Concerto Microcontrollers

Check for Samples: F28M36P63C

1 F28M36x (Concerto™) MCUs

1.1 Features

- Master Subsystem — ARM® Cortex™-M3
  - 125 MHz
  - Cortex™-M3 Core Hardware Logic Built-in Self Test
  - Embedded Memory
    - Up to 1MB Flash (ECC)
    - Up to 128KB RAM (ECC or Parity)
    - Up to 64KB Shared RAM
    - 2KB IPC Message RAM
  - 5 Universal Asynchronous Receiver/Transmitters (UARTs)
  - 4 Synchronous Serial Interfaces (SSIs)/ Serial Peripheral Interface (SPI)
  - 2 Inter-integrated Circuits (I2Cs)
  - Universal Serial Bus On-the-Go (USB-OTG) + PHY
  - 10/100 ENET 1588 MII
  - 2 Controller Area Networks (CANs)
  - 32-Channel Direct Memory Access (μDMA)
  - Dual Security Zones (128-Bit Password per Zone)
  - External Peripheral Interface (EPI)
  - Micro Cyclic Redundancy Check (μCRC) Module
  - 4 General-Purpose Timers
  - 2 Watchdog Timer Modules
  - Endianness: Little Endian
- Clocking
  - On-chip Crystal Oscillator/External Clock Input
  - Dynamic PLL Ratio Changes Supported
- 1.2-V Digital, 1.8-V Analog, 3.3-V I/O Design

- Interprocessor Communications (IPC)
  - 32 Handshaking Channels
  - 4 Channels Generate IPC Interrupts
  - Can be Used to Coordinate Transfer of Data Through IPC Message RAMs

- Up to 142 Individually Programmable, Multiplexed GPIO Pins
  - Glitch-free I/Os

- Control Subsystem — TMS320C28x™ 32-Bit CPU
  - 150 MHz
  - C28x Core Hardware Logic Built-in Self Test
  - Embedded Memory
    - Up to 512KB Flash (ECC)
    - Up to 36KB RAM (ECC or Parity)
    - Up to 64KB Shared RAM
    - 2KB IPC Message RAM
  - IEEE-754 Single-Precision Floating-Point Unit (FPU)
  - Viterbi, Complex Math, CRC Unit (VCU)
  - Serial Communications Interface (SCI)
  - Serial Peripheral Interface (SPI)
  - Inter-Integrated Circuit (I2C)
  - 6-Channel Direct Memory Access (DMA)
  - 12 Enhanced Pulse Width Modulator (ePWM) Modules
    - 24 Outputs (16 High-Resolution)
  - 6 32-Bit Enhanced Capture (eCAP) Modules
  - 3 32-Bit Enhanced Quadrature Encoder (eQEP) Modules
  - Multichannel Buffered Serial Port (McBSP)
  - External Peripheral Interface (EPI)
  - One Security Zone (128-Bit Password)
  - 3 32-Bit Timers
  - Endianness: Little Endian

- Analog Subsystem
  - Dual 12-Bit Analog-to-Digital Converters (ADCs)
  - Up to 2.88 MSPS
  - Up to 24 Channels
  - 4 Sample-and-Hold (S/H) Circuits
  - Up to 6 Comparators With 10-Bit Digital-to-Analog Converter (DAC)

- Package
  - 289-Ball ZWT Plastic Ball Grid Array (PBGA)

1.2 Description

The Concerto™ family is a multi-core system-on-chip microcontroller (MCU) with independent communication and real-time control subsystems. The F28M36x is the second series in the Concerto family.

The communications subsystem is based on the industry-standard 32-bit ARM® Cortex™-M3 CPU and features a wide variety of communication peripherals, including Ethernet 1588, USB OTG with PHY, CAN, UART, SSI, I2C, and an external interface.

The real-time control subsystem is based on TI's industry-leading proprietary 32-bit C28x™ Floating-Point CPU and features the most flexible and high-precision control peripherals, including ePWMs with fault protection, and encoders and captures—all as implemented by TI's C2000™ Piccolo™ and Delfino™ families. In addition, the C28-CPU has been enhanced with the addition of the Viterbi, Complex Math, CRC Unit (VCU) instruction accelerator that implements efficient Viterbi, Complex Arithmetic, 16-bit FFTs and CRC algorithms.

A high-speed analog subsystem and supplementary RAM memory is shared, along with on-chip voltage regulation and redundant clocking circuitry. Safety considerations also include Error Correction Code (ECC), Parity, and Code Secure Memory, as well as documentation to assist with system-level industrial safety certification.

1.3 Functional Block Diagram

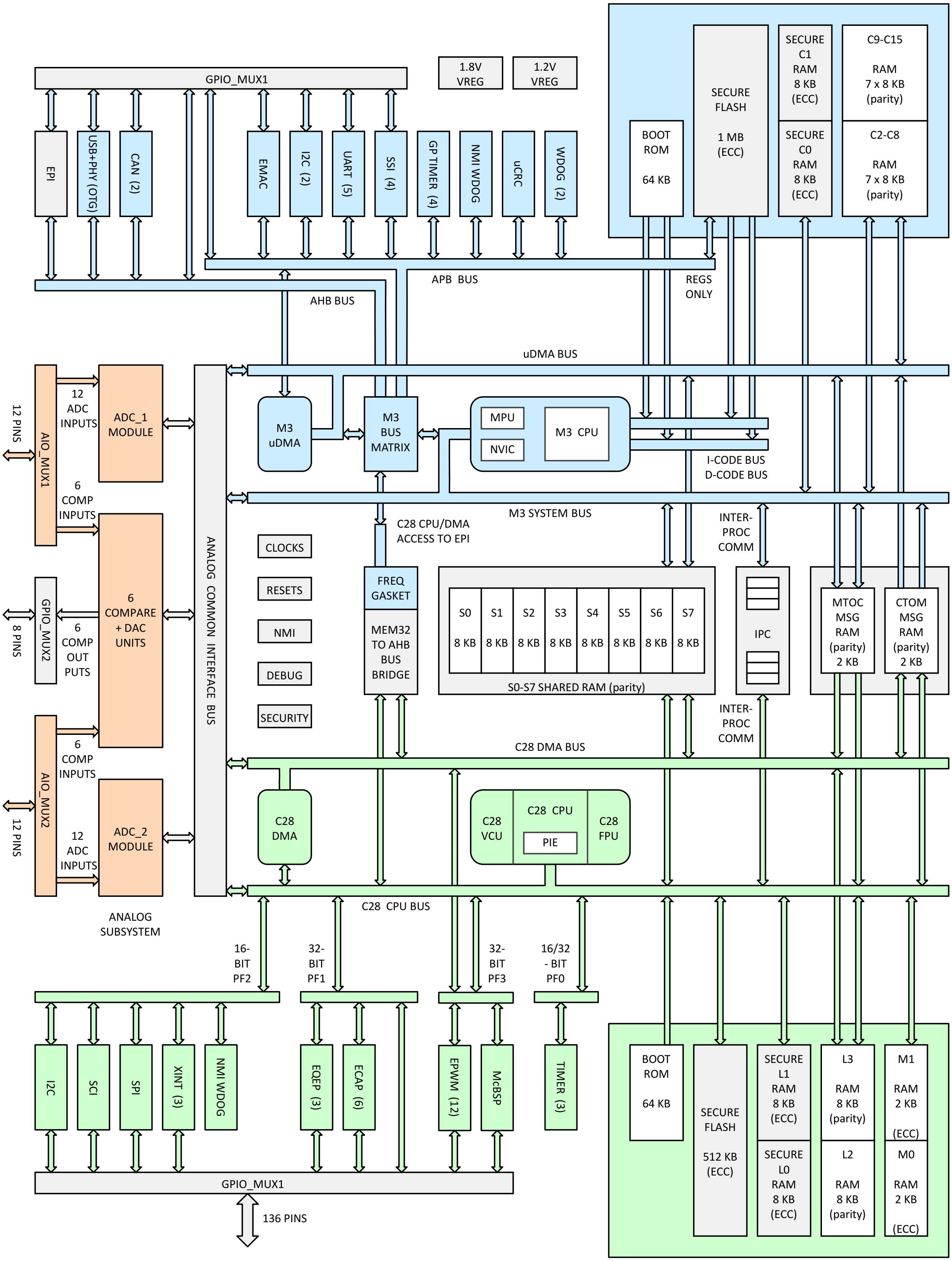


Figure 1-1. Functional Block Diagram

ADVANCE INFORMATION

2 Device Overview

The Concerto™ microcontroller (MCU) comprises three subsystems: the Master Subsystem, the Control Subsystem, and the Analog Subsystem. While the Master and Control Subsystem each have dedicated local memories and peripherals, they can also share data and events through shared memories and peripherals. The Analog Subsystem has two ADC converters and six Analog Comparators. Both the Master and Control Subsystems access the Analog Subsystem through the Analog Common Interface Bus (ACIB). The NMI Blocks force communication of critical events to the Master and Control Subsystem processors and their Watchdog Timers. The Reset Block responds to Watchdog Timer NMI Reset, External Reset, and other events to initialize subsystem processors and the rest of the chip to a known state. The Clocking Blocks support multiple low-power modes where clocks to the processors and peripherals can be slowed down or stopped in order to manage power consumption.

NOTE

Throughout this document, the Master Subsystem is denoted by the color "blue"; the Control Subsystem is denoted by the color "green"; and the Analog Subsystem is denoted by the color "orange".

2.1 Device Characteristics

Table 2-1 lists the features of the F28M36x devices.

Table 2-1. Hardware Features

FEATURE	TYPE(1)	P63C	P53C	H53C	H53B	H33C	H33B
Master Subsystem — ARM® Cortex™-M3							
Speed (MHz)(2)	–	125	125	100	100	100	100
Flash (KB)	–	1024	512	512	512	512	512
RAM ECC (KB)	–	16	16	16	16	16	16
RAM Parity (KB)	–	112	112	112	112	112	112
IPC Message RAM Parity (KB)	–	2	2	2	2	2	2
Security Zones	–	2	2	2	2	2	2
10/100 ENET 1588 MII	0	Yes	Yes	Yes	No	Yes	No
USB OTG FS	0	Yes	Yes	Yes	No	Yes	No
Synchronous Serial Interface (SSI)/ Serial Peripheral Interface (SPI)	0	4	4	4	4	4	4
Universal Asynchronous Receiver/Transmitter (UART)	0	5	5	5	5	5	5
Inter-integrated circuit (I2C)	0	2	2	2	2	2	2
Controller Area Network (CAN)	0	2	2	2	2	2	2
Direct Memory Access (µDMA)	0	32-ch	32-ch	32-ch	32-ch	32-ch	32-ch
External Peripheral Interface (EPI)	0	1	1	1	1	1	1
Micro Cyclic Redundancy Check (µCRC) Module	0	1	1	1	1	1	1
General-Purpose Timers	–	4	4	4	4	4	4
Watchdog Timer Modules	–	2	2	2	2	2	2
Control Subsystem — C28x Floating-Point Unit (FPU)/Viterbi, Complex Math, CRC Unit (VCU)							
Speed (MHz)(2)		150	150	150	150	150	150
Flash (KB)		512	512	512	512	512	512
RAM ECC (KB)		20	20	20	20	20	20
RAM Parity (KB)		16	16	16	16	16	16
IPC Message RAM Parity (KB)		2	2	2	2	2	2
Security Zones		1	1	1	1	1	1

Table 2-1. Hardware Features (continued)

FEATURE		TYPE[(1)]	P63C	P53C	H53C	H53B	H33C	H33B
Enhanced Pulse Width Modulator (ePWM) modules		2	12: 24 outputs					
High-Resolution PWM outputs		2	16 outputs					
Enhanced Capture (eCAP) modules/PWM outputs		0	6 (32-bit)					
Enhanced Quadrature Encoder (eQEP) modules		0	3 (32-bit)					
Fault Trip Zones		–	12 on any of 64 GPIO pins					
Multichannel Buffered Serial Port (McBSP)/ Serial Peripheral Interface (SPI)		1	1	1	1	1	1	1
Serial Communications Interface (SCI)		0	1	1	1	1	1	1
Serial Peripheral Interface (SPI)		0	1	1	1	1	1	1
Inter-integrated circuit (I2C)		0	1	1	1	1	1	1
Direct Memory Access (DMA)		0	6-ch	6-ch	6-ch	6-ch	6-ch	6-ch
External Peripheral Interface (EPI)		0	1	1	1	1	1	1
32-Bit Timers		–	3	3	3	3	3	3
Shared								
Supplemental RAM Parity (KB)			64	64	64	64	0	0
12-Bit ADC 1	MSPS[(3)]	3	2.88	2.88	2.88	2.88	2.88	2.88
	Conversion Time[(3)]		347 ns	347 ns	347 ns	347 ns	347 ns	347 ns
	Channels		12	12	12	12	12	12
	Sample-and-Hold (S/H)		2	2	2	2	2	2
12-Bit ADC 2	MSPS[(3)]	3	2.88	2.88	2.88	2.88	2.88	2.88
	Conversion Time[(3)]		347 ns	347 ns	347 ns	347 ns	347 ns	347 ns
	Channels		12	12	12	12	12	12
	Sample-and-Hold (S/H)		2	2	2	2	2	2
Comparators with Integrated DACs		0	6	6	6	6	6	6
Voltage Regulator			Yes – Uses 3.3-V Single Supply (3.3-V/1.2-V recommended for 125ºC)					
Clocking			See Section 2.10					
Additional Safety								
Master Subsystem			2 Watchdogs, NMI Watchdog: CPU, Memory					
Control Subsystem			NMI Watchdog: CPU, Memory					
Shared			Critical Register and I/O Function Lock Protection; RAM Fetch Protection					

Table 2-1. Hardware Features (continued)

FEATURE		TYPE(1)	P63C	P53C	H53C	H53B	H33C	H33B
Packaging								
Package Type	289-Ball ZWT Plastic Ball Grid Array		Available at Prototype Sampling					
Temperature options	T: –40°C to 105°C	–	Yes	Yes	Yes	Yes	Yes	Yes
	S: –40°C to 125°C	–	Yes	Yes	Yes	Yes	Yes	Yes
	Q: –40°C to 125°C(4)	–	No	No	No	No	No	No
Product status(5)		–	**x**F28M36...	**x**F28M36...	**x**F28M36...	**x**F28M36...	**x**F28M36...	**x**F28M36...

(1) A type change represents a major functional feature difference in a peripheral module. Within a peripheral type, there may be minor differences between devices that do not affect the basic functionality of the module. These device-specific differences are listed in the *TMS320x28xx, 28xxx DSP Peripheral Reference Guide* (literature number SPRU566) and in the peripheral reference guides.

(2) An integer divide ratio must be maintained between the C28x and Cortex™-M3 clock frequencies; thus, when the C28x is configured to run at maximum frequency of 150 MHz, the fastest allowable frequency for the Cortex™-M3 will be 75 MHz. If the Cortex™-M3 is configured to run at 125 MHz, the maximum frequency of the C28x is limited to 125 MHz.

(3) An integer divide ratio must be maintained between the C28x and ADC clock frequencies. All MSPS and Conversion Time values are based on the maximum C28x clock frequency.

(4) "Q" refers to Q100 qualification for automotive applications.

(5) The "**x**F28M36..." product status denotes an experimental device that is not necessarily representative of the final device's electrical specifications. See Section 7.1.2, Device Nomenclature, for descriptions of device stages.

2.2 Memory Maps

Section 2.2.1 shows the Control Subsystem Memory Map. Section 2.2.2 shows the Master Subsystem Memory Map.

2.2.1 Control Subsystem Memory Map

Table 2-2. Control Subsystem M0, M1 RAM

C DMA Access(1)	C Address (x16 Aligned)(1)	Control Subsystem M0, M1 RAM	Size (Bytes)
no	0000 0000 – 0000 03FF	M0 RAM (ECC)	2K
no	0000 0400 – 0000 07FF	M1 RAM (ECC)	2K

(1) The letter "C" refers to the Control Subsystem.

Table 2-3. Control Subsystem Peripheral Frame 0 (Includes Analog)

C DMA Access(1)	C Address (x16 Aligned)(1)	Control Subsystem Peripheral Frame 0 (Includes Analog)	Size (Bytes)
	0000 0800 – 0000 087F	Reserved	
no	0000 0880 – 0000 0890	Control Subsystem Device Configuration Registers (Read Only)	34
	0000 0891 – 0000 0ADF	Reserved	
no	0000 0AE0 – 0000 0AEF	C28x CSM Registers	32
	0000 0AF0 – 0000 0AFF	Reserved	
yes	0000 0B00 – 0000 0B0F	ADC1 Result Registers	32
	0000 0B10 – 0000 0B3F	Reserved	
yes	0000 0B40 – 0000 0B4F	ADC2 Result Registers	32
	0000 0B50 – 0000 0BFF	Reserved	
no	0000 0C00 – 0000 0C07	CPU Timer 0	16
no	0000 0C08 – 0000 0C0F	CPU Timer 1	16
no	0000 0C10 – 0000 0C17	CPU Timer 2	16
	0000 0C18 – 0000 0CDF	Reserved	
no	0000 0CE0 – 0000 0CFF	PIE Registers	64
no	0000 0D00 – 0000 0DFF	PIE Vector Table	512
no	0000 0E00 – 0000 0EFF	PIE Vector Table Copy (Read Only)	512
	0000 0F00 – 0000 0FFF	Reserved	
no	0000 1000 – 0000 11FF	C28x DMA Registers	1K
	0000 1200 – 0000 16FF	Reserved	
no	0000 1700 – 0000 177F	Analog Subsystem Control Registers	256
no	0000 1780 – 0000 17FF	C Hardware Logic BIST Registers	256
	0000 1800 – 0000 3FFF	Reserved	

(1) The letter "C" refers to the Control Subsystem.

Table 2-4. Control Subsystem Peripheral Frame 3

C DMA Access [1]	C Address (x16 Aligned) [1]	Control Subsystem Peripheral Frame 3	Size (Bytes)	M Address (Byte-Aligned) [2]	µDMA Access
no	0000 4000 – 0000 4181	C28x Flash Control Registers	772		
	0000 4182 – 0000 42FF	Reserved			
no	0000 4300 – 0000 4323	C28x Flash ECC Error Log Registers	72		
	0000 4324 – 0000 43FF	Reserved			
no	0000 4400 – 0000 443F	M Clock Control Registers [2]	128	400F B800 – 400F B87F	no
	0000 4440 – 0000 48FF	Reserved			
no	0000 4900 – 0000 497F	RAM Configuration Registers	256	400F B200 – 400F B2FF	no
	0000 4980 – 0000 49FF	Reserved			
no	0000 4A00 – 0000 4A7F	RAM ECC/Parity/Access Error Log Registers	256	400F B300 – 400F B3FF	no
	0000 4A80 – 0000 4DFF	Reserved			
no	0000 4E00 – 0000 4E3F	CtoM and MtoC IPC Registers	128	400F B700 – 400F B77F	no
	0000 4E40 – 0000 4FFF	Reserved			
yes	0000 5000 – 0000 503F	McBSP-A	128		
	0000 5040 – 0000 50FF	Reserved			
yes	0000 5100 – 0000 517F	EPWM1 (Hi-Resolution)	256		
yes	0000 5180 – 0000 51FF	EPWM2 (Hi-Resolution)	256		
yes	0000 5200 – 0000 527F	EPWM3 (Hi-Resolution)	256		
yes	0000 5280 – 0000 52FF	EPWM4 (Hi-Resolution)	256		
yes	0000 5300 – 0000 537F	EPWM5 (Hi-Resolution)	256		
yes	0000 5380 – 0000 53FF	EPWM6 (Hi-Resolution)	256		
yes	0000 5400 – 0000 547F	EPWM7 (Hi-Resolution)	256		
yes	0000 5480 – 0000 54FF	EPWM8 (Hi-Resolution)	256		
yes	0000 5500 – 0000 557F	EPWM9	256		
yes	0000 5580 – 0000 55FF	EPWM10	256		
yes	0000 5600 – 0000 567F	EPWM11	256		
yes	0000 5680 – 0000 56FF	EPWM12	256		
	0000 5700 – 0000 57FF	Reserved			

(1) The letter "C" refers to the Control Subsystem.
(2) The letter "M" refers to the Master Subsystem.

Table 2-5. Control Subsystem Peripheral Frame 1

C DMA Access[1]	C Address (x16 Aligned)[1]	Control Subsystem Peripheral Frame 1	Size (Bytes)
	0000 5800 – 0000 59FF	Reserved	
no	0000 5A00 – 0000 5A1F	ECAP1	64
no	0000 5A20 – 0000 5A3F	ECAP2	64
no	0000 5A40 – 0000 5A5F	ECAP3	64
no	0000 5A60 – 0000 5A7F	ECAP4	64
no	0000 5A80 – 0000 5A9F	ECAP5	64
no	0000 5AA0 – 0000 5ABF	ECAP6	64
	0000 5AC0 – 0000 5AFF	Reserved	
no	0000 5B00 – 0000 5B3F	EQEP1	128
no	0000 5B40 – 0000 5B7F	EQEP2	128
no	0000 5B80 – 0000 5BBF	EQEP3	128
	0000 5BC0 – 0000 5EFF	Reserved	
no	0000 5F00 – 0000 5FFF	C GPIO Group 1 Registers[1]	512
	0000 6000 – 0000 63FF	Reserved	
no	0000 6400 – 0000 641F	COMP1 Registers	64
no	0000 6420 – 0000 643F	COMP2 Registers	64
no	0000 6440 – 0000 645F	COMP3 Registers	64
no	0000 6460 – 0000 647F	COMP4 Registers	64
no	0000 6480 – 0000 649F	COMP5 Registers	64
no	0000 64A0 – 0000 64BF	COMP6 Registers	64
	0000 64C0 – 0000 6F7F	Reserved	
no	0000 6F80 – 0000 6FFF	C GPIO Group 2 Registers and AIO Mux Registers[1]	256

(1) The letter "C" refers to the Control Subsystem.

Table 2-6. Control Subsystem Peripheral Frame 2

C DMA Access[1]	C Address (x16 Aligned)[1]	Control Subsystem Peripheral Frame 2	Size (Bytes)
	0000 7000 – 0000 70FF	Reserved	
no	0000 7010 – 0000 702F	C28x System Control Registers	64
	0000 7030 – 0000 703F	Reserved	
no	0000 7040 – 0000 704F	SPI-A	32
no	0000 7050 – 0000 705F	SCI-A	32
no	0000 7060 – 0000 706F	NMI Watchdog Interrupt Registers	32
no	0000 7070 – 0000 707F	External Interrupt Registers	32
	0000 7080 – 0000 70FF	Reserved	
no	0000 7100 – 0000 717F	ADC1 Configuration Registers (Only 16-bit read/write access supported)	256
no	0000 7180 – 0000 71FF	ADC2 Configuration Registers (Only 16-bit read/write access supported)	256
	0000 7200 – 0000 78FF	Reserved	
no	0000 7900 – 0000 793F	I2C-A	128
	0000 7940 – 0000 7FFF	Reserved	

(1) The letter "C" refers to the Control Subsystem.

ADVANCE INFORMATION

Table 2-7. Control Subsystem RAMs

C DMA Access[1]	C Address (x16 Aligned)[1]	Control Subsystem RAMs	Size (Bytes)	M Address (Byte-Aligned)[2]	µDMA Access
no	0000 8000 – 0000 8FFF	L0 RAM (ECC, Secure)	8K		
no	0000 9000 – 0000 9FFF	L1 RAM (ECC, Secure)	8K		
yes	0000 A000 – 0000 AFFF	L2 RAM (Parity)	8K		
yes	0000 B000 – 0000 BFFF	L3 RAM (Parity)	8K		
yes	0000 C000 – 0000 CFFF	S0 RAM (Parity, Shared)	8K	2000 8000 – 2000 9FFF	yes
yes	0000 D000 – 0000 DFFF	S1 RAM (Parity, Shared)	8K	2000 A000 – 2000 BFFF	yes
yes	0000 E000 – 0000 EFFF	S2 RAM (Parity, Shared)	8K	2000 C000 – 2000 DFFF	yes
yes	0000 F000 – 0000 FFFF	S3 RAM (Parity, Shared)	8K	2000 E000 – 2000 FFFF	yes
yes	0001 0000 – 0001 0FFF	S4 RAM (Parity, Shared)	8K	2001 0000 – 2001 1FFF	yes
yes	0001 1000 – 0001 1FFF	S5 RAM (Parity, Shared)	8K	2001 2000 – 2001 3FFF	yes
yes	0001 2000 – 0001 2FFF	S6 RAM (Parity, Shared)	8K	2001 4000 – 2001 5FFF	yes
yes	0001 3000 – 0001 3FFF	S7 RAM (Parity, Shared)	8K	2001 6000 – 2001 7FFF	yes
	0001 4000 – 0003 F7FF	Reserved			
yes	0003 F800 – 0003 FBFF	CtoM MSG RAM (Parity)	2K	2007 F000 – 2007 F7FF	yes read only
yes read only	0003 FC00 – 0003 FFFF	MtoC MSG RAM (Parity)	2K	2007 F800 – 2007 FFFF	yes
	0004 0000 – 0004 7FFF	Reserved			
no	0004 8000 – 0004 8FFF	L0 RAM - ECC Bits	8K		
no	0004 9000 – 0004 9FFF	L1 RAM - ECC Bits	8K		
no	0004 A000 – 0004 AFFF	L2 RAM - Parity Bits	8K		
no	0004 B000 – 0004 BFFF	L3 RAM - Parity Bits	8K		
no	0004 C000 – 0004 CFFF	S0 RAM - Parity Bits	8K	2008 8000 – 2008 9FFF	no
no	0004 D000 – 0004 DFFF	S1 RAM - Parity Bits	8K	2008 A000 – 2008 BFFF	no
no	0004 E000 – 0004 EFFF	S2 RAM - Parity Bits	8K	2008 C000 – 2008 DFFF	no
no	0004 F000 – 0004 FFFF	S3 RAM - Parity Bits	8K	2008 E000 – 2008 FFFF	no
no	0005 0000 – 0005 0FFF	S4 RAM - Parity Bits	8K	2009 0000 – 2009 1FFF	no
no	0005 1000 – 0005 1FFF	S5 RAM - Parity Bits	8K	2009 2000 – 2009 3FFF	no
no	0005 2000 – 0005 2FFF	S6 RAM - Parity Bits	8K	2009 4000 – 2009 5FFF	no
no	0005 3000 – 0005 3FFF	S7 RAM - Parity Bits	8K	2009 6000 – 2009 7FFF	no
	0005 4000 – 0007 EFFF	Reserved			
no	0007 F000 – 0007 F3FF	M0 RAM - ECC Bits	2K		
no	0007 F400 – 0007 F7FF	M1 RAM - ECC Bits	2K		
no	0007 F800 – 0007 FBFF	CtoM MSG RAM - Parity Bits	2K	200F F000 – 200F F7FF	no
no	0007 FC00 – 0007 FFFF	MtoC MSG RAM - Parity Bits	2K	200F F800 – 200F FFFF	no
	0008 0000 – 0009 FFFF	Reserved			

(1) The letter "C" refers to the Control Subsystem.
(2) The letter "M" refers to the Master Subsystem.

