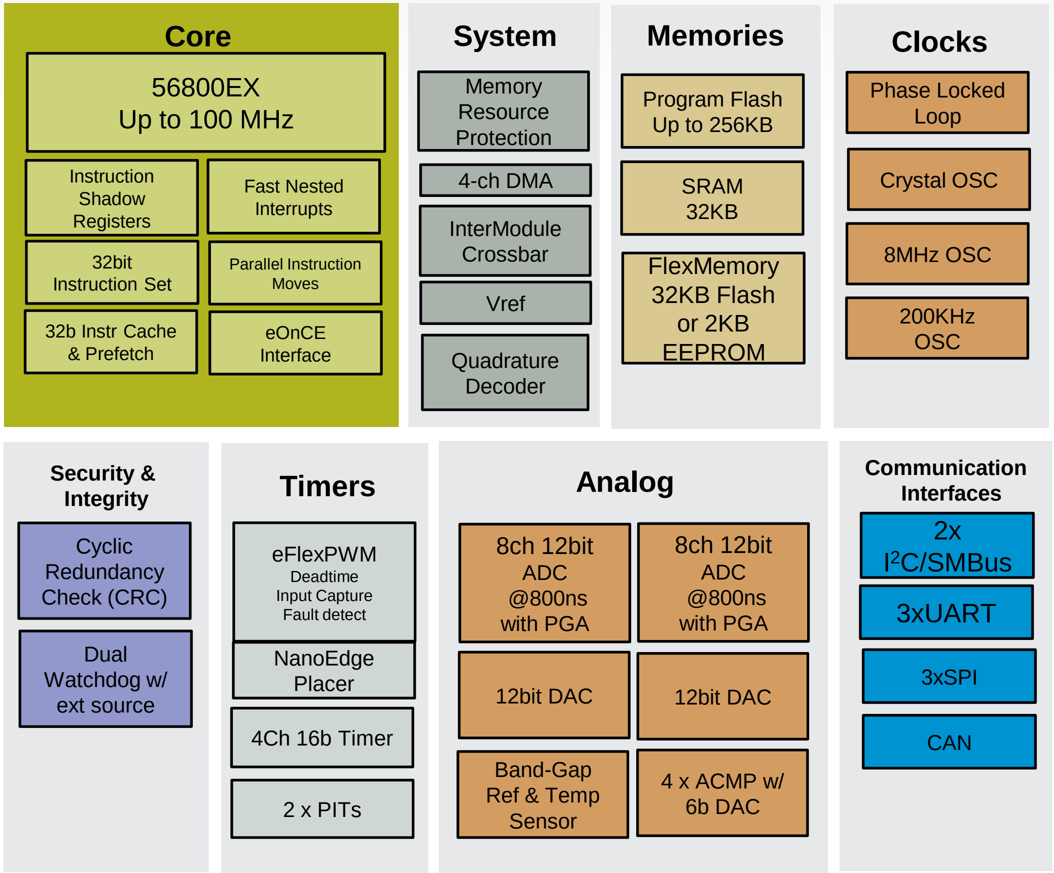
Key Features:

Core
• 56800EX @ 100MHz supporting fractional arithmetic with 4 accumulators, 8 cycle pipeline, separate program and data memory maps for parallel moves, single cycle math instructions, nested looping, and superfast interrupts that far outpace any competitive core on the market.

System
• Inter-module crossbar directly connecting any input and/or output with flexibility for additional logic functions (AND/OR/XOR/NOR)
• DMA controller for reduced core intervention when shifting data from peripherals
• Memory resource protection unit to ease safety certification

Timers
• eFlexPWM – Freescale's most advance timer for Digtial Power Conversion, up to 8ch and 312 pico-sec resolution, 4 independent time bases, with half cycle reloads for increased flexibility, automatic complimentary mode for ease of use and best in class performance

Analog
- 2x12-bit high-speed ADCs each with 800ns conversion rates
- 16 ch 16b SAR ADC that enables external sensors inputs and accurate system measurements
- 4 analog comparators with integrated 6-bit DACs that can enable emergency shutdown of the PWMs
- Integrated PGAs to increase the accuracy of ADC conversions on small voltages and currents



Others: 5-volt tolerant I/O for cost-effective board design
Freescale FlexMemory for simplified data storage

Packages: 48LQFP, 64LQFP, 80LQFP, 100LQFP

Temperature: -40 to +105C across all packages

KV10: 75MHz Cortex-M0+ 32KB Flash

Key Features:

Core/System
- 75MHz **Cortex-M0+** with 4ch DMA
  - Hardware Divide & SqrRoot
  - Bit Manipulation Engine

Memory
- 32KB Flash
- 8KB SRAM

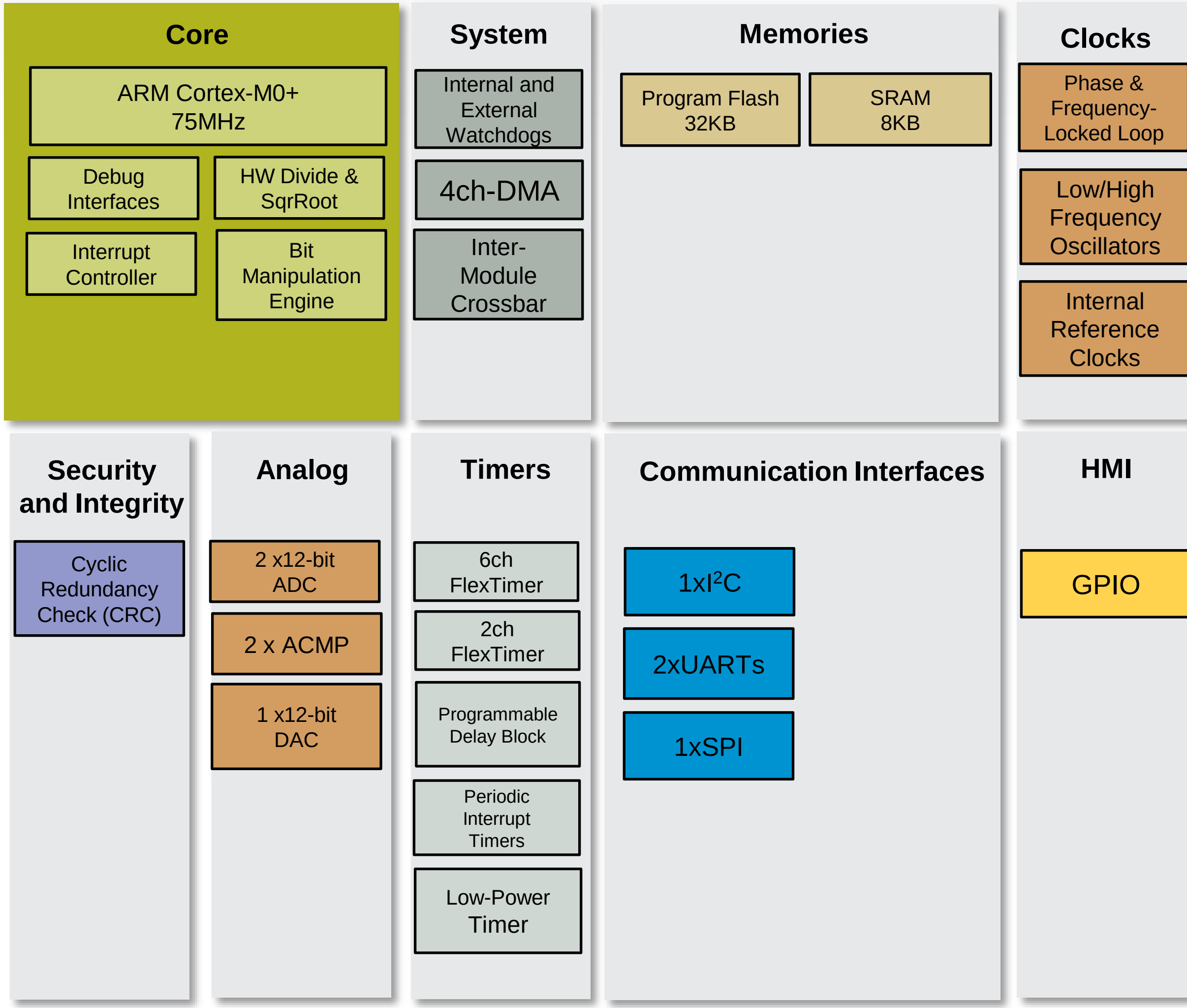
Communications
- Multiple serial ports

Analog
- 2 x 8ch 12-bit ADC
  - 1uS conversion time
- 1 x12-bit DAC
- 2 x ACMP w/ 6b DAC

Timers
- 1x6ch FlexTimer (PWM)
- 1x2ch FlexTimer (PWM/Quad Dec.)
- Programmable Delay Block

Others
- 32-bit CRC
- Intermodule Crossbar Switch
- Up to 35 I/Os
- 1.71V-3.6V; -40 to 105°C

Packages
32QFN, 32LQFP, 48LQFP



freescale™

KV10s: 100MHz Cortex M4 64K Flash

Key Features:

Core/System
- Cortex-M4 @ 100MHz

Memory
- 64KB Flash,
- 16KB SRAM

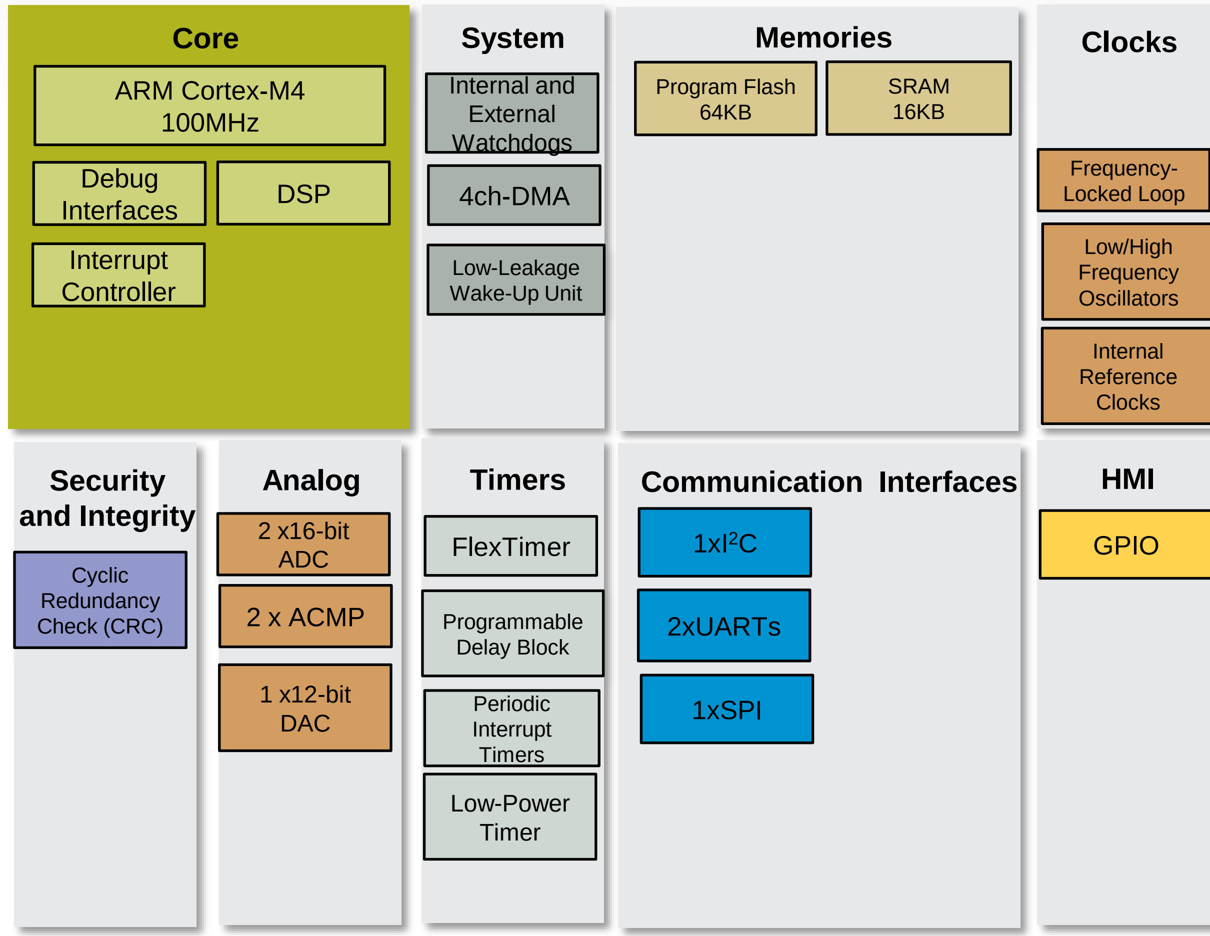
Communications
- Multiple serial ports

Analog
- 2 x16-bit ADC
- 1 x12-bit DAC
- 2 x ACMP

Timers
- 1x6ch FTM (PWM)
- 2x2ch FTM (PWM/Quad Dec.)
- Low Power Timer

Others
- Up to TBD I/Os
- 6 high-drive I/Os (20mA) – SPI/I2C
- 1.71V-3.6V; -40 to 105oC

Packages
32QFN, 48LQFP, 64LQFP



freescale™

KV10s: 100MHz Cortex M4 128K Flash

Key Features:

Core/System
- Cortex-M4 @ 100MHz

Memory
- 128KB Flash,
- 16KB SRAM

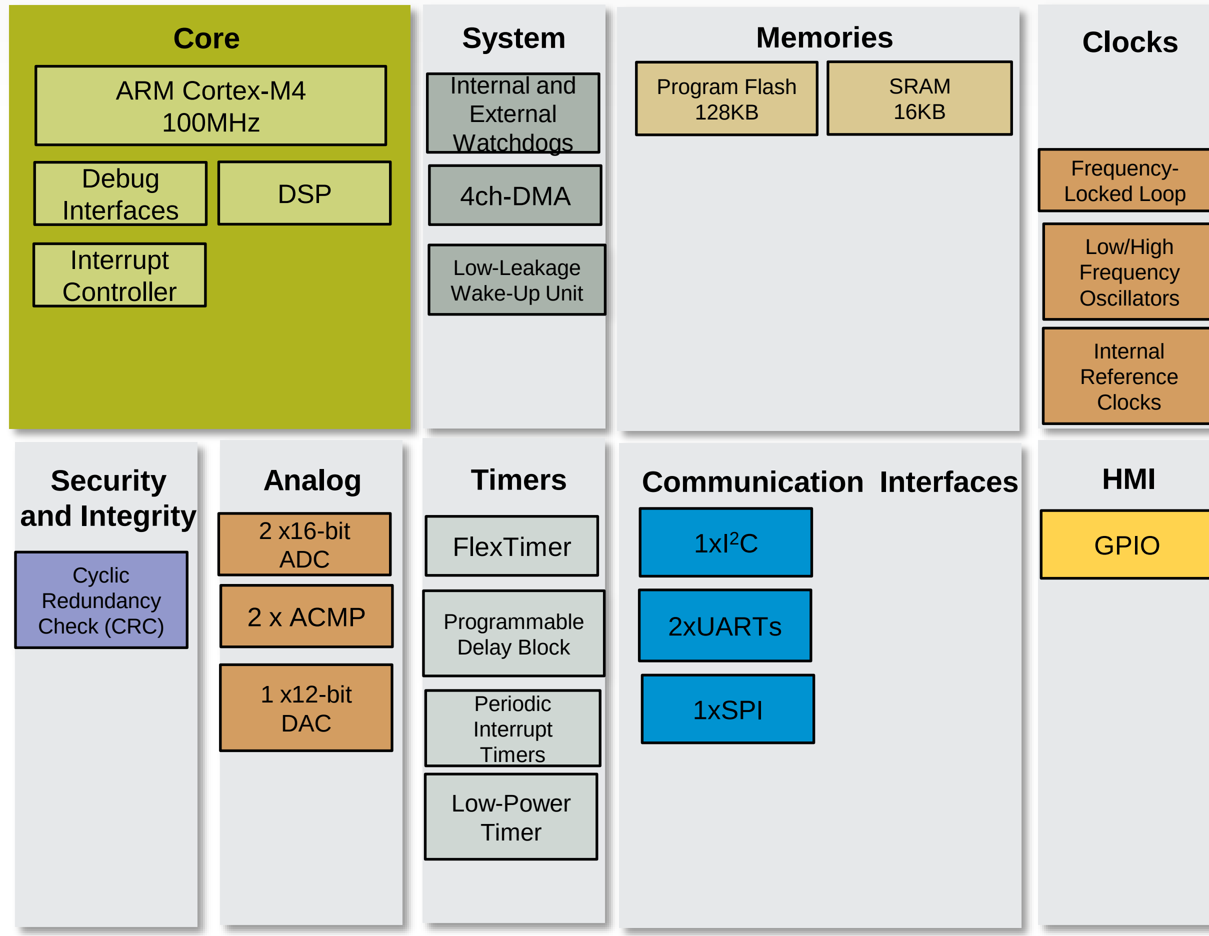
Communications
- Multiple serial ports

Analog
- 2 x16-bit ADC
- 1 x12-bit DAC
- 2 x ACMP

Timers
- 1x6ch FTM (PWM)
- 2x2ch FTM (PWM/Quad Dec.)
- Low Power Timer

Others
- Up to TBD I/Os
- 6 high-drive I/Os (20mA) – SPI/I2C
- 1.71V-3.6V; -40 to 105oC

Packages
32QFN, 48LQFP, 64LQFP



KV10s: 100MHz Cortex M4 FPU 128K Flash

Key Features:

Core/System
- Cortex-M4 @ 100MHz / FPU

Memory
- 128KB Flash,
- 24KB SRAM

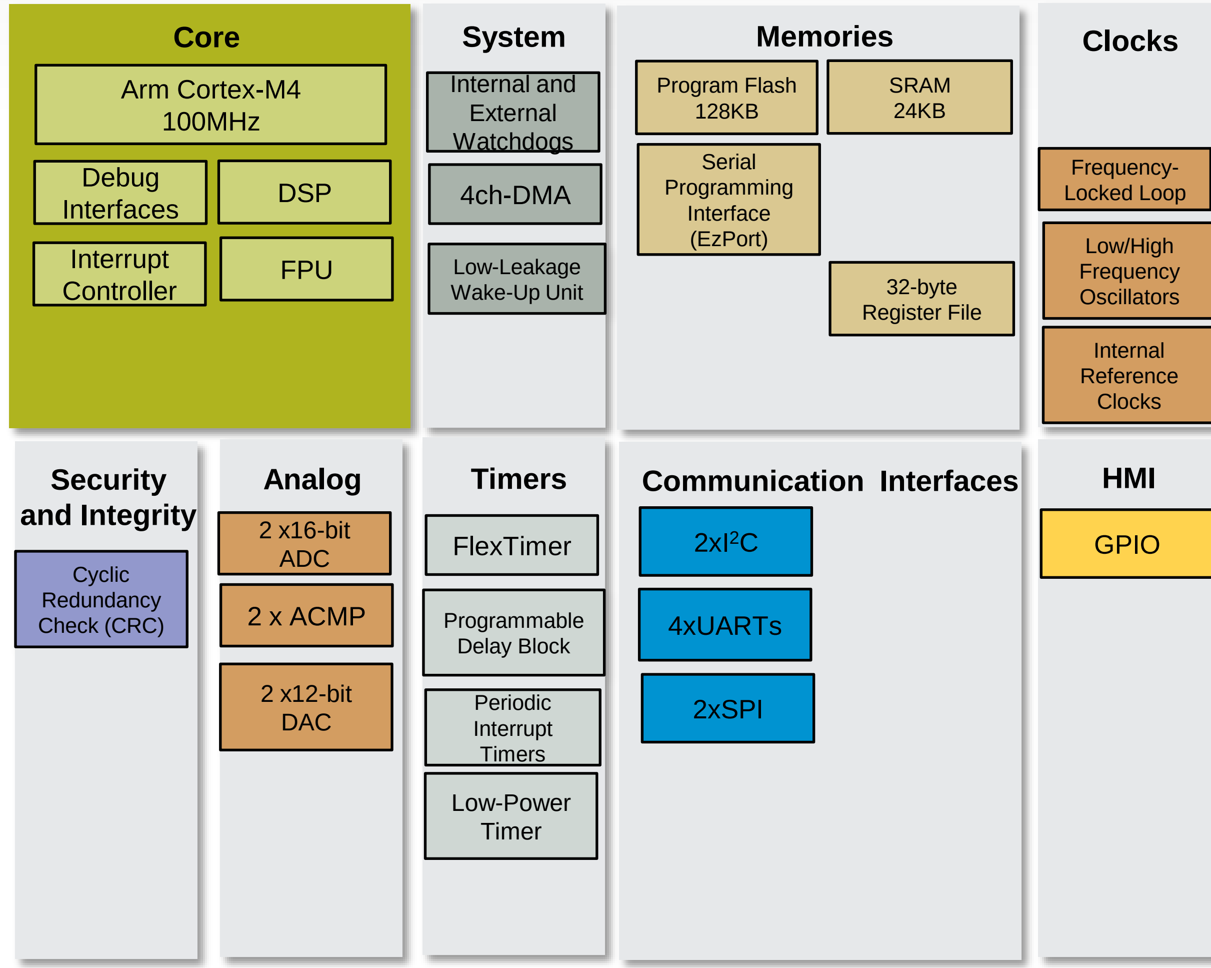
Communications
- Multiple serial ports

Analog
- 2 x16-bit ADC
- 1 x12-bit DAC
- 2 x ACMP

Timers
- 1x8ch FTM (PWM)
- 2x2ch FTM (PWM/Quad Dec.)
- Low Power Timer

Others
- Up to TBD I/Os
- 6 high-drive I/Os (20mA) – SPI/I2C
- 1.71V-3.6V; -40 to 105oC

Packages
64LQFP, 100LQFP



Standard Feature   Optional Feature

V10s –120MHz Cortex M4 FPU 512KB/256KB Flash

Key Features:

Core/System
- Cortex-M4 @ 120MHz / FPU

Memory
- up to 512KB Flash,
- up to 128KB SRAM
- FlexBus (External Bus Interface)

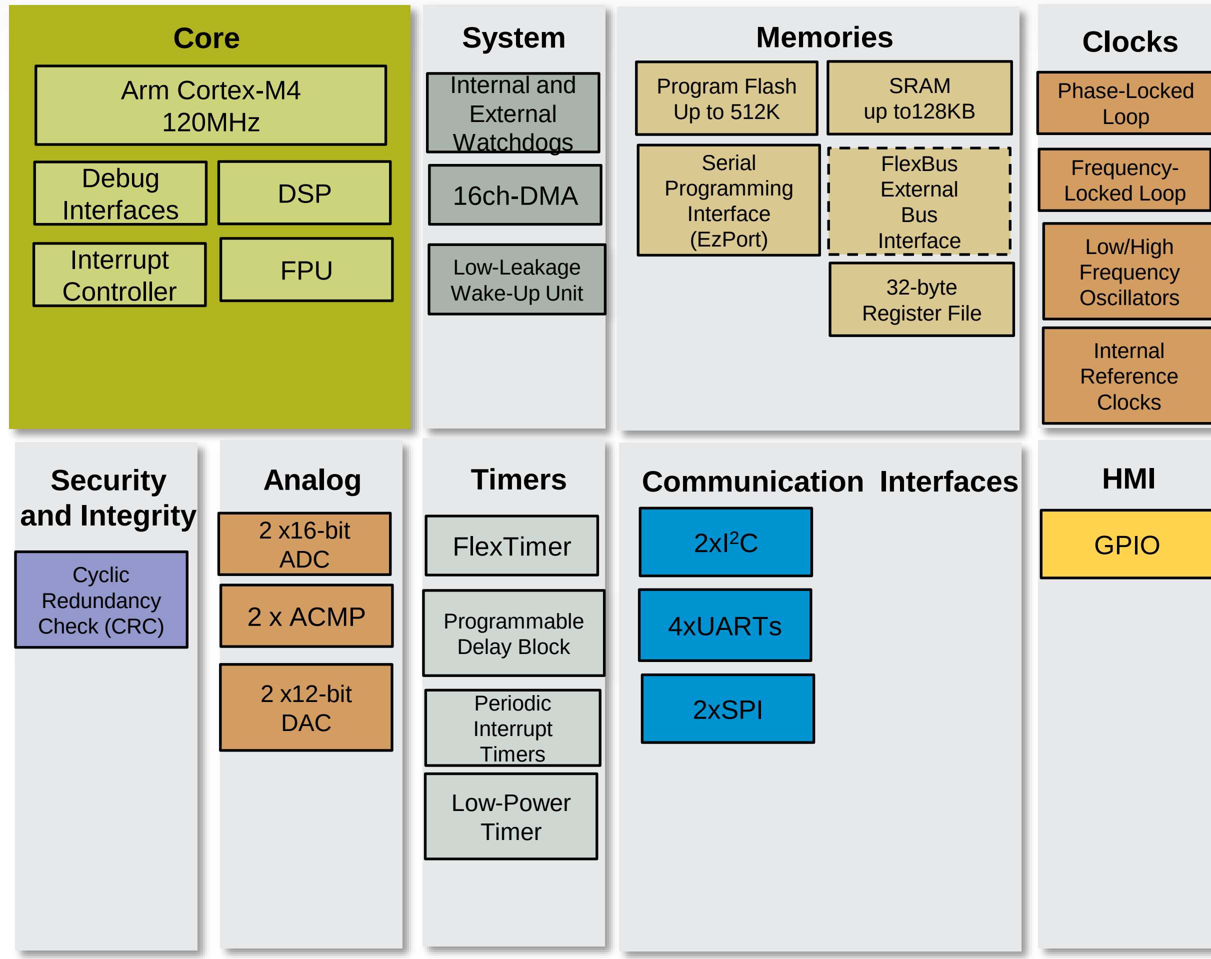
Communications
- Multiple serial ports

Analog
- 2 x16-bit ADC
- Up to 2 x12-bit DAC
- 2 x ACMP

Timers
- up to 2x8ch FTM (PWM)
- 2x2ch FTM (PWM/Quad Dec.)
- Low Power Timer

Others
- Up to TBD I/Os
- 6 high-drive I/Os (20mA) – SPI/I2C
- 1.71V-3.6V; -40 to 105oC

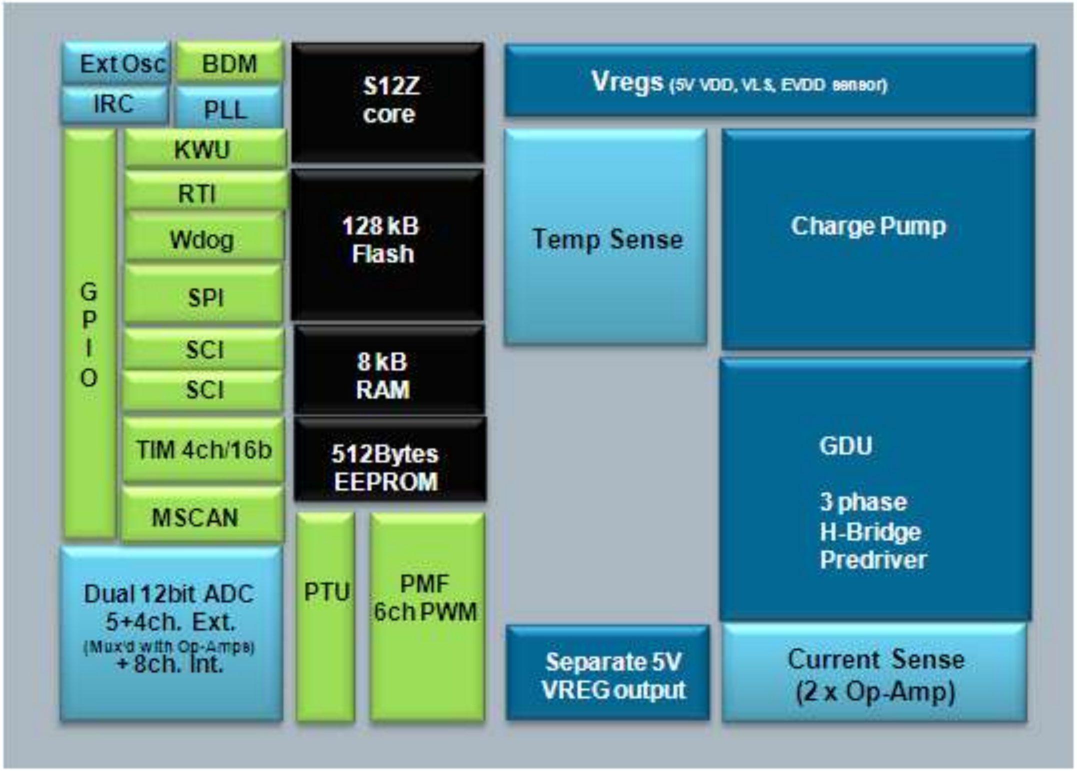
Packages
64LQFP, 100LQFP



S12ZVMC128

- **Target applications:**
  - BLDC motor control
  - DC motor control

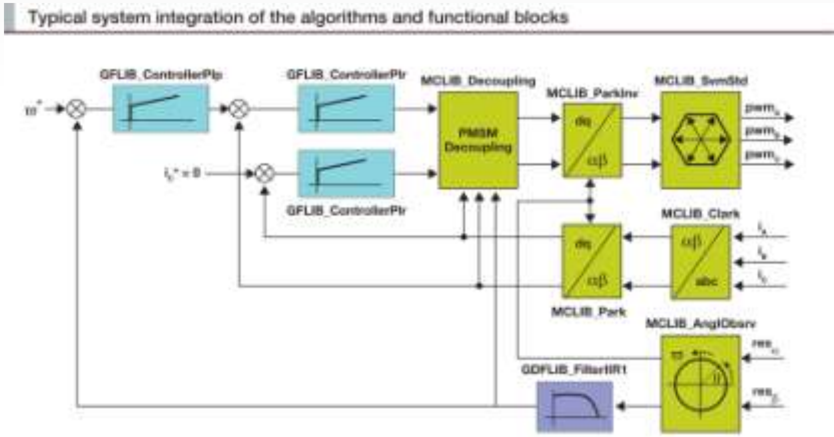
- **Key Features:**
  - S12Z CPU @ 50MHz bus speed
  - Embedded VREG
  - Separate 2nd VREG (to power external CAN phy)
  - Embedded GDU for 3ph BLDC
  - Embedded EE
  - 1x MSCAN controller
  - 2xSCI, 1xSPI
  - Dual 12bit ADC, synch with PWM
  - 20mA/5V EVDD sensor supply pin
  - 2x Op-amp for current sense (each needs 2 pins mux'd with ADC inputs)
  - 64LQFP-EP 10x10/0.5mm

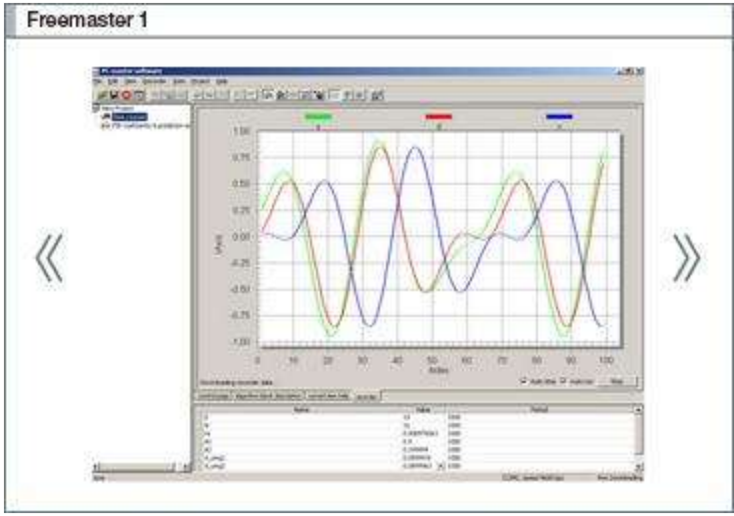


freescale™

Software Enablement – current offerings

- **Embedded Software Libraries**
  - Optimized fractional math, filtering and control functions
- **FreeMASTER**
  - Real-time debug monitor and data visualization tool
- **Motor Control Toolbox**
  - Automatic software generation for motor control applications (Qorivva and MPC5 MCUs)





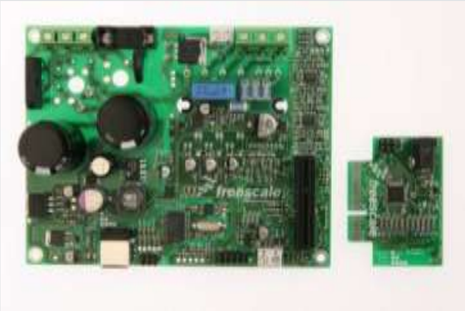
Hardware Enablement – current offerings

- **Low-voltage Tower motor control**
  - 3-phase, for low voltage BLDC and PMSM motors
- **Reference designs**
  - High and low voltage power stages
  - PMSM, BLDC, ACIM, SR motors

Motor Control H/W

Tower Power Stage – TWR-MC-LV3PH

- Target use: Motor Control Techniques Development
- Input voltage 12-24 VDC
- Output Current 5-10 Amps
- Compatible with FSL TWR cards
- Status: on FSL stock



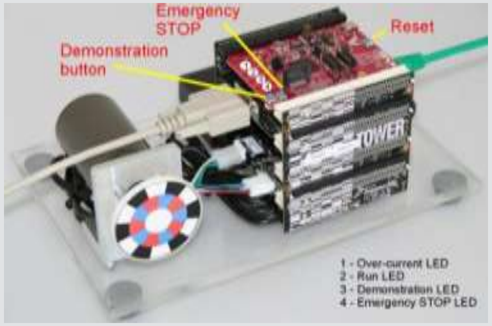
3-ph BLDC/PMSM High Voltage Drive

- Target use: appliance and industrial drives
- Input Voltage 115-230Vac, 50/60Hz
- Output Power 1kW
- Supporting: S08MP16, 56F80xx, 56F82xx, 56F84xx, K40
- Status: manufactured in Roznov, productization in 2013

3-in-1 Power Stage

- Target use: AirCon, washers
- Input Voltage 115-230Vac, 50/60Hz
- Output Power: Two Drives up to 1500W and up to 500W
- Using Nevis Daughter Card
- Status: prototype designed, handed over to A/P

Kinetis based Motor Control



BLDC Sensor-less Drive with MQX on Kinetis K60

- Sensor-less 3-phase trapezoidal BLDC motor control
- Motor Control algorithm running under MQX
- Control over web server or FreeMASTER
- Running on a Tower kit

Dual Sinusoidal PMSM for Industrial drive on K70

- Sensorless Sinusoidal FOC control algorithm with Encoder
- Targets industrial drives
- Running on Tower Kit with added dual motor control support

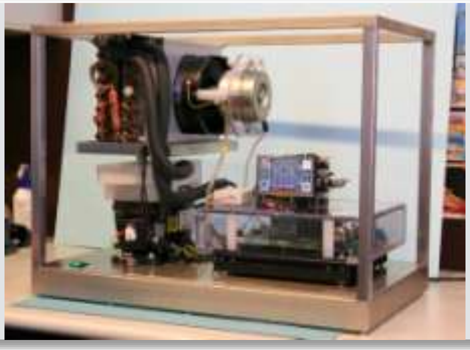
Sensorless PMSM on Kinetis K60

- Sensorless Sinusoidal FOC Drive
- Position and speed detection using dq back-emf observer and tracking observer
- Running on a Tower kit

