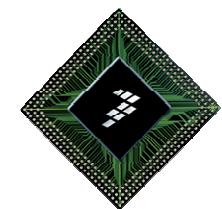


Power Architecture Training

Bolero (MPC5604B)



Smart Car Race - April, 2010

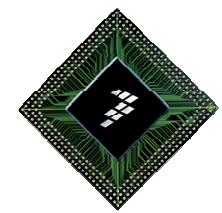
Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**TM
semiconductor



Designing with Freescale

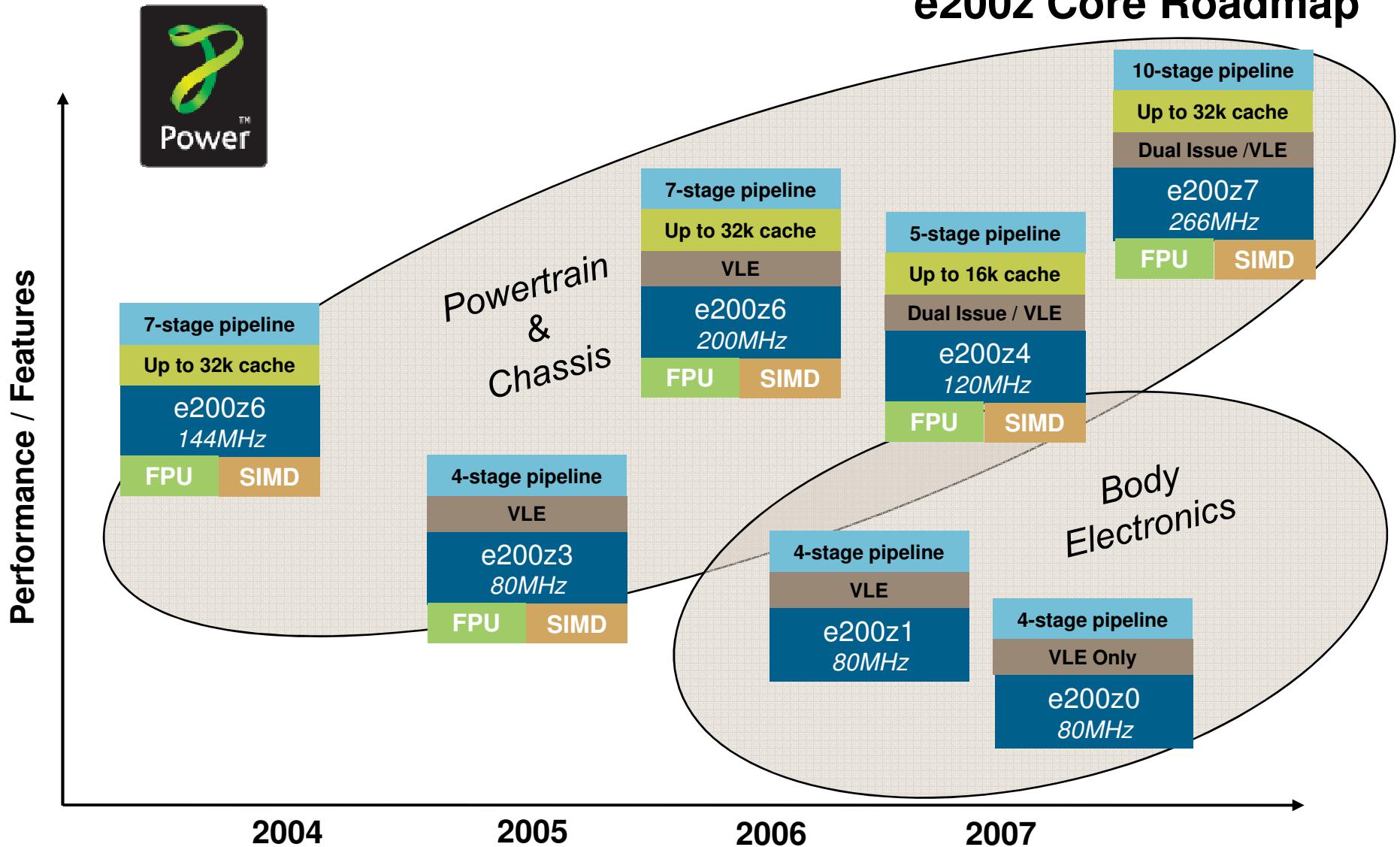
Power Architecture Overview



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**™
semiconductor

e200z Core Roadmap



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

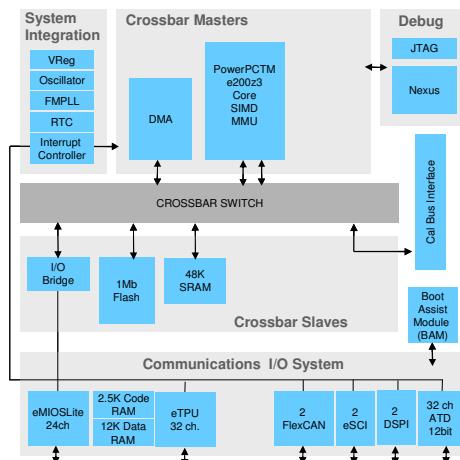
This document contains forward-looking statements based on current expectations, forecast and assumptions of Freescale that involves risk and uncertainties. Forward looking statements are subject to risk and uncertainties associated with Freescale business that could cause actual results to vary materially from those stated or implied by such forward-looking statements.



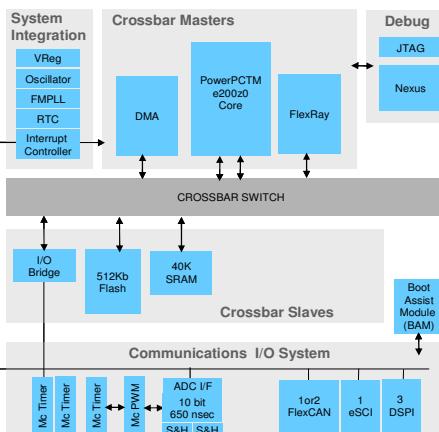


Cross Family Compatibility

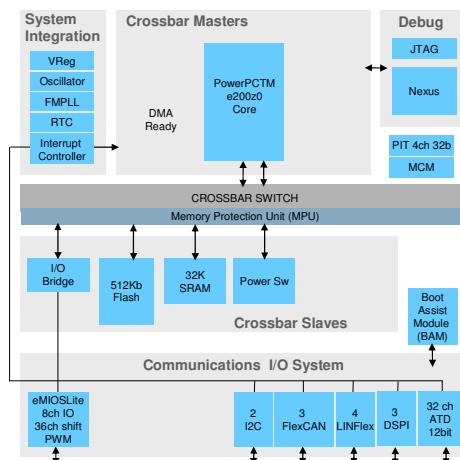
"Monaco"
(Powertrain)



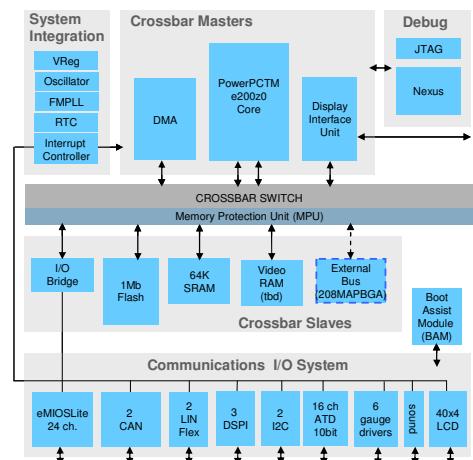
"Pictus"
(Steering/Airbag)



"Bolero"
(Body/Gateway)



"Spectrum"
(Inst Cluster)



32-bit standard architecture adopted across all product families

- Maximum IP reuse
- Optimized design and test flow
- Consolidated tool chain
- Strong Marketing message in compatibility

MPC5600 Family Part Numbers

Optional fields { 	M	— MC Qualification Status	Qual Status ↳ M: MC Status ↳ S: Auto Qualified ↳ P: PC Status
	P	— PowerPC Core	
	C	— Automotive Platform (C90)	
	56	— e200z0 Core version	
	0	Flash Size (Core dependent)	
	4	— Product	
	P	— Data Flash (Blank if none)	
	E	— FlexRay (Blank if none)	
	F	— 125C Temp Spec	
	M	— 144LQFP package	

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.



Bolero Family Line-up

Data Flash Code Flash	64Kb	64Kb	64Kb	64Kb	64Kb	64 Kb	Technology
4Mb				5647 Up to 256K	5647 Up to 256K	5647 Up to 256K	90nm 120 MHz 125oC T _a 200z4
3Mb				5646 Up to 256K	5646 Up to 256K	5646 Up to 256K	
2Mb				5645 Up to 256K	5645 Up to 256K	5645 Up to 256K	
1.5Mb			5607 Up to 96K	5607 Up to 96K			
1Mb			5606 Up to 80K	5606 Up to 80K			
768K		5605 Up to 64K	5605 Up to 64K	5605 Up to 64K			
512K		5604 Up to 48K	5604 Up to 48K				
384K		5603 Up to 40K	5603 Up to 28K				
256K		5602B/C Up to 32K	5602B/C Up to 24K				
256K	5602D Up to 20K	5602D Up to 20K					90nm 48 MHz 125oC T _a
128K	5601D Up to 16K	5601D Up to 16K					
Pin Out	64	100	144	176	208		QFP
						288 (TBD) (*1)*	BGA

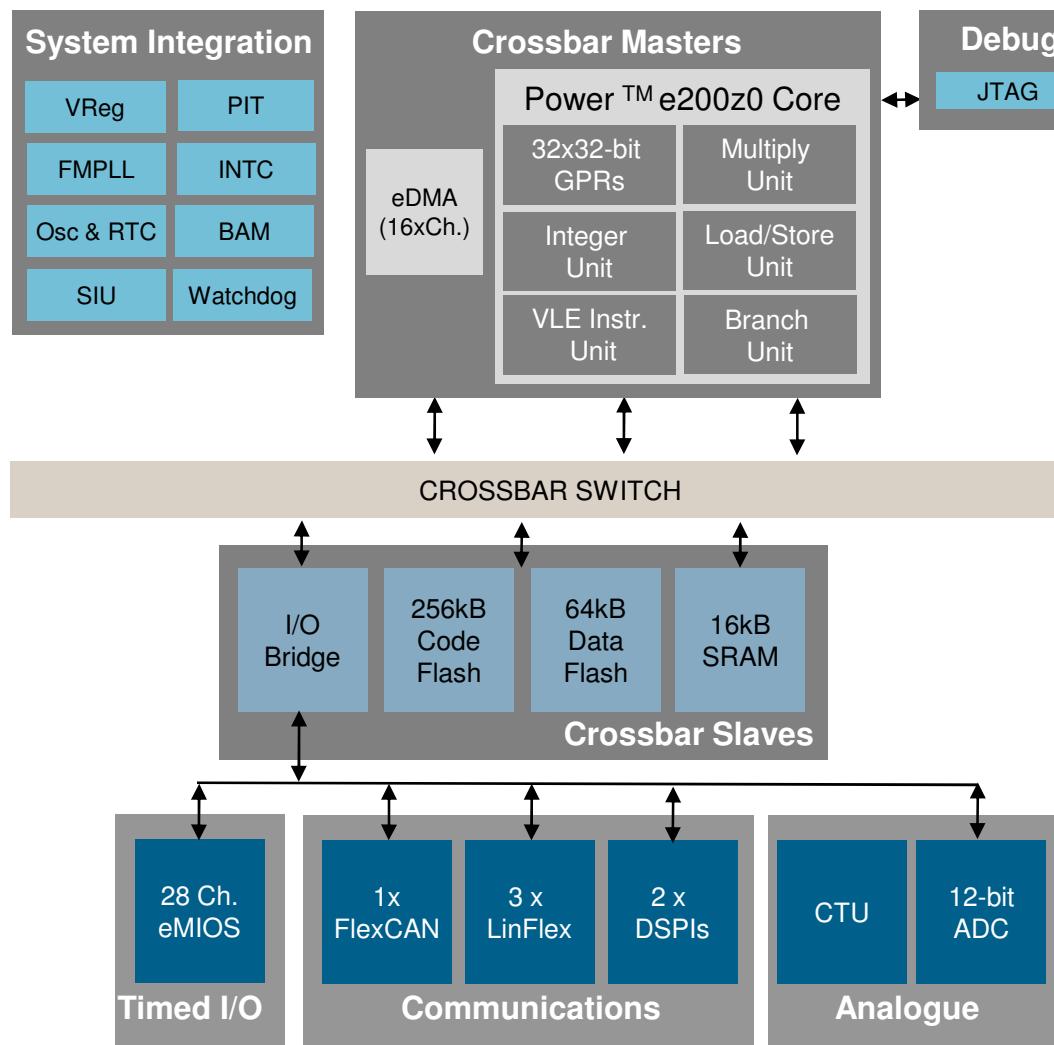
Device
Ram Size

Committed

Proposed

*1= package subject to confirmation.
All proposed parts features subject to change with out notice.

MPC5602D (Bolero 256K)



Core

- e200z0, Power Architecture with VLE, 48MHz, 4-stage pipelined RISC core.

Platform & System

- 256kB Code flash with prefetch, 64kB Data flash, 16kB system RAM all with ECC.
- Crossbar switch.
- Interrupt Controller.
- Direct Memory Access
 - Full 16 channel implementation.
- Nexus Class 1 level of debug, Class 2+ via emulation device

Timed I/O

- 28 channel eMIOS modules.
 - Supporting IOPWM, MC, IC & OC modes.

Communications

- 1 x FlexCAN Module
- 2 x SPI Interfaces.
- 3 x LinFLEX modules

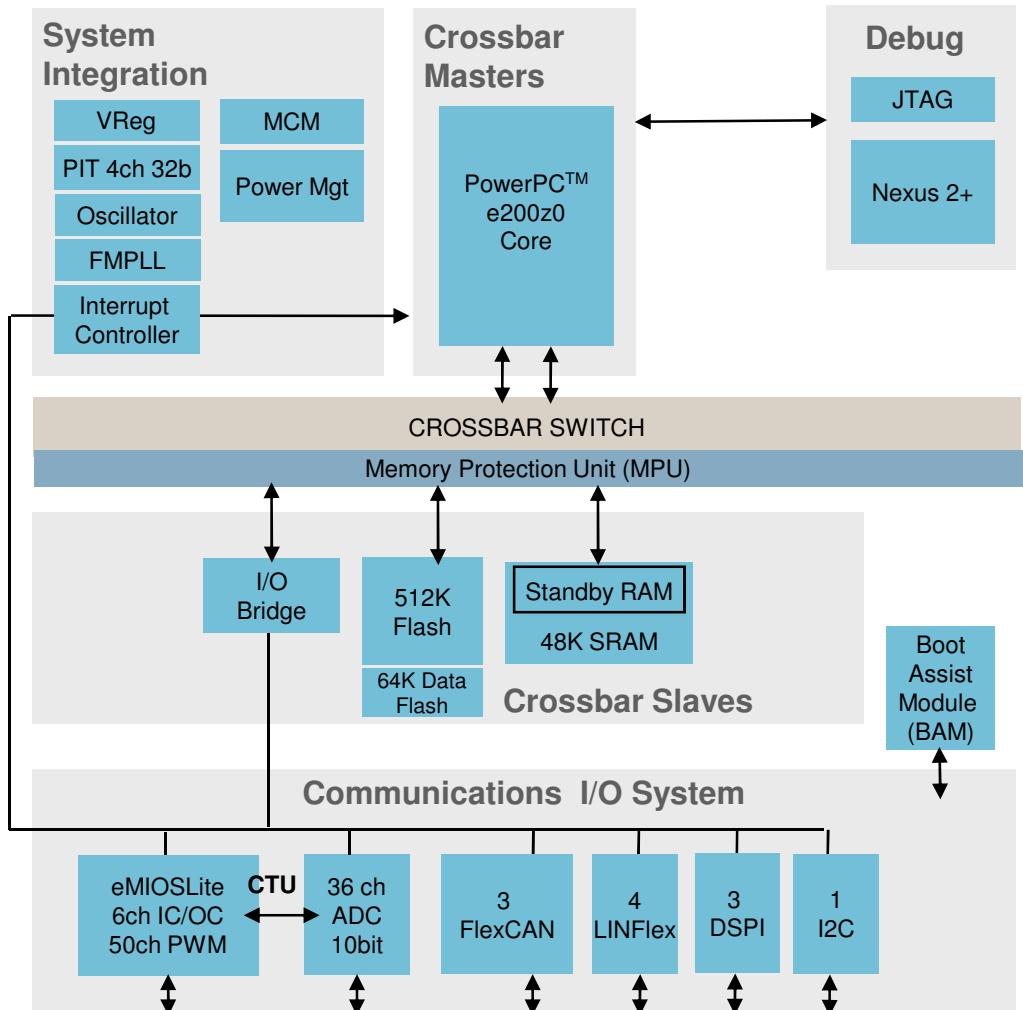
Analogue

- 32 x ADC channels
 - Supporting 12-bit accuracy.
- Cross triggering Unit
 - Direct eMIOS → ADC triggering.

Package

- 64 LQFP & 100LQFP
- Up to 79 GPIO

MPC5604B (Bolero 512K)



CORE

- PowerPC e200z0 core running 48-64MHz
- VLE ISA instruction set for superior code density
- Vectored interrupt controller
- Memory Protection Unit with 8 regions, 32byte granularity

MEMORY

- 512Kbyte embedded program Flash, 64KByte data flash
- 64Kbyte embedded data Flash (for EE Emulation)
- Up to 64MHz non-sequential access with 2WS
- ECC-enabled array with error detect/correct
- 48Kbyte SRAM (single cycle access, ECC-enabled)

COMMUNICATIONS

- 3x enhanced FlexCAN
 - 64 Message Buffers each, full CAN 2.0 spec
- 4x LINFlex
- 3x DSPI, 8-16 bits wide & chip selects
- 1x I²C

ANALOG

- 5V ADC 10-bit resolution

TIMED I/O

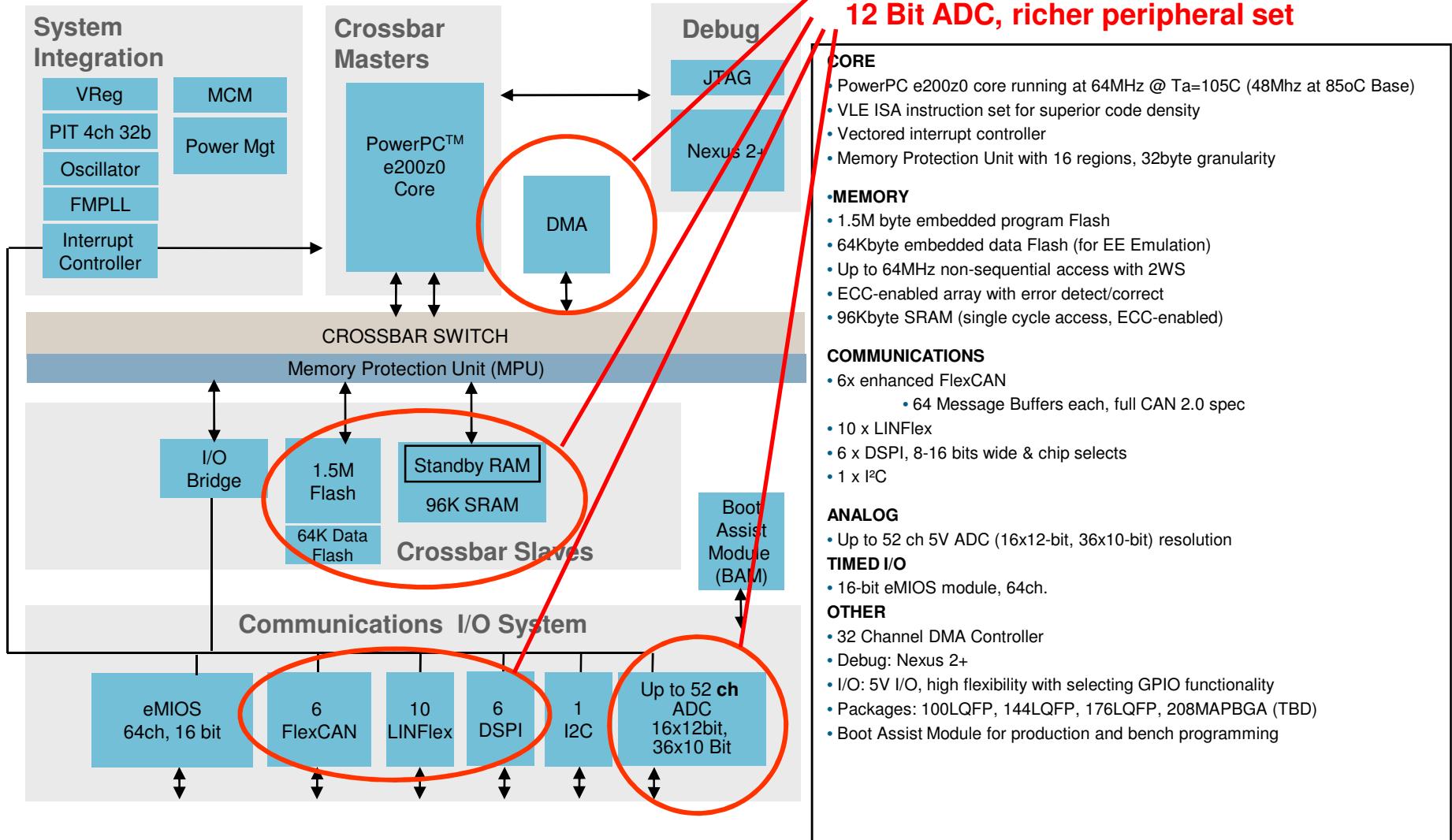
- 16-bit eMIOS module

OTHER

- CTU (Cross Triggering Unit) to sync ADC with PWM Channels
- Debug: Nexus 2+
- I/O: 5V I/O, high flexibility with selecting GPIO functionality
- Packages: 100LQFP, 144LQFP, 208MAPBGA (Development only)
- Boot Assist Module for production and bench programming

MPC5607 (Bolero 1.5M)

additional DMA, bigger memories, additional 12 Bit ADC, richer peripheral set



Bolero 100LQFP Compatibility

	MPC5602D* (256k)	MPC5604B (512k)	MPC5607B (1.5M)
Si Technology	90nm	90nm	90nm
CPU	e200z0	e200z0	e200z0
fbus	48MHz	64MHz	64MHz
ADC0 (10Bit)	no	yes	yes
ADC1 (12Bit)	yes	no	yes
DMA	yes	no	yes
Other Peripherals	same	same	same
100LQFP	yes	yes	yes
Pinout	a) 4 extra ADC channels (PA[3,7,10,11])	a) no ADC functionality on PA[3,7,10,11]	a) no ADC functionality on PA[3,7,10,11]
	b) 2 extra GPIO, or 2 ADC reference voltage pins *	b) 2 extra GPIO PD[11], PB[12] (Pins 59 & 60)	b) 2 extra ADC Supply Pins (Pins 59 & 60)
Tools	same	same	same

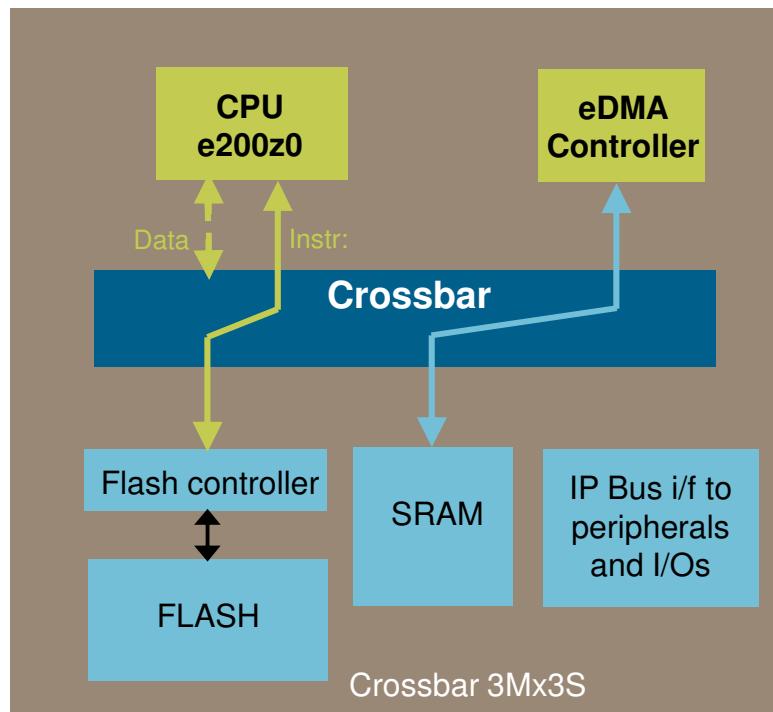
* Current plan of record is to add 2 GPIO's, but final decision depends on 12-bit ADC characterization results

Crossbar Switch Architecture MPC5605/6/7B

- ▶ The Crossbar increases overall system performance by providing up to **two simultaneous connections** between
 - 3 master ports
- ▶ e200z0 uses 2 master ports
 - 3 slave ports
- ▶ 32 bit address and 32 bit data on Bolero
- ▶ The eDMA offloads the processor tasks for data movement.
- ▶ Timers can initiate complex DMA sequences (channel link) to have automatic peripheral management without CPU load

Example Access:

- z0 instruction fetch from flash
- eDMA SRAM access

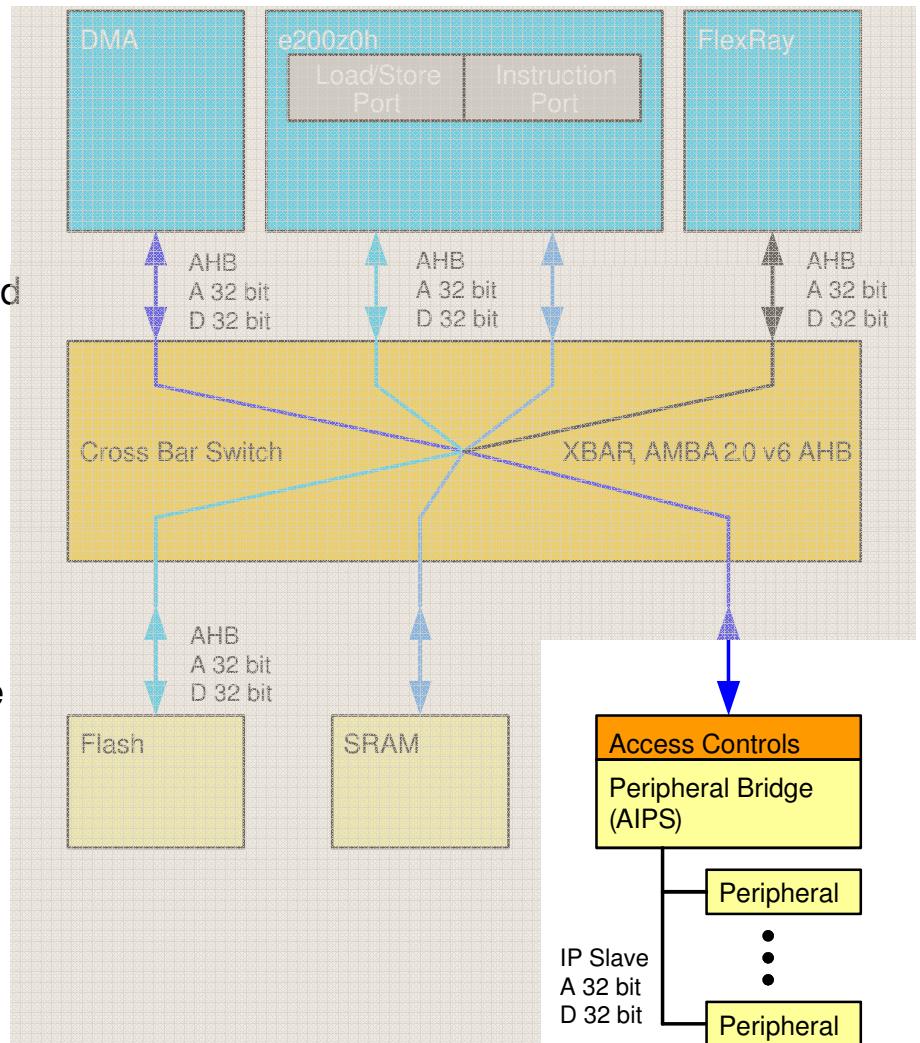


Platform overview - Busses

Peripheral bridge details

AHB 2.v6 to IP Skyblue interface (AIPS)

- ▶ AIPS supports the IPS slave bus (sky-blue) signals (FSL standard peripheral bus)
- ▶ Act as interface between the system bus and lower bandwidth peripherals
- ▶ Aligned 32-bit word accesses, half word accesses, and byte accesses are supported for the peripherals
- ▶ Data accesses that cross a 32-bit boundary are not supported
- ▶ Indicating to slave peripherals if an access is in supervisor or user mode.
- ▶ Peripheral read transactions require a minimum of 2 clocks (Two-clock read accesses are possible when the requested access size is 32-bits or smaller)
- ▶ Peripheral write transactions require a minimum of 3 clocks (Three clock write accesses are possible when the requested access size is 32-bits or smaller)

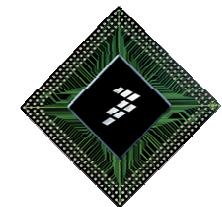




Designing with Freescale

Clocks – Clock Generation Module

- Overview of Platform Clocks
- FIRC (16 MHz), FXOSC (4 – 16 MHz), SIRC (128 KHz), SXOSC (32 KHz)
- FMPLL
- CMU



MPC560xB CGM Clock sources:

► Main Clock Sources

- 4-16 MHz External Crystal/Oscillator -> FXOSC*
- 16 MHz Internal RC Oscillator -> FIRC*
 - Default system clock after Reset
 - Trimmable

► Low Power Clock Sources

- 32 KHz External Crystal/Oscillator -> SXOSC* (not available on MPC5601/2D)
 - Low power oscillator
 - Dedicated for RTC/API
- 128 KHz Internal RC Oscillator -> SIRC*
 - Dedicated for RTC/API and watchdog
 - Trimmable

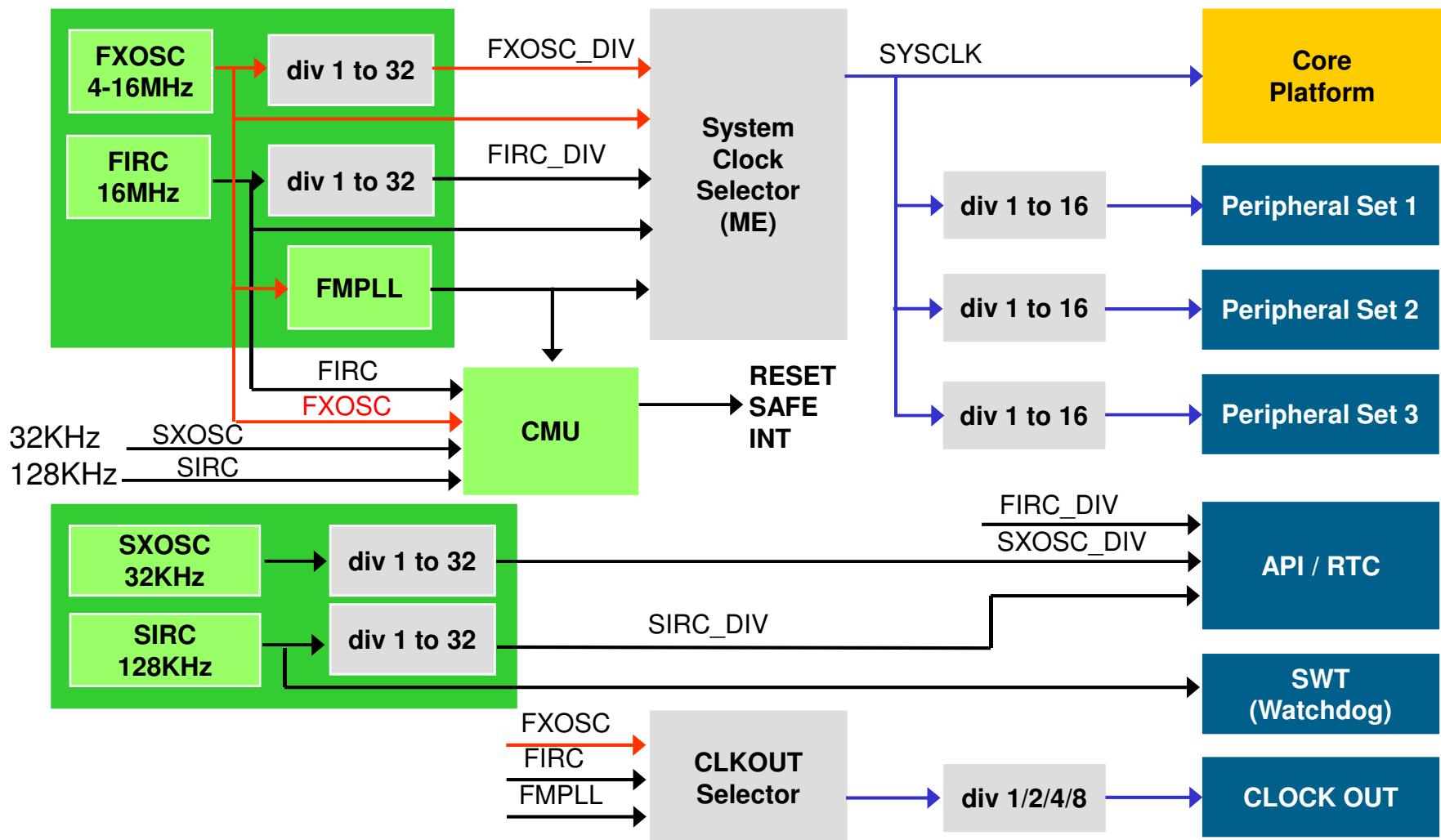
* Naming convention changed in MPC5604B RM Rev.2

MPC560xB Clock structure

- Frequency modulated PLL to reduce EMC
- Clock monitor unit for
 - **PLL clock monitoring** : detect if PLL leaves an upper or lower frequency boundary
 - **Crystal clock monitoring** : monitors the external crystal oscillator clock which must be greater than the internal RC clock divided by a division factor given by RCDIV[1:0] of CMU_CSR register.
 - **Frequency meter** : measure the frequency of a RCOSC versus a known reference clock XOSC

Platform overview – Clocks

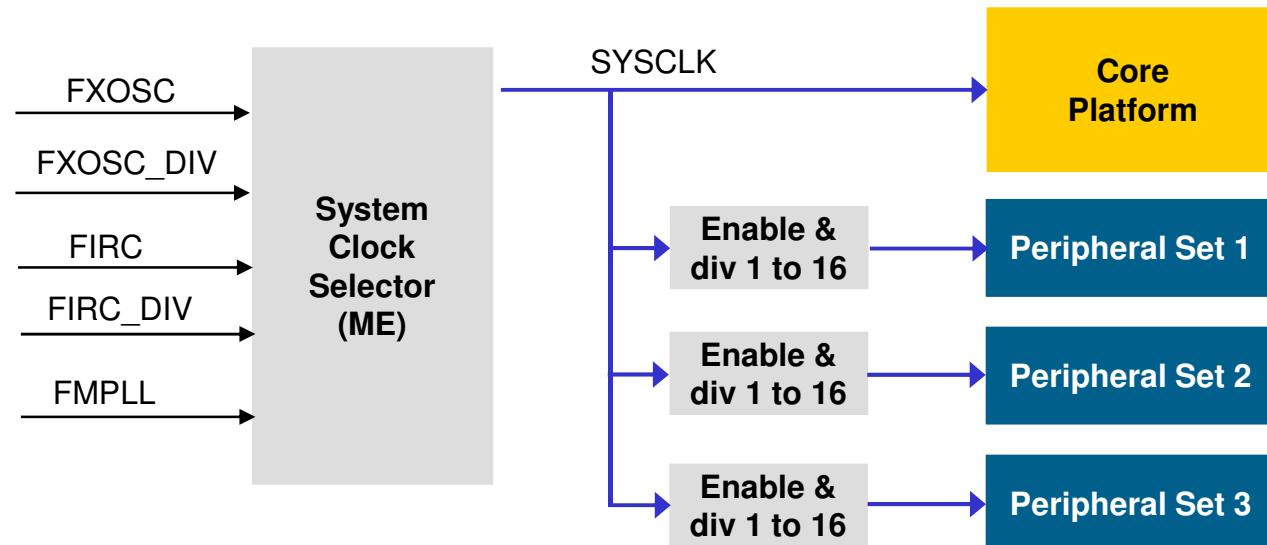
MPC560xB CGM Clocking Structure



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

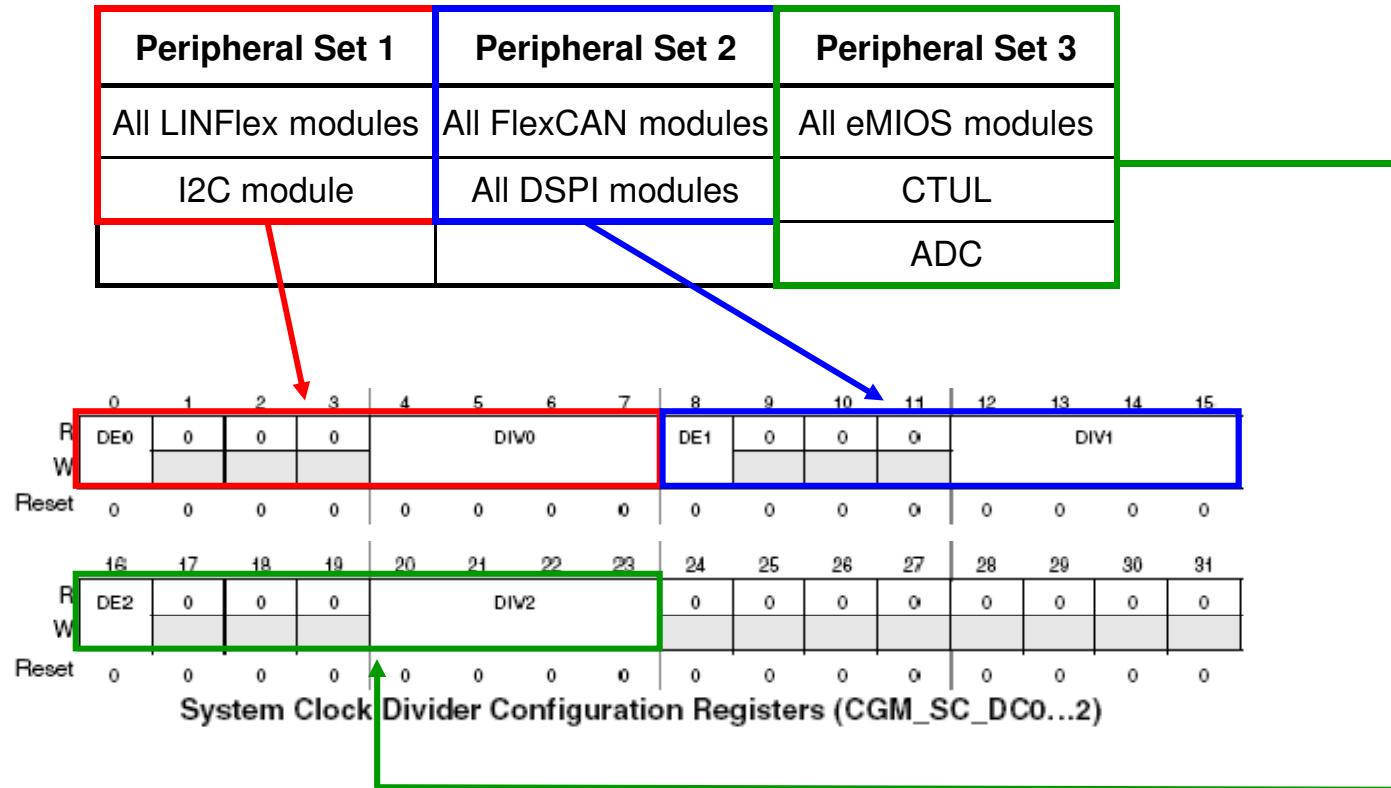
Platform overview – Clocks MPC560xB CGM System Clock

- ▶ Provides the clock (divided or not) to the Core/Peripherals
- ▶ Selected by ME_XXX_MC register



Platform overview – Clocks

MPC560xB CGM System Clock Divider Configuration Register



DEx: Peripheral Set x Divider Enable (Default value 1 = ON)

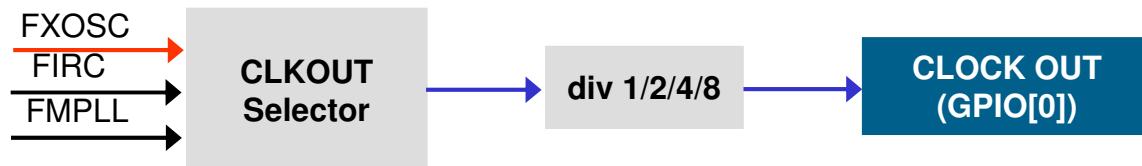
DIVx: Peripheral Set x Divider x Division Value (1..15)

Platform overview – Clocks

MPC560xB CGM Output Clock Description

- Clock output on the GPIO[0] (PA[0]) pin

Watch out: max pad slew rate!