Table 2-8. Control Subsystem Flash, ECC, OTP, Boot ROM

C DMA Access[1]	C Address (x16 Aligned)[1]	Control Subsystem Flash, ECC, OTP, Boot ROM	Size (Bytes)	M Address (Byte-Aligned)[2]	µDMA Access
no	0010 0000 – 0010 1FFF	Sector N (not available for 256KB Flash configuration)	16K		
no	0010 2000 – 0010 3FFF	Sector M (not available for 256KB Flash configuration)	16K		
no	0010 4000 – 0010 5FFF	Sector L (not available for 256KB Flash configuration)	16K		
no	0010 6000 – 0010 7FFF	Sector K (not available for 256KB Flash configuration)	16K		
no	0010 8000 – 0010 FFFF	Sector J (not available for 256KB Flash configuration)	64K		
no	0011 0000 – 0011 7FFF	Sector I (not available for 256KB Flash configuration)	64K		
no	0011 8000 – 0011 FFFF	Sector H (not available for 256KB Flash configuration)	64K		
no	0012 0000 – 0012 7FFF	Sector G	64K		
no	0012 8000 – 0012 FFFF	Sector F	64K		
no	0013 0000 – 0013 7FFF	Sector E	64K		
no	0013 8000 – 0013 9FFF	Sector D	16K		
no	0013 A000 – 0013 BFFF	Sector C	16K		
no	0013 C000 – 0013 DFFF	Sector B	16K		
no	0013 E000 – 0013 FFFF	Sector A (CSM password in the high address)	16K		
	0014 0000 – 001F FFFF	Reserved			
no	0020 0000 – 0020 7FFF	Flash - ECC Bits (1/8 of Flash used = 64 KBytes)	64K		
	0020 8000 – 0024 01FF	Reserved			
no	0024 0200 – 0024 03FF	TI OTP	1K		
	0024 0400 – 002F FFFF	Reserved			
yes	0030 0000 – 003F 7FFF	EPI0 (External Peripheral/Memory Interface)[3]	2G	6000 0000 – DFFF FFFF	yes
no	003F 8000 – 003F FFFF	C28x Boot ROM (64 KBytes)	64K		

(1) The letter "C" refers to the Control Subsystem.
(2) The letter "M" refers to the Master Subsystem.
(3) The Control Subsystem has no direct access to EPI in silicon revision 0 devices.

2.2.2 Master Subsystem Memory Map

Table 2-9. Master Subsystem Flash, ECC, OTP, Boot ROM

µDMA Access	M Address (Byte-Aligned)(1)	Master Subsystem Flash, ECC, OTP, Boot ROM	Size (Bytes)
no	0000 0000 – 0000 FFFF	Boot ROM - Dual-mapped to 0x0100 0000 (Both maps access same physical location.)	64K
	0001 0000 – 001F FFFF	Reserved	
no	0020 0000 – 0020 7FFF	Sector N (Zone 1 CSM password in the low address.)	32K
no	0020 8000 – 0020 FFFF	Sector M	32K
no	0021 0000 – 0021 7FFF	Sector L	32K
no	0021 8000 – 0021 FFFF	Sector K	32K
no	0022 0000 – 0023 FFFF	Sector J (not available for 256KB Flash configuration)	128K
no	0024 0000 – 0025 FFFF	Sector I (not available for 256KB or 512KB Flash configurations)	128K
no	0026 0000 – 0027 FFFF	Sector H (not available for 256KB or 512KB Flash configurations)	128K
no	0028 0000 – 0029 FFFF	Sector G (not available for 256KB or 512KB Flash configurations)	128K
no	002A 0000 – 002B FFFF	Sector F (not available for 256KB or 512KB Flash configurations)	128K
no	002C 0000 – 002D FFFF	Sector E (not available for 256KB Flash configuration)	128K
no	002E 0000 – 002E 7FFF	Sector D	32K
no	002E 8000 – 002E FFFF	Sector C	32K
no	002F 0000 – 002F 7FFF	Sector B	32K
no	002F 8000 – 002F FFFF	Sector A (Zone 2 CSM password in the high address.)	32K
	0030 0000 – 005F FFFF	Reserved	
no	0060 0000 – 0061 FFFF	Flash - ECC Bits (1/8 of Flash used = 128 KBytes)	128K
	0062 0000 – 0068 047F	Reserved	
no	0068 0480 – 0068 0FFF	TI OTP	2944
no	0068 1000	OTP – Security Lock	4
	0068 1004	Reserved	
	0068 1008	Reserved	
no	0068 100C	OTP – Zone 2 Flash Start Address	4
no	0068 1010	OTP – EMAC Address 0	4
no	0068 1014	OTP – EMAC Address 1	4
no	0068 1018	Reserved	
no	0068 101C	OTP – Main Oscillator Clock Frequency	4
	0068 0820 – 0070 01FF	Reserved	
no	0070 0200 – 0070 0203	OTP – ECC Bits – Application Use (1/8 of OTP used = 3 Bytes)	4
	0070 0204 – 00FF FFFF	Reserved	
no	0100 0000 – 0100 FFFF	Boot ROM – Dual-mapped to 0x0000 0000 (Both maps access same physical location.)	64K
	0101 0000 – 03FF FFFF	Reserved	

---

(1) The letter "M" refers to the Master Subsystem.

Table 2-9. Master Subsystem Flash, ECC, OTP, Boot ROM (continued)

µDMA Access	M Address (Byte-Aligned)(1)	Master Subsystem Flash, ECC, OTP, Boot ROM	Size (Bytes)
no	0400 0000 – 07FF FFFF	ROM/Flash/OTP/Boot ROM – Mirror-mapped ***for µCRC***. Accessing this area of memory by the µCRC peripheral will cause an access in 0000 0000 – 03FF FFFF memory space. Mirrored boot ROM: 0x0400 0000 – 0x0400 FFFF (Not dual-mapped ROM address) Mirrored Flash bank: 0x0420 0000 – 0x042F FFFF Mirrored Flash OTP: 0x0468 0000 – 0x0468 1FFF (Read cycles from this space cause the µCRC peripheral to continuously update data checksum inside a register, when reading a block of data.)	64M
	0800 0000 – 1FFF FFFF	Reserved	

Table 2-10. Master Subsystem RAMs

µDMA Access	M Address (Byte-Aligned)(1)	Master Subsystem RAMs	Size (Bytes)	C Address (x16 Aligned)(2)	C DMA Access(2)
no	2000 0000 – 2000 1FFF	C0 RAM (ECC, Secure)	8K		
no	2000 2000 – 2000 3FFF	C1 RAM (ECC, Secure)	8K		
yes	2000 4000 – 2000 5FFF	C2 RAM (Parity)	8K		
yes	2000 6000 – 2000 7FFF	C3 RAM (Parity)	8K		
yes	2000 8000 – 2000 9FFF	S0 RAM (Parity, Shared)	8K	0000 C000 – 0000 CFFF	yes
yes	2000 A000 – 2000 BFFF	S1 RAM (Parity, Shared)	8K	0000 D000 – 0000 DFFF	yes
yes	2000 C000 – 2000 DFFF	S2 RAM (Parity, Shared)	8K	0000 E000 – 0000 EFFF	yes
yes	2000 E000 – 2000 FFFF	S3 RAM (Parity, Shared)	8K	0000 F000 – 0000 FFFF	yes
yes	2001 0000 – 2001 1FFF	S4 RAM (Parity, Shared)	8K	0001 0000 – 0001 0FFF	yes
yes	2001 2000 – 2001 3FFF	S5 RAM (Parity, Shared)	8K	0001 1000 – 0001 1FFF	yes
yes	2001 4000 – 2001 5FFF	S6 RAM (Parity, Shared)	8K	0001 2000 – 0001 2FFF	yes
yes	2001 6000 – 2001 7FFF	S7 RAM (Parity, Shared)	8K	0001 3000 – 0001 3FFF	yes
yes	2001 8000 – 2001 9FFF	C4 RAM (Parity)	8K		
yes	2001 A000 – 2001 BFFF	C5 RAM (Parity)	8K		
yes	2001 C000 – 2001 DFFF	C6 RAM (Parity)	8K		
yes	2001 E000 – 2001 FFFF	C7 RAM (Parity)	8K		
yes	2002 0000 – 2002 1FFF	C8 RAM (Parity)	8K		
yes	2002 2000 – 2002 3FFF	C9 RAM (Parity)	8K		
yes	2002 4000 – 2002 5FFF	C10 RAM (Parity)	8K		
yes	2002 6000 – 2002 7FFF	C11 RAM (Parity)	8K		
yes	2002 8000 – 2002 9FFF	C12 RAM (Parity)	8K		
yes	2002 A000 – 2002 BFFF	C13 RAM (Parity)	8K		
yes	2002 C000 – 2002 DFFF	C14 RAM (Parity)	8K		
yes	2002 E000 – 2002 FFFF	C15 RAM (Parity)	8K		
	2003 0000 – 2007 EFFF	Reserved			
yes read only	2007 F000 – 2007 F7FF	CtoM MSG RAM (Parity)	2K	0003 F800 – 0003 FBFF	yes
yes	2007 F800 – 2007 FFFF	MtoC MSG RAM (Parity)	2K	0003 FC00 – 0003 FFFF	yes read only
no	2008 0000 – 2008 1FFF	C0 RAM - ECC Bits	8K		
no	2008 2000 – 2008 3FFF	C1 RAM - ECC Bits	8K		
no	2008 4000 – 2008 5FFF	C2 RAM - Parity Bits	8K		
no	2008 6000 – 2008 7FFF	C3 RAM - Parity Bits	8K		
no	2008 8000 – 2008 9FFF	S0 RAM - Parity Bits	8K	0004 C000 – 0004 CFFF	no

(1) The letter "M" refers to the Master Subsystem.
(2) The letter "C" refers to the Control Subsystem.

Table 2-10. Master Subsystem RAMs (continued)

µDMA Access	M Address (Byte-Aligned)[1]	Master Subsystem RAMs	Size (Bytes)	C Address (x16 Aligned)[2]	C DMA Access[2]
no	2008 A000 – 2008 BFFF	S1 RAM - Parity Bits	8K	0004 D000 – 0004 DFFF	no
no	2008 C000 – 2008 DFFF	S2 RAM - Parity Bits	8K	0004 E000 – 0004 EFFF	no
no	2008 E000 – 2008 FFFF	S3 RAM - Parity Bits	8K	0004 F000 – 0004 FFFF	no
no	2009 0000 – 2009 1FFF	S4 RAM - Parity Bits	8K	0005 0000 – 0005 0FFF	no
no	2009 2000 – 2009 3FFF	S5 RAM - Parity Bits	8K	0005 1000 – 0005 1FFF	no
no	2009 4000 – 2009 5FFF	S6 RAM - Parity Bits	8K	0005 2000 – 0005 2FFF	no
no	2009 6000 – 2009 7FFF	S7 RAM - Parity Bits	8K	0005 3000 – 0005 3FFF	no
no	2009 8000 – 2009 9FFF	C4 RAM - Parity Bits	8K		
no	2009 A000 – 2009 BFFF	C5 RAM - Parity Bits	8K		
no	2009 C000 – 2009 DFFF	C6 RAM - Parity Bits	8K		
no	2009 E000 – 2009 FFFF	C7 RAM - Parity Bits	8K		
no	200A 0000 – 200A 1FFF	C8 RAM - Parity Bits	8K		
no	200A 2000 – 200A 3FFF	C9 RAM - Parity Bits	8K		
no	200A 4000 – 200A 5FFF	C10 RAM - Parity Bits	8K		
no	200A 6000 – 200A 7FFF	C11 RAM - Parity Bits	8K		
no	200A 8000 – 200A 9FFF	C12 RAM - Parity Bits	8K		
no	200A A000 – 200A BFFF	C13 RAM - Parity Bits	8K		
no	200A C000 – 200A DFFF	C14 RAM - Parity Bits	8K		
no	200A E000 – 200A FFFF	C15 RAM - Parity Bits	8K		
	200B 0000 – 200F EFFF	Reserved			
no	200F F000 – 200F F7FF	CtoM MSG RAM - Parity Bits	2K	0007 F800 – 0007 FBFF	no
no	200F F800 – 200F FFFF	MtoC MSG RAM - Parity Bits	2K	0007 FC00 – 0007 FFFF	no
	2010 0000 – 21FF FFFF	Reserved			
yes	2200 0000 – 23FF FFFF	Bit Banded RAM Zone (Dedicated address for each RAM bit of Cortex™-M3 RAM blocks above)	32M		
yes	2400 0000 – 27FF FFFF	All RAM Spaces – Mirror-Mapped ***for µCRC***. Accessing this memory by the µCRC peripheral will cause an access to 2000 0000 – 23FF FFFF memory space. (Read cycles from this space cause the µCRC peripheral to continuously update data checksum inside a register when reading a block of data.)	64M		
	2800 0000 – 3FFF FFFF	Reserved			

Table 2-11. Master Subsystem Peripherals

µDMA Access	M Address (Byte-Aligned)(1)	Master Subsystem Peripherals	Size (Bytes)	C Address (x16 Aligned)(2)	C DMA Access(2)
yes	4000 0000 – 4000 0FFF	Watchdog Timer 0 Registers	4K		
yes	4000 1000 – 4000 1FFF	Watchdog Timer 1 Registers	4K		
	4000 2000 – 4000 3FFF	Reserved			
yes	4000 4000 – 4000 4FFF	M GPIO Port A (APB Bus)(1)	4K		
yes	4000 5000 – 4000 5FFF	M GPIO Port B (APB Bus)(1)	4K		
yes	4000 6000 – 4000 6FFF	M GPIO Port C (APB Bus)(1)	4K		
yes	4000 7000 – 4000 7FFF	M GPIO Port D (APB Bus)(1)	4K		
yes	4000 8000 – 4000 8FFF	SSI0	4K		
yes	4000 9000 – 4000 9FFF	SSI1	4K		
yes	4000 A000 – 4000 AFFF	SSI2	4K		
yes	4000 B000 – 4000 BFFF	SSI3	4K		
yes	4000 C000 – 4000 CFFF	UART0	4K		
yes	4000 D000 – 4000 DFFF	UART1	4K		
yes	4000 E000 – 4000 EFFF	UART2	4K		
yes	4000 F000 – 4000 FFFF	UART3	4K		
yes	4001 0000 – 4001 0FFF	UART4	4K		
	4001 1000 – 4001 FFFF	Reserved			
no	4002 0000 – 4002 07FF	I2C0 Master	2K		
no	4002 0800 – 4002 0FFF	I2C0 Slave	2K		
no	4002 1000 – 4002 17FF	I2C1 Master	2K		
no	4002 1800 – 4002 1FFF	I2C1 Slave	2K		
	4002 2000 – 4002 3FFF	Reserved			
yes	4002 4000 – 4002 4FFF	M GPIO Port E (APB Bus)(1)	4K		
yes	4002 5000 – 4002 5FFF	M GPIO Port F (APB Bus)(1)	4K		
yes	4002 6000 – 4002 6FFF	M GPIO Port G (APB Bus)(1)	4K		
yes	4002 7000 – 4002 7FFF	M GPIO Port H (APB Bus)(1)	4K		
	4002 8000 – 4002 FFFF	Reserved			
yes	4003 0000 – 4003 0FFF	GP Timer 0	4K		
yes	4003 1000 – 4003 1FFF	GP Timer 1	4K		
yes	4003 2000 – 4003 2FFF	GP Timer 2	4K		
yes	4003 3000 – 4003 3FFF	GP Timer 3	4K		
	4003 4000 – 4003 CFFF	Reserved			
yes	4003 D000 – 4003 DFFF	M GPIO Port J (APB Bus)(1)	4K		
	4003 E000 – 4003 FFFF	Reserved			
yes	4004 8000 – 4004 8FFF	ENET MAC0	4K		
	4004 9000 – 4004 FFFF	Reserved			
yes	4005 0000 – 4005 0FFF	USB MAC0	4K		
	4005 1000 – 4005 7FFF	Reserved			
yes	4005 8000 – 4005 8FFF	M GPIO Port A (AHB Bus)(1)	4K		
yes	4005 9000 – 4005 9FFF	M GPIO Port B (AHB Bus)(1)	4K		
yes	4005 A000 – 4005 AFFF	M GPIO Port C (AHB Bus)(1)	4K		
yes	4005 B000 – 4005 BFFF	M GPIO Port D (AHB Bus)(1)	4K		
yes	4005 C000 – 4005 CFFF	M GPIO Port E (AHB Bus)(1)	4K		
yes	4005 D000 – 4005 DFFF	M GPIO Port F (AHB Bus)(1)	4K		
yes	4005 E000 – 4005 EFFF	M GPIO Port G (AHB Bus)(1)	4K		

(1) The letter "M" refers to the Master Subsystem.
(2) The letter "C" refers to the Control Subsystem.

ADVANCE INFORMATION

ADVANCE INFORMATION

Table 2-11. Master Subsystem Peripherals (continued)

µDMA Access	M Address (Byte-Aligned)(1)	Master Subsystem Peripherals	Size (Bytes)	C Address (x16 Aligned)(2)	C DMA Access(2)
yes	4005 F000 – 4005 FFFF	M GPIO Port H (AHB Bus)(1)	4K		
yes	4006 0000 – 4006 0FFF	M GPIO Port J (AHB Bus)(1)	4K		
yes	4006 1000 – 4006 1FFF	M GPIO Port K (AHB Bus)(1)	4K		
yes	4006 2000 – 4006 2FFF	M GPIO Port L (AHB Bus)(1)	4K		
yes	4006 3000 – 4006 3FFF	M GPIO Port M (AHB Bus)(1)	4K		
yes	4006 4000 – 4006 4FFF	M GPIO Port N (AHB Bus)(1)	4K		
yes	4006 5000 – 4006 5FFF	M GPIO Port P (AHB Bus)(1)	4K		
yes	4006 6000 – 4006 6FFF	M GPIO Port Q (AHB Bus)(1)	4K		
yes	4006 7000 – 4006 7FFF	M GPIO Port R (AHB Bus)(1)	4K		
yes	4006 8000 – 4006 8FFF	M GPIO Port S (AHB Bus)(1)	4K		
	4006 9000 – 4006 FFFF	Reserved			
no	4007 0000 – 4007 3FFF	CAN0	16K		
no	4007 4000 – 4007 7FFF	CAN1	16K		
	4007 8000 – 400C FFFF	Reserved			
no	400D 0000 – 400D 0FFF	EPI0 (Registers only)	4K		
	400D 1000 – 400F 9FFF	Reserved			
no	400F A000 – 400F A303	M Flash Control Registers(3)	772		
	400F A304 – 400F A5FF	Reserved			
no	400F A600 – 400F A647	M Flash ECC Error Log Registers(3)	72		
	400F A648 – 400F AFFF	Reserved			
no	400F B000 – 400F B1FF	PBIST Control Registers	512		
no	400F B200 – 400F B2FF	RAM Configuration Registers	256	0000 4900 – 0000 497F	no
no	400F B300 – 400F B3FF	RAM ECC/Parity/Access Error Log Registers	256	0000 4A00 – 0000 4A7F	no
no	400F B400 – 400F B5FF	M CSM Registers(1)	512		
no	400F B600 – 400F B67F	µCRC	128		
	400F B680 – 400F B6FF	Reserved			
no	400F B700 – 400F B77F	CtoM and MtoC IPC Registers	128	0000 4E00 – 0000 4E3F	no
	400F B780 – 400F B7FF	Reserved			
no	400F B800 – 400F B87F	M Clock Control Registers(1)	128	0000 4400 – 0000 443F	no
no	400F B880 – 400F B8BF	M LPM Control Registers(1)	64		
no	400F B8C0 – 400F B8FF	M Reset Control Registers(1)	64		
no	400F B900 – 400F B93F	Device Configuration Registers	64	0000 0880 – 0000 0890 (Read Only)	
	400F B940 – 400F B97F	Reserved			
no	400F B980 – 400F B9FF	M Write Protect Registers(1)	128		
no	400F BA00 – 400F BA7F	M NMI Registers(1)	128		
	400F BA80 – 400F BAFF	Reserved			
no	400F BB00 – 400F BBFF	M HWBIST Registers	256		
	400F BC00 – 400F EFFF	Reserved			
no	400F F000 – 400F FFFF	µDMA Registers	4K		
	4010 0000 – 41FF FFFF	Reserved			
yes	4200 0000 – 43FF FFFF	Bit Banded Peripheral Zone (Dedicated address for each register bit of Cortex™-M3 peripherals above.)	32M		
	4400 0000 – 4FFF FFFF	Reserved			

(3) The letter "M" refers to the Master Subsystem.

Table 2-12. Master Subsystem Analog and EPI

µDMA Access	M Address (Byte-Aligned)(1)	Master Subsystem Analog and EPI	Size (Bytes)	C Address (x16 Aligned)(2)	C DMA Access(2)
	5000 0000 – 5000 15FF	Reserved			
yes	5000 1600 – 5000 161F	ADC1 Result Registers	32		
	5000 1620 – 5000 167F	Reserved			
yes	5000 1680 – 5000 169F	ADC2 Result Registers	32		
	5000 16A0 – 5FFF FFFF	Reserved			
yes	6000 0000 – DFFF FFFF	EPI0 (External Peripheral/Memory Interface)	2G	0030 0000 – 003F 7FFF(3)	yes

(1) The letter "M" refers to the Master Subsystem.
(2) The letter "C" refers to the Control Subsystem.
(3) The Control Subsystem has no direct access to EPI in silicon revision 0 devices.

Table 2-13. Cortex™-M3 Private Bus

µDMA Access	Cortex™-M3 Address (Byte-Aligned)	Cortex™-M3 Private Bus	Size (Bytes)
no	E000 0000 – E000 0FFF	ITM (Instrumentation Trace Macrocell)	4K
no	E000 1000 – E000 1FFF	DWT (Data Watchpoint and Trace)	4K
no	E000 2000 – E000 2FFF	FPB (Flash Patch and Breakpoint)	4K
	E000 3000 – E000 E007	Reserved	
no	E000 E008 – E000 E00F	System Control Block	8
no	E000 E010 – E000 E01F	System Timer	16
	E000 E020 – E000 E0FF	Reserved	
no	E000 E100 – E000 E4EF	Nested Vectored Interrupt Controller (NVIC)	1008
	E000 E4F0 – E000 ECFF	Reserved	
no	E000 ED00 – E000 ED3F	System Control Block	64
	E000 ED40 – E000 ED8F	Reserved	
no	E000 ED90 – E000 EDB8	Memory Protection Unit	41
	E000 EDB9 – E000 EEFF	Reserved	
no	E000 EF00 – E000 EF03	Nested Vectored Interrupt Controller (NVIC)	4
	E000 EF04 – FFFF FFFF	Reserved	

2.3 Master Subsystem

The Master Subsystem includes the Cortex™-M3 CPU, µDMA, Nested Vectored Interrupt Controller (NVIC), Cortex™-M3 Peripherals, and Local Memory. Additionally, the Cortex™-M3 CPU and µDMA can access the Control Subsystem through Shared Resources: IPC (CPU only), Message RAM, and Shared RAM; and read ADC Result Registers via the Analog Common Interface Bus. The Master Subsystem can also receive events from the NMI block and send events to the Resets block.

Figure 2-1 shows the Master Subsystem.

2.3.1 *Cortex™-M3 CPU*

The 32-bit Cortex™-M3 processor offers high performance, fast interrupt handling, and access to a variety of communication peripherals (including Ethernet and USB). The Cortex™-M3 features a Memory Protection Unit (MPU) to provide a privileged mode for protected operating system functionality. A bus bridge adjacent to the MPU can route program instructions and data on the I-CODE and D-CODE buses that connect to the Boot ROM and Flash. Other data is typically routed through the Cortex™-M3 System Bus connected to the local RAMs. The System Bus also goes to the Shared Resources block (also accessible by the Control Subsystem) and to the Analog Subsystem through the Analog Common Interface Bus (ACIB). Another bus bridge allows bus cycles from both the Cortex™-M3 System Bus and those of the µDMA bus to access the Master Subsystem peripherals (via the APB bus or the AHP bus).

Most of the interrupts to the Cortex™-M3 CPU come from the Nested Vectored Interrupt Controller (NVIC), which manages the interrupt requests from peripherals and assigns handling priorities. There are also several exceptions generated by Cortex™-M3 CPU that can return to the Cortex™-M3 as interrupts after being prioritized with other requests inside the NVIC. In addition to programmable priority interrupts, there are also three levels of fixed-priority interrupts of which the highest priority, level-3, is given to M3PORRST and M3SYSRST resets from the Resets block. The next highest priority, level-2, is assigned to the M3NMIINT, which originates from the NMI block. The M3HRDFLT (Hard Fault) interrupt is assigned to level-1 priority, and this interrupt is caused by one of the error condition exceptions (Memory Management, Bus Fault, Usage Fault) escalating to Hard Fault because they are not enabled or not properly serviced.

The Cortex™-M3 CPU has two low-power modes: Sleep and Deep Sleep.

2.3.2 *Cortex™-M3 Core Hardware Logic Built-In Test (LBIST)*

The Concerto™ microcontroller Cortex™-M3 CPU core includes a Logic Built-In Self Test (LBIST) controller for testing the CPU core logic for errors. Tests are initiated by software whenever convenient (at start-up, idle, and so on), which allows for periodic logic tests to ensure that the CPU core logic is working correctly. During a test cycle, all interrupts are logged by the LBIST controller and re-issued after the test cycle completes to ensure that no interrupts are missed. In the event of a logic error, the LBIST controller generates an NMI on both cores to signal that an error has been detected. This action allows for the software to gracefully handle any detected logic errors.