DSC based Motor Control

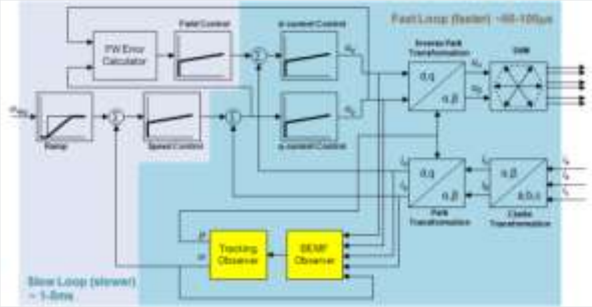
VF PMSM Compressor

- Sensorless sinusoidal FOC for compressor PMSM motor
- Control using K70 HMI with graphic touch display
- demo and s/w available



VF 3 in 1 Motor Control for AirCon with 56F84xxx

- 1.5 KW output power, support sensor-less PMSM motor control for both outdoor fan and compressor with FOC algorithm
- Support digital PFC (average current control)
- Demo, h/w and s/w available



Sensorless PMSM for fans on 56F82xxx

- New application being developed for sensorless sinusoidal PMSM FOC
- Includes Tuning Wizard for easy use
- Prototype for pre-programmed MC device

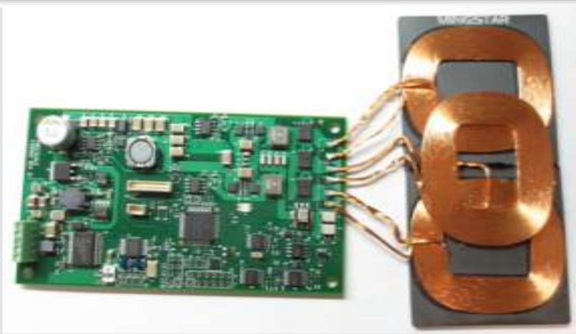
Power Conversion & Wireless Charging

200W SMPS with MC56F8013 and MC56F8257

- Primary Side: Two Phase Interleaved PFC
- Secondary Side: Half Bridge LLC Resonant Converter with Synchronous Rectification for 12V output
- Additional Synchronous Buck Converter for 5V output

Solar Micro-Inverter with MC56F8257

- 1-phase 200W non-isolated Micro Solar Inverter
- Includes Interleaved CrCM step-up converter with P&O and RCC MPPT, Sine inverter and output filter
- Project done in cooperation with Future distributor
- Project in finalization



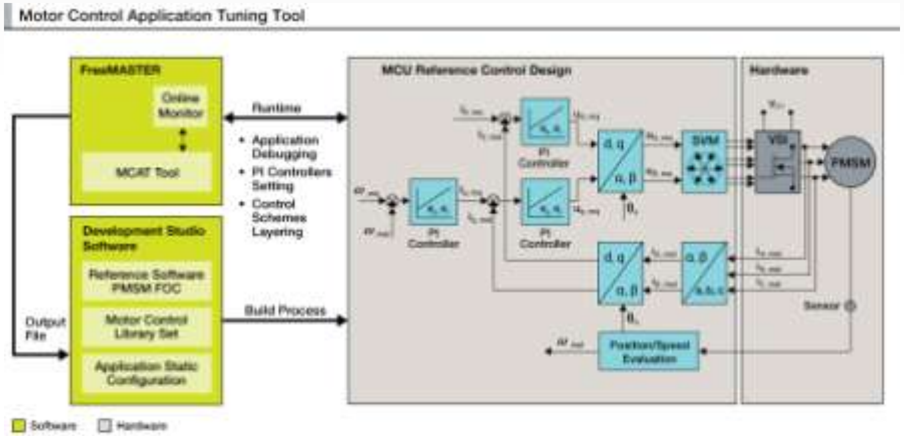
5 W A13 Automotive Wireless Charger

- Request to enhance the AVID/Fulton WC transmitter
- Add digital modulation / demodulation, Touch, CAN
- Project in definition

freescale™

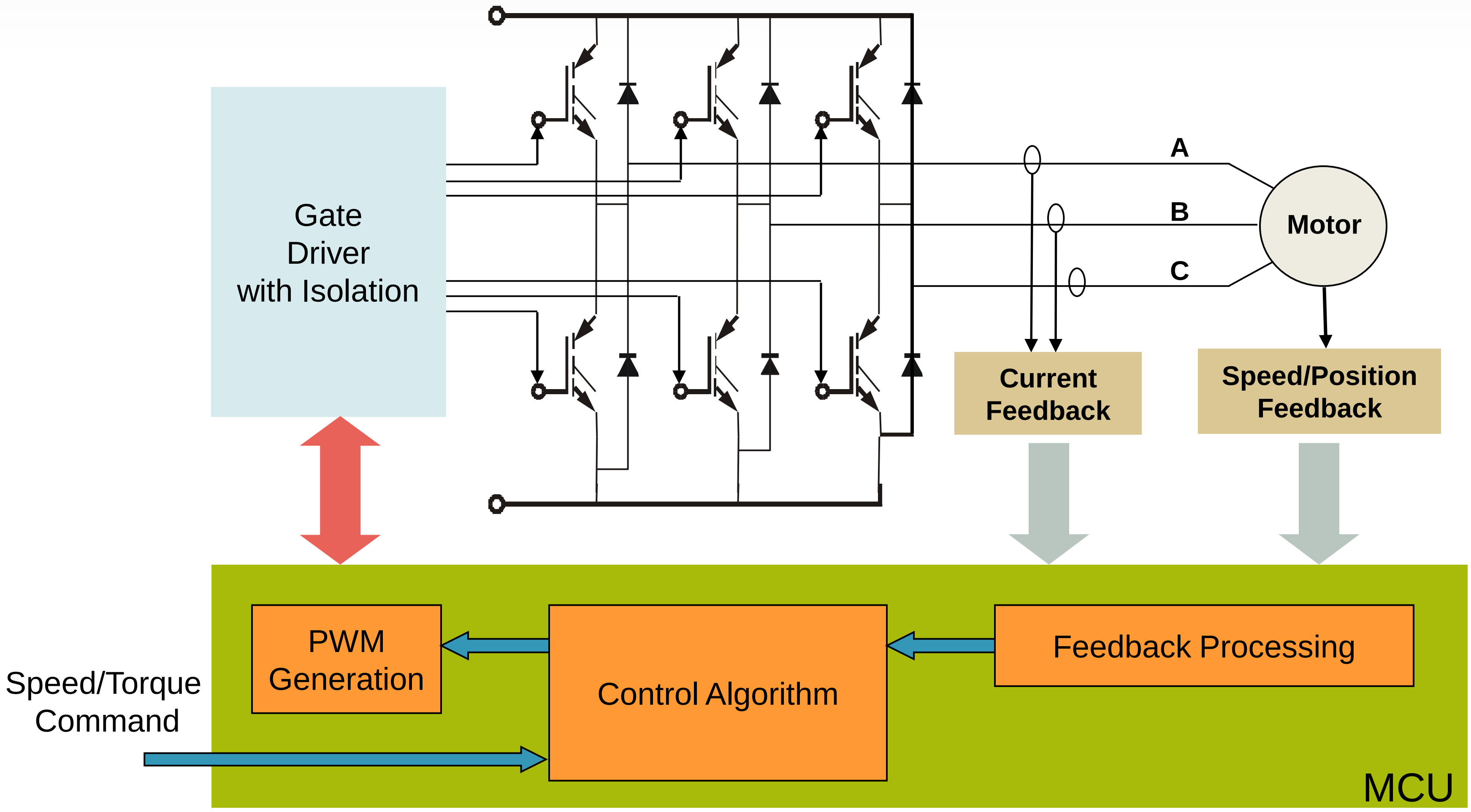
New enablement

- **MCAT**
  - Motor control application tuning tool
- **Floating-point control libraries**
  - For Cortex M4 FPU-enabled devices
- **Motor control toolbox**
  - For Kinetis V
- **Hardware divide and square root on the Kinetis V M0+**



Motor control technology

MCU Requirements for Motor Control Applications



MCU Requirements for Motor Control Applications

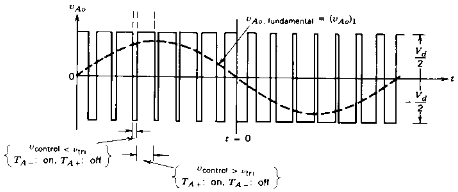
- **ADC Module**
  - We need to measure DC Bus voltage, Back-EM voltage, phase currents, DC Bus current, heatsink temperature
- **PWM module**
  - We need to generate 1 up 8 PWM according to motor type
- **Timer/Quadrature decoder**
  - We need to measure speed and rotor position from different sensors (hall sensors, quadrature encoder, tacho generator, sin/cos interface, etc.)
- **Built-in Comparator**
  - We need to detect fault conditions (over-current, over-voltage)
  - Allows to eliminate external comparators
  - Build in DAC allows SW control of fault level
- **User interface**
  - Communication interfaces, if required (SCI, SPI, CAN, I2C)
  - GPIO pins

Kinetis

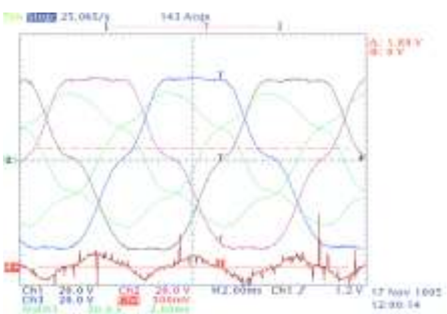
Motor Control Peripherals

PWM Signal Generation

- **Sinusoidal Controlled Motors**
  - AC Induction Motor, PM Synchronous Motor
  - PWM Requirements
    - Synchronized PWM Update
    - Complementary Signal Generation
    - Dead-time insertion
    - Fault Control



- **Block Commutated Motors**
  - BLDC Motor, SR Motor, Stepper Motor
    - Commutation is asynchronous to PWM generation
    - Software Control
    - Mask/Swap (Invert) Control
    - Fault Control

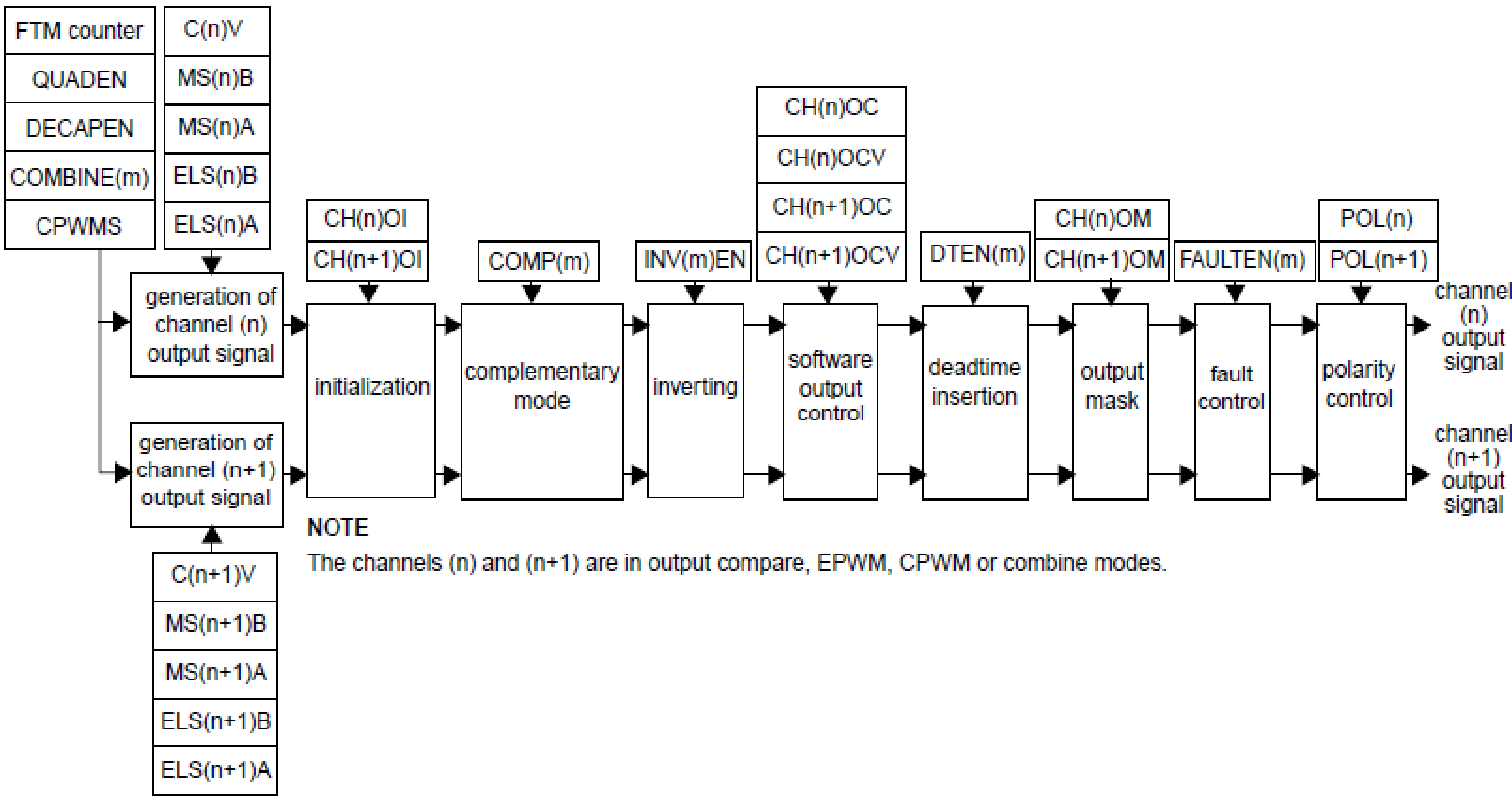


FlexTimer Module

- FTM source clock is selectable with prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128
- FTM has a 16-bit counter
- **2 up to 8 channels** (inputs/outputs)
- The **counting** can be **up** or **up-down**
- Each channel can be configured for input capture, output compare, or
- Input filter can be selected for some channels
- New **combined mode to generate a PWM signal** (with independent control of both edges of PWM signal)
- **Complementary outputs, include the deadtime insertion**
- **Software control of PWM outputs**
- Up to **4 fault inputs** for global fault control
- **The polarity of each channel is configurable**
- The generation of an interrupt per channel input capture/compare, counter overflow, at fault condition
- **Synchronized loading of write buffered FTM registers**
- **Write protection for critical registers**
- Backwards compatible with TPM
- **Dual edge capture for pulse and period width measurement**
- **Quadrature decoder with input filters, relative position counting and interrupt on**
- Position count or capture of position count on external event

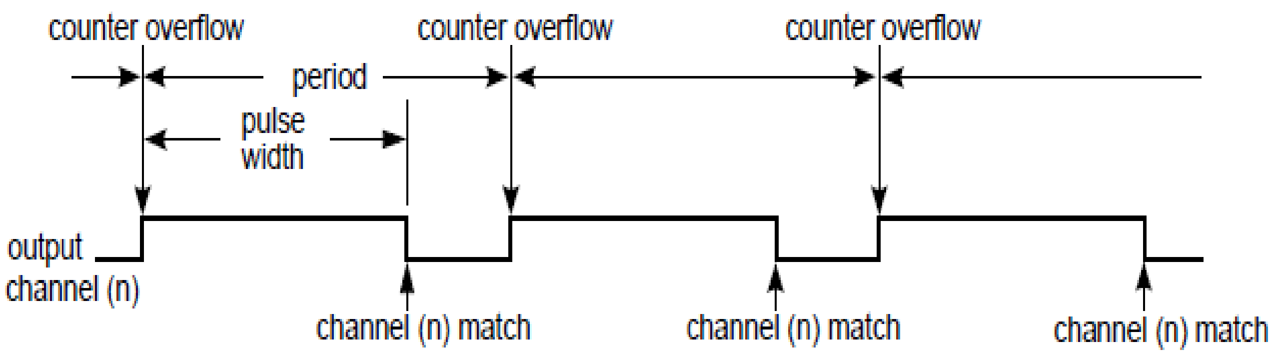
freescale™

FlexTimer Module Diagram



FTM Counting Modes

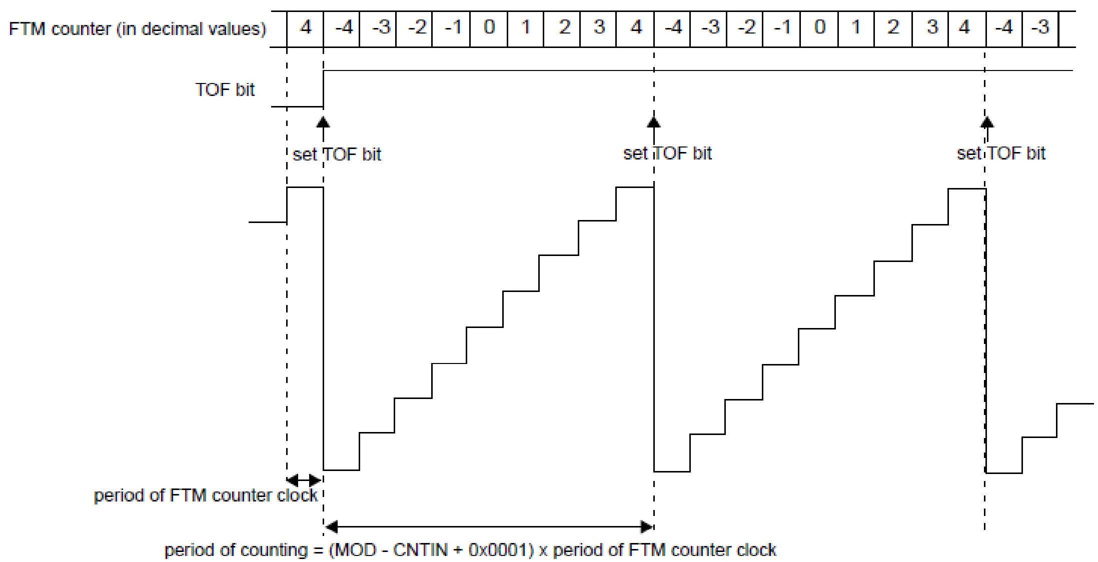
- **Edge Aligned PWM**
  - The frequency is defined by FTMx_MOD register
  - The duty cycle is defined by FTMx_CnV register



FTM Counting Modes

- **Edge Aligned PWM**
  - If FTMx_CNTIN register is set to non zero value, then the frequency defined as

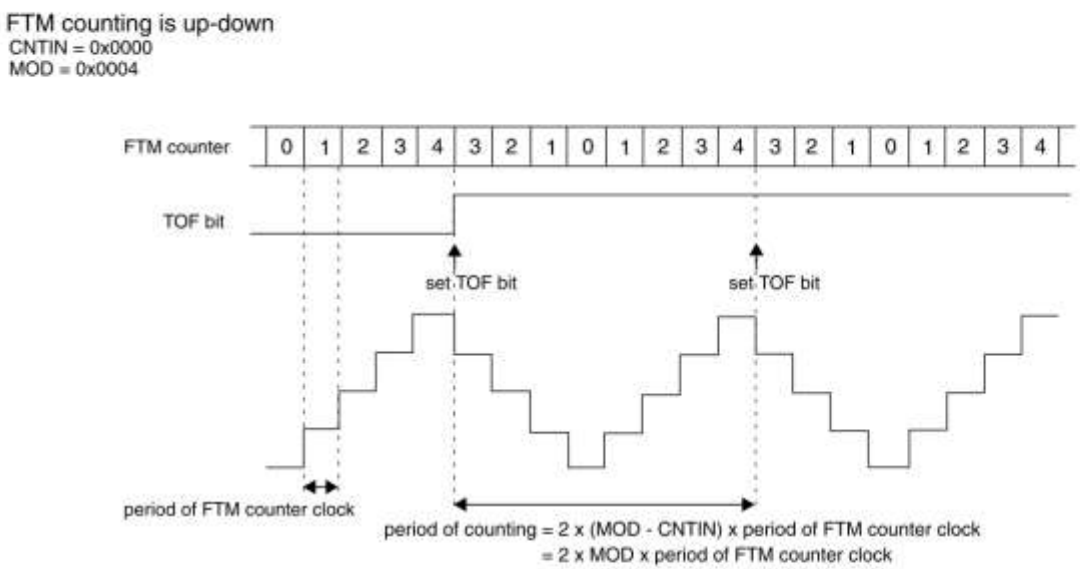
FTMx_MOD - FTMx_CNTIN +1



freescale™

FTM Counting Modes

- **Center Aligned PWM**
  - The frequency is defined by FTMx_MOD register
  - The duty cycle is defined by FTMx_CnV register

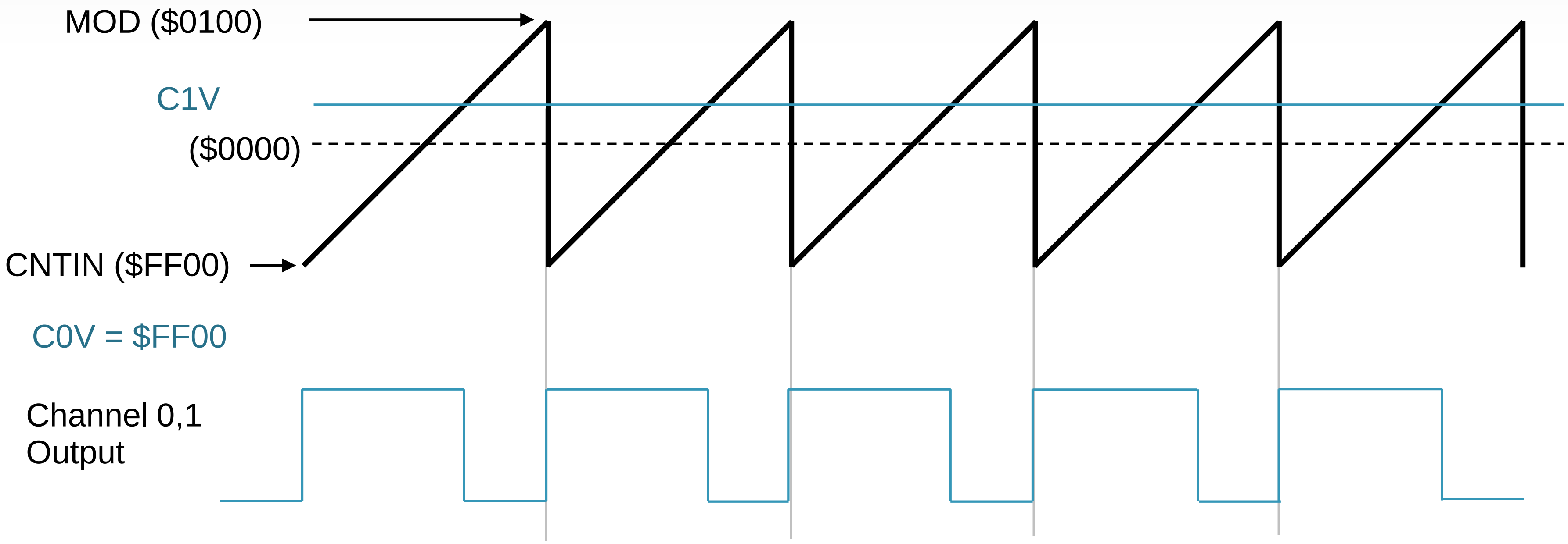


Note: *For Center Aligned PWM the FTMx_CNTIN has to be set to 0*

FTM Counting Modes

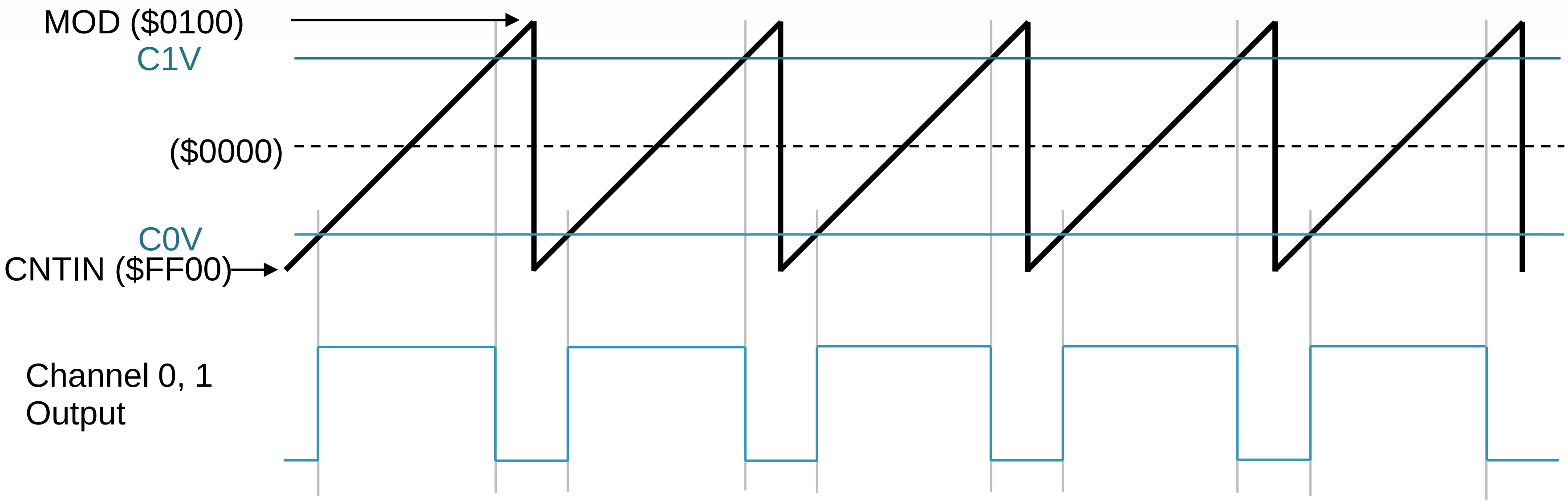
- **Combined PWM Mode**
  - Two FTM Channels are combined together to define one PWM signal
  - The channel n (FTMx_CnV register) defines rising edge of PWM signal
  - The channel (n+1) (FTMx_C(n+1)V register) defines falling edge of PWM signal
  - In independent mode both outputs generates two equal signals
  - In complementary mode both outputs generates two complementary signals

FlexTimer – Edge Aligned PWM Generation



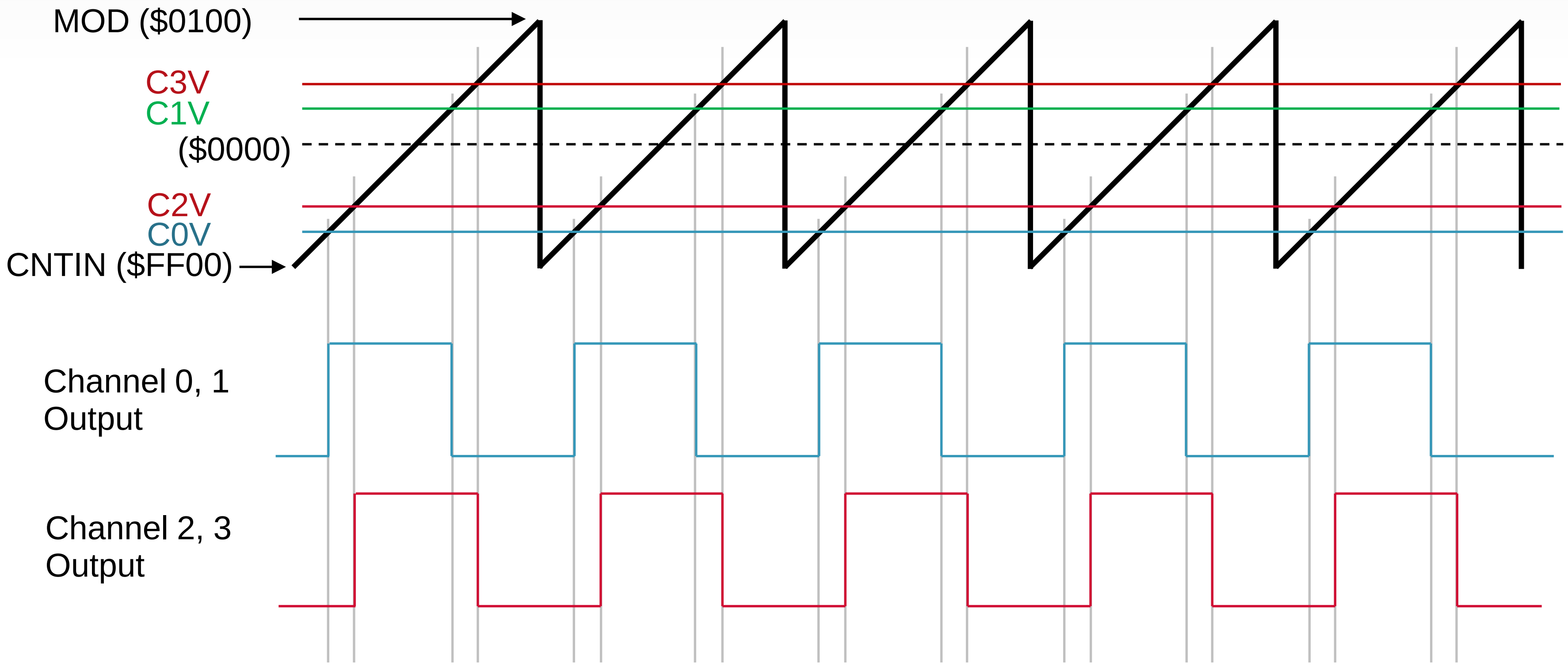
- All PWM-on values are set to the init value, and never changed again. Positive PWM-off values generate pulse widths above 50% duty cycle. Negative PWM-off values generate pulse widths below 50% duty cycle . This works well for bipolar waveform generation.

FlexTimer – Center Aligned PWM Generation



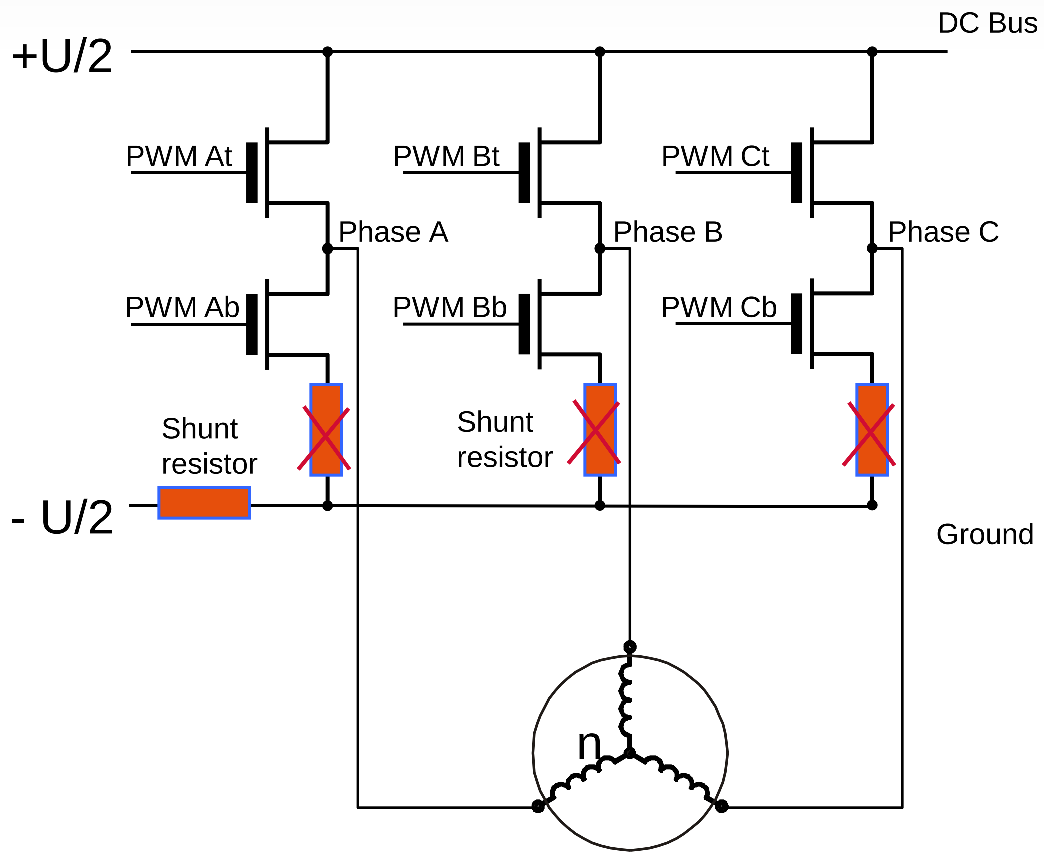
- When the Init value is the signed negative of the Modulus value, the PWM module works in signed mode. Center-aligned operation is achieved when the turn-on and turn-off values are the same number, but just different signs.

FlexTimer – Shifted PWM Generation



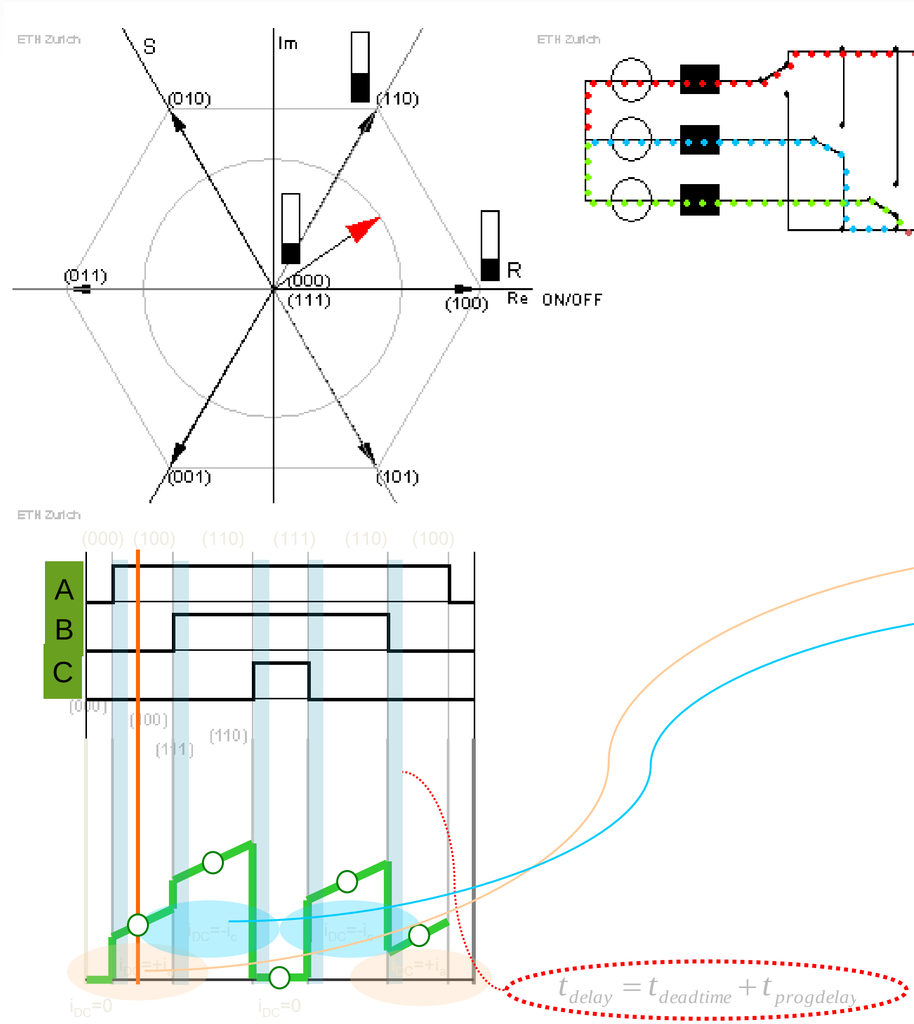
- In this example, both PWMs have the same duty-cycle. However, the edges are shifted relative to each other by simply biasing the compare values of one waveform relative to the other.