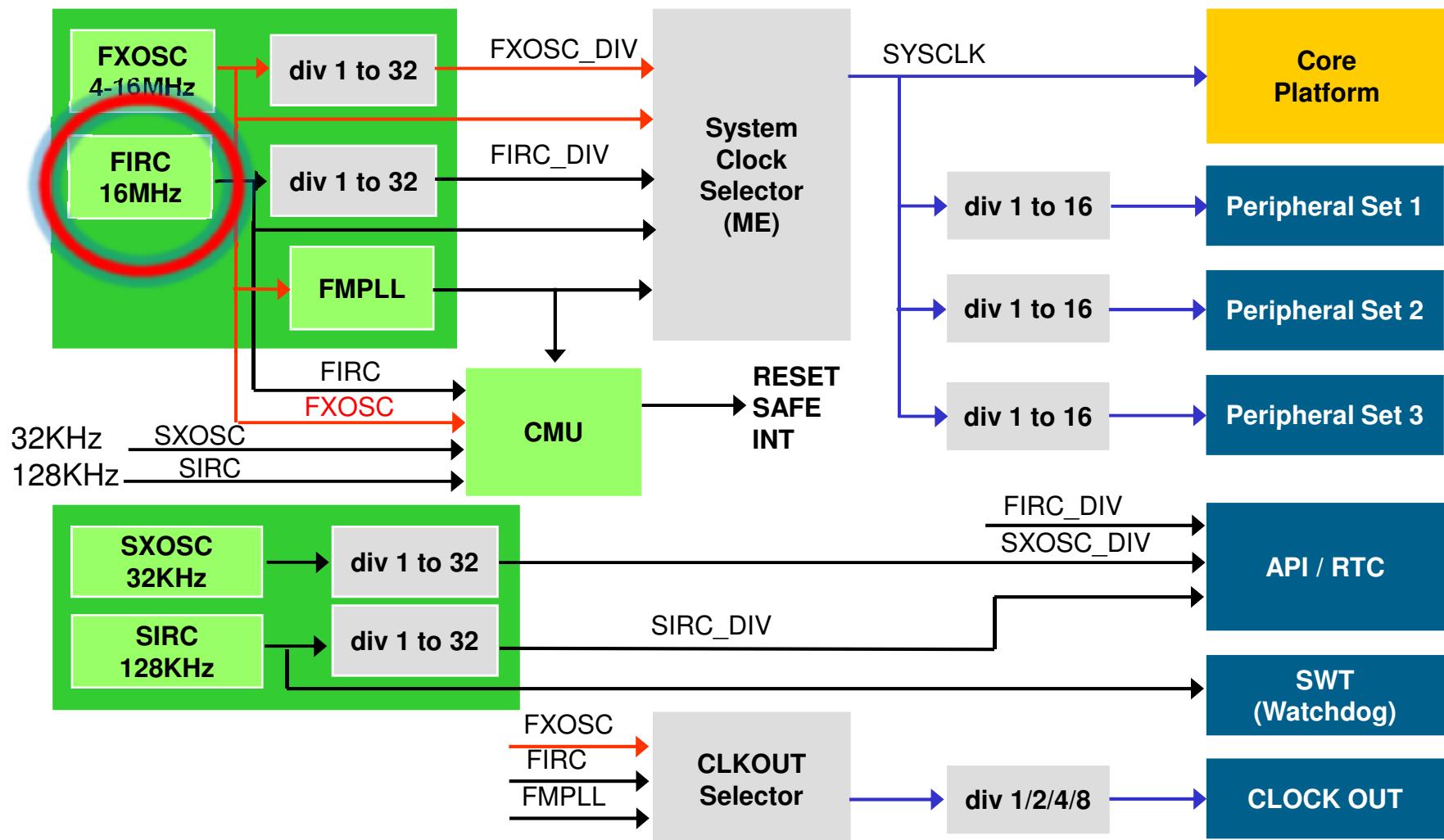


0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15	
R	0	0	SELDIV				SELCTL			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
W								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
W																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Output Clock Division Select Register (CGM_OCDS_SC)

SELDIV:	division by 1, 2, 4, 8
SELCTL:	clock source (XOSC, FIRC, PLL) selection

FIRC 16 MHz



CGM FIRC 16MHz

Digital Interface – Control Register 1/2

- ▶ Default clock after reset
- ▶ Division factors ranging from 1, 2, 3....32
- ▶ Trimming (+/- 1%)

CGM IRC 16MHz

Digital Interface – Control Register 2/2

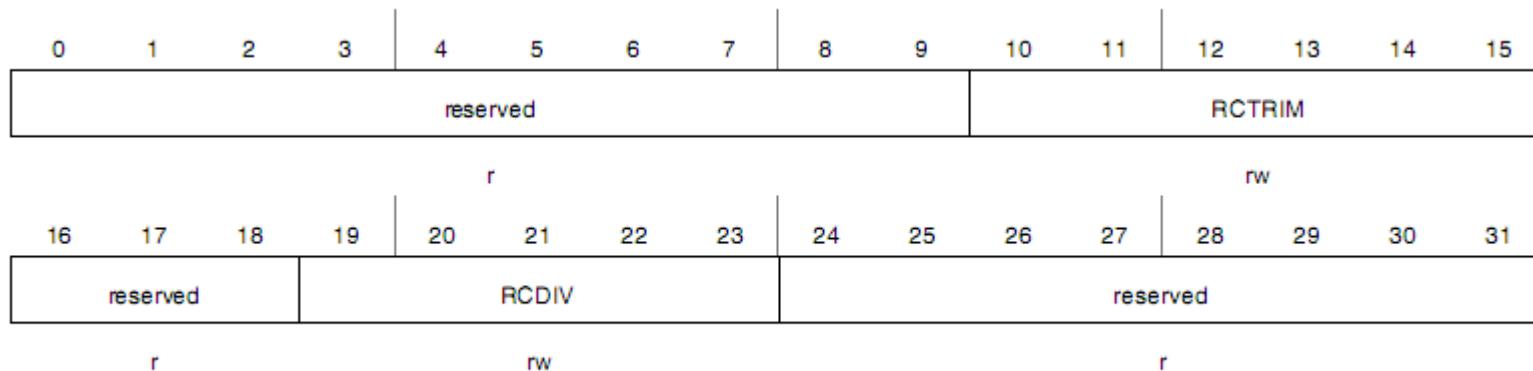
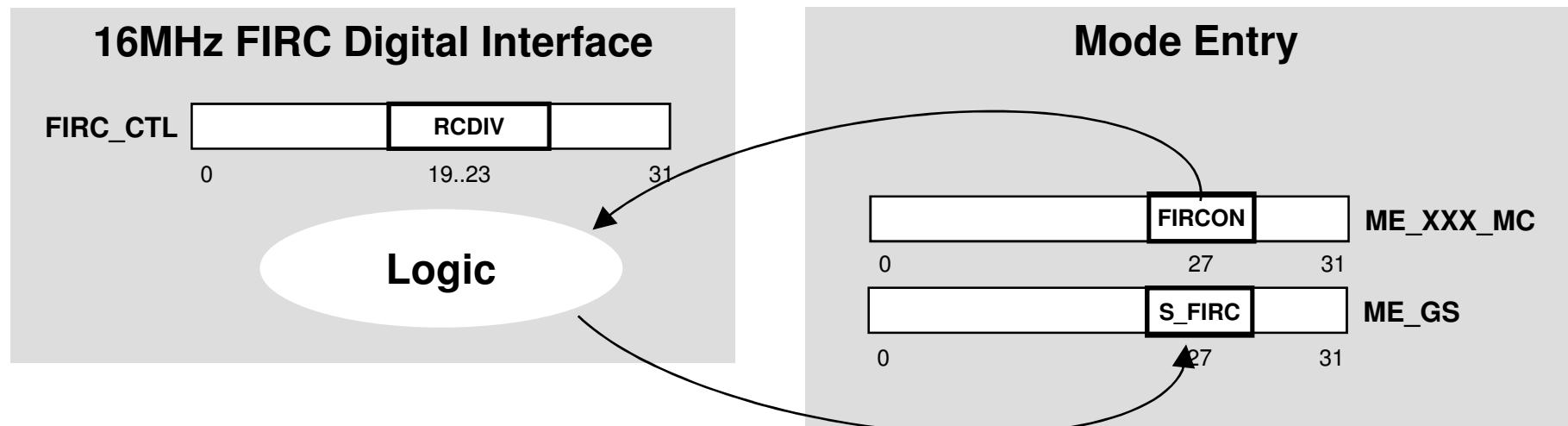


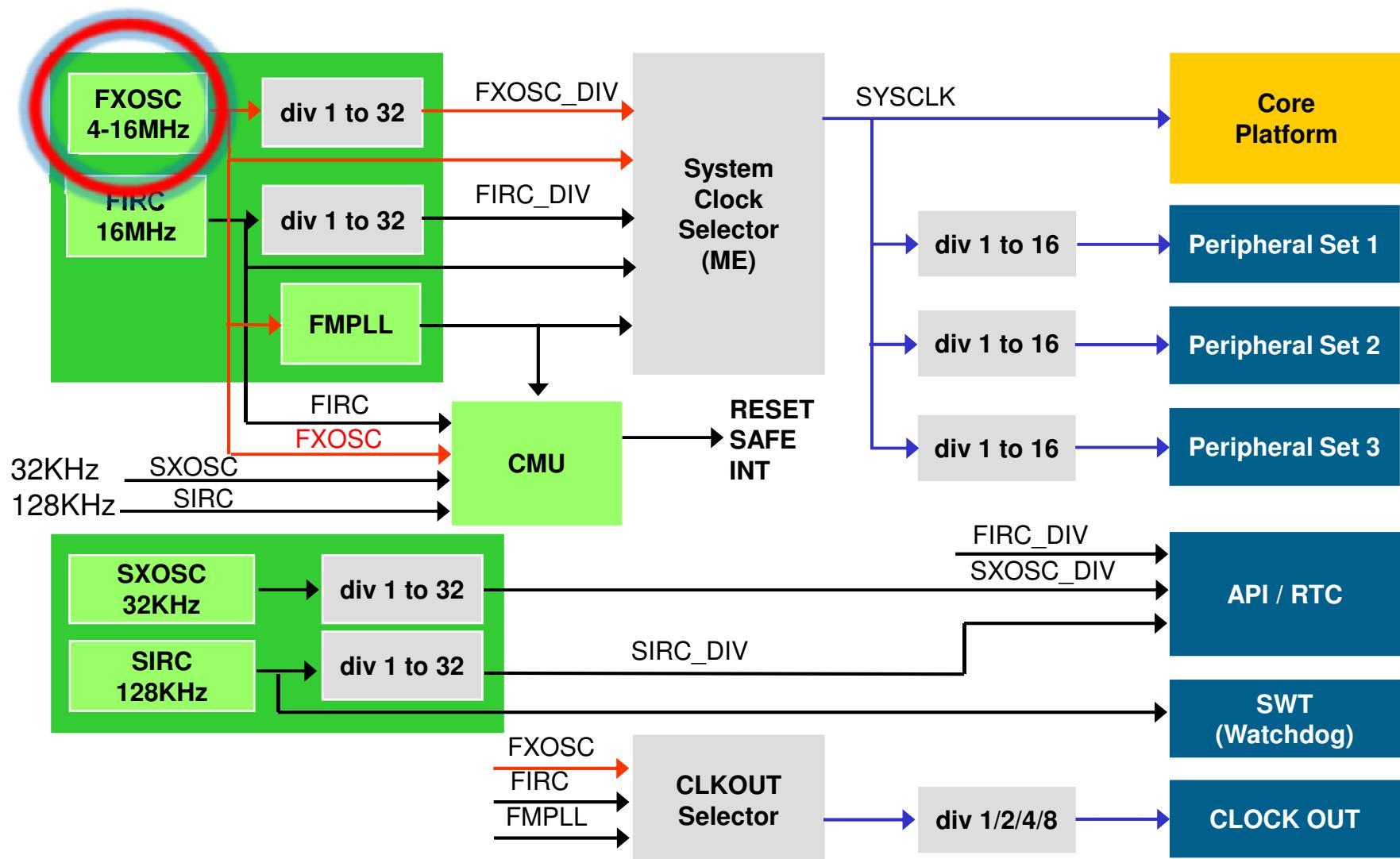
Table 3-16. RC Oscillator Control Register (RC_CTL)

Bit	Name	Description
10-15	RCTRIM	RC trimming bits
19-23	RCDIV	RC division factor

- ▶ Switching on/off controlled by Mode Entry
- ▶ Divider configured in the FIRC digital Interface



FXOSC



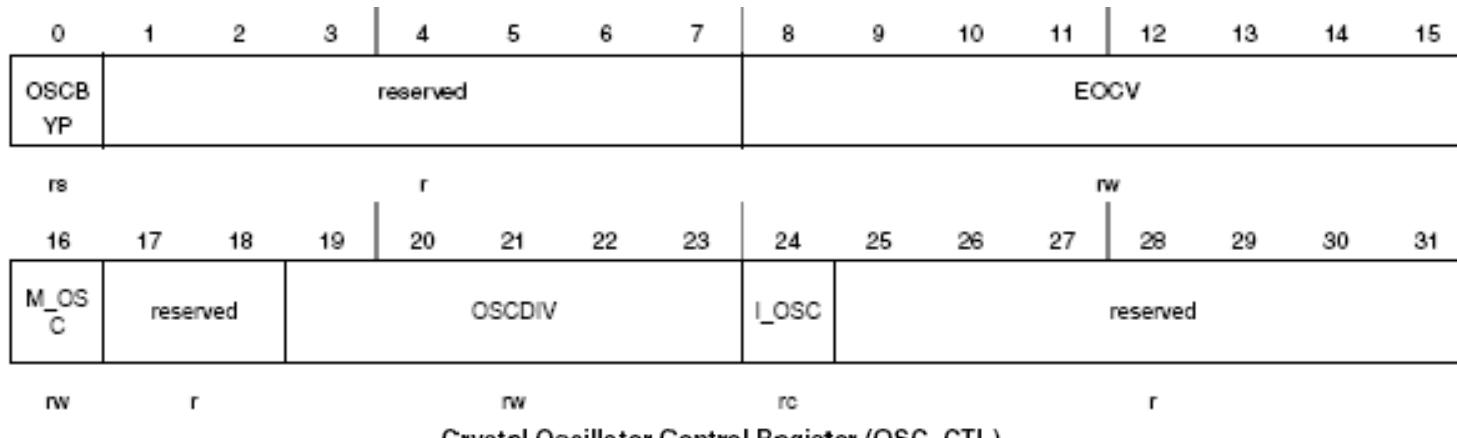
CGM FXOSC 4-16MHz

Digital Interface – Control Register

- ▶ Division factors ranging from 1, 2, 3....32
- ▶ Bypass mode to enable external clock source
- ▶ Clock ready interrupt flag
 - Configurable start-up time ($\text{EOCV}[7:0] * 512 = 0 .. 255*512$)

CGM FXOSC 4-16MHz

Digital Interface – Control Register

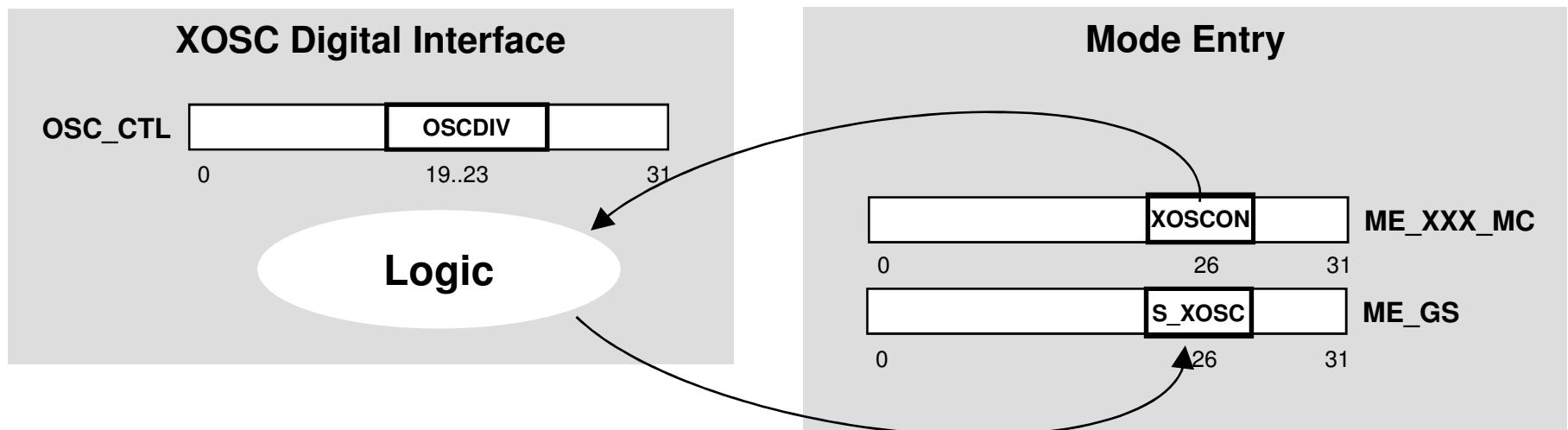


Bit	Name	Description
0	OSCBYP	Oscillator bypass
8-15	EOCV	End of count value
16	M_OSC	Oscillator available interrupt mask
19-23	OSCDIV	Crystal oscillator division factor
24	I_OSC	Oscillator available interrupt flag

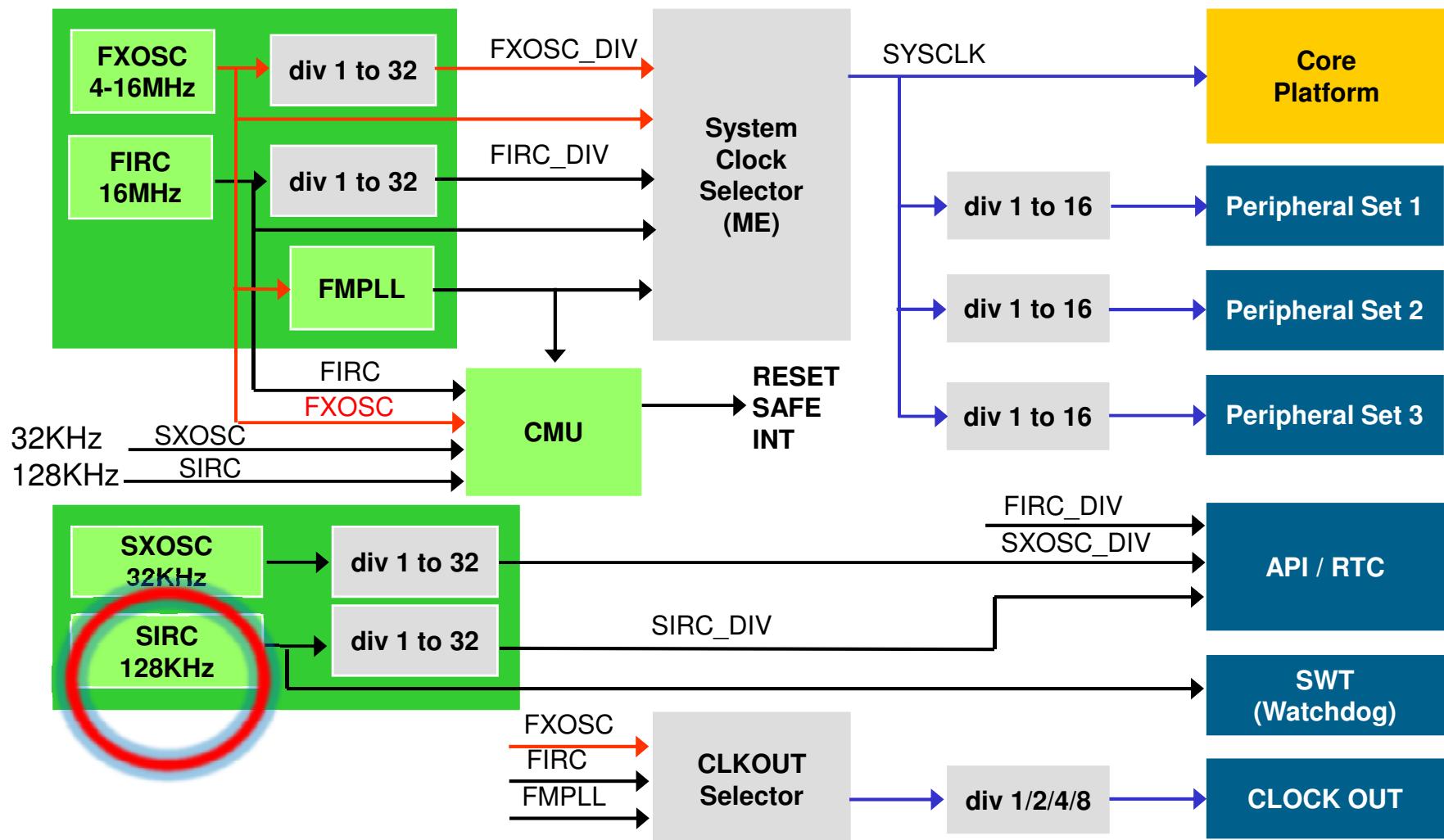
CGM FXOSC 4-16MHz

Application Control

- ▶ FXOSC Switching on/off controlled by Mode Entry
- ▶ FXOSC dividers configured in the FXOSC Digital Interface



SIRC 128 KHz



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

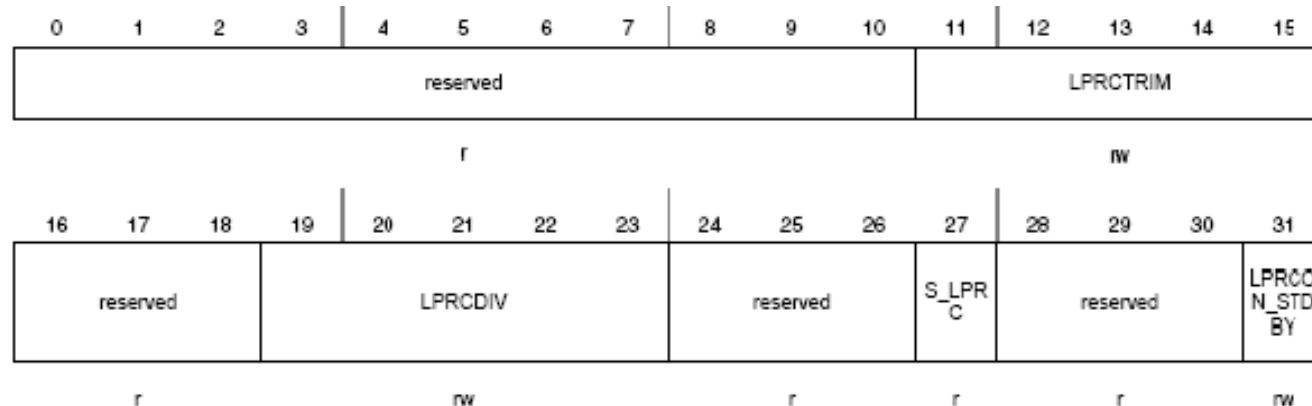
CGM SIRC 128KHz

Digital Interface – Control Register 1/2

- ▶ Control SIRC state (ON/OFF) in STANDBY Mode. In all other modes e.g. RUN, HALT, the SIRC is always ON
- ▶ Division factors ranging from 1, 2, 3....32
- ▶ Trimming (+/- 2%)

CGM SIRC 128KHz

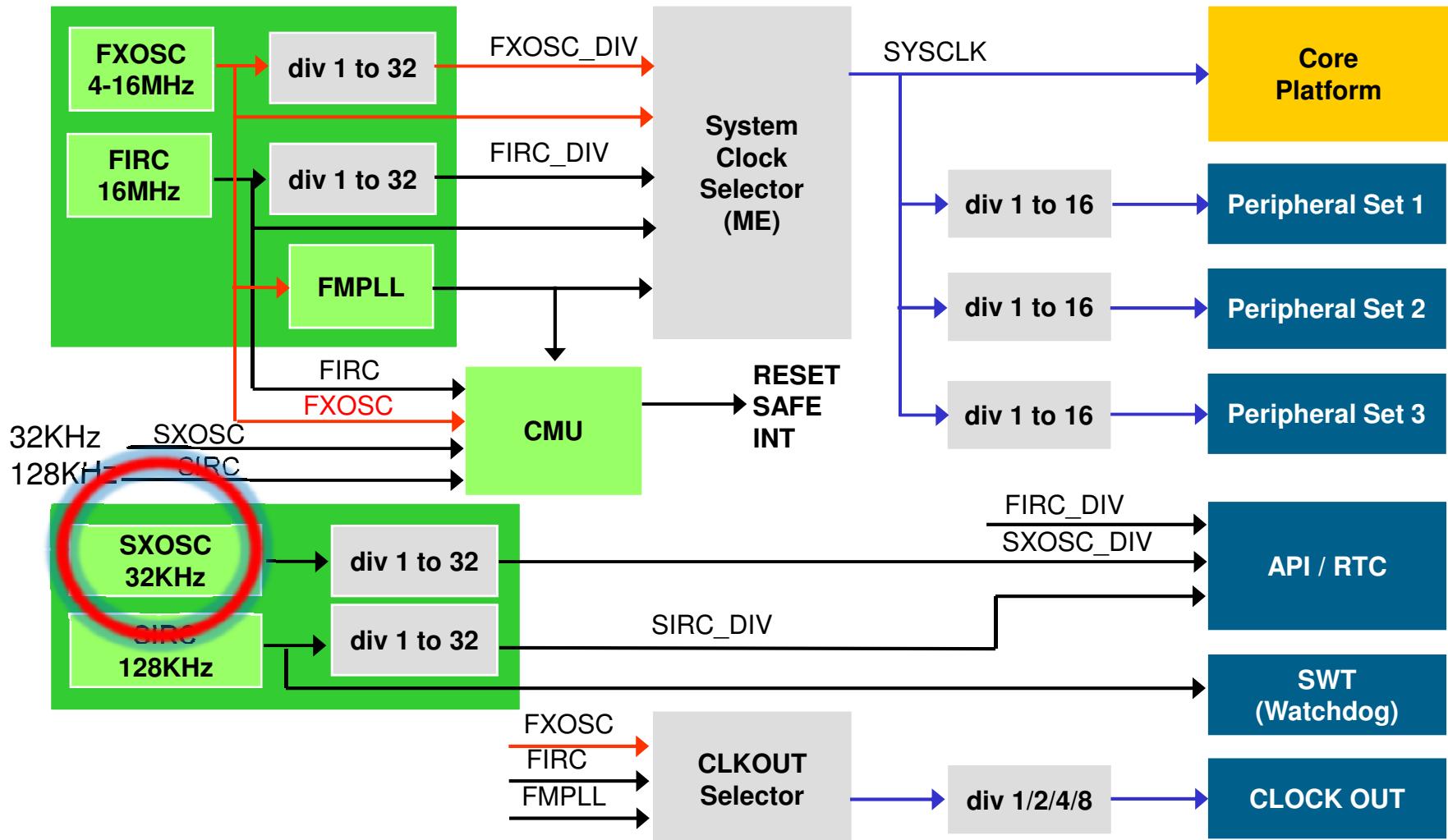
Digital Interface – Control Register 2/2



Low Power RC Control Register (LPRC_CTL)

Bit	Name	Description
11-15	LPRCTrim	Trimming bits
19-23	LPRCDIV	Division factor
27	S_LPRC	SIRC Clock Status
31	LPRCON_STDBY	Control in STANDBY mode

SXOSC 32 KHz



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

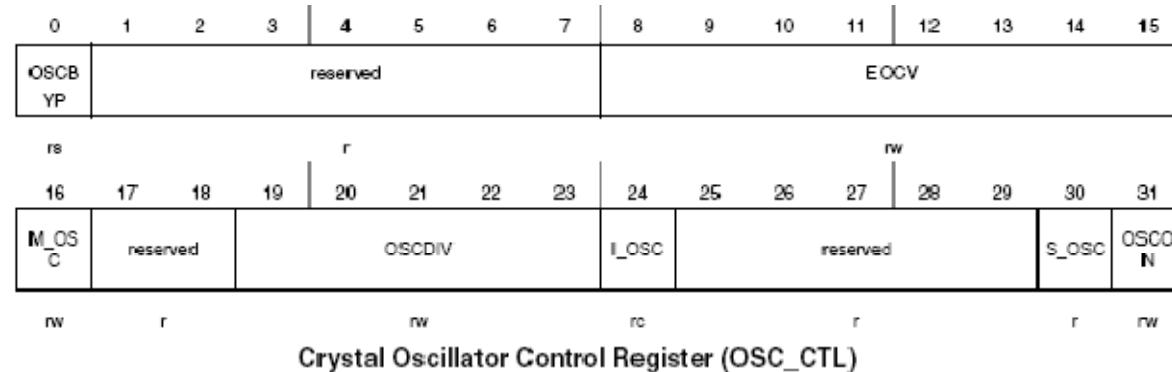
CGM SXOSC 32kHz

Digital Interface – Control Register 1/2

- ▶ SXOSC 32KHz power down control and status
- ▶ Division factors ranging from 1, 2, 3...32
- ▶ Bypass mode
- ▶ Clock ready interrupt flag
- ▶ Configurable start-up time ($\text{EOCV}[7:0] * 512 = 0 .. 255*512$)
- ▶ Not available on MPC5601/2D

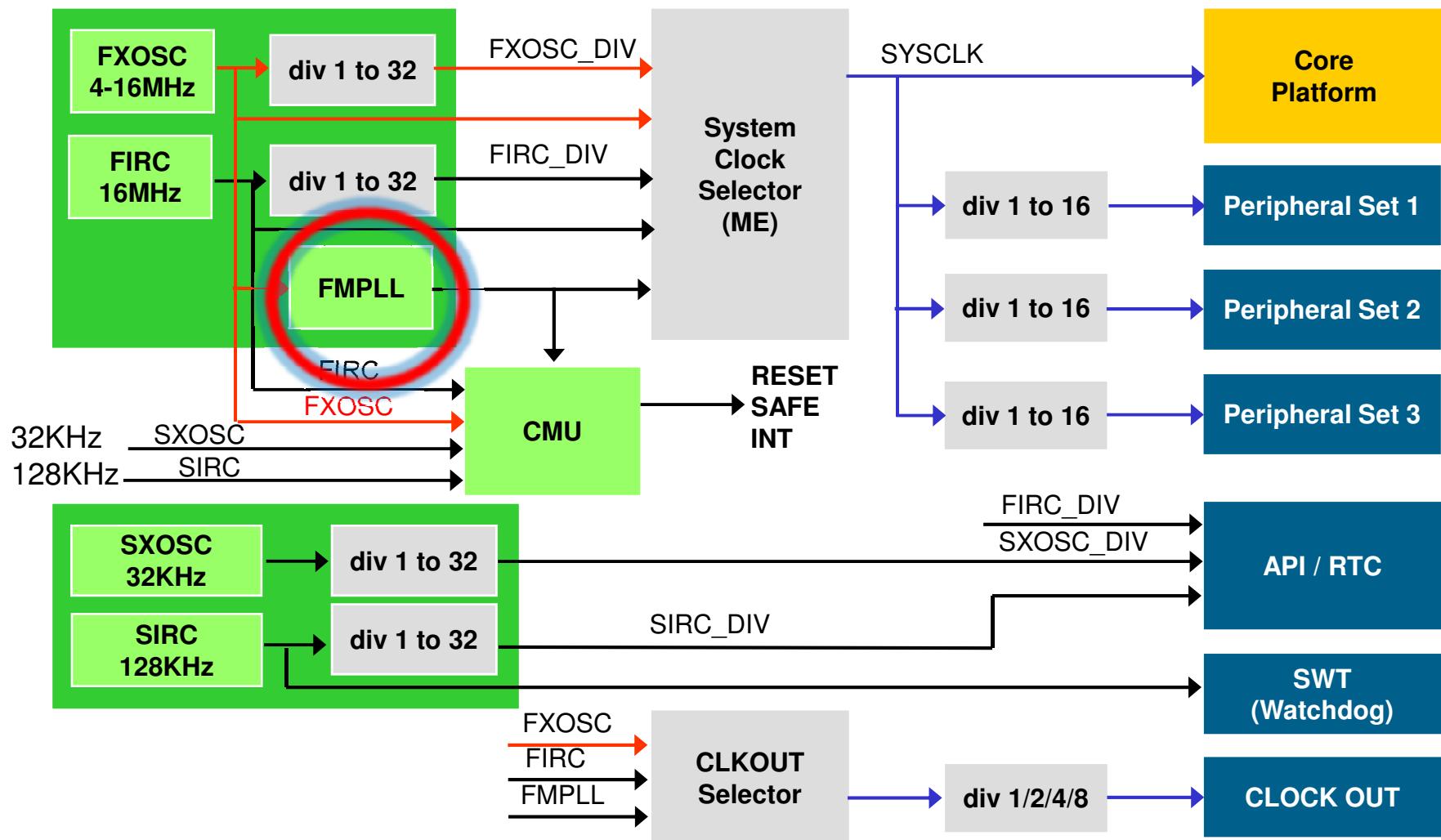
CGM SXOSC 32kHz

Digital Interface – Control Register 2/2



Bit	Bit name	Bit description
0	OSCBYP	Oscillator bypass
8-15	EOCV	End of count value
16	M_OSC	Oscillator available interrupt mask
19-23	OSCDIV	Crystal oscillator division factor
24	I_OSC	Oscillator available interrupt flag
30	S_OSC	Oscillator status
31	OSCON	Oscillator power down control

FMPLL

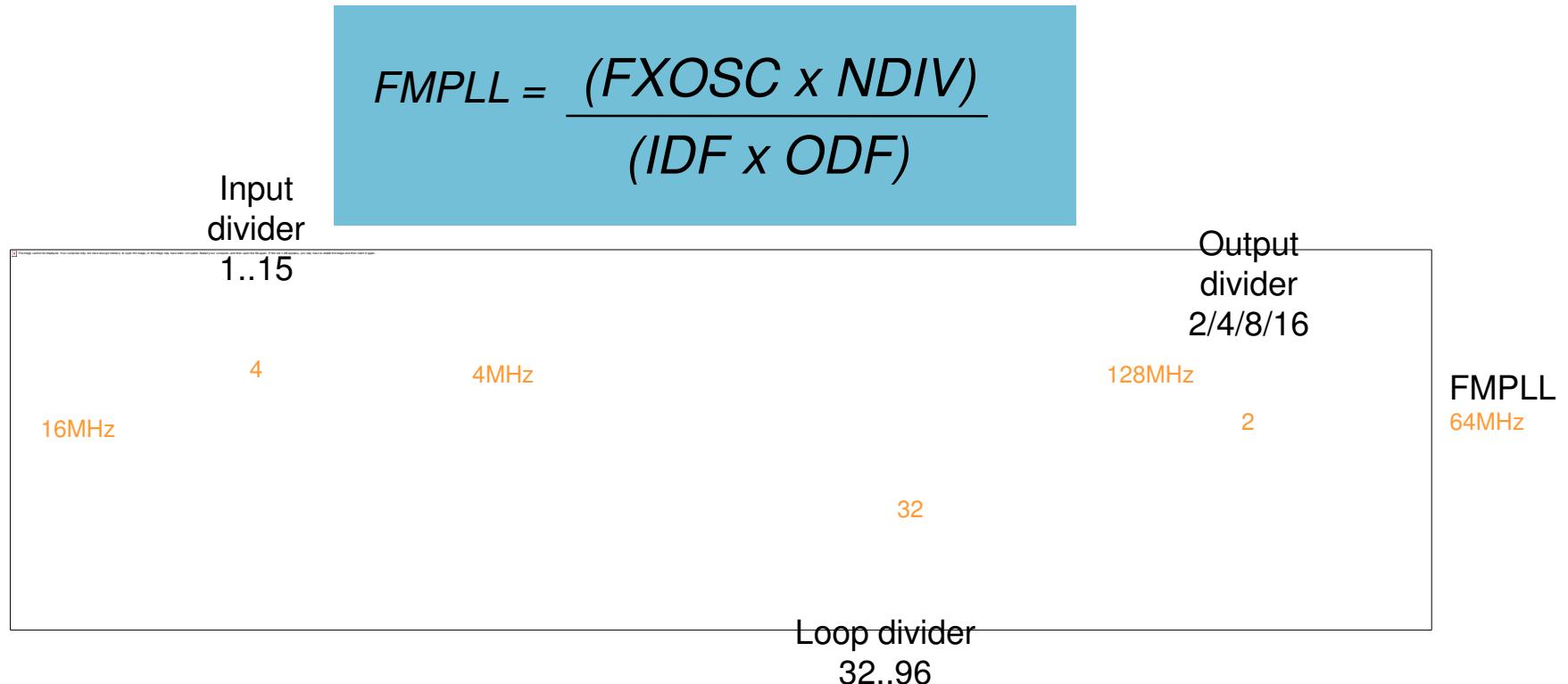


Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

CGM Frequency Modulated PLL (FMPLL)

- ▶ The purpose of the FMPLL is to generate a 64 MHz max (120MHz target for Bolero 2M/4M) system clock from the FXOSC.
- ▶ The FMPLL operating modes:
 - Power down
 - Normal
 - Normal with frequency modulation
 - Progressive clock switching
 - 1:1
- ▶ These modes are controlled by two registers:
 - Control Register (CR)
 - Modulation Register (MR)

- The PLL output clock frequency derives from the relation:

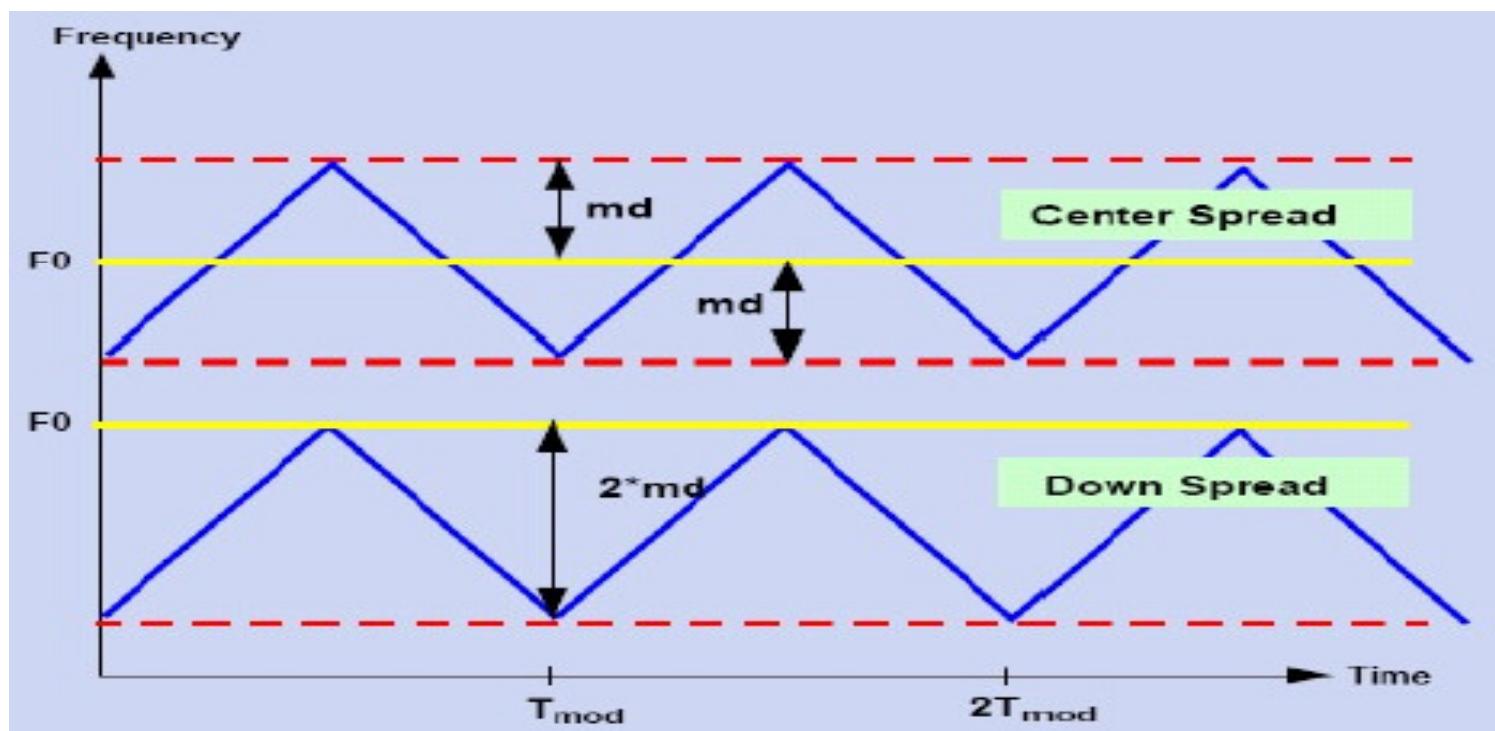


Example: FXOSC = 16MHz
 FMPLL = 64MHz → NDIV = 4 * IDF * ODF → NDIV = 32 , IDF = 4 , ODF = 2

CGM FMPLL

Frequency modulation

- Frequency modulated FMPLL
 - Modulation enabled/disabled through software
 - Triangle wave modulation
- Programmable modulation depth
 - $\pm 0.25\%$ to $\pm 4\%$ deviation from center spread frequency
 - -0.5% to -8% deviation from down spread frequency
 - Programmable modulation frequency dependent on reference frequency



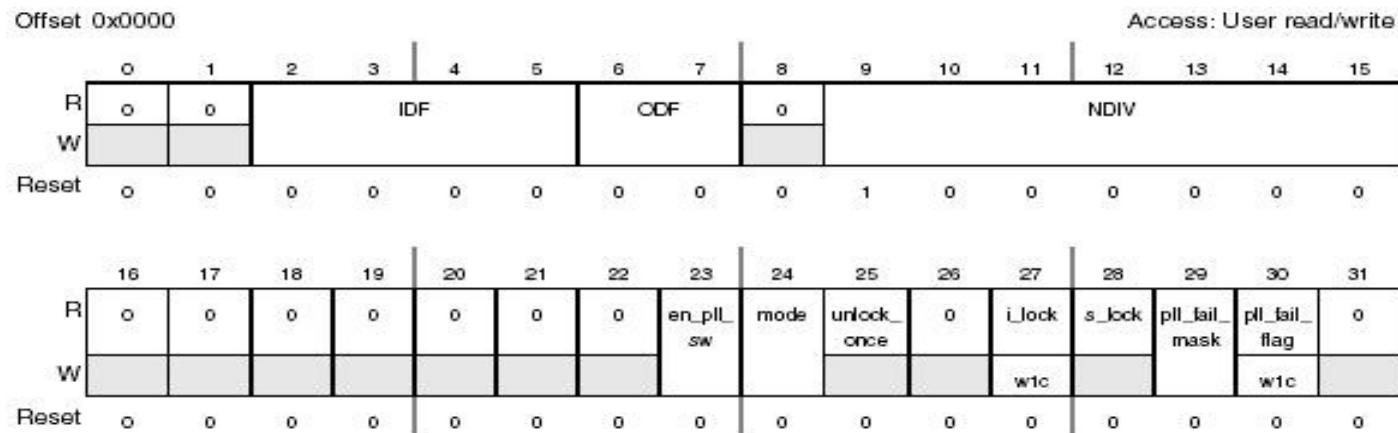
- ▶ FMPLL is enabled at Mode Entry ME_xxx_MC(PLLON)
- ▶ **Progressive clock switching** allows to switch FXOSC input clock to PLL output clock stepping through different division factors: This means that the current consumption gradually increases and so the voltage regulator has a better response.
 - This mode is enabled by setting bit PLL_CR(en_pll_sw), prior to enabling the PLL by setting the bit ME_xxx_MC(PLLON)
 - Divide FMPLL output by **8->4->2->1 for 8 clock cycles** each time

- ▶ The 1:1 mode is selected bit by asserting the CR(mode) bit.
- ▶ Then the input clock is divided by two and switched directly to the output clock.
- ▶

$$F_{FMPLL} = F_{FXOSC} / 2$$

CGM FMPLL

Control Register (CR) 1/2



Bit	Bit name	Bit description
2-5	IDF	The value of this field sets the PLL Input Divider
6-7	ODF	The value of this field sets the PLL Output Divider
9-15	NDIV	The value of this field sets the PLL Loop Divider

CGM FMPLL

Control Register (CR) 2/2

Bit	Bit name	Bit description
23	en_pll_sw	This bit is used to enable progressive clock switching.
24	mode	This bit is used to activate the 1:1 Mode.
25	unlock_once	This bit is a sticky indication of PLL loss of lock condition. Unlock_once is set when the PLL loses lock. Whenever the PLL reacquires lock, unlock_once remains set. Only por_rst_b can clear this bit.
27	i_lock	This bit is set by hardware whenever there is a lock/unlock event. It is cleared by software, writing 1.
28	s_lock	This bit is an indication of whether the PLL has acquired lock.
29	pll_fail_mask	This bit is used to mask the pll_fail output.
30	pll_fail_flag	This bit is asynchronously set by hardware whenever a loss of lock event occurs while PLL is switched on. It is cleared by software, writing 1.

► Steps to enable Frequency Modulation (FM):

1. Configure FM characteristics : MOD_PERIOD, INC_STEP.
2. Enable FM: Set FMPLL0_MR[FM_EN] = 1.
FM modulated mode can be enabled only when PLL is in lock state!

► MOD_PERIOD (Tmod) = round[Fref / 4xFmod]

- Fref: the feedback clock frequency = Fvco
- Fmod: modulation frequency

► INC_STEP = round{ [(2¹⁵ - 1) x NDIV x md] / [500 x MODPERIOD] }

- md: represents the peak modulation depth in percentage.
- Centre spread: pk-pk = +/- md
- Down spread: pk-pk = -2*md

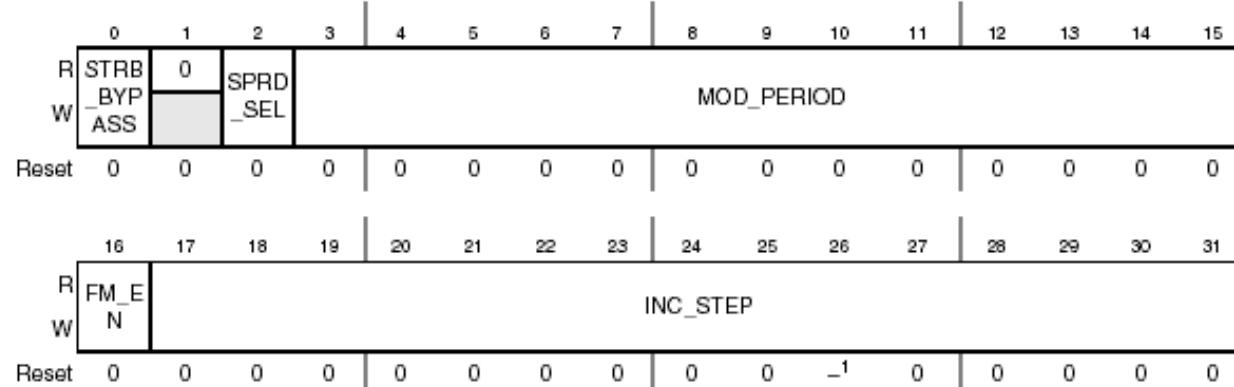
- ▶ Modulation parameters are latched in the PLL module in two different ways
 - **MR(STRB_BYPASS) =1** : parameters have to be changed only when PLL is in power down mode.
 - **MR(STRB_BYPASS) =0** : parameters are latched when modulation is enabled AND PLL is locked.
- ▶ The modulation depth in %
 - $md = [500 \times INCSTEP \times MODPERIOD] / [(2^{15} - 1) \times NDIV]$
- ▶ User must ensure that the product of INCSTEP and MODPERIOD is less than $2^{15} - 1$

CGM FMPLL

Modulation Register (MR)

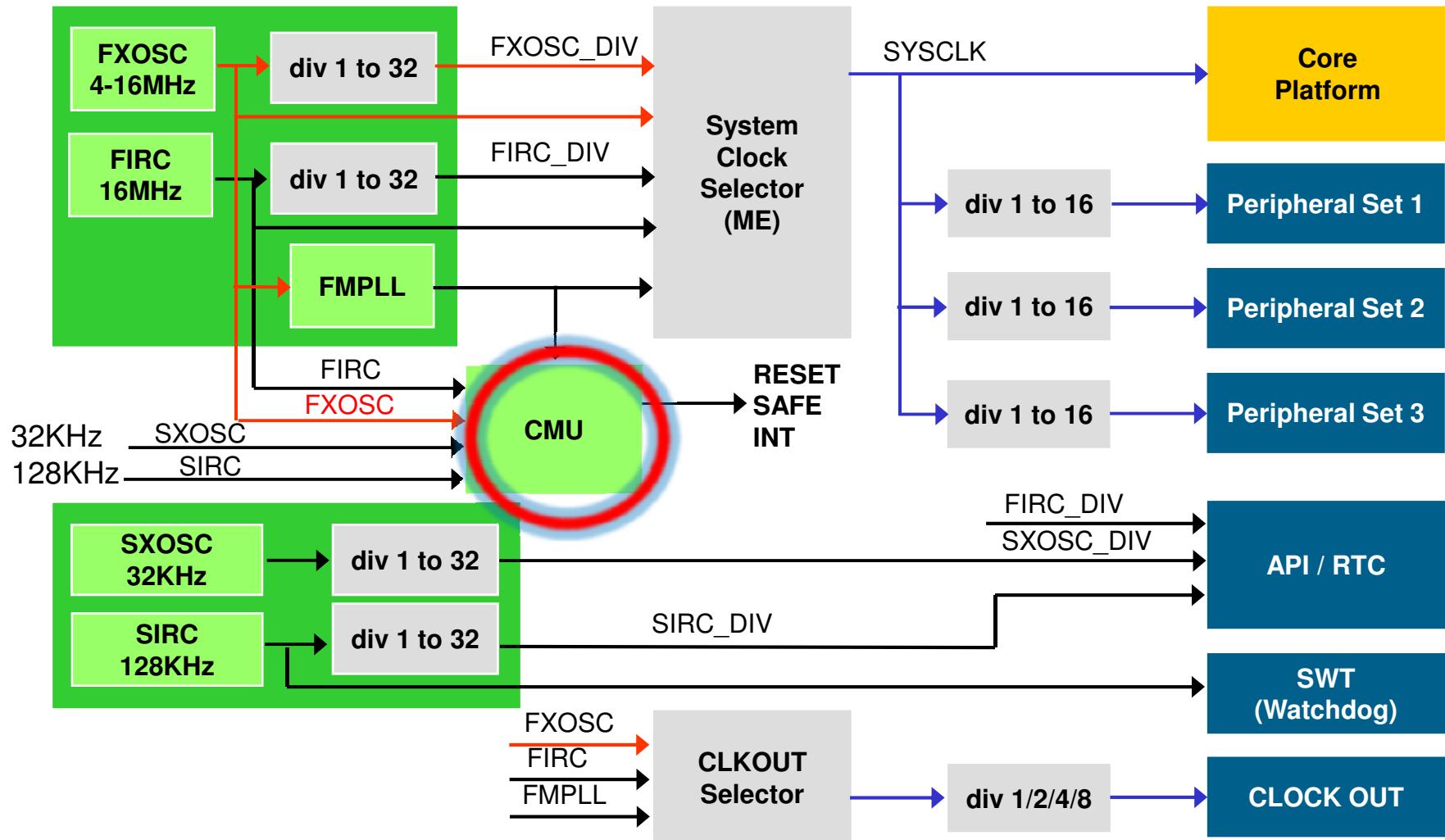
Offset 0x0004

Access: User read/write



Bit	Bit name	Bit description
0	STRB_BYPAS	Bypass the STRB signal used inside PLL to latch the correct values for control bits (INC_STEP, MOD_PERIOD and SPRD_SEL).
2	SPRD_SEL	The SPRD_SEL control the spread type in Frequency Modulation mode: centre spread or down spread.
3-15	MOD_PERIOD	The MOD_PERIOD field is the binary equivalent of the modulation period.
16	FM_EN	The FM_EN enables the frequency modulation
17-31	INC_STEP	The INC_STEP field is the binary equivalent of the incremental step value

Clock Monitor Unit (CMU)