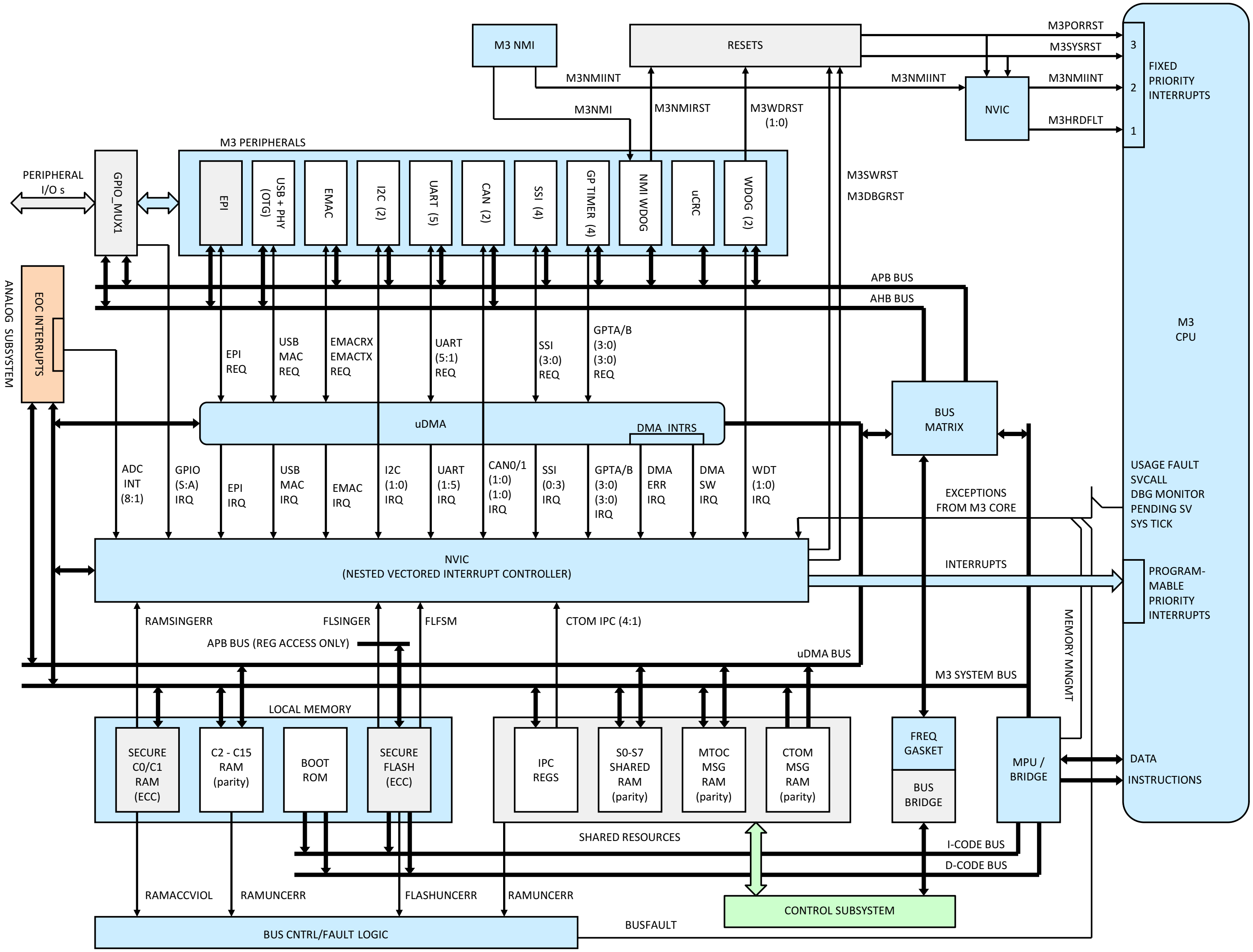


Figure 2-1. Master Subsystem

2.3.3 Cortex™-M3 DMA and NVIC

The Cortex™-M3 direct memory access (µDMA) module provides a hardware method of transferring data between peripherals, between memory, and between peripherals and memory without intervention from the Cortex™-M3 CPU. The Nested Vectored Interrupt Controller (NVIC) manages and prioritizes interrupt handling for the Cortex™-M3 CPU.

The Cortex™-M3 peripherals use REQ/DONE handshaking to coordinate data transfer requests with the µDMA. If a DMA channel is enabled for a given peripheral, REQ/DONE from the peripheral will trigger the data transfer, following which an IRQ request may be sent from the µDMA to the NVIC to announce to the Cortex™-M3 that the transfer has completed. If a DMA channel is not enabled for a given peripheral, REQ/DONE will directly drive IRQ to the NVIC so that the Cortex™-M3 CPU can transfer the data. For those peripherals that are not supported by the µDMA, IRQs are supplied directly to the NVIC, bypassing the DMA. This case is true for both Watchdogs, CANs, I2Cs, and the Analog-to-Digital Converters sending ADCINT[8:1] interrupts from the Analog Subsystem. The NMI Watchdog does not send any events to the µDMA or the NVIC (only to the Resets block).

2.3.4 Cortex™-M3 Interrupts

Table 2-14 shows all interrupt assignments for the Cortex™-M3 processor. Most interrupts (16–107) are associated with interrupt requests from Cortex™-M3 peripherals. The first 15 interrupts (1–15) are processor exceptions generated by the Cortex™-M3 core itself. These processor exceptions are detailed in Table 2-15.

Table 2-14. Interrupts from NVIC to Cortex™-M3

Interrupt Number (Bit in Interrupt Registers)	Vector Number	Vector Address or Offset	Description
–	0–15	0x0000.0000–0x0000.003C	Processor exceptions
0	16	0x0000.0040	GPIO Port A
1	17	0x0000.0044	GPIO Port B
2	18	0x0000.0048	GPIO Port C
3	19	0x0000.004C	GPIO Port D
4	20	0x0000.0050	GPIO Port E
5	21	0x0000.0054	UART0
6	22	0x0000.0058	UART1
7	23	0x0000.005C	SSI0
8	24	0x0000.0060	I2C0
9–17	25–33	–	Reserved
18	34	0x0000.0088	Watchdog Timers 0 and 1
19	35	0x0000.008C	Timer 0A
20	36	0x0000.0090	Timer 0B
21	37	0x0000.0094	Timer 1A
22	38	0x0000.0098	Timer 1B
23	39	0x0000.009C	Timer 2A
24	40	0x0000.00A0	Timer 2B
25–27	41–43	–	Reserved
28	44	0x0000.00B0	System Control
29	45	–	Reserved
30	46	0x0000.00B8	GPIO Port F
31	47	0x0000.00BC	GPIO Port G
32	48	0x0000.00C0	GPIO Port H
33	49	0x0000.00C4	UART2
34	50	0x0000.00C8	SSI1

Table 2-14. Interrupts from NVIC to Cortex™-M3 (continued)

Interrupt Number (Bit in Interrupt Registers)	Vector Number	Vector Address or Offset	Description
35	51	0x0000.00CC	Timer 3A
36	52	0x0000.00D0	Timer 3B
37	53	0x0000.00D4	I2C1
38–41	54–57	–	Reserved
42	58	0x0000.00E8	Ethernet Controller
44	60	0x0000.00F0	USB
45	61	–	Reserved
46	62	0x0000.00F8	µDMA Software
47	63	0x0000.00FC	µDMA Error
48–52	64–68	–	Reserved
53	69	0x0000.0114	EPI
54	70	0x0000.0118	GPIO Port J
55	71	0x0000.011C	GPIO Port K
56	72	0x0000.0120	GPIO Port L
57	73	0x0000.0124	SSI 2
58	74	0x0000.0128	SSI 3
59	75	0x0000.012C	UART3
60	76	0x0000.0130	UART4
61–63	77–79	–	Reserved
64	80	0x0000.0140	CAN1 INT0
65	81	0x0000.0144	CAN1 INT1
66	82	0x0000.0148	CAN1 INT0
67	83	0x0000.014C	CAN1 INT1
68–71	84–87	–	Reserved
72	88	0x0000.0160	ADCINT1
73	89	0x0000.0164	ADCINT2
74	90	0x0000.0168	ADCINT3
75	91	0x0000.016C	ADCINT4
76	92	0x0000.0170	ADCINT5
77	93	0x0000.0174	ADCINT6
78	94	0x0000.0178	ADCINT7
79	95	0x0000.017C	ADCINT8
80	96	0x0000.0180	CTOMIPC1
81	97	0x0000.0184	CTOMIPC2
82	98	0x0000.0188	CTOMIPC3
83	99	0x0000.018C	CTOMIPC4
84–87	100–103	–	Reserved
88	104	0x0000.01A0	RAM Single Error
89	105	0x0000.01A4	System / USB PLL Out of Lock
90	106	0x0000.01A8	M3 Flash Single Error
91	107	0x0000.01AC	PBIST Done
92–110	108–126	–	Reserved
111	127	0x0000.01FC	GPIO Port M
112	128	0x0000.0200	GPIO Port N
113–115	129–131	–	Reserved
116	132	0x0000.0210	GPIO Port P
117–123	133–139	–	Reserved

Table 2-14. Interrupts from NVIC to Cortex™-M3 (continued)

Interrupt Number (Bit in Interrupt Registers)	Vector Number	Vector Address or Offset	Description
124	140	0x0000.0230	GPIO Port Q
125–131	141–147	–	Reserved
132	148	0x0000.0250	GPIO Port R
133	149	0x0000.0254	GPIO Port S

Table 2-15. Exceptions from Cortex™-M3 Core to NVIC

Exception Type	Priority[1]	Vector Number	Vector Address or Offset	Activation
–	–	0	0x0000.0000	Stack top is loaded from the first entry of the vector table on reset.
Reset	–3 (highest)	1	0x0000.0004	Asynchronous
Non-Maskable Interrupt (NMI)	–2	2	0x0000.0008	Asynchronous On Concerto devices activated by clock fail condition, C28 PIE error, external M3GPIO NMI input signal, and C28 NMI WD timeout reset.
Hard Fault	–1	3	0x0000.000C	–
Memory Management	programmable	4	0x0000.0010	Synchronous
Bus Fault	programmable	5	0x0000.0014	Synchronous when precise and asynchronous when imprecise. On Concerto devices activated by memory access errors and RAM and flash uncorrectable data errors.
Usage Fault	programmable	6	0x0000.0018	Synchronous
–	–	7–10	–	Reserved
SVCall	programmable	11	0x0000.002C	Synchronous
Debug Monitor	programmable	12	0x0000.0030	Synchronous
–	–	13	–	Reserved
PendSV	programmable	14	0x0000.0038	Asynchronous
SysTick	programmable	15	0x0000.003C	Asynchronous
Interrupts	programmable	16 and above	0x0000.0040 and above	Asynchronous

(1) 0 is the default priority for all the programmable priorities

2.3.5 Cortex™-M3 Vector Table

Each peripheral interrupt of Table 2-14 is assigned an address offset containing the location of the peripheral interrupt handler (relative to the vector table base) for that particular interrupt (vector numbers 16–107).

Similarly, each exception interrupt of Table 2-15 (including Reset) is also assigned an address offset containing the location of the exception interrupt handler (relative to the vector table base) for that particular interrupt (vector numbers 1–15).

In addition to interrupt vectors, the vector table also contains the initial stack pointer value at table location 0.

Following system reset, the vector table base is fixed at address 0x0000.0000. Privileged software can write to the Vector Table Offset (VTABLE) register to relocate the vector table start address to a different memory location, in the range 0x0000 0200 to 0x3FFF FE00. Note that when configuring the VTABLE register, the offset must be aligned on a 512-byte boundary.

2.3.6 Cortex™-M3 Local Peripherals

The Cortex™-M3 local peripherals include two Watchdogs, an NMI Watchdog, four General-Purpose Timers, four SSI peripherals, two CAN peripherals, five UARTs, two I2C peripherals, Ethernet, USB + PHY, EPI, and µCRC (Cyclic Redundancy Check). The USB and EPI are accessible through the AHB Bus (Advanced High-Performance Bus). The EPI peripheral is also accessible from the Control Subsystem. The remaining peripherals are accessible through the APB Bus (Advanced Peripheral Bus). The APB and AHB bus cycles originate from the CPU System Bus or the µDMA Bus via a bus bridge.

While the Cortex™-M3 CPU has access to all the peripherals, the µDMA has access to most, with the exception of the µCRC, Watchdogs, NMI Watchdog, CAN peripherals, and the I2C peripheral. The Cortex™-M3 peripherals connect to the Concerto™ device pins via GPIO_MUX1. Most of the peripherals also generate event signals for the µDMA and the NVIC. The Watchdogs receive M3SWRST from the NVIC (triggered by software) and send M3WDRST[1:0] reset requests to the Reset block. The NMI Watchdog receives the M3NMI event from the NMI block and sends the M3NMIRST request to the Resets block.

See Section 6.2 for more information on the Cortex™-M3 peripherals.

2.3.7 Cortex™-M3 Local Memory

The Local Memory includes Boot ROM; Secure Flash with Error Correction Code (ECC); Secure C0/C1 RAM with ECC; and C2/C3 RAM with Parity Error Checking. The Boot ROM and Flash are both accessible through the I-CODE and D-CODE Buses. Flash registers can also be accessed by the Cortex™-M3 CPU through the APB Bus. All Local Memory is accessible from the Cortex™-M3 CPU; the C2/C3 RAM is also accessible by the µDMA.

Two types of error correction events can be generated during access of the Local Memory: uncorrectable errors and single errors. The uncorrectable errors (including one from the Shared Memories) generate a Bus Fault Exception to the Cortex™-M3 CPU. The less critical single errors go to the NVIC where they can result in maskable interrupts to the Cortex™-M3 CPU.

2.3.8 Cortex™-M3 Accessing Shared Resources and Analog Peripherals

There are several memories, digital peripherals, and analog peripherals that can be accessed by both the Master and Control Subsystems. They are grouped into Shared Resources and the Analog Subsystem.

The Shared Resources include the External Peripheral Interface (EPI), Inter-Processor Communications (IPC) registers, MTOC Message RAM, CTOM Message RAM, and eight individually configurable Shared RAM blocks. The RAMs of the Shared Resources block have Parity Error Checking.

The Message RAMs and the Shared RAMs can be accessed by the Cortex™-M3 CPU and µDMA. The MTOC Message RAM is intended for sending data from the Master Subsystem to the Control Subsystem, having r/w access for the Cortex™-M3/µDMA and read-only access for the C28x/DMA. The CTOM Message RAM is intended for sending data from the Control Subsystem to the Master Subsystem, having r/w access for the C28x/DMA and read-only access for the Cortex™-M3/µDMA.

The IPC registers provide up to 32 handshaking channels to coordinate the transfer of data through the Message RAMs by polling. Four of these channels are also backed up by four interrupts to PIE on the Control Subsystem side, and four interrupts to the NVIC on the Master Subsystem side (to reduce delays associated with polling).

The eight Shared RAM blocks are similar to the Message RAMs, in that the data flow is only one way; however, the direction of the data flow can be individually set for each block to be from Master to Control Subsystem or from Control to Master Subsystem.

The Analog Subsystem has ADC1, ADC2, and Analog Comparator peripherals that can be accessed through the Analog Common Interface Bus. The ADC Result Registers are accessible by CPUs and DMAs of the Master and Control Subsystems. All other Analog Peripheral Registers are accessible by the C28x CPU only. The Cortex™-M3 CPU accesses the ACIB through the System Bus, and the µDMA through the µDMA Bus. The ACIB arbitrates for access to the ADC and Analog Comparator registers between CPU/DMA bus cycles of the Master Subsystem with those of the Control Subsystem. In addition to managing bus cycles, the ACIB also transfers End-of-Conversion ADC interrupts to the Master Subsystem (as well as to the Control Subsystem). The eight EOC sources from ADC1 and the eight EOC sources from ADC2 are AND-ed together by the ACIB, with the resulting eight ADC interrupts going to destinations in both the Master Subsystem and the Control Subsystem.

See Section 6.1 for more information on shared resources and analog peripherals.

2.4 Control Subsystem

The Control Subsystem includes the C28x CPU/FPU/VCU, Peripheral Interrupt Expansion (PIE) block, DMA, C28x Peripherals, and Local Memory. Additionally, the C28x CPU and DMA have access to Shared Resources: IPC (CPU only), Message RAM, and Shared RAM; and to Analog Peripherals via the Analog Common Interface Bus.

Figure 2-2 shows the Control Subsystem.

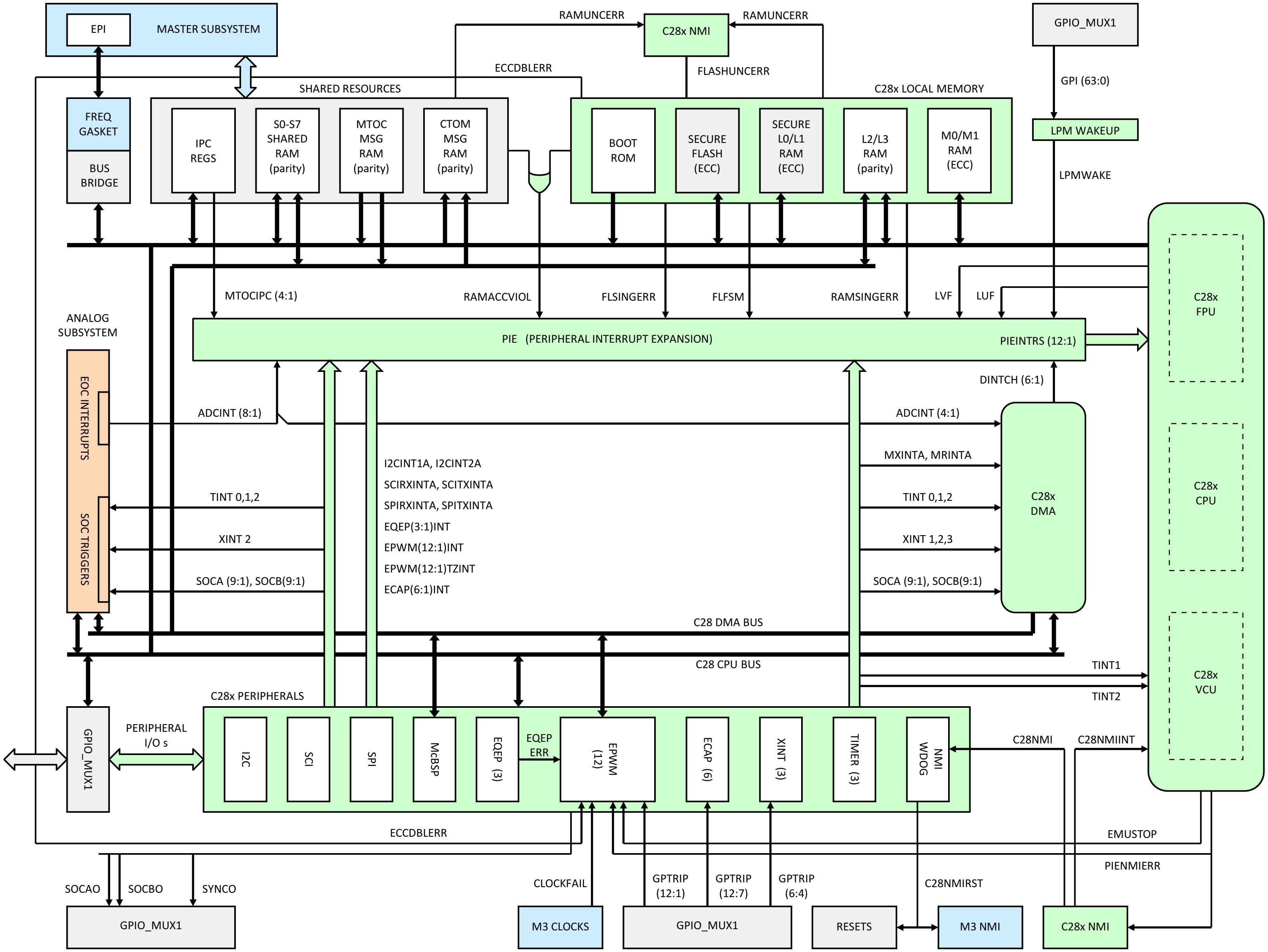


Figure 2-2. Control Subsystem

ADVANCE INFORMATION

2.4.1 C28x CPU/FPU/VCU

The F28M36x Concerto™ MCU family is a member of the TMS320C2000™ MCU platform. The Concerto™ C28x CPU/FPU has the same 32-bit fixed-point architecture as TI's existing Piccolo™ MCUs, combined with a single-precision (32-bit) IEEE 754 floating-point unit (FPU) of TI’s existing Delfino™ MCUs. Each F28M36x device is a very efficient C/C++ engine, enabling users to develop their system control software in a high-level language. Each F28M36x device also enables math algorithms to be developed using C/C++. The device is equally efficient at DSP math tasks and at system control tasks. The 32 x 32-bit MAC 64-bit processing capabilities enable the controller to handle higher numerical resolution problems efficiently. With the addition of the fast interrupt response with automatic context save of critical registers, the device is capable of servicing many asynchronous events with minimal latency. The device has an 8-level-deep protected pipeline with pipelined memory accesses. This pipelining enables the device to execute at high speeds without resorting to expensive high-speed memories. Special branch-look-ahead hardware minimizes the latency for conditional discontinuities. Special conditional store operations further improve performance. The VCU extends the capabilities of the C28x CPU and C28x+FPU processors by adding additional instructions to accelerate Viterbi, Complex Arithmetic, 16-bit FFTs, and CRC algorithms. No changes have been made to existing instructions, pipeline, or memory bus architecture. Therefore, programs written for the C28x are completely compatible with the C28x+VCU.

There are two events generated by the FPU block that go to the C28x Peripheral Interrupt Expansion (PIE): LVF and LUV. Inside PIE, these and other events from C28x peripherals and memories result in 12 PIE interrupts PIEINTS[12:1] into the C28x CPU. The C28x CPU also receives three additional interrupts directly (instead of through PIE) from Timer 1 (TINT1), from Timer 2 (TINT2), and from the NMI block (C28uNMIINT).

The C28x has two low-power modes: Idle and Standby.

2.4.2 C28x™ Core Hardware Logic Built-In Test (LBIST)

The Concerto™ microcontroller C28x CPU core includes a Logic Built-In Self Test (LBIST) controller for testing the CPU core logic for errors. Tests are initiated by software whenever convenient (at start-up, idle, and so on), which allows for periodic logic tests to ensure that the CPU core logic is working correctly. During a test cycle, all interrupts are logged by the LBIST controller and re-issued after the test cycle completes to ensure that no interrupts are missed. In the event of a logic error, the LBIST controller generates an NMI on both cores to signal that an error has been detected. This action allows for the software to gracefully handle any detected logic errors.

2.4.3 C28x Peripheral Interrupt Expansion (PIE)

The PIE block serves to multiplex numerous interrupt sources into a smaller set of interrupt inputs. The PIE block can support up to 96 peripheral interrupts. On the F28M36x, 72 of the possible 96 interrupts are used. The 96 interrupts are grouped into blocks of 8 and each group is fed into 1 of 12 CPU interrupt lines (INT1 to INT12). Each of 12 interrupt lines supports up to 8 simultaneously active interrupts. Each of the 96 interrupts has its own vector stored in a dedicated RAM block that can be overwritten by the user. The vector is automatically fetched by the CPU on servicing the interrupt. Eight CPU clock cycles are needed to fetch the vector and save critical CPU registers. Hence, the CPU can quickly respond to interrupt events. Prioritization of interrupts is controlled in hardware and software. Each individual interrupt can be enabled or disabled within the PIE block.

See Table 2-16 for PIE interrupt assignments.

Table 2-16. PIE Peripheral Interrupts(1)

CPU INTERRUPTS	PIE INTERRUPTS							
	INTx.8	**INTx.7**	**INTx.6**	**INTx.5**	**INTx.4**	**INTx.3**	**INTx.2**	**INTx.1**
INT1	C28.LPMWAKE (C28LPM) 0x0D4E	TINT0 (TIMER 0) 0x0D4C	Reserved – 0x0D4A	XINT2 – 0x0D48	XINT1 – 0x0D46	Reserved – 0x0D44	ADCINT2 (ADC) 0x0D42	ADCINT1 (ADC) 0x0D40
INT2	EPWM8_TZINT (ePWM8) 0x0D5E	EPWM7_TZINT (ePWM7) 0x0D5C	EPWM6_TZINT (ePWM6) 0x0D5A	EPWM5_TZINT (ePWM5) 0x0D58	EPWM4_TZINT (ePWM4) 0x0D56	EPWM3_TZINT (ePWM3) 0x0D54	EPWM2_TZINT (ePWM2) 0x0D52	EPWM1_TZINT (ePWM1) 0x0D50
INT3	EPWM8_INT (ePWM8) 0x0D6E	EPWM7_INT (ePWM7) 0x0D6C	EPWM6_INT (ePWM6) 0x0D6A	EPWM5_INT (ePWM5) 0x0D68	EPWM4_INT (ePWM4) 0x0D66	EPWM3_INT (ePWM3) 0x0D64	EPWM2_INT (ePWM2) 0x0D62	EPWM1_INT (ePWM1) 0x0D60
INT4	EPWM9_TZINT (ePWM9) 0x0D7E	EPWM10_TZINT (ePWM10) 0x0D7C	ECAP6_INT (eCAP6) 0x0D7A	ECAP5_INT (eCAP5) 0x0D78	ECAP4_INT (eCAP4) 0x0D76	ECAP3_INT (eCAP3) 0x0D74	ECAP2_INT (eCAP2) 0x0D72	ECAP1_INT (eCAP1) 0x0D70
INT5	EPWM9_INT (ePWM9) 0x0D8E	EPWM10_INT (ePWM10) 0x0D8C	Reserved – 0x0D8A	Reserved – 0x0D88	Reserved – 0x0D86	EQEP3_INT (eQEP3) 0x0D84	EQEP2_INT (eQEP2) 0x0D82	EQEP1_INT (eQEP1) 0x0D80
INT6	EPWM11_TZINT (ePWM11) 0x0D9E	EPWM12_TZINT (ePWM12) 0x0D9C	MXINTA (McBSPA) 0x0D9A	MRINTA (McBSPA) 0x0D98	Reserved – 0x0D96	Reserved – 0x0D94	SPITXINTA (SPIA) 0x0D92	SPIRXINTA (SPIA) 0x0D90
INT7	EPWM11_INT (ePWM11) 0x0DAE	EPWM12_INT (ePWM12) 0x0DAC	DINTCH6 (C28 DMA) 0x0DAA	DINTCH5 (C28 DMA) 0x0DA8	DINTCH4 (C28 DMA) 0x0DA6	DINTCH3 (C28 DMA) 0x0DA4	DINTCH2 (C28 DMA) 0x0DA2	DINTCH1 (C28 DMA) 0x0DA0
INT8	Reserved – 0x0DBE	Reserved – 0x0DBC	Reserved – 0x0DBA	Reserved – 0x0DB8	Reserved – 0x0DB6	Reserved – 0x0DB4	I2CINT2A (I2CA) 0x0DB2	I2CINT1A (I2CA) 0x0DB0
INT9	Reserved – 0x0DCE	Reserved – 0x0DCC	Reserved – 0x0DCA	Reserved – 0x0DC8	Reserved – 0x0DC6	Reserved – 0x0DC4	SCITXINTA (SCIA) 0x0DC2	SCIRXINTA (SCIA) 0x0DC0
INT10	ADCINT8 (ADC) 0x0DDE	ADCINT7 (ADC) 0x0DDC	ADCINT6 (ADC) 0x0DDA	ADCINT5 (ADC) 0x0DD8	ADCINT4 (ADC) 0x0DD6	ADCINT3 (ADC) 0x0DD4	ADCINT2 (ADC) 0x0DD2	ADCINT1 (ADC) 0x0DD0
INT11	Reserved – 0x0DEE	Reserved – 0x0DEC	Reserved – 0x0DEA	Reserved – 0x0DE8	MTOCIPCINT4 (IPC) 0x0DE6	MTOCIPCINT3 (IPC) 0x0DE4	MTOCIPCINT2 (IPC) 0x0DE2	MTOCIPCINT1 (IPC) 0x0DE0
INT12	LUF (C28FPU) 0x0DFE	LVF (C28FPU) 0x0DFC	EPI_INT (EPI) 0x0DFA	C28RAMACCVIOL (Memory) 0x0DF8	C28RAMSINGERR (Memory) 0x0DF6	Reserved – 0x0DF4	C28FLSINGERR (Memory) 0x0DF2	XINT3 (Ext. Int. 3) 0x0DF0

(1) Out of the 96 possible interrupts, 72 interrupts are currently used. The remaining interrupts are reserved for future devices. These interrupts can be used as software interrupts if they are enabled at the PIEIFRx level, provided none of the interrupts within the group is being used by a peripheral. Otherwise, interrupts coming in from peripherals may be lost by accidentally clearing their flag while modifying the PIEIFR. To summarize, there are two safe cases when the reserved interrupts could be used as software interrupts:
1) No peripheral within the group is asserting interrupts.
2) No peripheral interrupts are assigned to the group (example PIE group 11).