Single Shunt Current Reconstruction



3-ph AC Induction Motor
3-ph PM Synchronous Motor

Single Shunt Current Reconstruction - Analysis

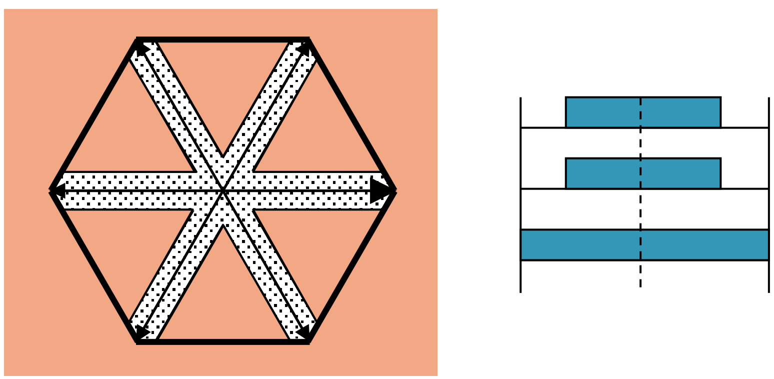


- Measurement Table

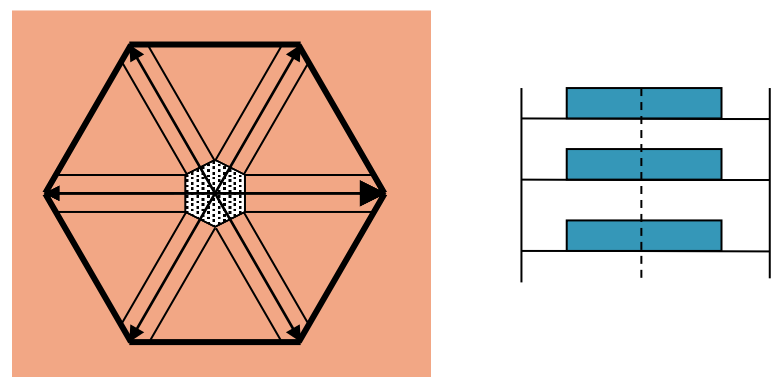
Voltage Vector	DC-Link current i_{dc}
V_1(100)	$+i_a$
V_2(110)	$-i_c$
V_3(010)	$+i_b$
V_4(011)	$-i_a$
V_5(001)	$+i_c$
V_6(101)	$-i_b$
V_7(111)	*0*
V_0(000)	*0*

Single Shunt Current Reconstruction - Issues

- Two current samples cannot be taken:
  1. Voltage vector is crossing a sector border
     - Only one sample can be taken
  2. Low modulation indexes
     - Sampling intervals too short
     - None of current samples can be taken



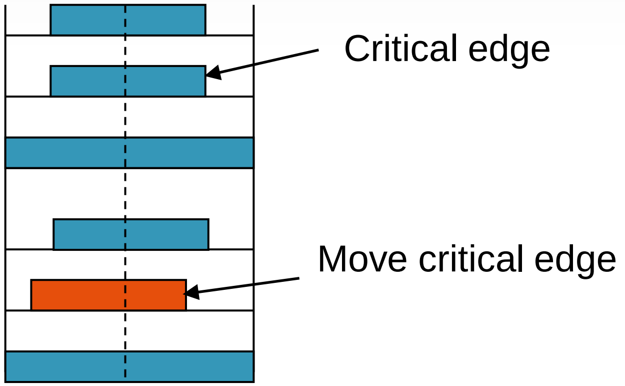
Passing Active Vector



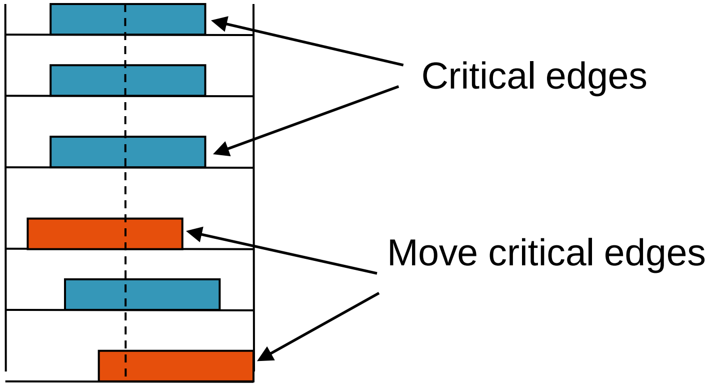
Low Modulation Index

Single Shunt Current Reconstruction - Solution

- Asymmetrical PWMs
  - Case 1 – Passing active vector:
    - Freeze center edge
    - Move one critical edge
    - Goes for higher modulation indexes



  - Case 2 – Low modulation indexes:
    - Freeze center edge
    - Move both side edges in opposite direction
    - Goes for low modulation indexes



Block Commutation

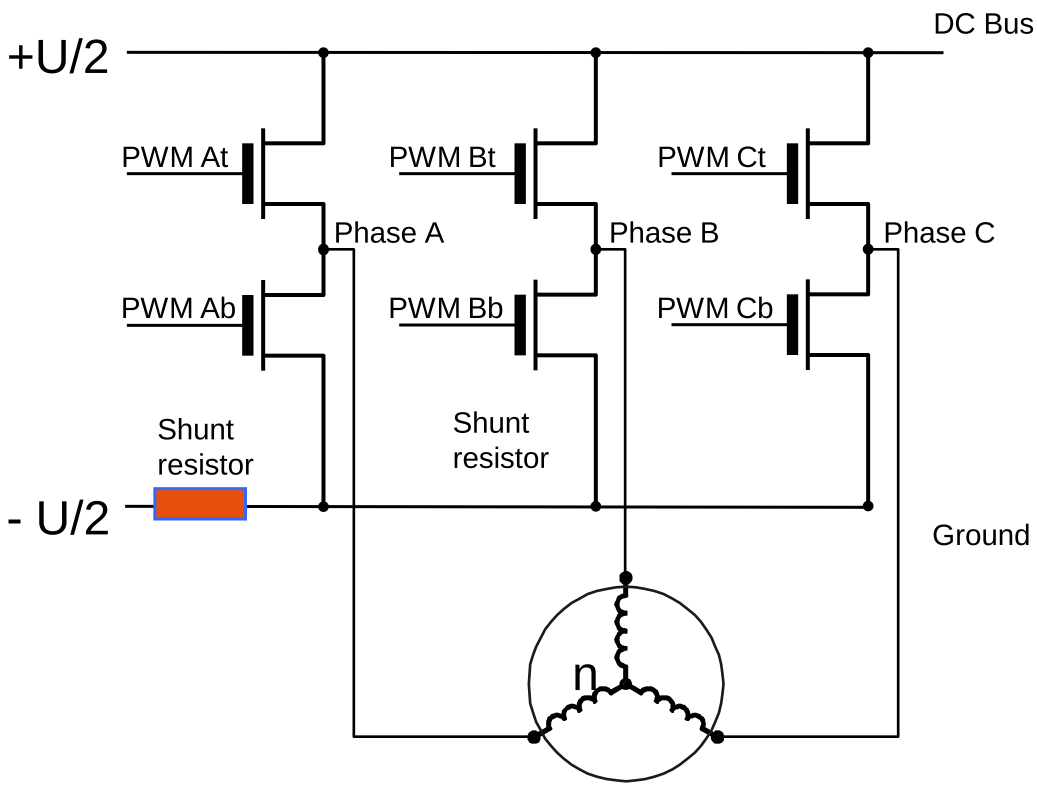
- The commutation depends on rotor position and it is asynchronous to PWM modulation
- The PWM outputs state has to be change at any time during PWM period but the PWM update is done at the end of the PWM period only
- The solution is to use mask, invert and sw control features on FlexTimer module

FTM Mask, Invert and SW Control Features

- MASK Control
  - The Mask feature disable PWM output regardless to duty cycle value
- Inverter Control
  - The Invert feature inverts signal going to complementary logic. It results in signals swap for top and bottom transistor.
  - This feature can be used in complementary mode
- Software Control
  - This feature set user value (0, 1) to PWM output regardless to duty cycle value

BLDC Motor Commutation

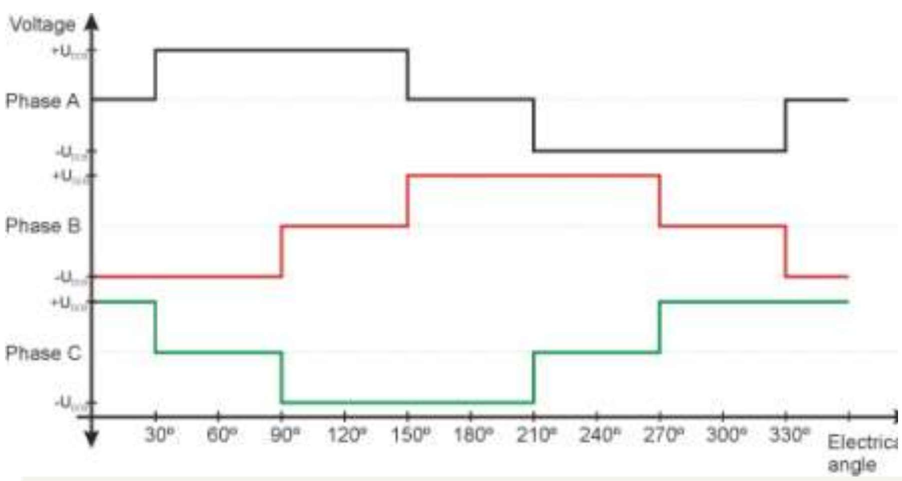
- Six Step BLDC Motor Control
  - Voltage applied on two phases only



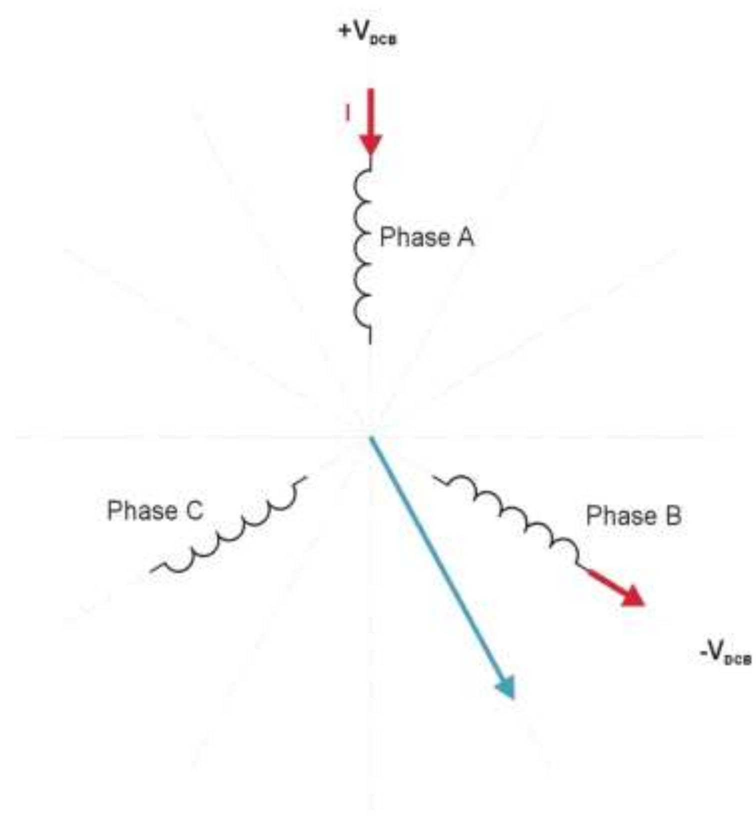
3-ph Brushless DC Motor

BLDC Motor Commutation

- Six Step BLDC Motor Control
  - Voltage applied on two phases only
  - It creates 6 flux vectors
  - Phases are powered based on rotor positi
  - The process is called commutation

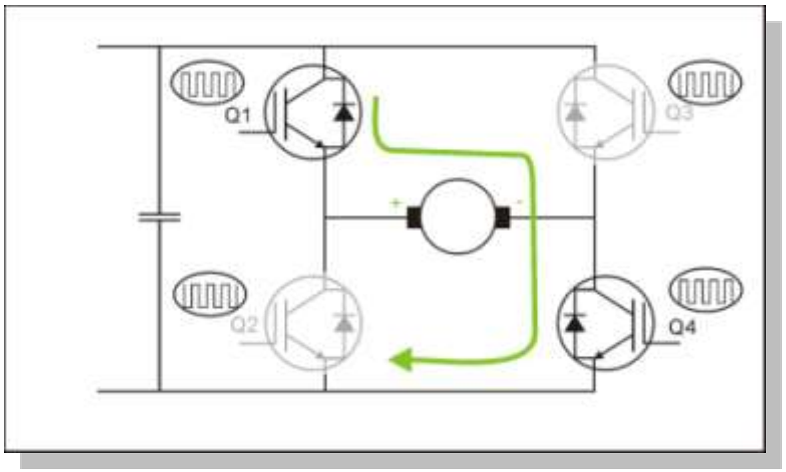


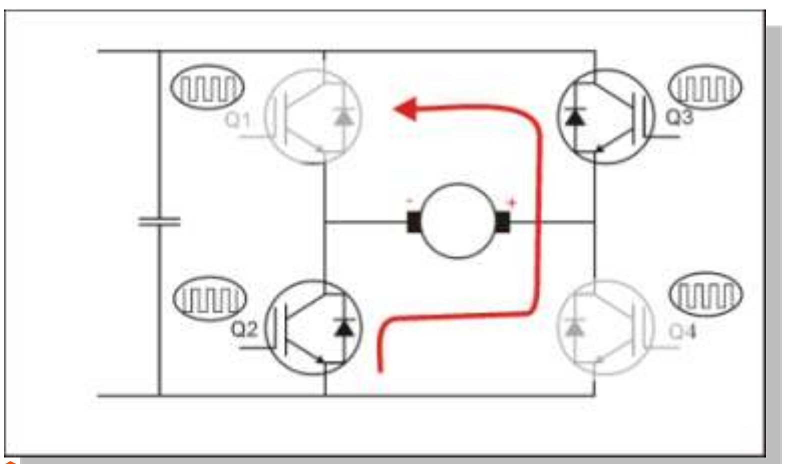
Phase voltages

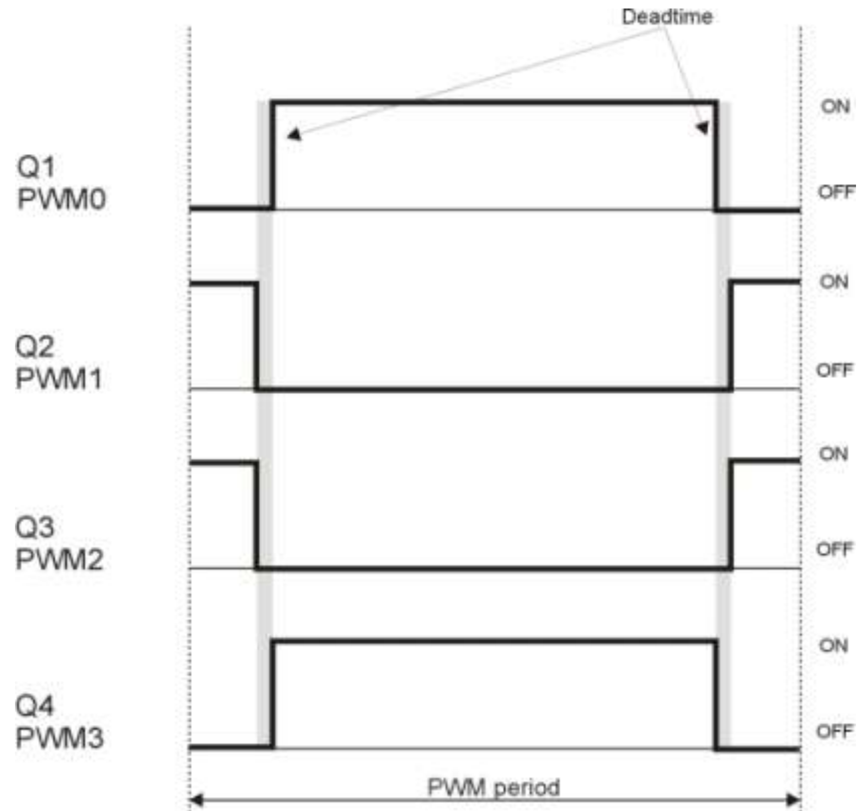


BLDC Motor Commutation

- Complementary bipolar PWM switching

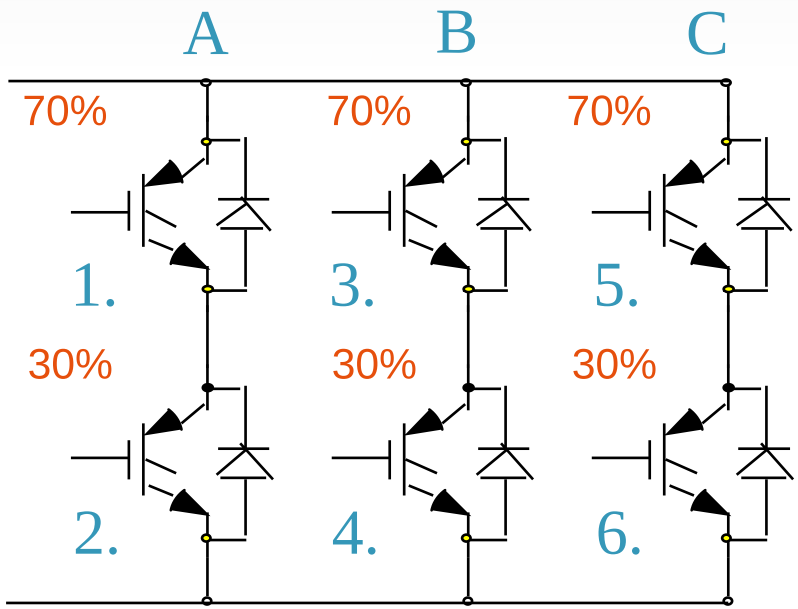






$Q1=Q4=PWM; \; Q2=Q3=\overline{Q1}$

BLDC Motor Commutation



PWM Output Mask Register
(FTMx_OUTMASK)

								0	0	0	0	0	0	0	0

FTM Inverting Control Register
(FTMx_INVCTRL)

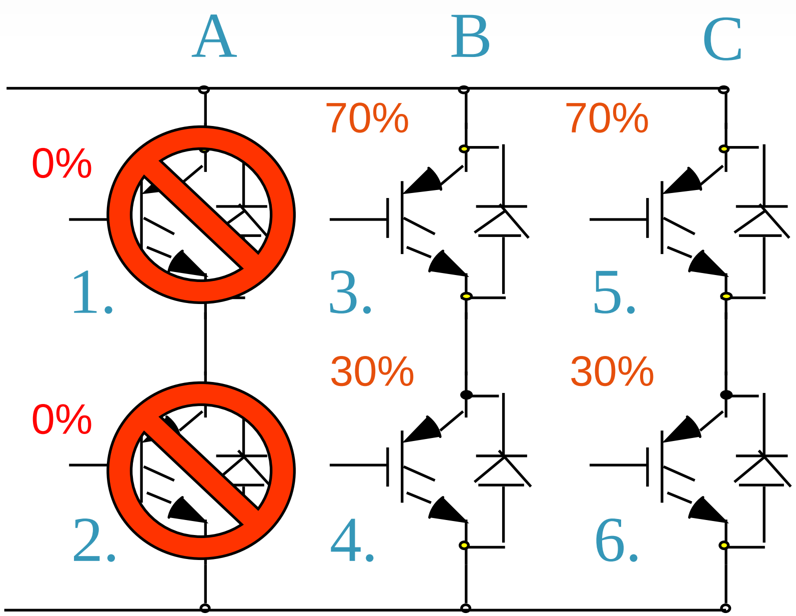
												0	0	0	0

All six FTMx_CnV registers are set to generate 70 % Duty Cycle
Complementary logic with deadtime enabled

Speed Control

freescale™

BLDC Motor Commutation



MASK

All six FTMx_CnV registers are set to generate 70 % Duty Cycle
Complementary logic with deadtime enabled

Speed Control

PWM Output Mask Register
(FTMx_OUTMASK)

								0	0	0	0	0	0	**1**	**1**

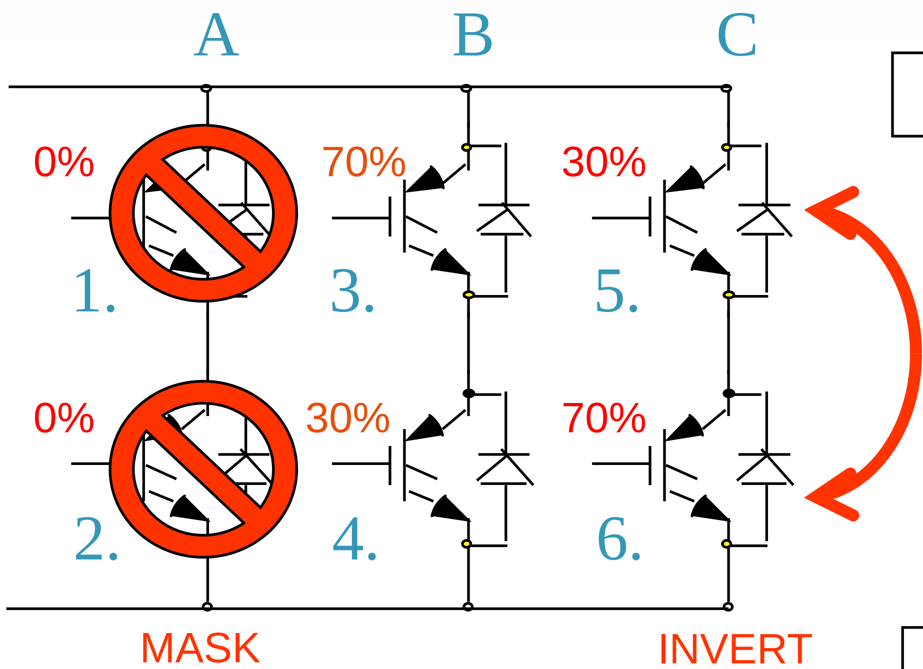
6. 5. 4. 3. 2. 1.
transistors

Mask will disable the complementary transistors pair

FTM Inverting Control Register
(FTMx_INVCTRL)

												0	0	0	0

BLDC Motor Commutation



PWM Output Mask Register
(FTMx_OUTMASK)

								0	0	0	0	0	0	**1**	**0**

6. 5. 4. 3. 2. 1.
transistors

Mask will disable the complementary transistors pair

FTM Inverting Control Register
(FTMx_INVCTRL)

												0	**1**	0	0

All six FTMx_CnV registers are set to generate 70 % Duty Cycle
Complementary logic with deadtime enabled

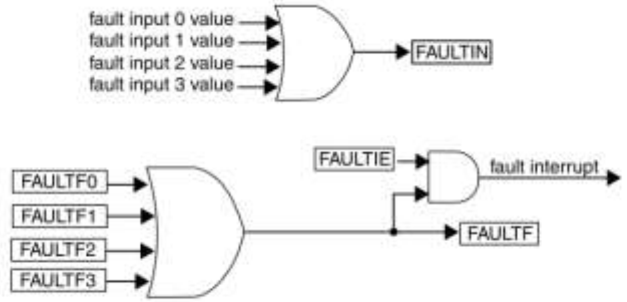
Speed Control

Commutation Control

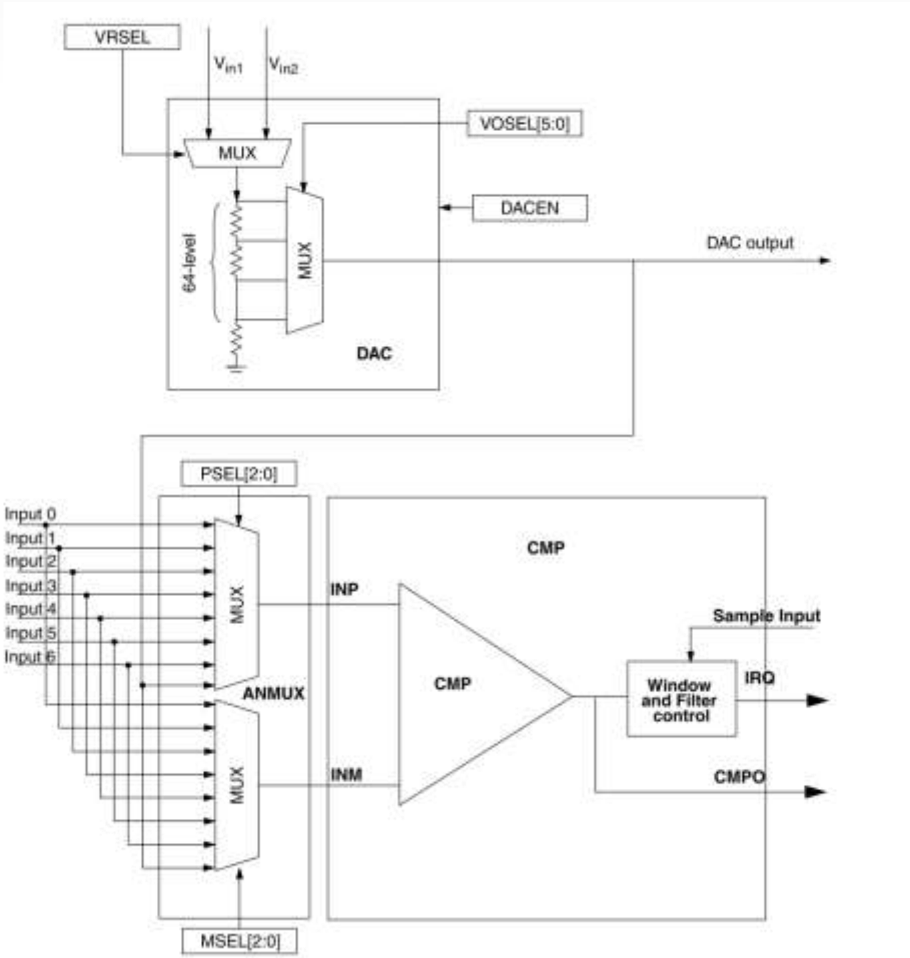
Invert reroutes the top and bottom control signals of complementary pair

FTM Fault Control

- There are four fault inputs ORed into single fault signal
- The fault signal disables all PWM outputs
- The polarity of the fault signal is user configurable
- The all inputs have input filter
- Manual or automatic clear fault control



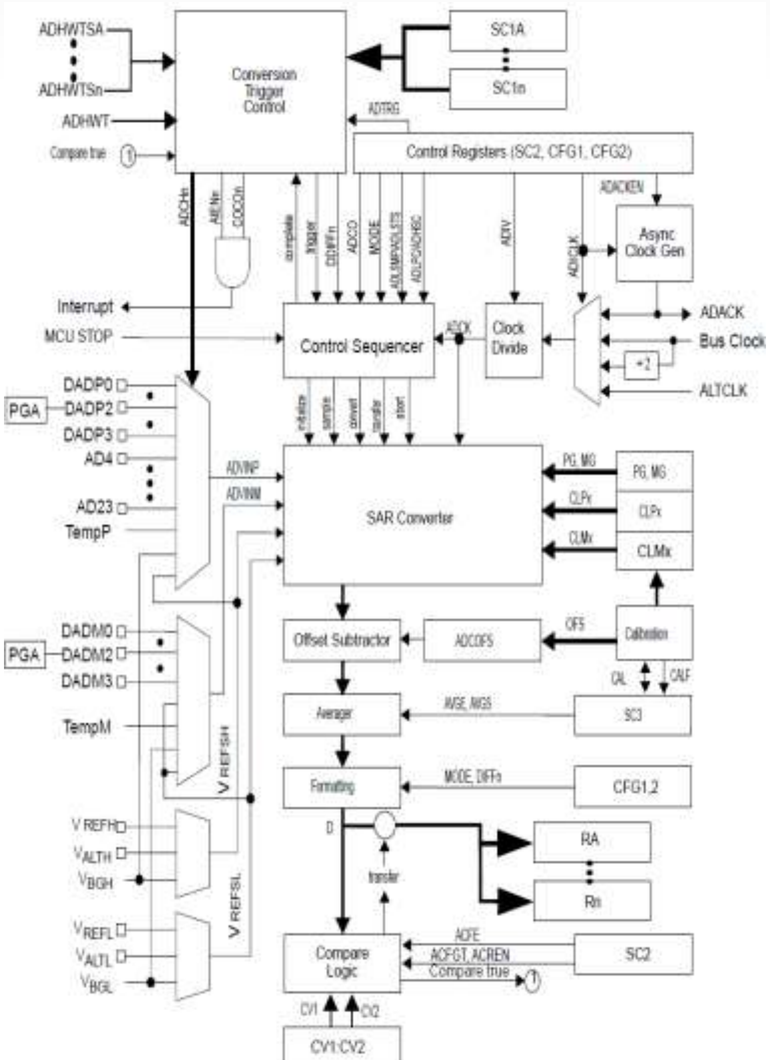
Build-in Comparator



- Continuous, Sampled, Windowed modes
- Programmable filter and hysteresis
- Up to eight independently selectable channels for positive and negative comparator inputs
- External pin inputs and several internal reference options including 6bit DAC, 12bit DAC, bandgap, VREF, OpAmp,
- 6-bit DAC
  - Output range (Vin/64) to Vin
  - VREF or VDD selectable as DAC reference

freescale™

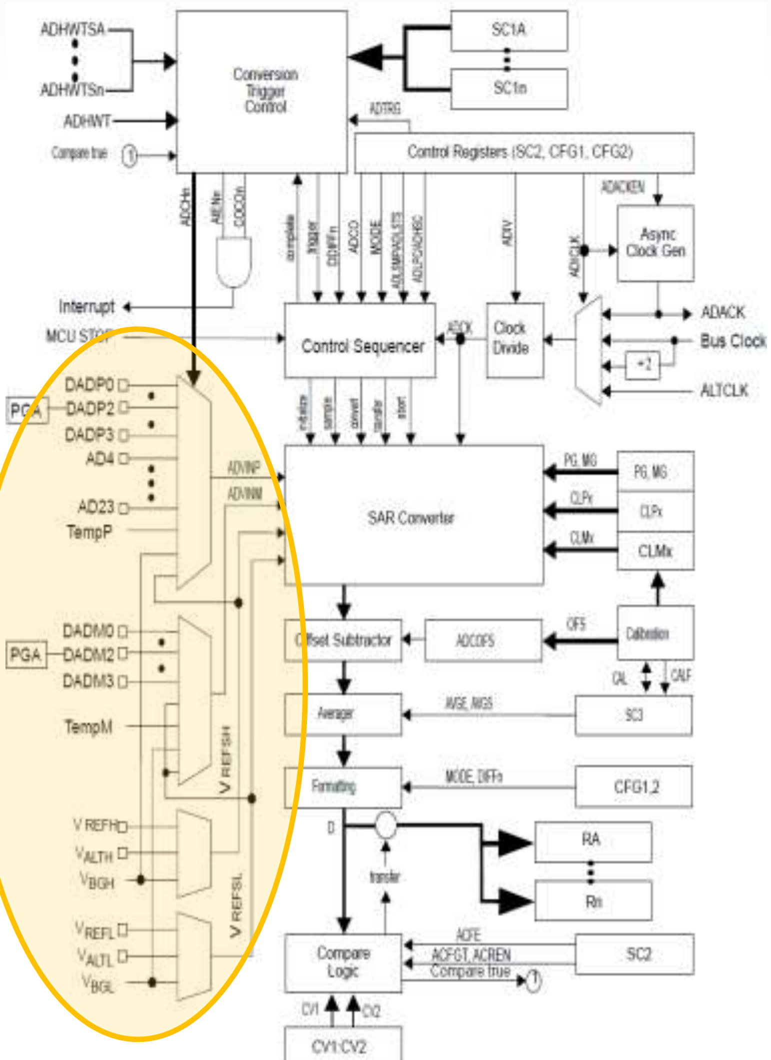
16-bit ADC – Analog Quantities Measurement