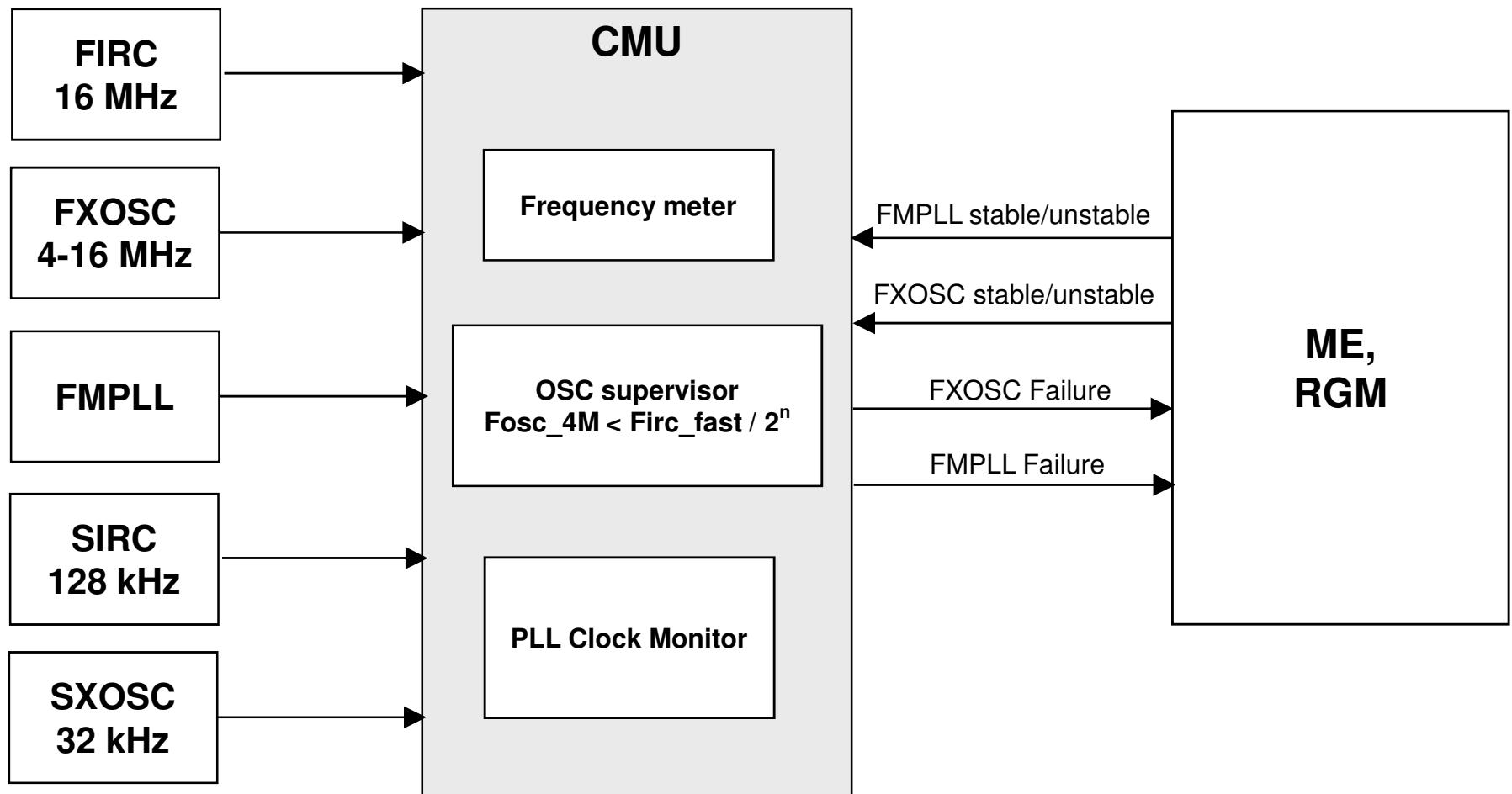
CGM Clock Monitoring Unit (CMU)

- ▶ The task of the CMU is to **permanently supervise** the integrity of the various product's clock sources, e.g. FXOSC or FMPLL, if either
 - FXOSC clock frequency lower than FIRC / 2^n
 - PLL clock frequency upper or lower frequency boundaries defined in CMU registers
- ▶ Then the **Mode Entry module** is notified in order to switch to SAFE mode with FIRC as clock source.
- ▶ The second task is **frequency measurement**. It allows to measure the deviation of a clock (FIRCs or SXOSC) by measuring its frequency versus FXOSC as reference clock. This measurement can be used to improve Real Time Counter precision based on FIRC (16MHz) or SXOSC (32kHz xtal)

CGM Clock Monitoring Unit (CMU)

- ▶ Main task: Permanently supervise integrity of clock sources
- ▶ Features:
 - External oscillator clock monitoring with respect to FIRC/n clock
 - FMPLL clock upper & lower frequency monitoring with respect to FIRC/4 clock
 - Event generation for various failures detected inside monitoring unit
 - RC oscillator frequency measurement
 - Useful for on-chip RC oscillator calibration
 - Useful for correcting/calculating time deviation of counter clocked by RC oscillator

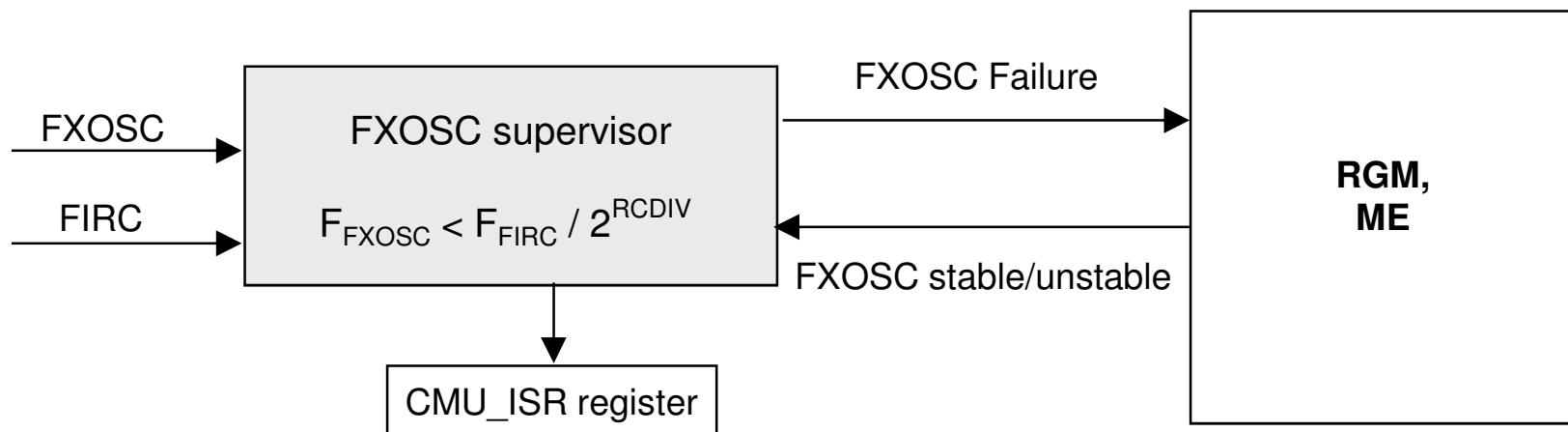
CGM CMU Block Diagram



CGM CMU MPC560xB

Crystal Clock Monitor

- ▶ Crystal clock monitor is active only when ME provides the info that FXOSC is valid
- ▶ If $\text{FXOSC} < \text{FIRC} / 2^{\text{RCDIV}}$ (CMU_CSR[RCDIV] bits), then
 - an event pending bit CMU_ISR[OLRI] is set.
 - a failure event is signaled to the RGM which in turn can generate a RESET, transition to SAFE mode, or generate an interrupt request



CMU: Crystal Clock Monitor Reset Default Values

► **CAUTION:** Before enabling crystal clock input in a mode configuration, verify the proper compare frequency divider.

- MPC560xB, MPC560xS: reset default RCDIV = 3, so
$$\text{Freq}_{\text{FIRC}} / 2^{\text{CMU_CSR[RCDIV]}} = 16 \text{ MHz} / 8 = 2 \text{ MHz}$$
- MPC560xP: reset default RCDIV = 0, so
$$\text{Freq}_{\text{FIRC}} / 2^{\text{CMU_CSR[RCDIV]}} = 16 \text{ MHz} / 1 = 16 \text{ MHz}$$

CMU: Crystal Clock Monitor OLR Event

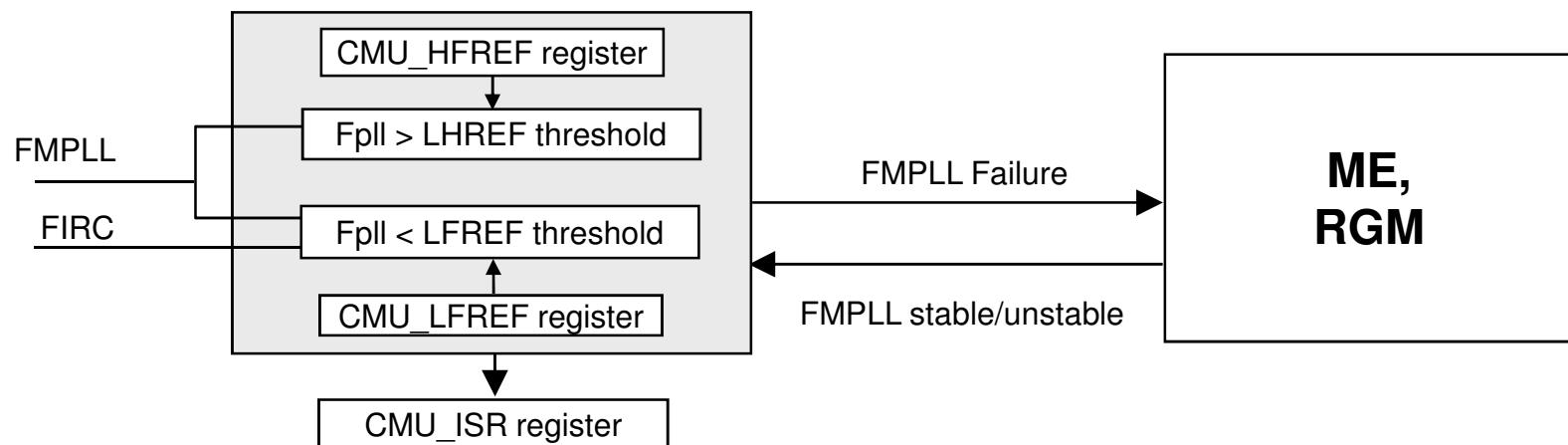
- ▶ Oscillator Less than Reference event occurs when the FXOSC appears too slow and sets:
 - ▶ **CMU_ISR[OLRI]**
 - No interrupt or other automatic action can be generated – just sets the bit
 - ▶ **RGM_FES[F_CMU_OLR]**
 - Action taken is per table below:

Functional Event Reset Disable: FXOSC freq. lower than reference	Functional Event Alternate Request: Alternate Request for FXOSC freq. lower than reference	Action
RGM_FERD [D_CMU_OLR]	RGM_FEAR [AR_CMU_OLR]	
0 (reset default)	-	Reset sequence
1	0 (reset default)	SAFE mode request
1	1	Interrupt request – INTC #56

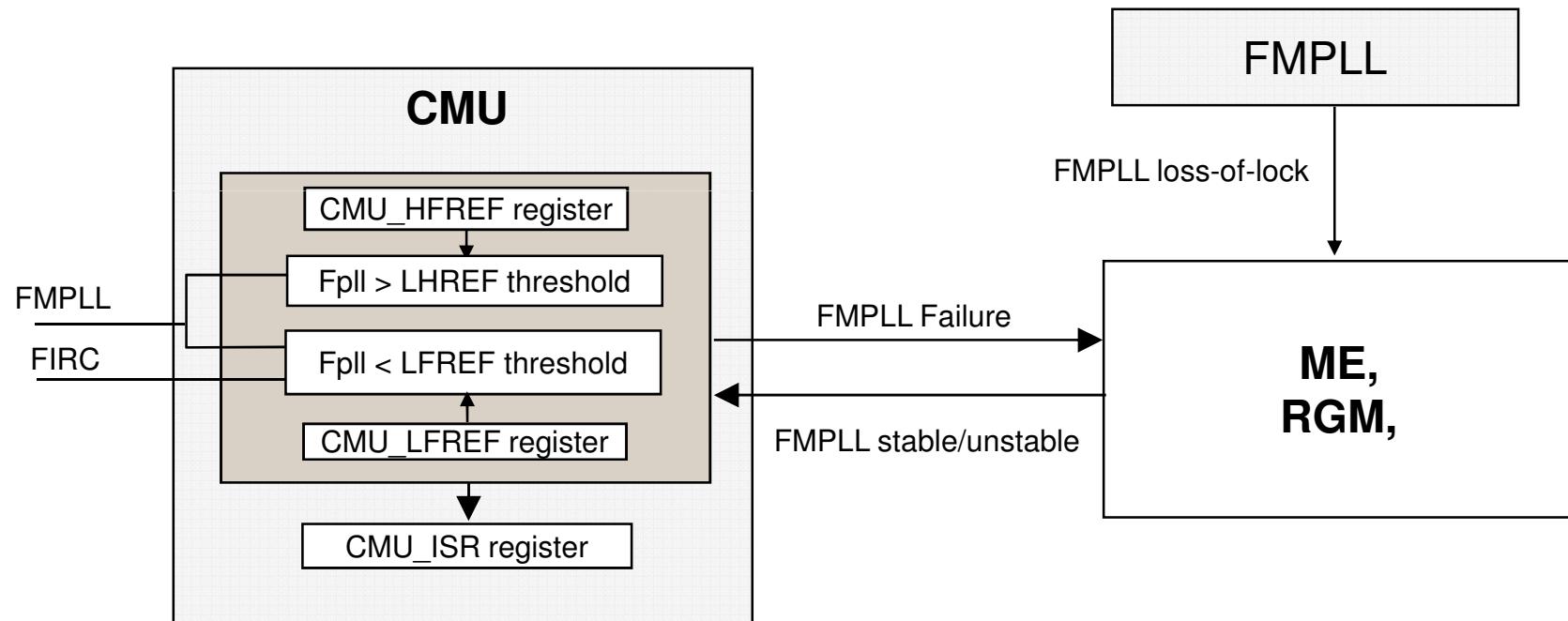
- ▶ FMPLL monitoring is enabled at CMU_CSR(CME)
- ▶ Monitoring is active only when ME provides the info that FMPLL is valid.
- ▶ Depending on the following conditions
 - If F_{FMPLL} is lower than HREF[11:0] bits in CMU_LFREFR
 - If F_{FMPLL} is higher than HREF[11:0] bits in CMU_HFREFR

► Then for each matching:

- a dedicated event pending bit in CMU_ISR is set.
- a failure event is output to the RGM & ME which can **generate** a transition to **SAFE mode**, an **interrupt** or a **reset** sequence.



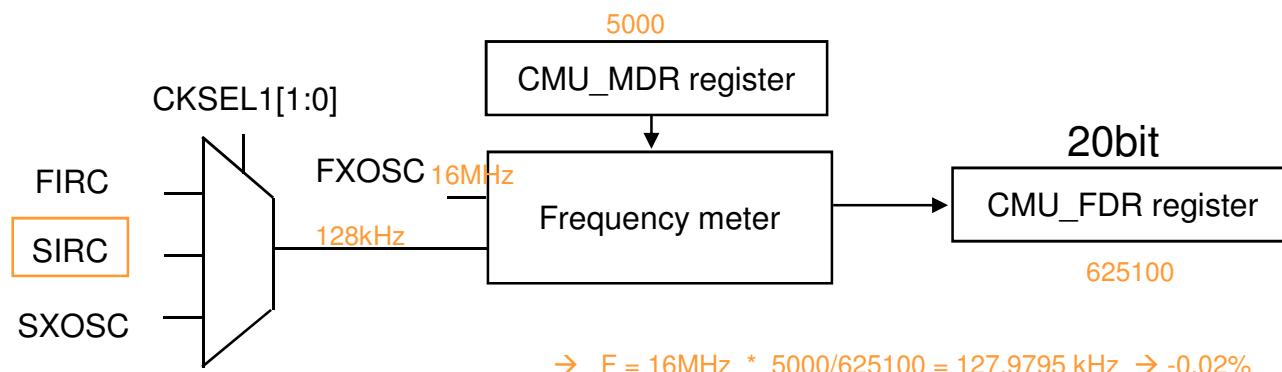
- ▶ The FMPLL also performs a monitoring of the output clock frequency and generate a **loss-of-lock** event to the RGM & ME which can generate a transition to **SAFE mode**, an interrupt or a **reset** sequence.



- ▶ The purpose of frequency meter is to calibrate the IRC oscillators and SXOSC (32 kHz) using a known frequency. FXOSC clock is the reference.

▶ How it works

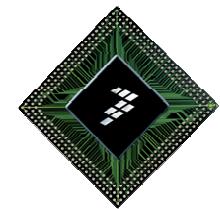
- Write in CMU_MDR the number of clock period to be counted
- CK_RC frequency measure starts/ends as soon as CMU_CSR(FSR) bit is set to 1/0.
- FSR bit is reset by hardware when the measure is finished, the CMU_FDR contains the number of FXOSC periods counted in the meantime.
- Then $F = \text{FXOSC} * \text{MDR} / \text{FDR}$





Designing with Freescale

Modes: Intro and RUN Modes



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

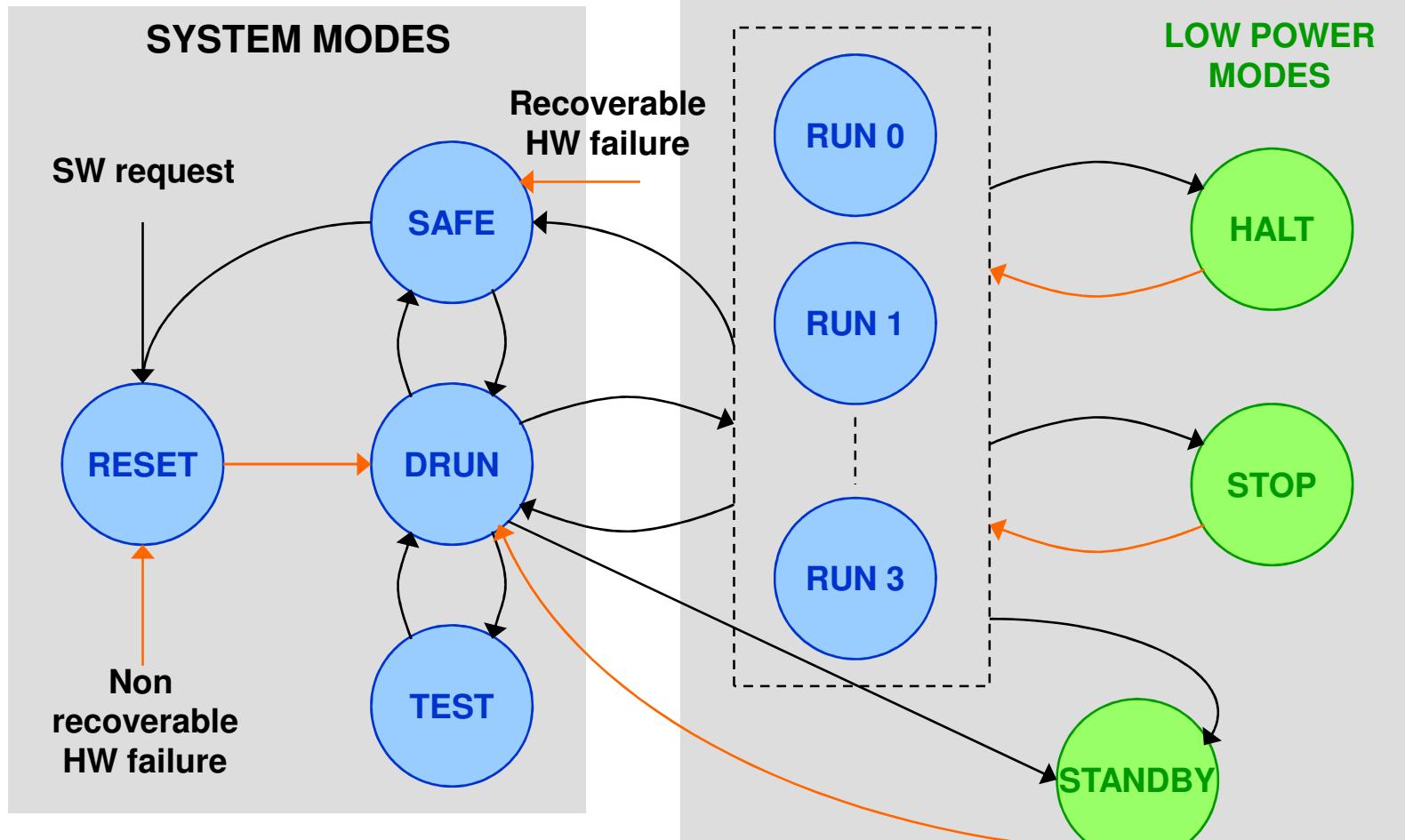
 **freescale**™
semiconductor

Mode Overview

- ▶ The Mode Entry Module (MC_ME) provides SYSTEM modes and USER modes :
 - SYSTEM: RESET, DRUN (Default RUN), SAFE and TEST
 - USER: RUN(0..3), HALT, STOP and STANDBY
- ▶ For each mode the following parameters are configured/controlled
 - System clock sources (ON/OFF)
 - System clock source selection
 - Flash power mode (ON, low power, power down)
 - Pad output driver state (Can disable Pad Output drivers, enabling high impedance mode)
 - Peripherals' clock (gated/clocked)

USER MODES

LOW POWER MODES



→ HW triggered transition
→ SW triggered transition

Device Start-up: MPC560xB System modes

- **RESET**

- Completely managed by HW
- The flash initialization is executed by hardware **while** device in **RESET**

- **DRUN (Default RUN)**

- Mode **automatically entered out of RESET or STANDBY**
- This mode is used by the application to **configure the device out of RESET or out of STANDBY**

- **SAFE**

- Mode **automatically entered on “recoverable HW failure detection”** like oscillator, PLL or voltage failure
- Device in a SAFE configuration with sysclk = FIRC & output at high impedance (if configured so)

- **TEST**

- Allow device self tests like flash checksum, RAM BIST

► RUN[0..3]

- Full performance available
- Support WAIT instruction to stop the core with the capability to restart with very short latency (< 4 system clocks)

► HALT

- Core stopped but system clock can remain the same as in RUN mode
- Selective peripheral clock gating
- Flash can be put in low power mode
- Useful to reduce device consumption during a slow serial communication e.g. LIN frame transmission or reception
- Exit by interrupt, pin transition, or RTC/API timeout

► STOP

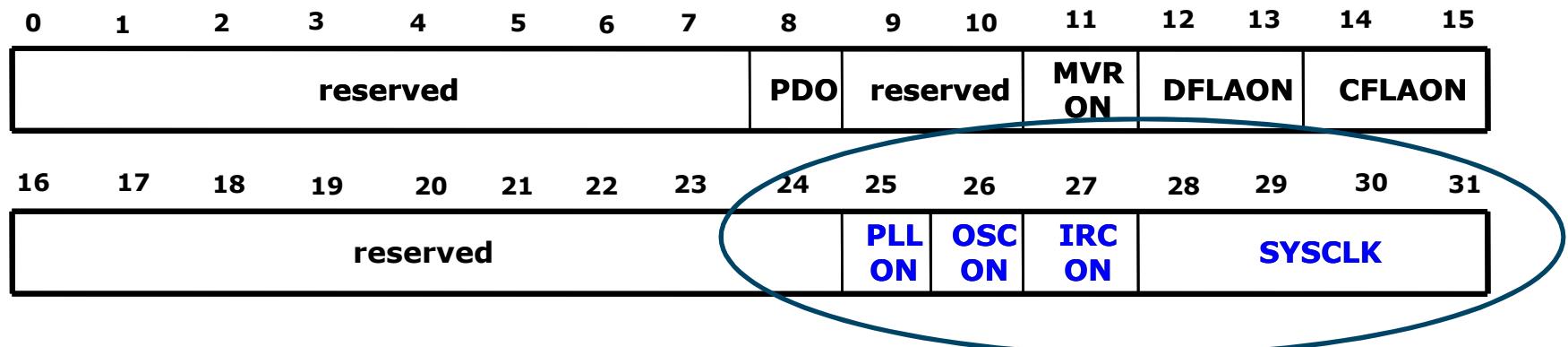
- Provides additional low power features beyond HALT, including:
 - System clock can be disabled
 - PLL is always disabled
 - Upon STOP mode exit, only FIRC can be system clock

► STANDBY

- Mode providing the lowest possible consumption
- Most functions (digital and analog) of the device are not powered
- Powers only the back-up logic (e.g. RTC/API, preserves wake-up inputs, part of SRAM.)
- On STANDBY exit, the processor uses the RESET vector or a SRAM Vector if enabled
 - WISR register in the Wake Up Unit can be used to verify the wake up source
- Exit by RTC/API timeout or pin transition

ME_xxx_MC - Mode Configuration Registers

- Each mode has a Mode Configuration register.
- Example: ME_DRUN_MC
- Key RUN mode configurations are circled:



- **PDO:** Disable pad outputs (put in hi Z)
- **MVRON:** control VREG on/off
- **CFLAON/DFLAON:**
 - control code / data flash module
 - Normal
 - Low Power
 - Power Down
- **PLLON:** control PLL on/off
- **OSCON:** control XOSC on/off
- **IRCON:** control IRC16M on/off
- **SYSCLK:** select system clock

Mode Configurations Example

- It is useful to keep a table of mode configurations used.
- Example below uses two USER modes. (Per AN2865 rev 4)

Table 93. Mode Configurations for MPC56xxB/P/S DSPI SPI to SPI Example
Modes are enabled in ME_ME Register.

Mode	Mode Config. Register	Settings									
		Mode Config. Register Value	sysclk Selection	Clock Sources				Memory Power Mode		Main Voltage Reg.	I/O Power Down Ctrl
				16MHz IRC	XOSC0	PLL0	PLL1 (MPC 56xxP/S only)	Data Flash	Code Flash		
DRUN	ME_DRUN_MC	0x001F 0010 (default)	16 MHz IRC	On	Off	Off	Off	Normal	Normal	On	Off
RUN0	ME_RUN0_MC	0x001F 007D	PLL0	On	On	On	Off	Normal	Normal	On	Off
Other modes are not used in example											

Device Start-up: MC_ME Mode Enable Register

- Modes must be enabled before entering them per the Mode Enable register
- An interrupt is generated if attempt is made to enter disabled mode

Address 0xC3FD_C008																Access: User read, Supervisor read/write, Test read/write																			
R				W				Reset				R				W				Reset				R				W				Reset			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R	0	0	STANDBY0	0	0	STOP0	0	HALT0	RUN3	RUN2	RUN1	RUN0	DRUN	SAFE	TEST	W	0	0	0	0	0	0	1	1	0	1	RESET	0	1	0	1	0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0			

Figure 5-4. Mode Enable Register (ME_ME)

This register allows a way to disable the device modes which are not required for a given device. RESET, SAFE, DRUN, and RUN0 modes are always enabled.

Device Start-up: MC_ME Mode Global Status Register

Address 0xC3FD_C000																Access: User read, Supervisor read, Test read									
R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15									
W	S_CURRENT_MODE	S_MTRANS	S_DC	0	0	S_PDO	0	0	S_MVR	S_DFLA	S_CFLA														
Reset	0 0 0 0	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1													
R	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	S_FPLL	S_FKOSC	S_FIRC	S_SYSCLK					
W	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0									

Figure 5-2. Global Status Register (ME_GS)

S_CURRENT_Mode - Current device mode status

0000 =RESET
 0001 =TEST
 0010 =SAFE
 0011 =DRUN
 0100 =RUN0
 0101 =RUN1
 0110 =RUN2
 0111 =RUN3
 1000 =HALT0
 1010 =STOP0
 1101 =STANDBY0

S_MTRANS - Mode transition status
 0 Mode transition process is not active
 1 Mode transition is ongoing

S_DC - Device current consumption status
 0 = Device consumption is low enough to allow powering down of main voltage regulator
 1 = Device consumption requires main voltage regulator to remain powered regardless of mode configuration

S_PDO - Output power-down status specifies output power-down status of I/Os. This bit is Asserted whenever outputs of pads are forced to high impedance state or the pads power sequence driver is switched off.

Device Start-up: MC_ME Mode Global Status Register (Cont'd)

Address 0xC3FD_C000																Access: User read, Supervisor read, Test read								
R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15								
W	S_CURRENT_MODE	S_ATRANS	S_DC	0	0	S_PIO	0	0	S_MVR	S_DFLA	S_CFLA													
Reset	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1								
R	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
W					0	0	0	0	S_ENRF	S_FXOSC	S_FIRC		S_SYSCLK											
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0								

Figure 5-2. Global Status Register (ME_GS)

MS_MVR - Main voltage regulator status
 0 = Main voltage regulator is not ready
 1 = Main voltage regulator is ready for use

S_DFLA - Data flash availability status
 00 = Data flash is not available
 01 = Data flash is in power-down mode
 10 = Data flash is in low-power mode
 11 = Data flash normal mode & available for use

S_FMPLL - FMPLL Locked Status
 0 = FMPLL is not stable
 1 = FMPLL is providing stable clock

F_FXOSC – Fast external Crystal osc (4-16MHz)
 0 = Not stable
 1= Stable

S_CFLA - Code flash availability status
 00 =Code flash is not available
 01 =Code flash is in power-down mode
 10 =Code flash is in low-power mode
 11 =Code flash is in normal mode and available for use

F_FIRC - Fast internal RC osc (16MHz)
 0 = Not stable
 1= Stable

S_SYSCLK - System clock switch status -Identifies clock currently used by the system.
 0000 =16 MHz int. RC osc.
 0001 =div. 16 MHz int. RC osc.
 0010 =4-16 MHz ext. XTAL osc.
 0011 =div. ext. XTALI osc.
 0100 =freq. mod. PLL
 1111 = System clock is disabled

Peripheral Clock Gating Control

- Each peripheral can be associated with a particular clock gating policy
- The policy is determined by two groups of peripheral configuration registers:
 - ME_RUN_PC0:7 for RUN modes
 - ME_LP_PC0:7 for Low Power modes
- Clocks to peripherals are gated off unless enabled for that mode
- Example (per AN2865 rev 4):

Table 94. Peripheral Configurations for MPC56xxB/P/S DSPI SPI to SPI Example
Low power modes are not used in example.

Peri- pheral Config.	Peri. Config. Register	Enabled Modes								Peripherals Selecting Configuration	
		RUN3	RUN2	RUN1	RUN0	DRUN	SAFE	TEST	RESET	Peripheral	PCTL Reg. #
PC1	ME_ RUNPC_ 1	0	0	0	1	0	0	0	0	DSPI 0 DSPI 1 SIUL (MPC56xxB/S)	4 5 68
Other peripheral configurations are not used in example											

MC_ME Peripheral Configuration Registers RUN Modes

Defines a selection of 8 possible RUN mode configurations for a peripheral

Address 0xC3FD_C080 - 0xC3FD_C09C																Access: User read, Supervisor read/write, Test read/write							
R				W				Reset				R				W				Reset			
0	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	17	18	19	0	0	0	0	0	0	0	0	RUN3	RUN2	RUN1	RUN0	DRUN	SAFE	TEST	RESET	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5-21. Run Peripheral Configuration Registers (ME_RUN_PC0...7)

These registers configure eight different types of peripheral behavior during run modes.

Device Start-up: MC_ME Peripheral Control Registers

For each peripheral, there is a ME_PCTLx register to control clock gating to that peripheral:

- selects one of the 8 **Run** peripheral set configurations
- selects one of the 8 **Low Power** peripheral set configurations
- enables/disables freezing the clock during debug

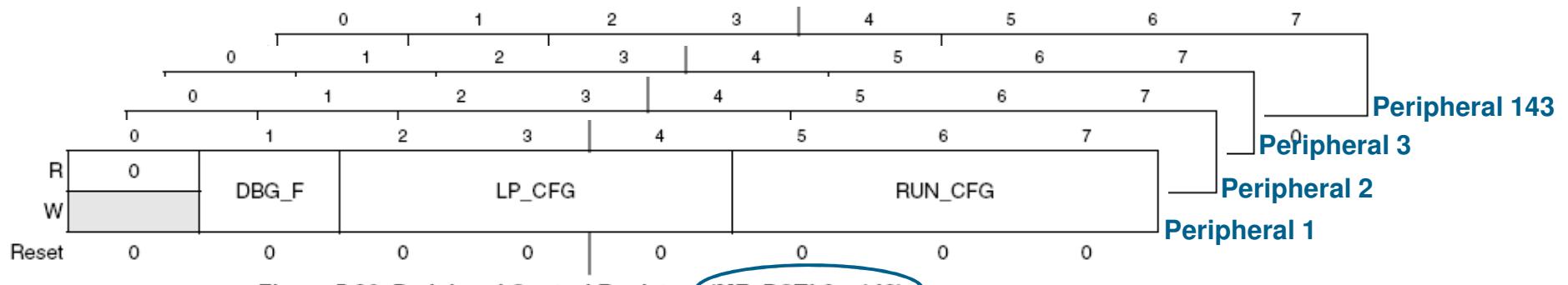


Figure 5-23. Peripheral Control Registers (ME_PCTL0...143)

These registers select the configurations during run and non-run modes for each peripheral.

Mode Entry: SW and HW Transitions

► Software handled transition

- A transition is requested writing a key protected sequence in ME_MCTL
- Mode Entry configures the modules according to the ME_xxx_MC register of the target mode
- Once all modules are ready the new mode is entered
- Transition completion signalling: status bit/interrupt
- Note: Modification of a ME_xxx_MC register (even the current one) is taken into account on next mode “xxx” entry

► Hardware triggered transition

- Exit from low power mode
- SAFE transition caused by HW failure
- RESET transition caused by HW failure

Device Start-up: MC_ME Peripheral Status Registers 0...3

Example: Peripheral Status Register 0

Address 0xC3FD_C060																Access: User read, Supervisor read, Test read									
R																									
0	1	2	3	4	5	6	7	S_DMA_CH_MUX	9	10	11	12	13	14	15	S_FlexCAN5	S_FlexCAN4	S_FlexCAN3	S_FlexCAN2	S_FlexCAN1	S_FlexCAN0				
W																									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	17	18	19	20	21	22	23	S_DSPI5	S_DSPI4	S_DSPI3	S_DSPI2	S_DSPI1	S_DSPI0												
R			S_LINFlex9	S_LINFlex8																					
W																									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

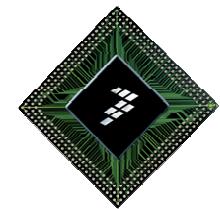
Figure 5-17. Peripheral Status Register 0 (ME_PS0)

- ▶ Review of Key Points
- ▶ RUN mode configurations allow
 1. Enabling/disabling system clock sources
 2. Selecting appropriate system clock
 3. Gating clocks to peripherals
- ▶ Peripheral clocks can be divided as needed on a set basis
- ▶ Example PLL: Initializing System Clock



Designing with Freescale

Real Time Clock / Autonomous Periodic Interrupt (RTC/API)



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**TM
semiconductor

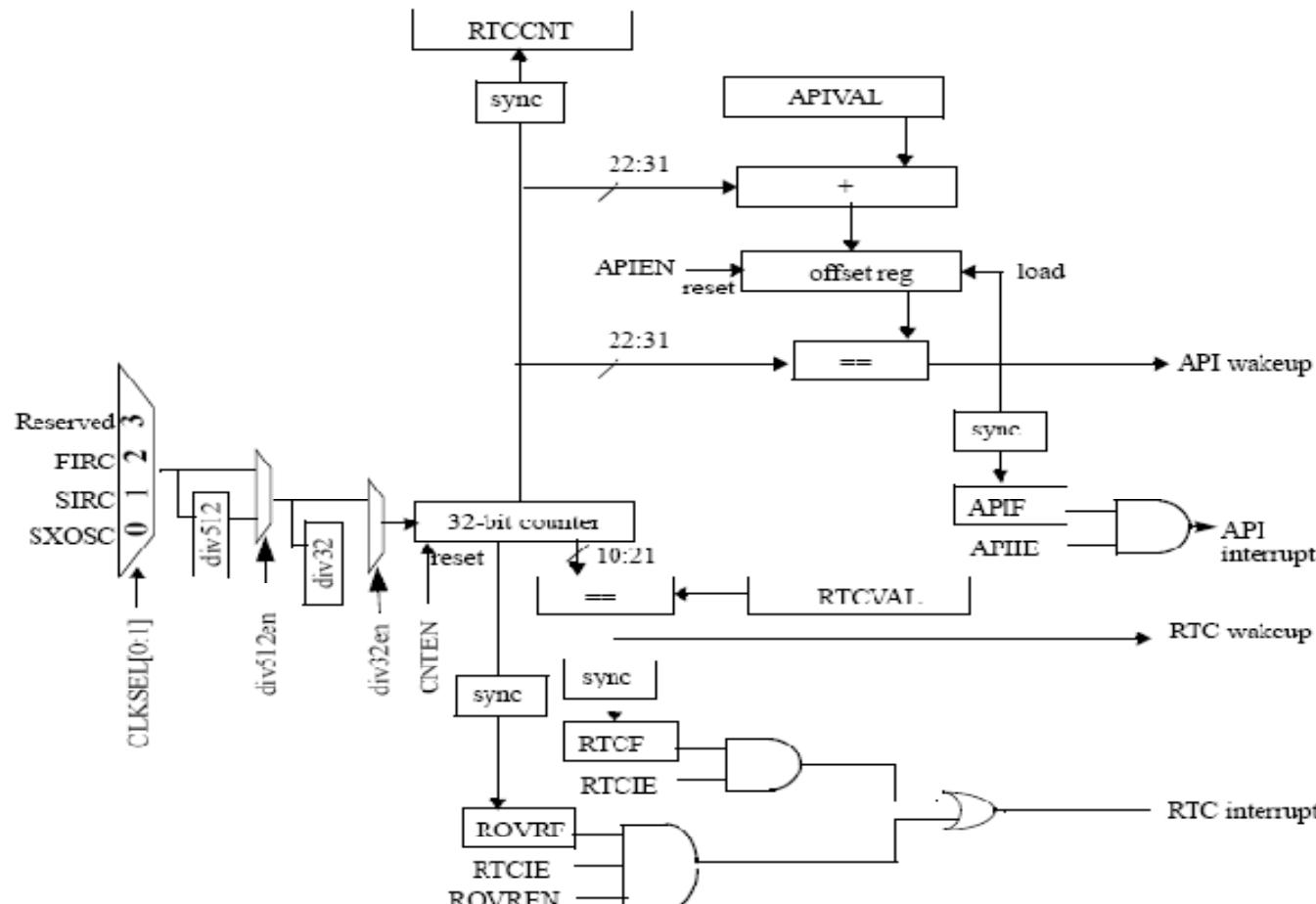
RTC/API Overview

- RTC is a free running counter used for time keeping applications
 - Can generate interrupt independent of RUN or Low Power Mode
 - Can first generate a wakeup for low power exit, then interrupt
 - Continues counting thru resets except for power up reset
- 3 Selectable counter clock sources with optional prescalers:
 - SIRC (128 KHz)
 - SXOSC (32 KHz)
 - FIRC (16 MHz)
- API provides regular timeouts for wakeup / interrupt
 - Compares lower 10 bits RTC to 10 bit compare value
 - At match, automatically adds programmed value for next compare
- RTC provides longer timeout for wakeup / interrupt
 - Compares 12 bits above API to 12 bit compare value

Freescale, the Freescale logo, AliVec, C-S, CopyNet, CodeWarrior, ColdFire, C-Ware, Intrinsic, PowerQUICC, StaCore, and Symphony are trademarks of Freescale Semiconductor, Inc. Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.



Peripherals RTC / API



RTC/API Clock, Divider, Timeouts (AN2865, rev 4, Table 49)

Clock Source	div512 RTC_RTCC [div512]	div 32 RTC_RTCC [div32]	RTC Counter Input Clock Frequency	RTC Counter Input Clock Period (1 / (RTC Counter Input Clock Freq.))	Min. API Timeout (1 × RTC Counter input clock period)	Max. API Timeout ((2 ¹⁰ - 1) × RTC Counter input clock period)	Min. RTC Timeout (2 ¹⁰ × RTC Counter input clock period)	Max. RTC Timeout ((2 ²² - 1) × RTC Counter input clock period)	RTC Rollover Timeout (2 ³² × RTC Counter input clock period)
128 kHz SIRC ¹ with SIRCDIV = 0	0	0	128 kHz	~7.8 µsec	~7.8 µsec	~ 8 msec	~ 8 msec	~31 sec	~8.9 hrs
	0	1	4 kHz	250 µsec	250 µsec	~ 256 msec	256 msec	~17 min	~12 days
	1	0	250 Hz	4 msec	4 msec	~ 4.09 sec	~4.10 sec	~4.4 hrs	~6.3 mo
	1	1	7.8125 Hz	128 msec	128 msec	~131 sec	~131 sec	~6 days	~17 yrs
16 MHz FIR ² (125 x SIRC)	0	0	16 MHz	62.5 nsec	62.5 nsec	~64 µsec	64 µsec	0.25 sec	~4.3 min
	0	1	500 kHz	2 µsec	2 µsec	~2 msec	2.048 msec	8 sec	~2.8 hrs
	1	0	31.25 kHz	32 µsec	32 µsec	~33 msec	~33 msec	~2.1 min	~1.5 days
	1	1	~977 Hz	1.024 msec	1.024 msec	~1048 sec	~1049 sec	~1.1 hrs	~1.6 mo
32 kHz SXOSC ³ or 128 kHz SIRC/4 ⁴	0	0	32 kHz	31.25 µsec	31.25 µsec	~32 msec	32 msec	~2.1 min	~1.5 days
	0	1	1 kHz	1 msec	1 msec	1.023 sec	1.024 sec	~1.1 hrs	~1.6 mo
	1	0	62.5 Hz	16 msec	16 msec	~16 sec	~16 sec	~18 hrs	~2.1 yrs
	1	1	~2 Hz	512 msec	512 msec	~524 sec	~524 sec	~24 days	~67 yrs
FXOSC ⁵	0	0	8 MHz	125 nsec	125 nsec	~128 sec	~128 sec	0.5 sec	~8.6 min
	0	1	250 kHz	4 µsec	4 µsec	~4.1 msec	~4.1 msec	16 sec	~4.6 hrs
	1	0	15.625 kHz	64 µsec	64 µsec	~65 msec	~66 msec	~4.3 min	~3 days
	1	1	~488 Hz	2.048 msec	2.048 msec	~2.1 sec	~2.1 sec	~2.3 hr	~3.2 mo

RTC/API – RTC Control Register

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
R	CNT EN	RTCI E	FRZ EN	ROV REN	RTCVAL[0:11]												
W																	
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
R	API EN	APIIE	CLKSEL[0:1]]	DIV5 12 EN	DIV3 2EN	APIVAL[0:9]											
W																	
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 32-5. RTC Control Register (RTCC)

RTC/API – RTC Status Register & RTC Count Register

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R			RTC													
W			F													
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R			API			ROV										
W			F			R										
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 32-6. RTC Status Register (RTCS)

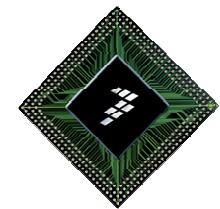
0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	3	3	
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	
R																											
W																											
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 32-7. RTC Counter Register (RTCCNT)



Designing with Freescale

Wake Up Unit



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**™
semiconductor

► External wakeup/interrupt support with

- 3 system interrupt vectors for up to 18 interrupt sources for Bolero MPC5602/3/4B
- 4 system interrupt vectors for up to 24 interrupt sources for Bolero MPC5605/6/7B
- analog glitch filter per each wakeup line
- independent interrupt mask
- edge detection
- configurable system wakeup triggering from all interrupt sources
- configurable pull-up (recommended)

► Non-maskable interrupt support with

- edge detection
- 1 NMI source with bypassable glitch filter
- independent interrupt destination: non-maskable interrupt, critical interrupt, or machine check request

► On-chip wakeup support

- 2 wakeup sources (RTC and API)
- wakeup status mapped to same register as external wakeup/interrupt status

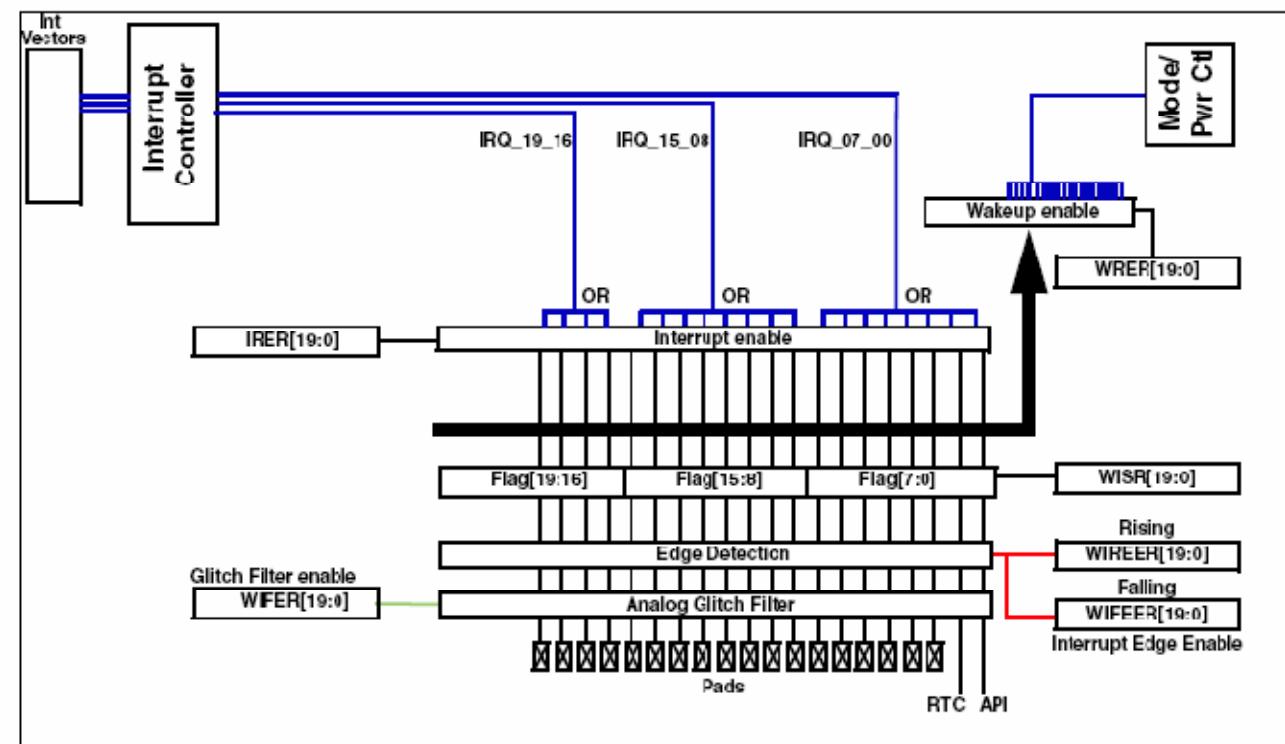
Freescale, the Freescale logo, Altivec, C-1, CodeTest, CodeWarrior, ColdFire, C-Ware, iMobileST, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

Peripherals Wakeup Unit block diagram

- ▶ The WKPU remains powered in low power mode
- ▶ Each pin can:

- Issue only a wakeup, or an interrupt, or both
- be sensitive on rising, falling or both edges
- has an analog glitch filter
- internal pull-up

In addition the chip provides external interrupts through different external pins



MPC560xB Wake-up line Interrupts

► Interrupt Vector 0

- API
- RTC
- PA[1], GPIO[1], E0UC[1], NMI, WKUP[2]
- PA[2], GPIO[2], E0UC[2], WKUP[3]
- PB[1], GPIO[17], CAN0RX, WKUP[4]
- PC[11], GPIO[43], CAN1RX, CAN4RX2, WKUP[5]
- PE[0], GPIO[64], E0UC[16], CAN5RX2, WKUP[6]
- PE[9], GPIO[73], CAN2RX3, CAN3RX2, E0UC[23], WKUP[7]

► Interrupt Vector 1

- PB[10], GPIO[26], ANS[2], WKUP[8]
- PA[4], GPIO[4], E0UC[4], WKUP[9]
- PA[15], GPIO[15], CS0_0, SCK_0, WKUP[10]
- PB[3], GPIO[19], LIN0RX, SCL, WKUP[11]
- PC[7], GPIO[39], LIN1RX, WKUP[12]
- PC[9], GPIO[41], LIN2RX, WKUP[13]
- PE[11], GPIO[75], LIN3RX, CS4_1, WKUP[14]
- PF[11], GPIO[91], WKUP[15] (not in 100-pin package)

► Interrupt Vector 2

- PF[13], GPIO[93], E1UC[26], WKUP[16] (not in 100-pin package)
- PG[3], GPIO[103], E1UC[12], WKUP[17] (not in 100-pin package)
- PG[5], GPIO[105], E1UC[14], WKUP[18] (not in 100-pin package)
- PA[0], GPIO[0], E0UC[0], CLKOUT, WKUP[19]

► Interrupt Vector 3 (on MPC5605/6/7B only)

- PG[7], GPIO[103], LIN6RX, E1UC[16], E1UC[30], WKUP[20] (on MPC5605/6/7B only)
- PG[9], GPIO[105], LIN7RX, E1UC[18], SCK_2, WKUP[21] (on MPC5605/6/7B only)
- PF[9], GPIO[89], CAN3RX, CAN2RX, E1UC[1], CS5_0, WKUP[22] (on MPC5605/6/7B only)
- PI[3], GPIO[131], LIN9RX, E0UC[31], WKUP[23] (on MPC5605/6/7B only)
- PI[1], GPIO[129], LIN8RX, E0UC[29], WKUP[24] (on MPC5605/6/7B only)

Wakeup Number	MPC560xB			MPC560xS		
	Port	SIU_PCR #	Wakeup IRQ to INTC	Port	SIU_PCR	Wakeup IRQ to INTC
WKUP0	API	n.a.	WakeUp_IRQ_0	PA0	PCR0	WakeUp_IRQ_0
WKUP1	RTC	n.a.	WakeUp_IRQ_0	PB1	PCR17	WakeUp_IRQ_0
WKUP2	PA1	PCR1	WakeUp_IRQ_0	PB3	PCR19	WakeUp_IRQ_0
WKUP3	PA2	PCR2	WakeUp_IRQ_0	PB4	PCR20	WakeUp_IRQ_0
WKUP4	PB1	PCR17	WakeUp_IRQ_0	PB9	PCR25	WakeUp_IRQ_0
WKUP5	PC11	PCR43	WakeUp_IRQ_0	PB10	PCR26	WakeUp_IRQ_0
WKUP6	PE0	PCR64	WakeUp_IRQ_0	PB12	PCR28	WakeUp_IRQ_0
WKUP7	PE9	PCR73	WakeUp_IRQ_0	PC0	PCR30	WakeUp_IRQ_1
WKUP8	PB10	PCR26	WakeUp_IRQ_1	PC10	PCR40	WakeUp_IRQ_1
WKUP9	PA4	PCR4	WakeUp_IRQ_1	PF0	PCR70	WakeUp_IRQ_1
WKUP10	PA15	PCR15	WakeUp_IRQ_1	PF2	PCR72	WakeUp_IRQ_1
WKUP11	PB3	PCR19	WakeUp_IRQ_1	PF3	PCR73	WakeUp_IRQ_1
WKUP12	PC7	PCR39	WakeUp_IRQ_1	PF5	PCR75	WakeUp_IRQ_1
WKUP13	PC9	PCR41	WakeUp_IRQ_1	PF6	PCR76	WakeUp_IRQ_1
WKUP14	PE11	PCR75	WakeUp_IRQ_1	PF8	PCR78	WakeUp_IRQ_2
WKUP15	PF11	PCR91	WakeUp_IRQ_1	PF11	PCR81	WakeUp_IRQ_2
WKUP16	PF13	PCR93	WakeUp_IRQ_2	PF13	PCR83	WakeUp_IRQ_2
WKUP17	PG3	PCR99	WakeUp_IRQ_2	PJ4	PCR108	WakeUp_IRQ_2
WKUP18	PG5	PCR101	WakeUp_IRQ_2	PJ6	PCR111	WakeUp_IRQ_2
WKUP19	PA0	PCR0	WakeUp_IRQ_2	API	n.a.	WakeUp_IRQ_2
WKUP20	(none)	(none)	(none)	RTC	n.a.	WakeUp_IRQ_2

Wakeup Sources (Table 50 AN2865 rev 4)

- ▶ Table shows wakeup sources for MPC5604B, MPC5606S
- ▶ Smaller packages do not contain all ports

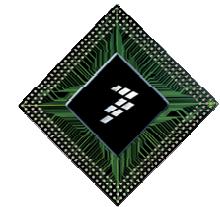
WKUP Registers

Name	Symbol	Description
Wakeup / Interrupt Status Flag Register	WISR	Flags event as defined by WIREER and WIFEER
Wakeup / Interrupt Rising-Edge Event Enable Register	WIREER	Enables rising-edge event
Wakeup / Interrupt Falling-Edge Event Enable Register	WIFEER	Enables falling-edge event
Interrupt Request Enable Register	IRER	Enables flags to cause interrupt req.
Wakeup / Interrupt Filter Enable Register	WIFER	Enables analog glitch filter on external pad input (filters glitch < 40 ns, passes signals > 1000 ns)
Wakeup / Interrupt Pull-up Enable Register	WIPUER	Enables pull-up on external pad (use for all pads to minimize leakage)
NMI Configuration Register	NCR	Configuration settings for NMI
NMI Status Flag Register	NSR	Holds NMI status flags



Designing with Freescale

System Integration Unit Lite (SIUL)



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**™
semiconductor

Peripherals

SIUL Introduction

► Pad Control and IOMUX configuration;

- Intended to configure the electrical parameters and mux'ing of each pad;
- may simplify PCB design by multiple alternate input / output functions

► GPIO ports with data control;

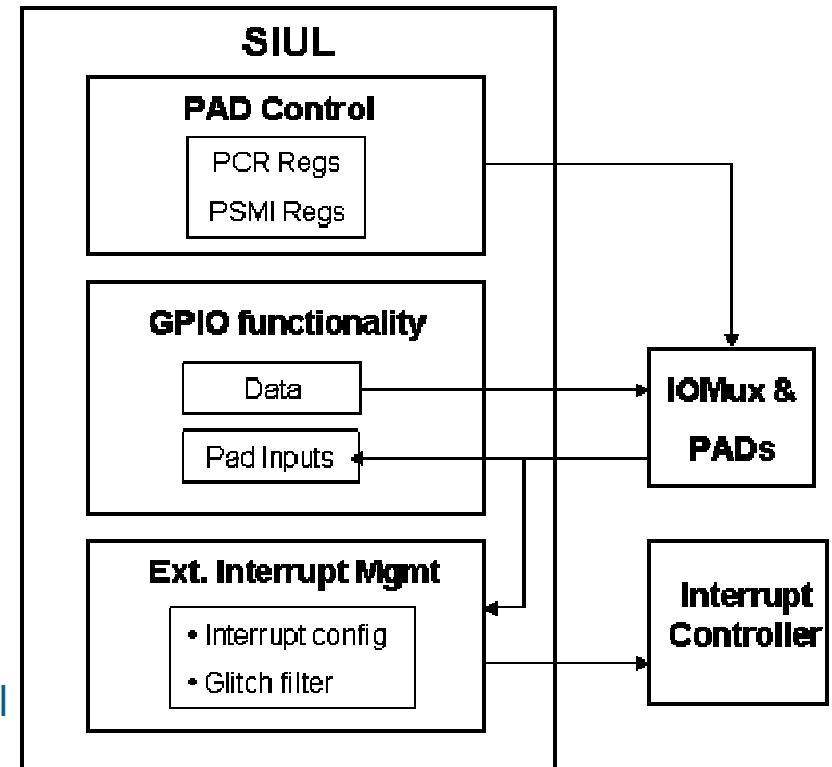
- Different access mechanisms to the GPIO data registers in order to allow port accesses or bit manipulation without the need of R-M-W operations

► External interrupt management

- Allows the enabling and configuration (such as filtering window, edge and mask setting) of digital glitch filters on each ext. IRQ;

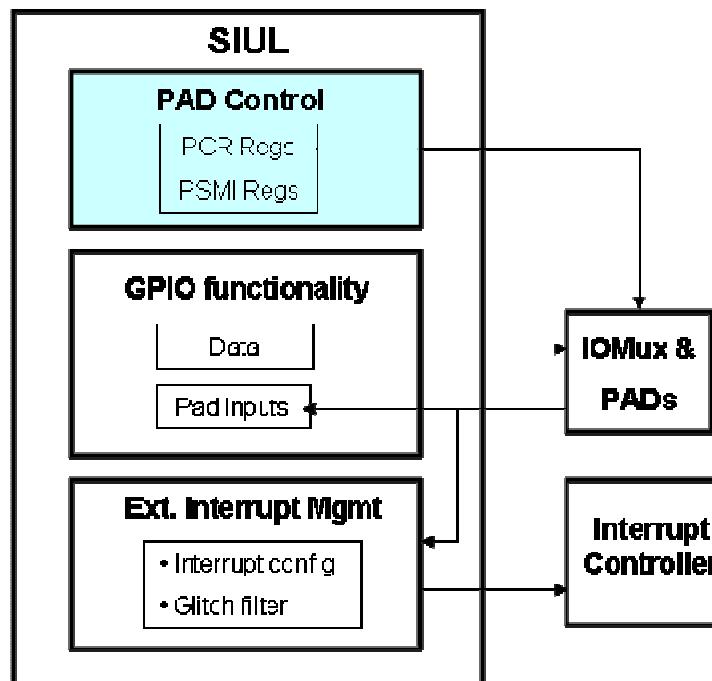
► MCU Identification

- Recognition of the specific MCU and version



SIUL Pad Control and IOMUX configuration overview

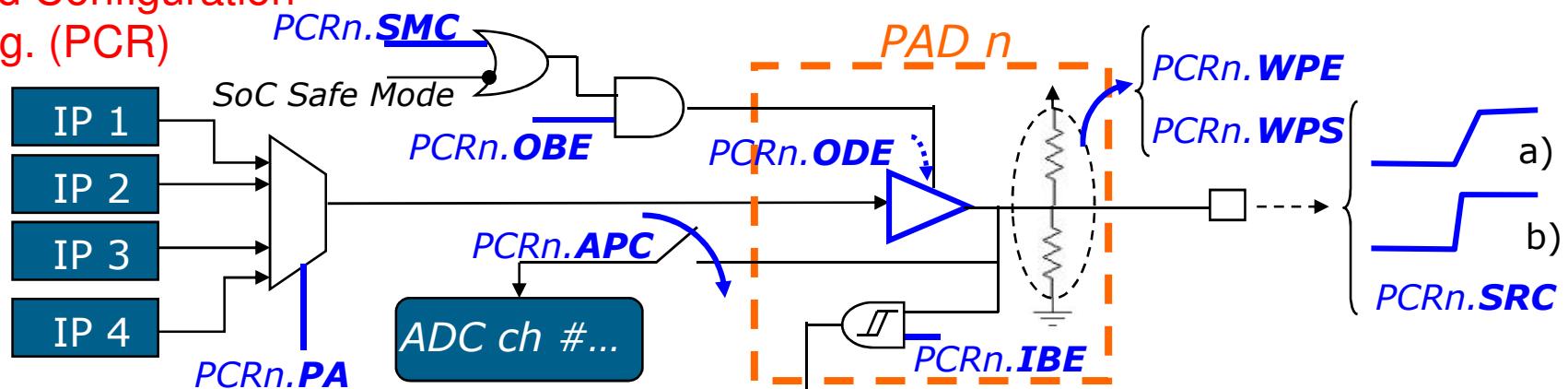
- ▶ Pad Control is managed through Pad Configuration Registers (PCRs)
- ▶ IOMUX configuration is managed through:
 - PCR Registers (output functionalities)
 - PSMI Registers (input functionalities)



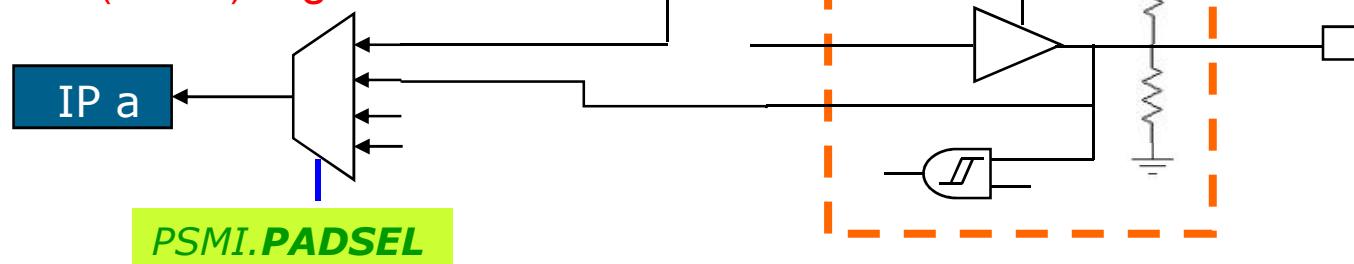
SIUL Pad Control and IOMUX config

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R		SMC	APC		PA	OBE	IBE			ODE		SRC	WPE	WPS		
W	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	

Pad Configuration Reg. (PCR)



Pad Select Multiplexed Inputs (PSMI) reg.



Freescale, the Freescale logo, Altivec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.



Peripherals

SIUL Pad Control and IOMUX configuration 2/4

Alternate functions are chosen by
PCR.PA bitfields:

PCR.PA = 00 -> AF0;
PCR.PA = 01 -> AF1;
PCR.PA = 10 -> AF2;
PCR.PA = 11-> AF3.

This is intended to select the output
functions;

For input functions,
PCR.IBE bit must be written to '1',
regardless of the values selected in
PCR.PA bitfields.

For this reason, the value corresponding
to an input only function is reported as “--”.

2.7 Functional ports A, B, C, D, E, F, G, H

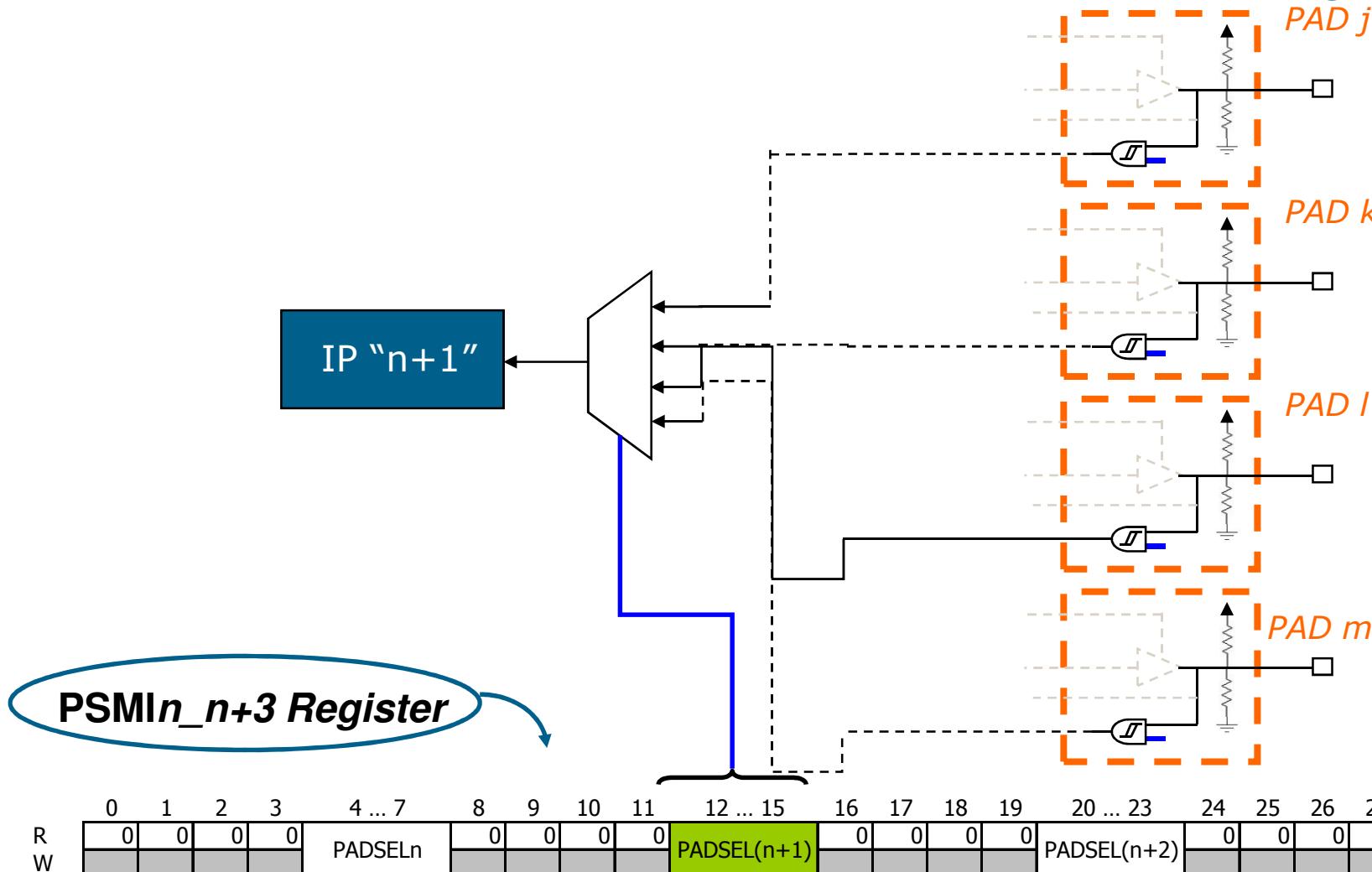
The functional port pins are listed in [Table 2-3](#).

Table 2-3. Functional Port Pin Descriptions

Port pin	PCR register	Alternat e function ¹	Function	Peripheral	I/O direction	Pad type	RESET config.	Pin No.			
								100 LQFP	144 LQFP	176 LQFP	208 MAP BGA ²
PA[0]	PCR[0]	AF0 AF1 AF2 AF3 --	GPIO[0] E0UC[0] CLKOUT E0UC[13] WKUP[19] ³	SIUL eMIOS 0 CGM eMIOS 0 WKPU	I/O I/O O I/O I	M	Tristat e	12	16	24	G4
PA[1]	PCR[1]	AF0 AF1 AF2 AF3 -- --	GPIO[1] E0UC[1] - - WKUP[2] ⁽³⁾ NMI ⁴	SIUL eMIOS 0 - - WKPU WKPU	I/O I/O - - I I	S	Tristat e	7	11	19	F3
PA[2]	PCR[2]	AF0 AF1 AF2 AF3 --	GPIO[2] E0UC[2] - - WKUP[3] ⁽³⁾	SIUL eMIOS 0 - - WKPU	I/O I/O - - I	S	Tristat e	5	9	17	F2
PA[3]	PCR[3]	AF0 AF1 AF2 AF3 --	GPIO[3] E0UC[3] LIN5TX - EIRQ[0]	SIUL eMIOS 0 LINFlex 5 - SIUL	I/O I/O O - I	S	Tristat e	68	90	114	K15

Peripherals

PSMI SIUL Pad Control and IOMux configuration 3/4



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.



Peripherals

SIUL Pad Control and IOMUX configuration 4/4

	0	1	2	3	4 ... 7	8	9	10	11	12 ... 15	16	17	18	19	20 ... 23	24	25	26	27	28 ... 31
R W	0	0	0	0	PADSELn	0	0	0	0	PADSEL(n+1)	0	0	0	0	PADSEL(n+2)	0	0	0	0	PADSEL(n+3)

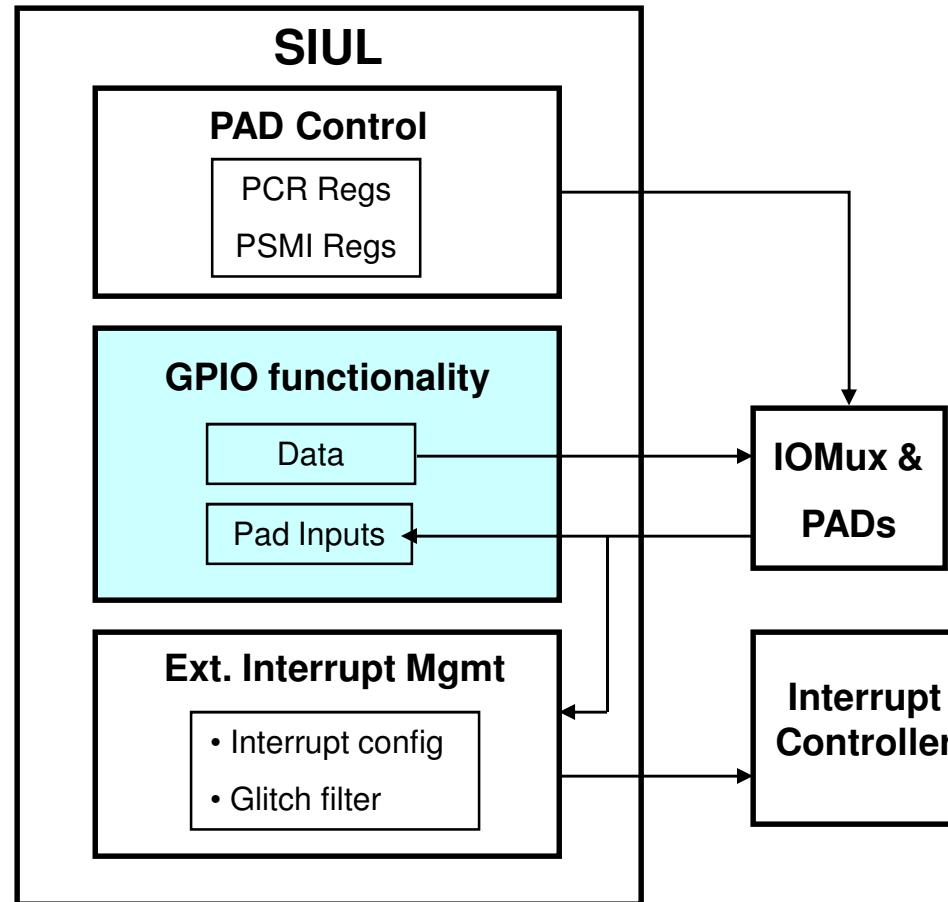
- ▶ Different pads can be chosen as possible inputs for a certain peripheral function.

Table 15-13. Peripheral input pin selection

PSMI registers	PADSEL fields	SIUL address offset	Function / Peripheral	Mapping ¹
PSMIO_3	PADSEL0	0x500	CAN1RX / FlexCAN 1	00: PCR[35] 01: PCR[43] 10: PCR[95] ²
	PADSEL1	0x501	CAN2RX / FlexCAN 2	0: PCR[73] 1: PCR[89] ²
	PADSEL2	0x502	CAN3RX / FlexCAN 3	00: PCR[36] 01: PCR[73] 10: PCR[89] ²
	PADSEL3	0x503	CAN4RX / FlexCAN 4	00: PCR[35] 01: PCR[43] 10: PCR[95] ²

Peripherals

SIUL GPIO Functionality



Peripherals

SIUL GPIO Functionality 1/2

► SIUL manages GPIO pads both:

- On individual base (R/W access to a single GPIO);
 - Access is done on a byte basis
- On port base (parallel access).

► Ports accesses can be:

- Data Read: 32-bit, 16-bit or 8-bit accesses
- Data Write: 32-bit, 16-bit or 8-bit (only if not masked access)

▪ **Masked Access:**

This mechanism allows support for port accesses or for bit manipulation **without** the need to use **read-modify-write** operations

Peripherals

SIUL GPIO Functionality 2/2

- ▶ All the pads have separate data inputs and data output registers:
 - The data **output registers** support both **read** and **write** operations to be performed.
 - The data **input registers** support **read** access only.
- ▶ When the pad is configured to use one of its alternate functions:
 - the **data input value reflects** the respective value of the pad (Input value to the associated IP from “outside world”).
 - If a write operation is performed to the data output register, it will not be reflected by the pad value until reconfigured to GPIO.

Peripherals

Register Description: Masked access to ports

- ▶ This mechanism allows port accesses or for bit manipulation without the need to use read-modify-write operations

▶ Mask Field:

- Each bit corresponds to one data bit in the MPPDO[x] register at the same bit location.

▶ Masked Parallel Pad Data Out:

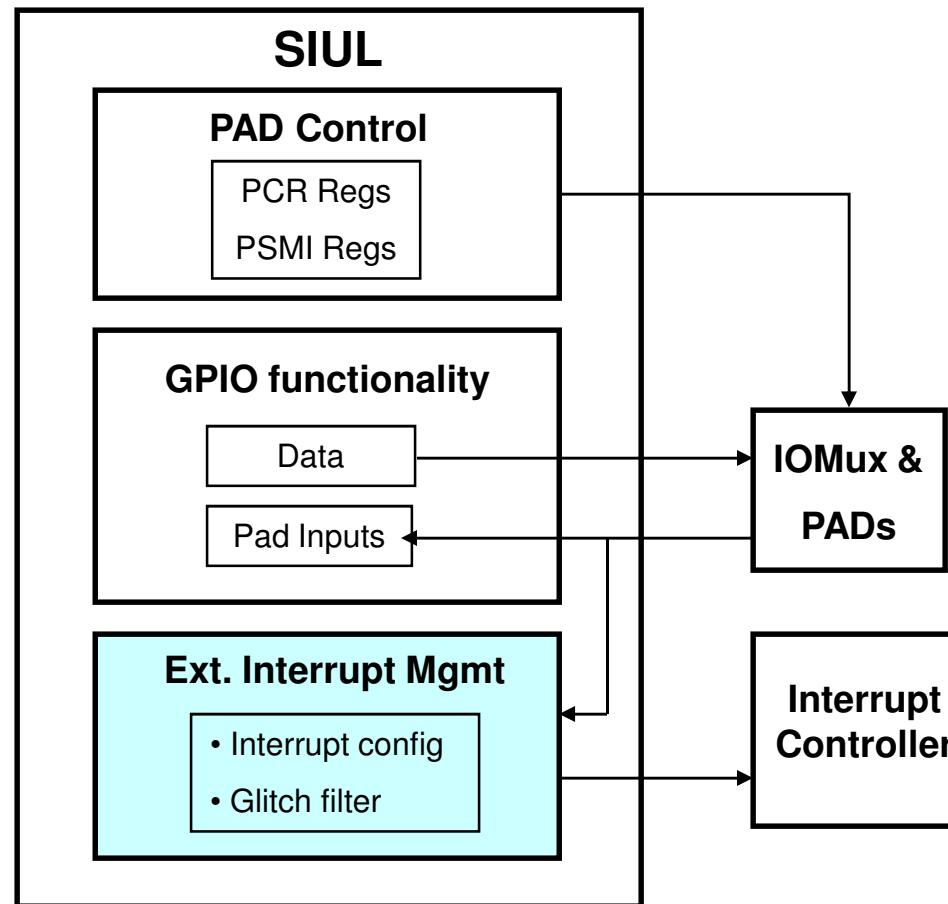
- The x and bit index define which MPPDO register bit is equivalent to which PDO register bit according to the following equation:
- $MPPDO[x][y] = PDO[(x*16)+y]$

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

- ▶ The MPGPDO[x] register may **only** be accessed with 32 bit writes. 8-bit or 16-bit writes will not modify any bits in the register and cause a transfer error response by the module. Read accesses will return 0.

Peripherals

SIUL External Interrupt Management



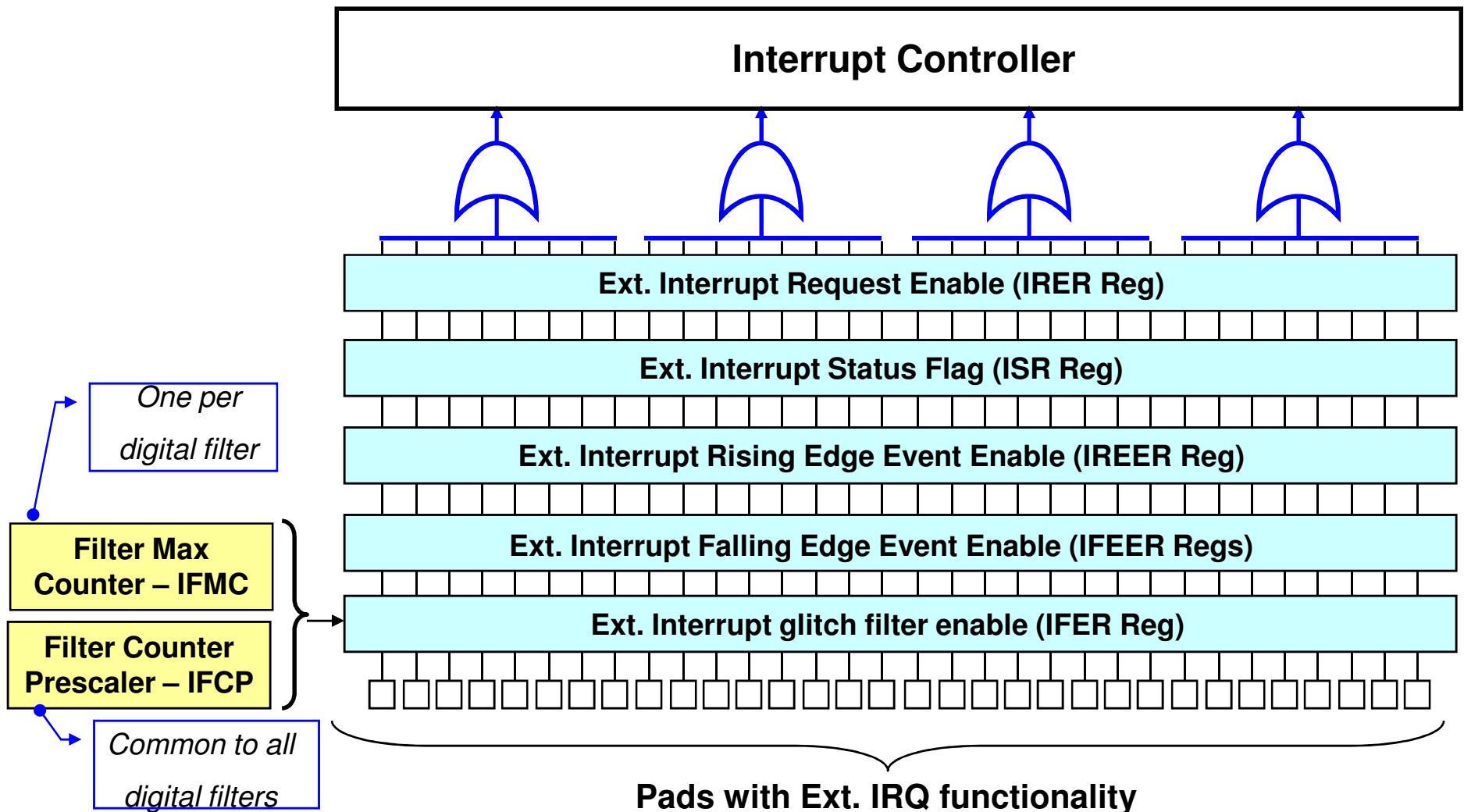
Peripherals

External IRQ Management: Feature set

- ▶ SIUL supports 4 interrupt vectors for 32 interrupt input signals
- ▶ 8 external interrupt sources can be grouped on a single interrupt line.
 - All of the external interrupt pads within a single group have equal priority.
- ▶ Each external interrupt line:
 - Can be enabled or disabled independently
 - Independent interrupt mask
 - Edge detection (rising, falling, both)
 - Each external interrupt supports an individual flag (clear-by-write-1)
 - have a digital glitch filter applied to it.
 - The glitch filters need a running internal oscillator clock to work. If such clock is not available, external interrupts will be effectively disabled when the glitch filter is enabled

SIUL External Interrupt Management

External IRQ Management: Block Scheme



MCU Identification Registers (MDIR)

- ▶ SIUL includes two registers that can be read by users and tools manufacturers to determine **what device is present** (which part number, pkg, flash size etc.) and to take the corresponding actions.
- ▶ - Debuggers use these registers to identify the device and its revision.

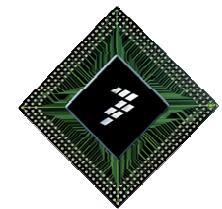
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R																	CSP			PKG		0	0									
W																																
MIDR1																																

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	S/F	Flash_Size_1	Flash_Size_2	0	0	0	0	0	0	0	0	0	0	0	0																	
W																																
Part_No																																



Designing with Freescale

Timed I/O (Watchdog, PIT, STM & eMIOS)



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**™
semiconductor

Periodic Interrupt Timer (PIT)

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.



► **Periodic Interrupt Timer (PIT) features:**

- ▶ 32-bit counter resolution
- ▶ clocked by system clock
- ▶ timer can generate interrupt (DMA trigger on MPC5601/2/5/6/7B)
- ▶ independent timeout periods for each timer
- ▶ Channel outputs can trigger ADC conversion
- ▶ MPC5602/3/4 -> 6 PITs
- ▶ MPC5605/6/7 -> 8 PITs

Peripherals

Bolero MPC5602/3/4B PIT Timer Module

PIT Nr.	Interrupt	Peripheral Trigger	DMA Trigger
0	YES	NO	YES
1	YES	NO	YES
2	YES	10-bit ADC	NO
3	YES	CTU ch28	NO
4	YES	NO	YES
5	YES	NO	YES

STM System Timer Module features

- The System Timer Module (STM) is a 32-bit timer designed to support commonly required system and application software timing functions.

Main features:

- One 32-bit modulus up counter with 8-bit prescaler (1-256)
- Four 32-bit compare channels, each with its own interrupt vector
- Compliant with the AUTOSAR Task Monitor scheme
- Counter can be stopped in debug mode

Enhanced Modular I/O System (eMIOS)

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

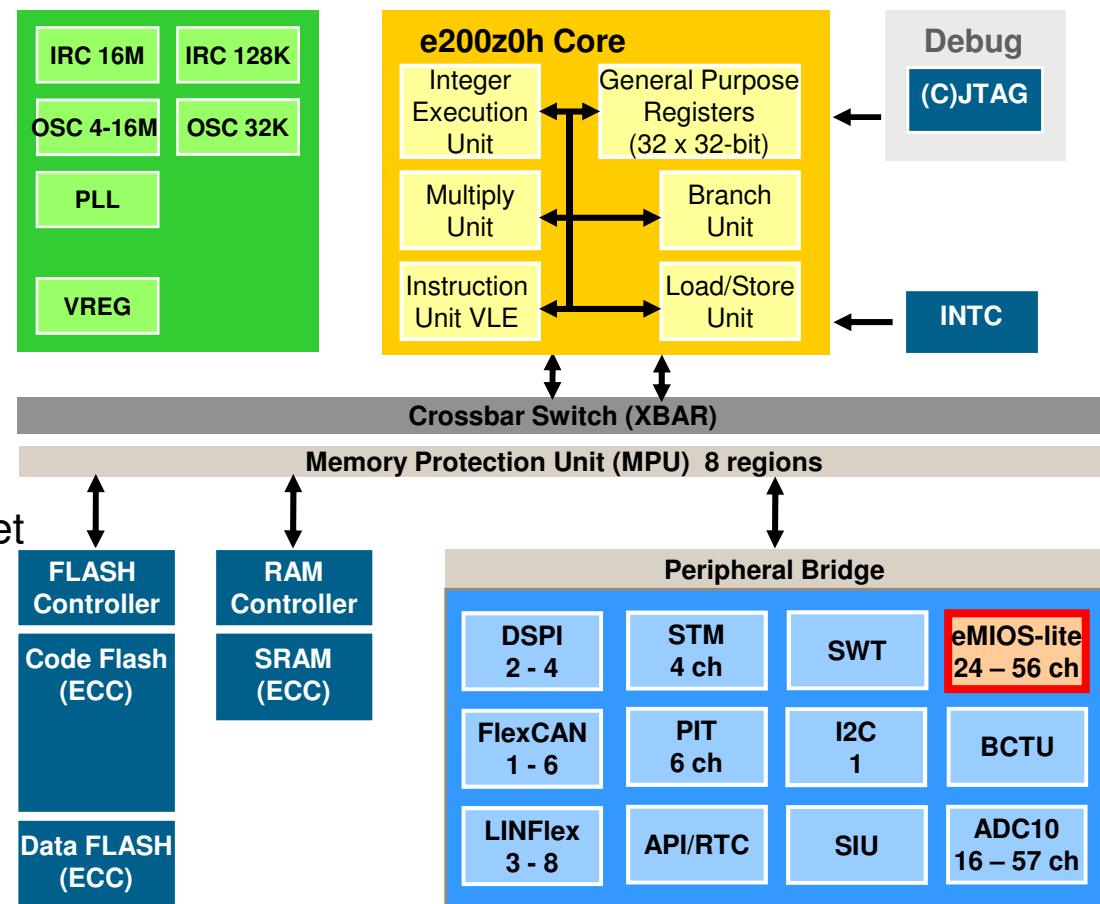


eMIOS260

Introduction

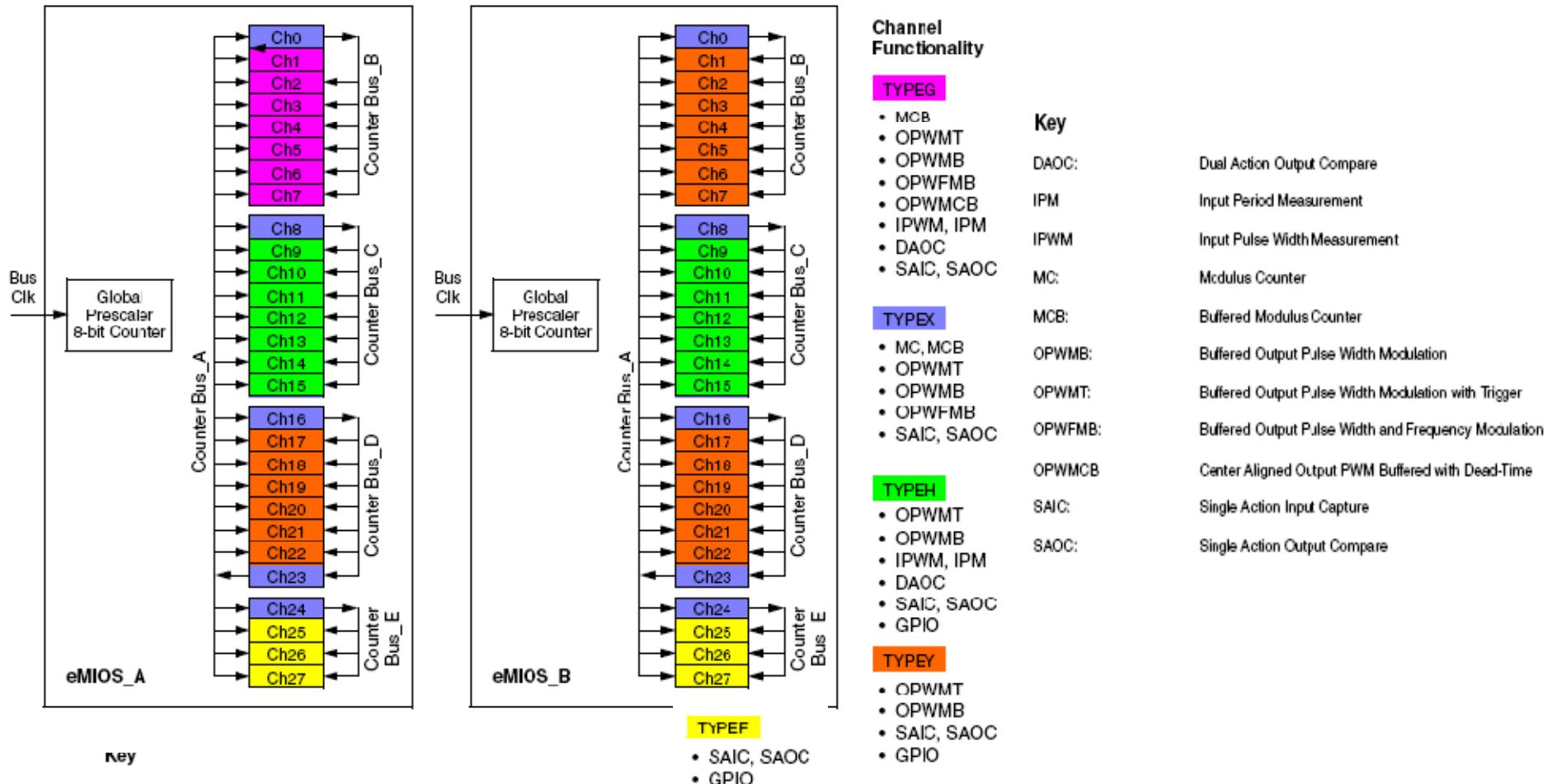
- The eMIOS260 used on BOLERO family is based on the existing eMIOS used on 5500 power architecture parts

- Provides various modes to generate or measure timed event signals.
- 24 to 56 Channels based on 3 channel types
- One new channel mode featuring lighting applications, OPWMT
- All other channel modes are subset of the unified channel structure on previous eMIOS.
- Consistent user interface with previous eMIOS implementation.