2.4.4 C28x DMA

The C28x direct memory access (DMA) module provides a hardware method of transferring data between peripherals, between memory, and between peripherals and memory without intervention from the CPU, thereby freeing up bandwidth for other system functions. Additionally, the DMA has the capability to orthogonally rearrange the data as the data is transferred as well as “ping-pong” data between buffers. These features are useful for structuring data into blocks for optimal CPU processing. The interrupt trigger source for each of the six DMA channels can be configured separately and each channel contains its own independent PIE interrupt to notify the CPU when a DMA transfer has either started or completed. Five of the six channels are exactly the same, while Channel 1 has one additional feature: the ability to be configured at a higher priority than the others.

2.4.5 C28x Local Peripherals

The C28x local peripherals include an NMI Watchdog, three Timers, four Serial Port Peripherals (SCI, SPI, McBSP, I2C), an External Peripheral Interface (EPI), and three types of Control Peripherals (ePWM, eQEP, eCAP). All peripherals are accessible by the C28x CPU via the C28x Memory Bus. Additionally, the McBSP and ePWM are accessible by the C28x DMA Bus. The EPI peripheral is also accessible from the Master Subsystem. The Serial Port Peripherals and the Control Peripherals connect to Concerto's pins via the GPIO_MUX1 block. Internally, the C28x peripherals generate events to the PIE block, C28x DMA, and the Analog Subsystem. The C28x NMI Watchdog receives a C28NMI event from the NMI block and sends a counter timeout event to the Cortex™-M3 NMI block and the Resets block to flag a potentially critical condition.

The ePWM peripheral receives events that can be used to trip the ePWM outputs EPWMxA and EPWMxB. These events include ECCDBLERR event from the C28x Local Memory, PIENMIERR and EMUSTOP events from the C28x CPU, and up to 12 trips from GPIO_MUX1.

See Section 6.3 for more information on C28x peripherals.

2.4.6 C28x Local Memory

The C28x Local Memory includes Boot ROM; Secure Flash with Error Correction Code (ECC); Secure L0/L1 RAM with ECC; L2/L3 RAM with Parity Error Checking; and M0/M1 with ECC. All local memories are accessible from the C28x CPU; the L2/L3 RAM is also accessible by the C28x DMA. Two types of error correction events can be generated during access of the C28x Local Memory: uncorrectable errors and single errors. The uncorrectable errors propagate to the NMI block where they can become the C28NMI to the C28x NMI Watchdog and the C28NMIINT non-maskable interrupt to the C28x CPU. The less critical single errors go to the PIE block where they can become maskable interrupts to the C28x CPU.

ADVANCE INFORMATION

2.4.7 C28x Accessing Shared Resources and Analog Peripherals

There are several memories, digital peripherals, and analog peripherals that can be accessed by both the Master and Control Subsystems. They are grouped into the Shared Resources and the Analog Subsystem.

The Shared Resources include the External Peripheral Interface (EPI), Inter-Processor Communications (IPC) registers, MTOC Message RAM, CTOM Message RAM, and eight individually configurable Shared RAM blocks.

The Message RAMs and the Shared RAMs can be accessed by the C28x CPU and DMA and have Parity-Error Checking. The MTOC Message RAM is intended for sending data from the Master Subsystem to the Control Subsystem, having r/w access for the Cortex™-M3/µDMA and read-only access for the C28x/DMA. The CTOM Message RAM is intended for sending data from the Control Subsystem to the Master Subsystem, having r/w access for the C28x/DMA and read-only access for the Cortex™-M3/µDMA.

The IPC registers provide up to 32 handshaking channels to coordinate transfer of data through the Message RAMs by polling. Four of these channels are also backed up by four interrupts to PIE on the Control Subsystem side, and four interrupts to the NVIC on the Master Subsystem side (to reduce delays associated with polling).

The eight Shared RAM blocks are similar to the Message RAMs, in that the data flow is only one way; however, the direction of the data flow can be individually set for each block to be from Master to Control Subsystem or from Control to Master Subsystem.

See Section 6.1 for more information on shared resources and analog peripherals.

2.5 Analog Subsystem

The Analog Subsystem has ADC1, ADC2, and six Analog Comparator + DAC units that can be accessed via the Analog Common Interface Bus. The ADC Result Registers are accessible by CPUs and DMAs of the Master and Control Subsystems. All other Analog Peripheral Registers are accessible by the C28x CPU only. The C28x CPU accesses the ACIB through the C28x Memory Bus, and the C28x DMA through the C28x DMA Bus. The ACIB arbitrates for access to ADC and Analog Comparator registers between CPU/DMA bus cycles of the C28x Subsystem with those of the Cortex™-M3 Subsystem. In addition to managing bus cycles, the ACIB also transfers Start-Of-Conversion triggers to the Analog Subsystem and returns End-Of-Conversion ADC interrupts to both the Master Subsystem and the Control Subsystem.

There are 22 possible SOC (Start-Of-Conversion) sources from the C28x Subsystem that are mapped to a total of 8 possible SOC triggers inside the Analog Subsystem (to ADC1 and ADC2).

Going the other way, eight EOC (End-Of-Conversion) sources from ADC1 and eight EOC sources from ADC2 are AND-ed together to form eight interrupts going to destinations in both the Master and Control Subsystems. Inside the C28x Subsystem, all eight EOC interrupts go to the PIE, but only four of the same eight go to the C28x DMA.

The Concerto™ MCU Analog Subsystem has two independent Analog-to-Digital Converters (ADC1, ADC2); six Analog Comparators + DAC units; and an Analog Common Interface Bus (ACIB) to facilitate analog data communications with Concerto's two digital subsystems (Cortex™-M3 and C28x).

Figure 2-3 shows the Analog Subsystem.

2.5.1 ADC1

The ADC1 consists of a 12-bit Analog-to-Digital converter with up to 16 analog input channels of which 12 are currently pinned out. The analog channels are internally pre-assigned to two Sample-and-Hold (S/H) units A and B, both feeding an Analog Mux whose output is converted to a 12-bit digital value and stored in ADC1 result registers. The two S/H units enable simultaneous sampling of two analog signals at a time. Additional channels or channel pairs are converted sequentially. Start-of-Conversion (SOC) triggers from the Control Subsystem initiate analog-to-digital conversions. End-of-Conversion (EOC) interrupts from ADCs notify the Master and Control Subsystems that the conversion results are ready to be read from ADC1 result registers.

See Section 6.1.1 for more information on ADC peripherals.

2.5.2 ADC2

The ADC2 consists of a 12-bit Analog-to-Digital converter with up to 16 analog input channels of which 12 are currently pinned out. The analog channels are internally preassigned to two Sample-and-Hold (S/H) units A and B, both feeding an Analog Mux whose output is converted to a 12-bit digital value and stored in the ADC2 result registers. The two S/H units enable simultaneous sampling of two analog signals at a time. Additional channels or channel pairs are converted sequentially. Start-of-Conversion (SOC) triggers from the Control Subsystem initiate analog-to-digital conversions. End-of-Conversion (EOC) interrupts from ADCs notify the Master and Control Subsystems that the conversion results are ready to be read from ADC2 result registers.

See Section 6.1.1 for more information on ADC peripherals.

ADVANCE INFORMATION

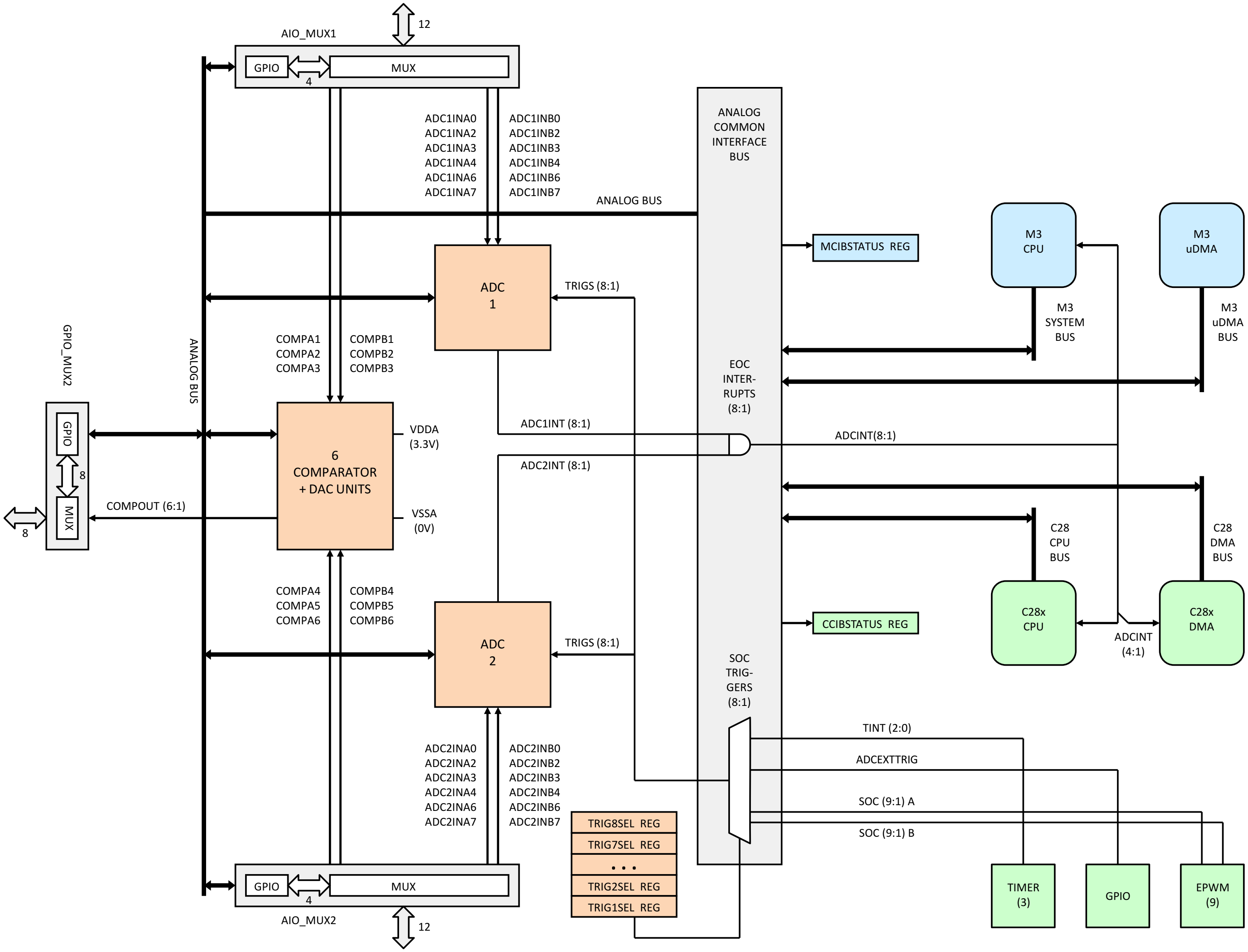


Figure 2-3. Analog Subsystem

2.5.3 Analog Comparator + DAC

There are six Comparator blocks enabling simultaneous comparison of multiple pairs of analog inputs, resulting in six digital comparison outputs. The external analog inputs that are being compared in the comparators come from AIO_MUX1 and AIO_MUX2 blocks. These analog inputs can be compared against each other or the outputs of 10-bit DACs (Digital-to-Analog Converters) inside individual Comparator modules. The six comparator outputs go to the GPIO_MUX2 block where they can be mapped to six out of eight available pins.

Note that in order to use these comparator outputs to trip the C28x EPWMA/B outputs, they must be first routed externally from pins of the GPIO_MUX2 block to selected pins of the GPIO_MUX1 block before they can be assigned to selected 12 ePWM Trip Inputs.

See Section 6.1.2 for more information on the analog comparator + DAC.

2.5.4 Analog Common Interface Bus (ACIB)

The ACIB links the Master and Control Subsystems with the Analog Subsystem. The ACIB enables the Cortex™-M3 CPU/µDMA and C28x CPU/DMA to access Analog Subsystem registers, to send SOC Triggers to the Analog Subsystem, and to receive EOC Interrupts from the Analog Subsystem. The Cortex™-M3 uses its System Bus and the µDMA Bus to read from ADC Result registers. The C28x uses its Memory Bus and the DMA bus to access ADC Result registers and other registers of the Analog Subsystem. The ACIB arbitrates between up to four possibly simultaneously occurring bus cycles on the Master/Control Subsystem side of ACIB to access the ADC and Analog Comparator registers on the Analog Subsystem side.

Additionally, ACIB maps up to 22 SOC trigger sources from the Control Subsystem to 8 SOC trigger destinations inside the Analog Subsystem (shared between ADC1 and ADC2), and up to 16 ADC EOC interrupt sources from the Analog Subsystem to 8 destinations inside the Master and Control Subsystems. The eight ADC interrupts are the result of AND-ing of eight EOC interrupts from ADC1 with 8 EOC interrupts from ADC2. The total of 16 possible ADC1 and ADC2 interrupts are sharing the 8 interrupt lines because it is unlikely that any application would need all 16 interrupts at the same time.

Eight registers (TRIG1SEL–TRIG8SEL) configure eight corresponding SOC triggers to assign 1 of 22 possible trigger sources to each SOC trigger.

There are two registers that provide status of ACIB to the Master Subsystem and to the Control Subsystem.

The Cortex™-M3 can read the MCIBSTATUS register to verify that the Analog Subsystem is properly powered up; the Analog System Clock (ASYSCLK) is present; and that the bus cycles, triggers, and interrupts are correctly propagating between the Master, Control, and Analog subsystems.

The C28x can read the CCIBSTATUS register to verify that the Analog Subsystem is properly powered up; the Analog System Clock (ASYSCLK) is present; and that the bus cycles, triggers, and interrupts are correctly propagating between the Master, Control, and Analog subsystems.

2.6 Master Subsystem NMIs

The Cortex™-M3 NMI Block generates an M3NMIINT non-maskable interrupt to the Cortex™-M3 CPU and an M3NMI event to the NMI Watchdog in response to potentially critical conditions existing inside or outside the Concerto™ MCU. When able to respond to the M3NMIINT interrupt, the Cortex™-M3 CPU may address the NMI condition and disable the NMI Watchdog. Otherwise, the NMI Watchdog counts out and an M3NMIRST reset signal is sent to the Resets block.

The inputs to the Cortex™-M3 NMI block include the C28NMIRST, PIENMIERR, CLOCKFAIL, ACIBERR, EXTGPIO, MLBISTERR, and CLBISTERR signals. The C28NMIRST comes from the C28x NMI Watchdog; C28NMIRST indicates that the C28x was not able to prevent the C28x NMI Watchdog counter from counting out. PIENMIERR indicates that an error condition was generated during the NMI vector fetch from the C28x Peripheral Interrupt Expansion (PIE) block. The CLOCKFAIL input comes from the Master Clocks Block, announcing a missing clock source to the Main Oscillator. ACIBERR indicates an abnormal condition inside the Analog Common Interface Bus. EXTGPIO comes from the GPIO_MUX1 to announce an external emergency. MLBISTERR is generated by the Cortex™-M3 core to signal that a BIST time-out or signature mismatch error has been detected. CLBISTERR is generated by the C28x core to signal that a BIST time-out or signature mismatch error has been detected.

The Cortex™-M3 NMI block can be accessed via the Cortex™-M3 NMI configuration registers—including the MNMIFLG, MNMIFLGCLR, and MNMIFLGFRC registers—to examine flag bits for the NMI sources, clear the flags, and force the flags to active state, respectively.

Figure 2-4 shows the Cortex™-M3 NMI and C28x NMI.

2.7 Control Subsystem NMIs

The C28x NMI Block generates a C28NMIINT non-maskable interrupt to the C28x CPU and a C28NMI event to the C28x NMI Watchdog in response to potentially critical conditions existing inside the Concerto™ MCU. When able to respond to the C28NMIINT interrupt, the C28x CPU may address the NMI condition and disable the C28x NMI Watchdog. Otherwise, the C28x NMI Watchdog counts out and the C28NMIRST reset signal is sent to the Resets block and the Cortex™-M3 NMI Block, where the Cortex™-M3 NMI Block can generate an NMI to the Cortex™-M3 processor.

The inputs to the C28x NMI block include the CLOCKFAIL, ACIBERR, RAMUNCERR, FLASHUNCERR, PIENMIERR, CLBISTERR, and MLBISTERR signals. The CLOCKFAIL input comes from the Clocks Block, announcing a missing clock source to the Main Oscillator. ACIBERR indicates an abnormal condition inside the Analog Common Interface Bus. The RAMUCERR and FLASHUNCERR announce the occurrence of uncorrectable error conditions during access to the Flash or RAM (local or shared). PIENMIERR indicates that an error condition was generated during NMI vector fetch from the C28x Peripheral Interrupt Expansion (PIE) block. MLBISTERR is generated by the Cortex™-M3 core to signal that a BIST time-out or signature mismatch error has been detected. CLBISTERR is generated by the C28x core to signal that a BIST time-out or signature mismatch error has been detected.

The C28x NMI block can be accessed via the C28x NMI configuration registers—including the CNMIFLG, CNMIFLGCLR, and CNMIFLGFRC registers—to examine flag bits for the NMI sources, clear the flags, and force the flags to active state, respectively.

Figure 2-4 shows the Cortex™-M3 NMI and C28x NMI.

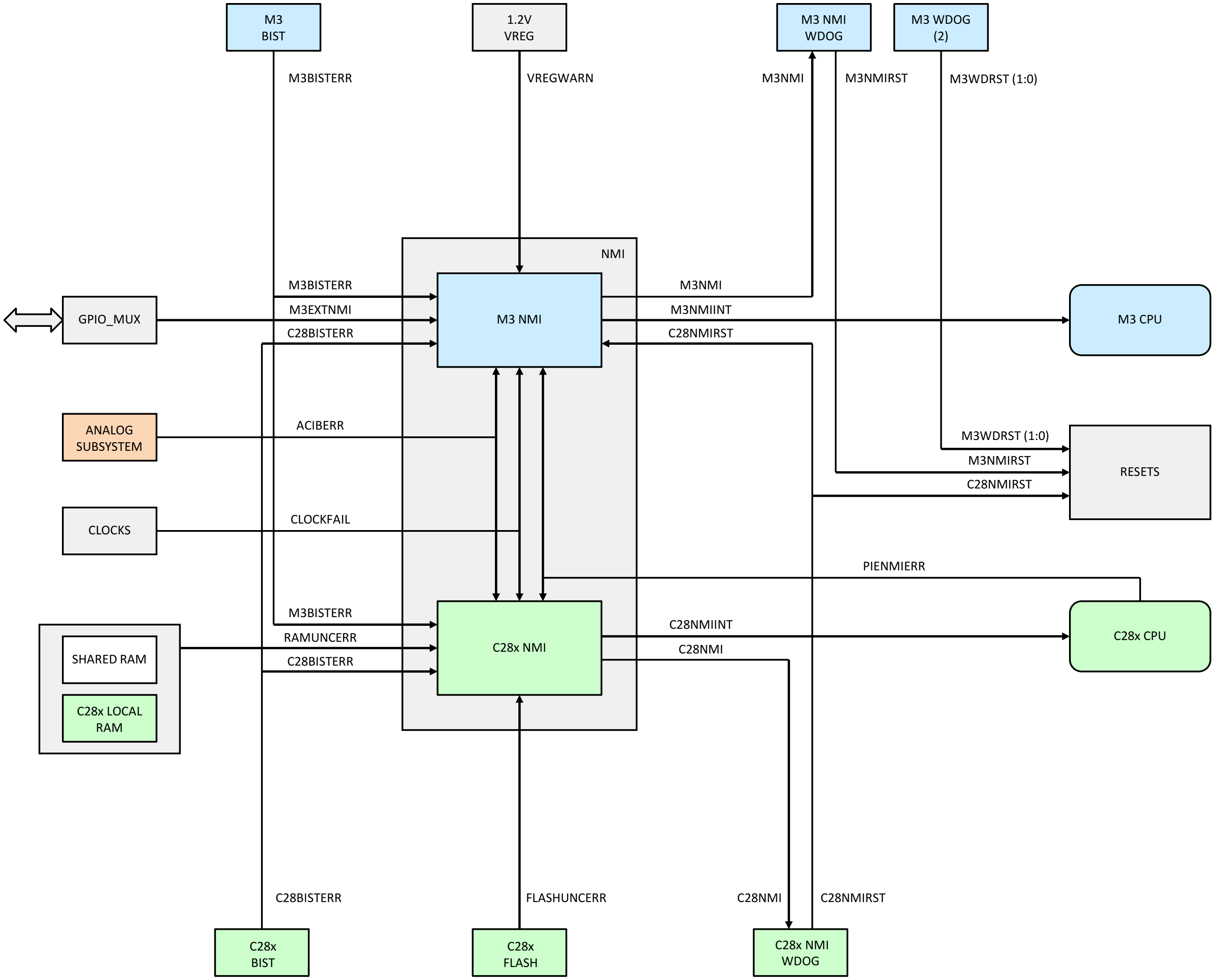


Figure 2-4. Cortex™-M3 NMI and C28x NMI

2.8 Resets

The Concerto™ MCU has two external reset pins: $\overline{XRS}$ for the Master and Control Subsystems and $\overline{ARS}$ for the Analog Subsystem. TI recommends that these two pins be externally tied together with a board signal trace.

The $\overline{XRS}$ pin can receive an external reset signal from outside into the chip, and the pin can drive a reset signal out from inside of the chip. A reset pulse driven into the $\overline{XRS}$ pin resets the Master and Control Subsystems. A reset pulse can also be driven out of the $\overline{XRS}$ pin by the Power-On Reset (POR) block of the Master and Control Subsystems (see Section 2.9). A reset pulse can be driven out of the $\overline{XRS}$ pin when the two Cortex™-M3 Watchdogs or the Cortex™-M3 NMI Watchdog time out.

There are some requirements on the $\overline{XRS}$ pin:

1. During power up, the $\overline{XRS}$ pin must be held low for at least eight X1 cycles after the input clock is stable. This requirement is to enable the entire device to start from a known condition.
2. During power down, the $\overline{XRS}$ pin must be pulled low at least 8 µs prior to V_{DDIO} reaching 1.5 V. This requirement is to enhance Flash reliability.
3. TI recommends that no voltage larger than 0.7 V be applied to any pin prior to powering up the device. Voltages applied to pins on an unpowered device can lead to unpredictable results.

The $\overline{ARS}$ pin can receive an external reset signal from outside into the chip, and the pin can drive a reset signal out from inside of the chip. A reset pulse driven into the $\overline{ARS}$ pin resets the Analog Subsystem. A reset pulse can be driven out of the $\overline{ARS}$ pin by the Power-On Reset (POR) block of the Analog Subsystem.

Figure 2-5 shows the resets.

2.8.1 *Cortex™-M3 Resets*

The Cortex™-M3 CPU and NVIC (Nested Vectored Interrupt Controller) are both reset by the POR (Power-On Reset) or the M3SYSRST reset signal. In both cases, the Cortex™-M3 CPU restarts program execution from the address provided by the reset entry in the vector table. A register can later be referenced to determine the source of the reset. The M3SYSRST signal also propagates to the Cortex™-M3 peripherals and the rest of the Cortex™-M3 Subsystem.

The $\overline{M3SYSRST}$ has four possible sources: $\overline{XRS}$, $\overline{M3WDOGS}$, $\overline{M3SWRST}$, and $\overline{M3DBGRST}$. The $\overline{M3WDOGS}$ is set in response to time-out conditions of the two Cortex™-M3 Watchdogs or the Cortex™-M3 NMI Watchdog. The $\overline{M3SWRST}$ is a software-generated reset output by the NVIC. The M3DBGRS is a debugger-generated reset that is also output by the NVIC. In addition to driving M3SYSRST, these two resets also propagate to the C28x Subsystem and the Analog Subsystem.

The M3RSNIN bit can be set inside the CRESCNF register to selectively reset the C28x Subsystem from the Cortex™-M3, and ACIBRST bit of the same register selectively resets the Analog Common Interface Bus. In addition to driving reset signals to other parts of the chip, the Cortex™-M3 can also detect a $\overline{C28SYSRST}$ reset being set inside the C28x Subsystem by reading the CRES bit of the CRESSTS register.

Cortex™-M3 software can also set bits in the SRCR register to selectively reset individual Cortex™-M3 peripherals, provided they are enabled inside the DC (Device Configuration) register. The Reset Cause register (MRESC) can be read to find out if the latest reset was caused by External Reset, POR, Watchdog Timer 0, Watchdog Timer 1, or Software Reset from NVIC.