- Up to 4 pairs of differential and 24 single-ended external analog inputs
- Single or continuous conversion (automatic return to idle after single conversion)
- Configurable sample time and conversion speed/power
- Input clock selectable from up to four sources
- Operation in low power modes
- Asynchronous clock source for lower noise operation
- Selectable hardware conversion trigger with hardware channel select
- Automatic compare with interrupt for less-than, greater-than or equal-to, within range,or out-of-range, programmable value
- Temperature sensor
- Hardware average function
- Selectable voltage reference: external or alternate
- Self-calibration mode
- Programmable Gain Amplifier (PGA) with up to x64 gain

freescale™

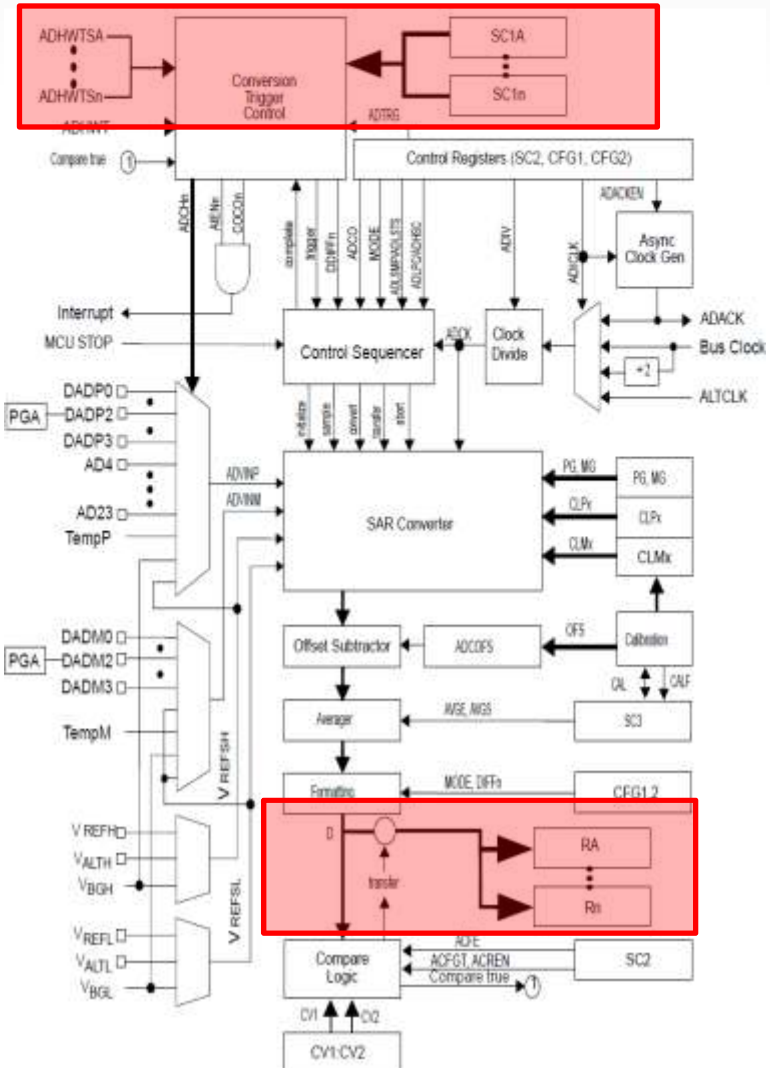
ADC – Analog Inputs Section



- Up to **24 single ended** channels and **4 differential** channels
- **Internal channel connections from:**
  - **DAC,**
  - **Temp Sensor**
  - **PMC Bandgap**
  - **Vrefh, Vrefl**
  - **Vref_Out**
- VREF selection from:
  - **Vrefh,Vrefl** external pin pair or
  - **VREF module**
- **Channel Interleaving** on s.e. and diff. channels

freescale™

Multiple Channel Select and Result Registers

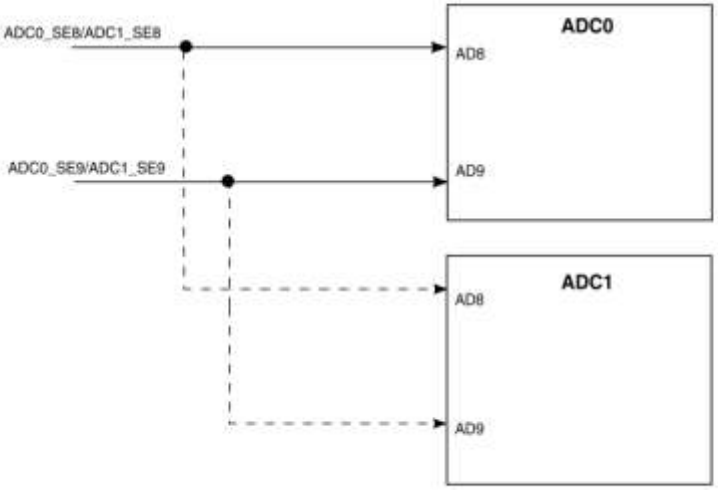


- Multiple **ADC_SC1n** registers are used to select channels and conversion modes for the ADC
- Each **ADC_SC1n** register contains its own channel selection bitfield interrupt enable and conversion complete flag to allow flexibility in the interrupt handling
- Programmable Delay block hardware triggers (and also other trig sources) can be sent to the ADC to initiate conversions at pre-set time intervals for detailed control of ADC conversion timing
- Results for each ADCSC1 are stored in individual result registers **ADC_Rn**
- **2 sets** of control (ADC_SC1n) and result (ADC_Rn) registers implemented on available Kinetic devices
- **up to 4 ADC modules** available on Kinetis devices

3-phase Current Measurement

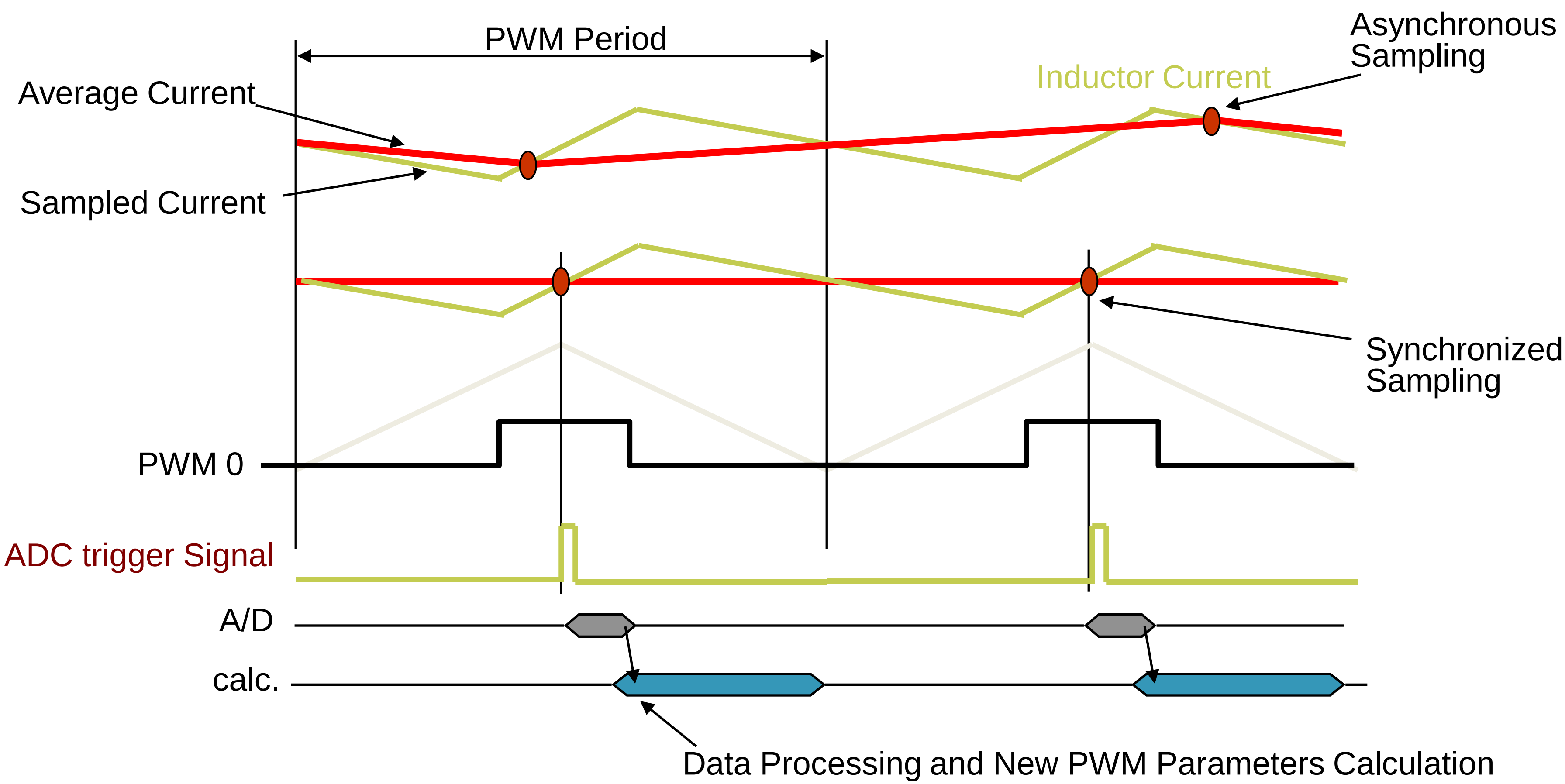
- The ADC provides one differential and one single ended input channel connected to both ADC modules
- It can be utilized with advantage for 3-phase current measurement
- We need to measure two phase currents in parallel (any combination)
- This requires to have one phase connected to both ADC modules
- Therefore it is desirable to connect one phase current signal to interleaved channel

Phase A => ADC0
Phase B => ADC1
Phase C => ADC0/ADC1



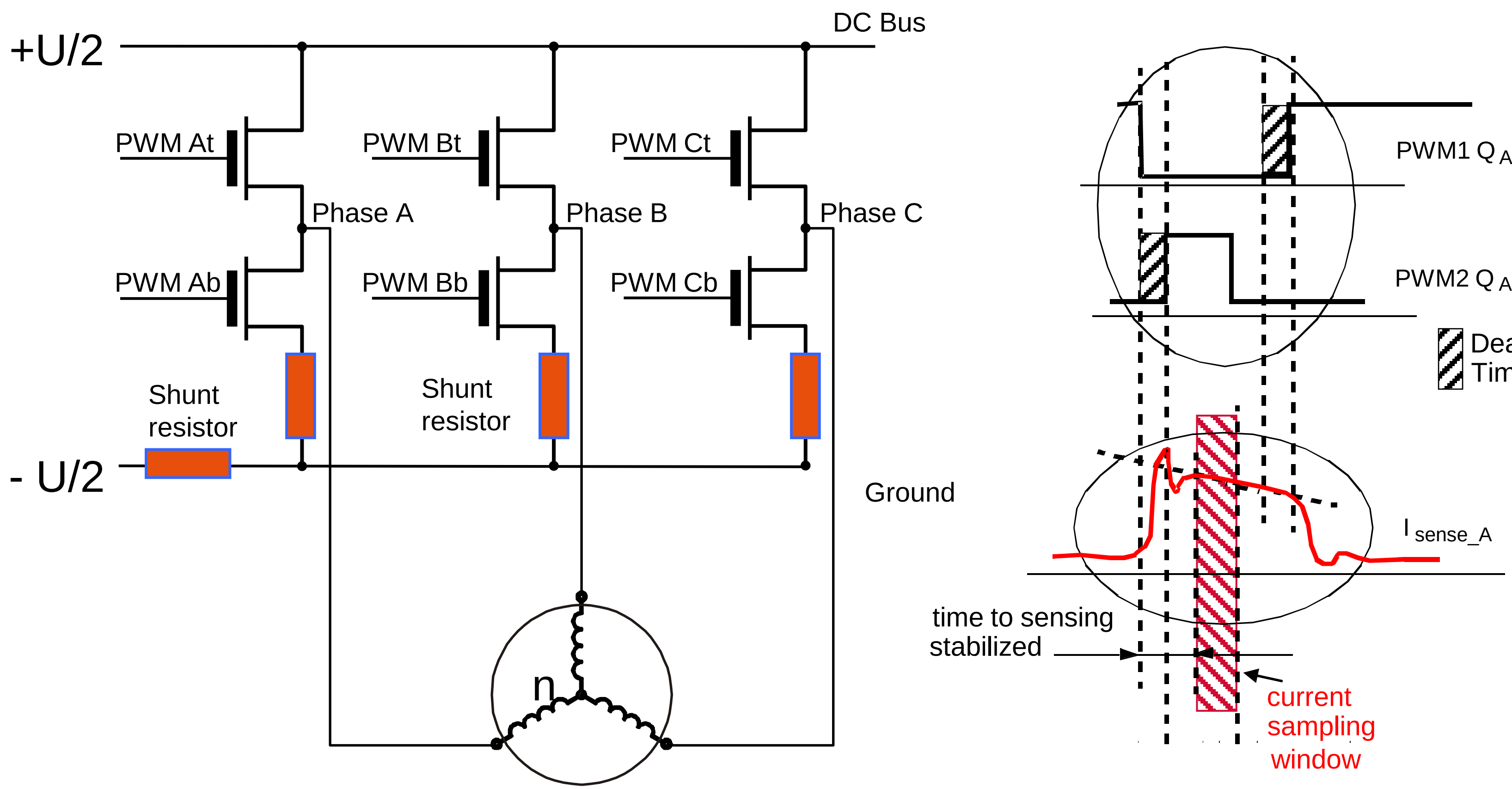
ADC to PWM Synchronization - Why Needed?

- ADC sampling helps to filter the measured current - antialiasing



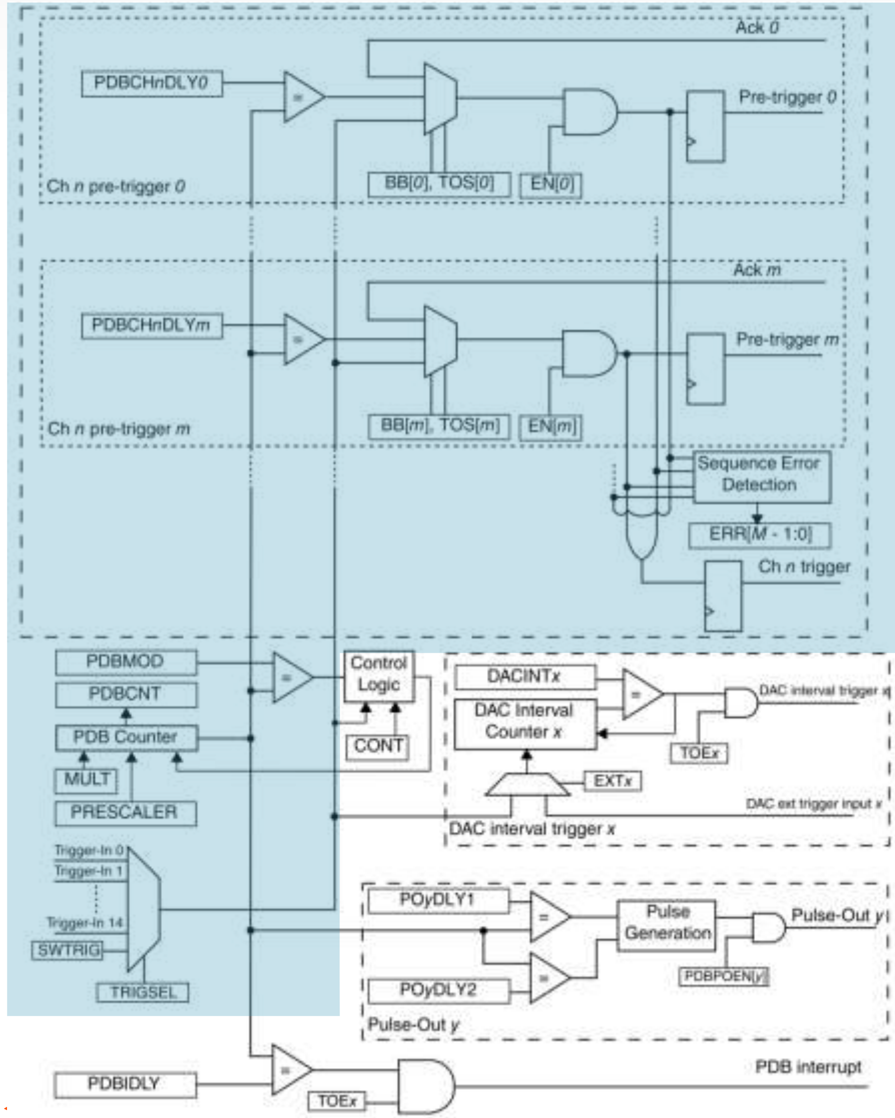
ADC to PWM Synchronization - Why Needed?

- Phase current can be sensed for certain time only



3-ph AC Induction Motor
3-ph PM Synchronous Motor

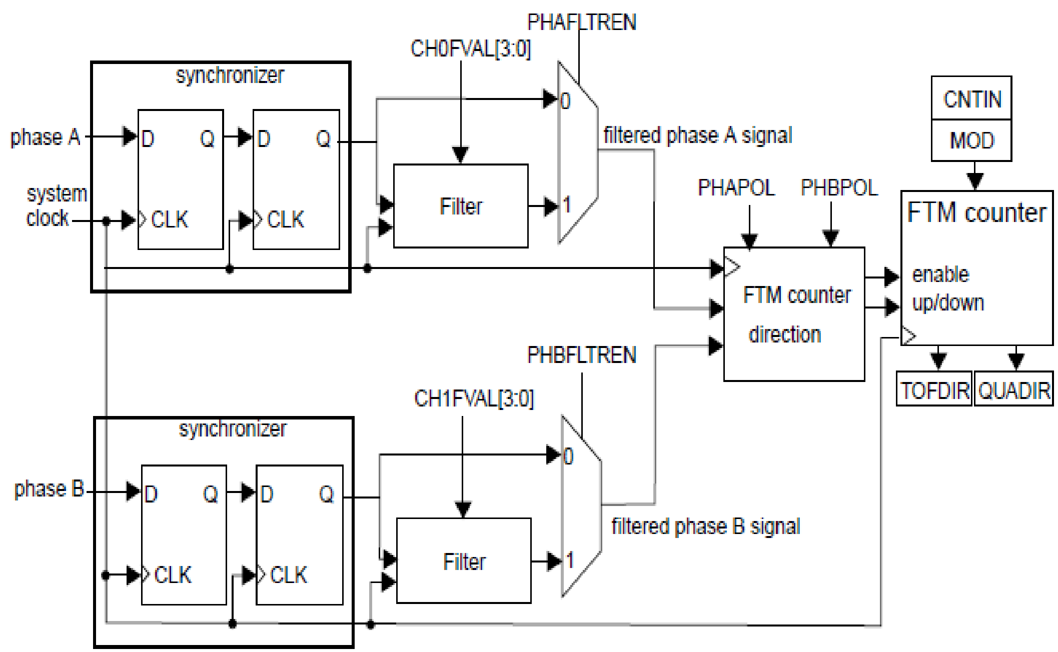
Programmable Delay Block (PDB)



- The PDB provides delays between input and output triggers
- Up to 4 channels available (one for each ADC) with two pretriggers
- Trigger 0 => Sample A
- Trigger 1 => Sample B

Speed/Position Measurement

- The FlexTimer can be used for Speed/Position Measurement
- Quadrature Mode
  - The FTM is capable to decode signals from quadrature encoder
  - There are input filters for both A and B inputs



freescale™

Speed/Position Measurement

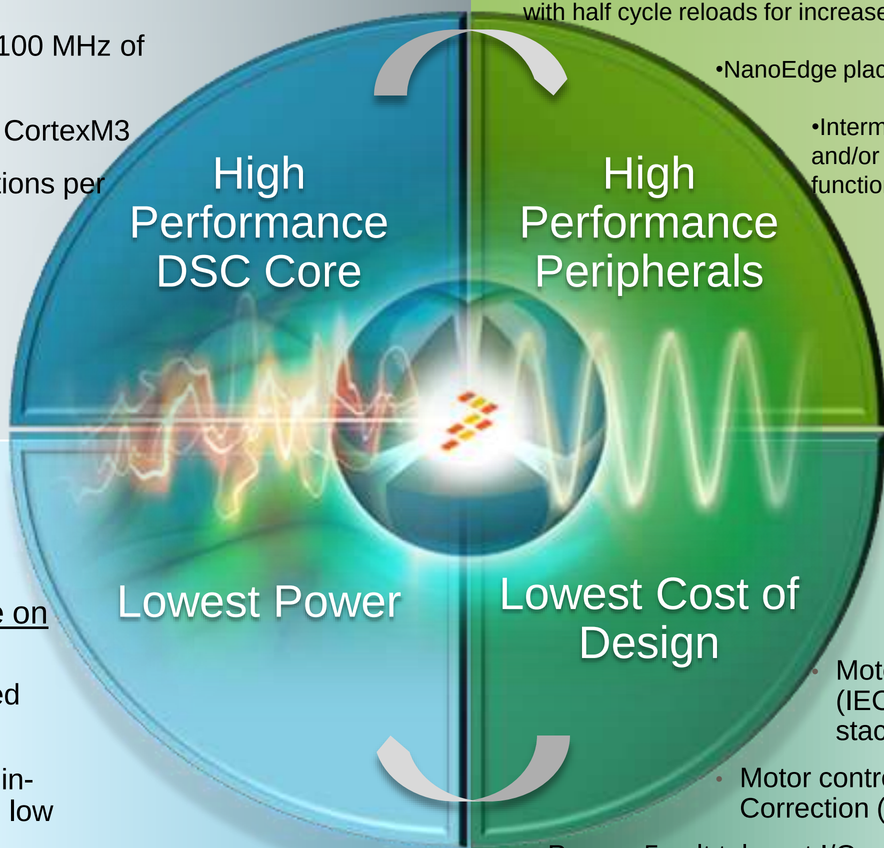
- **FlexTimer Dual Capture Capability**
  - FTM is capable to capture two consecutive edges
  - **The One-shot Capture mode**
    - Captures two edges and disable capturing
  - **The Continuous Capture mode**
    - The edges are captured continuously
  - **Pulse width measurement with both positive/negative polarity**
  - **Period measurement**
    - Between two consecutive edges of the same polarity
    - Between two consecutive rising/falling edges

DSC Series

Motor Control Peripherals

MC56F827x – Half the Power, Twice the Performance

(100MHz in a 5x5 32pin QFN!)



<u>100MHz DSP 32-BIT 56800EX Hawk V3 core</u>

- Fastest DSC in its class with 100 MHz of performance
- FIR Filter 6x faster than ARM CortexM3
- The highest number of operations per cycle of any MCU in its class
- Fractional arithmetic
- Nested looping
- Superfast interrupt

- eFlexPWM – Freescale's most advance timer for Digtial Power Conversion with up to 8ch and 312pico-sec resolution, supported by 4 independent time bases, with half cycle reloads for increased flexibility and best in class performance
- NanoEdge placer to implement fractional delays
- Intermodule Cross-Bar directly connecting any input and/or output with flexibility for additional logic functions (AND/OR/XOR/NOR)
- DAC with hardware Waveform generation support
- Very high speed ADCs capture events real time.

<u>The lowest power DSC available on the market</u>

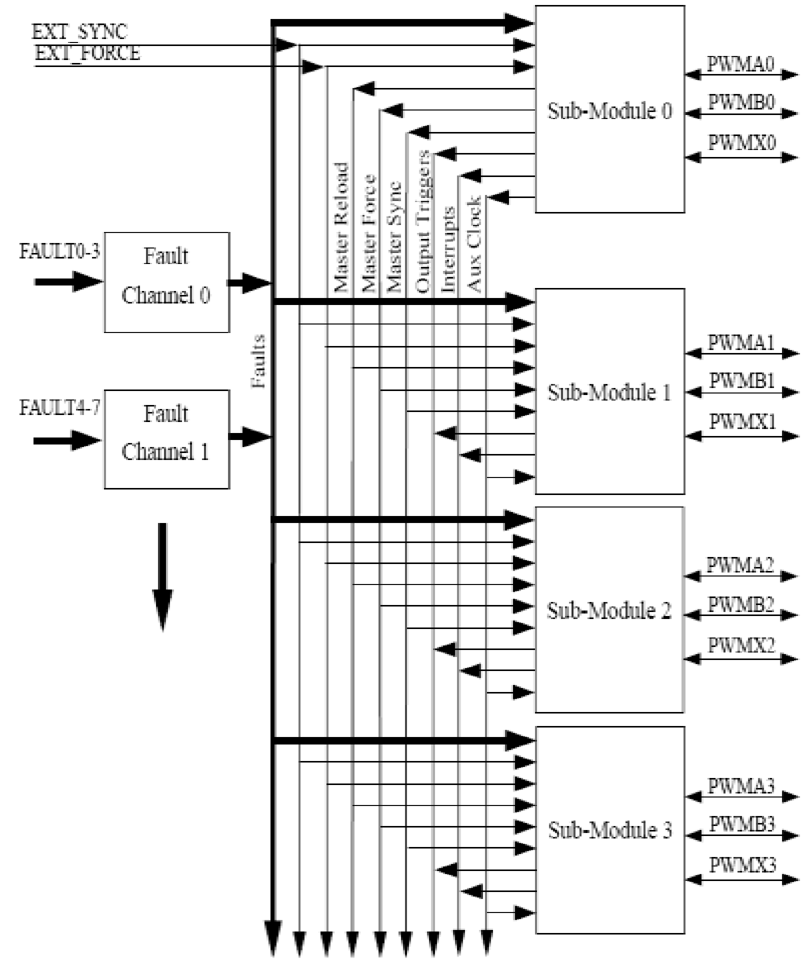
- Less than 0.4mA/Mhz at full speed run
- Concurrent operations offer best-in-class execution times and overall low power run rates.

<u>Advanced Integration & development speed</u>

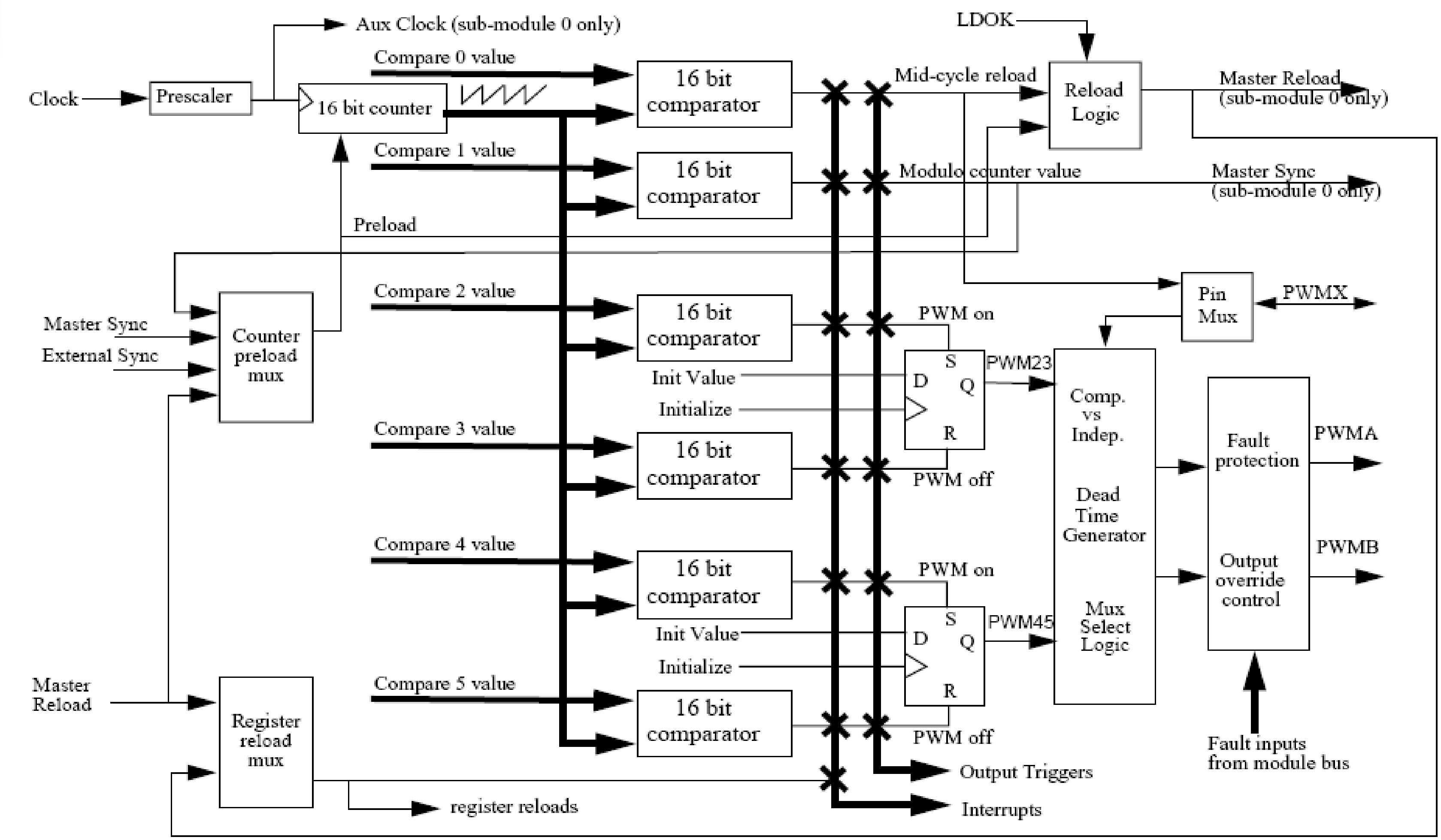
- A high level of on-chip integration lowers external Op Amp and capacitor costs.
- Motor Control, Power Control, Safety (IEC60730) Libraries, PMBus software stack, PLC software stack.
- Motor control with integrated Power Factor Correction (PFC) reducing chip count.
- Proven 5 volt tolerant I/O and Peripheral Crossbar enable greater flexibility and system cost reduction.
- Development tools, including FREEMaster

Enhanced Flex Pulse Width Modulator (eFlexPWM)

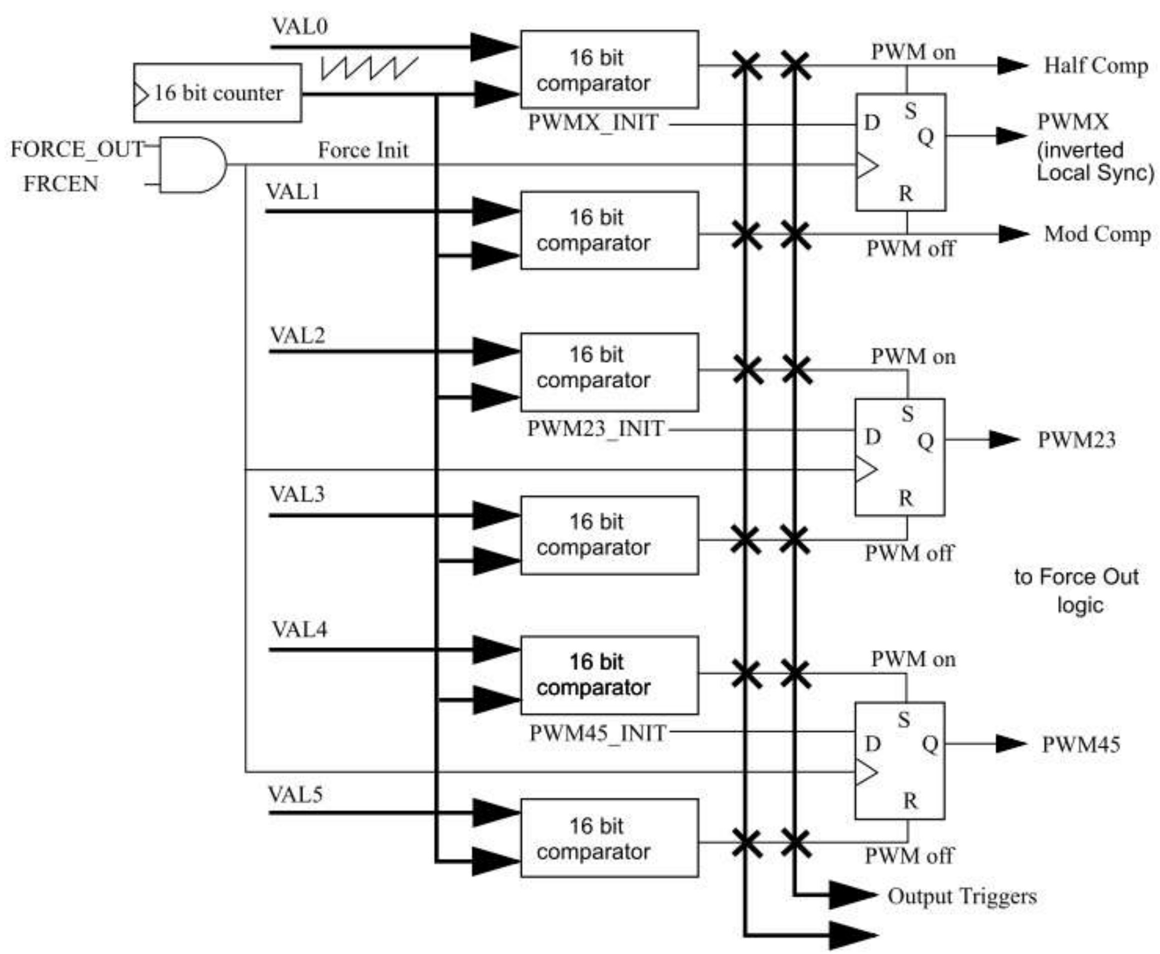
- Four independent sub-modules with own time base, two PWM outputs + 1 auxiliary PWM input/output
- 16-bits resolution for center, edge aligned, and asymmetrical PWMs
- Fractional delay for enhanced resolution of the PWM period and edge placement
- Complementary pairs or independent operation
- Independent control of both edges of each PWM output
- Synchronization to external hardware or other PWM sub-modules
- Double buffered PWM registers
- Integral reload rates from 1 to 16 include half cycle reload
- Half cycle reload capability
- Multiple output trigger events per PWM cycle
- Support for double switching PWM outputs
- Fault inputs can be assigned to control multiple PWM outputs
- Programmable filters for fault inputs
- Independently programmable PWM output polarity
- Independent top and bottom deadtime insertion
- Individual software control for each PWM output
- Software control, and swap features via FORCE_OUT event
- Compare/capture functions for unused PWM channels
- Enhanced dual edge capture functionality