Peripherals

EMIOS channel configuration MPC5604B

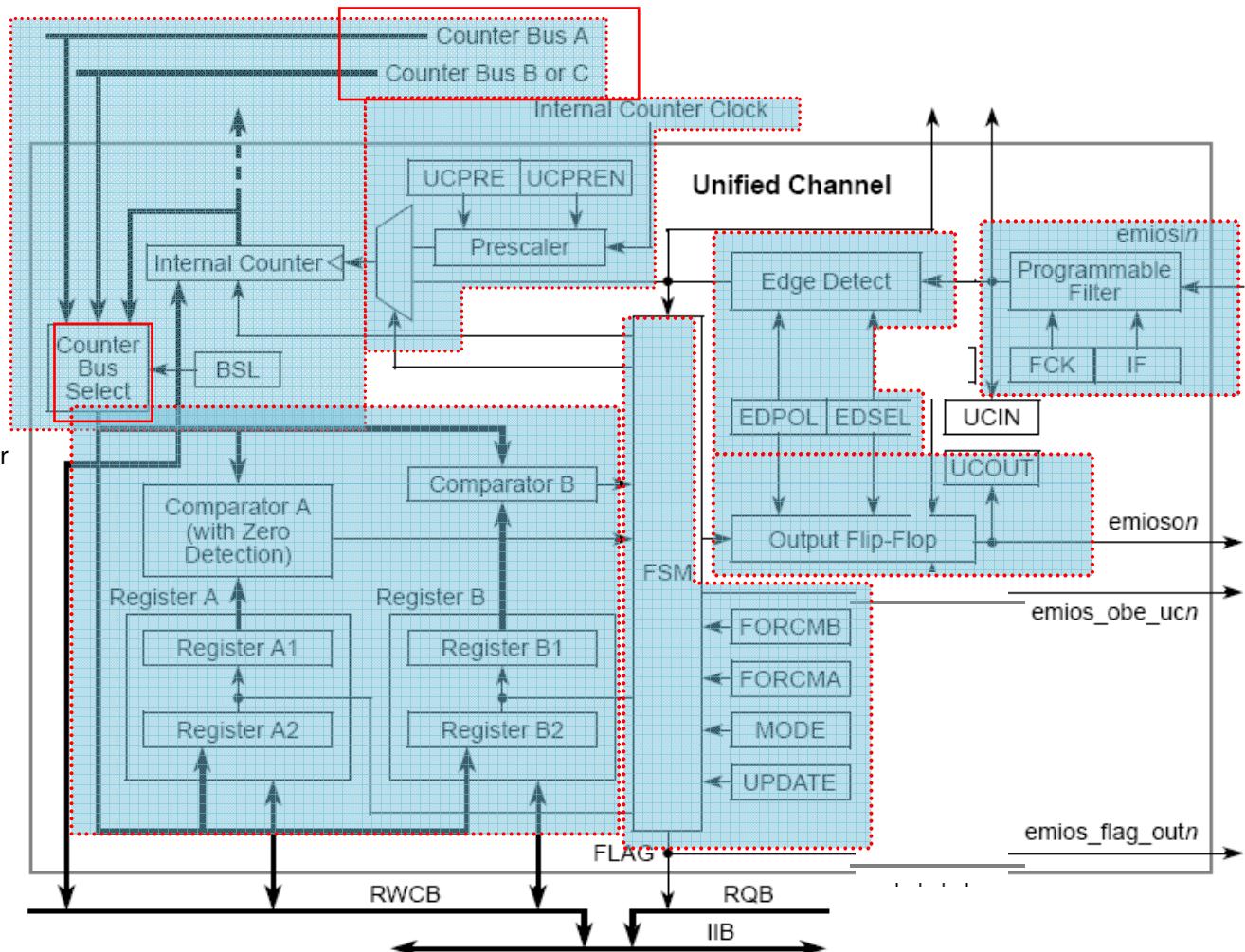


Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

eMIOS260

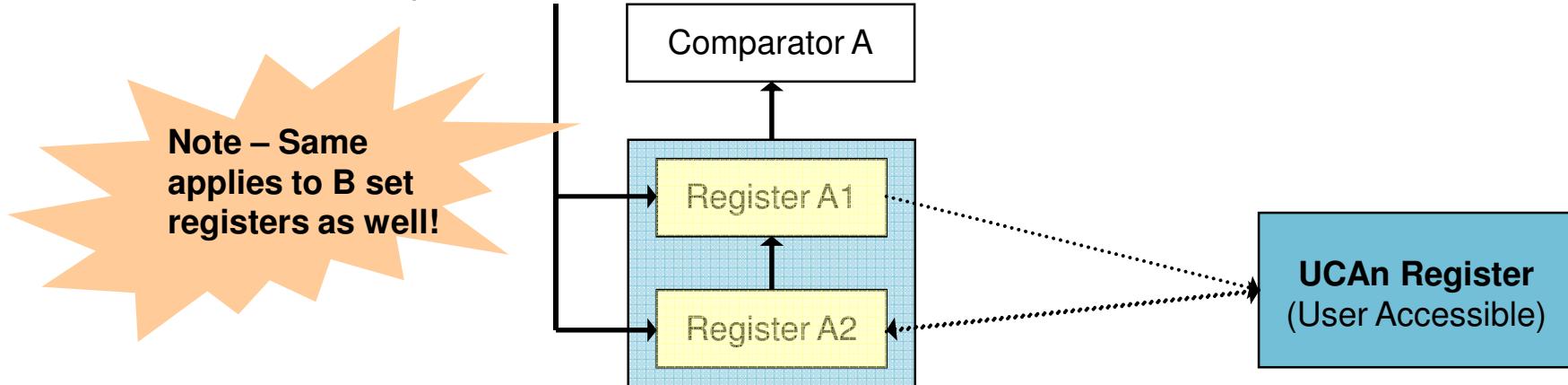
Unified Channel Features

- Selectable time base for each counter bus
- Programmable Clock Prescaler
- Double buffered data registers and comparators
- State Machine (with mode control)
- Programmable as input or output:
 - Input Edge Detect as rising, falling or toggle.
- Programmable digital input filter



eMIOS260 Double Buffered A and B Registers

- ▶ A and B data registers are double buffered to provide a mechanism for safe update of the A and B register values
 - This also enables very small pulse / period generation or measurement since updates can happen in current period
- ▶ The channel A user registers are an address mapped link to either the A1 or A2 register (determined automatically by the mode of the unified channel).
 - For output modes, data is typically written to the A2 register
 - For input capture modes, data is latched into either A1 or A2 depending on mode.
 - All of this is transparent to the user

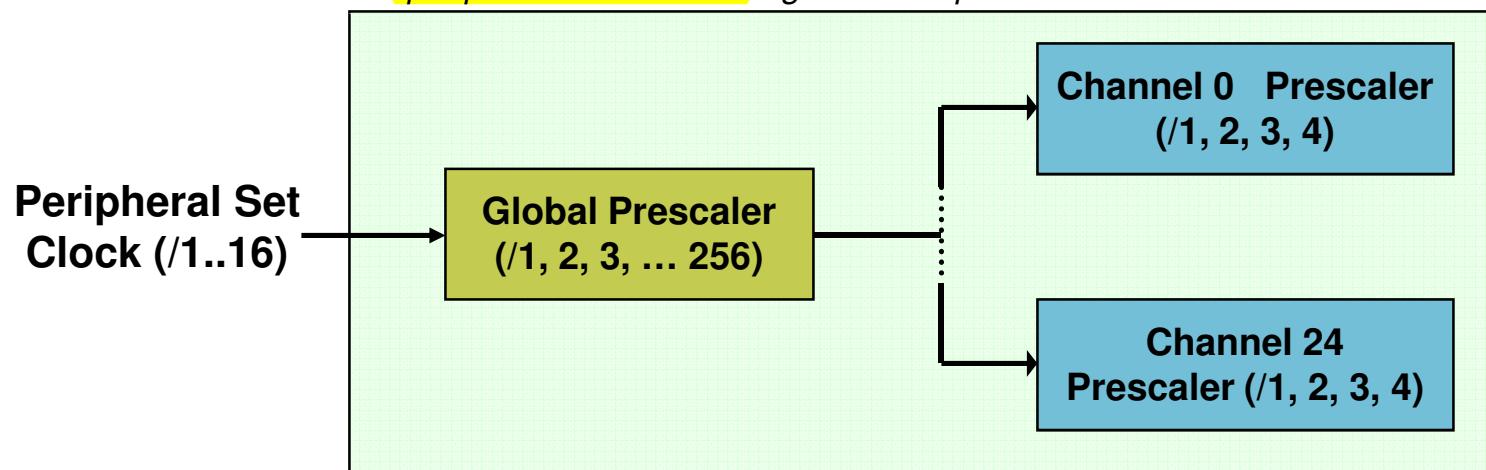


Peripherals

EMIOS Channel Types MPC5604B

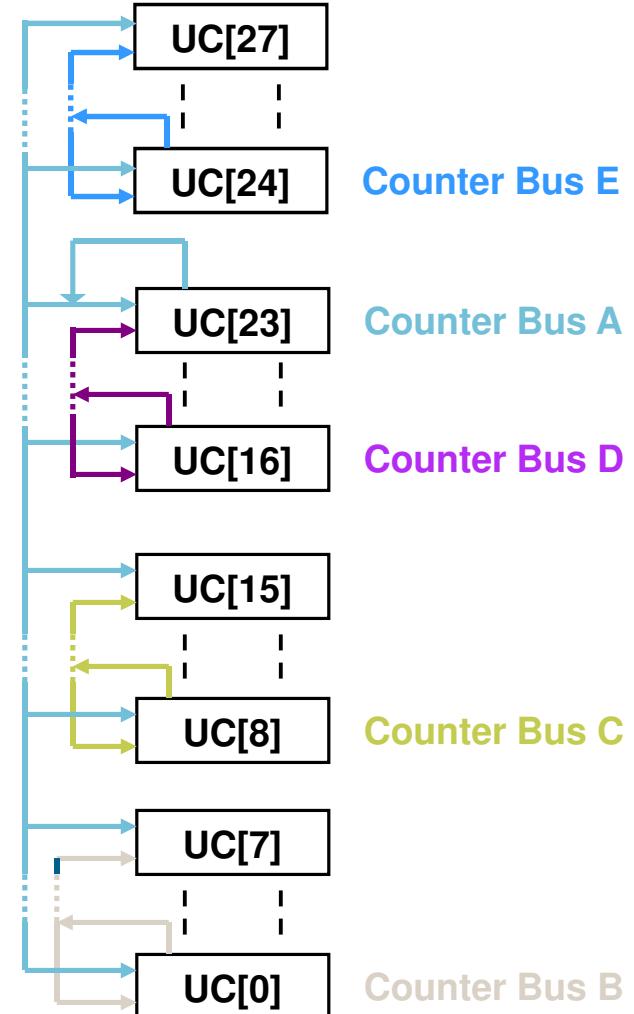
Description	Name	Channel Type				
		Type X	Type Y	Type F	Type G	Type H
General Purpose Input / Output	GPIO	X	X	X	X	X
Single Action Input Capture	SAIC	X	X	X	X	X
Single Action Output Compare	SAOC	X	X	X	X	X
Modulus Counter	MC	X	-	-	-	-
Modulus Counter Buffered (Up / Down)	MCB	X	-	-	X	-
Input Pulse Width Measurement	IPWM	-	-	-	X	X
Input Period Measurement	IPM	-	-	-	X	X
Double Action Output Compare	DAOC	-	-	-	X	X
Output Pulse Width and Frequency Modulation Buffered	OPWFMB	X	-	-	X	-
Center aligned Output PWM Buffered with dead time	OPWMCB	-	-	-	X	-
Output Pulse Width Modulation Buffered	OPWMB	X	X	-	X	X
Output Pulse Width Modulation Trigger	OPWMT	X	X	-	X	X

- ▶ Channel clocks are derived from the **peripheral set clock**
- ▶ There are global and channel prescalers for flexibility
 - Note that both prescalers must be enabled!!!
 - **Caution** – Only **eMIOS0** channels 0-8, 16, 23, 24 and eMIOS1 channels 0, 8, 16, 23, 24 have an **internal counter**
- ▶ *The eMIOS is clocked from the “peripheral set 3 clock” signal which provides clock dividers from 1 to 16*



Peripherals EMIOS counter buses

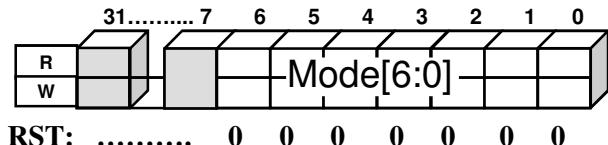
- The BOLERO family has 5 shared counter busses allowing common counter bus timing across multiple channels
 - **Counter Bus A** is shared with all channels and driven from channel 23
 - **Counter bus B** is shared with channels 0 to 7 and driven from channel 0
 - **Counter bus C** is shared with channels 8 to 15 and driven from channel 8
 - **Counter bus D** is shared with channels 16 to 23 and driven from channel 16
 - **Counter bus E** is shared with channels 24 to 27 and driven from channel 24



Peripherals

EMIOS mode selection

EMIOS_CCR[n] Mode Fields



Note: This is only an example.

Implemented modes may be different.

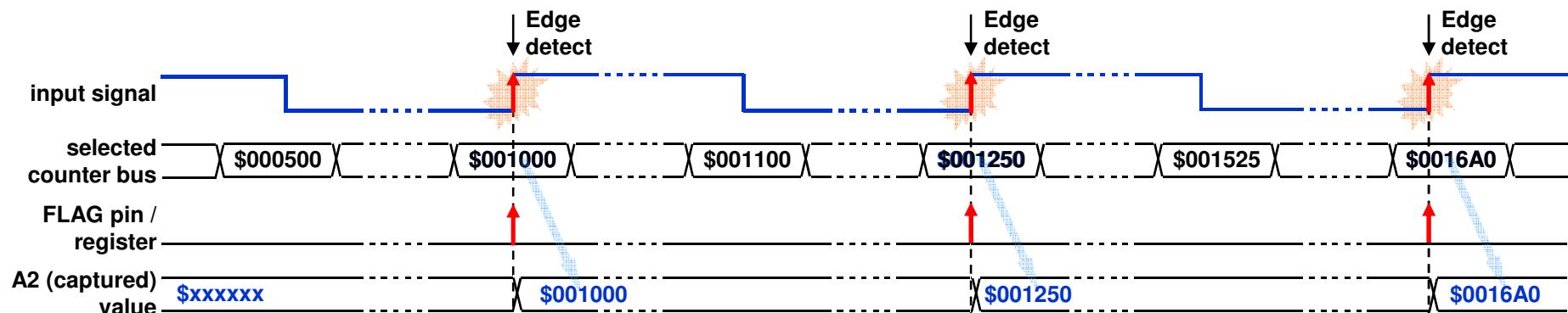
MODE[0:6]	Mode of operation
000_0000	General Purpose Input/Output (input)
000_0001	General Purpose Input/Output (output)
000_0010	Single Action Input Capture
000_0011	Single Action Output Compare
000_0100	Input Pulse width Measurement
000_0101	Input Period Measurement
000_011b	Double Action Output compare
000_1000	
000_1111	Reserved
001_0bbb	Modulus Counter
001_1000	
010_0101	Reserved
010_0110	Output Pulse Width Modulation with Trigger
010_0111	
100_1111	Reserved
101_000b	Modulus Counter Buffered (Up counter)
101_0010	
101_0011	Reserved
101_010b	Modulus Counter Buffered (Up/Down counter)
101_0110	
101_1001	Reserved
101_10b0	Output Pulse Width and Frequency Modulation Buffered
101_11bb	Center Aligned Output Pulse Width Modulation Buffered
110_00b0	Output Pulse Width Modulation Buffered
110_0100	
111_1111	Reserved

Peripherals

EMIOS - Single Action Input Capture

Returns the **value** of the counter bus on an edge match of an input signal .

- Can use Internal or Modulus counter
- Can match on Rising, Falling or Toggle determined by state of EDPOL, EDSEL



Notes:

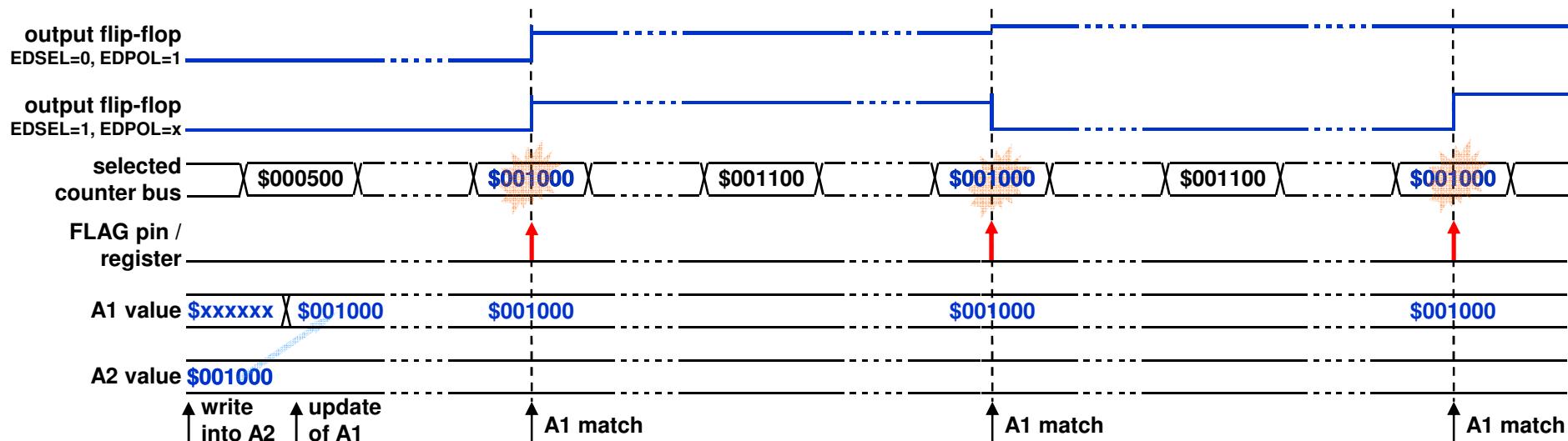
- When edge is detected, flag is set and counter bus value is captured in register A2. User reads this value from UCA[n] register.
- UCB[n] = Cleared and cannot be written

Peripherals

EMIOS - Single Action Output Compare

Generates an output on a counter bus match

- Can use Internal or Modulus counter
- Can set output to go HIGH, LOW or TOGGLE, based on the state of EDPOL and EDSEL



Notes:

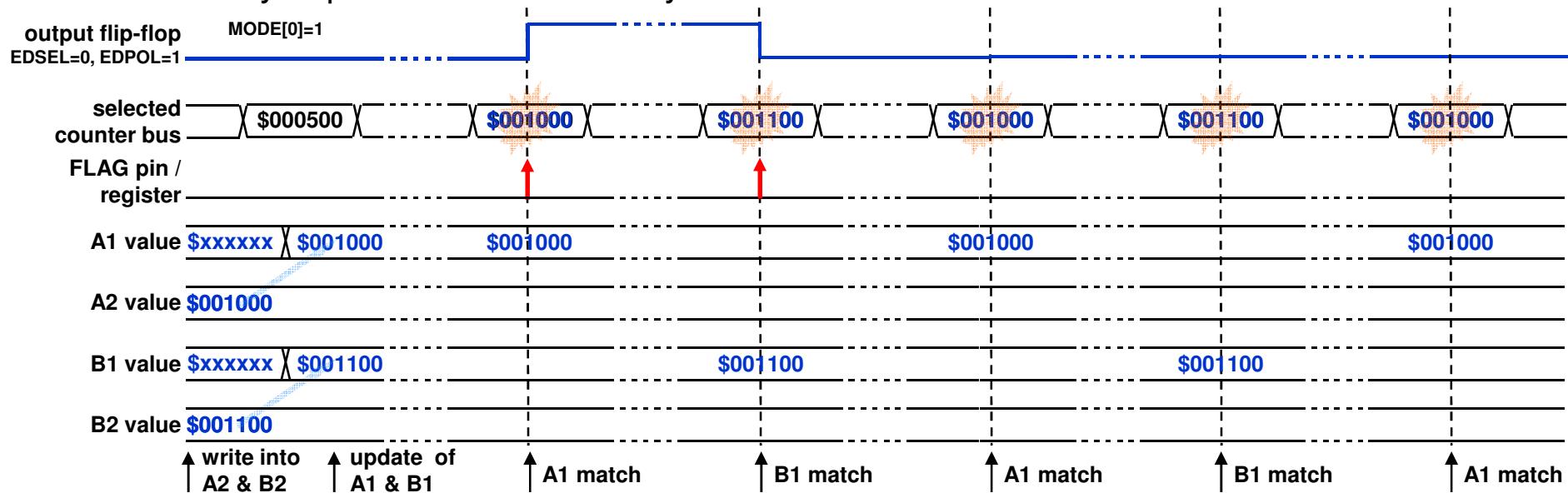
- Write the desired counter bus value to create a match into UCA[n] (A2n) which is buffered into A1.
- A comparator match of A1 results in an output event, defined by status of EDPOL and EDSEL

Peripherals

EMIOS - Double Action Output Compare

Generates an output pulse

- Can use Internal or Modulus counter
- Polarity of pulse is determined by value of EDPOL



Notes:

- Write the desired pulse leading edge into UCA[n] (A2n) and the falling edge into UCB[n] (B2n) which are buffered into A1 and B1.
- On a comparator A match, the output is set to the value of EDPOL. FLAG is set if MODE0=1
- On a comparator B match, the output is set to the inverse of EDPOL. FLAG is set

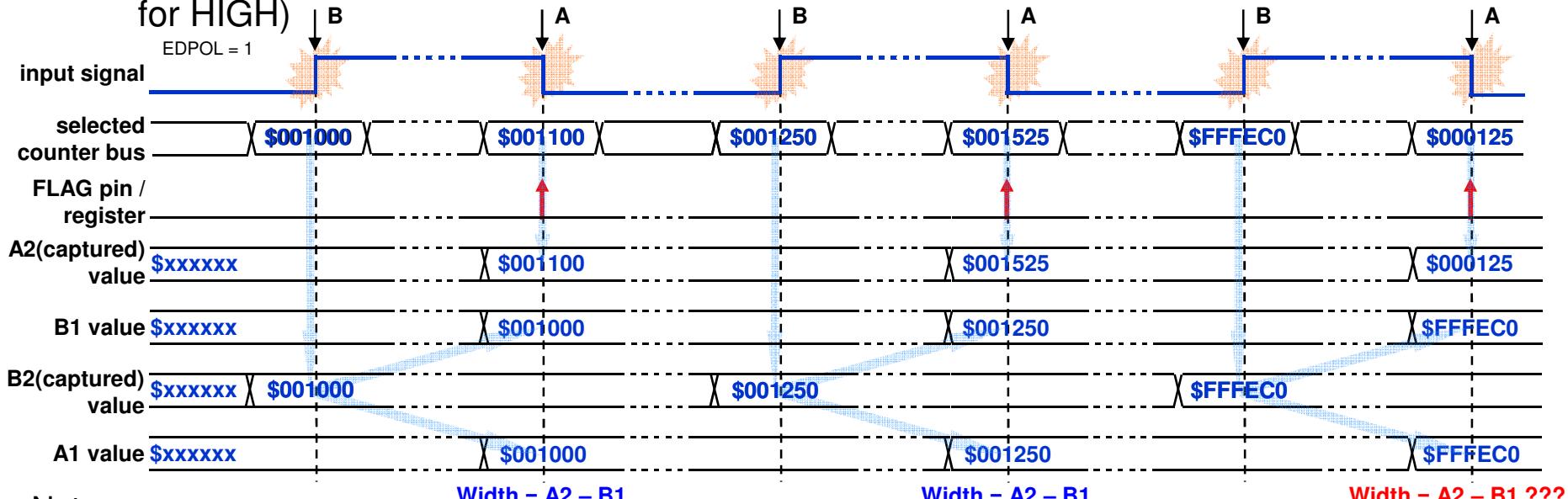


Peripherals

EMIOS - Input Pulse Width Measurement

Determines the **width** (in counter bus clock ticks) of an input pulse width

- Can use Internal or Modulus counter
- Can be configured to measure HIGH or LOW pulses by state of EDPOL bit (EDPOL=1 for HIGH)



Notes:

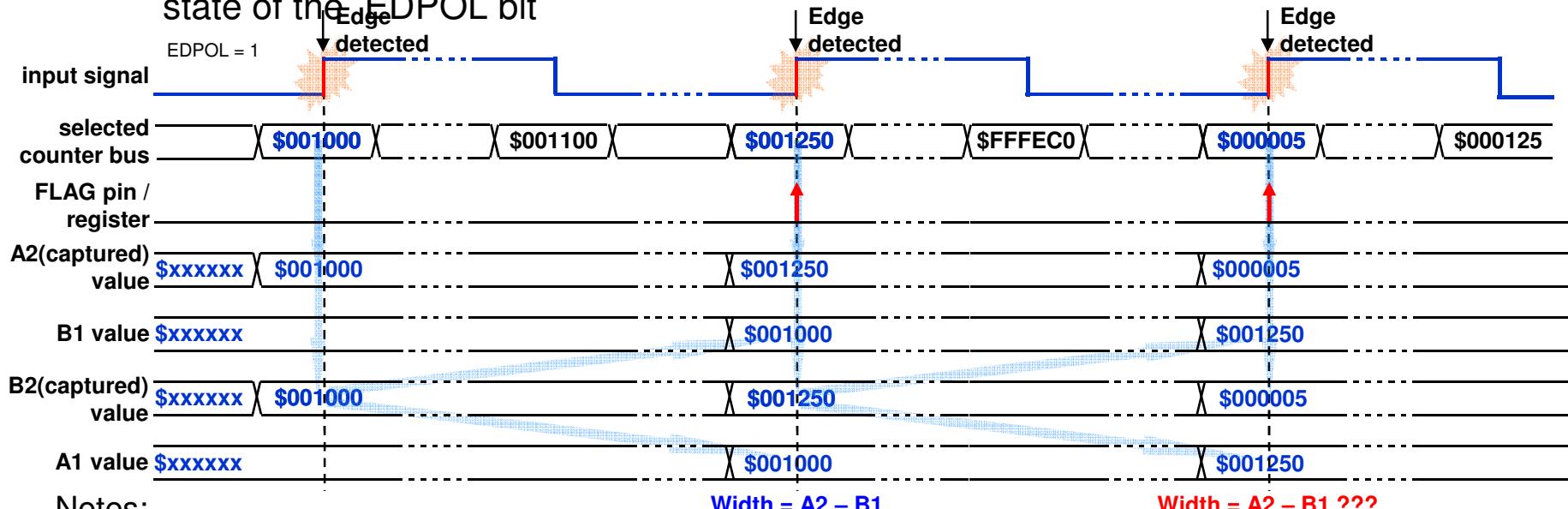
- Leading edge is captured into B2[n]. (EDPOL Determines if leading edge is high or low).
- Trailing edge is captured into A2[n] and Flag is set
- Pulse width is calculated by subtracting UCBn (B1) from UCAn (A2)
- **Caution** – If pulse has spanned a counter bus period, then need to take care to modify calculation....
Width = (UCAn + Counter Bus Period) - UCBn

Peripherals

EMIOS - Input Period Measurement

Determines the **period** (in counter bus clock ticks) of an input pulse width

- Can use Internal or Modulus counter
- Can be configured to measure between 2 HIGH or 2 LOW edges, determined by the state of the EDPOL bit



Notes:

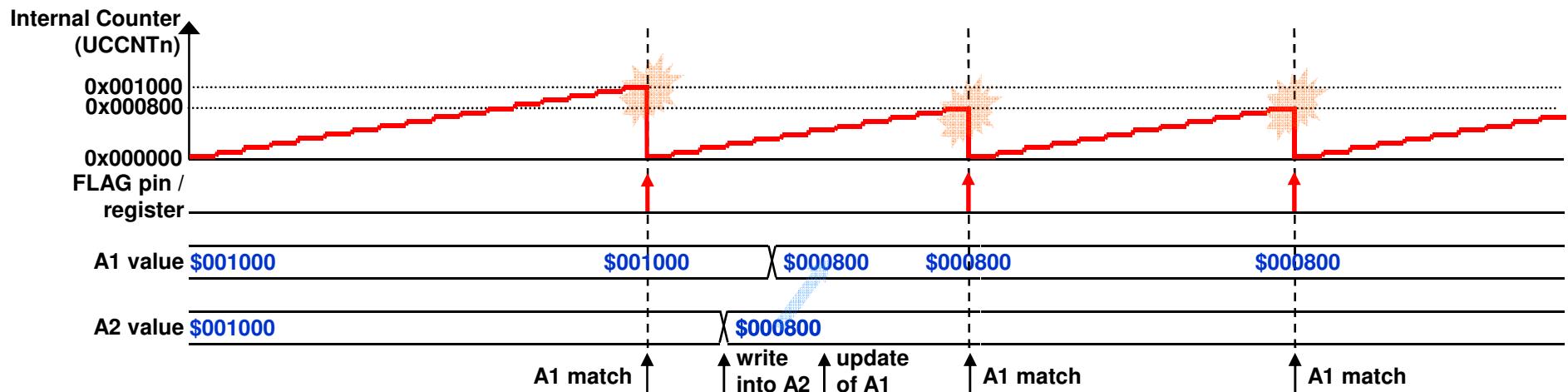
- When the edge of the selected polarity is detected, counter value is captured into A2[n] and B2[n], the data previously held in B2[n] is captured into A1[n] and B1[n], and Flag is set.
- Period is calculated by subtracting UCBn (B1) from UCAn (A2)
- **Caution** – If period of input signal has spanned a counter bus period, then need to take care to modify calculation.... Width = (UCAn + Counter Bus Period) - UCBn

Peripherals

EMIOS - Modulus Counter Mode – UP Counter

Generates a time base **which can be shared with other channels** through the **internal counter buses**

- Can use Internal or External (input channel pin) counter



Notes:

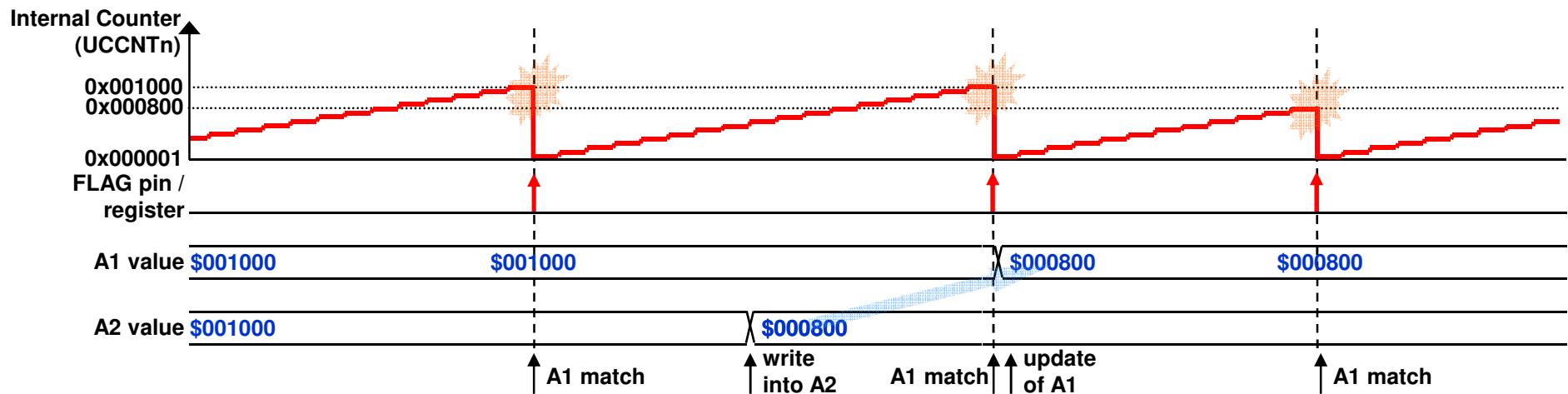
- On a comparator A match, FLAG is set and the internal counter is set to value \$0.
- A change of the A2 register makes the A1 register be updated at the next clock.
- Caution** – If when entering MC mode the internal counter value is upper than register UCA[n] value, then it will wrap at the maximum counter value (\$FFFFFF) before matching A1.

Peripherals

EMIOS - Modulus Counter Buffer Mode – UP Counter

Generates a time base which can be shared with other channels through the internal counter buses

- Can use Internal or External (input channel pin) counter



Notes:

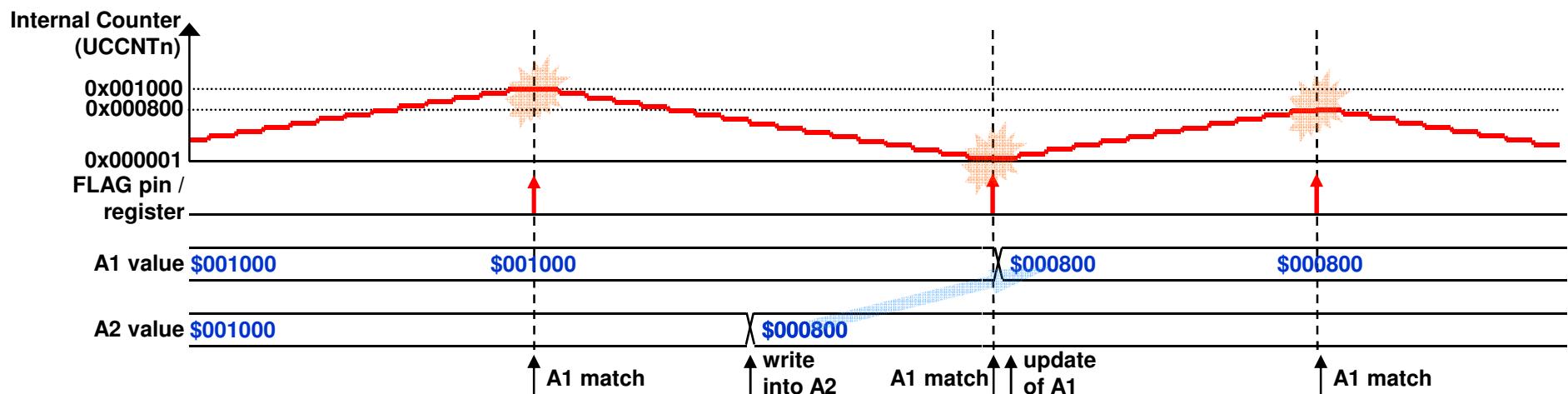
- On a comparator A match, FLAG is set and the internal counter is set to value \$1.
- Allowing smooth transitions, a change of the A2 register makes the A1 register be updated when the internal counter reaches the value \$1.
- Caution** – If when entering MCB mode the internal counter value is upper than register UCA[n] value, then it will wrap at the maximum counter value (\$FFFFFF) before matching A1.

Peripherals

EMIOS - Modulus Counter Buffer Mode – UP/DOWN Counter

Generates a time base which can be shared with other channels through the internal counter buses

- Can use Internal or External (input channel pin) counter

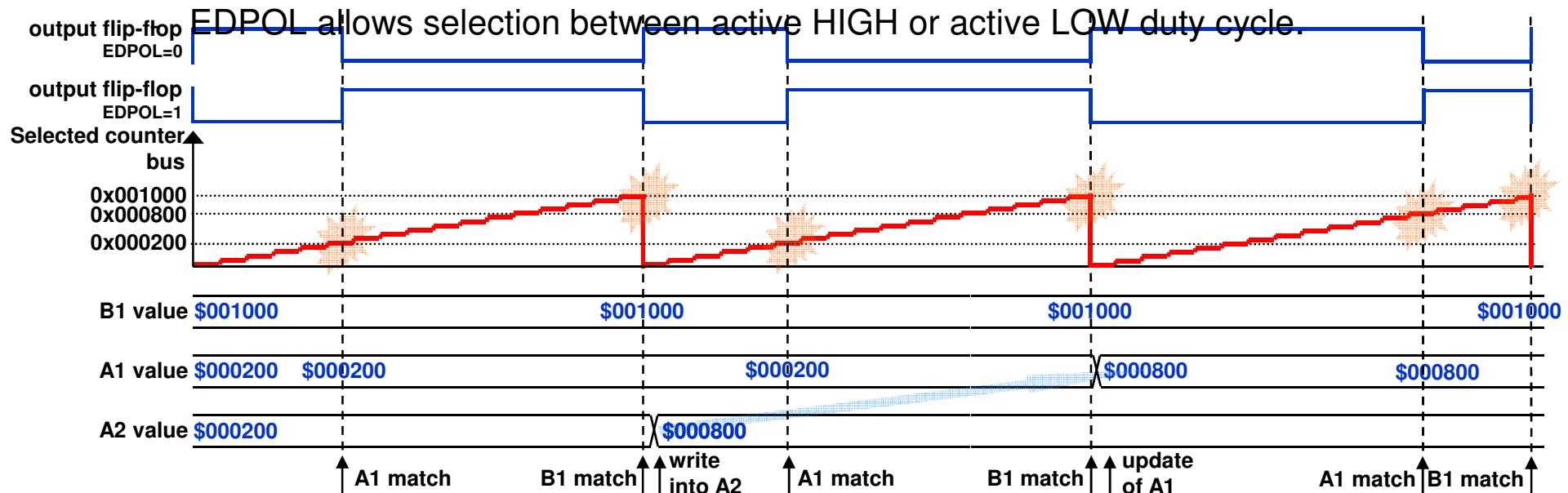


Notes:

- On a comparator A match, FLAG is set and the internal counter is set to value \$1.
- Allowing smooth transitions, a change of the A2 register makes the A1 register be updated when the internal counter reaches the value \$1.
- **Caution** – If when entering MCB mode the internal counter value is upper than register UCA[n] value, then it will wrap at the maximum counter value (\$FFFFFF) before matching A1.

Generates a simple output PWM signal

- Requires **INTERNAL** Counter



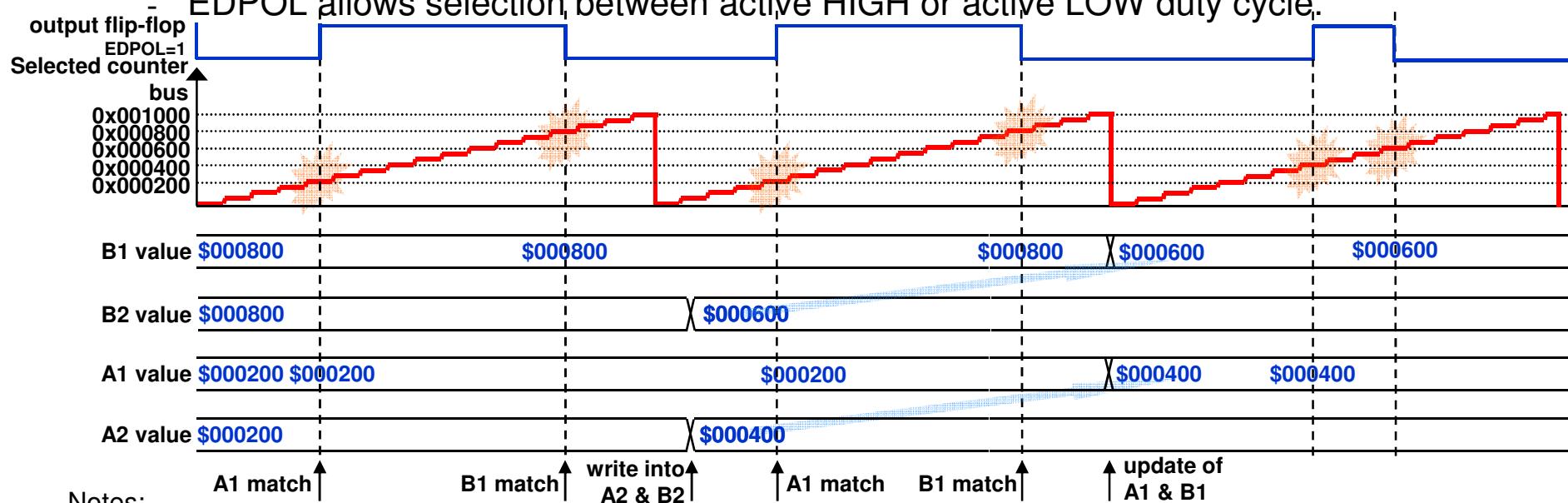
Notes:

- Duty Cycle = UCA[n] (A1) + 1, Period = UCB[n] (B1) + 1
- On Comparator A1 match, Output pin is set to value of EDPOL
- On Comparator B1 match, Output pin is set to complement of EDPOL and Internal counter is reset
- The transfers from register B2[n] to B1[n] and from register A2[n] to A1[n] are performed at the first clock of the next cycle.
- FLAGS can be generated only on B1 matches or on both A1 and B1 matches depending on MODE[5] bit.

Generates a simple output PWM signal

- Can use Internal or Modulus counter

- EDPOL allows selection between active HIGH or active LOW duty cycle.



Notes:

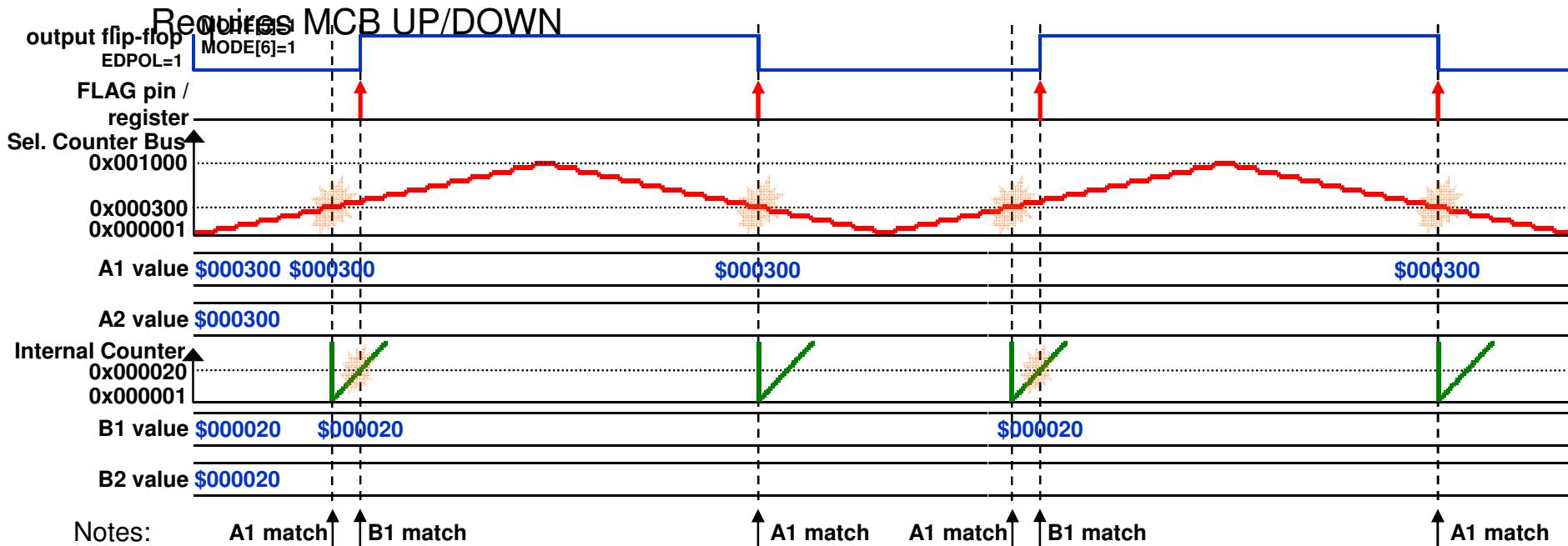
- Write UCA[n] (A1) with Leading Edge. Write UCB[n] (B1) with trailing edge
- On Comparator A1 match, Output pin is set to value of EDPOL
- On Comparator B1 match, Output pin is set to complement of EDPOL
- The transfers from register B2[n] to B1[n] and from register A2[n] to A1[n] are performed at the first clock of the next cycle.
- FLAGS can be generated only on B1 matches or on both A1 and B1 matches depending on MODE[5] bit.

Peripherals

EMIOS - OPWMCB

Generates a center aligned PWM output signal with dead time insertion

- MODE[6] bit selects between trailing and leading dead time insertion, respectively.
- EDPOL allows selection between active HIGH or active LOW duty cycle.



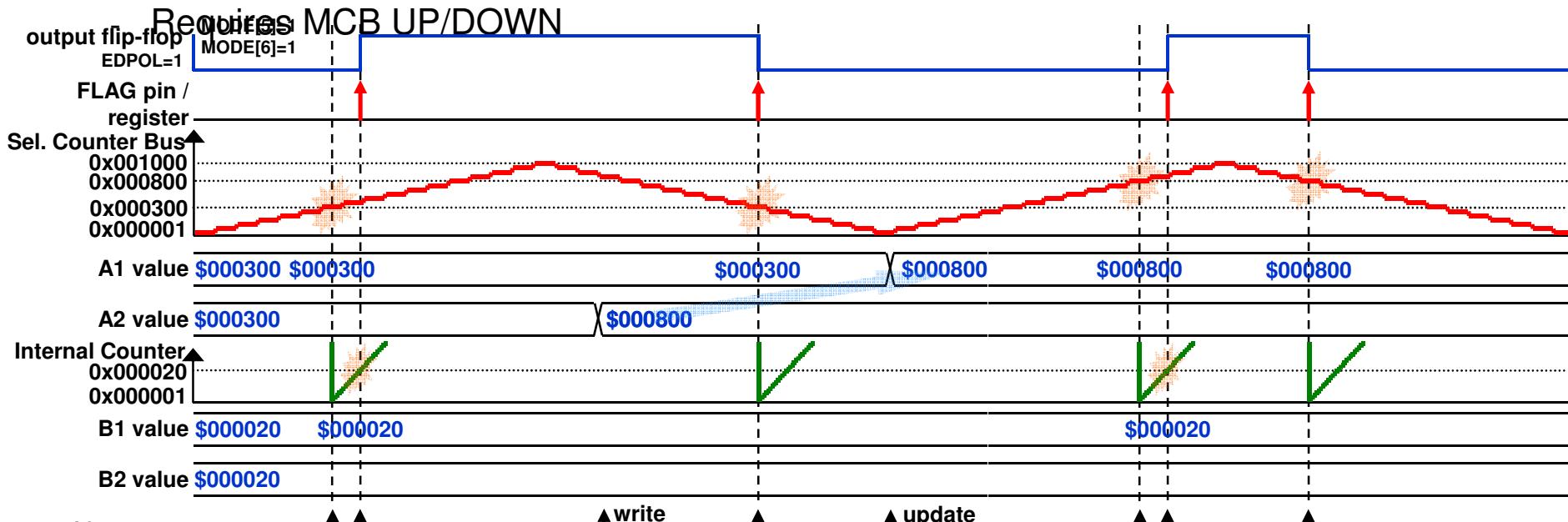
- Period = MCB Period, Dead Time = B1[n], Duty Cycle = $2 * (\text{Period} - \text{A1}[n]) - \text{Dead Time}$
- On the selected dead time insertion edge:
 - On Comparator A1 match (Selected Counter Bus), the internal counter is set to \$1
 - On Comparator B1 match (Internal Counter), Output pin is set to the value of EDPOL
- On the non-selected dead time insertion edge:
 - On Comparator A1 match, Output pin is set to complement of EDPOL and internal counter set to \$1

Peripherals

EMIOS - OPWMCB

Generates a center aligned PWM output signal with dead time insertion

- MODE[6] bit selects between trailing and leading dead time insertion, respectively.
- EDPOL allows selection between active HIGH or active LOW duty cycle.



Notes: A1 match B1 match write into A2 A1 match update of A1 A1 match B1 match A1 match

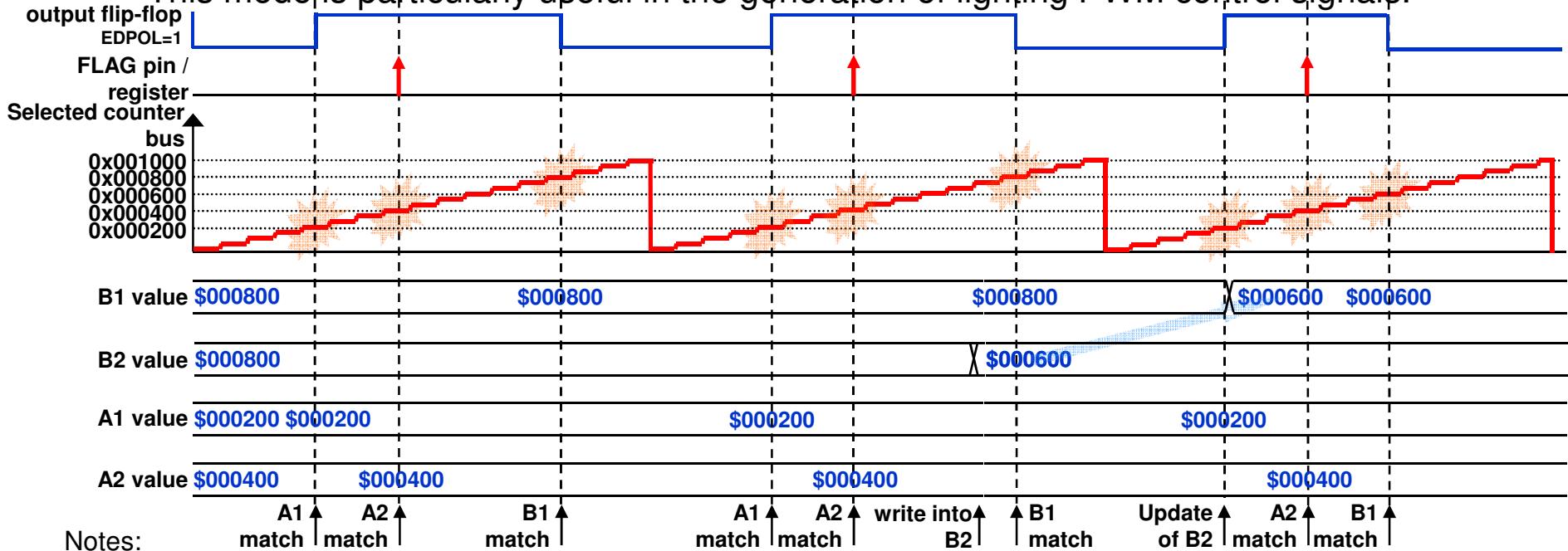
- FLAGS can be generated only on B1 matches or on both A1 and B1 matches depending on MODE[5] bit.
- The transfers from register B2[n] to B1[n] and from register A2[n] to A1[n] are performed at the first clock of the next cycle.
- To achieve 100% duty cycle, A1[n] must be set to \$1. To achieve 0% duty cycle, A1[n] must be set to a value greater than the maximum value of the selected time base.
- **Caution** – The internal counter should not reach \$0 as consequence of a rollover.

Peripherals

EMIOS - OPWMT

Generates a PWM signal with a fixed offset and a trigger signal

- Intended to be used with other channels in the same mode with shared common time base
- This mode is particularly useful in the generation of lighting PWM control signals.