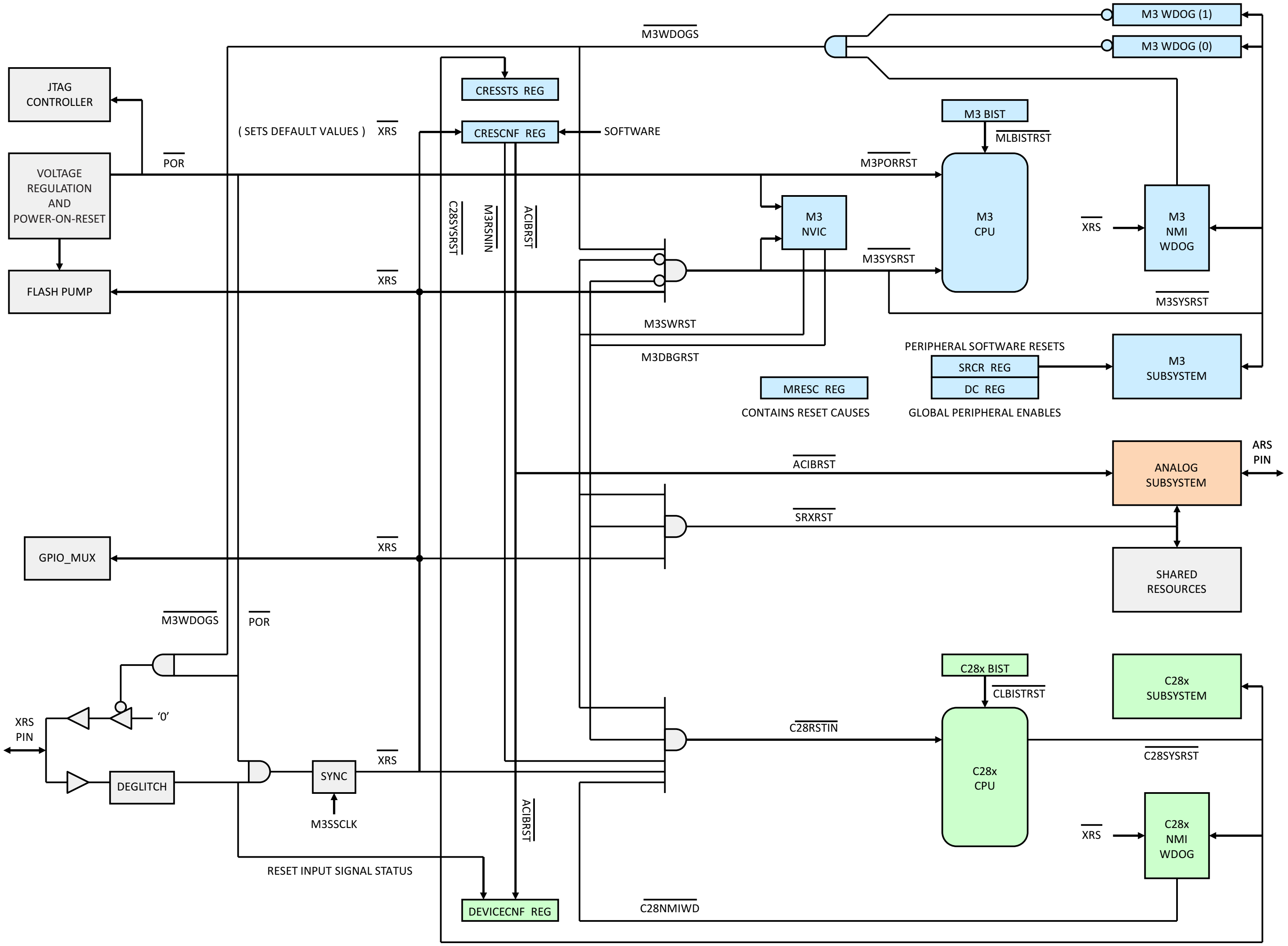


Figure 2-5. Resets

2.8.2 C28x Resets

The C28x CPU is reset by the C28RSTIN signal, and the C28x CPU in turn resets the rest of the C28x Subsystem with the C28SYSRST signal. When reset, the C28x restarts program execution from the address provided at the top of the Boot ROM Vector Table.

The C28RSTIN has five possible sources: XRS, C28NMIWD, M3SWRST, M3DBGRST, and the M3RSNIN. The C28NMIWD is set in response to time-out conditions of the C28x NMI Watchdog. The M3SWRST is a software-generated reset output by the NVIC. The M3DBGRS is a debugger-generated reset that is also output by the NVIC. These two resets must be first enabled by the Cortex™-M3 processor in order to propagate to the C28x Subsystem. M3RSNIN reset comes from the Cortex™-M3 Subsystem to selectively reset the C28x Subsystem from Cortex™-M3 software.

The C28x processor can learn the status of the internal ACIBRST reset signal and the external XRS pin by reading the DEVICECNF register.

2.8.3 Analog Subsystem and Shared Resources Resets

Both the Analog Subsystem and the resources shared between the C28x and Cortex™-M3 subsystems (IPC, MSG RAM, Shared RAM) are reset by the SRXRST reset signal. Additionally, the Analog Subsystem is also reset by the internal ACIBRST signal from the Cortex™-M3 Subsystem and the external ARS pin, (should be externally tied to the XRS pin), which can be reset by the POR circuitry.

The SRXRST has three possible sources: XRS, M3SWRST, and M3DBGRST. The M3SWRST is a software-generated reset output by the NVIC. The M3DBGRS is a debugger-generated reset that is also output by the NVIC. These two resets must be first enabled by the Cortex™-M3 processor in order to propagate to the Analog Subsystem and the Shared Resources.

Although EPI is a shared peripheral, it is physically located inside the Cortex™-M3 Subsystem; therefore, EPI is reset by M3SYSRST.

2.8.4 Device Boot Sequence

Concerto's boot sequence is used to configure the Master Subsystem and the Control Subsystem for execution of application code. The boot sequence involves both internal resources, and resources external to the device. These resources include: Master Subsystem Bootloader code (M-Bootloader) factory-programmed inside the Master Subsystem Boot ROM (M-Boot ROM); Control Subsystem Bootloader code (C-Bootloader) factory-programmed inside the Control Subsystem Boot ROM (C-Boot ROM); four GPIO_MUX pins for Master boot mode selection; internal Flash and RAM memories; and selected Cortex™-M3 and C28x peripherals for loading the application code into the Master and Control Subsystems.

The boot sequence starts when the Master Subsystem comes out of reset, which can be caused by device power up, external reset, debugger reset, software reset, Cortex™-M3 watchdog reset, or Cortex™-M3 NMI watchdog reset. While the M-Bootloader starts executing first, the C-Bootloader starts soon after, and then both bootloaders work in tandem to configure the device, load application code for both processors (if not already in the Flash), and branch the execution of each processor to a selected location in the application code.

Execution of the M-Bootloader commences when an internal reset signal goes from active to inactive state. At that time, the Control Subsystem and the Analog Subsystem continue to be in reset state until the Master Subsystem takes them out of reset. The M-Bootloader first initializes some device-level functions, then the M-Bootloader initializes the Master Subsystem. Next, the M-Bootloader takes the Control Subsystem and the Analog Subsystem/ACIB out of reset. When the Control Subsystem comes out of reset, its own C-Bootloader starts executing in parallel with the M-Bootloader. After initializing the Control Subsystem, the C-Bootloader enters the C28x processor into the idle mode (to wait for the M-Bootloader to wake up the C28x processor later via the MTOCIPC1 interrupt). Next, the M-Bootloader reads four GPIO pins (see Table 2-17) to determine the boot mode for the rest of the M-Bootloader operation.

Table 2-17. Master Subsystem Boot Mode Selection

Boot Mode #	Master Subsystem Boot Modes	PF2_GPIO34 (BOOT_3)(1)	PF3_GPIO35 (BOOT_2)(1)	PG7_GPIO47 (BOOT_1)(1)	PG3_GPIO43 (BOOT_0)(1)
0(2)	Boot from Parallel GPIO	0	0	0	0
1(2)	Boot to Master Subsystem RAM	0	0	0	1
2(2)	Boot from Master Subsystem serial peripherals (UART0/SSI0/I2C0)	0	0	1	0
3(2)	Boot from Master Subsystem CAN interface	0	0	1	1
4(2)	Boot from Master Subsystem Ethernet interface	0	1	0	0
5(2)(4)	Not supported (Defaults to Boot-to-Flash), future boot from Cortex™-M3 USB	0	1	0	1
6(2)(4)	Not supported (Defaults to Boot-to-Flash)	0	1	1	0
7(2)(4)	Boot to Master Subsystem Flash memory	0	1	1	1
8	Not supported (Defaults to Boot-to-Flash)	1	0	0	0
9(4)	Boot from Master Subsystem serial peripheral – SSI0 Master	1	0	0	1
10(4)	Boot from Master Subsystem serial peripheral – I2C0 Master	1	0	1	0
11(4)	Not supported (Defaults to Boot-to-Flash)	1	0	1	1
12(3)	Boot from Master Subsystem Ethernet interface	1	1	0	0
13(4)	Not supported (Defaults to Boot-to-Flash)	1	1	0	1
14(4)	Not supported (Defaults to Boot-to-Flash)	1	1	1	0
15(4)	Boot to Master Subsystem Flash memory	1	1	1	1

(1) By default, GPIO terminals are not pulled up (they are floating).

(2) Boot Modes 0–7 are pin-compatible with the F28M35x members of the Concerto family (they use same GPIO terminals).

(3) Boot Mode 12 is the same as Boot Mode 4, except it uses a different set of GPIO terminals.

(4) This Boot Mode uses a faster Flash power-up sequence. The maximum supported OSCCLK frequency for this mode is 30 MHz.

Boot Mode 7 and Boot Mode 15 cause the Master program to branch execution to the application in the Master Flash memory. This branching requires that the Master Flash be already programmed with valid code; otherwise, a hard fault exception is generated and the Cortex™-M3 goes back to the above reset sequence. (Therefore, for a factory-fresh device, the M-Bootloader will be in a continuous reset loop until the emulator is connected and a debug session started.) If the Master Subsystem Flash has already been programmed, the application code will start execution. Typically, the Master Subsystem application code will then establish data communication with the C28x [through the IPC (Interprocessor Communications peripheral)] to coordinate the rest of the boot process with the Control Subsystem. Boot Mode 15 (Fast Boot to Flash Mode) supported on this device is a special boot to Flash mode, which configures Flash for a faster power up, thus saving some boot time. Boot Mode 7 and other modes which default to Flash do not configure Flash for a faster power up like Boot Mode 15 does. Note that following reset, the internal pullup resistors on GPIOs are disabled. Therefore, Boot Mode 15, for example, will typically require four external pullups.

Boot Mode 1 causes the Master boot program to branch to Cortex™-M3 RAM, where the Cortex™-M3 processor starts executing code that has been preloaded earlier. Typically, this mode is used during development of application code meant for Flash, but which has to be first tested running out of RAM. In this case, the user would typically load the application code into RAM using the debugger, and then issue a debugger reset, while setting the four boot pins to 0001b. From that point on, the rest of the boot process on the Master Subsystem side is controlled by the application code.

Boot Modes 0, 2, 3, 4, 9, 10, and 12 are used to load the Master application code from an external peripheral before branching to the application code. This process is different from the process in Boot Modes 1, 7, and 15, where the application code was either already programmed in Flash or loaded into RAM by the emulator. If the boot mode selection pins are set to 0000b, the M-Bootloader (running out of M-Boot ROM) will start uploading the Master application code from preselected Parallel GPIO_MUX pins. If the boot pins are set to 0010b, the application code will be loaded from the Master Subsystem UART0, SSI0, or I2C0 peripheral. (SSI0 and I2C0 are configured to work in Slave mode in this Boot Mode.) If the boot pins are set to 0011b, the application code will be loaded from the Master Subsystem CAN interface. Furthermore, if the boot pins are set to 0100b, the application code will be loaded through the Master Subsystem Ethernet interface; the IOs used in this Boot Mode are compatible with the F28M35x device. If the boot pins are set to 1001b or 1010b, then the application code will be loaded through the SSI0 or I2C0 interface, respectively. SSI0 and I2C0 loaders work in Master Mode in this boot mode. If the boot pins are set to 1100b, then the application code will be loaded through the Master Subsystem Ethernet interface; the IOs used in this Boot Mode are F28M36x IOs, which are available only in a BGA package.

Regardless of the type of boot mode selected, once the Master application code is resident in Master Flash or RAM, the next step for the M-Bootloader is to branch to Master Flash or RAM. At that point, the application code takes over control from the M-Bootloader, and the boot process continues as prescribed by the application code. At this stage, the Master application program typically establishes communication with the C-Bootloader, which by now, would have already initialized the Control Subsystem and forced the C28x to go into Idle mode. To wake the Control Subsystem out of Idle mode, the Master application issues the Master-to-Control-IPC-interrupt 1 (MTOCIPCINT1) . Once the data communication has been established through the IPC, the boot process can now also continue on the Control Subsystem side.

The rest of the Control Subsystem boot process is controlled by the Master Subsystem application issuing IPC instructions to the Control Subsystem, with the C-Bootloader interpreting the IPC commands and acting on them to continue the boot process. At this stage, a boot mode for the Control Subsystem can be established. The Control Subsystem boot modes are similar to the Master Subsystem boot modes, except for the mechanism by which they are selected. The Control Subsystem boot modes are chosen through the IPC commands from the Master application code to the C-Bootloader, which interprets them and acts accordingly. The choices are, as above, to branch to already existing Control application code in Flash, to branch to preloaded code in RAM (development mode), or to upload the Control application code from one of several available peripherals (see Table 2-18). As before, once the Control application code is in place (in Flash or RAM), the C-Bootloader branches to Flash or RAM, and from that point on, the application code takes over.

ADVANCE INFORMATION

Table 2-18. Control Subsystem Boot Mode Selection

Control Subsystem Boot Modes	MTOCIPCBOOTMODE Register Value	Description
BOOT_FROM_RAM	0x0000 0001	Upon receiving this command from the Master Subsystem, C-Boot ROM will branch to the Control Subsystem RAM entry point location and start executing code from there.
BOOT_FROM_FLASH	0x0000 0002	Upon receiving this command, C-Boot ROM will branch to the Control Subsystem FLASH entry point and start executing code from there.
BOOT_FROM_SCI	0x0000 0003	Upon receiving this command, C-Boot ROM will boot from the Control Subsystem SCI peripheral.
BOOT_FROM_SPI	0x0000 0004	Upon receiving this command, C-Boot ROM will boot from the Control Subsystem SPI interface.
BOOT_FROM_I2C	0x0000 0005	Upon receiving this command, C-Boot ROM will boot from the Control Subsystem I2C interface.
BOOT_FROM_PARALLEL	0x0000 0006	Upon receiving this command, C-Boot ROM will boot from the Control Subsystem GPIO.
BOOT_FROM_SPI (1)	0x0000 0007	Upon receiving this command, C-Boot ROM will boot from the Control Subsystem SPI interface.

(1) MTOCBOOTMODE 0x0000 0001–MTOCBOOTMODE 0x0000 0006 are compatible with the F28M35x members of the Concerto family, but MTOCBOOTMODE 0x0000 0007 uses GPIO terminals that are not available on the F28M35x.

The boot process can be considered completed once the Cortex™-M3 and C28x are both running out of their respective application programs. Note that following the boot sequence, the C-Bootloader is still available to interpret and act upon an assortment of IPC commands that can be issued from the Master Subsystem to perform a variety of configuration, housekeeping, and other functions.

2.9 Internal Voltage Regulation and Power-On-Reset Functionality

While Concerto's analog functions draw power from a single dedicated external power source—V_{DDA}, its digital circuits are powered by three separate rails: 3.3-V V_{DDIO}, 1.8-V V_{DD18}, and 1.2-V V_{DD12}. This section describes the sourcing, regulation, and power-on-reset (POR) functionality for these three digital power rails.

Concerto devices can be internally divided into an Analog Subsystem and a Digital Subsystem (having the Cortex™-M3-based Master Subsystem and the C28x-based Control Subsystem). The Digital Subsystem uses V_{DD12} to power the two processors, internal memory, and peripherals. The Analog Subsystem uses V_{DD18} to power the digital logic associated with the analog functions. Both Digital and Analog Subsystems share a common V_{DDIO} rail to power their 3.3-V I/O buffers through which Concerto's digital signals communicate with the outside world.

The Analog and Digital Subsystems each have their own power-on-reset (POR) circuits that operate independently. With the $\overline{\text{ARS}}$ and $\overline{\text{XRS}}$ reset pins externally tied together, both systems can come out of reset together, and can also be put in reset together by driving both reset pins low. See Figure 2-6 for a snapshot of the voltage regulation and POR functions provided within Concerto's Analog and Digital Subsystems.

2.9.1 Analog Subsystem's Internal 1.8-V VREG

The internal 1.8-V Voltage Regulator (VREG) generates V_{DD18} power from V_{DDIO}. The 1.8-V VREG is enabled by pulling the $\overline{\text{VREG18EN}}$ pin to a low state. When enabled, the 1.8-V VREG provides 1.8 V to digital logic associated with the analog functions of the Analog Subsystem.

When the internal 1.8-V VREG function is enabled, the 1.8 V power no longer has to be provided externally; however, a 1.2-µF (10% tolerance) capacitor is required for each V_{DD18} pin to stabilize the internally generated voltages. These load capacitors are not required if the internal 1.8-V VREG is disabled, and the 1.8 V is provided from an external supply.

Note that while removing the need for an external power supply, enabling the internal VREG will increase the V_{DDIO} power consumption.

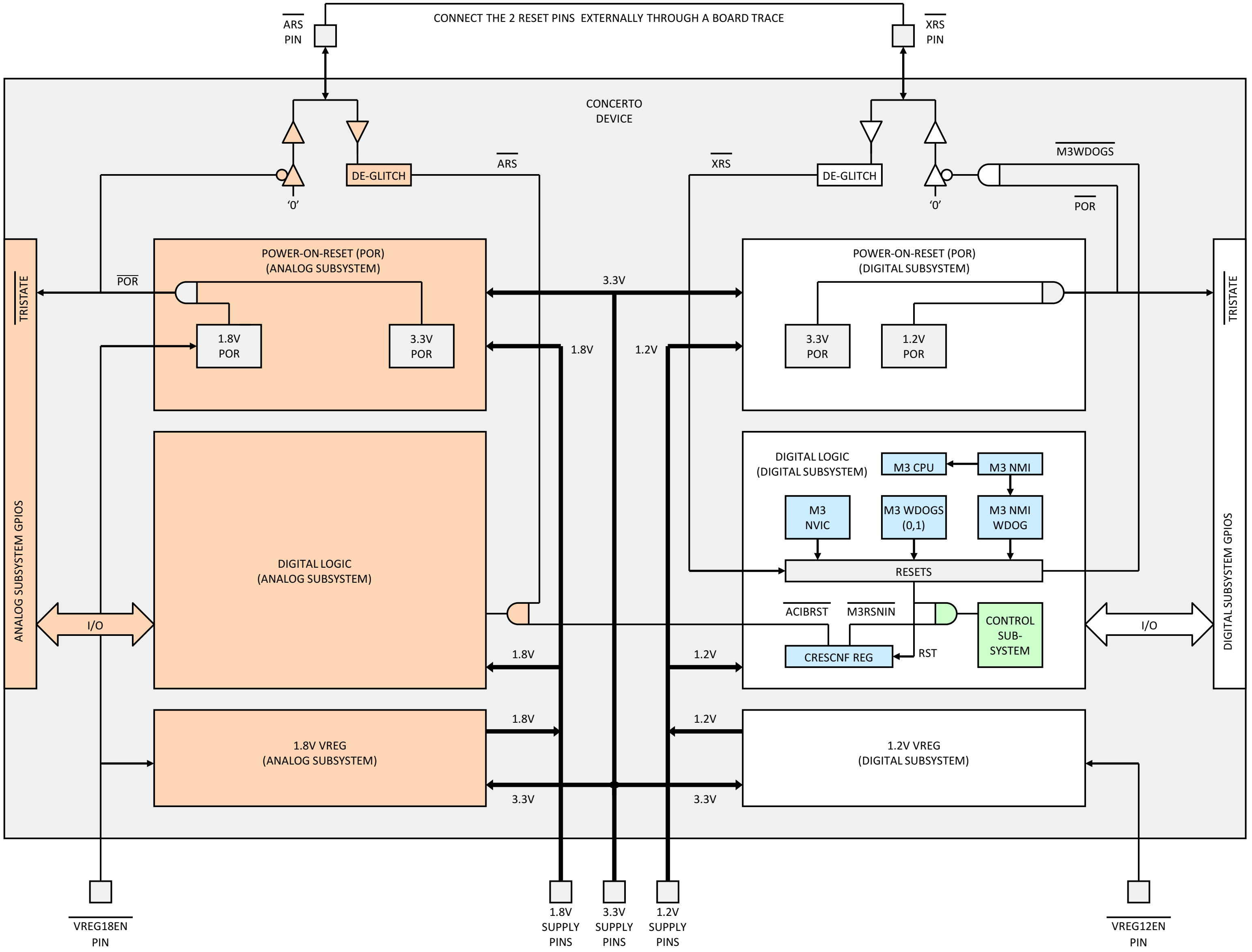


Figure 2-6. Voltage Regulation and Monitoring

2.9.2 Digital Subsystem's Internal 1.2-V VREG

The internal 1.2-V VREG generates V_{DD12} power from V_{DDIO}. The 1.2-V VREG is enabled by pulling the $\overline{\text{VREG12EN}}$ pin to a low state. When enabled, the 1.2-V VREG internally provides 1.2 V to digital logic associated with the processors, memory, and peripherals of the Digital Subsystem.

When the internal 1.2-V VREG function is enabled, the 1.2 V power no longer has to be provided externally; however, the minimum and maximum capacitance required for each V_{DD12} pin to stabilize the internally generated voltages are 250 nF and 750 nF, respectively. These load capacitors are not required if the internal 1.2-V VREG is disabled and the 1.2 V is provided from an external supply.

Note that while removing the need for an external power supply, enabling the internal VREG will increase the V_{DDIO} power consumption.

2.9.3 Analog and Digital Subsystems' Power-On-Reset Functionality

The Analog and Digital Subsystems' each have a Power-On-Reset (POR) circuit that creates a clean reset throughout the device enabling glitchless GPIOs during the power-on procedure. The POR function keeps both $\overline{\text{ARS}}$ and $\overline{\text{XRS}}$ driven low during device power up.

While in most applications, the POR generated reset has a long enough duration to also reset other system ICs, some applications may require a longer lasting pulse. In these cases, the $\overline{\text{ARS}}$ and $\overline{\text{XRS}}$ reset pins (which are open-drain) can also be driven low to match the time the device is held in a reset state with the rest of the system.

When POR drives the $\overline{\text{ARS}}$ and $\overline{\text{XRS}}$ pins low, the POR also resets the digital logic associated with both subsystems and puts the GPIO pins in a high impedance state.

In addition to the POR reset, the Digital Subsystem’s Resets block also receives reset inputs from the NVIC, the Cortex™-M3 Watchdogs (0, 1), and from the Cortex™-M3 NMI Watchdog. The resulting reset output signal is then fed back to the $\overline{\text{XRS}}$ pin after being AND-ed with the POR reset (see Figure 2-6).

On a related note, only the Master Subsystem comes out of reset immediately following a device power up. The Control and Analog Subsystems continue to be held in reset until the Master Processor (Cortex™-M3) brings them out of reset by writing a "1" to the M3RSNIN and ACIBRST bits of the CRESCNF Register (see Figure 2-6).

2.9.4 Connecting $\overline{\text{ARS}}$ and $\overline{\text{XRS}}$ Pins

In most Concerto applications, TI recommends that the $\overline{\text{ARS}}$ and $\overline{\text{XRS}}$ pins be tied together by external means such as through a signal trace on a PCB board. Tying the $\overline{\text{ARS}}$ and $\overline{\text{XRS}}$ pins together ensures that all reset sources will cause both the Analog and Digital Subsystems to enter the reset state together, regardless of where the reset condition occurs.

2.10 Input Clocks and PLLs

Concerto devices have multiple input clock pins from which all internal clocks and the output clock are derived. Figure 2-7 shows the recommended methods of connecting crystals, resonators, and oscillators to pins X1/X2 and XCLKIN.

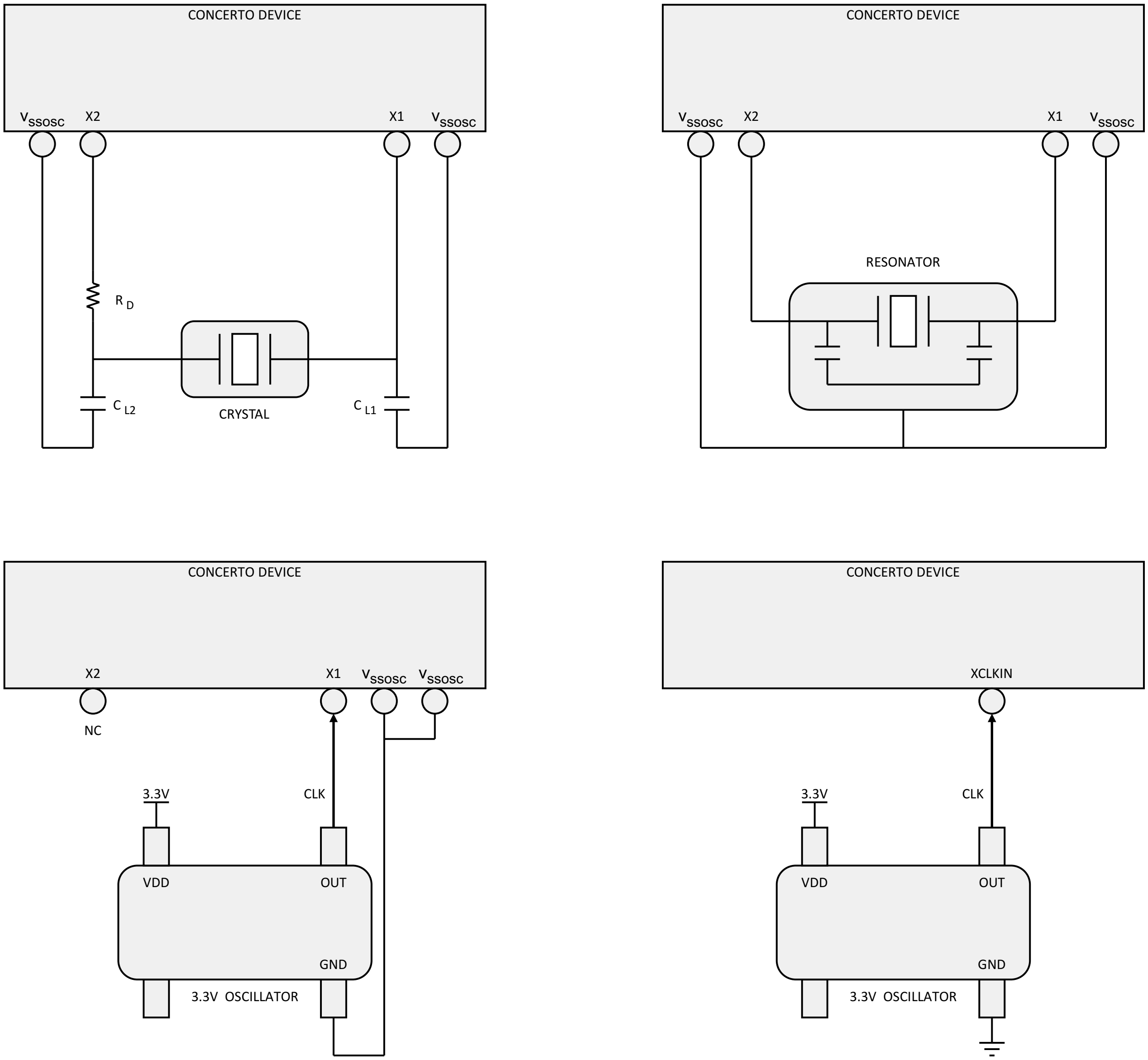


Figure 2-7. Connecting Input Clocks to a Concerto Device

2.10.1 Internal Oscillator (Zero-Pin)

Each Concerto device contains a zero-pin internal oscillator. This oscillator outputs two fixed-frequency clocks: 10MHZCLK and 32MHZCLK. These clocks are not configurable by the user. They are used inside the Master Subsystem to implement low-power modes. The 10MHZCLK is also used by the Missing Clock Detect circuit.