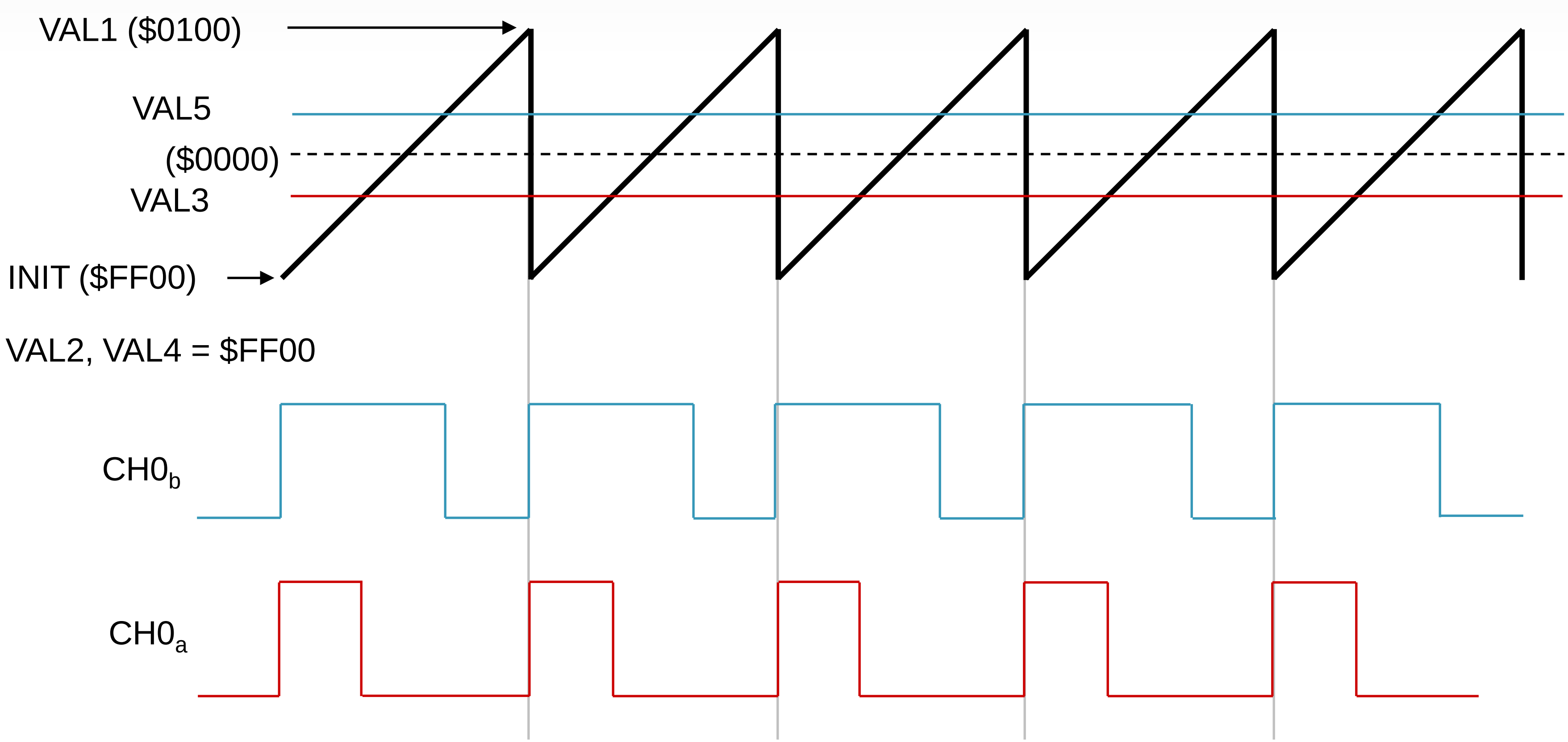
eFlexPWM - Sub-Module Detail



eFlexPWM – PWM Generation

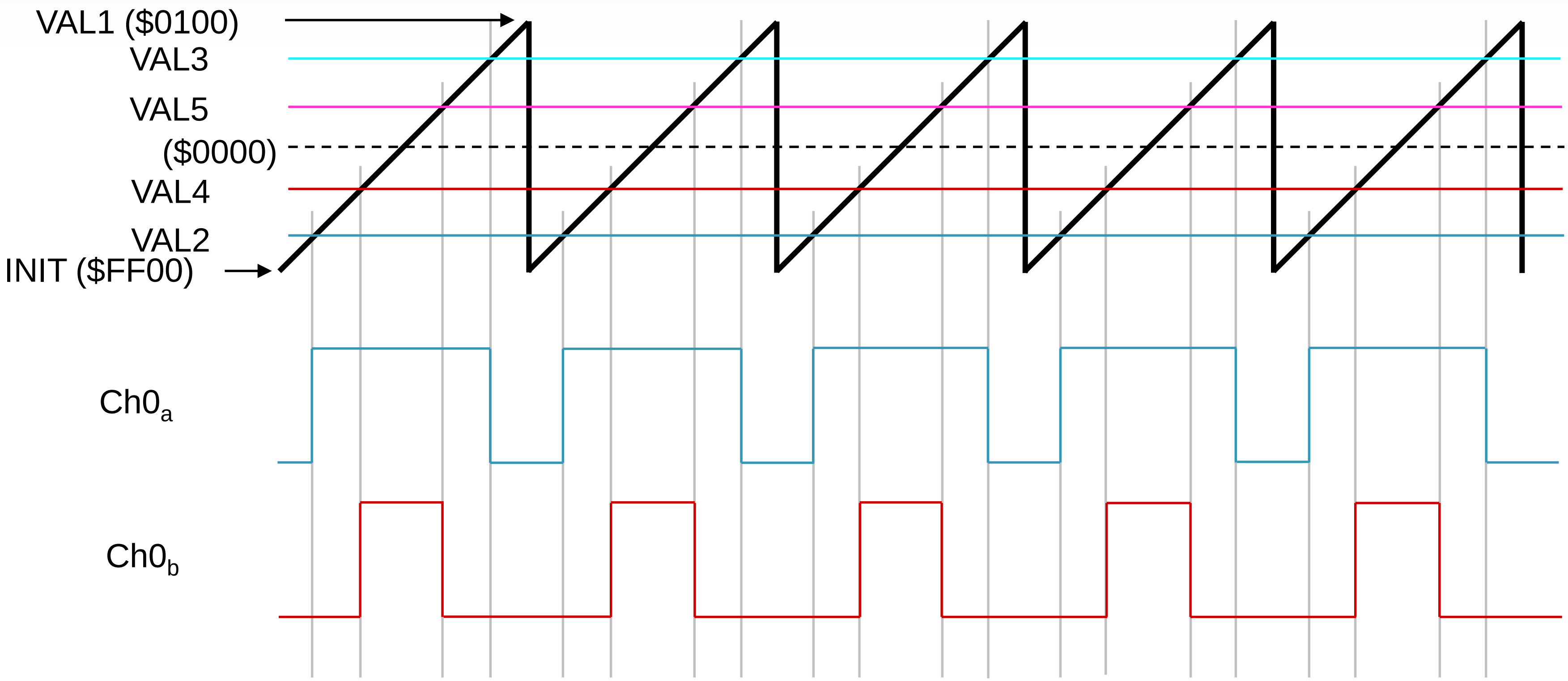


eFlexPWM – Edge Aligned PWM Generation



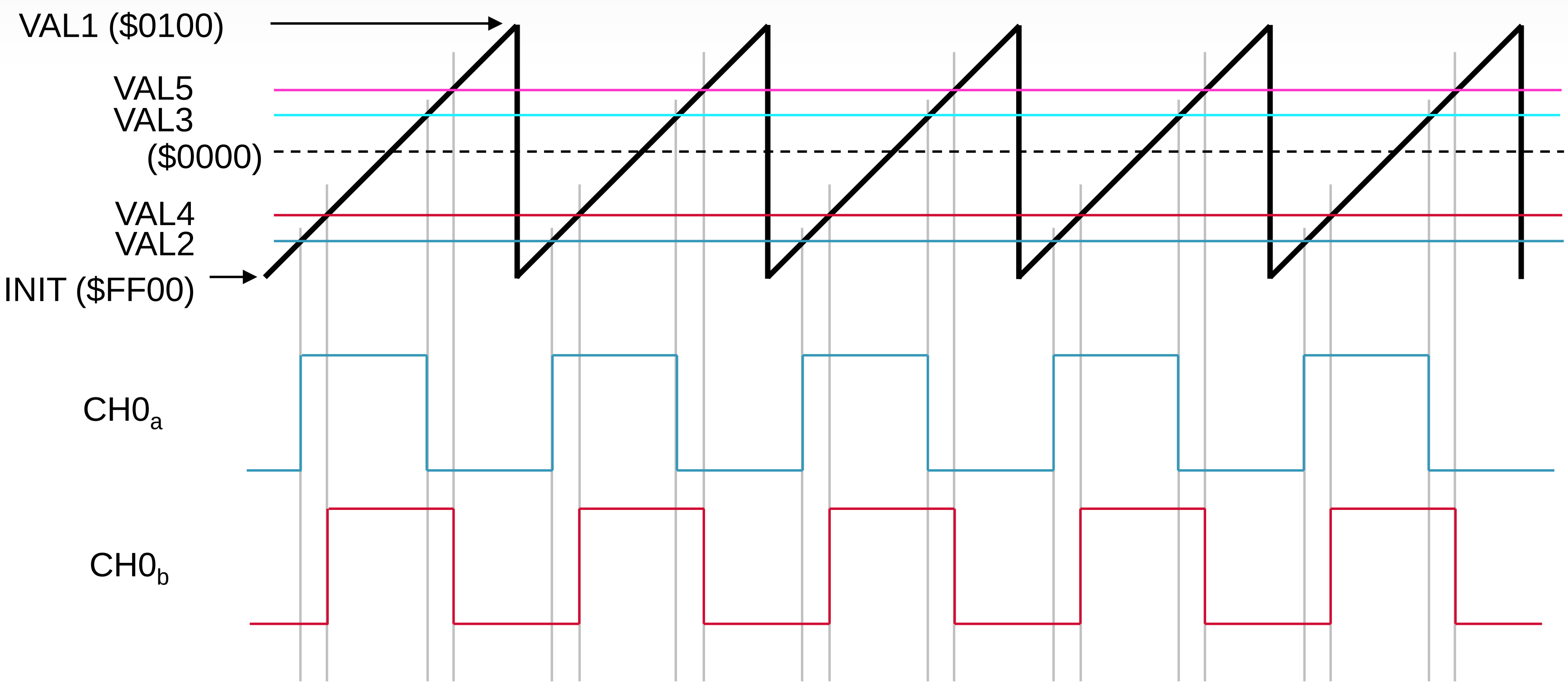
- All PWM-on values are set to the init value, and never changed again. Positive PWM-off values generate pulse widths above 50% duty cycle. Negative PWM-off values generate pulse widths below 50% duty cycle . This works well for bipolar waveform generation.

eFlexPWM – Center Aligned PWM Generation



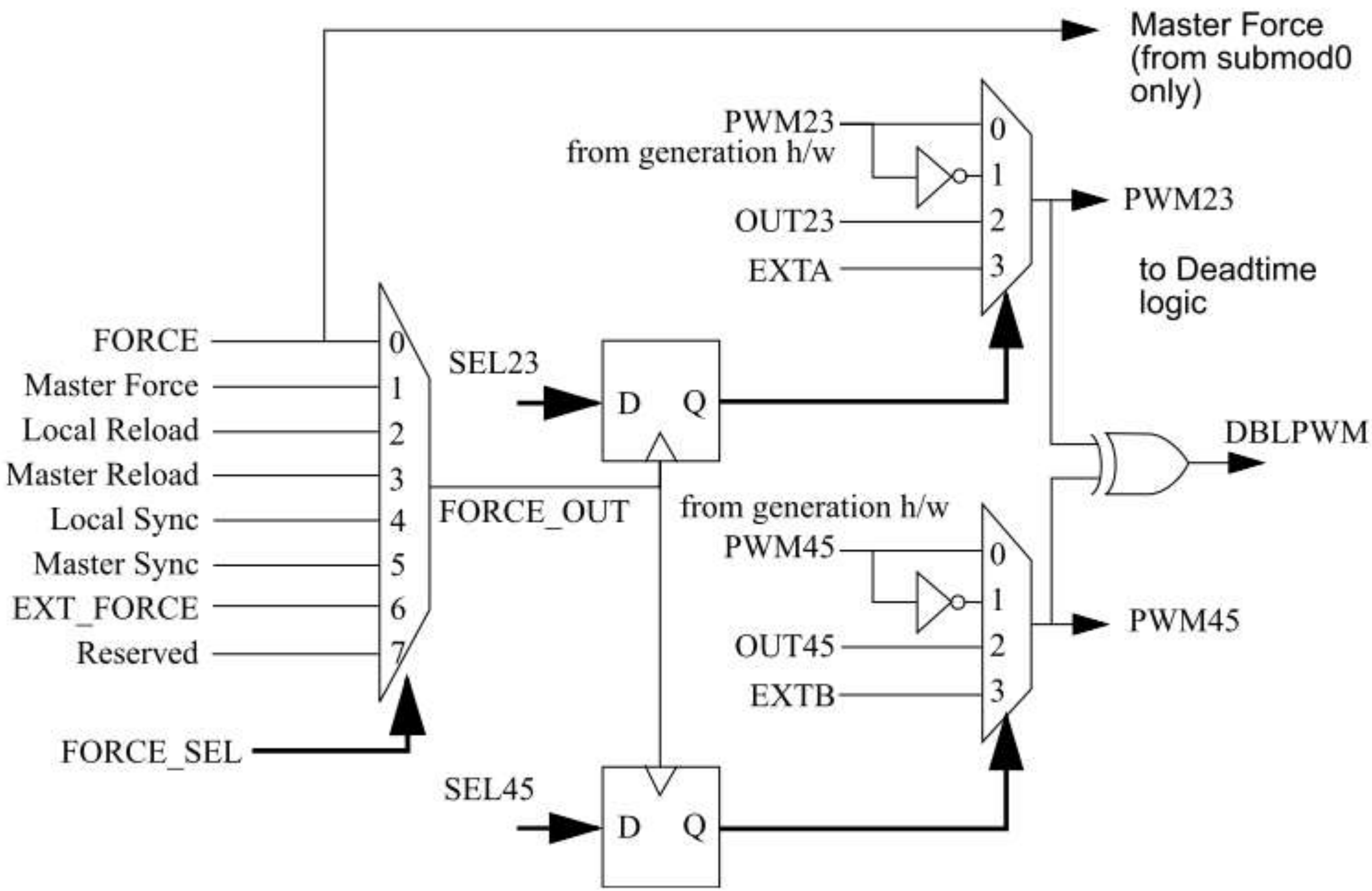
- When the Init value is the signed negative of the Modulus value, the PWM module works in signed mode. Center-aligned operation is achieved when the turn-on and turn-off values are the same number, but just different signs.

eFlexPWM – Shifted PWM Generation

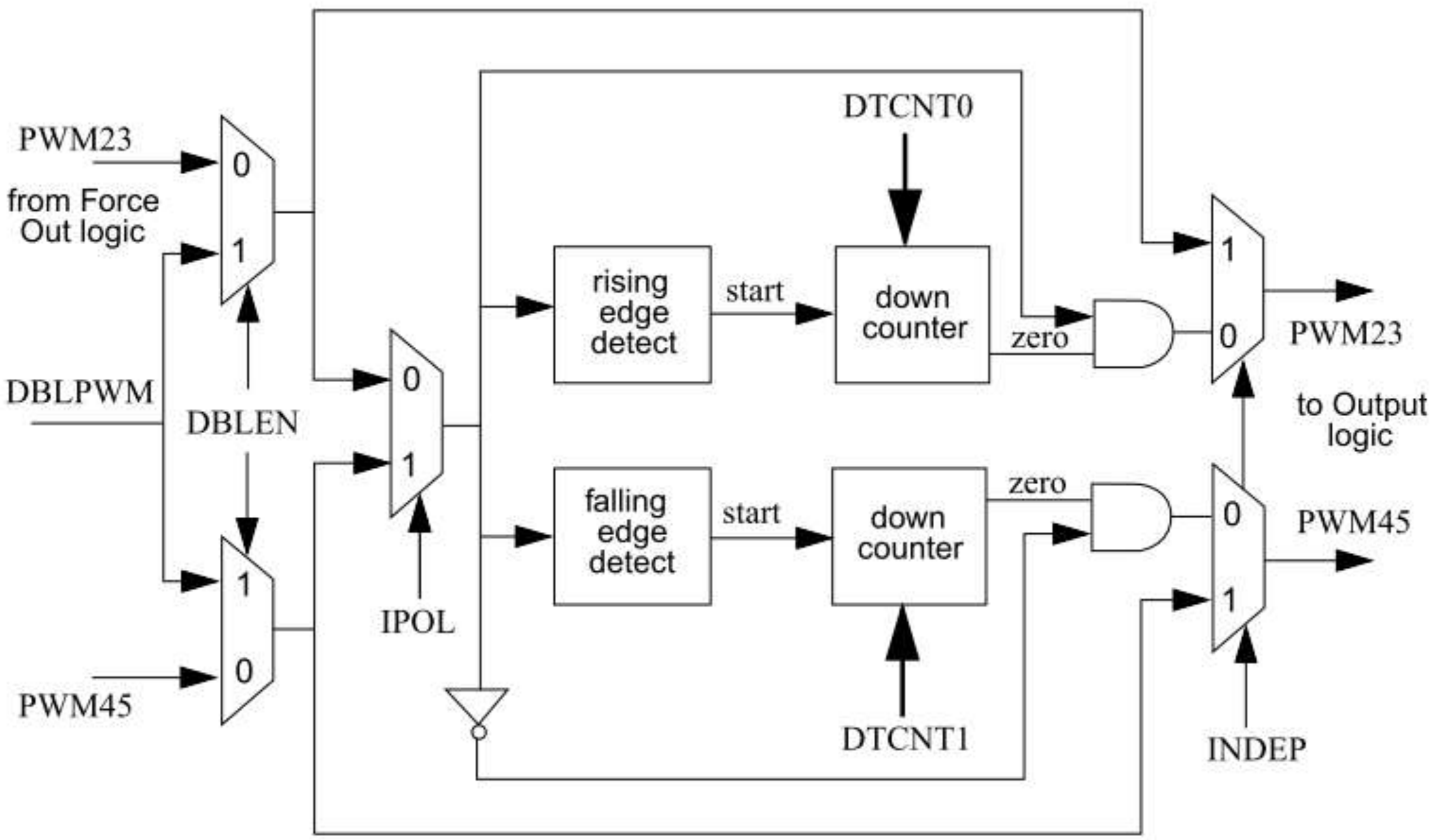


- In this example, both PWMs have the same duty-cycle. However, the edges are shifted relative to each other by simply biasing the compare values of one waveform relative to the other.

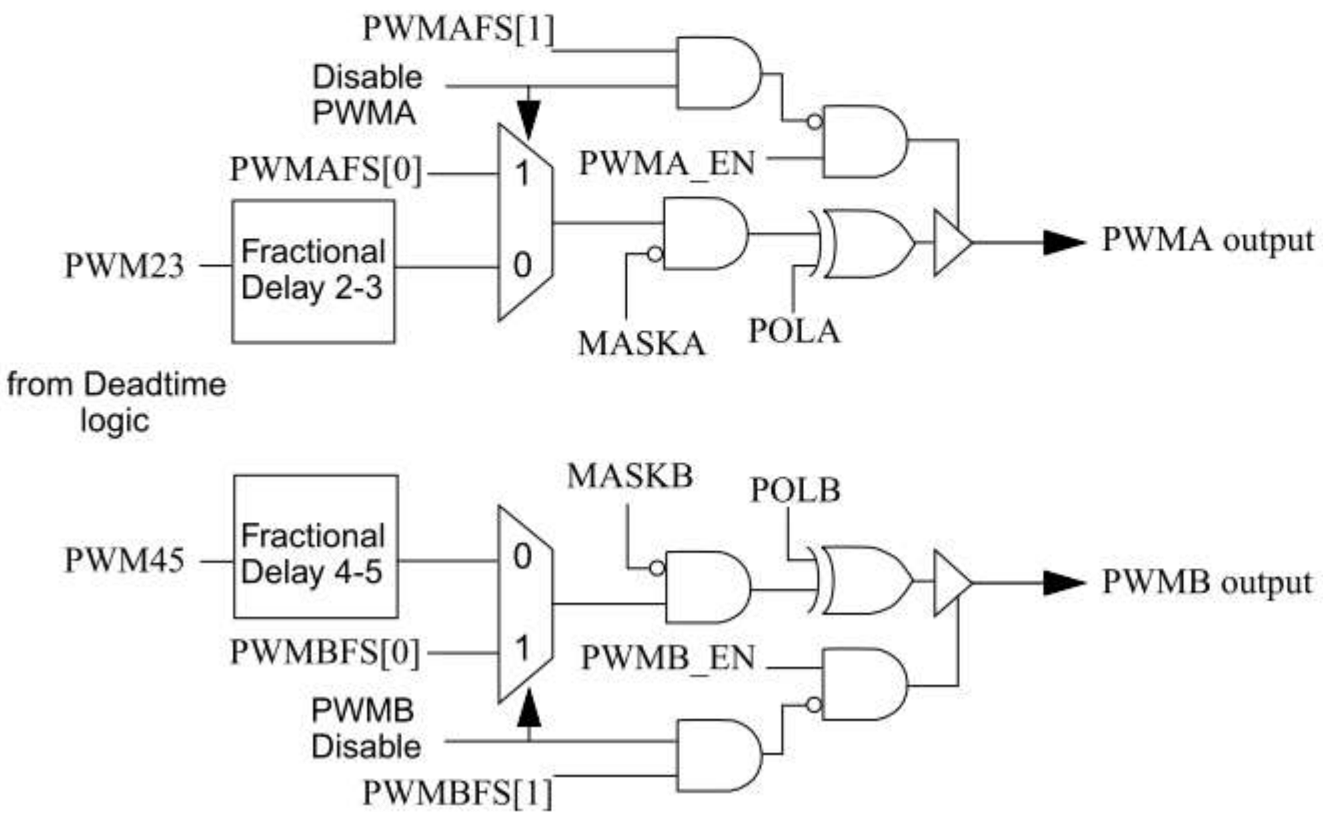
eFlexPWM – Force Output Logic



eFlexPWM – Complementary and Deadtime Logic

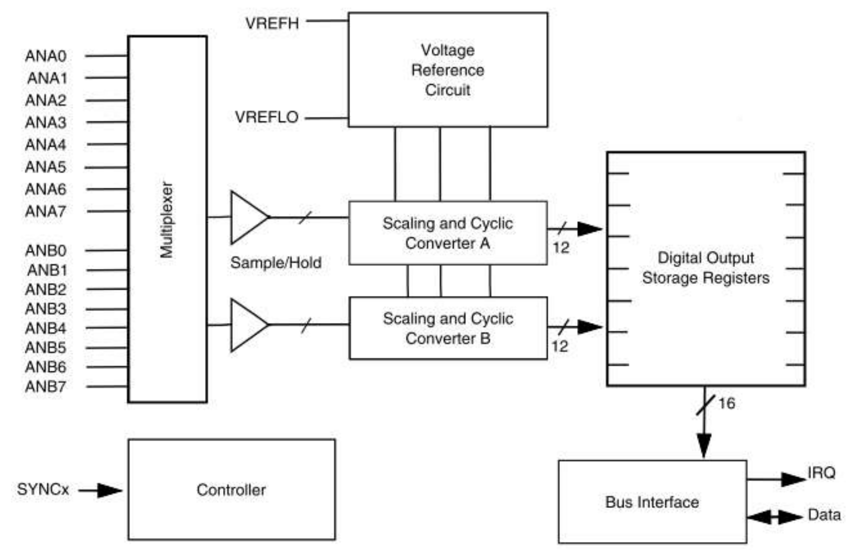


eFlexPWM – Fractional Delay and Output Logic

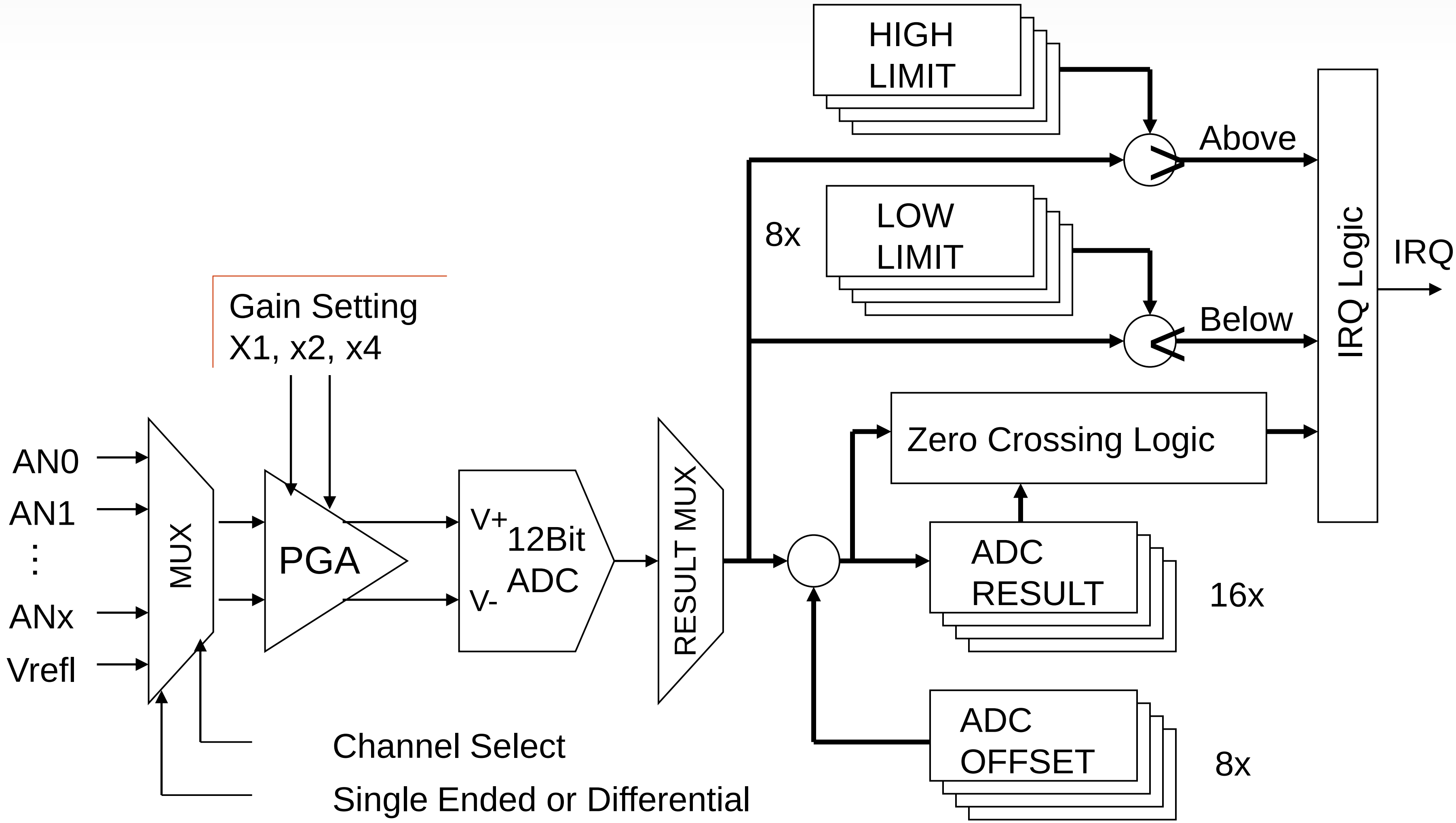


A/D Converter

- 12-bit resolution
- Maximum ADC clock frequency of 20 MHz with 50 ns period
- Sampling rate up to 6.66 million samples per second
- Single conversion time of 8.5 ADC clock cycles (8.5 × 50 ns = 450 ns)
- Additional conversion time of 6 ADC clock cycles (6 × 50 ns = 300 ns)
- ADC to PWM synchronization through the SYNC0/1 input signal sequentially scans and stores up to sixteen measurements
- Scans and stores up to eight measurements each on two ADC converters operating simultaneously and in parallel
- Scans and stores up to eight measurements each on two ADC converters operating asynchronously to each other in parallel
- Multi-triggering support
- Gains the input signal by x1, x2, or x4
- Optional interrupts at end of scan if an out-of-range limit is exceeded or there is a zero crossing
- Optional sample correction by subtracting a pre-programmed offset value
- Signed or unsigned result
- Single-ended or differential inputs
- PWM outputs with hysteresis for three of the analog inputs

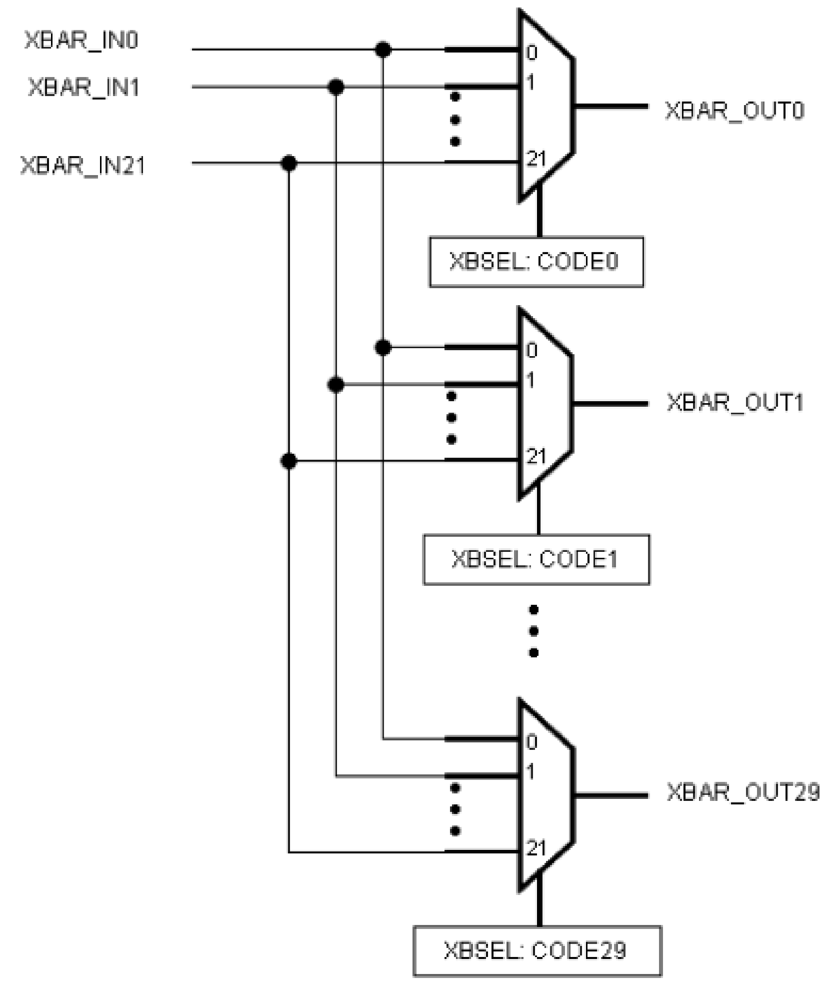


A/D Converter

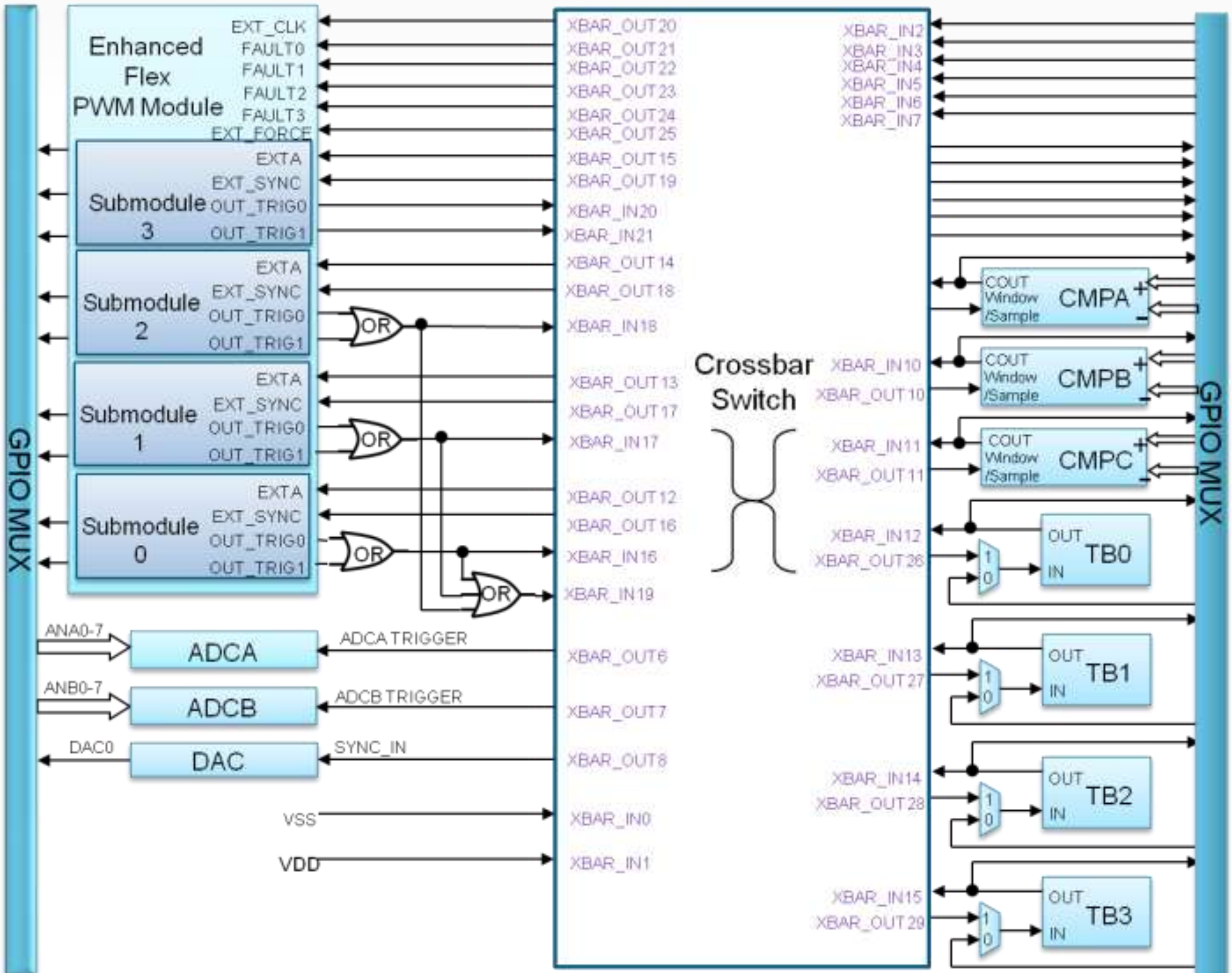


Crossbar Switch - MC56F824x/5x

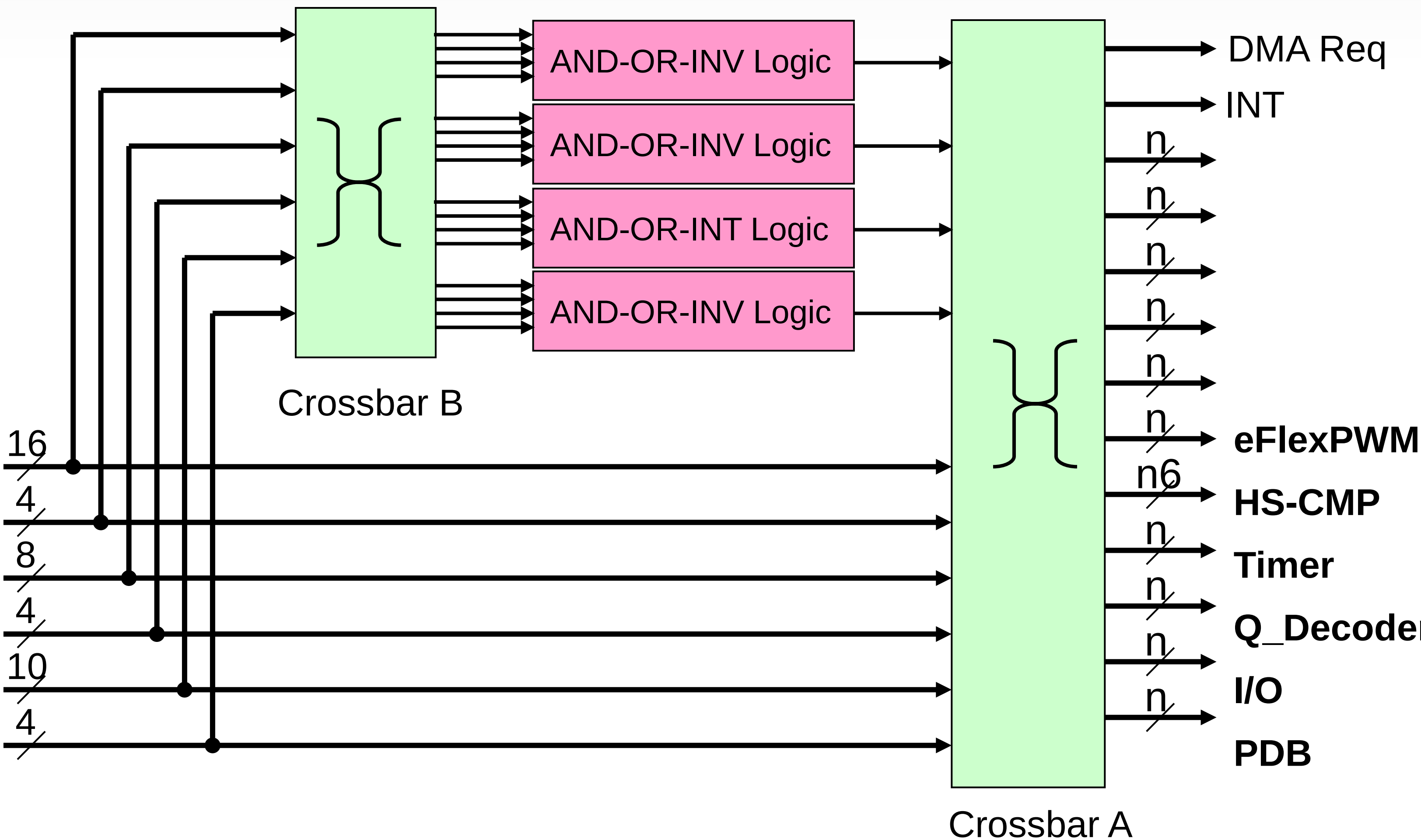
- Flexible signal interconnection among peripherals
- Connects any of 22 signals on left side to the output on right side (multiplexer)
- Total 30 multiplexers
- All multiplexers share the same set of 22 signals
- Increase flexibility of peripheral configuration according to user needs



Crossbar Inter-module Connection - MC56F824x/5x



Crossbar Inter-module Connection - MC56F84xx



Motor control Enablement

Freescale Embedded Software and Motor Control Libraries

Algorithms divided into four sub-libraries:

- **General Function Library (GFLIB)** contains math, trigonometric, look-up table and control functions. These software modules are basic building blocks.
- **Motor Control Library (MCLIB)** contains vector modulation, transformation and specific motor related functions to build digitally controlled motor drives.
- **General Digital Filter Library (GDFLIB)** contains filter functions for signal conditioning.
- **Advanced Control Library (ACLIB)** contain functions to enable building the variable speed AC motor drive systems with field oriented control techniques without position or speed transducer (for Cortex-M4 contain **Back-EMF observer d,q** and **Tracking Observer**).