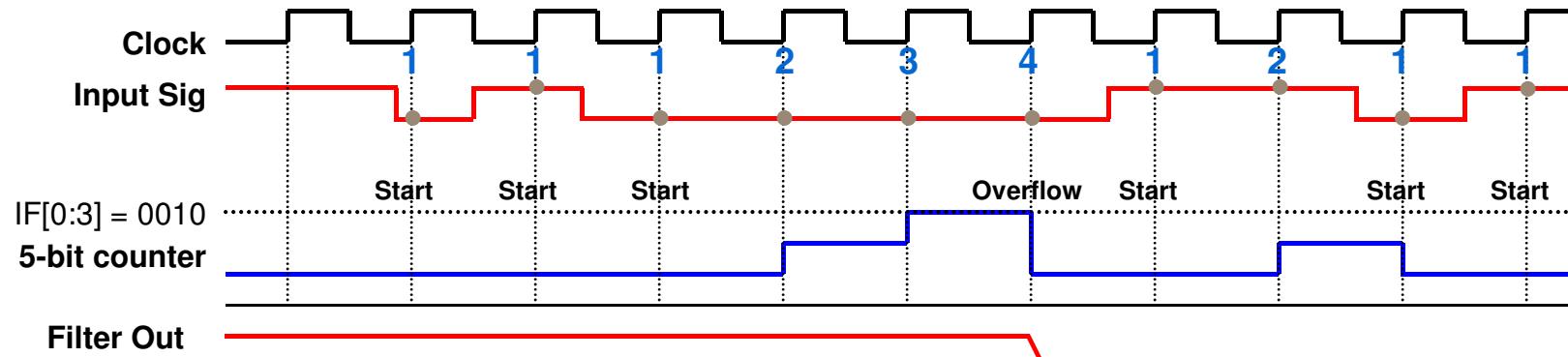
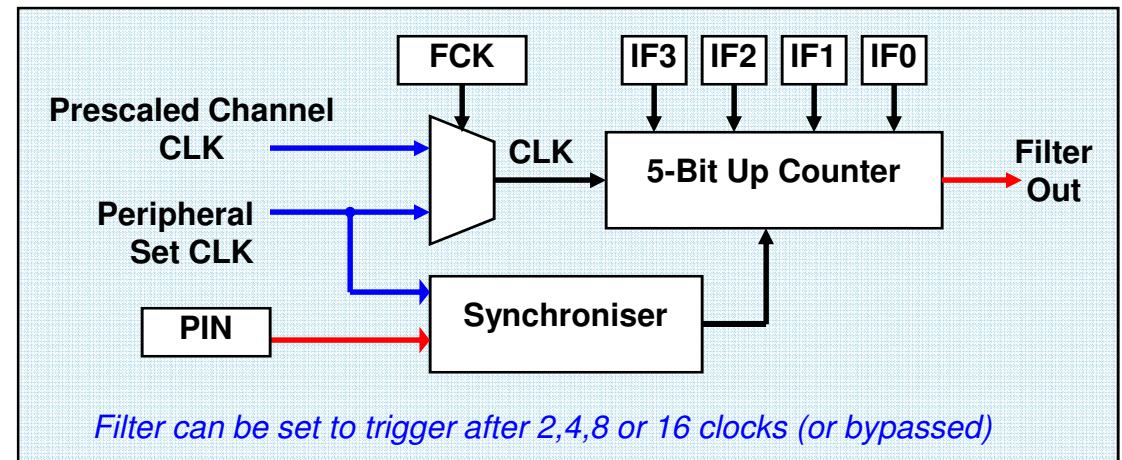


- ▶ If changing an operating mode
 - first change the channel's mode to mode 0 (GPIO)
 - then update A[n] and B[n] registers with the correct values for the next operating mode
 - then write the new operating mode to the CCR[n] register
- ▶ If a Channel is changed from one mode to another without performing this procedure, matches can occur in random time if the contents of A[n] or B[n] were not updated.

Peripherals

EMIOS - Programmable Input Filter

- Filter Consists of a **5 bit programmable up-counter**, clocked by either the channel or peripheral set clock (defined by FCK)
- **Input signal is synchronised to system clock.**
- When the synchroniser output changes state, the counter starts counting up
- If the synchroniser state remains stable for the desired number of selected clocks, the counter overflows on next high clock edge, counter resets and filter output changes

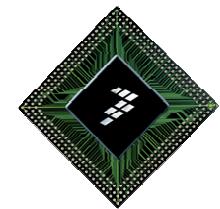


- ▶ Review of Key Points
- ▶ eMIOS channels can be configured for desired timing function
 - Input functions: SAIC, IPM, IPWM, GPIO
 - Output functions: DAOC, OPWM, OPWMT, OPWMC, OPWF, GPIO
 - Modulus counter
- ▶ Global, local or individual channel time bases
 - Input functions can share common time base
 - Output functions can be synchronized
- ▶ Example eMIOS OPWM, Modulus Counter



Designing with Freescale

Core – Programming Model

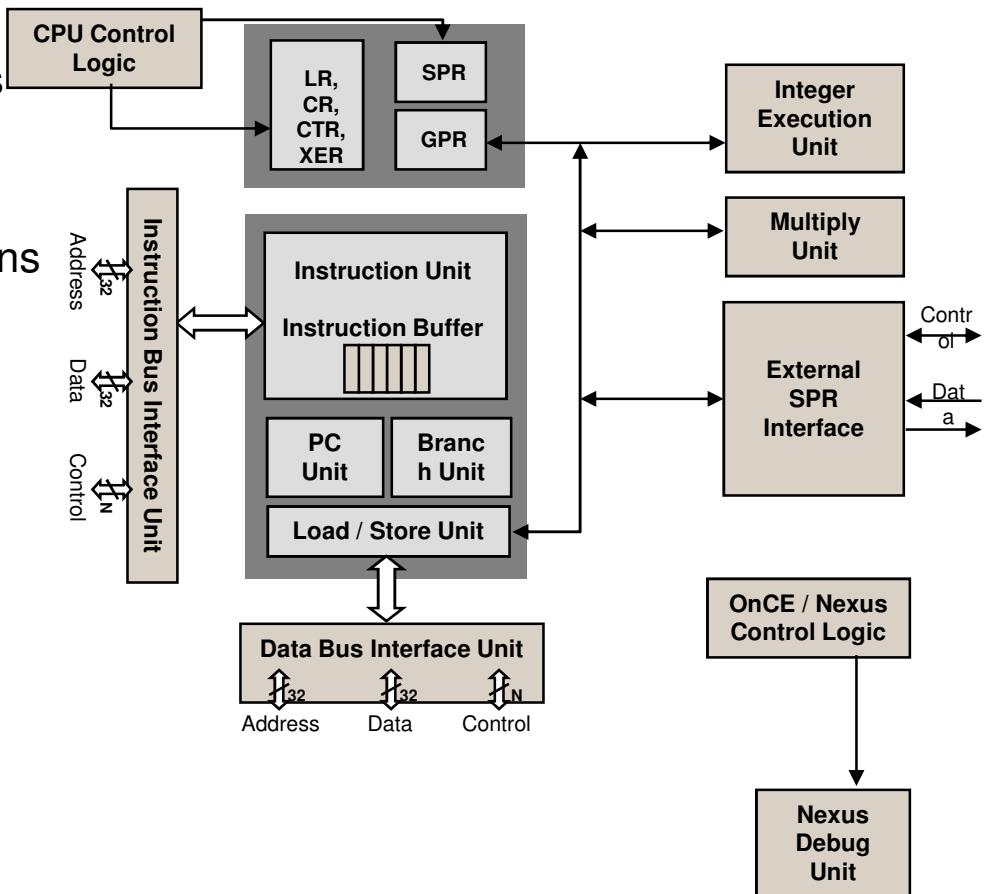


Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

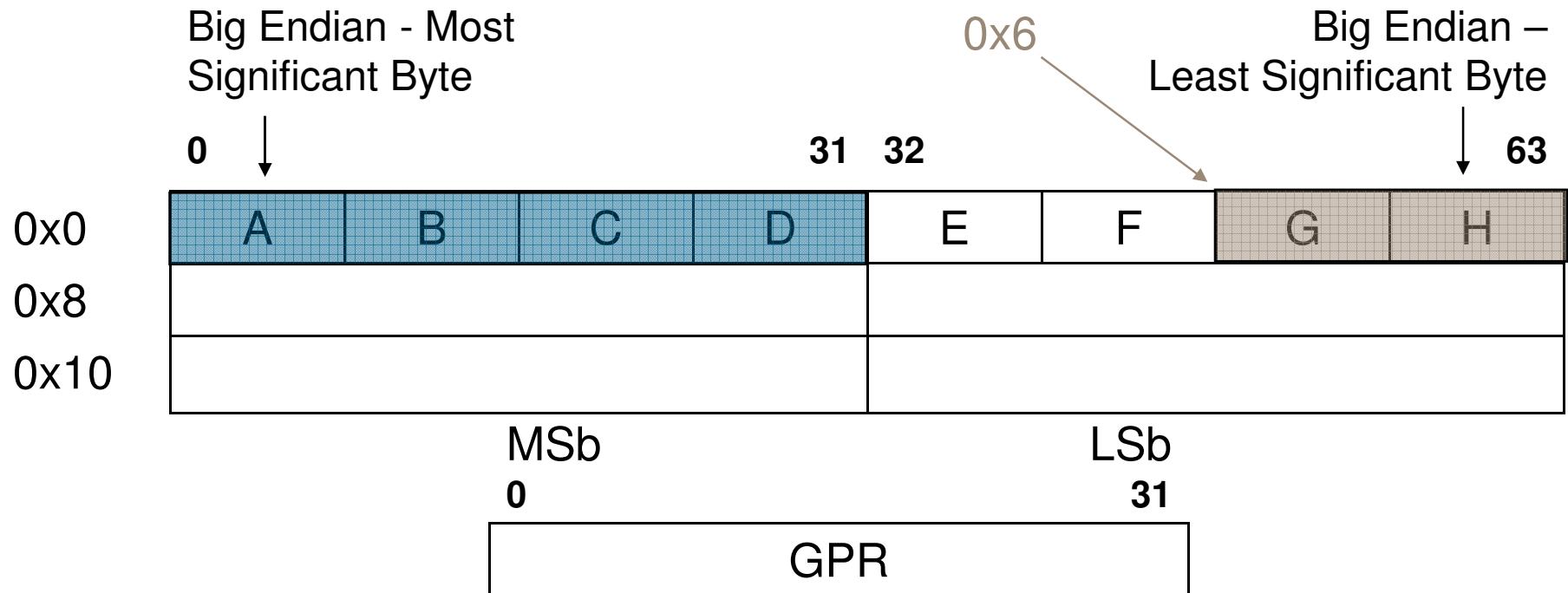
 **freescale**™
semiconductor

e200z0 Overview: Core Diagram (e200z0)

- ▶ Single issue architecture, 32-bit CPU
- ▶ 32-bit PowerPC Book E VLE-only
- ▶ Harvard architecture
 - Independent instruction and data buses
- ▶ Multiple execution units:
 - **Integer Unit:**
 - arithmetic, logical, etc. instructions
 - **Instruction Unit:**
 - single cycle execution of successful look-ahead branches
 - **Load / Store Unit:**
 - pipelined for single cycle execution
 - 1 cycle load latency
 - Big endian support only
 - Misaligned access support
 - **Branch processing unit:**
 - Dedicated branch address calculation adder
 - Branch target buffer (BTB) for branch acceleration



e200z0 Overview: Data Organization in Memory



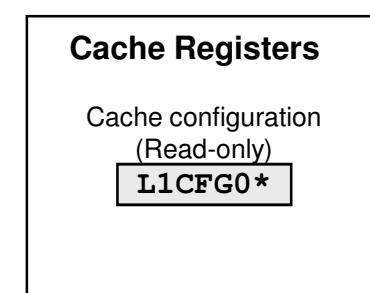
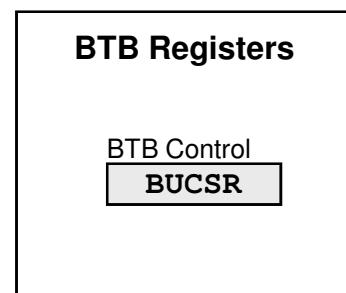
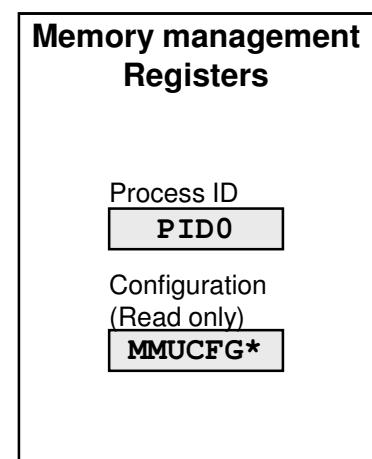
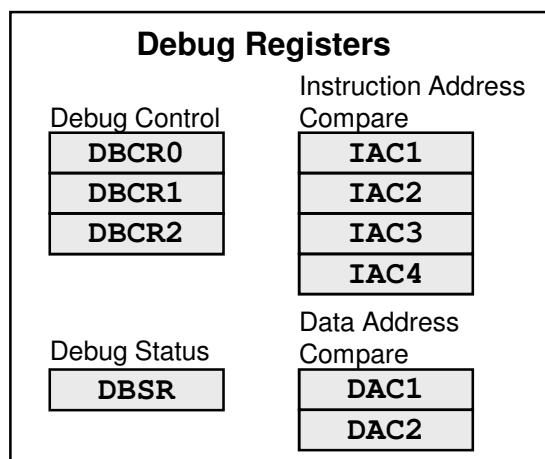
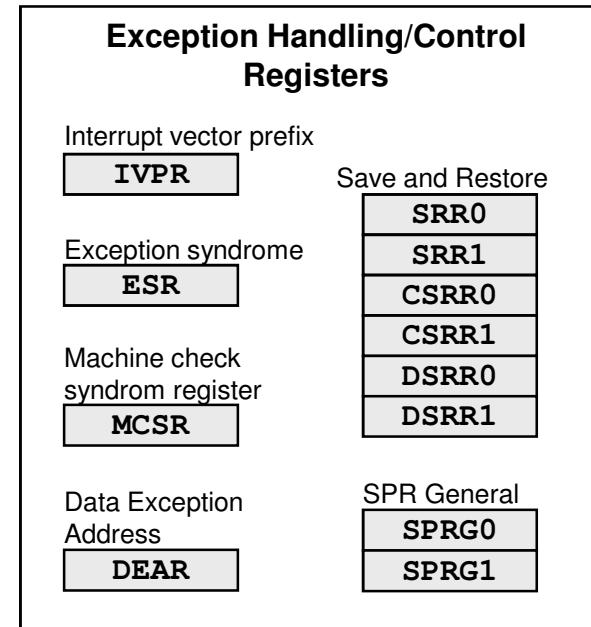
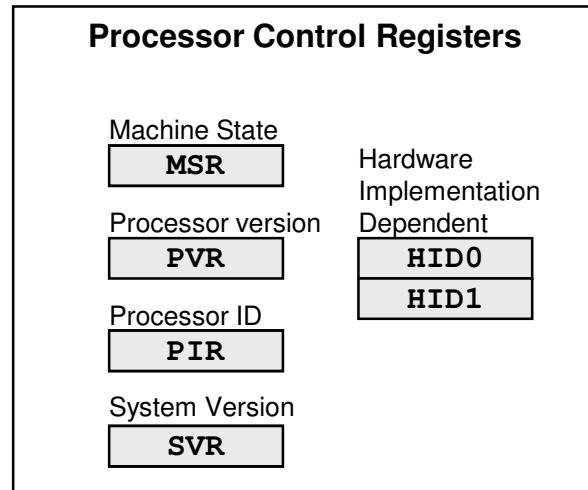
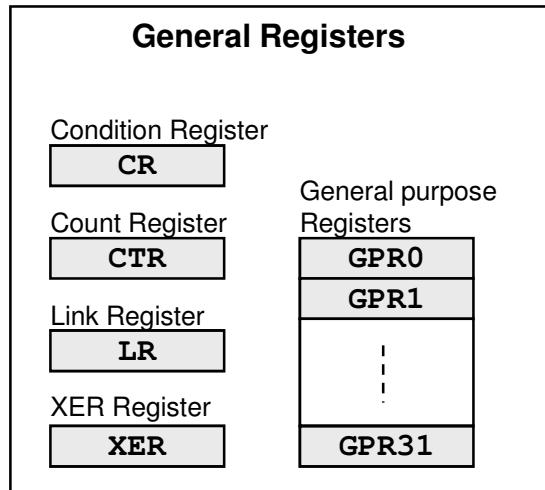
- ▶ All BookE instructions are 32 bits wide.
- ▶ Power architecture is naturally Big Endian, but has switch for Little Endian
- ▶ Examples:

Load word from address 0x0 loads the word “ABCD”

Load half word from address 0x6 loads the half word “GH”

►e200z0 Registers

e200z0 Registers



* Read-only for backward compatibility

e200z0 Registers

General Registers

Register	Description
GPR0-GPR31	Thirty-two 32-Bit GPRs (GPR0–GPR31) serve as data source or destination registers for integer instructions and provide data for generating addresses.
CTR	Count register holds a loop count that can be decremented during execution of appropriately coded branch instructions. It also provides the branch target address for the Branch Conditional to Count Register (bcctr , bcctr1) instructions.
LR	Link Register provides the branch target address for the Branch Conditional to Link Register (bclr , bclr1) instructions, and is used to hold the address of the instruction that follows a branch and link instruction, typically used for linking to subroutines.
XER	Integer Exception Register indicates overflow and carries for integer operations.
CR	Condition register

e200z0 Registers

Processor Control Registers

Register	Description
MSR	The MSR register defines state of the processor.
PVR	This register is a <u>read-only</u> register that identifies the version (model) and revision level of the processor.
PIR	This <u>read-only</u> register is provided to distinguish the processor from other processors in the system.
SVR	<u>Read-only</u> register that specifies a particular implementation of a Zen-based system by a particular business unit at their discretion.
HID0, HID1	Hardware implementation-dependent registers controlling various processor and system functions (setting power modes, debug unit enable etc.)

e200z0 Registers

Interrupt Registers

Register	Description
SRR0 SRR1	Save / Restore Registers 0 & 1 are used to save the machine state on a non-critical interrupt. SRR0 saves the address of the instruction at which execution resumes following the end of an interrupt. SRR1 contains the contents of the MSR when the interrupt is taken.
CSRR0 CSRR1	Save / Restore registers 0 & 1 are used to save the state machine state on a critical interrupt.
DSRR0 DSRR1	Save / Restore registers 0 & 1 are used to save the state machine state on a debug interrupt when enabled HID0[DAPUEN]=1 otherwise CSRR registers are used
IVPR	The Interrupt Vector Prefix Register together with hardwire offsets provide the address of the interrupt handler for different classes of interrupts.
DEAR	The Data Exception Address Register is set to the address of the faulting instruction after most Data Storage Interrupts or Alignment Interrupts.
MCSR	This register provides a syndrome to differentiate between the different kinds of conditions which can generate a Machine Check.
ESR	The Exception Syndrome Register provides a syndrome to differentiate between the different kinds of exceptions which can generate the same interrupts.

e200z0 Registers

Debug Registers

Register	Description
DBCR0-2	These registers provide control for enabling and configuring debug events
DBSR	Debug event status register
IAC1-4	These registers contain addresses and/or masks which are used to specify Instruction Address Compare debug event.
DAC1-2	These registers contain addresses and/or masks which are used to specify Instruction Address Compare debug event.

e200z0 Registers

Key registers overview - Machine State Register (MSR)

► Manipulated during exceptions

- Saved when interrupt occurs to one of the save/restore registers (SRR1, CSRR1, DSRR1)
- Restored when returning from exception

► Accessible by mtmsr and mfmsr instructions

Key Machine State Register Options

Bit	Name	Description	Reset value
14	CE	Critical interrupt Enable	0 (Disabled)
16	EE	External interrupt Enable	0 (Disabled)
17	PR	Problem state (0/1 – Supervisor/User mode)	0 (Supervisor)
19	ME	Machine check Enable	0 (Disabled)
22	DE	Debug interrupt Enable	0 (Disabled)

e200z0 Registers

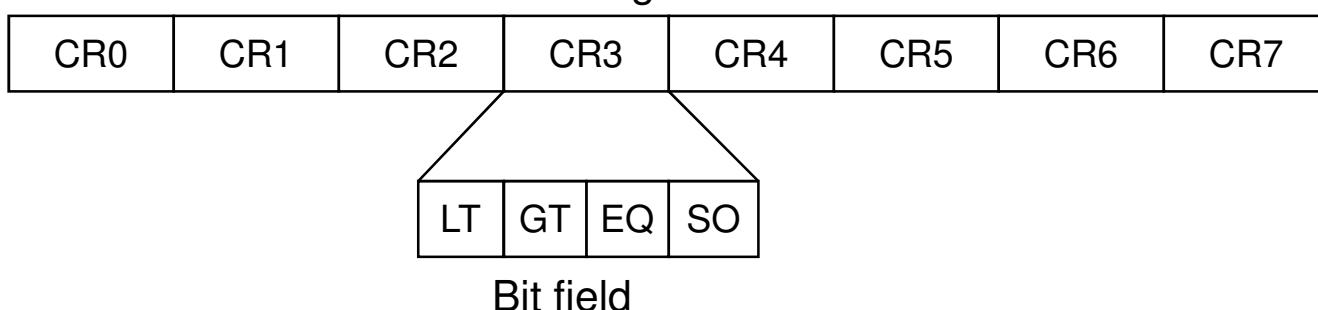
Key registers overview – General Purpose Registers (GPRs)

- 32, 32-bit registers
 - Symbol is r0:r31 for assembly language.
 - Also called gprs or GPRs in documentation
- All integer operands MUST be in gprs
 - If an operand is in memory, it must first be loaded into a gpr with a “load” instruction
 - If a result is to be stored in memory, a “store” instruction is used to store a gpr into memory.

e200z0 Registers

Key register overview - Condition register (CR)

- ▶ Condition Register (CR) has eight identical 4-bit comparison result fields
- ▶ Bit-Fields reflect results of certain arithmetic operations and provides a mechanism for testing and branching.
 - LT less then flag
 - GT greater then flag
 - EQ equal flag
 - SO summary overflow



e200z0 Registers

Special purpose registers (SPRs)

- Most of core registers are special purpose registers (SPRs)
- Each SPR register has the number used in the instruction syntax to access it
- Two assembly instruction are used to read and write
 - `mfspr gpr_dest,spr_src` read from special function register
 - `mtspr spr_dest,gpr_src` write to special function register
- Few SPRs have synchronization requirements for access (ie. HID0)
- Response to invalid SPR access depends on privileged level
 - Illegal exception

General	Alternative	Simplified	Description
<code>mtspr 9,r4</code>	<code>mtspr CTR,r4</code>	<code>mtctr r4</code>	Copy contents of gpr4 to spr9 (CTR)
<code>mtspr 22,r0</code>	<code>mtspr DEC,r0</code>	<code>mtdec r0</code>	Copy contents of gpr0 to spr22 (DEC)
<code>mfspr r11,1</code>	<code>mfspr r11,XER</code>	<code>mfker r11</code>	Copy contents of spr1 (XER) to gpr 11
<code>mfspr r0,26</code>	<code>mfspr r0,SRR0</code>	<code>mfsrr0 r0</code>	Copy contents of spr26 (SRR0) to gpr 0

e200z0 Registers

Application Binary Interface (ABI) Register Use 1 of 2

- ▶ The Application Binary Interface (ABI) for the e200 is the e500 ABI, which is based on the Embedded Application Interface (EABI)
- ▶ This ABI defines certain registers to have **dedicated** uses:
 - Stack pointer is r1
 - Small data area pointers r2 and r13
- ▶ If registers need to be saved across a function call:
 - **Nonvolatile** registers must be saved by called function
 - **Volatile** registers must be saved by calling function

e200z0 Registers

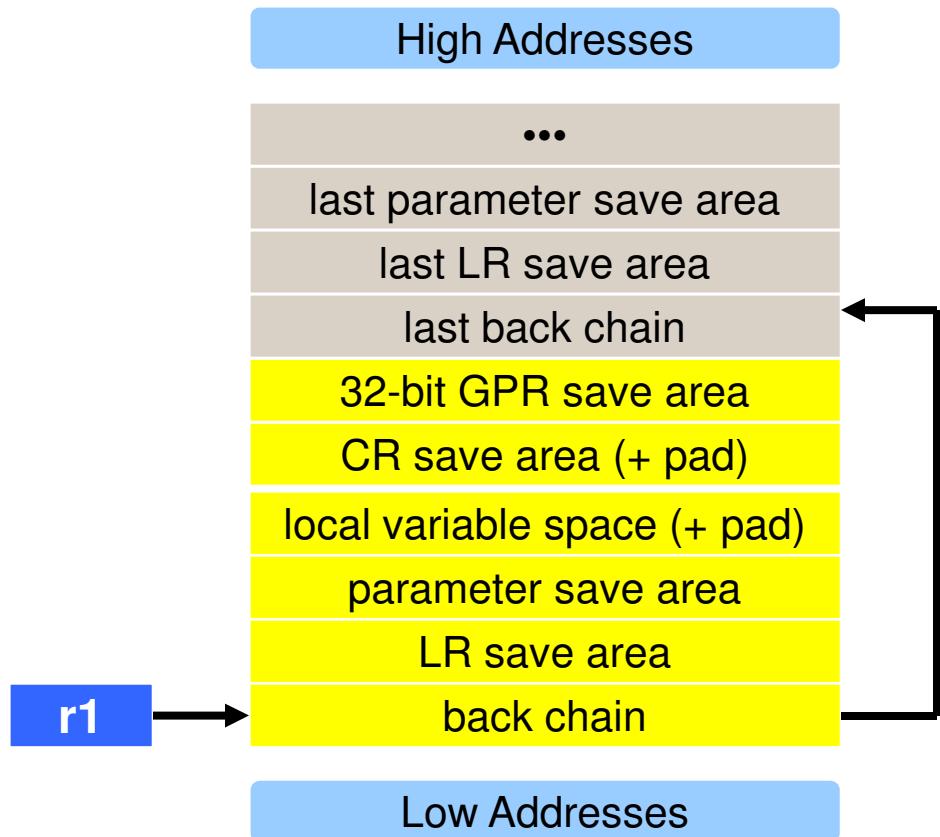
ABI Register Use 2 of 2

Register	Function	Type
r0	Function linkage	volatile
r1	Stack frame pointer	dedicated
r2	Small data area 2 pointer	dedicated
r3-r4	Parameters / Return values	volatile
r5-r10	Parameters	volatile
r11-r12	Function linkage	volatile
r13	Small data area pointer	dedicated
r14-r30	Local variables	non-volatile
r31	Local variables / Env pointer	non-volatile
cr0-cr1	Condition register fields	volatile
cr2-cr4	Condition register fields	non-volatile
cr5-cr7	Condition register fields	volatile
lr	Link register	volatile
ctr	Count register	volatile
xer	Integer exception register	volatile

Programming Model: ABI Stack Frames

Stack grows from high to low addresses

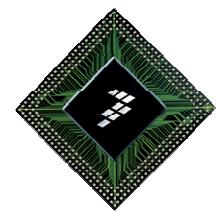
- Only created when necessary
- Size varies as needed
- 16-byte alignment required





Designing with Freescale

Interrupts



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale™**
semiconductor

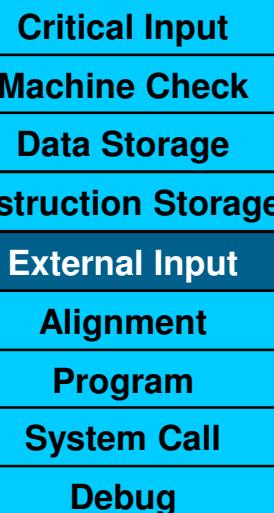
MPC5604B Interrupt Structure

- ▶ **Interrupt Requests from**
- ▶ **Interrupt Controller (INTC)**



INTC in
Software
Vector
Mode

- ▶ **Interrupt Requests from**
- ▶ **Core Exceptions (e200z0)**



CPU
Interrupt

CPU Core

INTC interrupts are assigned one of 16 priority levels, enabling preemption

Interrupt Behavior

Interrupt is recognized

Hardware context switch:

1. **Stores address of next instruction or instruction causing interrupt**
 - Stored into special purpose reg. **Save & Restore Reg 0 (SRR0)**
2. **Stores Machine State (MSR bits 16:31):**
 - Stored into special purpose reg. **Save & Restore Reg 1 (SRR1)**
3. **Alters Machine State:** All bits are cleared in **MSR** except ME
4. **Alters Instruction Pointer:** points to unique **interrupt vector**

Software Interrupt handler (at interrupt vector)

- Execute your handler code, including save/restore registers
- Last instruction, **rfi***, (return from interrupt):
 - Restores MSR bits 16:31 from **SRR1**
 - Restores instruction pointer from **SRR0**

* “Critical” and “debug” interrupts use rcfi and rdfi instruction instead of rfi.

C Function Interrupt Handler -- What to put on stack: EABI “Volatile” Registers

Register	Function	Type
r0	Function linkage	volatile
r1	Stack frame pointer	dedicated
r2	Small data area 2 pointer	dedicated
r3-r4	Parameters / Return values	volatile
r5-r10	Parameters	volatile
r11-r12	Function linkage	volatile
r13	Small data area pointer	dedicated
r14-r30	Local variables	non-volatile
r31	Local variables / Env pointer	non-volatile
cr0-cr1	Condition register fields	volatile
cr2-cr4	Condition register fields	non-volatile
cr5-cr7	Condition register fields	volatile
lr	Link register	volatile
ctr	Count register	volatile
xer	Integer exception register	volatile

e200z0 Core Interrupts

- ▶ External input Exception (EE)
- ▶ enables/disables all interrupts from the interrupt controller.

- ▶ Each exception type can normally be individually enabled or disabled in the **Machine State Register (MSR)**

Core Interrupt Type	IVOR #	IVPR Offset
Critical Input	IVOR 0	0x000
Machine Check	IVOR 1	0x010
Data Storage	IVOR 2	0x020
Instruction Storage	IVOR 3	0x030
External Input	IVOR 4	0x040
Alignment	IVOR 5	0x050
Program	IVOR 6	0x060
System Call	IVOR 8	0x080
Debug	IVOR 15	0x0E0

All e200 Core Interrupts

<u>IVOR #</u>	<u>Interrupt Type</u>	<u>Enables¹</u>	<u>State Saved In</u>	<u>Examples</u>
IVOR0	Critical Input	CE	CSRR0:1	Non-maskable interrupt (pins PD[10], PD[11])
IVOR1	Machine Check	ME	CSRR0:1	ISI, ITLB error on 1 st instr'n of exception handler
IVOR2	Data Storage	-	SRR0:1	Incorrect privilege mode for R/W access
IVOR3	Instruction Storage	-	SRR0:1	Incorrect privilege mode for instruction
IVOR4	External Input	EE, src	SRR0:1	Peripherals, IRQ pins, software
IVOR5	Alignment	-	SRR0:1	Load or store operand not word aligned
IVOR6	Program	-	SRR0:1	Illegal instruction, trap
IVOR7²	FP Unavailable	-	SRR0:1	FP instruction attempt with MSR[FP]=0
IVOR8	System Call	-	SRR0:1	System call, "sc", instruction
IVOR10²	Decrementer	EE, DIE	SRR0:1	Decrementer timeout
IVOR11²	Fixed-Interval Timer	EE, FIE	SRR0:1	Fixed-interval timer timeout
IVOR12²	Watchdog Timer	CE, WIE	CSRR0:1	Watchdog timeout when ENW=1, WIS=0
IVOR13²	Data TLB Error	-	SRR0:1	Data TLB miss in MMU
IVOR14²	Instruct'n TLB Error	-	SRR0:1	Instruction TLB miss in MMU
IVOR15	Debug	DE, IDM	CSSR0:1	ROM Debugger when HID0[DAPUEN]=0
		DE, IDM	DSRR0:1	ROM Debugger when HID0[DAPUEN]=1

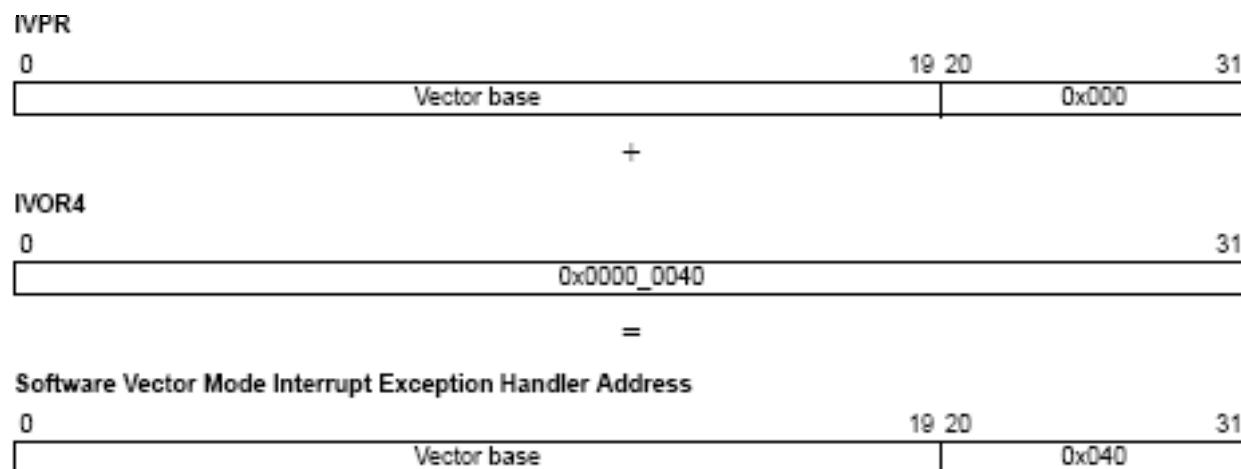
1 CE, ME, EE, DE are in MSR. DIE, FIE, WIE are in TCR. "src" is individual enable for each INTC source.

Debug interrupt, IVOR15, also requires EDM = 0 (EDM and IDM are in DBCR0).

2 Only on e200z1,3,6; unused on e200z0

Interrupt Vector Core Interrupts & INTC Software Vector Mode Interrupts

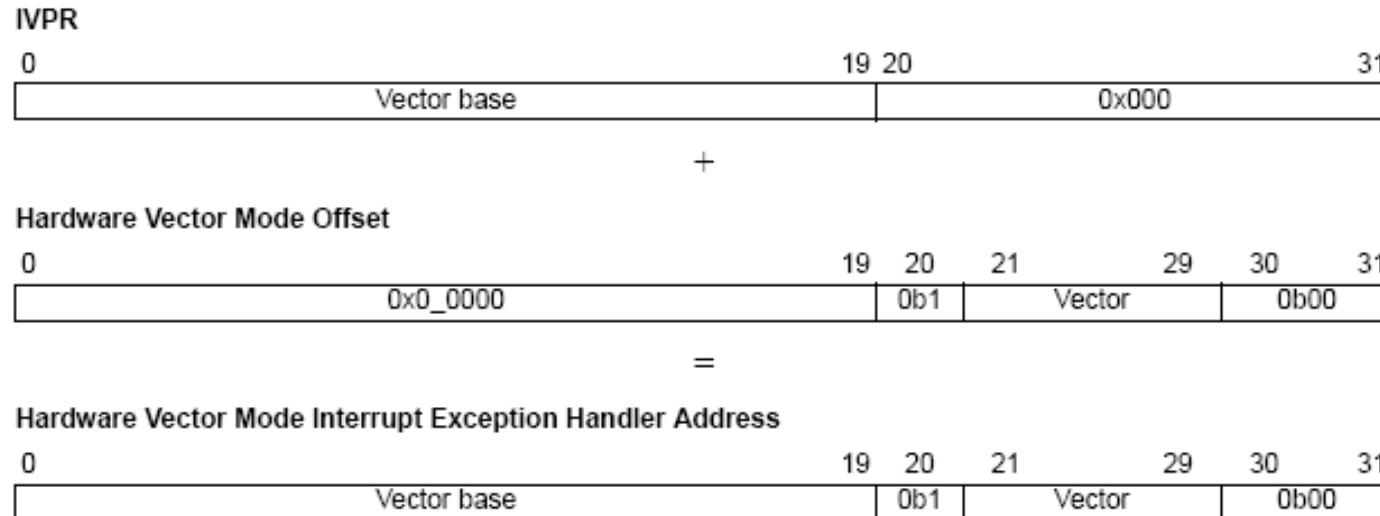
- The interrupt vector address is composed of two components:
 - Vector base address used as a prefix (special purpose register IVPR bits 0:19)
 - A fixed offset base on the IVOR #
- Each interrupt vector address would contain your branch instruction to that handler.
- Example:



Interrupt Vector INTC Hardware Vector Mode Interrupts

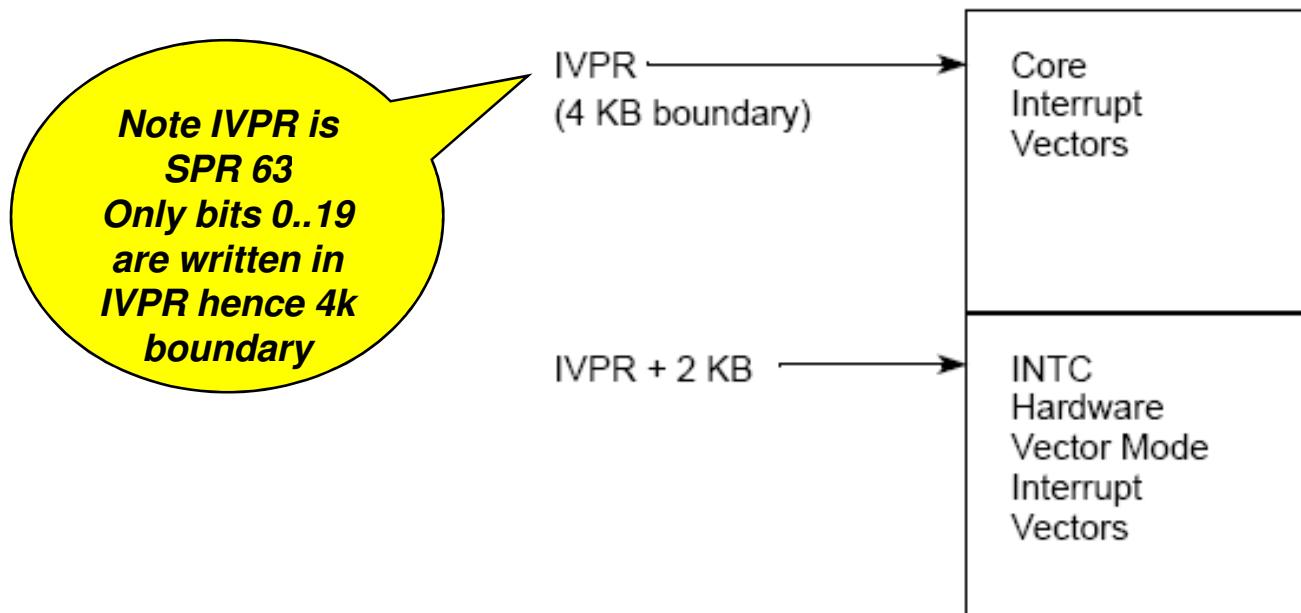
► In hardware vector mode, IVOR4 is not used!

- Each INTC interrupt has a unique 4 byte vector entry in the vector table at IVPR+2KB
- The table entry is calculated by adding the 4 byte vector to the IVPR
- There is no common handler and each interrupt ISR has it's own prologue and epilogue

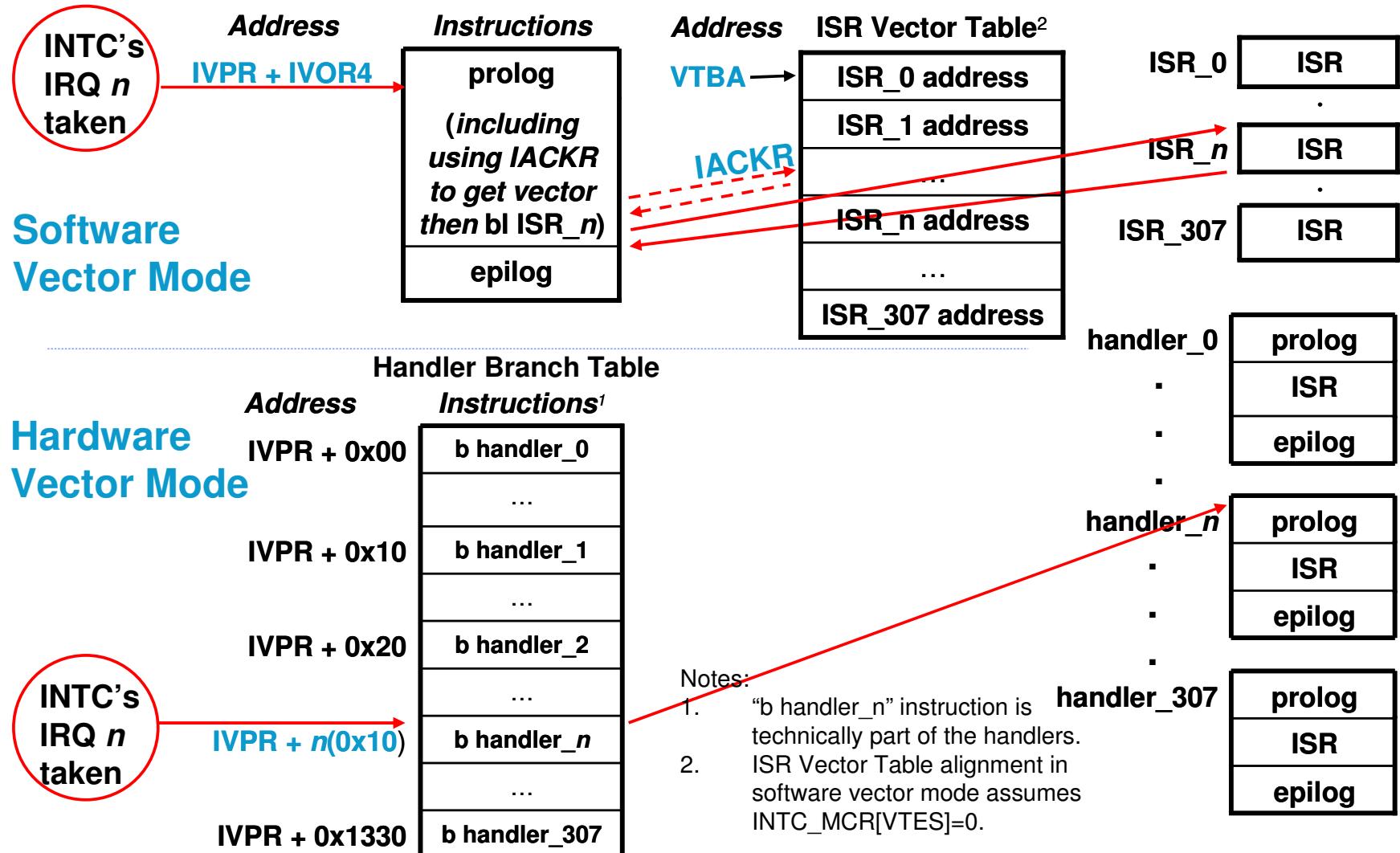


Interrupt controller INTC Hardware and Software Vector Mode

- ▶ The interrupt vectors are located on a 4KByte boundary for e200z0.
 - Hardware vector mode vectors are located at IVPR (Interrupt Vector Prefix Register) plus a 2Kbyte offset



Interrupt controller Software & Hardware Vector Mode



SW vs HW Vector Mode Handler

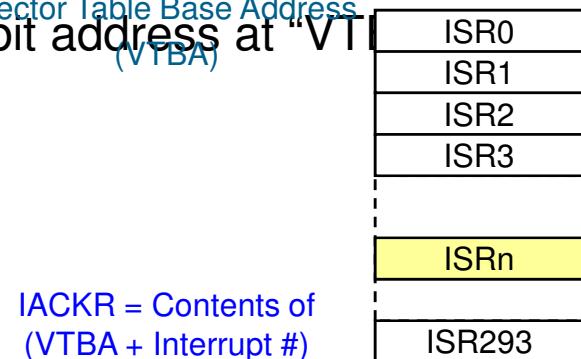
Software Vector Mode (<i>HVEN = 0</i>)	Hardware Vector Mode (<i>HVEN = 1</i>)
<p>HW:</p> <ul style="list-style-type: none"> - Backs up machine state to SRR0:1 - Disables interrupts except CE, ME, DE - Takes External Input Interrupt based on IVPR and offset 0x40 (for “IVOR4”) 	<p>HW:</p> <ul style="list-style-type: none"> - Backs up machine state to SRR0:1 - Disables interrupts except CE, ME, DE - Takes unique IRQ vector based on IVPR and offset which matches INTVEC
<p>SW Prolog:</p> <ul style="list-style-type: none"> - Saves SRR0:1* - Reads INTC_IACKR[INTVEC] - Re-enables MSR[EE]* - Saves other registers 	<p>SW Prolog:</p> <ul style="list-style-type: none"> - Saves SRR0:1* - Re-enables MSR[EE]* - Saves other registers
<p>SW:</p> <ul style="list-style-type: none"> - branches per INTC_IACKR[INTVEC] 	
SW ISR (clears interrupt flag)	SW ISR (clears interrupt flag)
<p>SW Epilog:</p> <ul style="list-style-type: none"> - Executes <i>mbar</i> to ensure IRQ flag cleared - Restores most registers - Disables EE* and writes to INTC_EOIR - Restores remaining registers and returns (rfi) 	<p>SW Epilog:</p> <ul style="list-style-type: none"> - Executes <i>mbar</i> to ensure IRQ flag cleared - Restores most registers - Disables EE* and writes to INTC_EOIR - Restores remaining registers and returns (rfi)

* When nesting



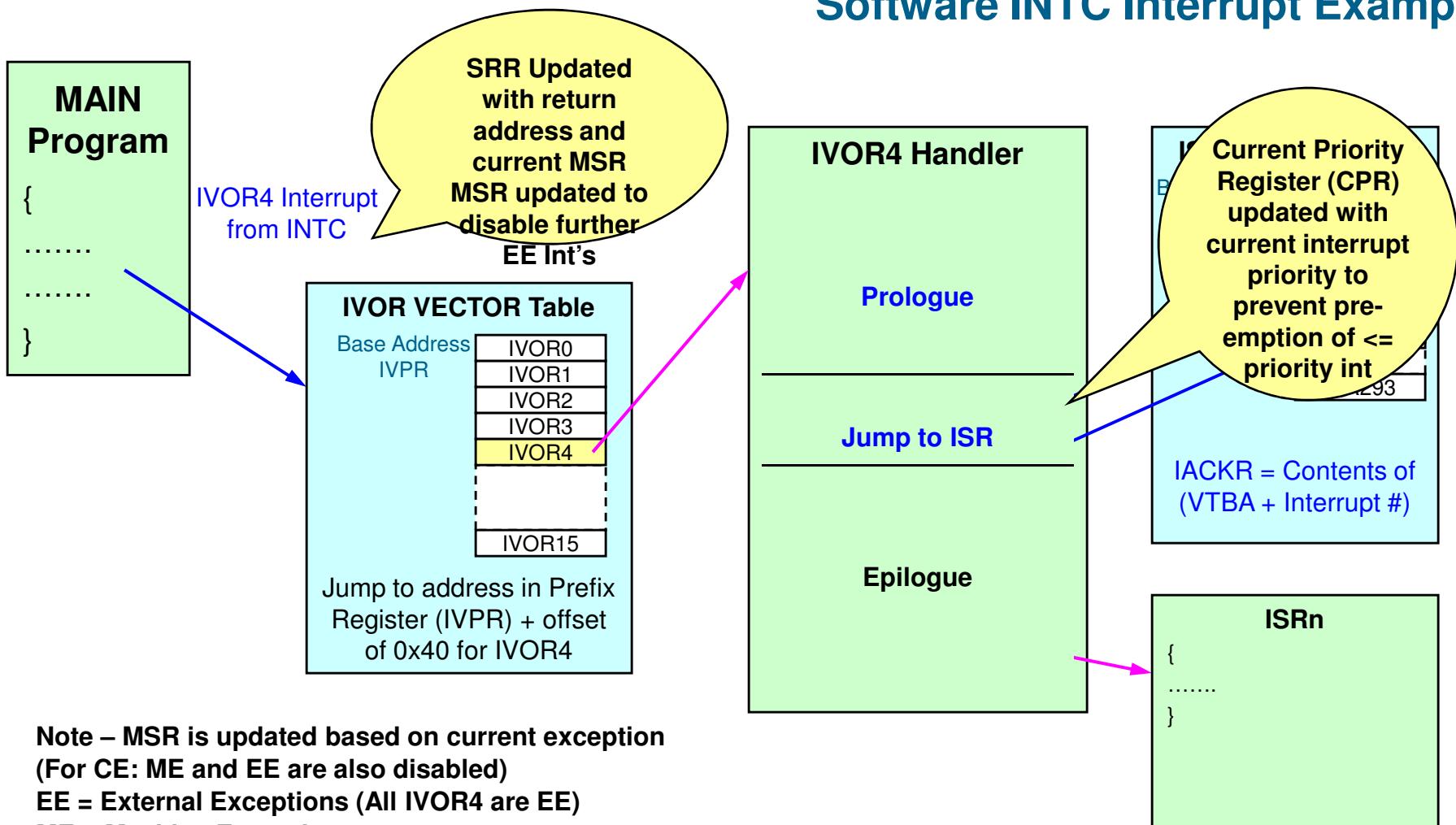
Interrupt controller Software Vector Mode Interrupt Acknowledge

- ▶ The INTC has a an Interrupt Acknowledge Register (IACKR) for each core to identify the interrupt source number and perform “housekeeping”.
 - **INTC_IACKR** value is used as an offset into an ISR vector table for these INTC software mode interrupts
 - Reading this register acknowledges the interrupt has taken place and prevents the same interrupt occurring again
 - Reading IACKR also calculates and returns the address of the relevant Interrupt Service Routine based on reading the 32-bit address at “**VTB**”



IACKR = Contents of
(VTBA + Interrupt #)

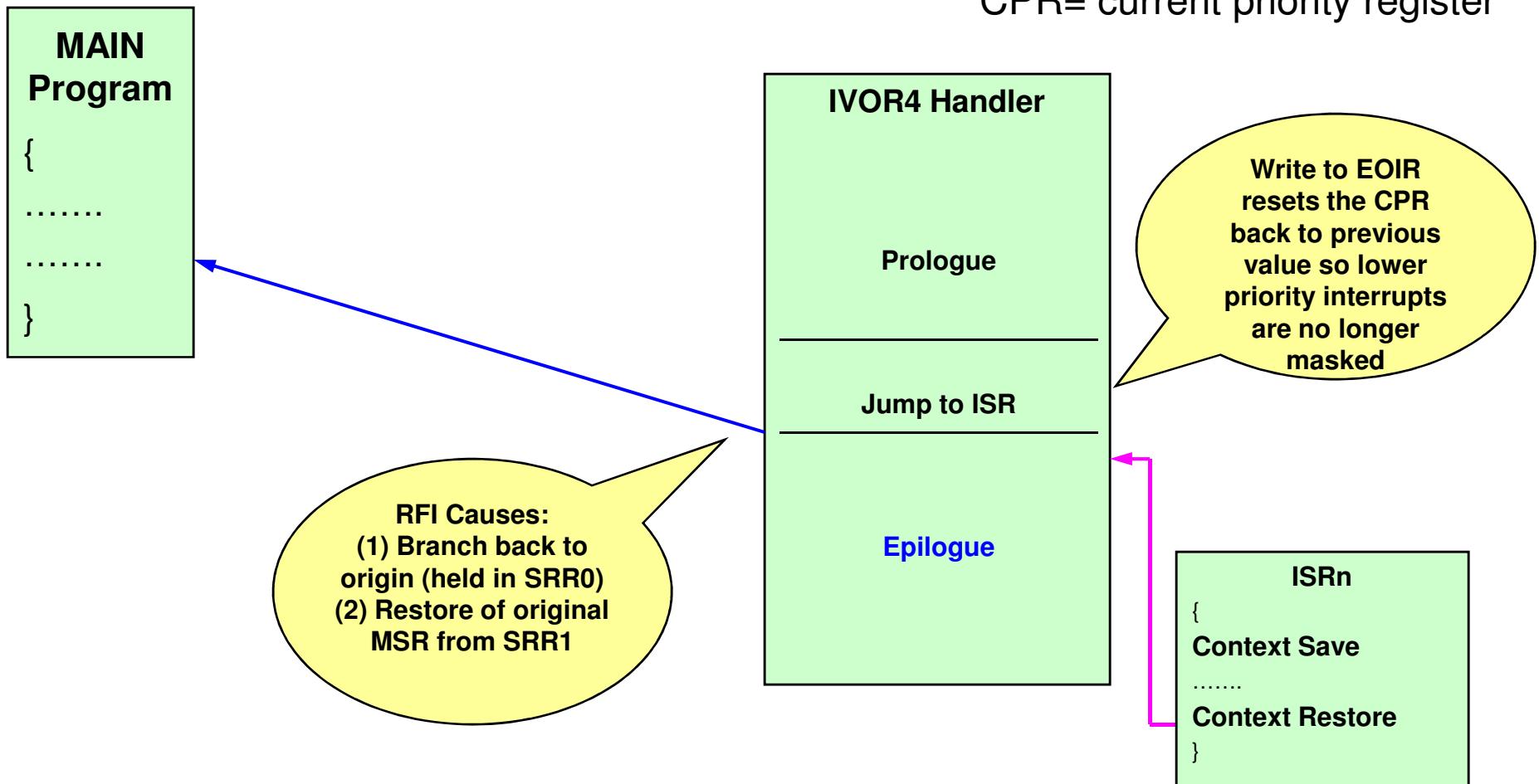
Interrupt controller Software INTC Interrupt Example



**Note – MSR is updated based on current exception
(For CE: ME and EE are also disabled)
EE = External Exceptions (All IVOR4 are EE)
ME = Machine Exceptions
CE = Critical Exceptions (Incl Watchdog)**

Interrupt controller Software INTC Interrupt Example

CPR= current priority register



Interrupt controller Hardware Vector Mode Details

► In hardware vector mode, IVOR4 is not used!