2.10.2 Crystal Oscillator/Resonator (Pins X1/X2 and V_{SSOSC})

The main oscillator circuit connects to an external crystal through pins X1 and X2. If a resonator is used (version of a crystal with built-in load capacitors), its ground terminal should be connected to the pin V_{SSOSC} (not board ground). The V_{SSOSC} pin should also be used to ground the external load capacitors connected to the two crystal terminals as shown in Figure 2-7.

2.10.3 External Oscillators (Pins X1, V_{SSOSC}, XCLKIN)

Concerto has two pins (X1 and XCLKIN) into which a single-ended clock can be driven from external oscillators or other clock sources. When connecting an external clock source through the X1 terminal, the X2 terminal should be left unconnected. Most internal clocks of this device are derived from the X1 clock input (or X1/X2 crystal) . The XCLKIN clock is only used by the USB PLL and CAN peripherals. Figure 2-7 shows how to connect external oscillators to the X1 and XCLKIN terminals.

When connecting an external oscillator through the X1 terminal, use good design practices to minimize EMI as well as clock jitter induced by external noise sources. Minimize the loop area formed between the forward current path (from the oscillator OUT terminal to the MCU X1 terminal) and the return path (from the MCU V_{SSOSC} terminal to the oscillator GND terminal). In this case, the external oscillator should be grounded only through the V_{SSOSC} pin.

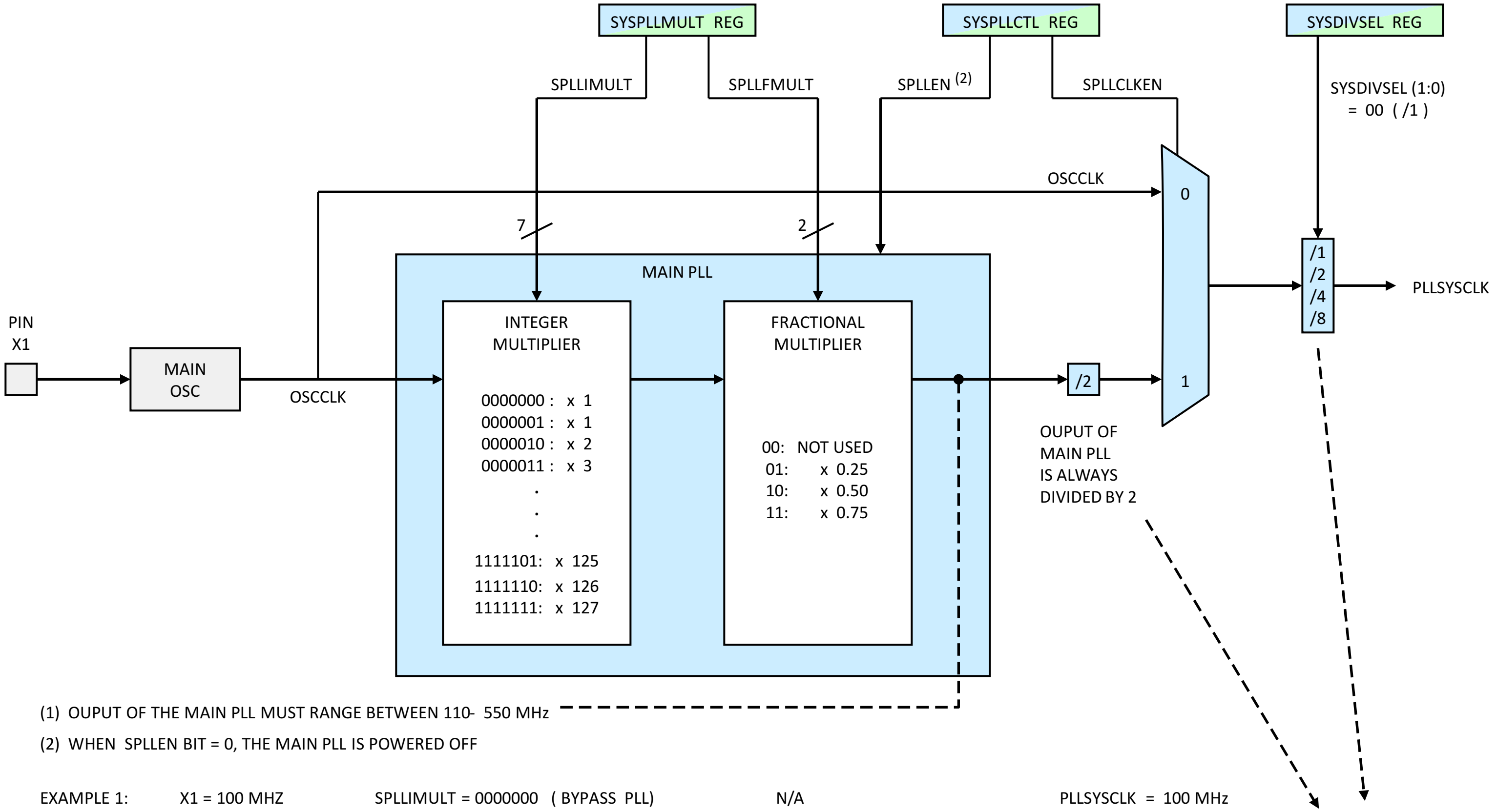
Locate the external oscillator as close to the MCU as practical. Ideally, the return ground trace should be an isolated trace directly underneath the forward trace or run adjacent to the trace on the same layer. Spacing should be kept minimal, with any other nearby traces double-spaced away, so that the electromagnetic fields created by the two opposite currents cancel each other out as much as possible, thus reducing parasitic inductances that radiate EMI.

While the above is the preferred method of connecting external oscillators to the X1 terminal, the ground pin of an oscillator or another clock source (for example, an FPGA) can also be connected to the board's ground plane, in which case the MCU V_{SSOSC} terminal should be left unconnected.

The XCLKIN terminal does not have a dedicated ground pin (like the V_{SSOSC} for the X1 pin); thus, when using an external oscillator to input clock through XCLKIN, the ground pin of that oscillator should be connected to the board ground (see Figure 2-7).

2.10.4 Main PLL

The Main PLL uses the reference clock from pins X1 (external oscillator) or X1/X2 (external crystal/resonator). The input clock is multiplied by an integer multiplier and a fractional multiplier as selected by the SPLLIMULT and SPLLFMULT fields of the SYSPLLMULT register. For example, to achieve PLL multiply of 28.5, the integer multiplier should be set to 28, and the fractional multiplier to 0.5. The output clock from the Main PLL must be between 110 MHz and 550 MHz. The PLL output clock is then divided by 2 before entering a mux that selects between this clock and the PLL input clock – OSCCLK (used in PLL bypass mode). The PLL bypass mode is selected by setting the SPLLIMULT field of the SYSPLLMULT register to 0. The output clock from the mux next enters a divider controlled by the SYSDIVSEL register, after which the output clock becomes the PLLSYSCLK. Figure 2-8 shows the Main PLL function and configuration examples. Table 2-19 to Table 2-22 list the integer multiplier configuration values.



(1) OUPUT OF THE MAIN PLL MUST RANGE BETWEEN 110- 550 MHz

(2) WHEN SPLLEN BIT = 0, THE MAIN PLL IS POWERED OFF

EXAMPLE 1:	X1 = 100 MHZ	SPLLIMULT = 0000000 (BYPASS PLL)	N/A	PLLSYSCLK = 100 MHz
EXAMPLE 2:	X1 = 10 MHz	SPLLIMULT = 0010100 (x 20)	SPLLFMULT = 00 (NOT USED)	PLLSYSCLK = [(10 x 20) / 2] / 1 = 100 MHz
EXAMPLE 3:	X1 = 10 MHz	SPLLIMULT = 0010100 (x 20)	SPLLFMULT = 10 (x 0.50)	PLLSYSCLK = [(10 x 20.5) / 2] / 1 = 102.5 MHz

Figure 2-8. Main PLL

Table 2-19. Main PLL Integer Multiplier Configuration (Bypass PLL to x 31)

SPLLIMULT(6:0)	MULT VALUE
0000000 b	Bypass PLL
0000001 b	x 1
0000010 b	x 2
0000011 b	x 3
0000100 b	x 4
0000101 b	x 5
0000110 b	x 6
0000111 b	x 7
0001000 b	x 8
0001001 b	x 9
0001010 b	x 10
0001011 b	x 11
0001100 b	x 12
0001101 b	x 13
0001110 b	x 14
0001111 b	x 15
0010000 b	x 16
0010001 b	x 17
0010010 b	x 18
0010011 b	x 19
0010100 b	x 20
0010101 b	x 21
0010110 b	x 22
0010111 b	x 23
0011000 b	x 24
0011001 b	x 25
0011010 b	x 26
0011011 b	x 27
0011100 b	x 28
0011101 b	x 29
0011110 b	x 30
0011111 b	x 31

Table 2-20. Main PLL Integer Multiplier Configuration (x 32 to x 63)

SPLLIMULT(6:0)	MULT VALUE
0100000 b	x 32
0100001 b	x 33
0100010 b	x 34
0100011 b	x 35
0100100 b	x 36
0100101 b	x 37
0100110 b	x 38
0100111 b	x 39
0101000 b	x 40
0101001 b	x 41
0101010 b	x 42
0101011 b	x 43
0101100 b	x 44
0101101 b	x 45
0101110 b	x 46
0101111 b	x 47
0110000 b	x 48
0110001 b	x 49
0110010 b	x 50
0110011 b	x 51
0110100 b	x 52
0110101 b	x 53
0110110 b	x 54
0110111 b	x 55
0111000 b	x 56
0111001 b	x 57
0111010 b	x 58
0111011 b	x 59
0111100 b	x 60
0111101 b	x 61
0111110 b	x 62
0111111 b	x 63

Table 2-21. Main PLL Integer Multiplier Configuration (x 64 to x 95)

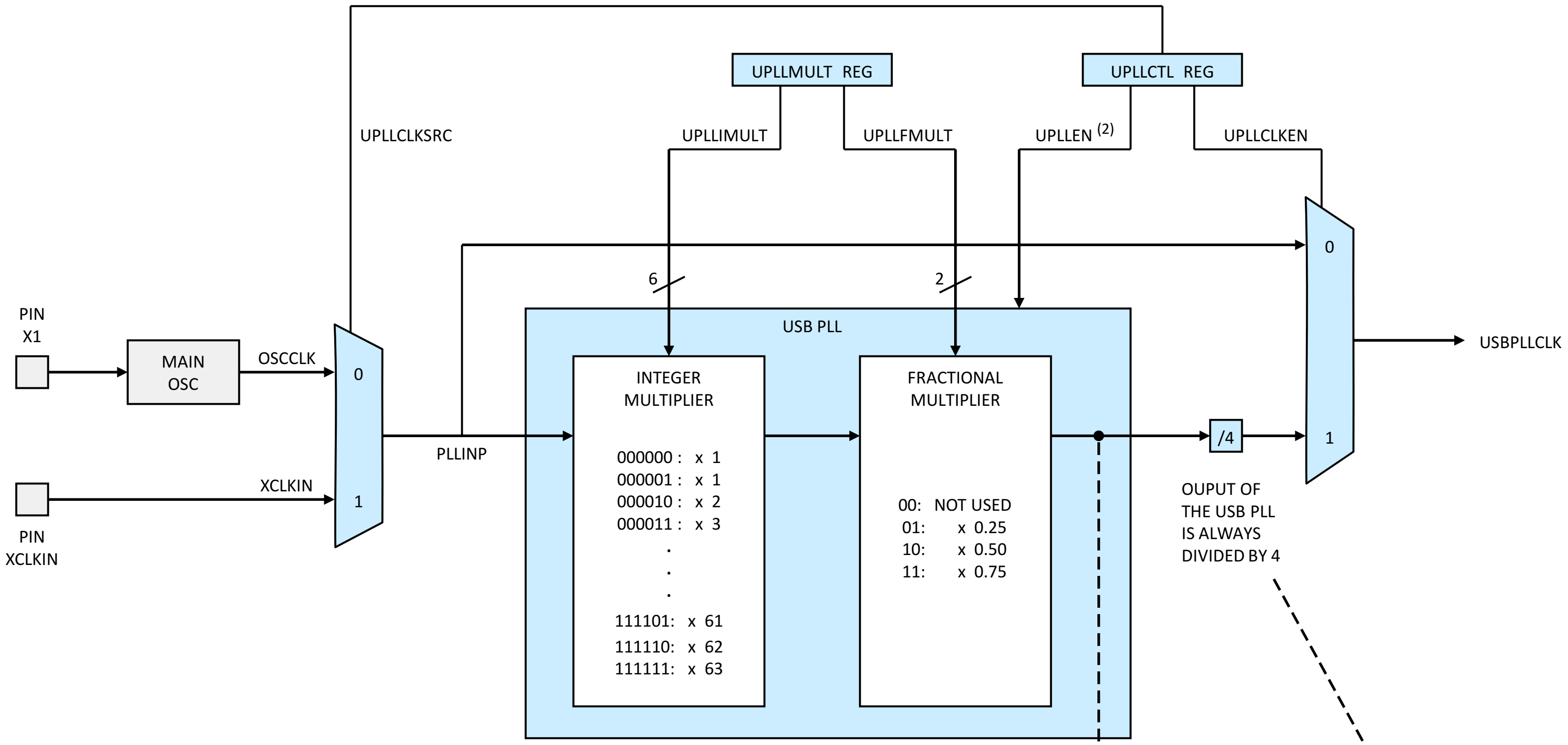
SPLLIMULT(6:0)	MULT VALUE
1000000 b	x 64
1000001 b	x 65
1000010 b	x 66
1000011 b	x 67
1000100 b	x 68
1000101 b	x 69
1000110 b	x 70
1000111 b	x 71
1001000 b	x 72
1001001 b	x 73
1001010 b	x 74
1001011 b	x 75
1001100 b	x 76
1001101 b	x 77
1001110 b	x 78
1001111 b	x 79
1010000 b	x 80
1010001 b	x 81
1010010 b	x 82
1010011 b	x 83
1010100 b	x 84
1010101 b	x 85
1010110 b	x 86
1010111 b	x 87
1011000 b	x 88
1011001 b	x 89
1011010 b	x 90
1011011 b	x 91
1011100 b	x 92
1011101 b	x 93
1011110 b	x 94
1011111 b	x 95

Table 2-22. Main PLL Integer Multiplier Configuration (x 96 to x 127)

SPLLIMULT(6:0)	MULT VALUE
1100000 b	x 96
1100001 b	x 97
1100010 b	x 98
1100011 b	x 99
1100100 b	x 100
1100101 b	x 101
1100110 b	x 102
1100111 b	x 103
1101000 b	x 104
1101001 b	x 105
1101010 b	x 106
1101011 b	x 107
1101100 b	x 108
1101101 b	x 109
1101110 b	x 110
1101111 b	x 111
1110000 b	x 112
1110001 b	x 113
1110010 b	x 114
1110011 b	x 115
1110100 b	x 116
1110101 b	x 117
1110110 b	x 118
1110111 b	x 119
1111000 b	x 120
1111001 b	x 121
1111010 b	x 122
1111011 b	x 123
1111100 b	x 124
1111101 b	x 125
1111110 b	x 126
1111111 b	x 127

2.10.5 USB PLL

The USB PLL uses the reference clock selectable between the input clock arriving at the XCLKIN pin, or the internal OSCCLK (originating from the external crystal or oscillator via the X1/X2 pins). An input mux selects the source of the USB PLL reference based on the UPLLCLKSRC bit of the UPLLCTL Register (see Figure 2-9). The input clock is multiplied by an integer multiplier and a fractional multiplier as selected by the UPLLIMULT and UPLLFMULT fields of the UPLLMULT register. For example, to achieve PLL multiply of 28.5, the integer multiplier should be set to 28, and the fractional multiplier to 0.5. The output clock from the USB PLL must always be 240 MHz. The PLL output clock is then divided by 4—resulting in 60 MHz that the USB needs—before entering a mux that selects between this clock and the PLL input clock (used in the PLL bypass mode). The PLL bypass mode is selected by setting the UPLLIMULT field of the UPLLMULT register to 0. The output clock from the mux becomes the USBPLLCLK (there is not another clock divider). Figure 2-9 shows the USB PLL function and configuration examples. Table 2-23 and Table 2-24 list the integer multiplier configuration values.



(1) OUPUT OF THE USB PLL MUST BE ALWAYS 240MHz (SO THAT USBPLLCLK IS 60MHZ)

(2) WHEN UPLLEN BIT = 0, THE USB PLL IS POWERED OFF

EXAMPLE 1:	X1 OR XCLKIN = 60 MHZ	UPLLIMULT = 000000 (BYPASS PLL)	N/A	PLLSYSCLK = 60 MHz	
EXAMPLE 2:	X1 OR XCLKIN = 10 MHz	UPLLIMULT = 011000 (x 24)	UPLLFMULT = 00 (NOT USED)	PLLSYSCLK = (10 x 24)	/ 4 = 60 MHz
EXAMPLE 3:	X1 OR XCLKIN = 64 MHz	UPLLIMULT = 000011 (x 3)	UPLLFMULT = 11 (x 0.75)	PLLSYSCLK = (64 x 3.75)	/ 4 = 60 MHz

Figure 2-9. USB PLL

ADVANCE INFORMATION

Table 2-23. USB PLL Integer Multiplier Configuration (Bypass PLL to x 31)

SPLLIMULT(5:0)	MULT VALUE
000000 b	Bypass PLL
000001 b	x 1
000010 b	x 2
000011 b	x 3
000100 b	x 4
000101 b	x 5
000110 b	x 6
000111 b	x 7
001000 b	x 8
001001 b	x 9
001010 b	x 10
001011 b	x 11
001100 b	x 12
001101 b	x 13
001110 b	x 14
001111 b	x 15
010000 b	x 16
010001 b	x 17
010010 b	x 18
010011 b	x 19
010100 b	x 20
010101 b	x 21
010110 b	x 22
010111 b	x 23
011000 b	x 24
011001 b	x 25
011010 b	x 26
011011 b	x 27
011100 b	x 28
011101 b	x 29
011110 b	x 30
011111 b	x 31

Table 2-24. USB PLL Integer Multiplier Configuration (x 32 to x 63)

SPLLIMULT(5:0)	MULT VALUE
100000 b	x 32
100001 b	x 33
100010 b	x 34
100011 b	x 35
100100 b	x 36
100101 b	x 37
100110 b	x 38
100111 b	x 39
101000 b	x 40
101001 b	x 41
101010 b	x 42
101011 b	x 43
101100 b	x 44
101101 b	x 45
101110 b	x 46
101111 b	x 47
110000 b	x 48
110001 b	x 49
110010 b	x 50
110011 b	x 51
110100 b	x 52
110101 b	x 53
110110 b	x 54
110111 b	x 55
111000 b	x 56
111001 b	x 57
111010 b	x 58
111011 b	x 59
111100 b	x 60
111101 b	x 61
111110 b	x 62
111111 b	x 63

ADVANCE INFORMATION

2.11 Master Subsystem Clocking

The internal PLLSYSCLK clock, normally used as a source for all Master Subsystem clocks, is a divided-down output of the Main PLL or X1 external clock input, as defined by the SPLLCKEN bit of the SYSPLLCTL register.

There is also a second oscillator that internally generates two clocks: 32KHZCLK and 10MHZCLK. The 10MHZCLK is used by the Missing Clock Circuit to detect a possible absence of an external clock source to the Main Oscillator that drives the Main PLL. Detection of a missing clock results in a substitution of the 10MHZCLK for the PLLSYSCLK. The CLKFAIL signal is also sent to the NMI Block and the Control Subsystem where this signal can trip the ePWM peripherals.

The 32KHZCLK and 10MMHZCLK clocks are also used by the Cortex™-M3 Subsystem as possible sources for the Deep Sleep Clock.

There are four registers associated with the Main PLL: SYSPLLCTL, SYSPLLMULT, SYSPLLSTAT and SYSDIVSEL. Typically, the Cortex™-M3 processor writes to these registers, while the C28x processor has read access. The C28x can request write access to the above registers through the CLKREQEST register. Cortex™-M3 can regain write ownership of these registers through the MCLKREQUEST register.

The Master Subsystem operates in one of three modes: Run Mode, Sleep Mode, or Deep Sleep Mode. Table 2-25 shows the Master Subsystem low-power modes and their effect on both CPUs, clocks, and peripherals. Figure 2-10 shows the Cortex™-M3 clocks and the Master Subsystem low-power modes.

Table 2-25. Master Subsystem Low-Power Modes

Cortex™-M3 Low-Power Mode	State of Cortex™-M3 CPU	Clock to Cortex™-M3 Peripherals	Register Used to Gate Clocks to Cortex™-M3 Peripherals	Main PLL	USB PLL	Clock to C28x	Clock to Shared Resources	Clock to Analog Subsystem
Run	Active	M3SSCLK(1)	RCGC	On	On	PLLSYSCLK(2)	PLLSYSCLK(2)	ASYSCLK(3)
Sleep	Stopped	M3SSCLK(1)	RCGC or SCGC(4)	On	On	PLLSYSCLK(2)	PLLSYSCLK(2)	ASYSCLK(3)
Deep Sleep	Stopped	M3DSDIVCLK(5)	RCGC or DCGC(4)	Off	Off	Off	Off	Off

(1) PLLSYSCLK or OSCCLK divided-down per the M3SSDIVSEL register. In case of a missing source clock, M3SSCLK becomes 10MHZCLK divided-down per the M3SSDIVSEL register.
(2) PLLSYSCLK normally refers to the output of the Main PLL divided-down per the SYSDIVSEL register. In case the PLL is bypassed, the PLLSYSCLK becomes the OSCCLK divided-down per the SYSDIVSEL register. In case of a missing source clock, the 10MHZCLK is substituted for the PLLSYSCLK.
(3) PLLSYSCLK or OSCCLK divided-down per the CCLKCTL register. In case of a missing source clock, ASYSCLK becomes 10MHZCLK.
(4) Depends on the ACG bit of the RCC register.
(5) 32KHZCLK or 10MHZCLK or OSCCLK chosen/divided-down per the DSLPCLKCFG register, then again divided by the M3SSDIVSEL register (source determined inside the DSLPCLKCFG register).

2.11.1 Cortex™-M3 Run Mode

In Run Mode, the Cortex™-M3 processor, memory, and most of the peripherals are clocked by the M3SSCLK, which is a divide-down version of the PLLSYSCLK (from Main PLL). The USB is clocked from a dedicated USB PLL, the CAN peripherals are clocked by M3SSCLK, OSCCLK, or XCLKIN, and one of two watchdogs (WDOG1) is also clocked by the OSCCLK. Clock selection for these peripherals is accomplished via corresponding peripheral configuration registers. Clock gating for individual peripherals is defined inside the RCGS register. RCGS, SCGS, and DCGS clock-gating settings only apply to peripherals that are enabled in a corresponding DC (Device Configuration) register.

Execution of the WFI instruction (Wait-for-Interrupt) shuts down the HCLK to the Cortex™-M3 CPU and forces the Cortex™-M3 Subsystem into Sleep or Deep Sleep low-power mode, depending on the state of the SLEEPDEEP bit of the Cortex™-M3 SYSCTRL register. To come out of a low-power mode, any properly configured interrupt event terminates the Sleep or Deep Sleep Mode and returns the Cortex™-M3 processor/subsystem to Run Mode.

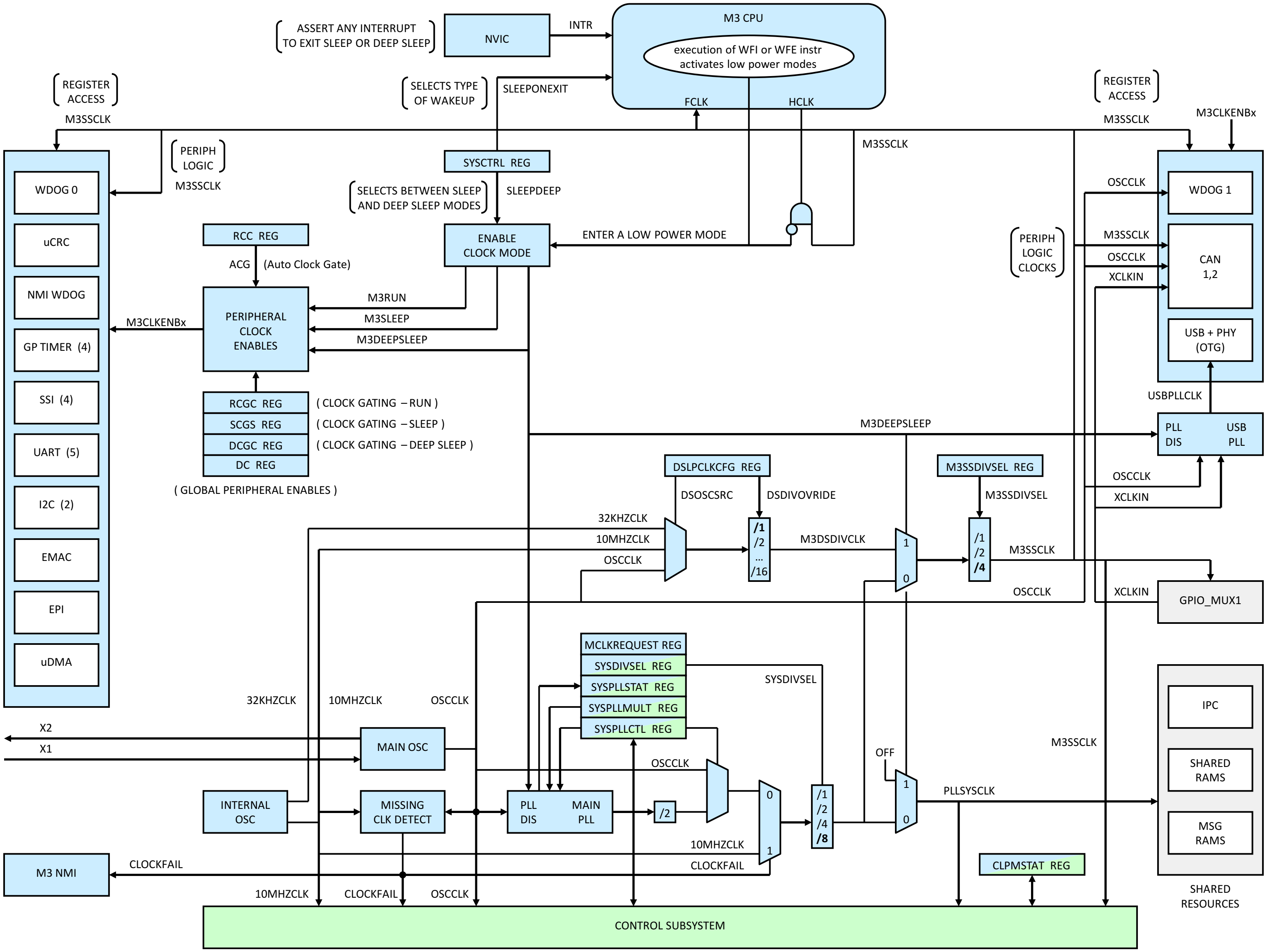


Figure 2-10. Cortex™-M3 Clocks and Low-Power Modes

2.11.2 Cortex™-M3 Sleep Mode

In Sleep Mode, the Cortex™-M3 processor and memory are prevented from clocking, and thus the code is no longer executing. The gating for the peripheral clocks may change based on the ACG bit of the RCC register. When ACG = 0, the peripheral clock gating is used as defined by the RCGS registers (same as in Run Mode); and when ASC = 1, the clock gating comes from the SCGS register. RCGS and SCGS clock-gating settings only apply to peripherals that are enabled in a corresponding DC register. Peripheral clock frequency for the enabled peripherals in Sleep Mode is the same as during the Run Mode.