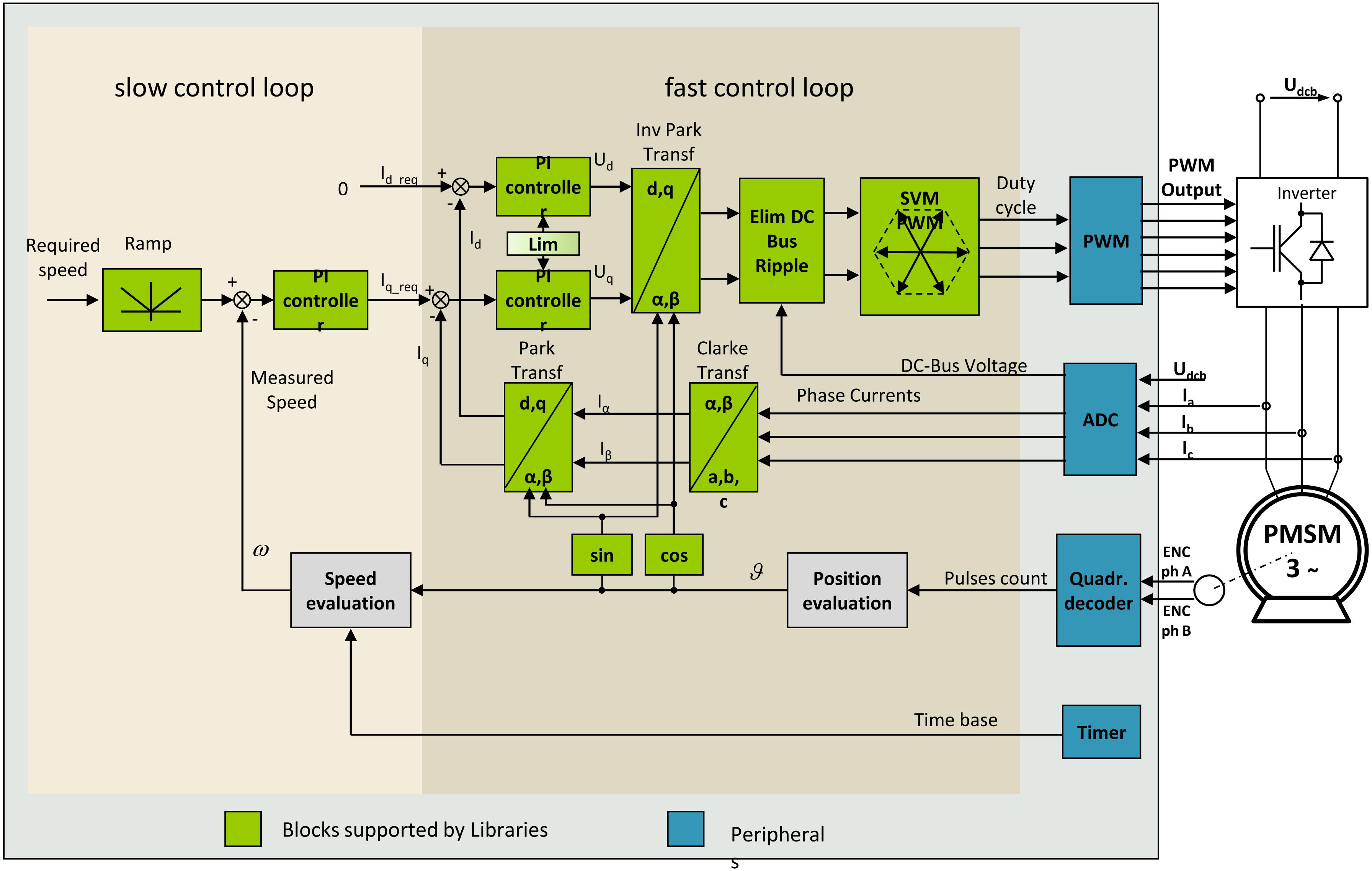
GFLIB - General Function Library	GMCLIB - Motor Control Library	GDFLIB - Digital Filter Library
• Sine, cosine, tangent • Square root • Ramp • Limitation on input signal • Proportional-integral (PI) controller of parallel form	• Clark, inverse clark • Park, inverse park • Vector limitation • DC bus voltage ripple elimination • Standart space vector modulation • PM motor decoupling	• 1^{st} and 2^{nd} order IIR filter • Moving average filter

Develop an Application using Libraries

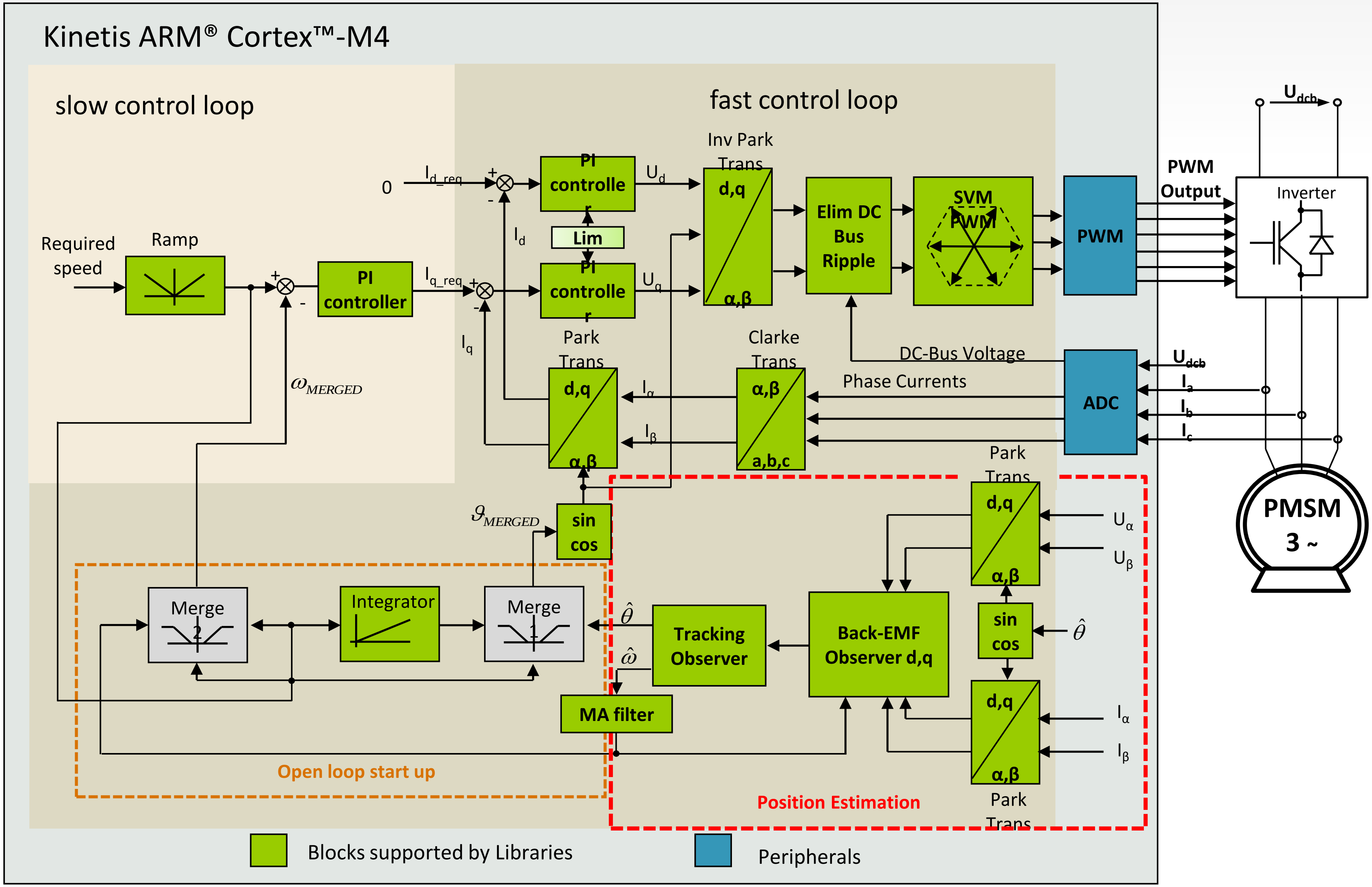
- The coding of the fast control loop of the PMSM vector control using libraries is then limited to peripherals handling and calling of the libraries functions, while passing the addresses of the application structures

```
....
  // Iq current PI controllers
  uDQReq.s32Arg2 = GFLIB_ControllerPIpAW(iDQErr.s32Arg2,&qAxisPI);
  // inverse Park trf for voltages
  GMCLIB_ParkInv(&uAlBeReq,&thRotElSyst,&uDQReq);
 // Elimination of DC bus ripple
 elimDcbRip.s32ArgDcBusMsr  = uDCBus;
 GMCLIB_ElimDcBusRip(&uAlBeReqDCB,&uAlBeReq,&elimDcbRip);
  // Calculation of Standard space vector modulation
  svmSector = GMCLIB_SvmStd(&pwm32,&uAlBeReqDCB);
....
```

PMSM Vector Control w/ Encoder built on Libraries

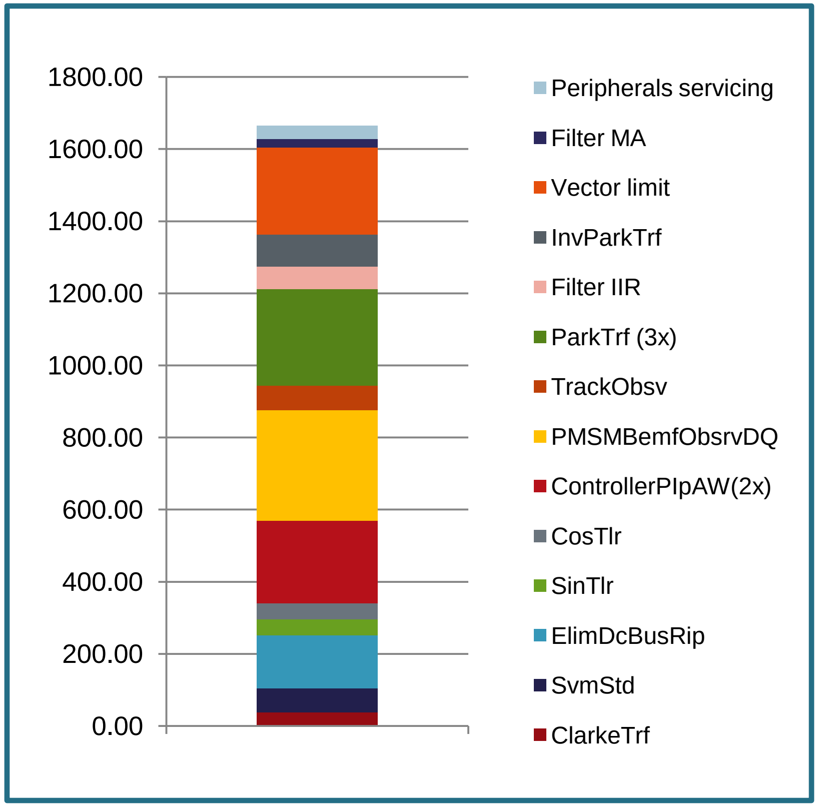


MSM Sensorless Vector Control built on Libraries

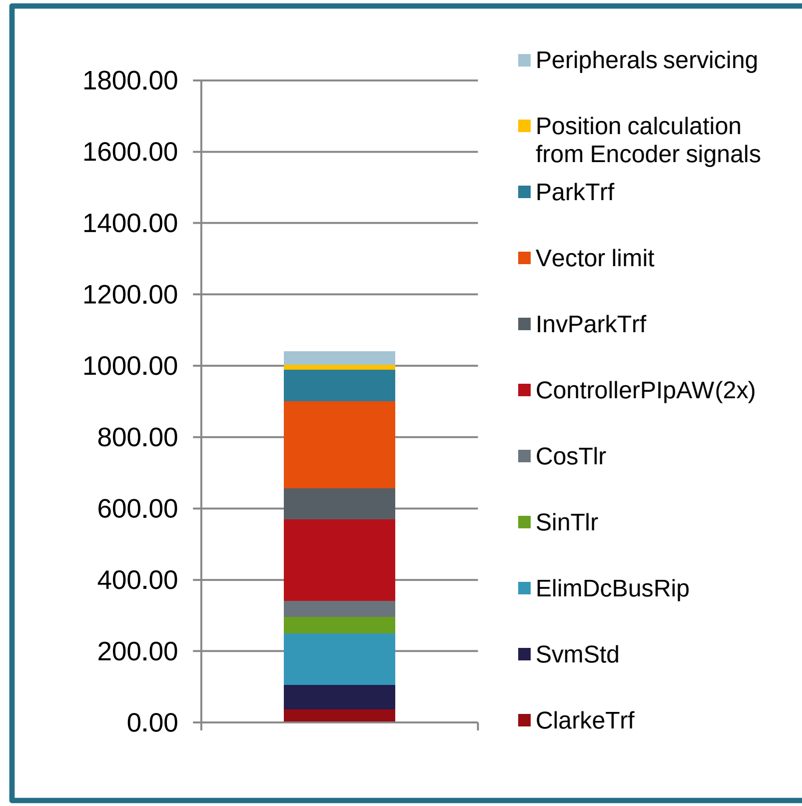


Machine Cycles – Fast Control Loop of PMSM FOC

Sensorless Solution



Encoder based solution



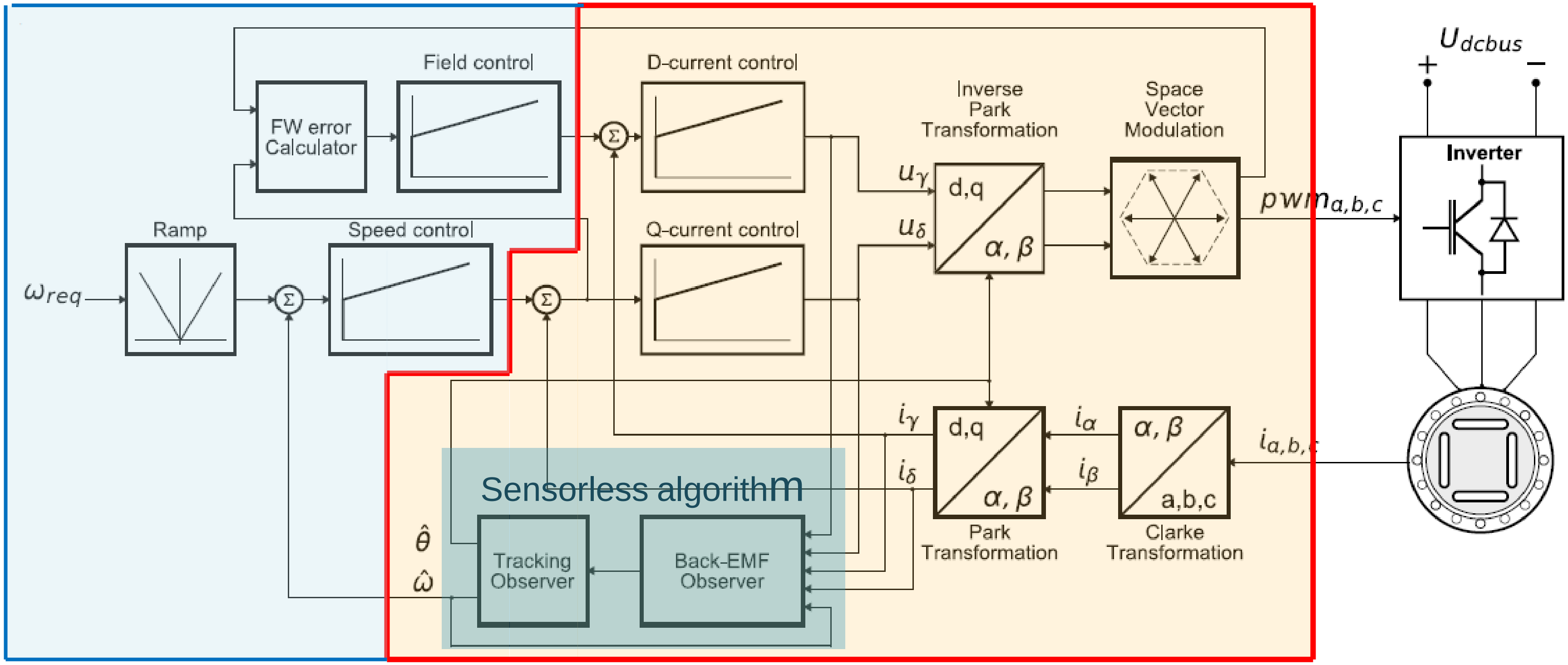
PMSM Sensorless Vector Control Algorithm

Slow (speed) control loop

- Executed in 1-5msec loop
- represents just like 1% of the CPU performance, <u>neglected for the benchmark</u>

Fast (current) control loop

- Executed in 25-200usec loop
- CPU load should be <40%
- critical for sensorless FOC, <u>target of the benchmark</u>



ARM Cortex-M0+/M4 Comparison

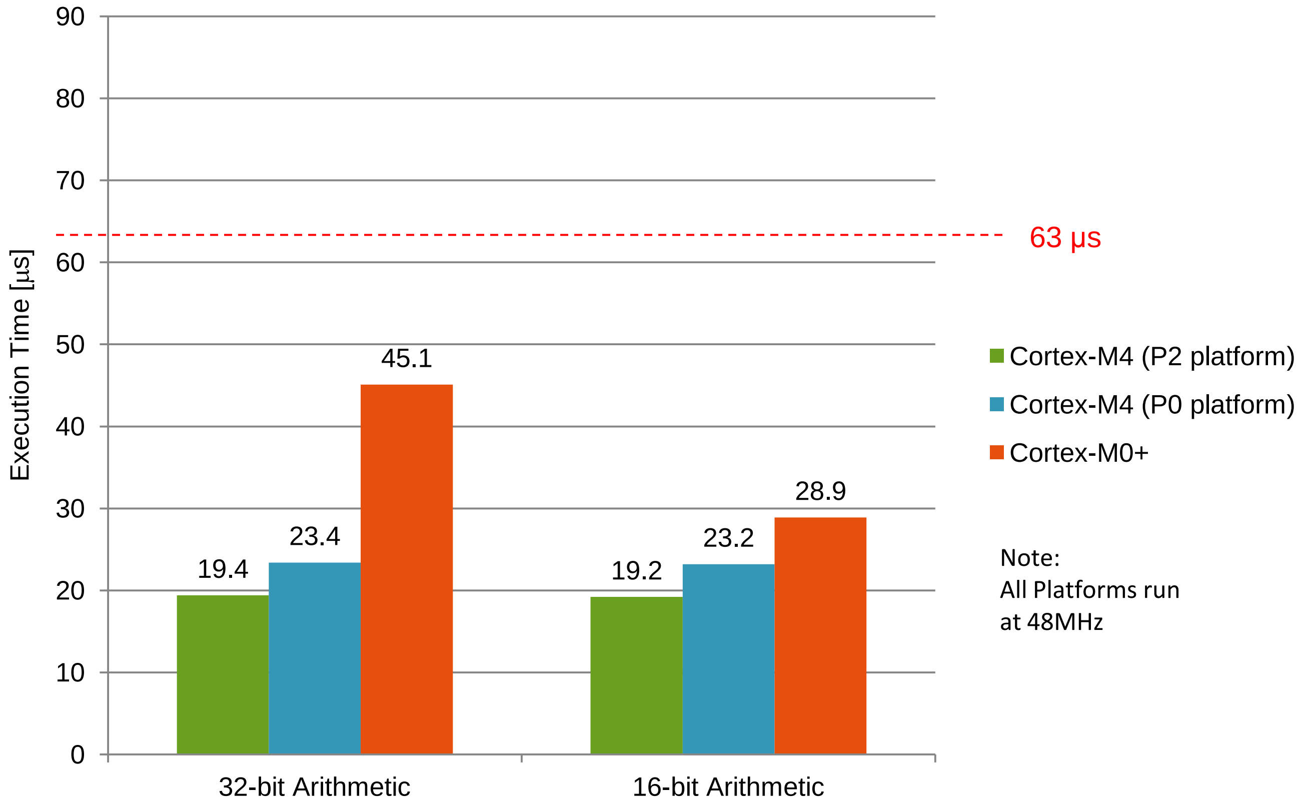
- Results for Sensored PMSM Vector Control Algorithm

Core	CPU Clock Cycles	Execution Time in µs at 48 MHz CPU clock	Code size (Bytes)
32-bit arithmetic 32 x 32 = 64			
Cortex-M0+	2385	49.6	2648
Cortex-M4 (50 MHz platform)	1122	23.4	2032
Cortex-M4 (100 MHz platform)	931	19.4	2032
16-bit arithmetic 16 x 16 = 32, 32 x 32 = 32 (result 32 LSB)			
Cortex-M0+	1938	40.3	1990
Cortex-M4 (50 MHz platform)	1115	23.2	1848
Cortex-M4 (100 MHz platform)	925	19.2	1848

freescale™

ARM Cortex-M0+/M4 Comparison

- Results for Sensored PMSM Vector Control Algorithm



Note:
All Platforms run
at 48MHz

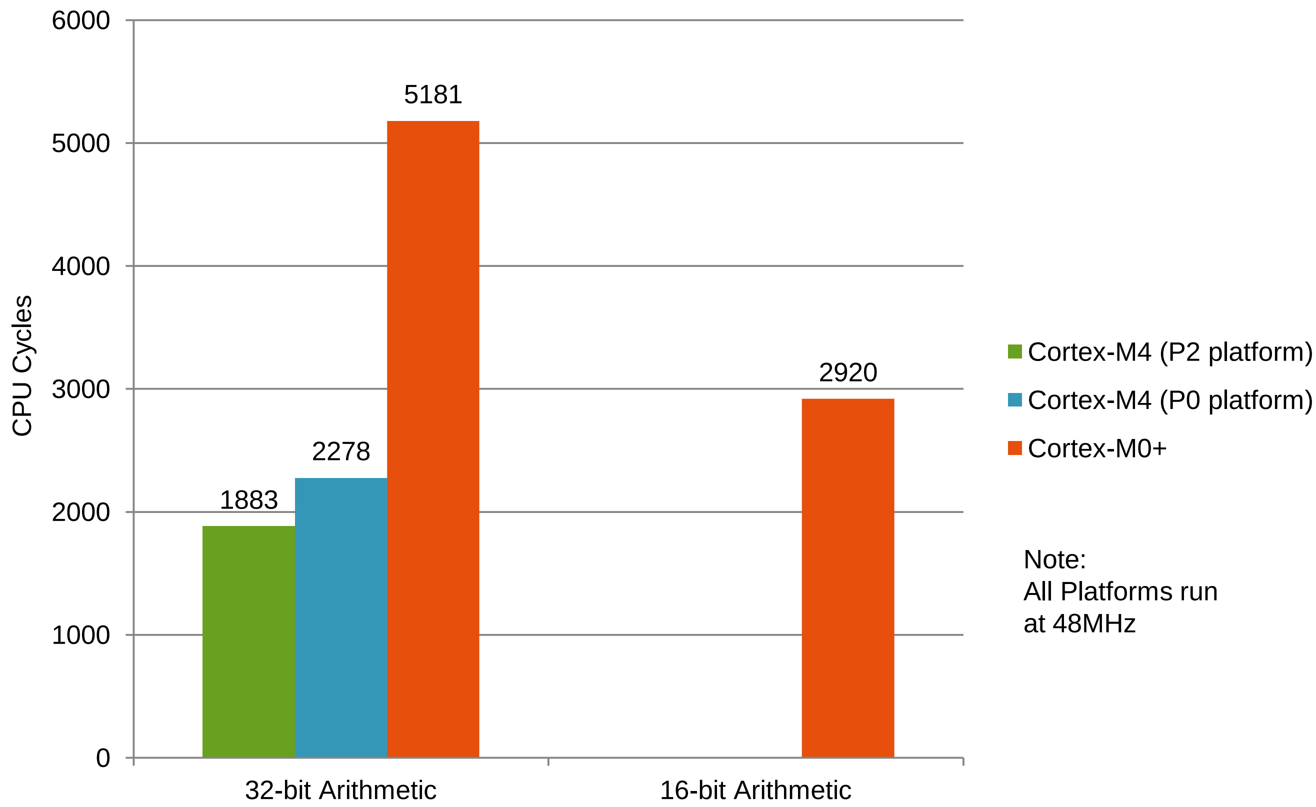
ARM Cortex-M0+/M4 Comparison

- Results for Sensorless PMSM Vector Control Algorithm

Core	CPU Clock Cycles	Execution Time in μs at 48 MHz CPU clock	Code size (Bytes)
32-bit arithmetic 32 x 32 = 64			
Cortex-M0+	5181	108	4202
Cortex-M4 (50 MHz platform)	2278	47.4	3382
Cortex-M4 (100 MHz platform)	1883	39.2	3382
16-bit arithmetic 16 x 16 = 32, 32 x 32 = 32 (result 32 LSB)			
Cortex-M0+	2920	60.8	4030

ARM Cortex-M0+/M4 Comparison

- Results for Sensorless PMSM Vector Control Algorithm

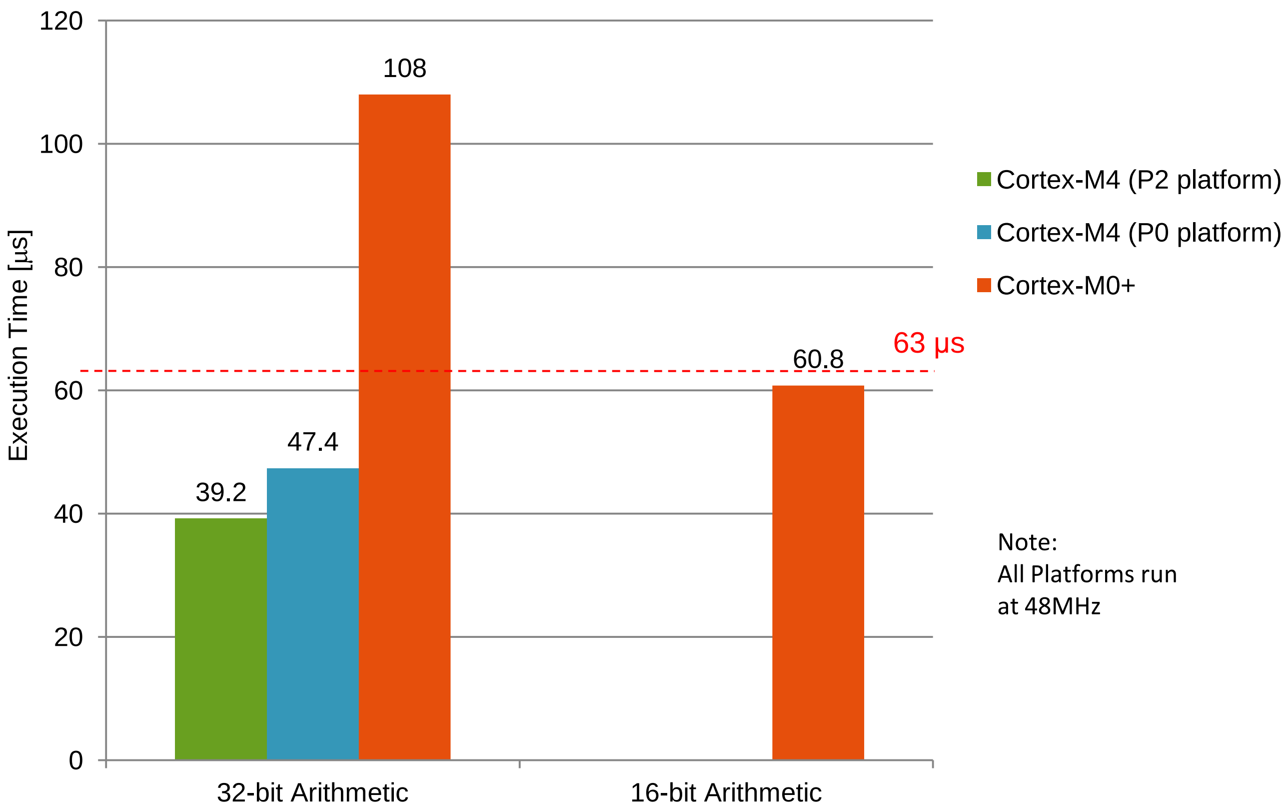


Note:
All Platforms run
at 48MHz

freescale™

ARM Cortex-M0+/M4 Comparison

- Results for Sensorless PMSM Vector Control Algorithm



Note:
All Platforms run
at 48MHz

Results for sensorless PMSM Vector Control Algorithm

	Cycles	Exec. Time [ms]
Cortex-M4 50MHz RAM (Kinetis K)	1843	36.9
Cortex-M4 100 MHz RAM (Kinetis K)	1843	18.4
Cortex-M4 50MHz FLASH (Kinetis K)	1979	39.6
Cortex-M4 100MHz FLASH (Kinetis K)	2628	26.3
DSC Hawk V3	1612	16.1

ARM Cortex-M0+/M4 Comparison

Summary

- The Cortex-M0+ is slower by 175% than Cortex-M4 using 32-bit arithmetic
  - This is due to missing 32-bit instruction
- The Cortex-M0+ cannot run Sensorless PMSM FOC in 32-bit arithmetic every PWM period (65μs)
- The Cortex-M0+ is on the limit to run Sensorless PMSM FOC in 16-bit arithmetic every PWM period (65μs). But it can run the algorithm every second period (130μs).

Plain Cortex M0+ vs HW SQRT and Divide

- New KV10 75 MHz devices include hardware SQRT and Divide to offload the CPU from these operations.
- Biggest cycle consumer for CM0+ CPU

Benchmark

- The sensorless PMSM application calculates 3 DIV and 1 SQRT in fast current loop.
- 2xDIV in dc bus ripple elimination
- 1xDIV in ArcusTangent (used in sensorless observer)
- 1xSQRT in Limitation

- **SW Divide = 180 to 360 cycles/divide**
- **HW Divide = 20 cycles/divide**
-
- **Optimized_SW_SQRT = 201 cycles/SQRT**
- **HW_SQRT = 13 cycles/SQRT**

HW SQRT and DIV improve up to 26% performance

Tuning the application constants with help of FreeMASTER

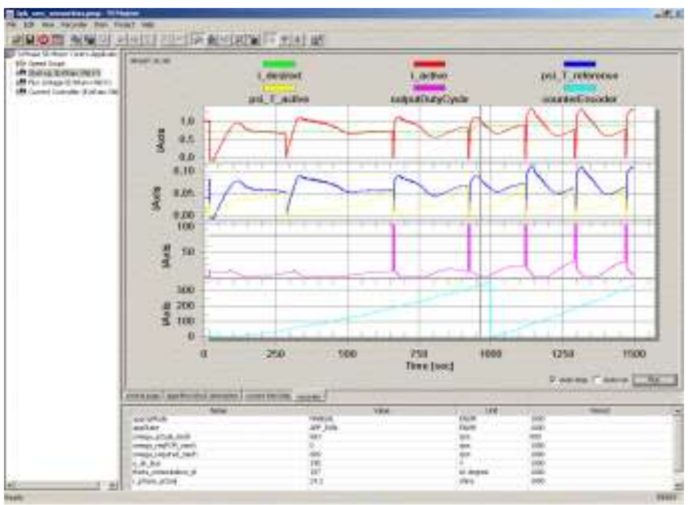
- The most challenging task for the developer is the setting of the application constants, sometimes trial-error method must be used when the system (drive) parameters are difficult to identify:
  - ***P*** and ***I*** constants of the regulators
  - Filter constants
  - Constants of the position estimation algorithms
  - Tuning the merging process when switching from the open loop start-up to full sensorless mode

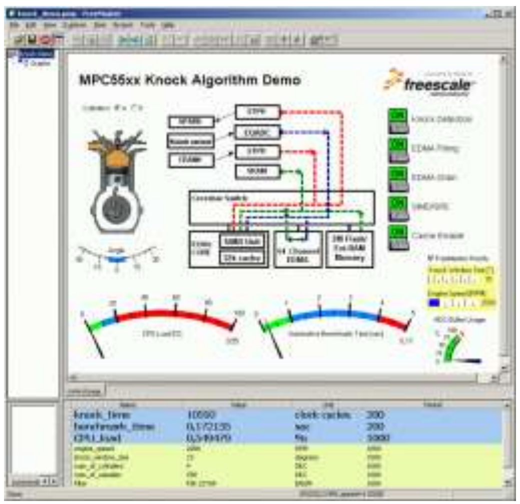
freescale™

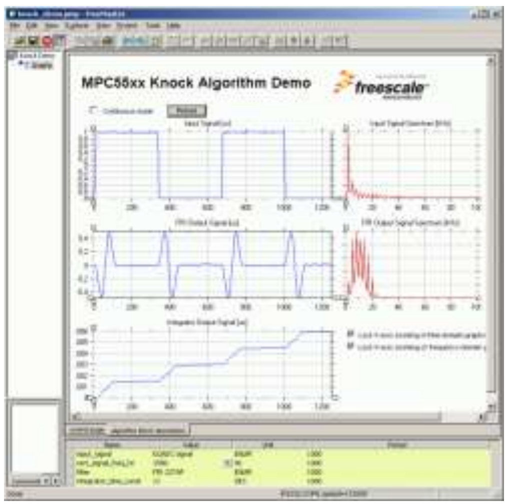
What is FREEMASTER ?