- Each INTC interrupt has a unique 4 byte vector entry in the vector table at IVPR+2KB
- The table entry is calculated by adding the 4 byte vector to the IVPR
- There is no common handler and each interrupt ISR has it's own prologue and epilogue

IVPR

0	19 20	31
Vector base		0x000

+

Hardware Vector Mode Offset

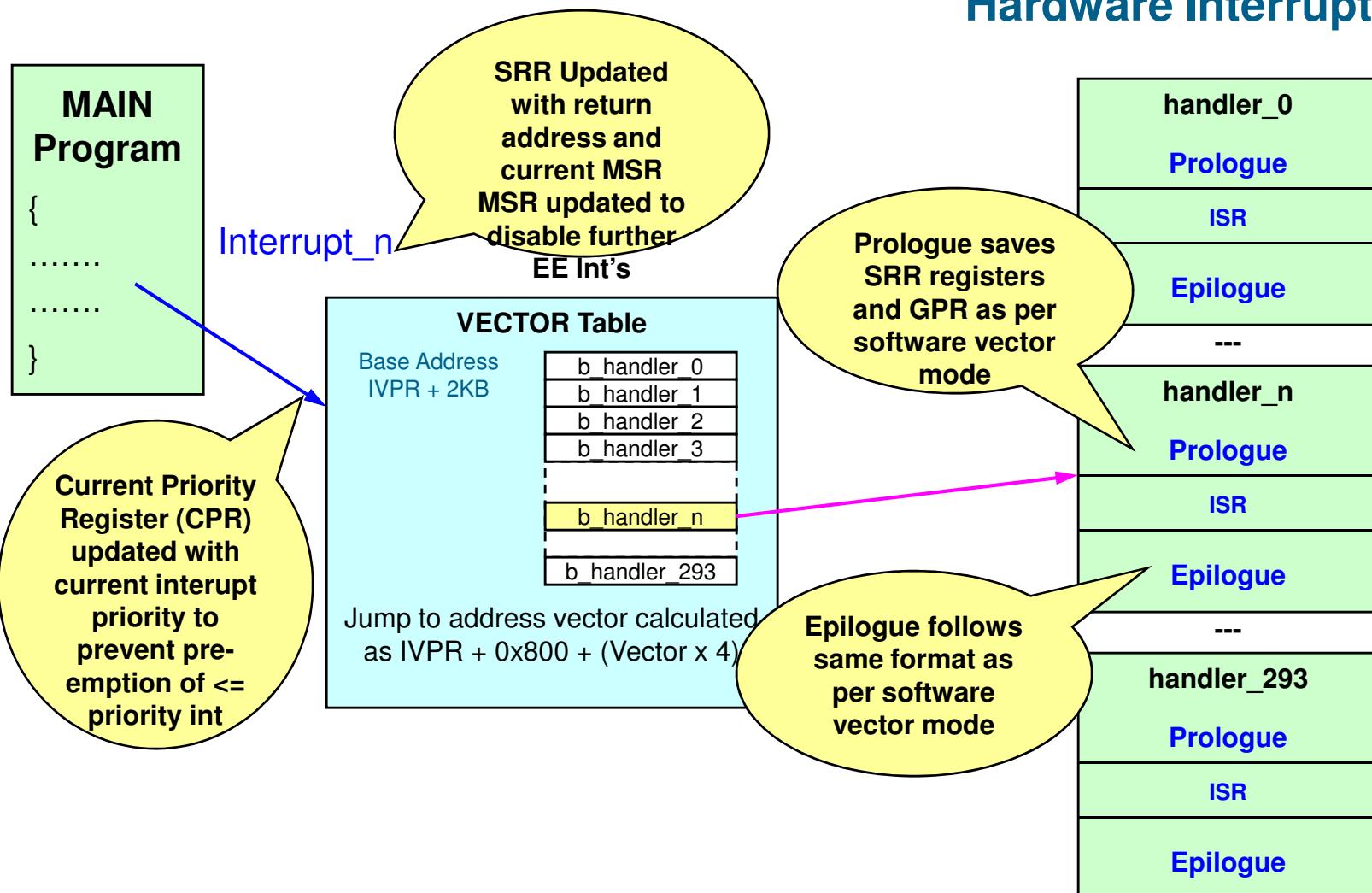
0	19	20	21	29	30	31
0x0_0000		0b1	Vector		0b00	

=

Hardware Vector Mode Interrupt Exception Handler Address

0	19	20	21	29	30	31
Vector base		0b1	Vector		0b00	

Interrupt controller Hardware Interrupt Example



Interrupt controller INTC Preemption

- ▶ Interrupt sources are assigned **1 of 16 priority levels**
 - 15 is highest priority, 0 is lowest
 - Each interrupt source's priority is specified in its **Priority Select Register** (8 bits wide), **INTC_PSRx**
- ▶ The Interrupt Controller records the current interrupt's priority.
 - The current priority is in the **Current Priority Register**, **INTC_CPR**
 - Only interrupts with a priority higher than current priority can be recognized, allowing preemption.
- ▶ Preempted priorities are automatically pushed/popped to/from a **LIFO** in the interrupt controller.

Core Interrupts: Critical Interrupt

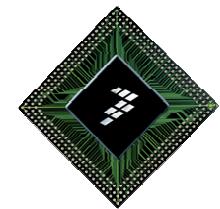
- ▶ Typical interrupts: the processor automatically saves the machine state into Save and Restore Registers SRR0 and SRR1
 - Last interrupt handler instruction is “return from interrupt” (rfi)
- ▶ Critical Interrupt: A second level of interrupt, saving machine state into registers CSRR0 and CSRR1
 - A critical interrupt can be taken while a regular interrupt is handled (i.e. when MSR[EE] = 0)
 - Last interrupt handler instruction is “critical return from interrupt” (crfi)
- ▶ Critical interrupt only applies to certain exceptions:
 - Critical Input (requires MSR[CE] set)
 - Machine Check (requires MSR[ME] set)
 - Debug (requires MSR[DE] set)
 - The debug exception could be configured to use DSRR0:1 and drfi by setting HID0[DAPUEN]

- ▶ Review of Key Points
- ▶ INTC Software Vector Mode
 - Interrupt vector is fetched by software
 - Memory efficient due to common prologue, epilogue
- ▶ INTC Hardware Vector Mode
 - Interrupt vector is automatically fetched by hardware
 - Saves some time
- ▶ Example INTC SW mode with VLE, or
INTC HW mode with VLE



Designing with Freescale

EEPROM Emulation



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**™
semiconductor

Memories: EEPROM Emulation Overview

- ▶ EEPROM supported using emulation in Data Flash based on RWW functionality
- ▶ How to handle the data depends on two main factors
 - Size of data to be stored (& consistency of size)
 - Frequency of update
- ▶ Two basic data schemes offered – fixed and variable length
 - Variable record length scheme more efficient when large amounts of data stored
 - However, data read software more complex & slower
 - Freescale drivers are implementing variable length record scheme
- ▶ Emulated EEPROM using 2 sectors (1 of 16KB Data Flash each)
- ▶ Autosar EEPROM emulation is part of the Autosar package
- ▶ Also non-Autosar EEPROM emulation software is available (in Beta for C90 Flash)
- ▶ Flash Low-Level Driver is also available in Beta

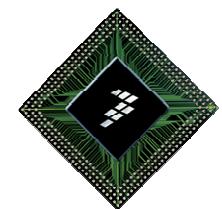
Freescale, the Freescale logo, Altivec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.





Designing with Freescale

ADC, CTU



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale™**
semiconductor

1.5 MB MPC560xB Mapping: Signals, ADC, CTU

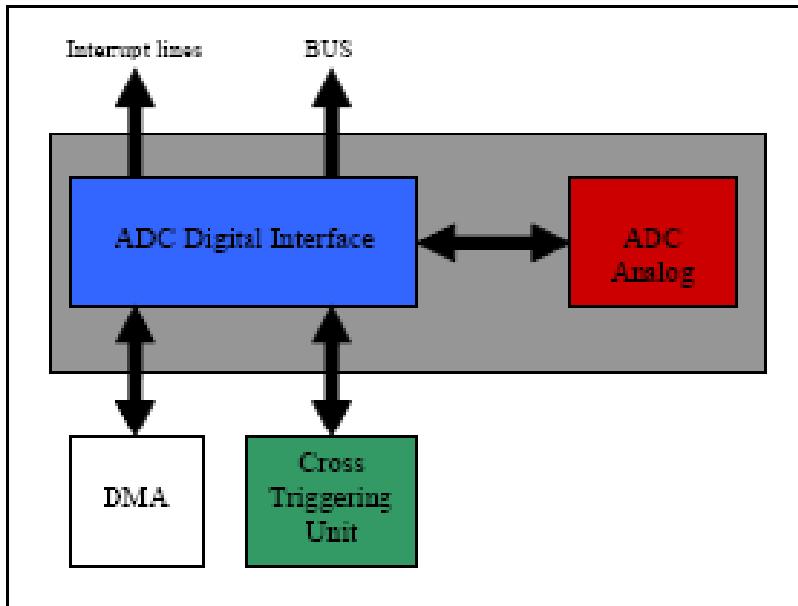
(per MPC5607B Microcontroller Ref. Manual Rev 2)

Signal Name (per Table 2-3) <i>* Not on MPC5604B/C</i>	ADC Channel # (per Fig 23-1)		CTU EVTCFTRx [CHANNEL_VALUE] (per Table 30-6)
	ADC0	ADC1*	
<i>ANP 0:15*</i>	<i>0:15</i>	<i>0:15</i>	<i>0:15</i>
-	-	-	-
ANS 0:15	32:47	-	16:31
<i>ANS 16:27*</i>	<i>48:59</i>	-	
-	-	-	-
ANX 0, MA 0:7	64:71	-	32:39
ANX 1, MA 0:7	72:79	-	40:47
ANX 2, MA 0:7	80:87	-	48:55
ANX 3, MA 0:7	88:95	-	56:63

MPC560xB ADC Trigger Options

Option	Selected Channels	Trigger Initiation	Trigger Enables
Normal • Scan or • One shot modes	Multiple , as selected in NCMR	<ul style="list-style-type: none"> Software sets NSTART bit 	None
Injected • One shot mode only	Multiple , as selected in JCMR	<ul style="list-style-type: none"> Software sets JSTART bit <i>or</i> PIT2 expires 	<ul style="list-style-type: none"> Software: none PIT2: PIT2 configured and MCR[JTRGEN]
Cross Trigger Unit (CTU)	Single , as selected in CTU_EVTCFGRx [CHANNEL_VALUE]	<ul style="list-style-type: none"> PIT3, PIT7 <i>or</i> eMIOS channels (up to 46) 	<ul style="list-style-type: none"> CTU_EVTCFGRx MCR[CTUEN & ADC_SEL] EMIOS_CHx_CCR [DMA, FEN]

ADC overview

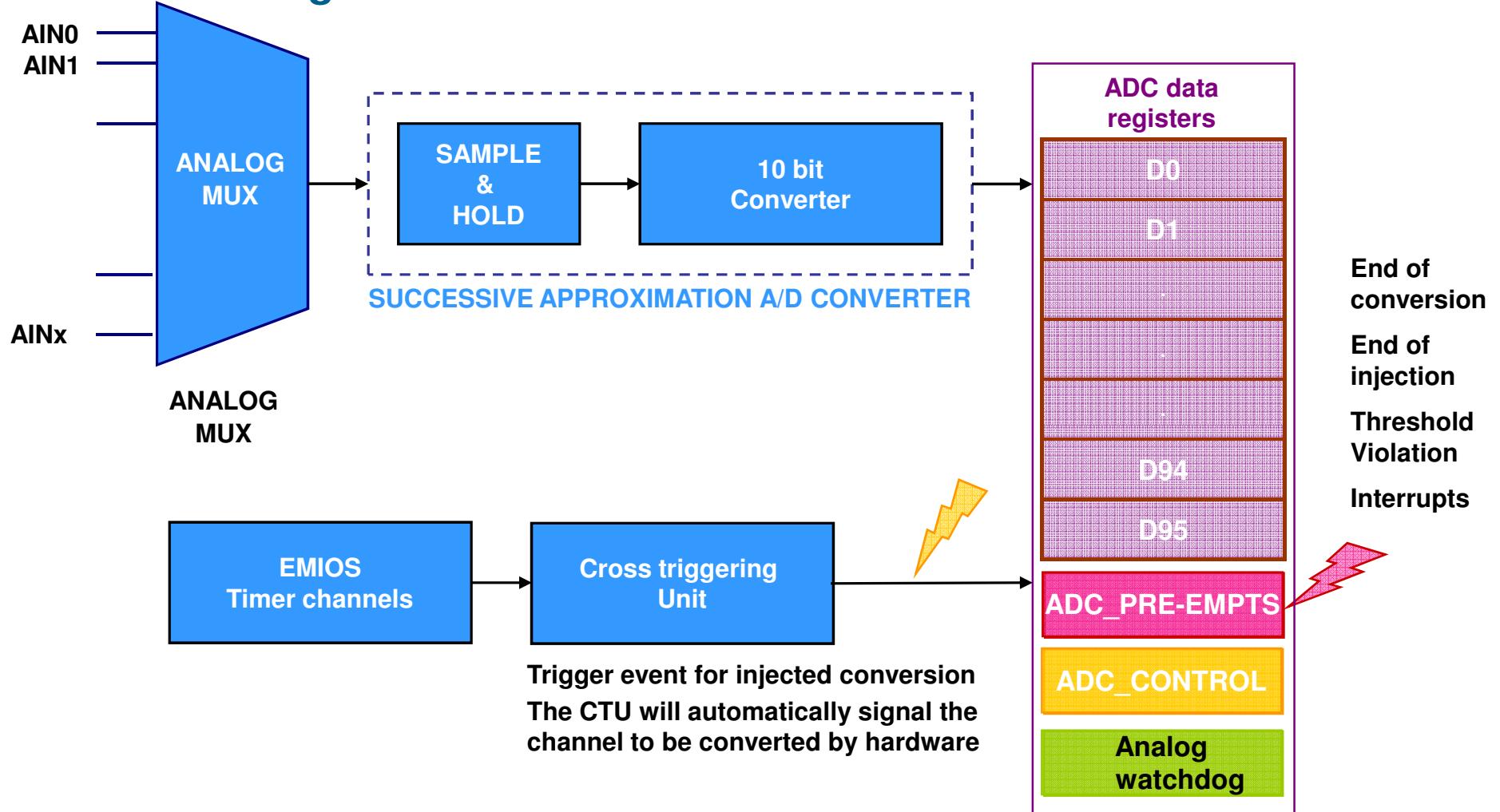


10-bit ADC resolution, on MPC5605/6/7B additional 12 Bit ADC

- Supports conversions time down to 650ns
- internal clock will be system clock/2
- ADC is specified from 6MHz to 32MHz
- Up to 36 single ended inputs channels, expandable to 64 channels with external multiplexers
- Internally multiplexed channels
 - 10-bit ± 2 counts accuracy (TUE) available for 16ch
 - 10-bit ± 3 counts accuracy (TUE) available for up to 20ch
- Externally multiplexed channels
 - 10-bit ± 3 counts accuracy (TUE) available for up to 32ch
 - Internal control to support generation of external analog multiplexer selection
- Dedicated result register for channel
- 3 independently configurable sample and conversion times
- Support for one-shot, scan, injection and triggered injected (CTU) conversion modes
- Independently configurable parameters for channels

Peripherals

ADC Block Diagram



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

Peripherals

ADC conversion modes

Normal conversion mode

one-shot mode or scan mode possible



Injected conversion of channels I and J

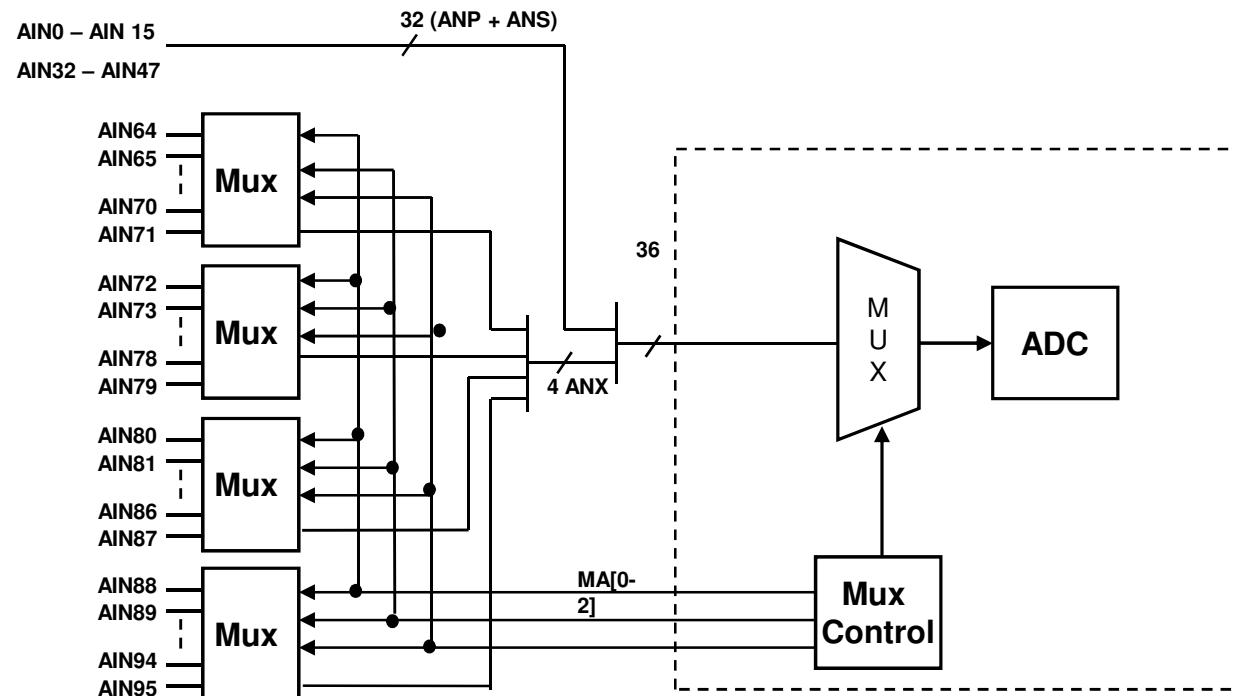


The ongoing channel conversion is interrupted and the injected conversion chain is processed first. After the injected chain is converted the normal conversion resumes from the last aborted channel.

Normal conversion resumes from the last aborted channel.

Peripherals External ADC multiplexing

- ▶ The ADC offers hardware support for external multiplexing
- ▶ Each of the 4 dedicated internal channels (ANX pins) can support up to 8 external multiplexed channels, yielding up to 32 channels with dedicated internal result registers
- ▶ 3 external multiplexer select channels (MA[0-2], PE[5-7]) are provided in the Bolero pinout



Presampling (1)

- ▶ Presampling is used to charge the sample and hold capacitor with a defined value before the actual sampling of the channel
- ▶ This allows to avoid that in case of an open channel (failure), the sample and hold capacitor remains charged at the previously converted value, simulating a valid channel voltage
- ▶ For presampling, the channel can be connected to one of four internally generated voltages
- ▶ Presampling can be enabled/disabled individually per channel
 - PSR0 – presampling for channel 0 to 31(precision channels)
 - PSR1 - presampling for channel 32 to 63 (extended internal channels)
 - PSR2 - presampling for channel 64 to 95 (external channels)
- ▶ PRECONV bit in Presampling Control Register (PSCR) defines if presampling is followed by the conversion.
 - PRECONV = 1 - presampling is followed by the conversion
 - PRECONV = 0 - sampling will be bypassed and conversion of presampled data will be done

Presampling (2)

Figure 237. Presampling sequence

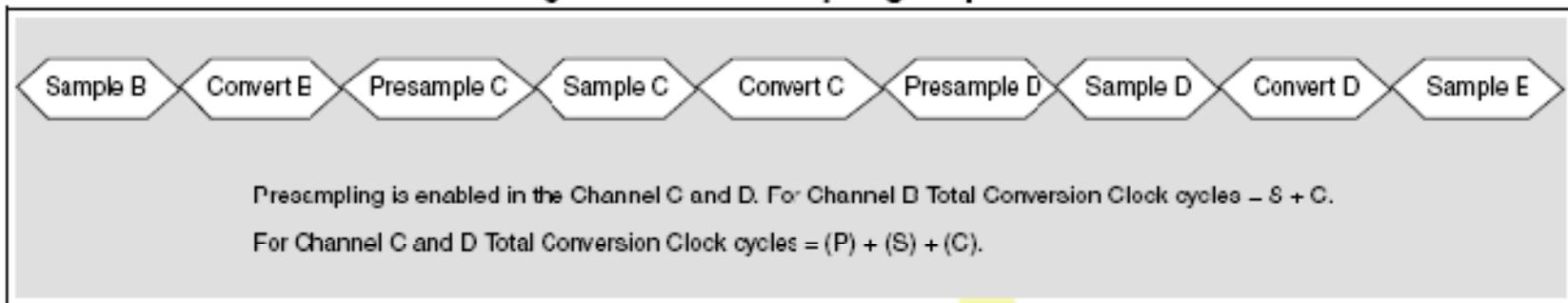
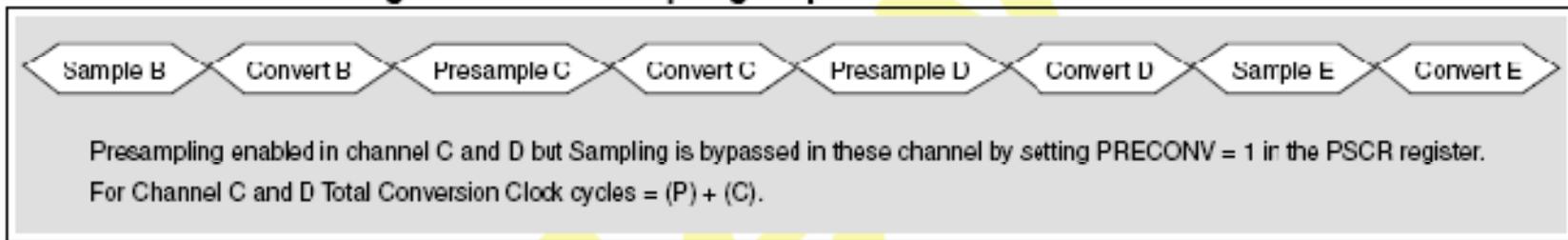
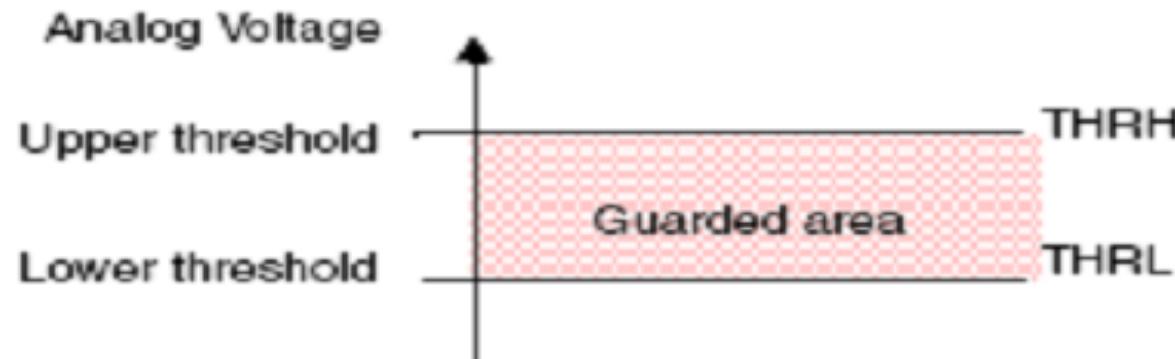


Figure 238. Presampling sequence with PRECONV = 1



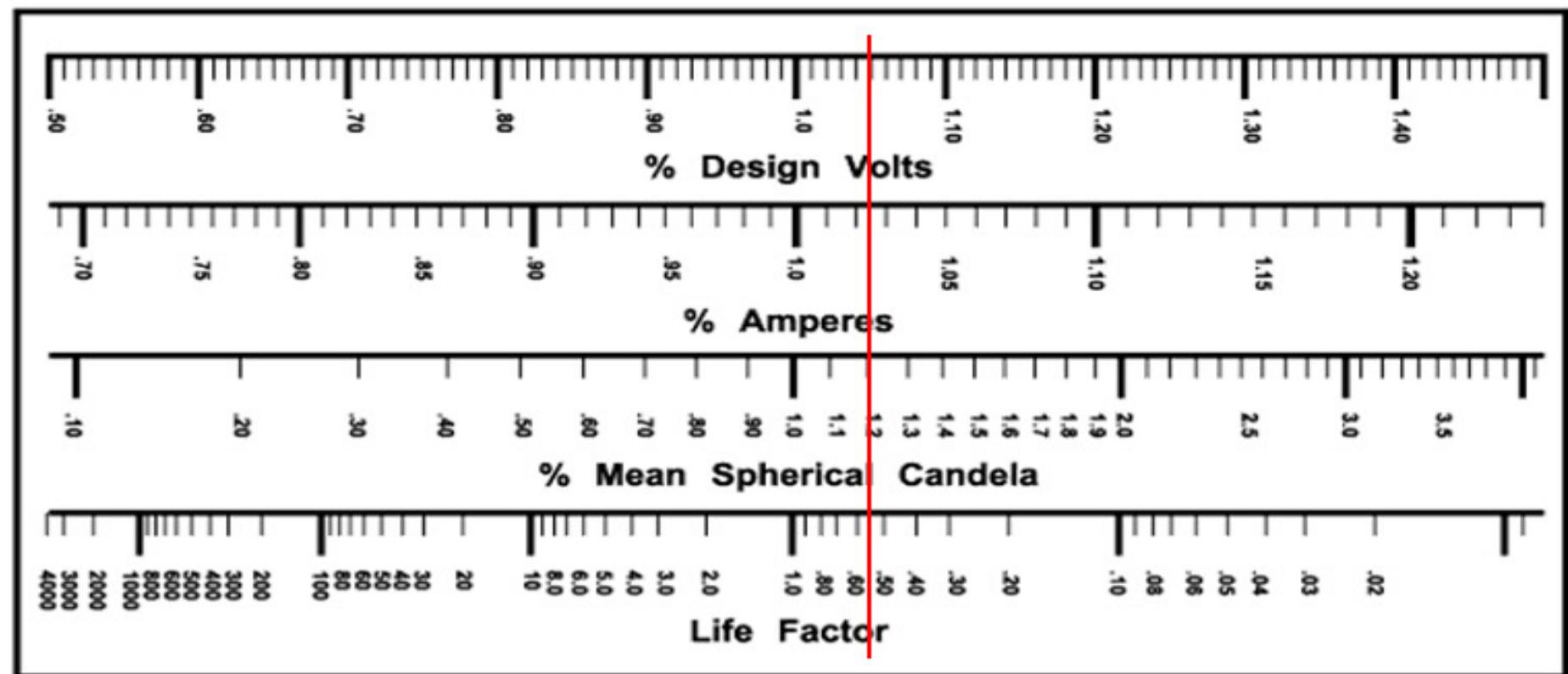
Programmable Analog Watchdog

- ▶ **4 to 6 Analog Watchdogs** can monitor any ADC channel
 - ▶ Number depends on device implementation
- ▶ Programmable UPPER and LOWER threshold
- ▶ Dedicated ADC (WD) Interrupt on UPPER and/or LOWER threshold violation
- ▶ Can be used in a lighting application to trim SMARTMOS devices to prolong LED life
 - Use EMIOS (OPWMT mode) -> CTU -> ADC to continually monitor LED voltage
 - 0% CPU loading
 - Only use CPU in event of ADC watchdog interrupt



Programmable Analog Watchdog

- ▶ 5% over voltage decreases life expectancy of LED by >40%



ADC Mode Priority Summary

- ▶ When a trigger pulse and the injected channel are received internally by the **CTU**, the conversion starts. The CTUSTART bit in the MSR is set automatically at this point and it is also automatically reset when the triggered injected conversion is completed.
- ▶ If an **injected conversion** (programmed by the user by setting the JSTART bit) is ongoing and a pulse is received on the CTU trigger input line, then the injected channel conversion chain is aborted and only the CTU injected conversion proceeds. Aborting the injected conversion, the JSTART bit in the MSR is reset to '0'. That abort is signalled through the status bit JABORT in the MSR.
- ▶ If **normal conversion** is ongoing and a pulse is received on the CTU trigger input line, then ongoing channel conversion is aborted and the triggered injected conversion is processed. When it is finished, normal conversion resumes from the channel at which the normal conversion was aborted.
- ▶ If **another CTU trigger** pulse is received before the end of conversion, that request is discarded.
- ▶ When a **normal conversion** is requested during CTU conversion (CTUSTART bit = '1'), the normal conversion starts when CTU conversion is completed (CTUSTART reset).
- ▶ Otherwise, when an **injected conversion** is requested during CTU conversion, that later conversion is discarded and the JSTART bit in the MCR is immediately reset.

ADC Channel Masks

- ▶ Individual channel mask bits enable:
 - ▶ Normal conversion
 - ▶ Injected conversion

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
rw															
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
rw															

Figure 23-27. Normal Conversion Mask Registers (NCMR[0..2])

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
rw															
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
rw															

Figure 23-28. Injected Conversion Mask Registers (JCMR[0..2])

ADC – Channel Data Registers

- ▶ Each channel has a data register containing:
- ▶ **VALID** indication of new value written (cleared when read)
- ▶ **OVERW**rite data indication that previous conversion data is overwritten by a new conversion
- ▶ **RESULT** indication of conversion mode: normal, injected, CTU
- ▶ **CDATA** with channel's converted data value

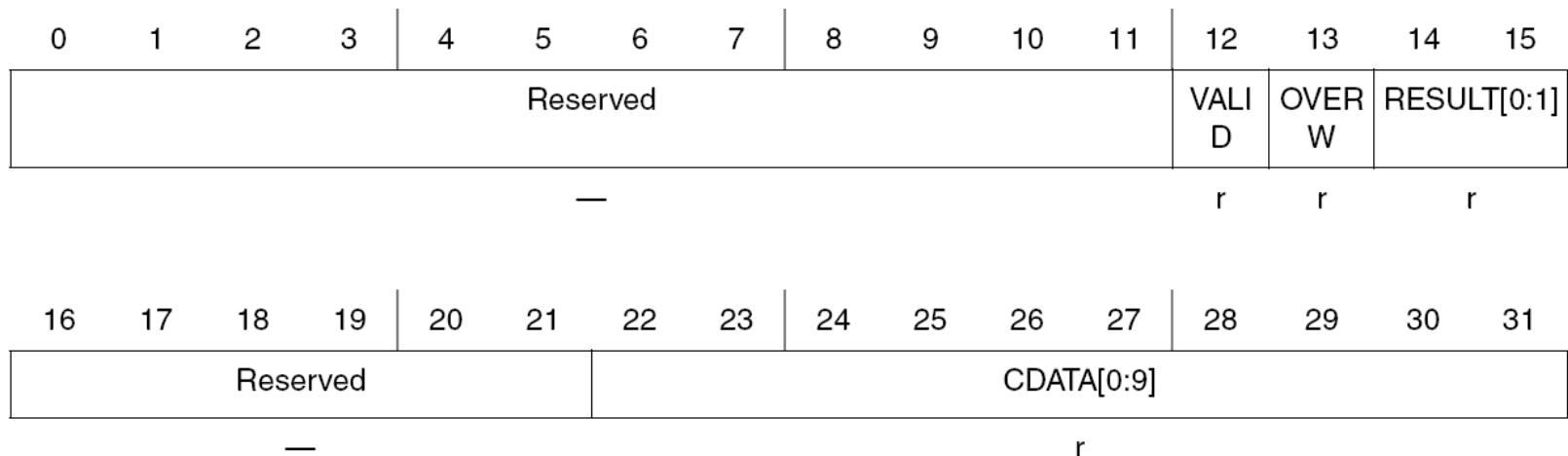


Figure 23-31. Channel Data Register (CDR[0..95])

ADC Interrupts

- ▶ Each ADC converter has two interrupt vectors to the INTC:
- ▶ **ADC_EOC**: ADC End of conversion
 - End of (normal) chain conversion
 - End of (normal) channel conversion (channels selected separately)
 - End of injected chain conversion
 - End of injected channel conversion
 - End of CTU conversion
- ▶ **ADC_WD**: ADC watchdog
 - Watchdog0:3 or 5 high thresholds exceeded
 - Watchdog0:3 or 5 low thresholds exceeded

CTU Lite

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.



- ▶ The Cross Triggering Unit Lite on the Bolero family is a link between timers (eMIOS or PIT) and the ADC
- ▶ The CTU Lite automatically transforms timer events into ADC conversions without main CPU intervention
- ▶ The real-time behavior (synchronization) between timer events and ADC conversions is guaranteed

CTU Lite Block Diagram

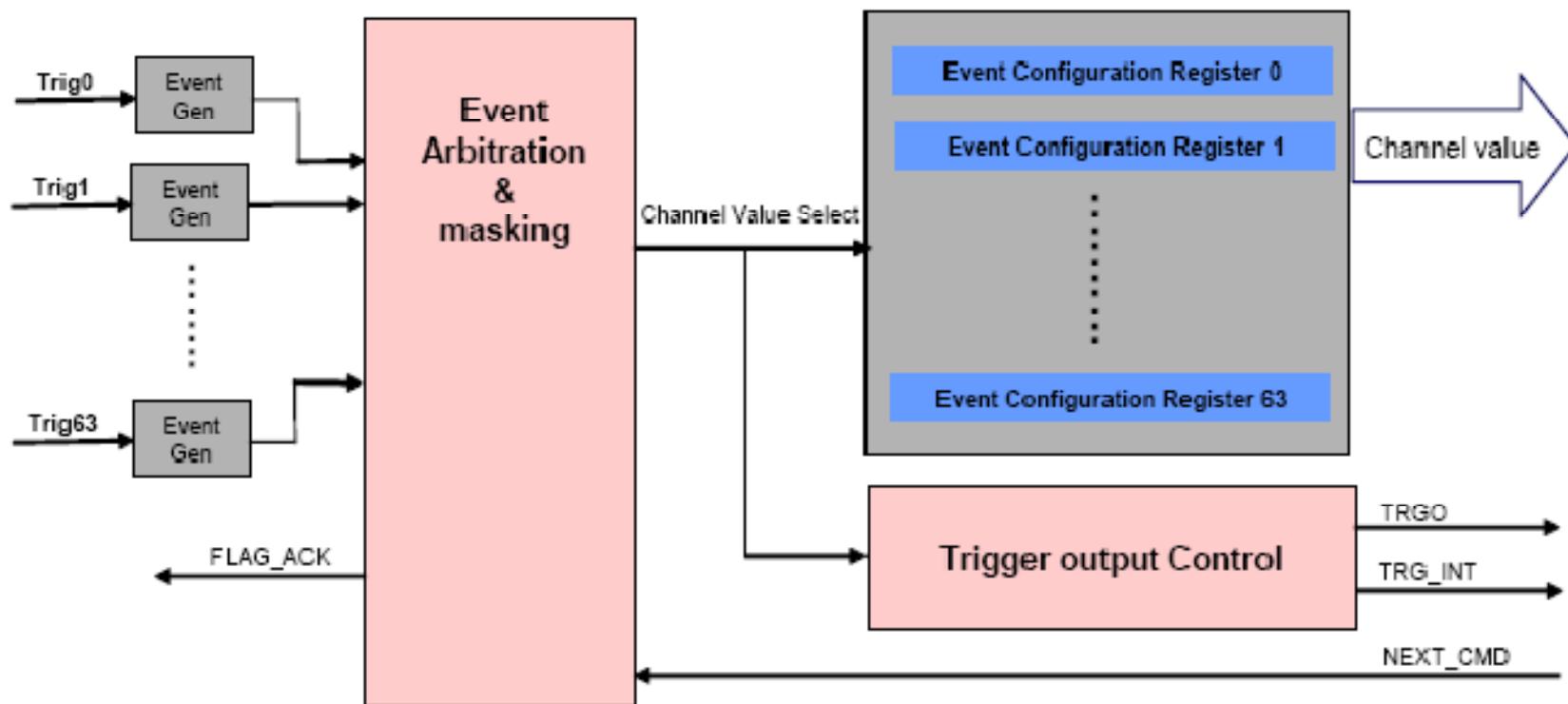
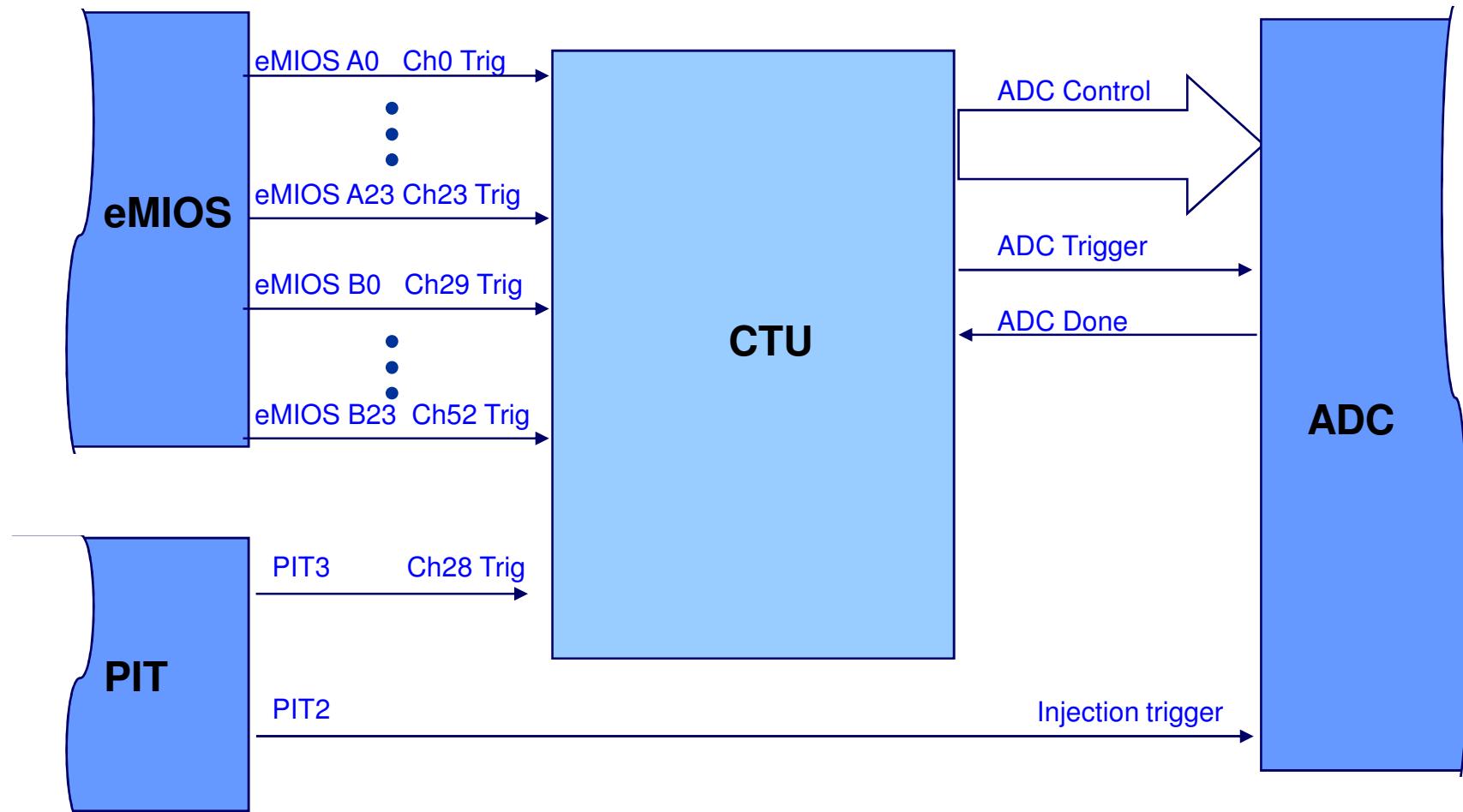
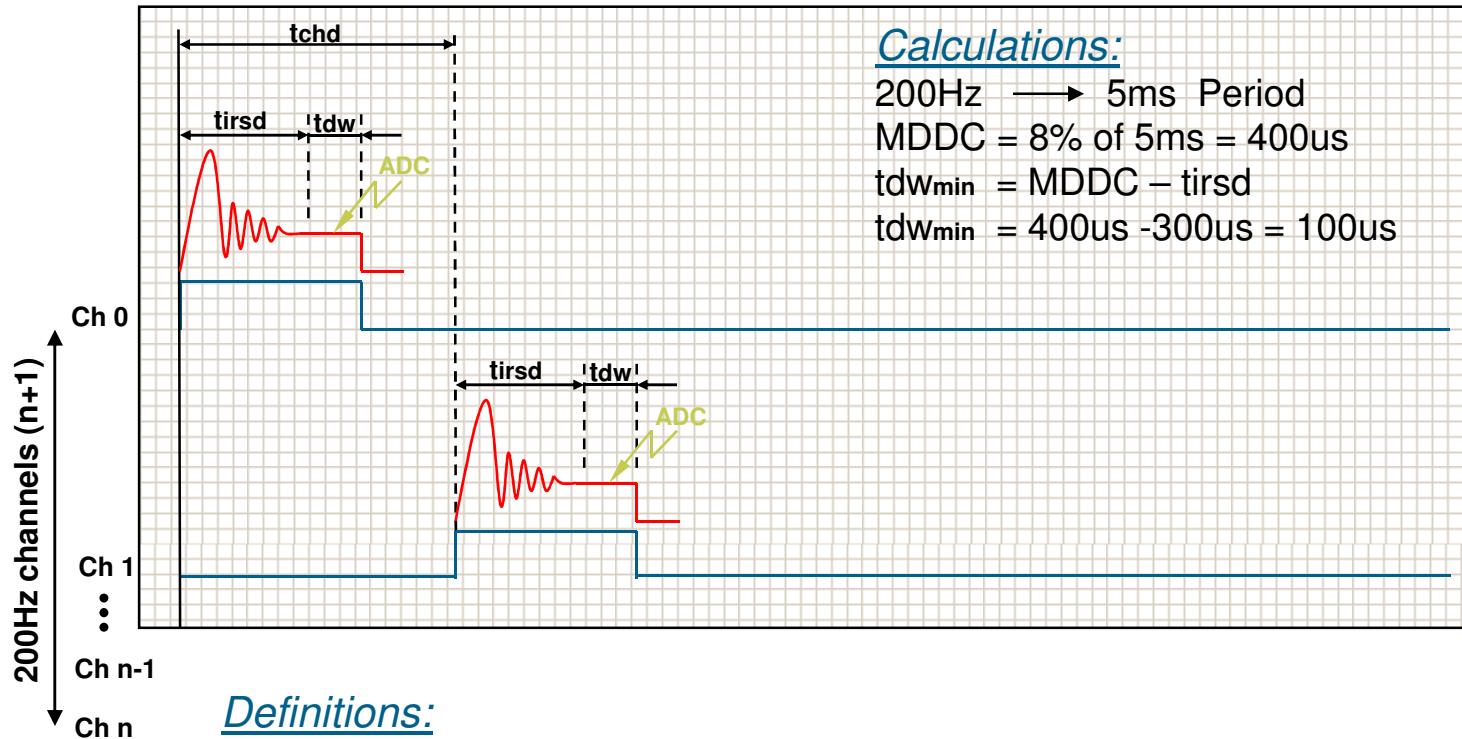


Figure 252. Cross Triggering Unit block diagram

MPC5602/3/4B eMIOS260/OPWMT Trigger event



PWM & Diagnosis Timing



Calculations:

200Hz → 5ms Period

MDDC = 8% of 5ms = 400us

tdw_{min} = MDDC - tirsd

tdw_{min} = 400us - 300us = 100us

Definitions:

PWM 200Hz (0.35% resolution, xybit)

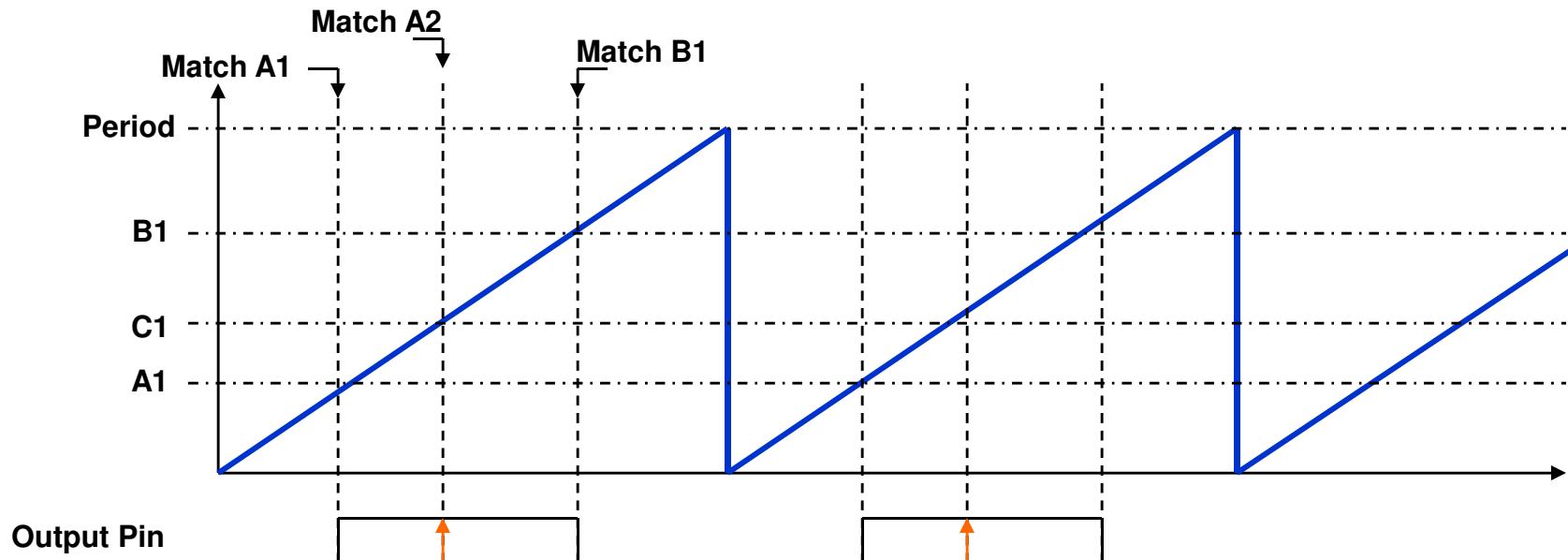
MDDC: 8% (*Min Diag Duty Cycle*: For Duty cycles <8% no diagnosis is required)

tchd: 1ms (*Channel Delay* to control inrush current as well as EMC on ECU level)

tirsd: 300us (*Inrush Delay* to ensure ADC measurement takes place once current is stable)

tdw_{min}: 100us (*Diagnosis Window* = Min. Diag. Duty Cycle - Inrush Delay)

eMIOS - OPWMT Mode



Period: the period of the PWM is defined by a Modulus Counter channel.