Sleep Mode is terminated by any properly configured interrupt event. Exiting from the Sleep Mode depends on the Sleeponexit bit of the SYSCTRL register. When the Sleeponexit bit is 1, the processor will temporarily wake up only for the duration of the ISR of the interrupt causing the wake-up. After that, the processor goes back to Sleep Mode. When the Sleeponexit bit is 0, the processor wakes up permanently (for the ISR and thereafter).

2.11.3 Cortex™-M3 Deep Sleep Mode

In Deep Sleep Mode, the Cortex™-M3 processor and memory are prevented from clocking and thus the code is no longer executing. The Main PLL, USB PLL, ASYSCLK to the Analog Subsystem, and input clock to the C28x CPU and Shared Resources are turned off. The gating for the peripheral clocks may change based on the ACG bit of the RCC register. When ACG = 0, the peripheral clock gating is used as defined by the RCGS registers (same as in Run Mode); and when ASC = 1, the clock gating comes from the DCGS register. RCGS and DCGS clock gating settings only apply to peripherals that are enabled in a corresponding DC register.

Peripheral clock frequency for the enabled peripherals in Deep Sleep Mode is different from the Run Mode. One of three sources for the Deep Sleep clocks (32KHZCLK, 10MHZCLK, or OSCLK) is selected with the DSOSCSRC bits of the DSLPCLKCFG register. This clock is divided-down according to DSDIVOVRIDE bits of the DSLPCLKCFG register. The output of this Deep Sleep Divider is further divided-down per the M3SSDIVSEL bits of the D3SSDIVSEL register to become the Deep Sleep Clock. If 32KHXCLK or 10MHZCLK is selected in Deep Sleep mode, the internal oscillator circuit (that generates OSCCLK) is turned off.

The Cortex™-M3 processor should enter the Deep Sleep mode only after first confirming that the C28x is already in the Standby mode. Typically, just before entering the Standby mode, the C28x will record in the CLPMSTAT that it is about to do so. The Cortex™-M3 processor can read the CLPMSTAT register to check if the C28x is in Standby mode, and only then should the Cortex™-M3 processor go into Deep Sleep. The reason for the Cortex™-M3 processor to confirm that the C28x is in Standby mode before the Cortex™-M3 processor enters the Deep Sleep mode is that the Deep Sleep mode shuts down the clock to C28x and its peripherals, and if this clock shutdown is not expected by the C28x, unintended consequences could result for some of the C28x control peripherals.

Deep Sleep Mode is terminated by any properly configured interrupt event. Exiting from the Deep Sleep Mode depends on the Sleeponexit bit of the SYSCTRL register. When the Sleeponexit bit is 1, the processor will temporarily wake up only for the duration of the ISR of the interrupt causing the wake-up. After that, the processor goes back to Deep Sleep Mode. When the Sleeponexit bit is 0, the processor wakes up permanently (for the ISR and thereafter).

2.12 Control Subsystem Clocking

The CLKIN input clock to the C28x processor is normally a divided-down output of the Main PLL or X1 external clock input. There are four registers associated with the Main PLL: SYSPLLCTL, SYSPLLMULT, SYSPLLSTAT and SYSDIVSEL. Typically, the Cortex™-M3 processor writes to these registers, while the C28x processor has read access. The C28x can request write access to the above registers through the CLKREQEST register. The Cortex™-M3 can regain write ownership of these registers through the MCLKREQUEST register.

Individual C28x peripherals can be turned on or off by gating C28SYSCLK to those peripherals, which is done via the CPCLKCR0,2,3 registers.

The C28x processor outputs two clocks: C28CPUCLK and C28SYSCLK. The C28SYSCLK is used by C28x peripherals, C28x Timer 0, C28x Timer 1, and C28x Timer 2. C28x Timer 2 can also be clocked by OSCCLK or 10MHZCLK (see Figure 2-11). The C28CPUCLK is used by the C28x CPU, FPU, VCU, and PIE.

The Control Subsystem operates in one of three modes: Normal Mode, Idle Mode, or Standby Mode. Table 2-26 shows the Control Subsystem low-power modes and their effect on the C28x CPU, clocks, and peripherals. Figure 2-11 shows the Control Subsystem clocks and low-power modes.

Table 2-26. Control Subsystem Low-Power Modes(1)

C28x Low-Power Mode	State of C28x CPU	C28CPUCLK(2)	C28SYSCLK(3)	Registers Used to Gate Clocks to C28x Peripherals
Normal	Active	On	On	CPCLKCR0,1,3
Idle	Stopped	Off	On	CPCLKCR0,1,3
Standby	Stopped	Off	Off	N/A

(1) The input clock to the C28x CPU is PLLSYSCLK from the Master Subsystem. This clock is turned off when the Master Subsystem enters the Deep Sleep mode.
(2) C28CPUCLK is an output from the C28x CPU. C28CPUCLK clocks the C28x FPU, VCU, and PIE.
(3) C28SYSCLK is an output from the C28x CPU. C28SYSCLK clocks C28x peripherals.

2.12.1 C28x Normal Mode

In Normal Mode, the C28x processor, Local Memory, and C28x peripherals are clocked by the C28SYSCLK, which is derived from the C28CLKIN input clock to the C28x processor. The FPU, VCU, and PIE are clocked by the C28CPUCLK, which is also derived from the C28CLKIN. Timer 2 can also be clocked by the TMR2CLK, which is a divided-down version of one of three source clocks—C28SYSCLK, OSCCLK, and 10MHZCLK—as selected by the CLKCTL register. Additionally, the LOSPCP register can be programmed to provide a dedicated clock (C28LSPCLK) to the SCI, SPI, and McBSP peripherals; and the HISPCP register can be programmed to provide a dedicated clock (C28HSPCLK) to stretch three outputs from ePWM peripherals.

Clock gating for individual peripherals is defined inside the CPCLKCR0,1,3 registers. Execution of the IDLE instruction stops the C28x processor from clocking and activates the IDLES signal. The IDLES signal is gated with two LPM bits of the CPCLKCR0 register to enter the C28x Subsystem into Idle mode or Standby Mode.

2.12.2 C28x Idle Mode

In Idle Mode, the C28x processor stops executing instructions and the C28CPUCLK is turned off. The C28SYSCLK continues to run. Exit from Idle Mode is accomplished by any enabled interrupt or the C28NMIINT (C28x non-maskable interrupt).

Upon exit from Idle Mode, the C28CPUCLK is restored. If LPMWAKE interrupt is enabled, the LPMWAKE ISR is executed. Next, the C28x processor starts fetching instructions from a location immediately following the IDLE instruction that originally triggered the Idle Mode.

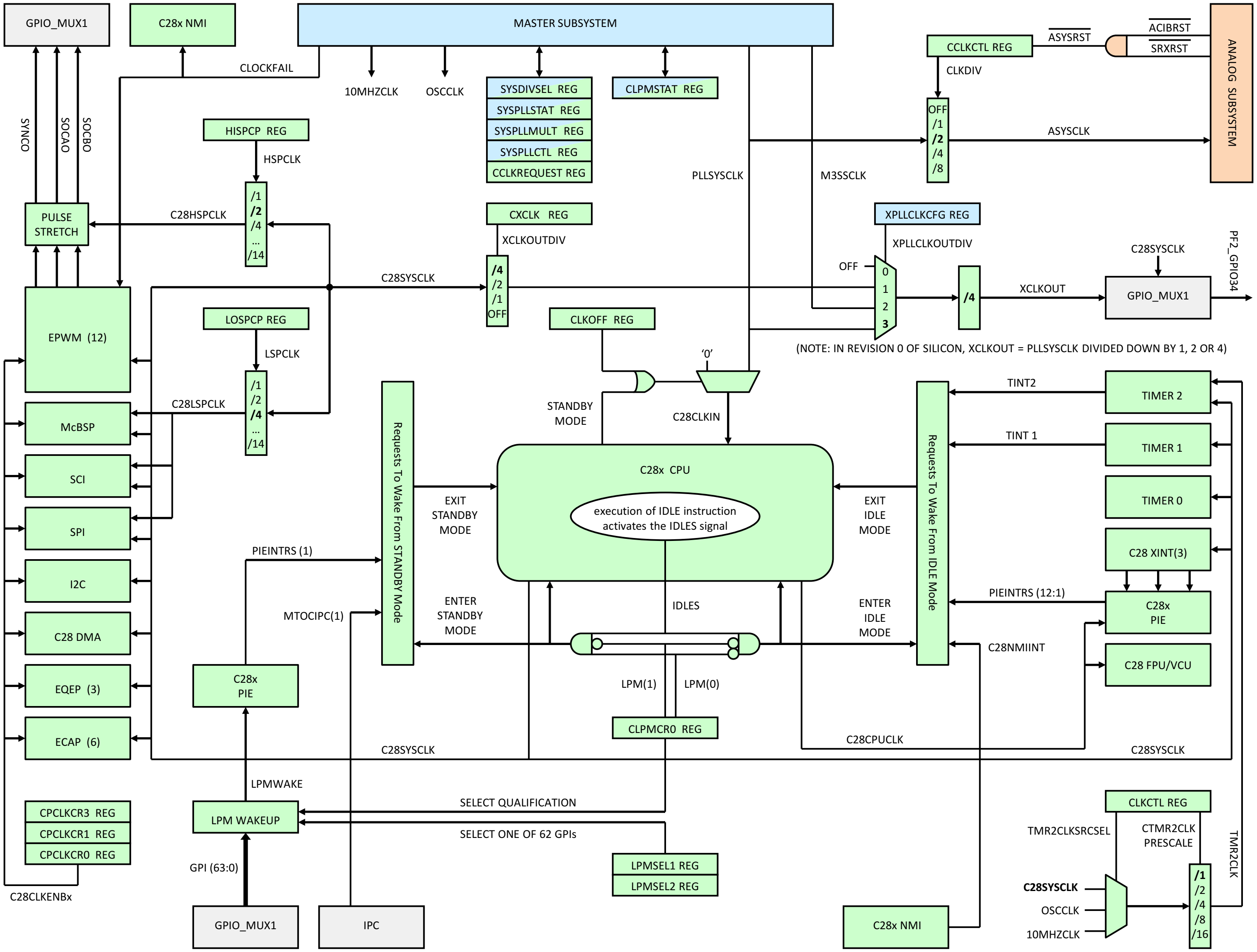


Figure 2-11. C28x Clocks and Low-Power Modes

2.12.3 C28x Standby Mode

In Standby Mode, the C28x processor stops executing instructions and the C28CLKIN, C28CPUCLK, and C28SYSCLK are turned off. Exit from Standby Mode is accomplished by one of 64 GPIOs from the GPIO_MUX1 block, or MTOCIPCINT1 (interrupt from MTOC IPC peripheral). The wakeup GPIO selected inside the GPIO_MUX block enters the Qualification Block as the LPMWAKE signal. Inside the Qualification Block, the LPMWAKE signal is sampled per the QUALSTDBY bits (bits [7:2] of the CPCLKCR0 register) before propagating into the wake request logic.

Cortex™-M3 should use CLPMSTAT register bits to tell the C28x to go into Standby mode before going into Deep Sleep mode. Otherwise, the clock to the C28x will be turned off suddenly when the control software is not expecting this clock to shut off. When the device is in Deep Sleep/Standby mode, wake-up should happen only from the Master Subsystem, since all C28x clocks are off (C28CLKIN, C28CPUCLK, C28SYSCLK), thus preventing the C28x from waking up first.

Upon exit from STANDBY Mode, the C28CLKIN, C28SYSCLK, and C28CPUCLK are restored. If the LPMWAKE interrupt is enabled, the LPMWAKE ISR is executed. Next, the C28x processor starts fetching instructions from a location immediately following the IDLE instruction that originally triggered the Standby Mode.

NOTE

For GPIO_MUX1 pins PF6_GPIO38 and PG6_GPIO46, only the corresponding USB function is available on silicon revision 0 devices (GPIO and other functions listed in Table 3-1 are not available).

2.13 Analog Subsystem Clocking

The Analog Subsystem is clocked by ASYSCLK, which is a divided-down version of the PLLSYSCLK as defined by CLKDIV bits of the CCLKCTL register. The CCLKCTL register is exclusively accessible by the C28x processor. The CCLKCTL register is reset by $\overline{\text{ASYSRST}}$, which is derived from two Analog Subsystem resets—$\overline{\text{ACIBRST}}$ and $\overline{\text{SRXRST}}$. Therefore, while normally the C28x controls the frequency of ASYSCLK, it is possible for the Cortex™-M3 software to restore the ASYSCLK to its default value by resetting the Analog Subsystem.

The ASYSCLK is shut down when the Cortex™-M3 processor enters the Deep Sleep mode.

2.14 Shared Resources Clocking

The IPC, Shared RAMs, and Message RAMs are clocked by PLLSYSCLK. EPI is clocked by M3SSCLK. The PLLSYSCLK normally refers to the output of the Main PLL divided-down per the SYSDIVSEL register. In case the PLL is bypassed, the PLLSYSCLK becomes the OSCCLK divided-down per the SYSDIVSEL register. In case of a missing source clock, the 10MHZCLK is substituted for the PLLSYSCLK.

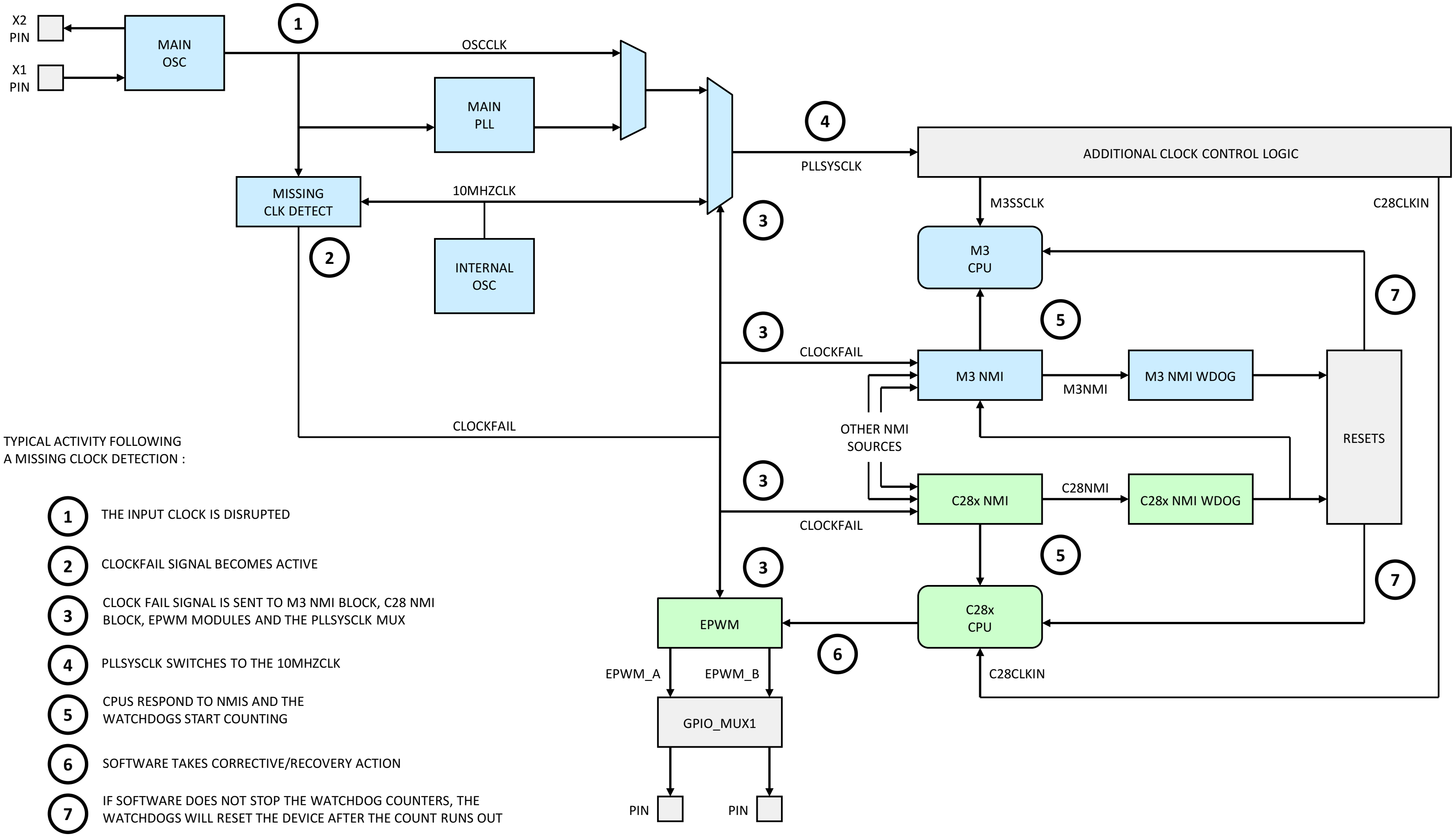
Although EPI is a shared peripheral, it is physically located inside the Cortex™-M3 Subsystem; therefore, EPI is clocked by M3SSCLK.

2.15 Loss of Input Clock (NMI Watchdog Function)

The Concerto devices use two type of input clocks. The main clock, for clocking most of the digital logic of the Master, Control, and Analog subsystems, enters the chip through pins X1 and X2 when using external crystal or just pin X1 when using an external oscillator. The second clock enters the chip through the XCLKIN pin and this second clock can be used to clock the USB PLL and CAN peripherals. Only the main clock has a built-in Missing Clock Detection circuit to recognize when the clock source vanishes and to enable other chip components to take corrective or recovery action from such event (see Figure 2-12).

The Missing Clock Detection circuit itself is clocked by the 10MHZCLK (from an internal zero-pin oscillator) so that, if the main clock disappears, the circuit is still working. Immediately after detecting a missing source clock, the Missing Clock Detection circuit outputs the CLOCKFAIL signal to the Cortex™-M3 NMI circuit, the C28x NMI, ePWM peripherals, and the PLLSYSCLK mux. When the PLLSYSCLK mux senses an active CLOCKFAIL signal, the PLLSYSCLK mux revives the PLLSYSCLK using the 10MHZCLK. Simultaneously, the ePWM peripherals can use the CLOCKFAIL signal to stop down driving motor control outputs. The NMI blocks respond to the CLOCKFAIL signal by sending an NMI interrupt to a corresponding CPU, while starting the associated NMI watchdog counter.

If the software does not respond to the clock-fail condition, the watchdog timers will overflow, resulting in the device reset. If the software does react to the NMI, the software can prevent the impending reset by disabling the watchdog timers, and then the software can initiate necessary corrective action such as switching over to an alternative clock source (if available) or the software can initiate a shut-down procedure for the system.



TYPICAL ACTIVITY FOLLOWING A MISSING CLOCK DETECTION :

① THE INPUT CLOCK IS DISRUPTED

② CLOCKFAIL SIGNAL BECOMES ACTIVE

③ CLOCK FAIL SIGNAL IS SENT TO M3 NMI BLOCK, C28 NMI BLOCK, EPWM MODULES AND THE PLLSYSCLK MUX

④ PLLSYSCLK SWITCHES TO THE 10MHZCLK

⑤ CPUS RESPOND TO NMIS AND THE WATCHDOGS START COUNTING

⑥ SOFTWARE TAKES CORRECTIVE/RECOVERY ACTION

⑦ IF SOFTWARE DOES NOT STOP THE WATCHDOG COUNTERS, THE WATCHDOGS WILL RESET THE DEVICE AFTER THE COUNT RUNS OUT

Figure 2-12. Missing Clock Detection

2.16 GPIOs and Other Pins

Most Concerto external pins are shared among many internal peripherals. This sharing of pins is accomplished through several I/O muxes where a specific physical pin can be assigned to selected signals of internal peripherals.

Most of the I/O pins of the Concerto™ MCU can also be configured as programmable GPIOs. Exceptions include the X1 and X2 oscillator inputs; the $\overline{\text{XRS}}$ digital reset and $\overline{\text{ARS}}$ analog reset; the $\overline{\text{VREG12EN}}$ and $\overline{\text{VREG18EN}}$ internal voltage regulator enables; and five JTAG pins. The 144 primary GPIOs are grouped in 2 programmable blocks: GPIO_MUX1 block (136 pins) and GPIO_MUX2 block (8 pins). Additionally, eight secondary GPIOs are available through the AIO_MUX1 block (four pins) and AIO_MUX2 block (four pins). Figure 2-13 shows the GPIOs and other pins.

2.16.1 GPIO_MUX1

One-hundred and thirty-six pins of the GPIO_MUX1 block can be selectively mapped through corresponding sets of registers to all Cortex™-M3 peripherals, to all C28x peripherals, to 136 General-Purpose Inputs, to 136 General-Purpose Outputs, or a mixture of all of the above. The first 64 pins of GPIO_MUX1 (GPIO0–GPIO63) can also be mapped to 12 ePWM Trip Inputs, 6 eCAP inputs, 3 External Interrupts to the C28x PIE, and the C28x Standby Mode Wakeup signal (LMPWAKE). Additionally, each GPIO_MUX1 pin can have a pullup enabled or disabled. By default, all pullups and outputs are disabled on reset, and all pins of the GPIO_MUX1 block are mapped to Cortex™-M3 peripherals (and not to C28x peripherals).

Figure 2-14 shows the internal structure of GPIO_MUX1. The blue blocks represent the Master Subsystem side of GPIO_MUX1, and the green blocks are the Control Subsystem side. The grey block in the center, Pin-Level Mux, is where the GPIO_MUX1 pins are individually assigned between the two subsystems, based on how the configuration registers are programmed in the blue and green blocks (see Figure 2-15 for the configuration registers).

Pin-Level Mux assigns Master Subsystem peripheral signals, Control Subsystem peripheral signals, or GPIOs to the 136 GPIO_MUX1 pins. In addition to connecting peripheral I/Os of the two subsystems to pins, the Pin-Level Mux also provides other signals to the subsystems: XCLKIN and GPIO[S:A] IRQ signals to the Master Subsystem, plus GPTRIP[12:1] and GPI[63:0] signals to the Control Subsystem. XCLKIN carries a clock from an external pin to USB PLL and CAN modules. The 17 GPIO[S:A] IRQ signals are interrupt requests from selected external pins to the NVIC interrupt controller. The 12 GPTRIP[12:1] signals carry trip events from selected external pins to C28x control peripherals—ePWM, eCAP, and eQEP. Sixty-four GPI signals go to the C28x LPM GPIO Select block where one of them can be selected to wake up the C28x CPU from Low-Power Mode. One-hundred and thirty-six (136) GPI signals go to the C28x QUAL block where they can be configured with a qualification sampling period (see Figure 2-15).

The configuration registers for the muxing of Master Subsystem peripherals are organized in 17 sets (A–S), with each set being responsible for eight pins. The first nine sets of these registers (A–J) are programmable by the Cortex™-M3 CPU via the AHB bus or the APB bus. The remaining sets of registers (K–S) are programmable by the AHB bus only. The configuration register for the muxing of Control Subsystem peripherals are organized in five sets (A–E), with each set being responsible for up to 32 pins. These registers are programmable by the C28x CPU via the C28x CPU bus. Figure 2-15 shows set A of the Master Subsystem GPIO configuration registers, set A of the Control Subsystem registers, and the muxing logic for one GPIO pin as driven by these registers.