- Real-time Monitor
- Graphical Control Panel
- Demonstration Platform

FOR YOUR EMBEDDED APPLICATION







FREEMASTER

as a Real-time Monitor

FREEMASTER as a Real-time Monitor

- Connects to an embedded application
  - SCI, UART
  - JTAG/EOnCE (56F8xxx only)
  - BDM (HCS08, HCS12 only)
  - CAN Calibration Protocol
  - Ethernet, TCP/IP
  - Any of the above remotely over the network
- Enables access to application memory
  - Parses ELF application executable file
  - Parses DWARF debugging information in the ELF file
  - Knows addresses of global and static C-variables
  - Knows variable sizes, structure types, array dimensions etc.
- Serial Communication Driver
  - Completely Interrupt-Driven – LONG INTERRUPT
  - Mixed Interrupt and Polling Modes – SHORT INTERRUPT
  - **Completely Poll-Driven → preferred mode, run typically in main() loop**

FREEMASTER as a Real-time Monitor

Application control and monitor

Live graphs, variable watches, and graphical control page

Real-time operation monitor

FREEMASTER as a Real-time Monitor

- Variable Transformations
  - Variable value can be transformed to the custom unit
  - Variable transformations may reference other variable values
  - Values are transformed back when writing a new value to the variable

- Application Commands
  - Command code and parameters are delivered to an application for arbitrary processing
  - After processed (asynchronously to a command delivery) the command result code is returned to the PC

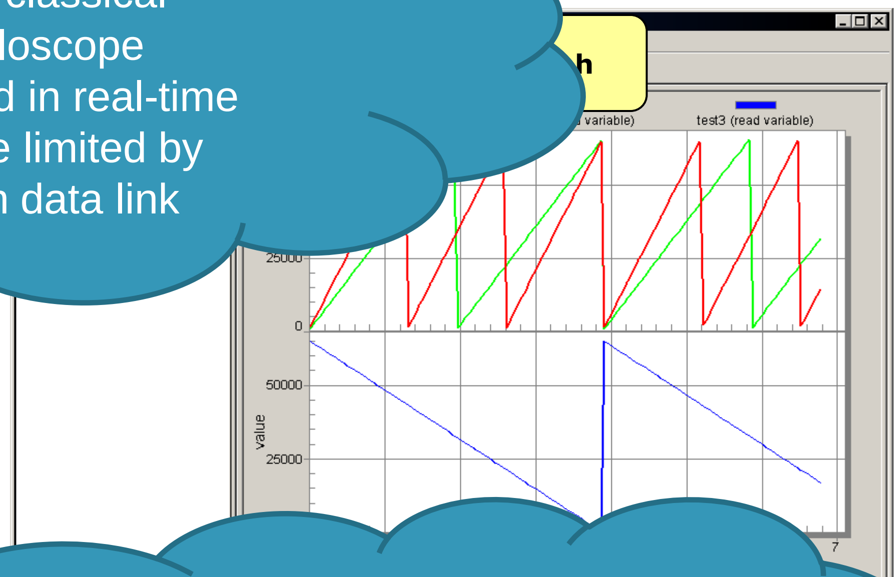
- Ability to protect memory regions
  - Describing variables visible to FreeMASTER
  - Declaring variables as read-write to read-only for FreeMASTER
  - the access is guarded by the embedded-side driver

freescale™

FREEMASTER

- Displays the variable values in various formats:
  - **Text, tabular grid**
    - variable name
    - value as hex, dec or number
    - min, max values
    - number-to-text labels
  - **Real-time waveforms**
    - up to 8 variables simultaneously in an oscilloscope-like graph
  - **High-speed recorded data**
    - up to 8 variables in on-board memory **transient recorder**

- similar to the classical hardware oscilloscope
- variables read in real-time
- sampling time limited by communication data link



- variables recorded by the embedded-side timer periodic ISR
- after requested number of samples data stored in Recorder buffer
- sample very fast actions
- buffer download can be defined

FREEMASTER as a Real-time Monitor

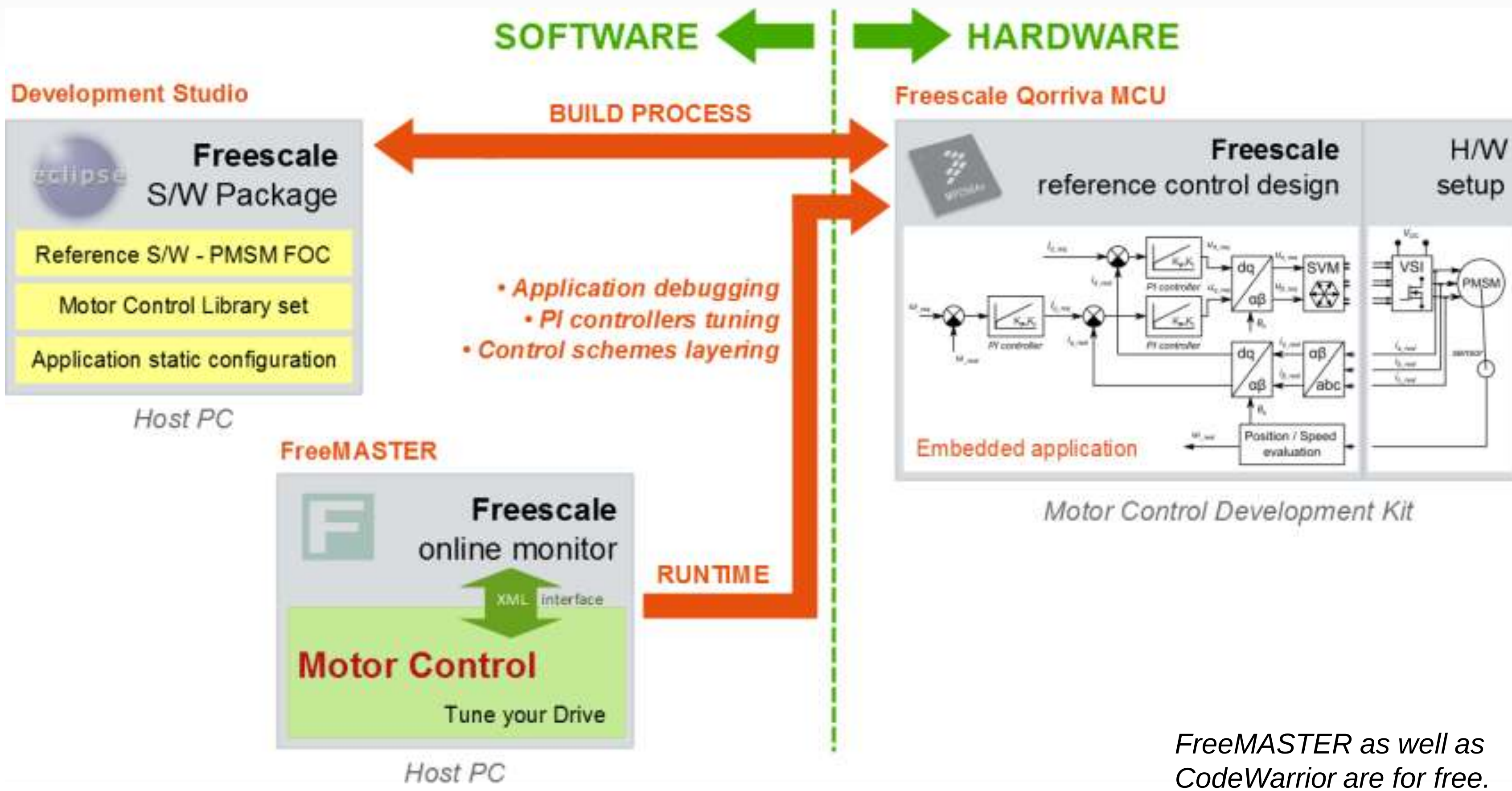
Highlights:

- FreeMASTER helps developers to debug or tune their applications
- Replaces debugger in situations when the processor core can not be simply stopped (e.g. motor control)
- Recorder may be used to visualize transitions in near 10-us resolution

freescale™

Motor Control Application Tuning Tool

Software Concept



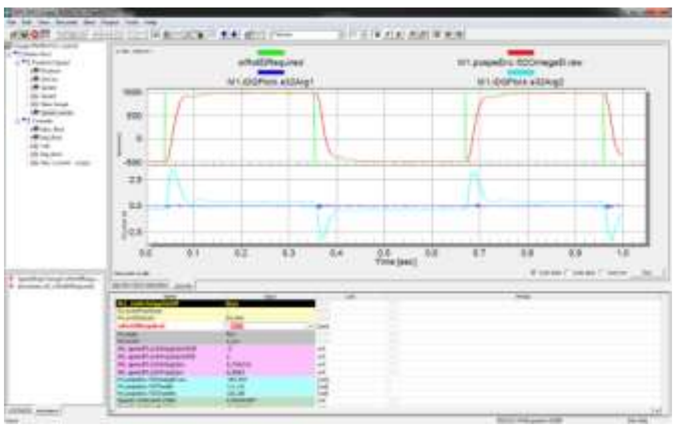
FreeMASTER as well as CodeWarrior are for free.

www.freescale.com/FreeMASTER
www.freescale.com/CodeWarrior

Motor Control Development Kit Series – Content

Out-of-the-box experience offers:

- Complete schematics of the Development Kit HW.
- Complete source code of the Development Kit SW application
- Math and Motor Control libraries (MCLib) in object code
- FreeMASTER & MCAT interface to easy application visualization / control
- Extensive documentation including User guide, Quick Start Guide and Fact sheet.

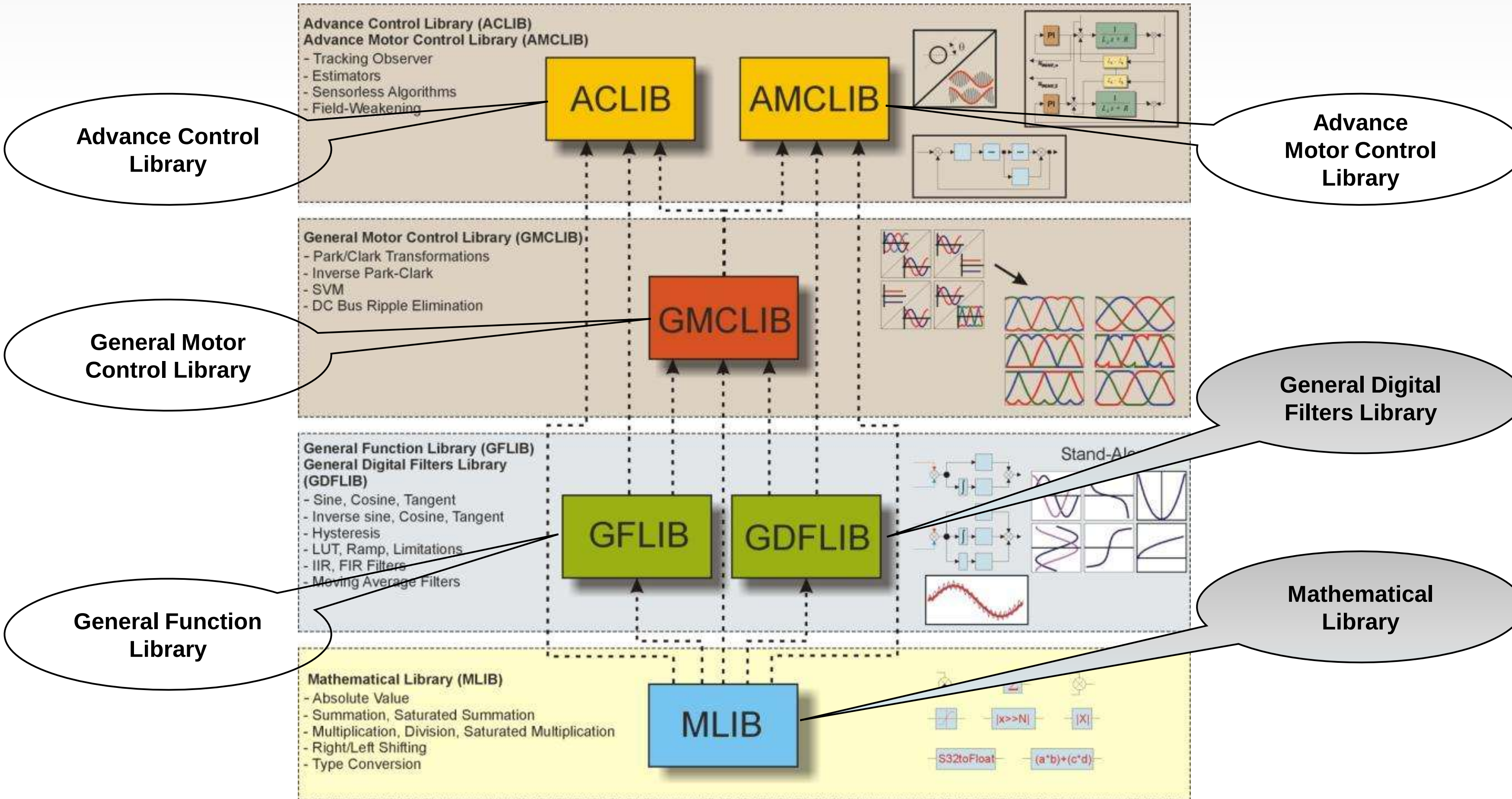


FreeMASTER Scope

FreeMASTER HTML based Control Page

www.freescale.com/AutoMCDevKits

Math and Motor Control Library Set

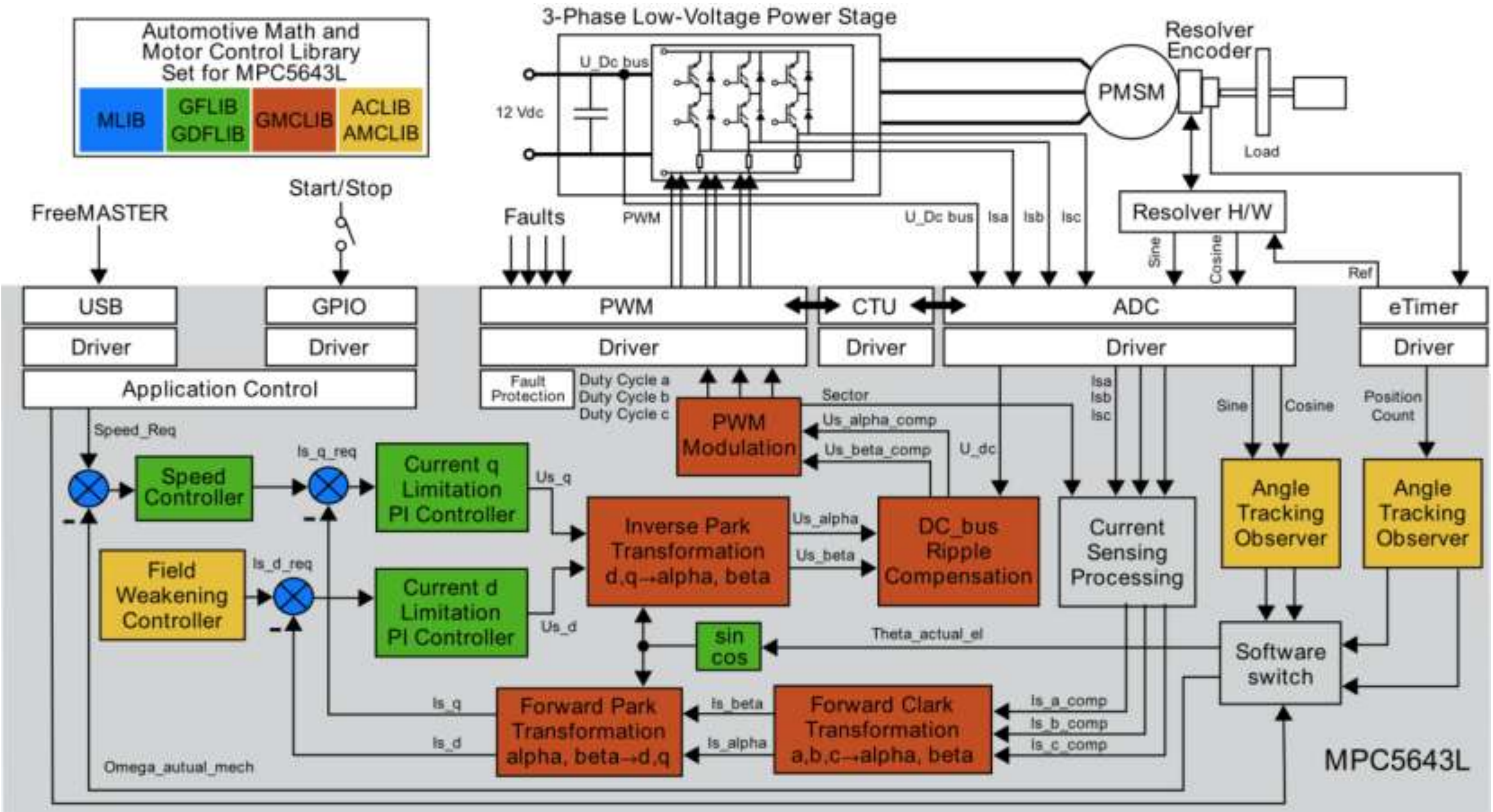


www.freescale.com/AutoMCLib

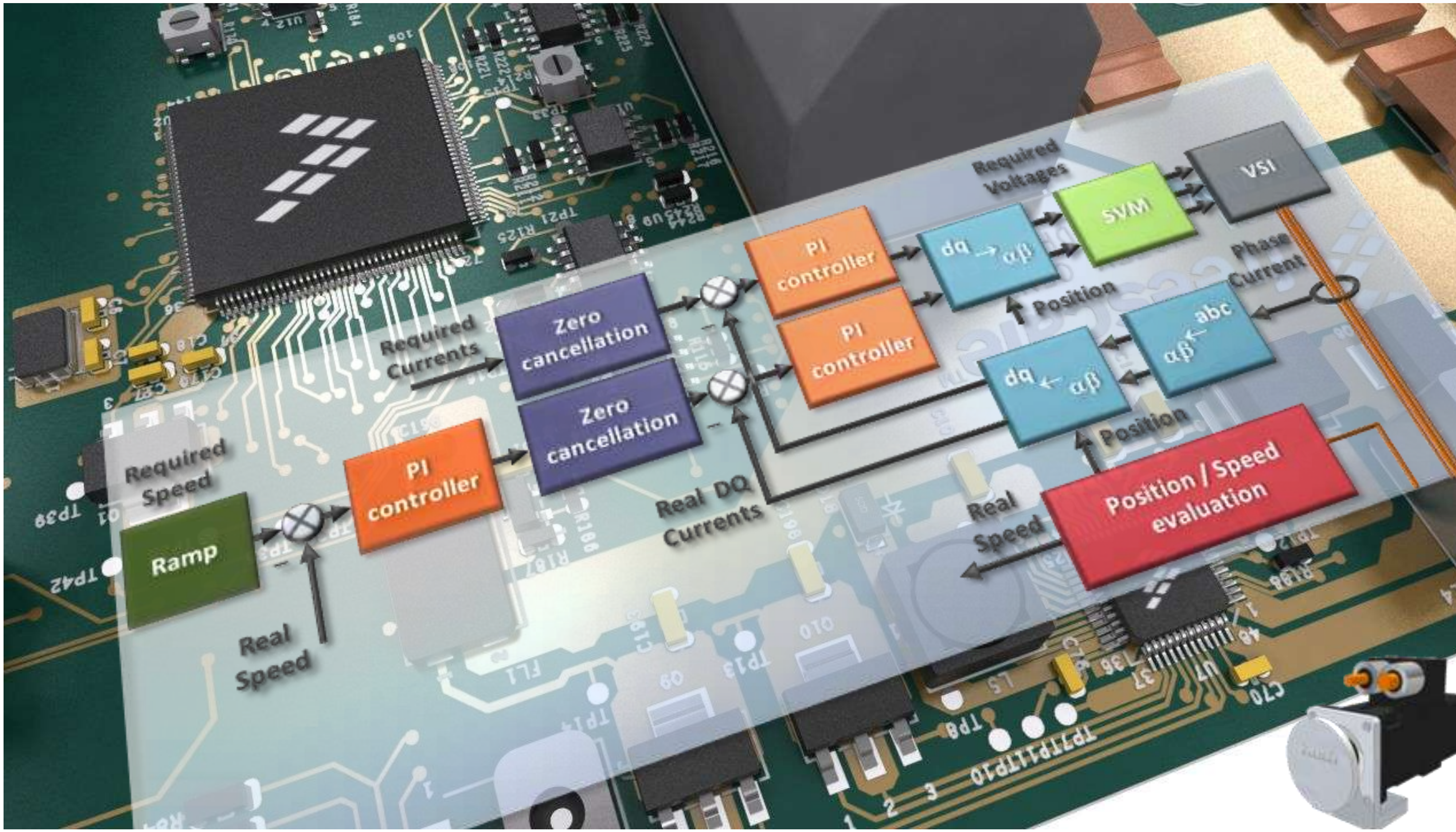
freescale™

PMSM Field Oriented Control

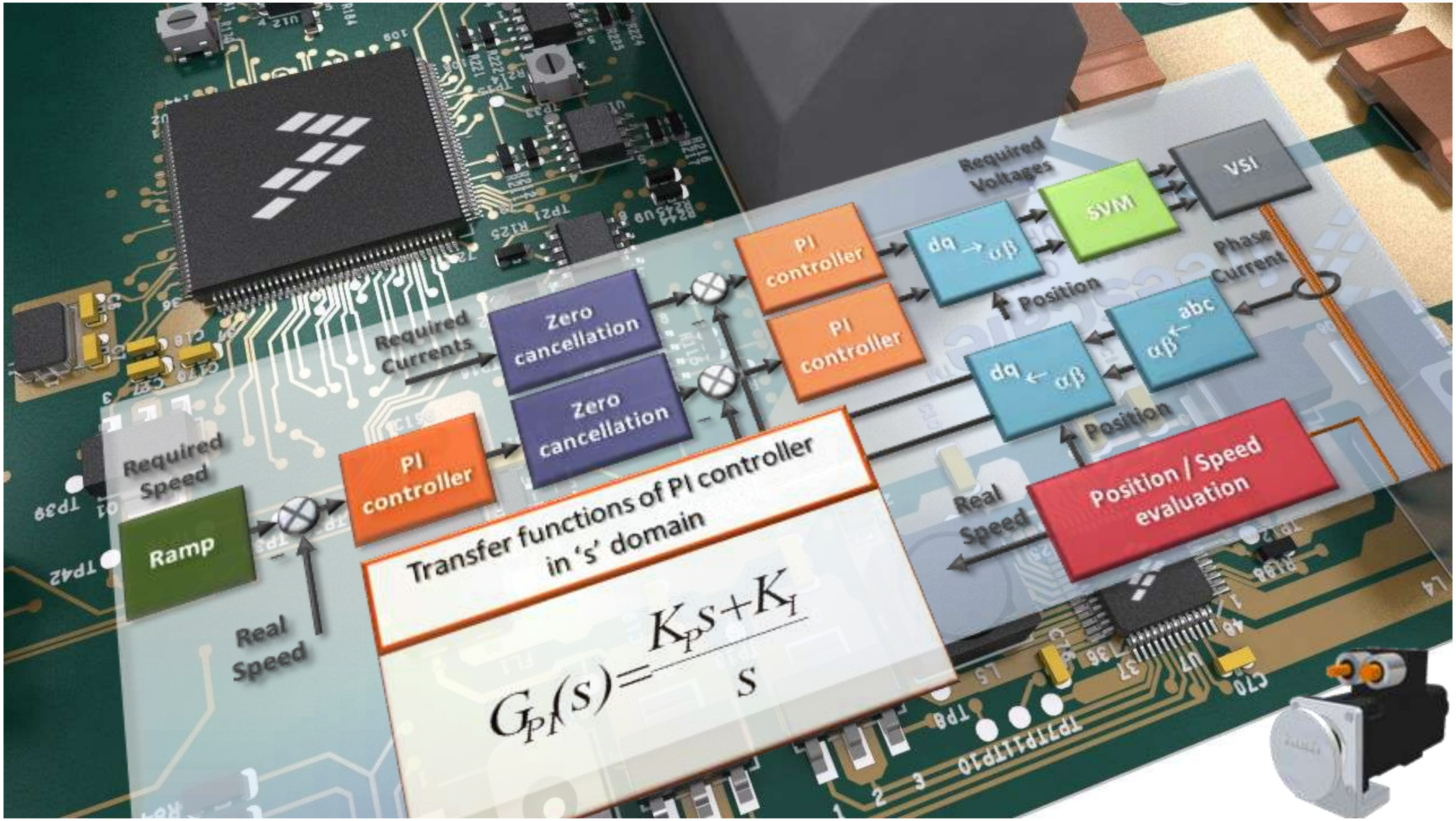
MCLib Application Example for MPC5643L Development Kit