A1 Value: define the leading edge (or shift) of the PWM channel. Buffering is not needed as the value of the shift must not change on the fly.

B1 Value: define the trailing edge (or duty cycle) of the PWM channel

B2 Value: buffered value of trailing edge

B1 update: transfer from B2 to B1 takes place at A1 match

EDPOL: define the output polarity

A2 Value: define the sampling point for the analog diagnostic. It can be configured anywhere within the PWM period.

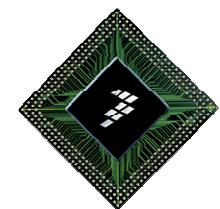
CTU features details:

- 64 timer events
- Each timer event can be assigned corresponding ADC channel
- Only one ADC conversion can be triggered at a time
- HW arbitration when simultaneous event occur
- Event priorities are HW defined
- Single cycle delayed trigger output. The trigger output is a combination of 64 (generic value) input flags/events connected to different timers in the system.
- Maskable interrupt generation whenever a trigger output is generated
- One event configuration register dedicated to each timer event allows to define the corresponding ADC channel
- Acknowledgment signal to eMIOS/PIT for clearing the flag
- Synchronization with ADC to avoid collision



Designing with Freescale

SPI



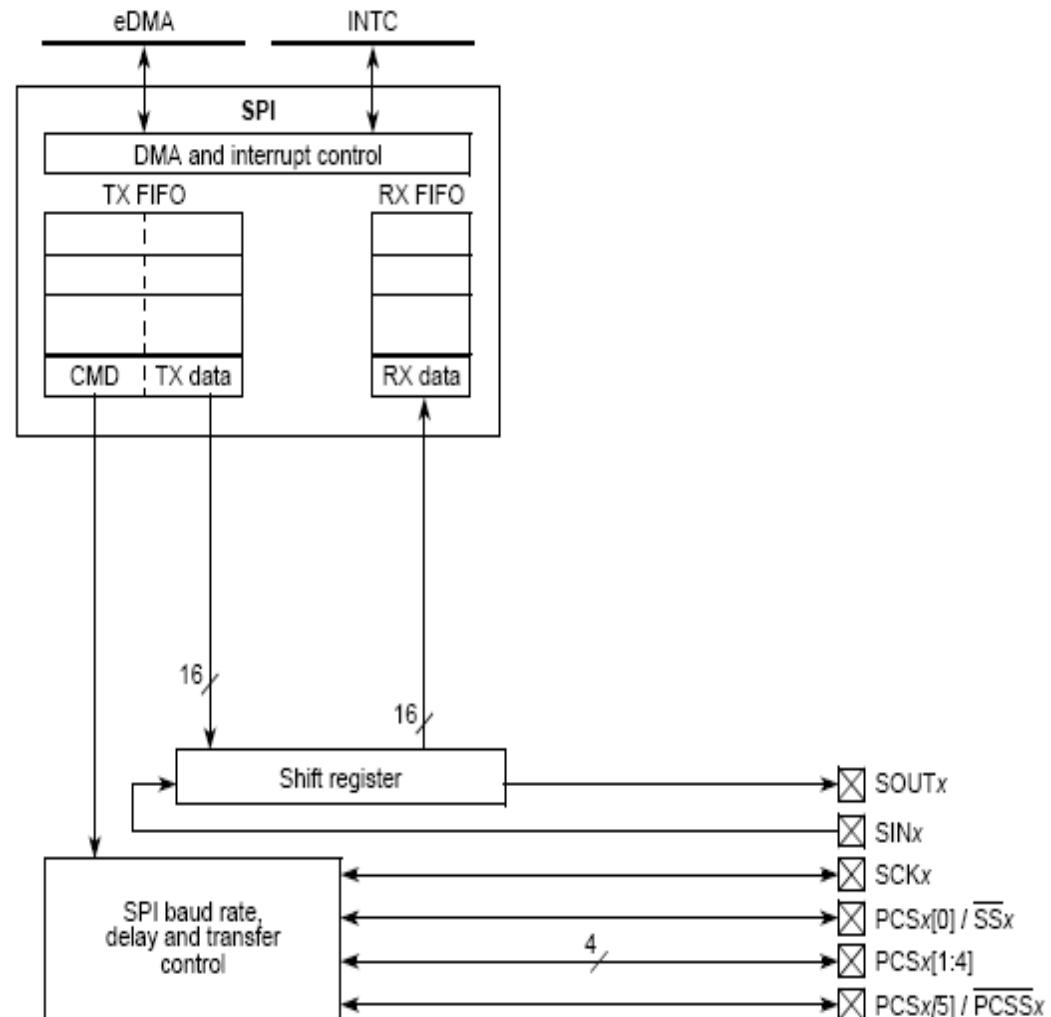
Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**™
semiconductor

Peripherals

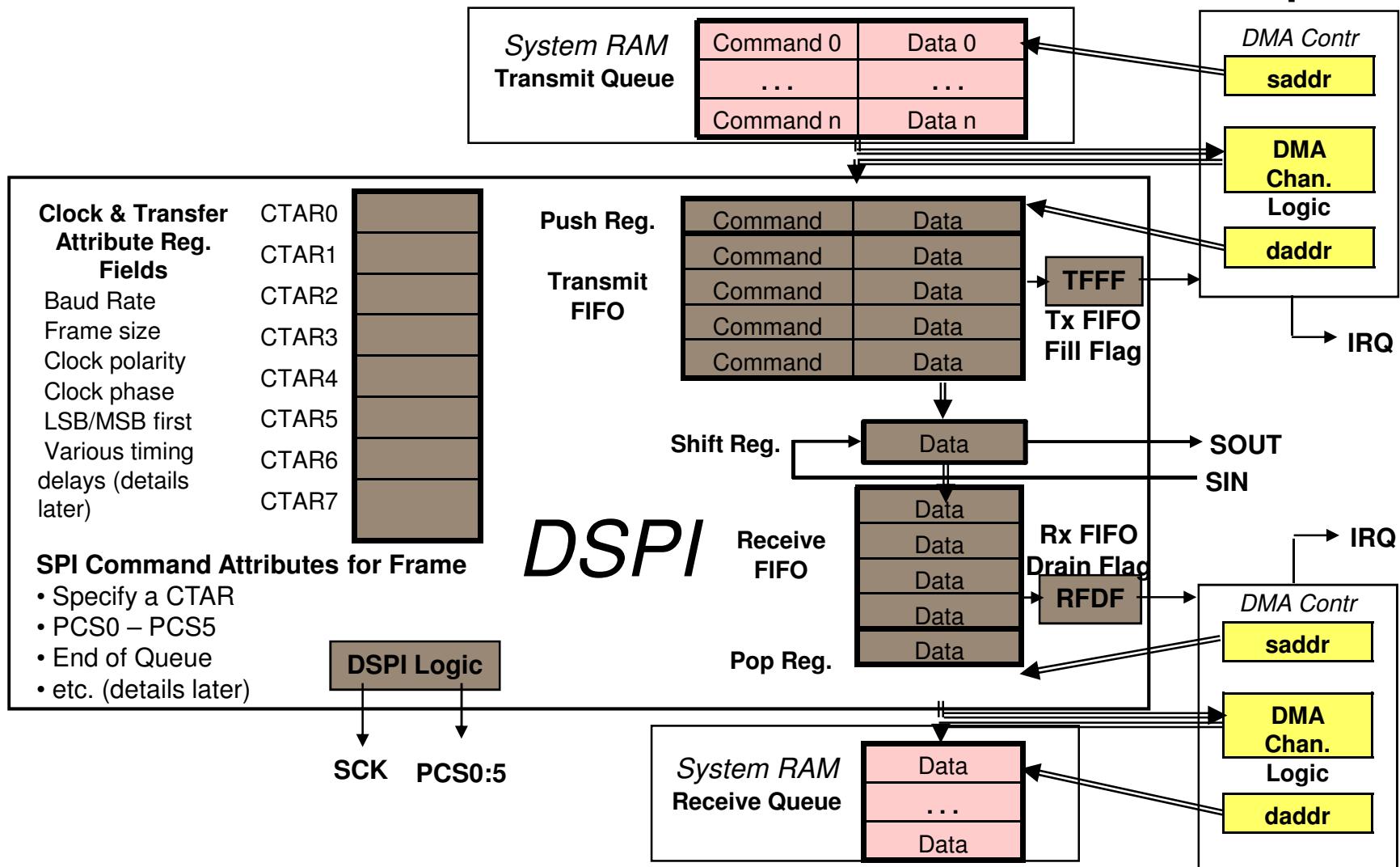
SPI Overview

- ▶ High speed, full duplex, three wire synchronous interface
- ▶ Master and slave modes supported
- ▶ Six Peripheral Chip Selects
 - Expandable to 64 with external demultiplexer
 - Deglitching support of up to 32 chip selects when external mux used
- ▶ SPI Queue support
 - Buffered transfers using 4 deep Tx FIFO and 4 deep Rx FIFO for regular SPI modules
 - 2 DMA requests for filling and emptying Tx and Rx FIFOs
 - FIFO visibility for debugging
- ▶ 6 Interrupt conditions



Peripherals

SPI Master Mode Operation



Peripherals

SPI Slave Mode Operation

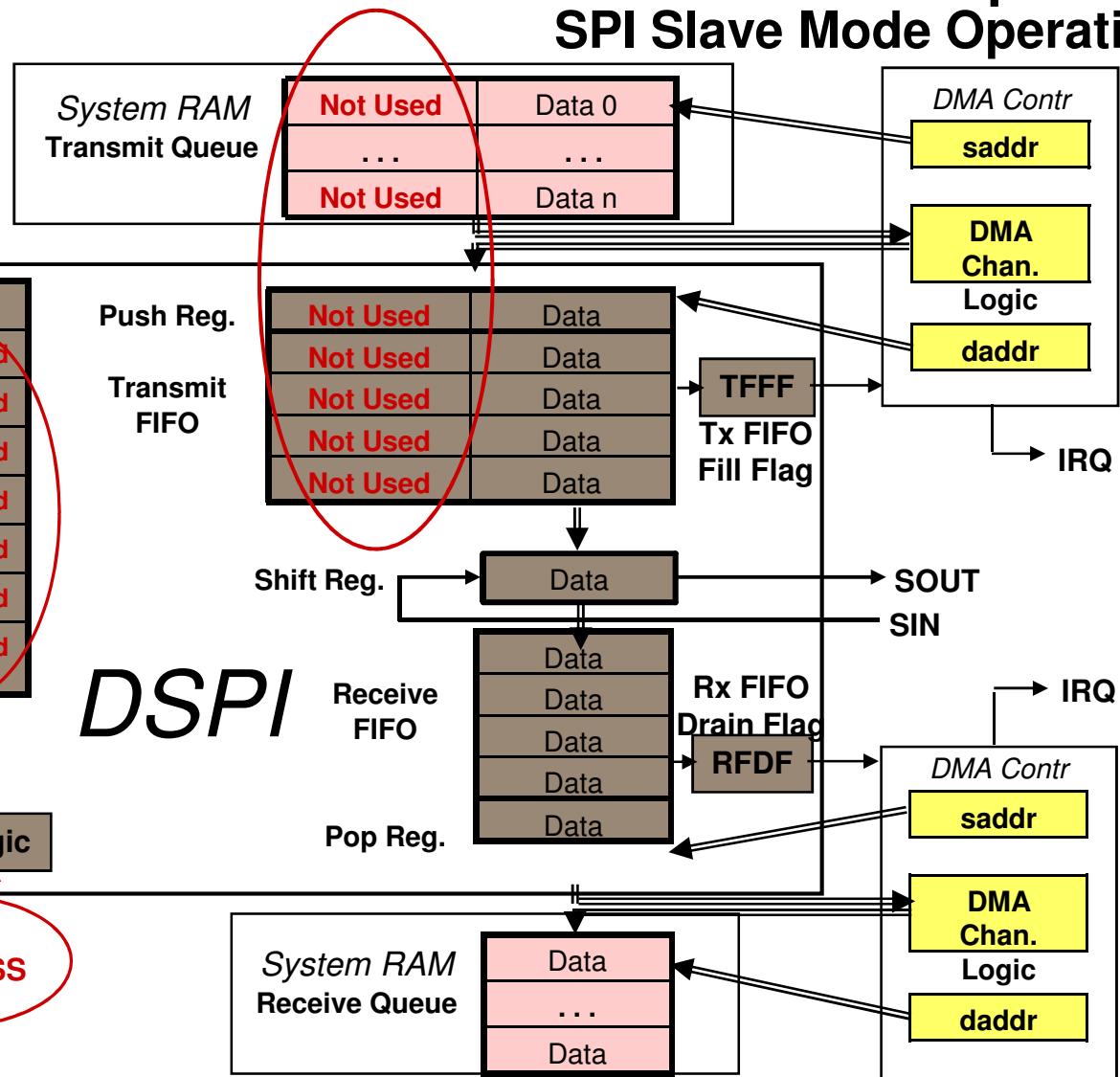
Slave Mode differences from Master Mode are circled.

Clock & Transfer Attribute Reg. Fields	CTAR0	CTAR1	CTAR2	CTAR3	CTAR4	CTAR5	CTAR6	CTAR7
Baud Rate	Not Used							
Frame size								
Clock polarity								
Clock phase								
LSB/MSB first								
Various timing delays (details later)								

- SPI Command Attributes for Frame**
- Specify a CTAR
 - PCS0 – PCS5
 - End of Queue
 - etc. (details later)

DSPI Logic

SCK SS



Peripherals

Automatic SPI message management

► eDMA

- ▶ Transfer based on descriptors in RAM; capability to link descriptors
- ▶ Transfers can be initiated by timers and then handled based on SPI interrupts
- ▶ Automatic data gathering and scattering

► DSPI

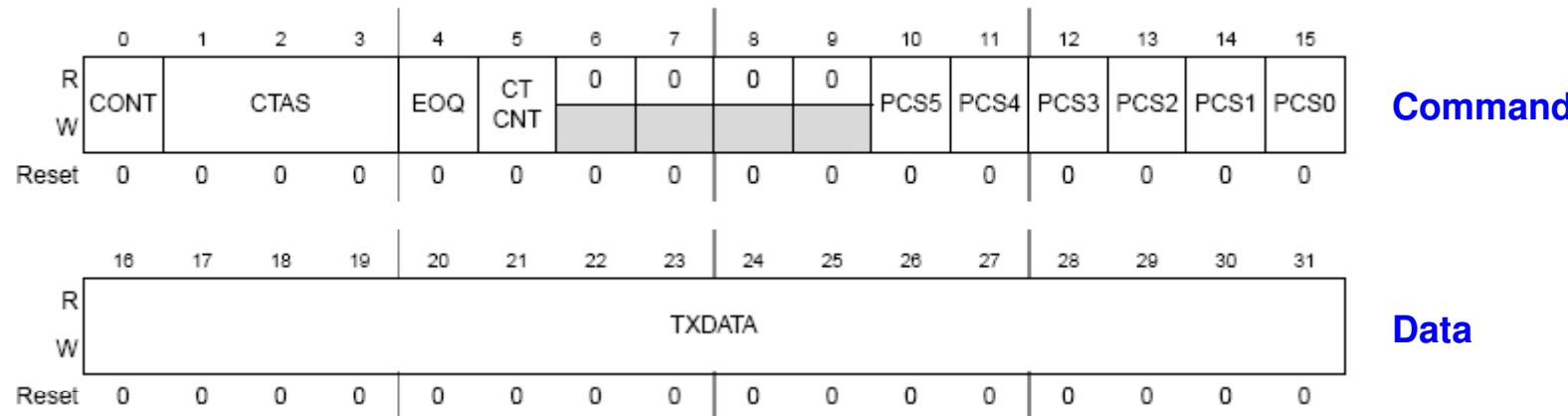
- ▶ Configurable SPI parameters per chip select
- ▶ Message based communication (automatic chip select activation/deactivation)
- ▶ DMA request signals (ex : transmit FIFO not full) allows to sustain data transmission

► eDMA + DSPI

- ▶ SPI Message scattering and message gathering from/to different message areas
- ▶ High speed SPI for communication (e.g. with Park Pilot ASIC)
- ▶ Automatic SPI PWM Generation
- ▶ **Automatic SPI message management without CPU load**

DSPI: Command and Data Word

- In master mode, to initiate a transmission, a 32-bit word is copied into the push register



- CONT** Continuous Peripheral Chip Select Enable (between transfers)
- CTAS** Clock and Transfer Attributes Select (Defines which CTAR to use)
- EOQ** Allows host to state that current transfer is last in Queue
- CTCNT** Clears SPI transfer Counter
- PSCn** Defines which peripheral chip select will be used for transfer
- TXData** Data to be transferred with the associated command word criteria

Freescale, the Freescale logo, Altivec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.



32-bit Module Configuration Register

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	MSTR	CONT_SCKE	DIS_RXF	DCONF	FRZ	MTFE	PCS_SE	ROOE	0	0	PCSI_S5	PCSI_S4	PCSI_S3	PCSI_S2	PCSI_S1	PCSI_S0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	MDIS	DIS_TXF	DIS_RXF	CLR_TXF	CLR_RXF	SMPL_PT		0	0	0	0	0	0	0	HALT
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

- ▶ **MSTR** Selects between master and slave modes
- ▶ **CONT_SCKE** Continuous SCLK enable sets SCK to run continuously
- ▶ **DCONF** Selects DSPI configuration (SPI, DSI or CSI)
- ▶ **FRZ** Freeze - Allows DSPI transfers to be halted (on next frame) in debug mode
- ▶ **MTFE** Modified Timing Format Enable (Delays sample until later in SCK period)
- ▶ **PCSSE** Peripheral Chip Select Strobe Enable allows PCS to be used as active low PCS strobe
- ▶ **ROOE** Receive FIFO Overflow Overwrite Enable. If FIFO is full determines whether to ignore incoming data or overwrite existing data in FIFO
- ▶ **PCSI_n** Peripheral Chip Select Inactive State (low or high)

32-bit Module Configuration Register

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	MSTR	INT _W SCK _I	DCONF		FRZ	MTFE	PCS SE	ROOE	0	0	PCSI S5	PCSI S4	PCSI S3	PCSI S2	PCSI S1	PCSI S0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	MDIS	DIS_ TXF	DIS_ RXF	CLR_ TXF	CLR_ RXF	SMPL_PT		0	0	0	0	0	0	0	HALT
W	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

- ▶ **MDIS** Module Disable. Stops clock to non memory mapped logic resulting in power saving
- ▶ **DIS_TXF** Disable Transmit FIFO (and operate as simplified double buffered SPI)
- ▶ **DIS_RXF** Disable RX FIFO
- ▶ **CLR_TXF** Clear TX FIFO – Flushes TX FIFO
- ▶ **CLR_RXF** Clear RX FIFO
- ▶ **SIMPL_PT** Sample Point – Defines when DSPI master samples SIN (serial Data In) to allow delayed sampling.
- ▶ **HALT** – Provides mechanism for software to start and stop DSPI transfers

Note – Only the HALT and MDIS bits can be written while DSPI is running

DSPI: Status Register 1

► 32-bit Status Register

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TCF	TXRXS	0	EOQF	TFUF	0	TFFF	0	0	0	0	0	RFOF	0	RFDF	0
W	w1c			w1c	w1c		w1c						w1c		w1c	
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TXCTR				TXNXTPTR				RXCTR				POPNXTPTR			
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCF Transmit complete flag (all bits in current transfer have been shifted out)

TXRXS TX and RX status – Indicates if TX and RX operations are enabled or disabled

EOQF End of Queue Flag – Indicates current transmission is last entry in queue

TFUF Transmit FIFO Underflow Flag – Slave mode only and indicates transmit FIFO is empty

TFFF Transmit FIFO Fill Flag – Indicates there is room in TX FIFO

RFOF Receive FIFO Overflow Flag – Overflow condition has occurred and data has been lost

RFDF Receive FIF Drain Flag – RX FIFO has data in it and can be drained.

DSPI: Status Register 2

► 32-bit Status Register

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TCF	TXRXS	0	EOQF	TFUF	0	TFFF	0	0	0	0	0	RFOF	0	RFDF	0
W	w1c			w1c	w1c		w1c						w1c		w1c	
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TXCTR				TXNXTPTR				RXCTR				POPNXTPTR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXCTR TX FIFO Counter – Indicated number of valid entries **currently** in FIFO

- Incremented when new entry pushed onto push register, decremented when entry transmitted

TXNXTPTR Transmit Next Pointer – Points to TX FIFO entry that will be transmitted next

RXCTR RX FIFO counter – Indicates number of entries **currently** in RX FIFO

- Incremented when new entry received, decremented when entry read from POP register

POPNXTPTR Pop Next Pointer – Points to RX FIFO entry that will be read next

DSPI: Queue Operation

- ▶ **DSPI_TCR** contains number of SPI transfers made
- ▶ Cleared by **CTCNT** bit in command word
- ▶ **EOQ** bit in command word allows an end of queue flag to be set, **EOQF** in the status register
- ▶ **EOQF_RE** allows the EOQF to generate an interrupt request when the EOQF flag is set.

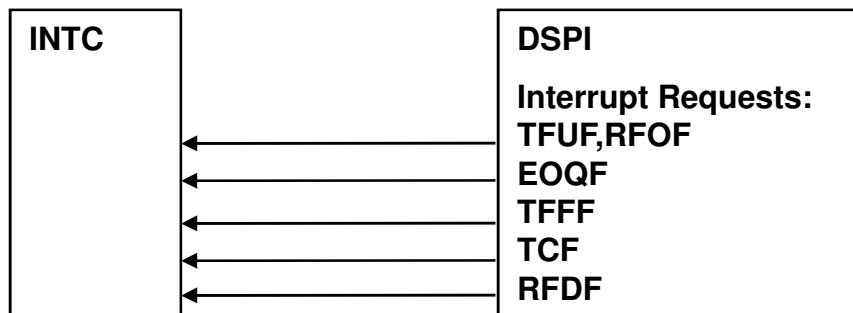
SPI_TCNT																
R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
W	DSPI_TCR – Timer Count Register															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	18	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
W	TCF	TXRXS	0	EOQF	TFUF	0	FFFF	0	0	0	0	0	RFOF	0	RFDF	0
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
DSPI_SR – Status Register																
R	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
TXCTR	TXNXTPTR				RXCTR				POPNXTPTR							
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
W	TCF_ RE	0	0	EOQF_ RE	TFUF_ RE	0	FFFF_ RE	FFFF_ DIRS	0	0	0	0	RFOF_ RE	0	RFDF_ RE	RFDF_ DIRS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DSPI_RSER – Request Select and Enable Register																
R	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DSPI: Basic process for a SPI transfer

- ▶ Initialize **MCR** (following shows critical fields)
 - **MSTR** = 1; for Master Mode
 - **PCSIS_x** = x; Define active state of chip selects (high or low)
 - **MDIS** = 0; Enable DSPI module (enables DSPI Clocks).
- ▶ Define CTAR registers - Clock and Transfer Attributes Register
- ▶ Setup Interrupt / DMA registers [DSPI_RSER; DSPI DMA / Interrupt Request Select and Enable Register]
- ▶ Enable DSPI by clearing **HALT** bit in **DSPI_MCR**
- ▶ To initiate a SPI transfer, write DATA and Command word into DSPI Push Register **DSPI_PUSHR**
- ▶ Received data is in DSPI Pop register **DSPI_POPR**

DSPI: IRQ/DMA Request Sources

Transmit	Receive
TFFF: Tx FIFO not full – “Fill Flag” (IRQ or DMA)	RFDF: Rx FIFO not empty - “Drain Flag” (IRQ or DMA)
TCF: Transfer of current frame complete (IRQ)	RFOF: Frame received while FIFO full (IRQ)
EOQF: End of queue reached (IRQ)	
TFUF: Attempt to transmit with empty FIFO (IRQ)	



Using DMA for SPI – (1)

Basic Example: Transferring 4 bytes of data (0x01 23 45 67) though SPI

Transmit data interleaved with COMMAND

COMMAND defines SPI Timing, baudrate, and CS to be used

Tx Data in SRAM

ADDRESS	DATA
0x4000 0000	xxxx
0x4000 0002	0123
0x4000 0004	xxxx
0x4000 0006	4567
0x4000 0008	zzzz
0x4000 000A	zzzz
0x4000 000C	zzzz
0x4000 000E	zzzz

DMA ch for SPI Tx

Source: 0x4000 0000
Destination: SPI DATA
Size: 32bit
Source offset: 4 bytes
Desti. offset: zero
Minor counter: 8
Major counter: 0



Data transfer can be started by

- Software
- Tx Empty
- PIT

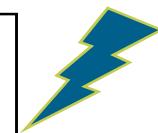
DSPI PUSHR
DSPI POPR

Rx Data in SRAM

ADDRESS	DATA
0x4000 0010	FEDC
0x4000 0012	BA98
0x4000 0014	yyyy
0x4000 0016	yyyy
0x4000 0018	yyyy
0x4000 001A	yyyy
0x4000 001C	yyyy
0x4000 001E	yyyy

DMA ch for SPI Rx

Source: SPI DATA
Destination: 0x4000 0010
Size: 16bit
Source offset: zero
Desti. offset: 2 bytes
Minor counter: 4
Major counter: 0



Data transfer started by
• Successful data received

At the end of Major loop, Interrupt can be generated to CPU

Appendix: Circuit Notes

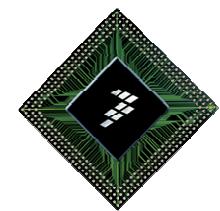
- ▶ 1) JTAG interface:
 - ▶ PULLUPs recommended for JTAG input pins
 - ▶ TDI - 10k PULLUP to VDD_HV
 - ▶ TCK - 10k PULLUP to VDD_HV
 - ▶ TMS - 10K PULLUP to VDD_HV
 - ▶
- ▶ 2) Nexus interface
 - ▶ Mictor M38x connector has 38 pins. On page there is a 40-pin connector, not sure what the reason for this is ?
 - ▶ Assuming Mictor 38x connector will be used:
 - ▶ -> pin 12 VREF, please use direct connection to Bolero VDD_HV (without 1K PULLUP)
 - ▶
 - ▶ -> Bolero does not need the connection of
 - ▶ DO_WD_DISABLE_NEXUS
 - ▶ from pin 27 of mictor connector.
 - ▶
 - ▶ -> mictor connector pin37 VALTREF should be connected via PULLUP to to Bolero JTAG I/O supply VDD_HV
 - ▶ -> mictor connector pin36 MSEO1 via 10k PULLDOWN to GND
 - ▶ -> mictor connector pin38 MSEO0 should be connected to Bolero MSEO pin
 - ▶
 - ▶ 3) Power Supply:
 - ▶ -> VDD_BV decoupling. Current spec. is 400nF min (CDEC1 value in datasheet). We plan to change this to 100nF min for 5V operation in next datasheet release (400nF will most likely apply to 3.3V operation only). Please note that due to cap. tolerance, temperature variation and aging the nominal capacitance can decrease by up to 35% during lifetime.
 - ▶ Another reference: 2009-08-06 Bolero HW-Design Guidelines Rev0.5_VF

- ▶ Review of Key Points
- ▶ Flexible SPI
 - 4 to 16 bits data per frame
 - Timing attributes, including baud rate, on a per frame basis
 - DMA (if implemented) for automatic filling and draining SPI transmit and receive FIFOs
- ▶ Example DSPI: SPI to SPI



Designing with Freescale

Debugging, Tools



Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**™
semiconductor

- Developed in house
- Developed offshore
- 3rd Party ownership

SW and Tools Market Coverage

Autosar environment (SW & Tools)

Autosar MCAL 2.1

Autosar OS

Autosar MCAL 3.0

Autosar BSW

Autosar OS

Tresos Auto Core

Software Drivers

FlexRay Drivers

Flash Drivers

Vector CAN Driver

LIN 2.1 Drivers

FEE Drivers

Sound Generation Drv

J2602 Drivers

MC Complex Drv

Stepper Motor Drivers

Software Libraries

Graphic Library

DSP lib

General Motor Control

eTPU+ Libs

Tools

RApID Init

Simulink Support

RApID Toolbox

System Simulation

Compiler & Debugger

GHS Compiler

Lauterbach Debugger

PLS Debugger

GNU Compiler

iSystem Debugger

P&E Debugger

GHS Time Machine

Cosmic

eTPU Compiler

Wind River Compiler

CodeWarrior Compiler

Starterkit and evaluation boards

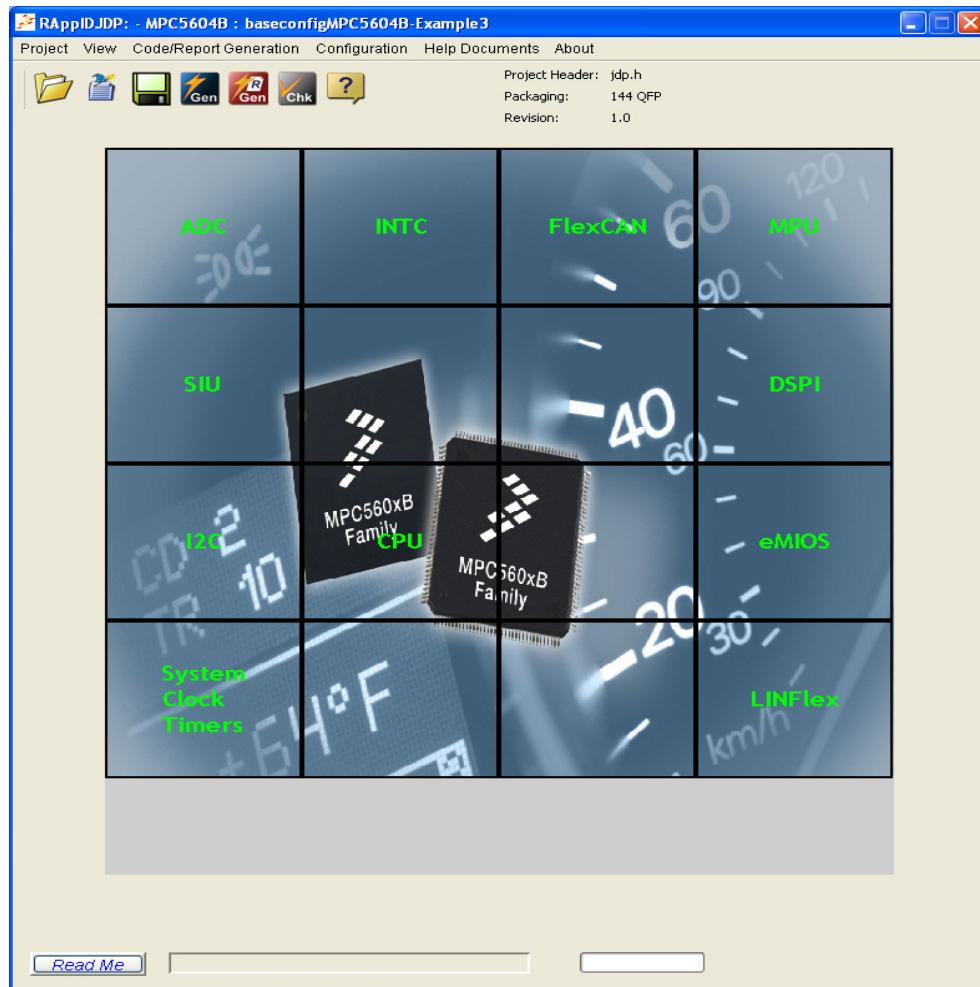
EVB

Mini Modules

Adapters

Debug, Software & Tools

RAppID for the 560xB Family example



- ▶ Supports all MPC560xB family members
- ▶ Enhancements:
 - ▶ Wizard workflow to allocate pins to functions (peripherals)
 - ▶ Initialization code for single/dual core startup from CRT0 upward
 - ▶ Interrupt and Exception handler software frame work generation
 - ▶ Ability to define section map and place code into any section desired

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

 **freescale**™ semiconductor

Debug, Software & Tools

Pin Allocation Wizard - Screenshot

Pin Allocation : baseconfig *

Pin Allocation Wizard

PAD	PAD	AN8	PAD	PA10 I	PK0 I	TXDG	PB2 I	PB6 I	PCS8 I	RXD H	eMIO8 I	PC7 I	PC 10 I	PAD	PAD				
PAD	PAD	PAD	PAD	PA12 I	PK1 I	RXD G	PB3 I	PB7 I	PCS9 I	eMIO9 I	PC4 I	PC8 I	PC 11 I	PAD	PC 12 I				
AN7	AN6	PAD	AN9	AN11	PA15 I	PCS5 I	PB4 I	PB8 I	TXD H	eMIO11 I	PC5 I	PC9 I	PAD	PCS15 I	PCS14 I				
AN5	AN4	AN3	PAD	AN10	PA14 I	PB1 I	PB5 I	PCS14 I	PAD	PC2 I	PC6 I	PAD	PCS13 I	CNTXA	CNRXA				
AN2	AN1	AN0	PAD	MPC5516G in 208 BGA												PAD	CNRXB	PD0 I	PD4 I
PA10 I	PH12 I	PH11 I	PH10 I													PE5 I	TXDA	RXDA	RXD B
PA15 I	PH9 I	PA14 I	PA13 I													TXDB	PCS82 I	PD11 I	PAD
PH8 I	PA12 I	PA11 I	PAD													PE7 I	PCS80 O	SCKB O	PE8 I
RXD F	TxD F	RXD E	PA10 I													PE9 I	SOUTB	PE11 I	PE10 I
PA9 I	PA8 I	PA4 I	PA3 I													PE12 I	PD15 I	PAD	PCS12 I
PH2 I	PH1 I	PH0 I	PAD													PE13 I	PCS11 I	PCS80 O	PE14 I
PG15 I	PG14 I	PG13 I	PG12 I													PE9 I	PE15 I	SINA	PAD
PG11 I	PG10 I	PG9 I	PAD	PAD	PF15 I	PF12 I	PF9 I	PAD	AD2 I	AD0 I	PF0 I	PAD	PE4 I	PAD	PAD				
PAD	PG8 I	PAD	PG3 I	PG0 I	PF14 I	PF11 I	AD13 I	PA6 I	AD4 I	AD1 I	PF1 I	PE8 I	PAD	PAD	PAD				
PG7 I	PAD	PG5 I	PG2 I	PH15 I	RXD D	PF10 I	PA7 I	AD11 I	AD0 I	PAD	PF2 I	PAD	PAD	PAD	PAD				
PAD	PG6 I	PG4 I	PG1 I	PH14 I	PAD	PF9 I	AD12 I	ADS I	AD10 I	PAD	AD9 I	PAD	PAD	PAD	PAD				

Part and Package

Part: MPC5516G Package: 208 BGA

DSPI	EBI	ECADC	FlexCAN	FlexRay	GPIO	I2C	MISC	eMIO8	eSCI
Functions		Input				Output			
I2C Clock		<input checked="" type="checkbox"/> PD8 <input type="checkbox"/> PH0				<input type="checkbox"/> PD8 <input type="checkbox"/> PH0			
I2C Data		<input type="checkbox"/> PD9 <input type="checkbox"/> PH1				<input type="checkbox"/> PD9 <input type="checkbox"/> PH1			

Address/Data 0 PAD PJ0 Allocated as Input
Address/Data 1 PAD PJ1 Allocated as Input
Address/Data 2 PAD PJ2 Allocated as Input
Address/Data 3 PAD PJ3 Allocated as Input
Address/Data 4 PAD PJ4 Allocated as Input
Address/Data 5 PAD PJ5 Allocated as Input
Address/Data 9 PAD PF3 Allocated as Input
Address/Data 10 PAD PF4 Allocated as Input
Address/Data 11 PAD PF5 Allocated as Input
Address/Data 12 PAD PF6 Allocated as Input
Address/Data 13 PAD PF7 Allocated as Input
EQADC AN 00 PAD PA0 Allocated as Input
EQADC AN 01 PAD PA1 Allocated as Input
EQADC AN 02 PAD PA2 Allocated as Input
EQADC AN 03 PAD PA3 Allocated as Input
EQADC AN 04 PAD PA4 Allocated as Input
EQADC AN 05 PAD PA5 Allocated as Input
EQADC AN 06 PAD PA6 Allocated as Input
EQADC AN 07 PAD PA7 Allocated as Input
EQADC AN 08 PAD PA8 Allocated as Input
EQADC AN 09 PAD PA9 Allocated as Input
EQADC AN 10 PAD PA10 Allocated as Input
EQADC AN 11 PAD PA11 Allocated as Input
CAN_A Rx PAD PD1 Allocated as Input
CAN_A Tx PAD PD0 Allocated as Output
CAN_B Rx PAD PD2 Allocated as Input
Channel0 PAD PC0 Allocated as Input
Channel1 PAD PC1 Allocated as Input
Channel3 PAD PC3 Allocated as Input
eSCL_A Rx PAD PD7 Allocated as Input
eSCL_B Rx PAD PD9 Allocated as Input
eSCL_B Tx PAD PD8 Allocated as Output
eSCL_E Rx PAD PH5 Allocated as Input
eSCL_D Rx PAD PF13 Allocated as Input
eSCL_A Tx PAD PD6 Allocated as Output
eSCL_F Tx PAD PH6 Allocated as Output
eSCL_F Rx PAD PH7 Allocated as Input
eSCL_G Rx PAD PB13 Allocated as Input
eSCL_G Tx PAD PB12 Allocated as Output
eSCL_H Tx PAD PB14 Allocated as Output
eSCL_H Rx PAD PB15 Allocated as Input

Previous Next Exit Wizard

Evaluation boards



 **freescale™**
Alliance Member

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.

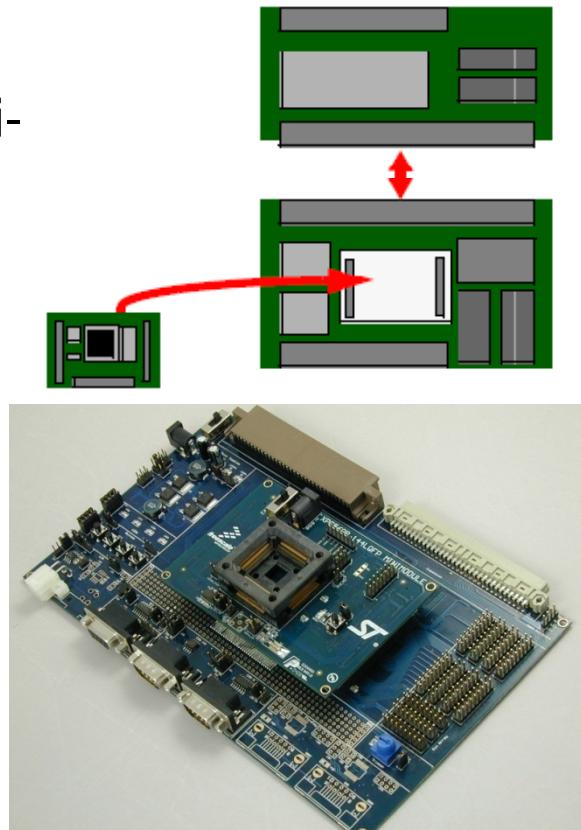
 **freescale™**
semiconductor

Debug, Software & Tools

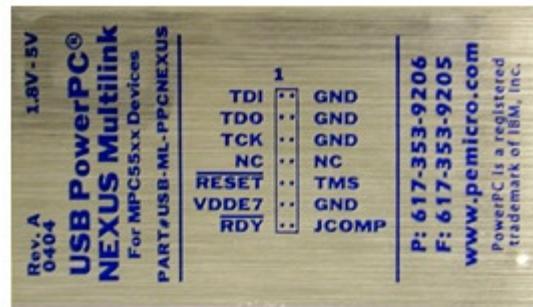
XPC56xxEVB - Evaluation System

- ▶ XPC Evaluation board for XPC56xx devices. Allow to evaluate and develop the whole range of XPC devices.

- Full modular design: motherboard, mini-module for each device, standardized connectors
- Standard communication transceivers and connectors
- User's buttons, jumpers and LED's
- Device mini-module connector



Nexus Class 0 – JTAG Interface for MPC55xx/56xx



- ▶ The USB Multi link is a Nexus Class 0 = JTAG Interface
- ▶ MULTILINK Interface is delivered with XPC560BKIT144/176/208 Starter Kits.
- ▶ It can also be purchased separately.

Development Tools – An Existing Ecosystem

	Compilers	Debuggers	Simulators	Eval Boards	Initialization Tools	Modeling and Code
CodeWarrior 	✓ (v2.2)	works w/ any debugger		✓		
Green Hills 	✓	✓	✓			
Wind River 	✓	✓				
GNU 	✓					
Lauterbach 		✓	✓			
iSystem 		✓		✓		
P&E Micro 		✓				
RAppID Init 					✓	
dSpace 						✓
MathWorks 						✓

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions Logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SmartMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2010.