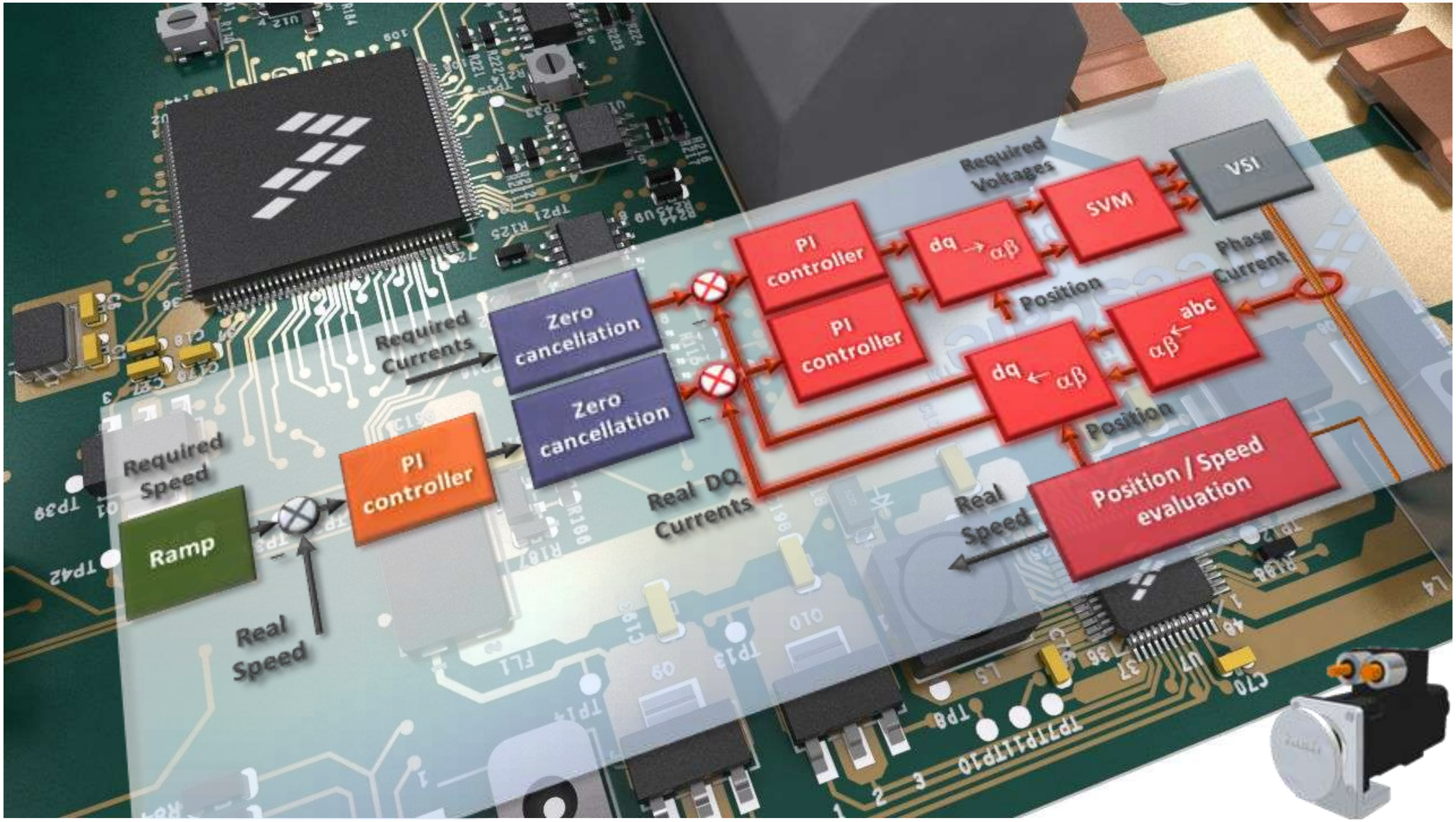
Field Oriented Control



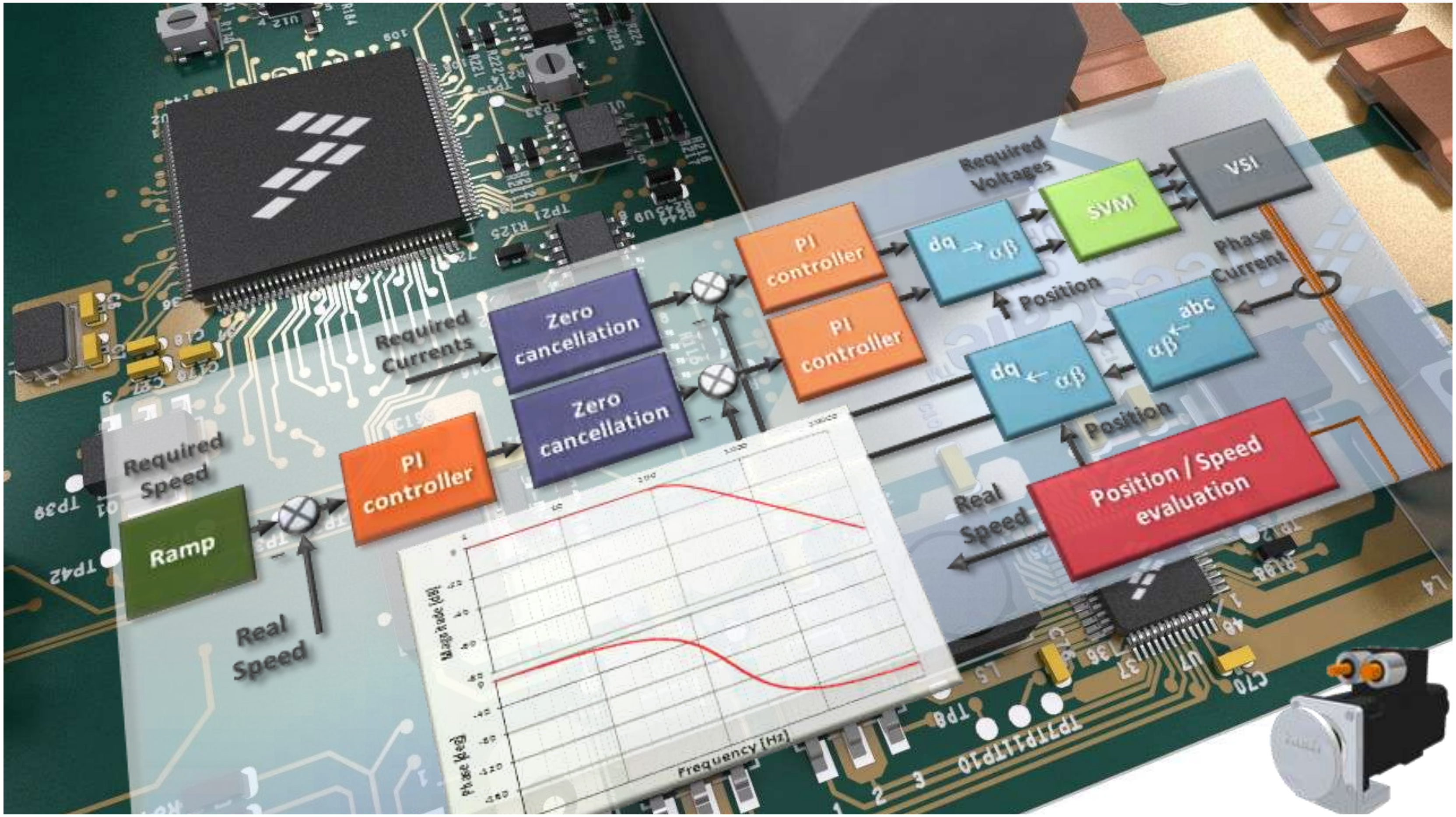
PI Controllers



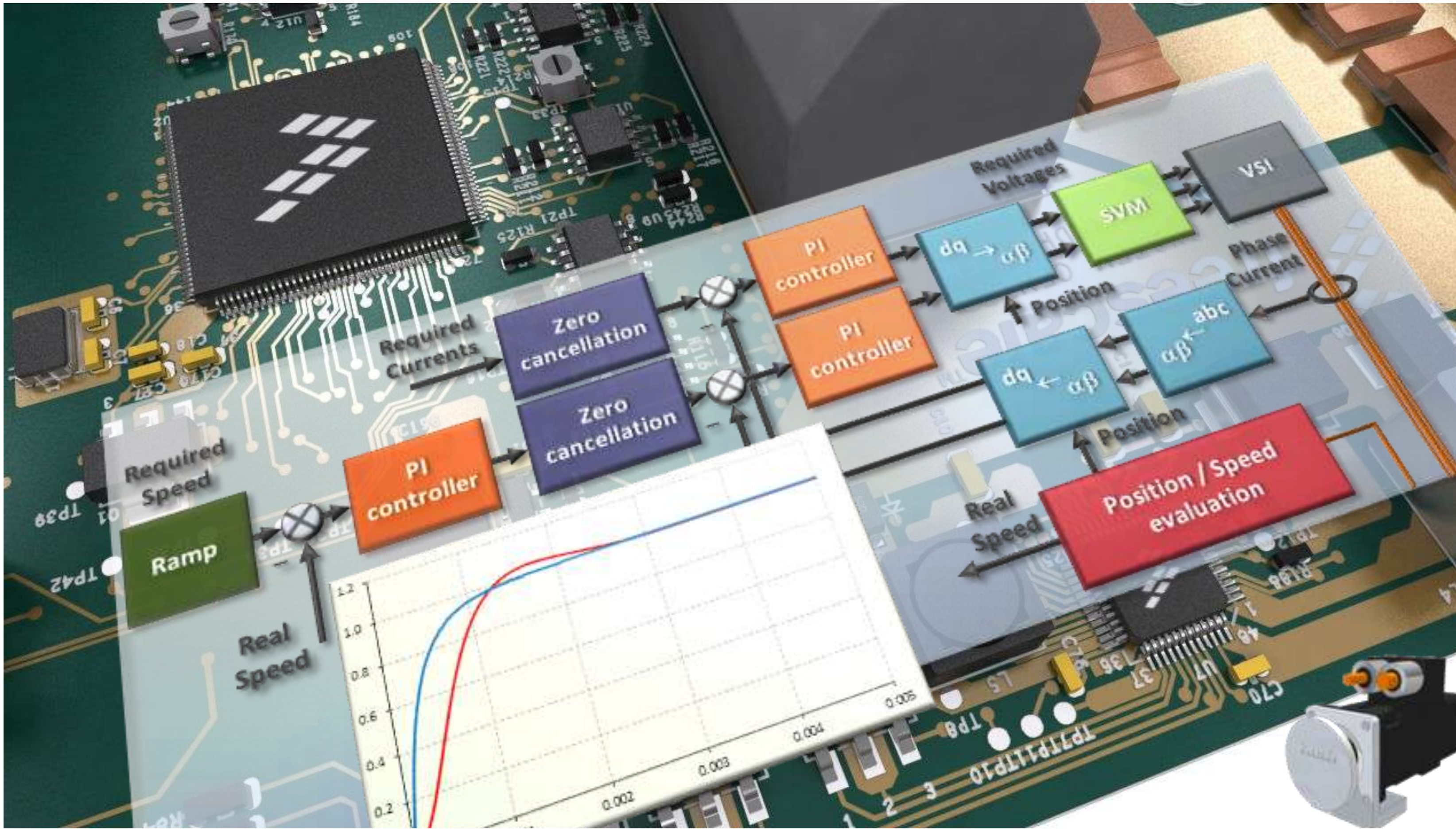
Current Control Loop



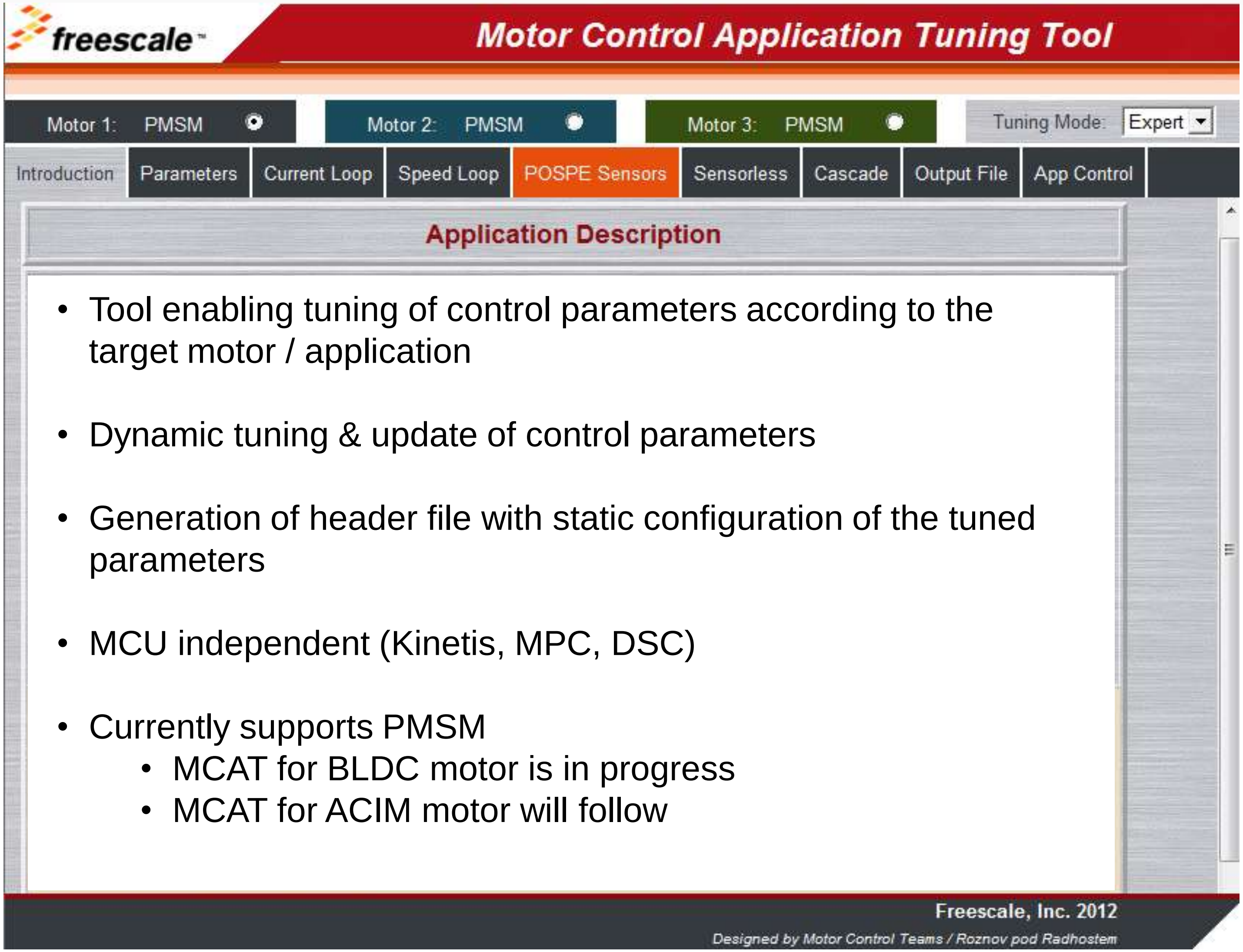
Control Loop Bandwidth



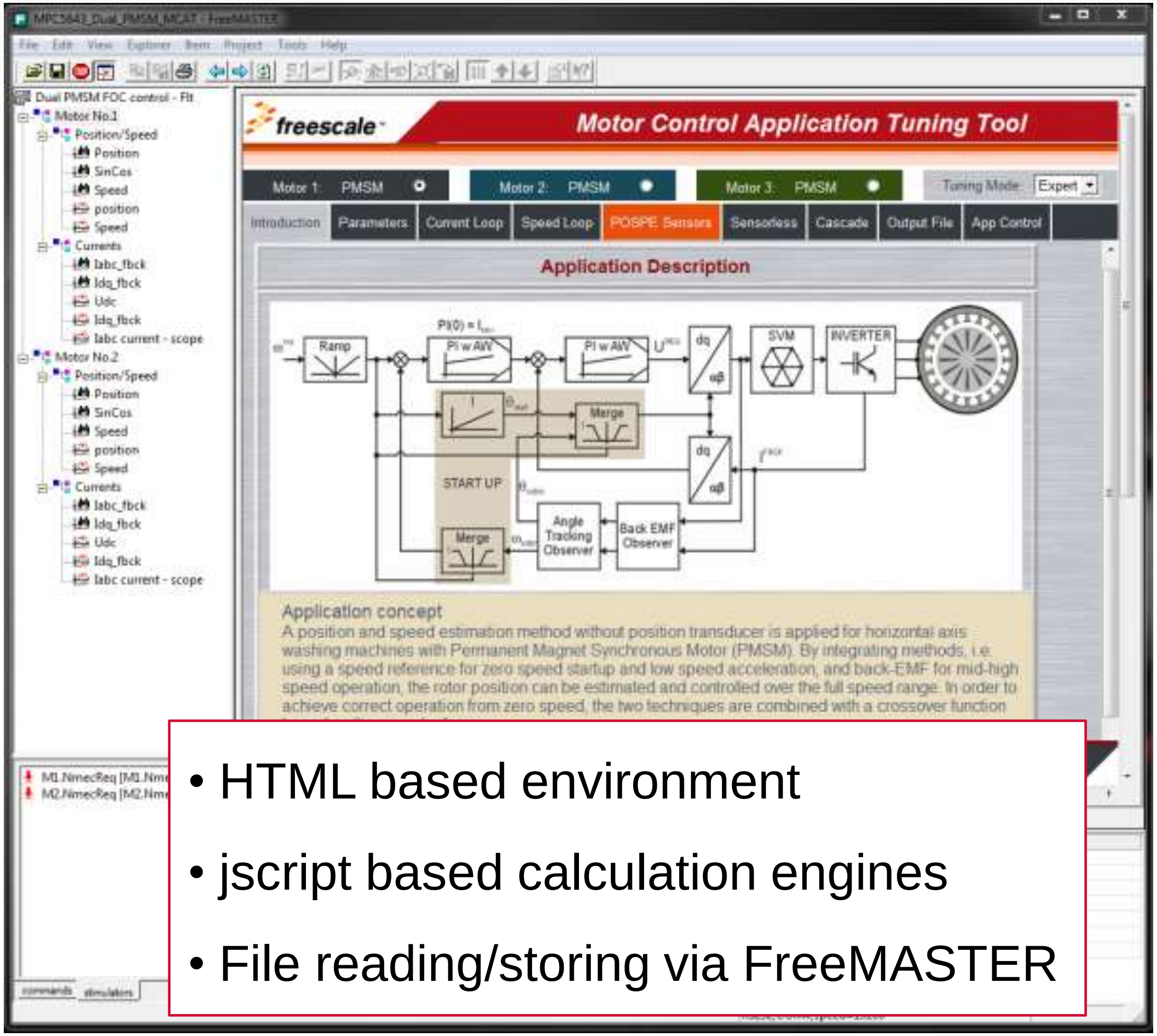
Speed Overshoot



MCAT Introduction & Features



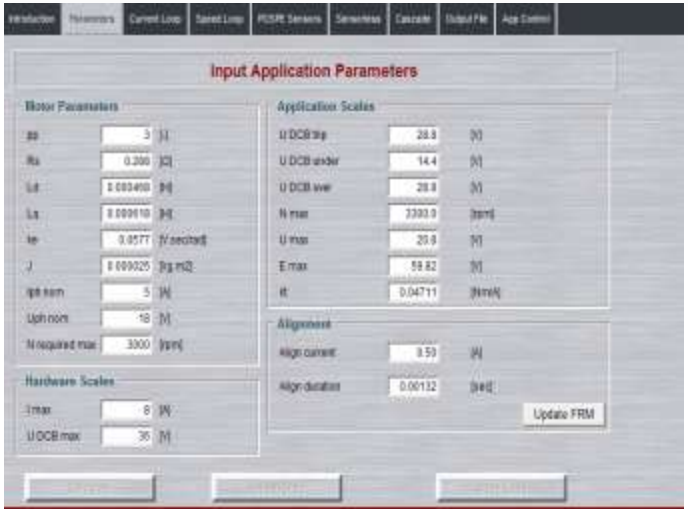
FreeMASTER with MCAT



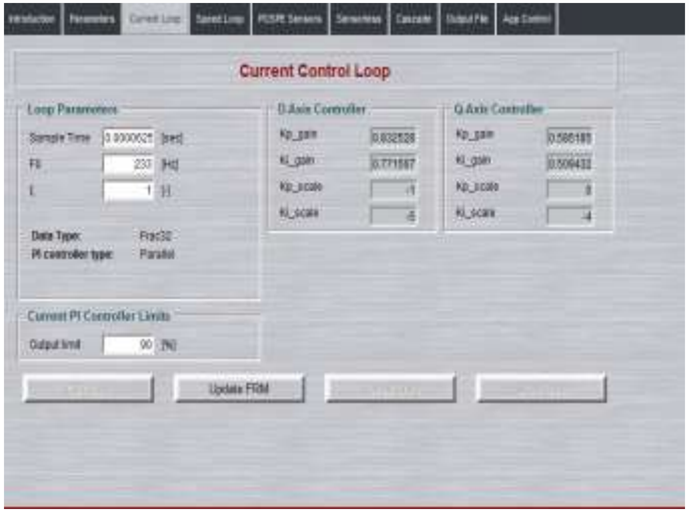
- HTML based environment
- jscript based calculation engines
- File reading/storing via FreeMASTER

Steps to Tune the Current Loop

1. Parameter Setting-Up

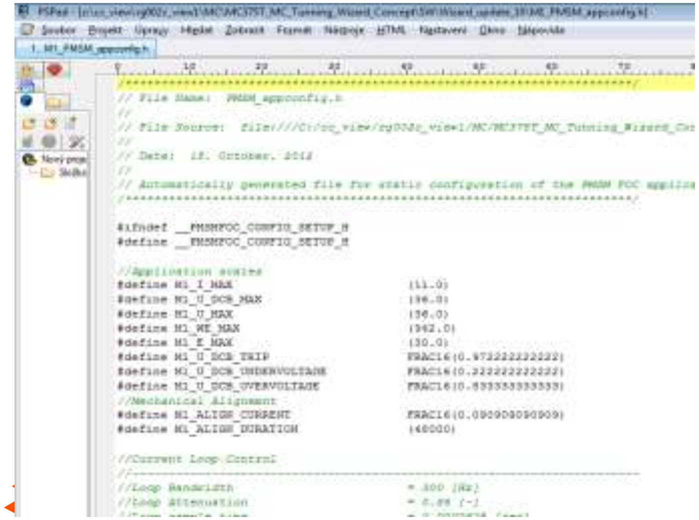


2. Control Loop Tuning

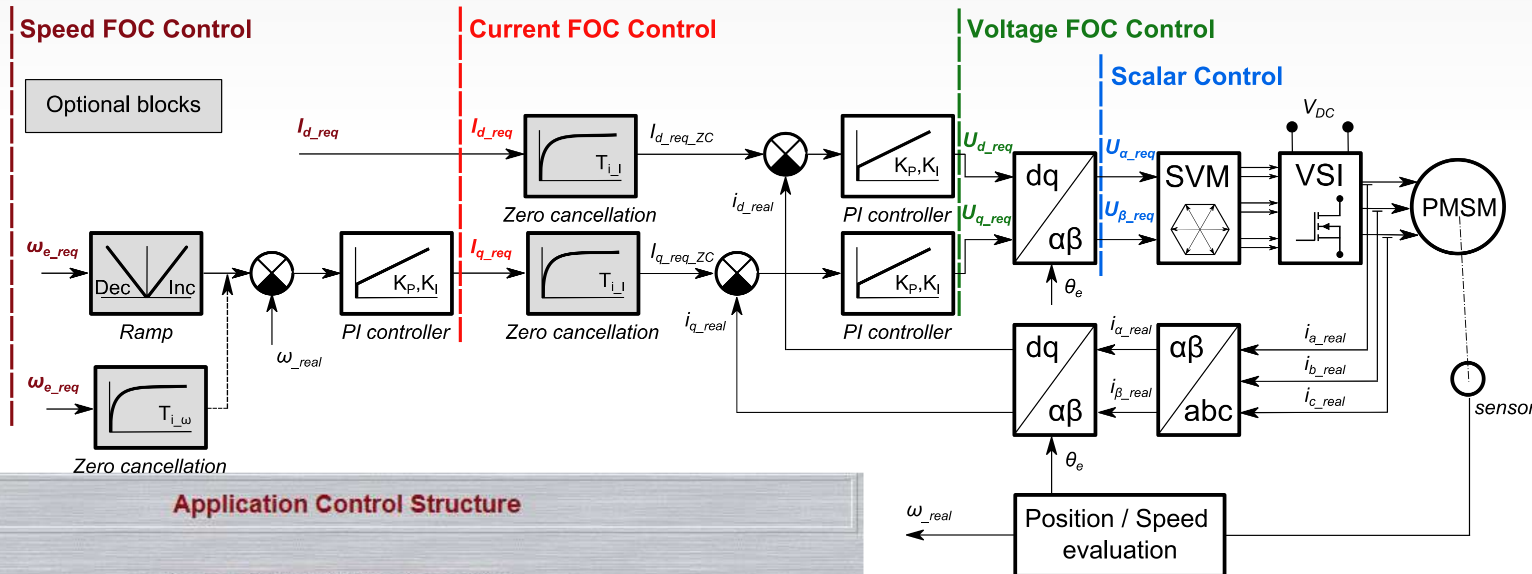


3. Output Control Constant Preview

4. Generated .h file



MCAT Control Structure Selector



Open loop control
no need any current, position or speed feedback

Voltage control – position required
no need any current and speed feedback

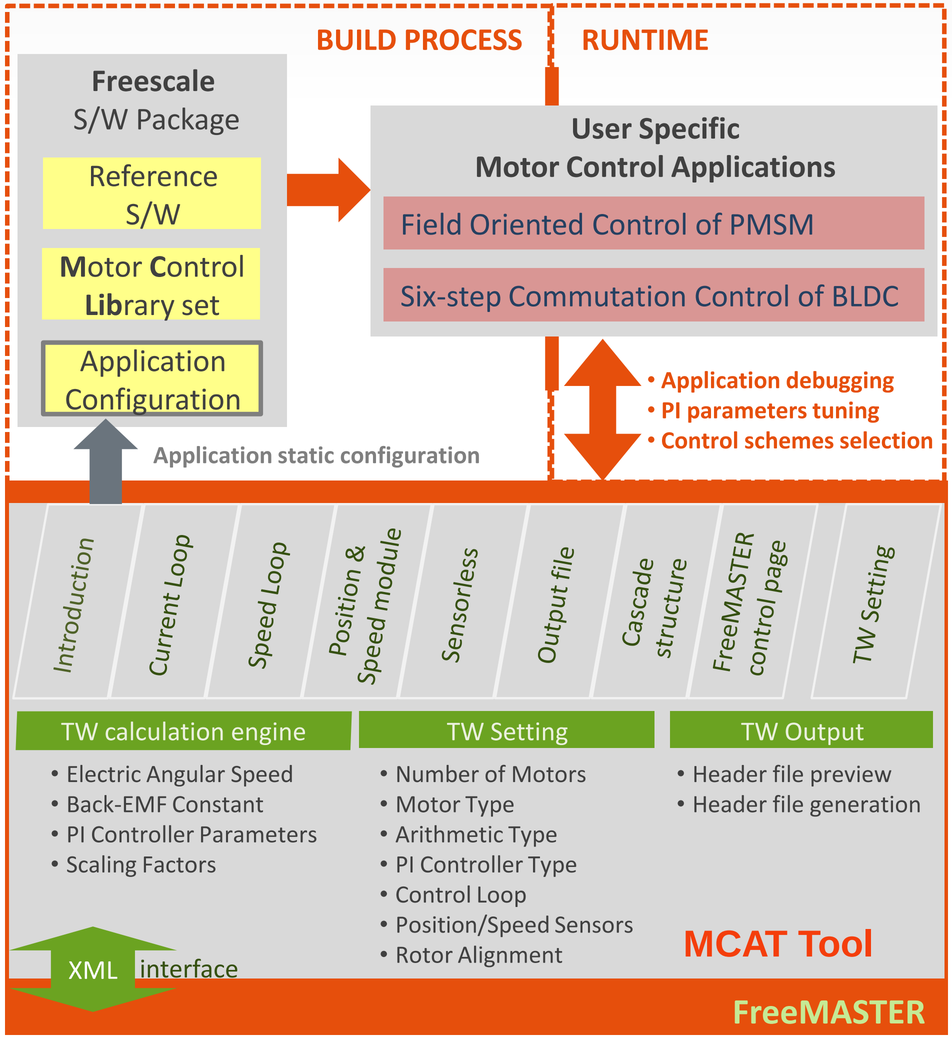
Current control – current, position required
no need any speed feedback

Speed control - current, position and speed required

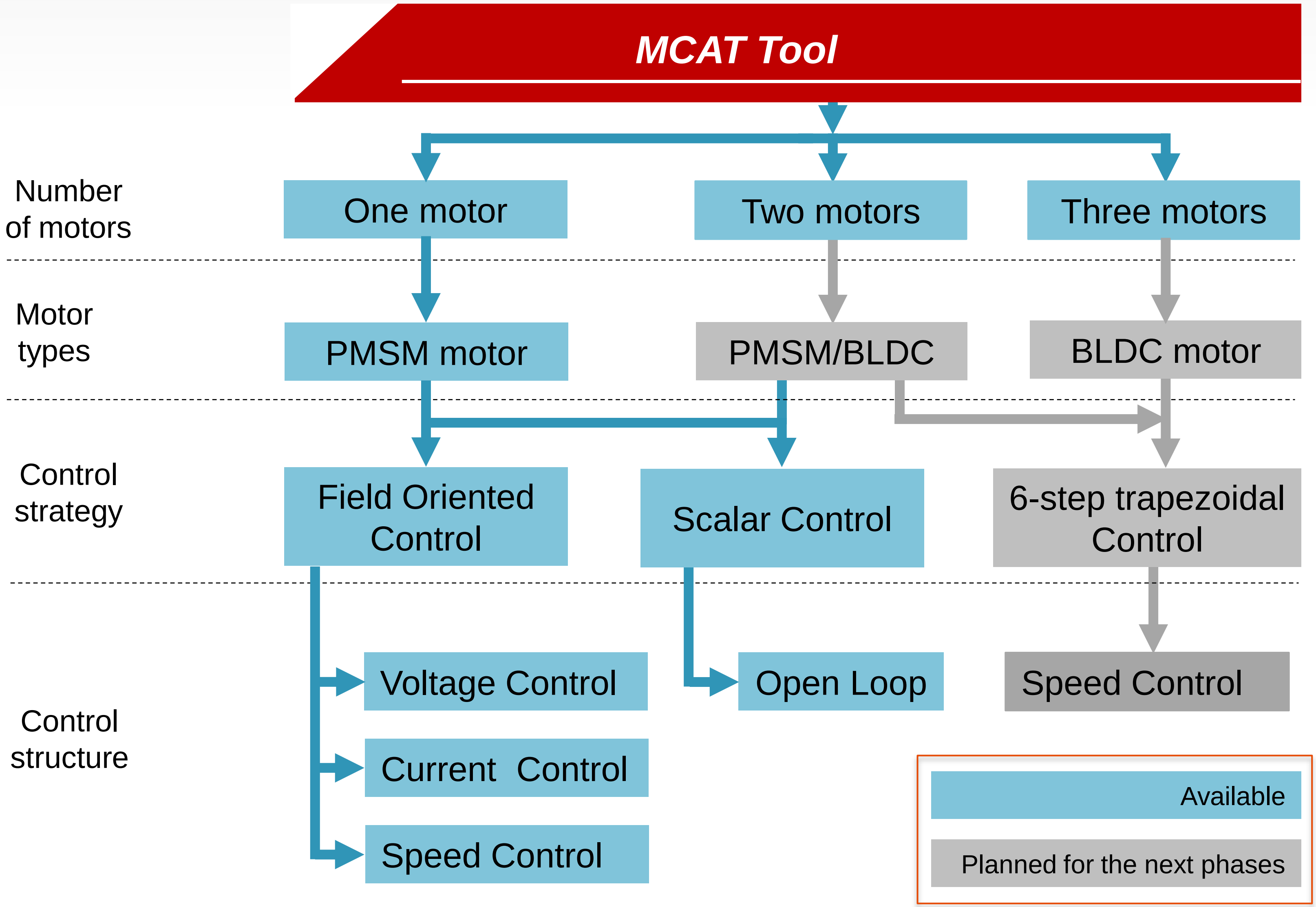
MCAT Tab Menu Summary

- *Introduction* - user defined basic application description
- *Parameters* - obligatory input motor and application parameters
- *Current Loop* - inner control loop implemented as parallel or recurrent PI controller with optional zero cancelation compensation in feed-forward path
- *Speed Loop* - outer control loop implemented as parallel or recurrent PI controller with optional speed ramp or zero cancelation in feed-forward path filter of speed feedback
- *Position & Speed Module* - selection among several sensor type
  - resolver ATO, encoder ATO, encoder ETIMER,
- *Sensorless Module* - setting of BEMF observer and tracking observer for algorithms that estimate position and speed of PM synchronous motor
- *Output File* - preview and generation of output header file that contains all required application and control constants
- *Cascade Structure* - sophisticated switch of cascade control structure enabling the selection of required control loop
  - scalar control - open loop voltage control
  - voltage FOC - dq voltages are input reference signals
  - current FOC - dq currents are input reference signals
  - speed FOC - required speed is an input reference signal
- *App control* - inner FreeMASTER control page for application graphical control

Motor Control Application Tuning Tool

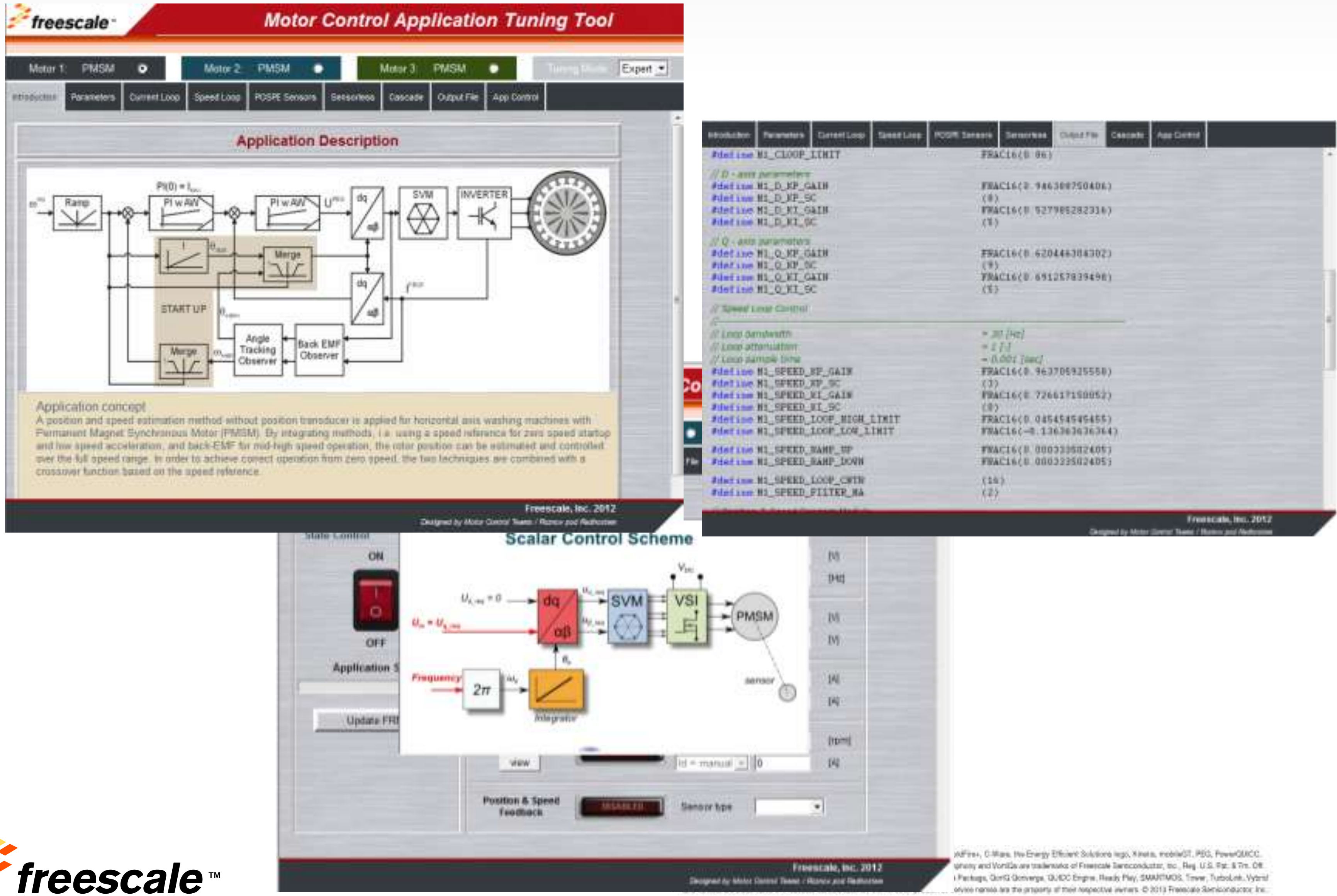


MCAT Tool – Features & Motor Types Supported

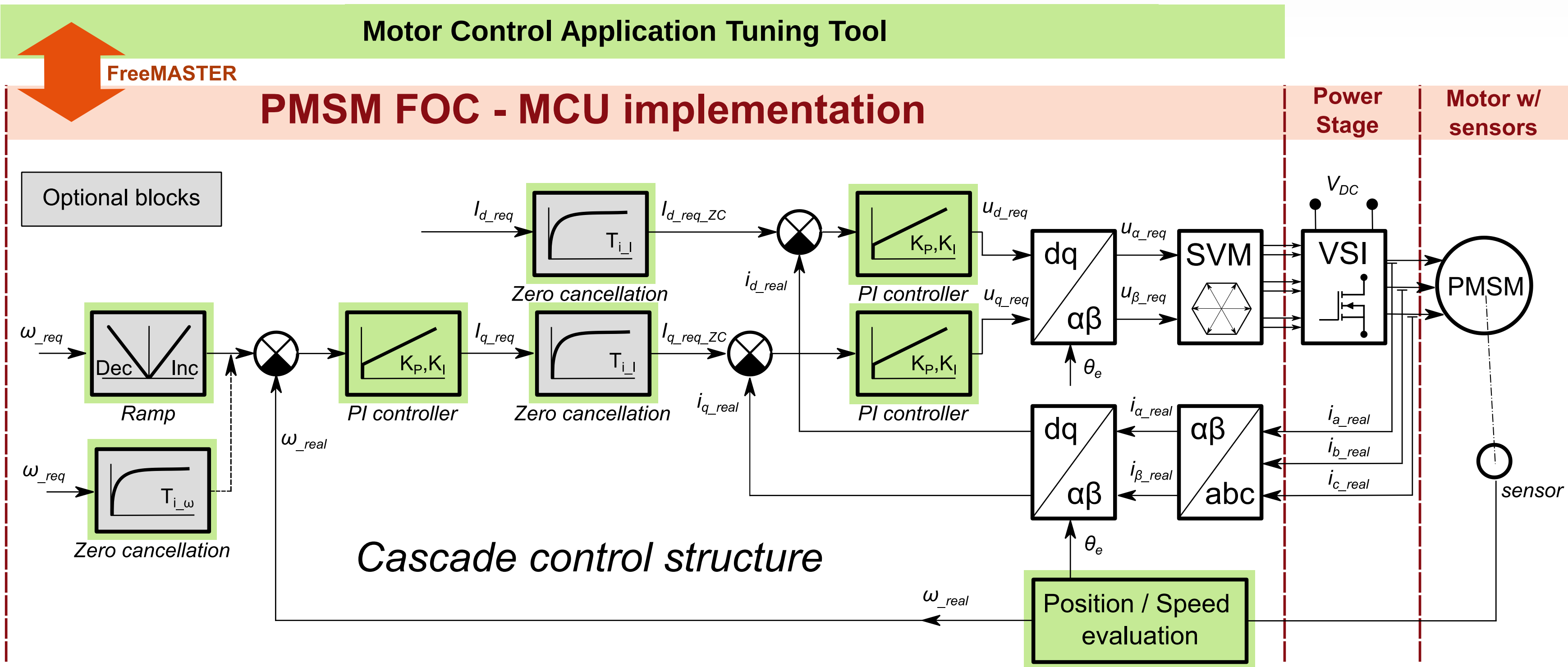


freescale™

MCAT Tool – Look and Feel

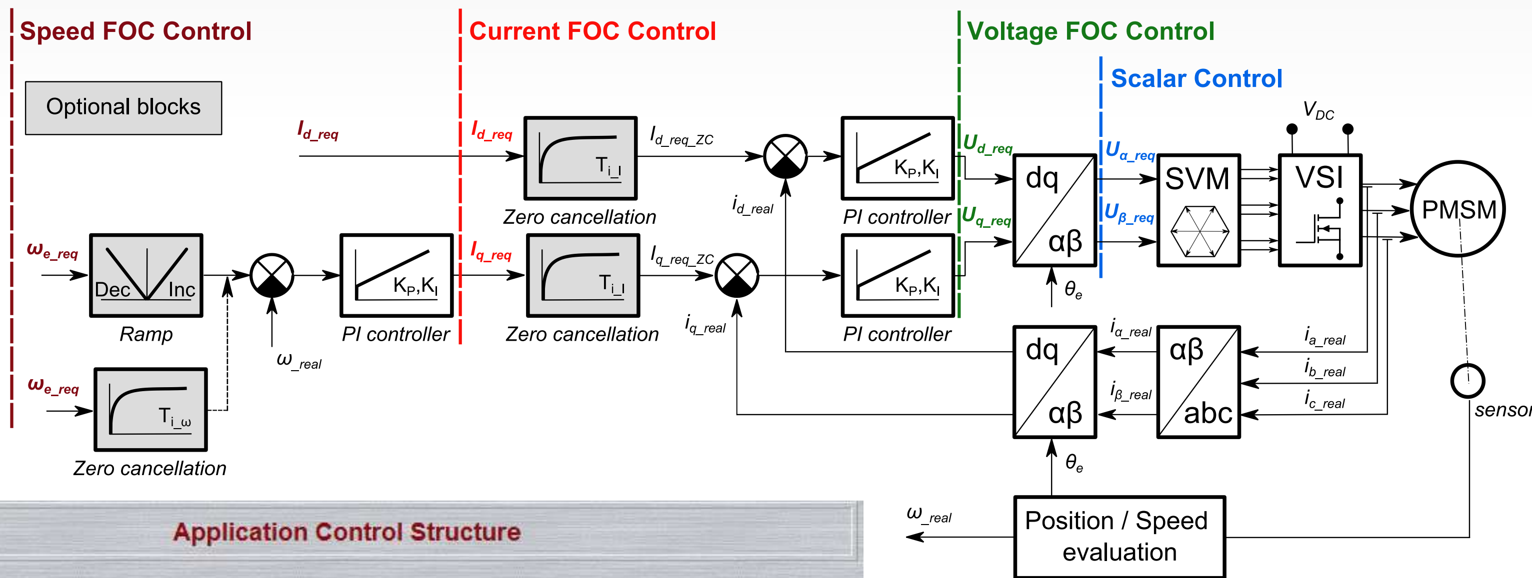


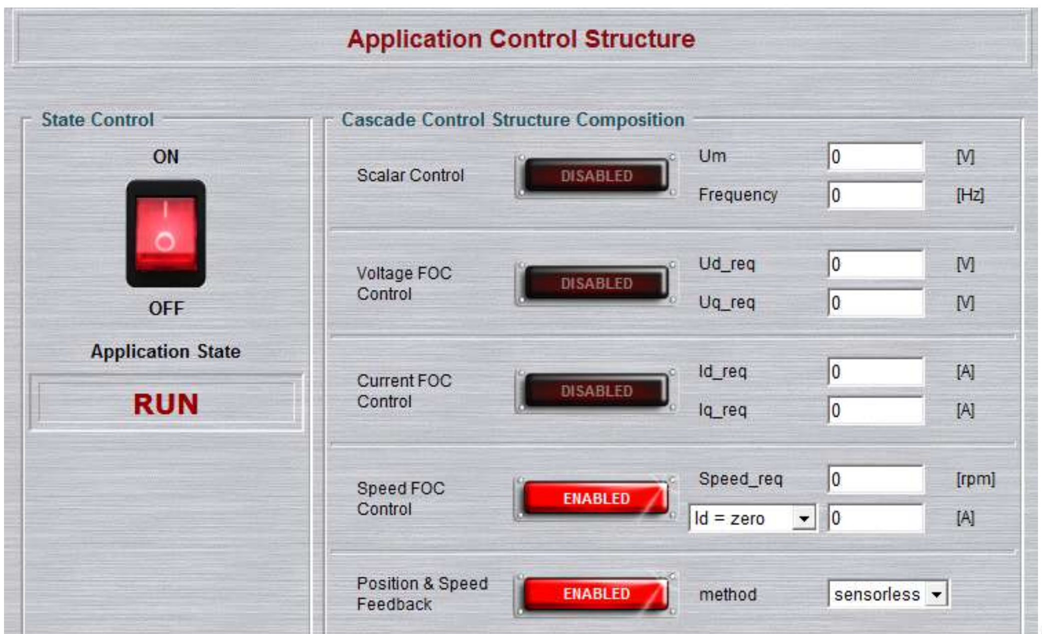
MCAT Tool Control Structure Parameters Tuning



The calculation of the PI controllers parameters is based on Pole-placement method, which is the one of the most popular technique in control theory. The Pole placement control method applied to closed-loop system leads to desired system behavior.

MCAT Tool Structure Selector





Open loop control
no need any current, position or speed feedback

Voltage control – position required
no need any current and speed feedback

Current control – current, position required
no need any speed feedback

Speed control - current, position and speed required

Hands on

Tuning of a motor control application

- We will use MCAT and FreeMASTER to tune an FOC sample code to run with a new motor.

Steps

- We will run the process step by step.
- Follow my lead.

Q&A

References

- Embedded Software and Motor Control Libraries
  - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FSLESL
- FreeMASTER
  - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FREEMASTER
- Tower low-voltage 3-phase motor control board
  - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=TWR-MC-LV3PH&tid=m32TWR
- MCAT
  - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MCATSW&fsrch=1&sr=1

NXP

freescale™