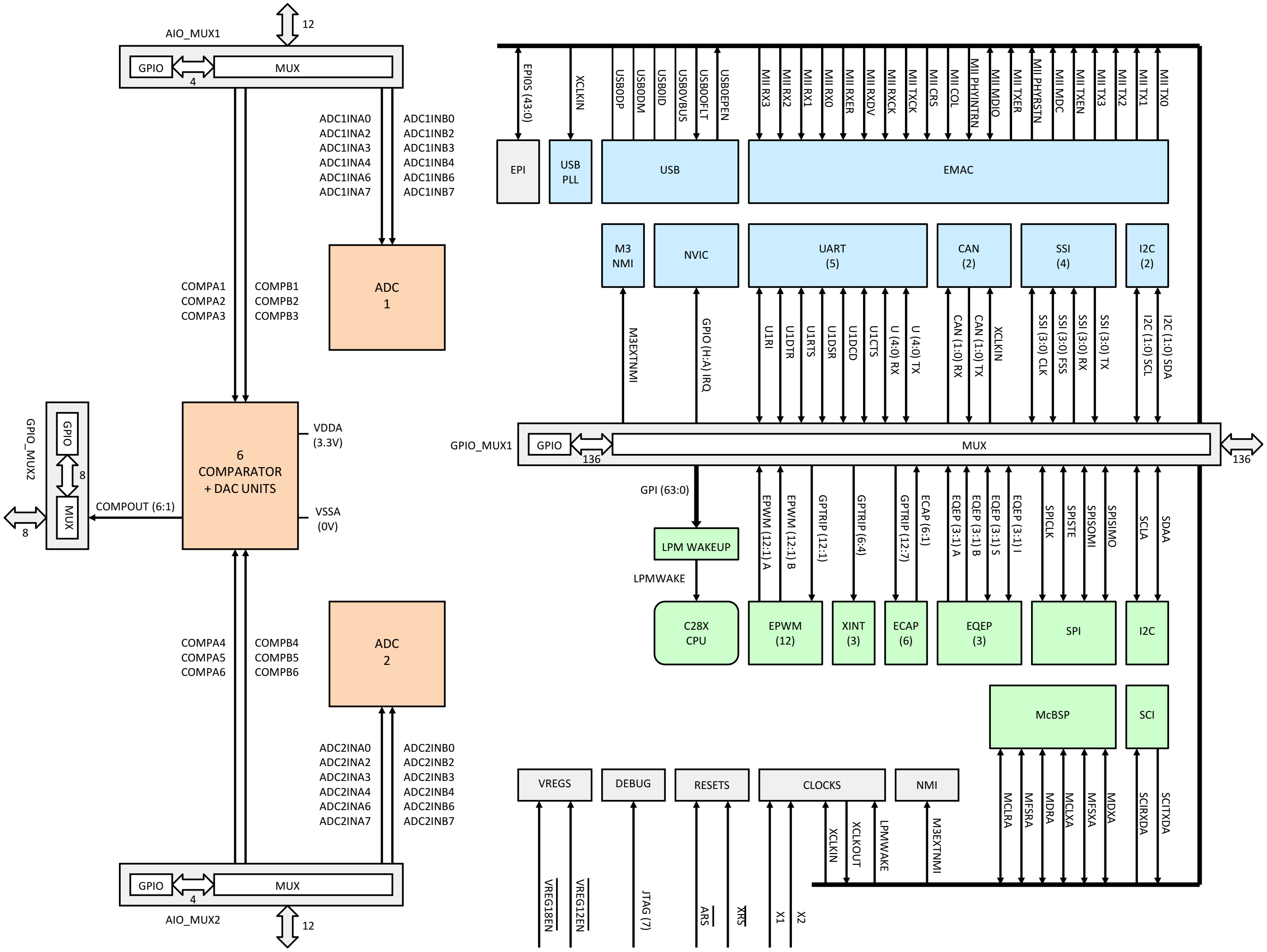


Figure 2-13. GPIOs and Other Pins

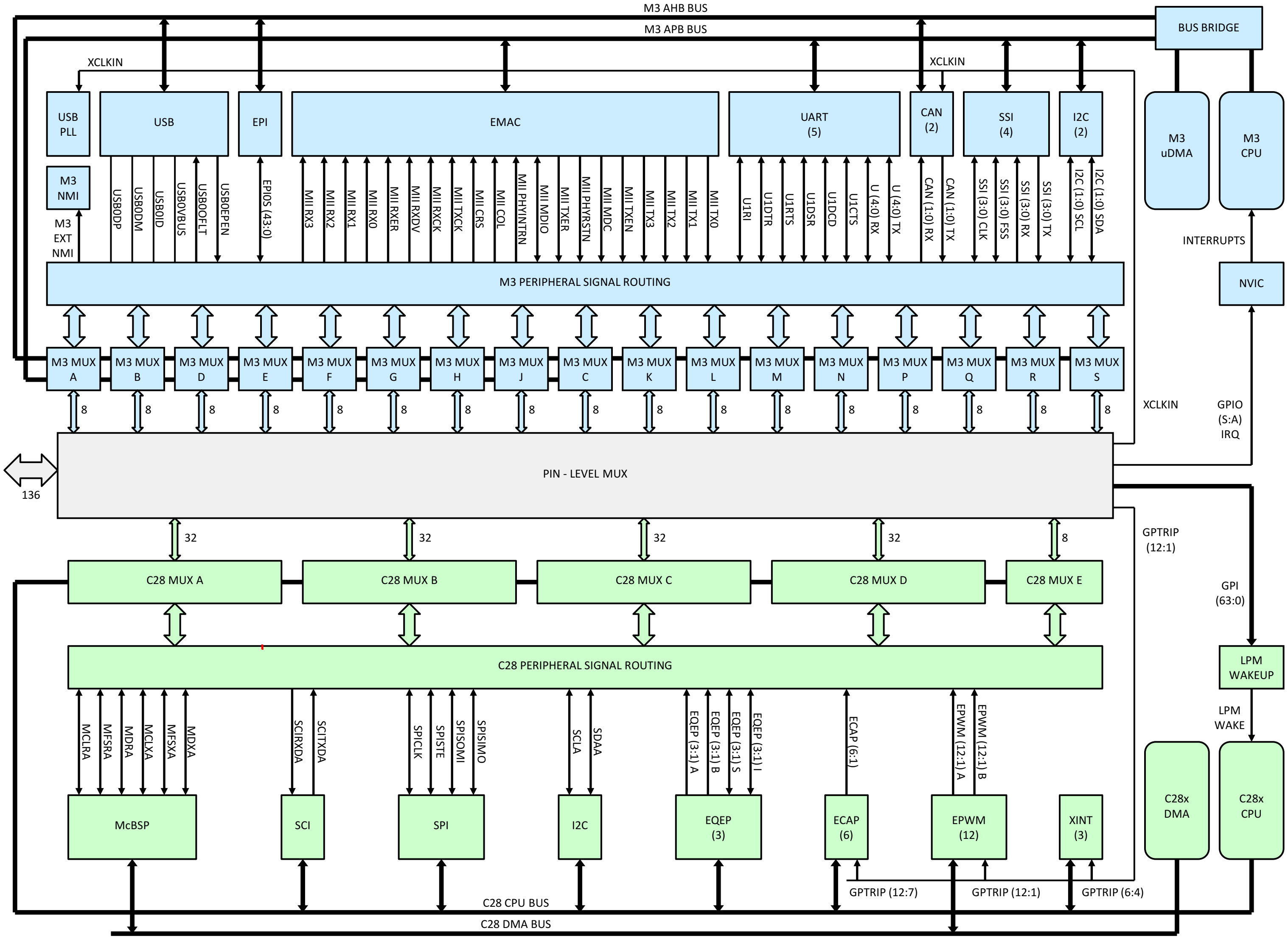


Figure 2-14. GPIO_MUX1 Block

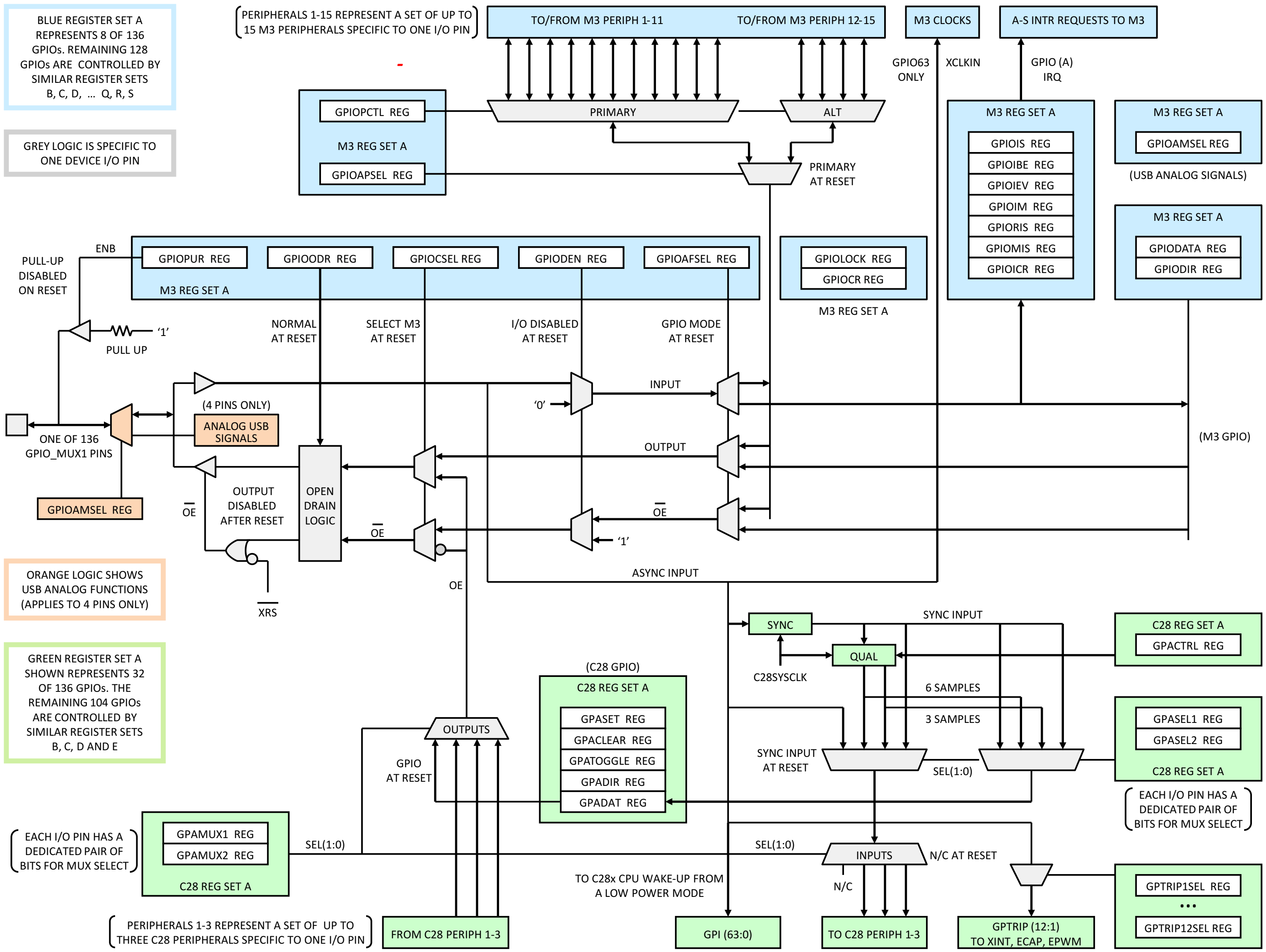


Figure 2-15. GPIO_MUX1 Pin Mapping Through Register Set A

For each of the 8 pins in set A of the Cortex™-M3 GPIO registers, register GPIOPCTL selects between 1 of 11 possible primary Cortex™-M3 peripheral signals, or 1 of 4 possible alternate peripheral signals. Register GPIOAPSEL then picks one output to propagate further along the muxing chain towards a given pin. The input takes the reverse path. See Table 2-27 and Table 2-28 for the mapping of Cortex™-M3 peripheral signals to GPIO_MUX1 pins.

Similarly, on the C28x side, GPAMUX1 and GPAMUX2 registers select 1 of 4 possible C28x peripheral signals for each of 32 pins of set A. The selected C28x peripheral output then propagates further along the muxing chain towards a given pin. The input takes the reverse path. See Table 2-29 for the mapping of C28x peripheral signals to GPIO_MUX1 pins.

In addition to passing mostly digital signals, four GPIO_MUX1 pins can also be assigned to analog signals. The GPIO Analog Mode Select (GPIOAMSEL) Register is used to assign four pins to analog USB signals. PF6_GPIO38 becomes USB0VBUS, PG2_GPIO42 becomes USB0DM, PG5_GPIO45 becomes USB0DP, and PG6_GPIO46 becomes USB0ID. When analog mode is selected, these four pins are not available for digital GPIO_MUX1 options as described above.

Another special case is the External Oscillator Input signal (XCLKIN). This signal, available through pin PJ7_GPIO63, is directly tied to USBPLLCLK (clock input to USB PLL) and two CAN modules. XCLKIN is always available at these modules where it can be selected through local registers.

NOTE

For GPIO_MUX1 pins PF6_GPIO38 and PG6_GPIO46, only the corresponding USB function is available on silicon revision 0 devices (GPIO and other functions listed in Table 3-1 are not available).

Table 2-27. GPIO_MUX1 Pin Assignments (M3 Primary Modes)(1)

Analog Mode (USB Pins)	Device Pin Name	M3 Primary Mode 1	M3 Primary Mode 2	M3 Primary Mode 3	M3 Primary Mode 4	M3 Primary Mode 5	M3 Primary Mode 6	M3 Primary Mode 7	M3 Primary Mode 8	M3 Primary Mode 9	M3 Primary Mode 10	M3 Primary Mode 11
–	PA0_GPIO0	U0RX	–	–	–	–	–	–	I2C1SCL	U1RX	–	–
–	PA1_GPIO1	U0TX	–	–	–	–	–	–	I2C1SDA	U1TX	–	–
–	PA2_GPIO2	SSI0CLK	–	MMI_TXD2	–	–	–	–	–	–	–	–
–	PA3_GPIO3	SSI0FSS	–	MMI_TXD1	–	–	–	–	–	–	–	–
–	PA4_GPIO4	SSI0RX	–	MMI_TXD0	–	CAN0RX	–	–	–	–	–	–
–	PA5_GPIO5	SSI0TX	–	MMI_RXDV	–	CAN0TX	–	–	–	–	–	–
–	PA6_GPIO6	I2C1SCL	CCP1	MMI_RXCK	–	–	CAN0RX	–	USB0EPEN	U1CTS	–	–
–	PA7_GPIO7	I2C1SDA	CCP4	MMI_RXER	–	–	CAN0TX	CCP3	USB0PFLT	U1DCD	–	–
–	PB0_GPIO8	CCP0	–	–	–	U1RX	–	–	–	–	–	–
–	PB1_GPIO9	CCP2	–	–	CCP1	U1TX	–	–	–	–	–	–
–	PB2_GPIO10	I2C0SCL	–	–	CCP3	CCP0	–	–	USB0EPEN	–	–	–
–	PB3_GPIO11	I2C0SDA	–	–	–	–	–	–	USB0PFLT	–	–	–
–	PB4_GPIO12	–	–	–	U2RX	CAN0RX	–	U1RX	EPI0S23	–	–	–
–	PB5_GPIO13	–	CCP5	CCP6	CCP0	CAN0TX	CCP2	U1TX	EPI0S22	–	–	–
–	PB6_GPIO14	CCP1	CCP7	–	–	–	CCP5	–	EPI0S37(2)	–	–	–
–	PB7_GPIO15	–	–	–	NMI	–	–	MII_RXD1	EPI0S36(2)	–	–	–
–	PD0_GPIO16	PWM0	CAN0RX	–	U2RX	U1RX	CCP6	MII_RXDV	–	U1CTS	–	–
–	PD1_GPIO17	PWM1	CAN0TX	–	U2TX	U1TX	CCP7	MII_TXER	–	U1DCD	CCP2	–
–	PD2_GPIO18	U1RX	CCP6	–	CCP5	–	–	–	EPI0S20	–	–	–
–	PD3_GPIO19	U1TX	CCP7	–	CCP0	–	–	–	EPI0S21	–	–	–
–	PD4_GPIO20	CCP0	CCP3	–	MII_TXD3	–	–	–	–	U1RI	EPI0S19	–
–	PD5_GPIO21	CCP2	CCP4	–	MII_TXD2	–	–	–	–	U2RX	EPI0S28	–
–	PD6_GPIO22	Fault0	–	–	MII_TXD1	–	–	–	–	U2TX	EPI0S29	–
–	PD7_GPIO23	IDX0	–	CCP1	MII_TXD0	–	–	–	–	U1DTR	EPI0S30	–
–	PE0_GPIO24	PWM4	SSI1CLK	CCP3	–	–	–	–	EPI0S8	USB0PFLT	–	–
–	PE1_GPIO25	PWM5	SSI1FSS	–	CCP2	CCP6	–	–	EPI0S9	–	–	–
–	PE2_GPIO26	CCP4	SSI1RX	–	–	CCP2	–	–	EPI0S24	–	–	–
–	PE3_GPIO27	CCP1	SSI1TX	–	–	CCP7	–	–	EPI0S25	–	–	–
–	PE4_GPIO28	CCP3	–	–	–	U2TX	CCP2	MII_RXD0	EPI0S34(2)	–	–	–
–	PE5_GPIO29	CCP5	–	–	–	–	–	–	EPI0S35(2)	–	–	–
–	PE6_GPIO30	–	–	–	–	–	–	–	–	U1CTS	–	–
–	PE7_GPIO31	–	–	–	–	–	–	–	–	U1DCD	–	–

(1) Blank fields represent Reserved functions.
(2) This muxing option is only available on silicon Revision A devices; this muxing option is not available on silicon Revision 0 devices.

Table 2-27. GPIO_MUX1 Pin Assignments (M3 Primary Modes)(1) (continued)

Analog Mode (USB Pins)	Device Pin Name	M3 Primary Mode 1	M3 Primary Mode 2	M3 Primary Mode 3	M3 Primary Mode 4	M3 Primary Mode 5	M3 Primary Mode 6	M3 Primary Mode 7	M3 Primary Mode 8	M3 Primary Mode 9	M3 Primary Mode 10	M3 Primary Mode 11
–	PF0_GPIO32	CAN1RX	–	–	MII_RXCK	–	–	–	–	U1DSR	–	–
–	PF1_GPIO33	CAN1TX	–	–	MII_RXER	–	–	–	–	U1RTS	CCP3	–
–	PF2_GPIO34	–	–	$\overline{\text{MII\_PHYINTR}}$	–	–	–	–	EPI0S32[2]	SSI1CLK	–	–
–	PF3_GPIO35	–	–	MII_MDC	–	–	–	–	EPI0S33[2]	SSI1FSS	–	–
–	PF4_GPIO36	CCP0	–	MII_MDIO	–	–	–	–	EPI0S12	SSI1RX	–	–
–	PF5_GPIO37	CCP2	–	MII_RXD3	–	–	–	–	EPI0S15	SSI1TX	–	–
USB0VBUS	PF6_GPIO38	CCP1	–	MII_RXD2	–	–	–	–	EPI0S38[2]	–	U1RTS	–
–	PF7_GPIO39	–	–	–	–	–	–	–	–	–	–	–
–	PG0_GPIO40	U2RX	–	I2C1SCL	–	–	–	USB0EPEN	EPI0S13	–	–	–
–	PG1_GPIO41	U2TX	–	I2C1SDA	–	–	–	–	EPI0S14	–	–	–
USB0DM	PG2_GPIO42	–	–	MII_COL	–	–	–	–	EPI0S39[2]	–	–	–
–	PG3_GPIO43	–	–	MII_CRS	–	–	–	–	–	–	–	–
–	PG4_GPIO44	–	–	–	–	–	–	–	–	–	–	–
USB0DP	PG5_GPIO45	CCP5	–	MII_TXEN	–	–	–	–	EPI0S40[2]	–	U1DTR	–
USB0ID	PG6_GPIO46	–	–	MII_TXCK	–	–	–	–	EPI0S41[2]	–	U1RI	–
–	PG7_GPIO47	–	–	MII_TXER	–	–	–	–	CCP5	EPI0S31	–	–
–	PH0_GPIO48	CCP6	–	$\overline{\text{MII\_PHYRST}}$	–	–	–	–	EPI0S6	–	–	–
–	PH1_GPIO49	CCP7	–	–	–	–	–	–	EPI0S7	–	–	–
–	PH2_GPIO50	–	–	–	–	–	–	–	EPI0S1	MII_TXD3	–	–
–	PH3_GPIO51	–	–	–	USB0EPEN	–	–	–	EPI0S0	MII_TXD2	–	–
–	PH4_GPIO52	–	–	–	USB0PFLT	–	–	–	EPI0S10	MII_TXD1	–	SSI1CLK
–	PH5_GPIO53	–	–	–	–	–	–	–	EPI0S11	MII_TXD0	–	SSI1FSS
–	PH6_GPIO54	–	–	–	–	–	–	–	EPI0S26	MII_RXDV	–	SSI1RX
–	PH7_GPIO55	–	–	MII_RXCK	–	–	–	–	EPI0S27	–	–	SSI1TX
–	PJ0_GPIO56	–	–	MII_RXER	–	–	–	–	EPI0S16	–	–	I2C1SCL
–	PJ1_GPIO57	–	–	–	–	–	–	–	EPI0S17	USB0PFLT	–	I2C1SDA
–	PJ2_GPIO58	–	–	–	–	–	–	–	EPI0S18	CCP0	–	–
–	PJ3_GPIO59	–	–	–	–	–	–	–	EPI0S19	U1CTS	CCP6	–
–	PJ4_GPIO60	–	–	–	–	–	–	–	EPI0S28	U1DCD	CCP4	–
–	PJ5_GPIO61	–	–	–	–	–	–	–	EPI0S29	U1DSR	CCP2	–
–	PJ6_GPIO62	–	–	–	–	–	–	–	EPI0S30	U1RTS	CCP1	–
–	PJ7_GPIO63/ XCLKIN	–	–	–	–	–	–	–	–	U1DTR	CCP0	–

Table 2-27. GPIO_MUX1 Pin Assignments (M3 Primary Modes)(1) (continued)

Analog Mode (USB Pins)	Device Pin Name	M3 Primary Mode 1	M3 Primary Mode 2	M3 Primary Mode 3	M3 Primary Mode 4	M3 Primary Mode 5	M3 Primary Mode 6	M3 Primary Mode 7	M3 Primary Mode 8	M3 Primary Mode 9	M3 Primary Mode 10	M3 Primary Mode 11
–	PC0_GPIO64	–	–	–	–	–	–	–	EPI0S32(2)	–	–	–
–	PC1_GPIO65	–	–	–	–	–	–	–	EPI0S33(2)	–	–	–
–	PC2_GPIO66	–	–	–	–	–	–	–	EPI0S37(2)	–	–	–
–	PC3_GPIO67	–	–	–	–	–	–	–	EPI0S36(2)	–	–	–
–	PC4_GPIO68	CCP5	–	MII_TXD3	–	CCP2	CCP4	–	EPI0S2	CCP1	–	–
–	PC5_GPIO69	CCP1	–	–	–	CCP3	USB0EPEN	–	EPI0S3	–	–	–
–	PC6_GPIO70	CCP3	–	–	–	U1RX	CCP0	USB0PFLT	EPI0S4	–	–	–
–	PC7_GPIO71	CCP4	–	–	CCP0	U1TX	USB0PFLT	–	EPI0S5	–	–	–

Table 2-28. GPIO_MUX1 Pin Assignments (M3 Alternate Modes)(1)

Analog Mode (USB Pins)	Device Pin Name	M3 Alternate Mode 12	M3 Alternate Mode 13	M3 Alternate Mode 14	M3 Alternate Mode 15
–	PA0_GPIO0	–	–	–	–
–	PA1_GPIO1	–	–	–	SSI1FSS
–	PA2_GPIO2	–	–	U1CTS	–
–	PA3_GPIO3	–	–	U1DCD	SSI1CLK
–	PA4_GPIO4	–	–	U1DSR	–
–	PA5_GPIO5	–	–	U1RTS	–
–	PA6_GPIO6	–	–	U1DTR	–
–	PA7_GPIO7	MII_RXD1	–	U1RI	–
–	PB0_GPIO8	–	SSI2TX	CAN1TX	U4TX
–	PB1_GPIO9	–	SSI2RX	–	–
–	PB2_GPIO10	–	SSI2CLK	CAN1RX	U4RX
–	PB3_GPIO11	–	SSI2FSS	U1RX	–
–	PB4_GPIO12	–	–	CAN1TX	SSI1TX
–	PB5_GPIO13	–	–	CAN1RX	SSI1RX
–	PB6_GPIO14	MII_CRS	I2C0SDA	U1TX	SSI1CLK
–	PB7_GPIO15	–	I2C0SCL	U1RX	SSI1FSS
–	PD0_GPIO16	MII_RXD2	SSI0TX	CAN1TX	USB0EPEN
–	PD1_GPIO17	MII_COL	SSI0RX	CAN1RX	USB0PFLT
–	PD2_GPIO18	–	SSI0CLK	U1TX	CAN0RX
–	PD3_GPIO19	–	SSI0FSS	U1RX	CAN0TX
–	PD4_GPIO20	–	–	U3TX	CAN1TX
–	PD5_GPIO21	–	–	U3RX	CAN1RX
–	PD6_GPIO22	–	–	I2C1SDA	U1TX
–	PD7_GPIO23	–	–	I2C1SCL	U1RX
–	PE0_GPIO24	–	SSI3TX	CAN0RX	SSI1TX
–	PE1_GPIO25	–	SSI3RX	CAN0TX	SSI1RX
–	PE2_GPIO26	–	SSI3CLK	U2RX	SSI1CLK
–	PE3_GPIO27	–	SSI3FSS	U2TX	SSI1FSS
–	PE4_GPIO28	–	U0RX	EPI0S38(2)	USB0EPEN
–	PE5_GPIO29	MII_TXER	U0TX	–	USB0PFLT
–	PE6_GPIO30	MII_MDIO	CAN0RX	–	–
–	PE7_GPIO31	MII_RXD3	CAN0TX	–	–
–	PF0_GPIO32	–	I2C0SDA	TRACED2	–
–	PF1_GPIO33	–	I2C0SCL	TRACED3	–
–	PF2_GPIO34	–	–	TRACECLK	XCLKOUT
–	PF3_GPIO35	–	U0TX	TRACED0	–
–	PF4_GPIO36	–	U0RX	–	–
–	PF5_GPIO37	–	–	–	MII_TXEN
USB0VBUS	PF6_GPIO38	–	–	–	–
–	PF7_GPIO39	–	–	CAN1TX	–

(1) Blank fields represent Reserved functions.
(2) This muxing option is only available on silicon Revision A devices; this muxing option is not available on silicon Revision 0 devices.

Table 2-28. GPIO_MUX1 Pin Assignments (M3 Alternate Modes)(1) (continued)

Analog Mode (USB Pins)	Device Pin Name	M3 Alternate Mode 12	M3 Alternate Mode 13	M3 Alternate Mode 14	M3 Alternate Mode 15
–	PG0_GPIO40	MII_RXD2	U4RX	–	MII_TXCK
–	PG1_GPIO41	MII_RXD1	U4TX	–	MII_TXER
USB0DM	PG2_GPIO42	–	–	–	–
–	PG3_GPIO43	MII_RXDV	–	TRACED1	–
–	PG4_GPIO44	–	–	CAN1RX	–
USB0DP	PG5_GPIO45	–	–	–	–
USB0ID	PG6_GPIO46	–	–	–	–
–	PG7_GPIO47	–	–	–	MII_CRS
–	PH0_GPIO48	–	SSI3TX	–	MII_TXD3
–	PH1_GPIO49	MII_RXD0	SSI3RX	–	MII_TXD2
–	PH2_GPIO50	–	SSI3CLK	–	MII_TXD1
–	PH3_GPIO51	–	SSI3FSS	–	MII_TXD0
–	PH4_GPIO52	–	U3TX	–	MII_COL
–	PH5_GPIO53	–	U3RX	–	MII_PHYRST
–	PH6_GPIO54	MII_TXEN	SSI0TX	–	MII_PHYINTR
–	PH7_GPIO55	MII_TXCK	SSI0RX	–	MII_MDC
–	PJ0_GPIO56	–	SSI0CLK	–	MII_MDIO
–	PJ1_GPIO57	MII_RXDV	SSI0FSS	–	MII_RXD3
–	PJ2_GPIO58	MII_RXCK	SSI0CLK	U0TX	MII_RXD2
–	PJ3_GPIO59	MII_MDC	SSI0FSS	U0RX	MII_RXD1
–	PJ4_GPIO60	MII_COL	SSI1CLK	–	MII_RXD0
–	PJ5_GPIO61	MII_CRS	SSI1FSS	–	MII_RXDV
–	PJ6_GPIO62	MII_PHYINTR	U2RX	–	MII_RXER
–	PJ7_GPIO63/ XCLKIN	MII_PHYRST	U2TX	–	MII_RXCK
–	PC0_GPIO64	–	–	–	MII_RXD2
–	PC1_GPIO65	–	–	–	MII_COL
–	PC2_GPIO66	–	–	–	MII_TXEN
–	PC3_GPIO67	–	–	–	MII_TXCK
–	PC4_GPIO68	–	–	–	–
–	PC5_GPIO69	–	–	–	–
–	PC6_GPIO70	–	–	–	–
–	PC7_GPIO71	–	–	–	–
–	PK0_GPIO72	–	SSI0TX	–	–
–	PK1_GPIO73	–	SSI0RX	–	–
–	PK2_GPIO74	–	SSI0CLK	–	–
–	PK3_GPIO75	–	SSI0FSS	–	–
–	PK4_GPIO76	MII_TXEN	SSI0TX	–	–
–	PK5_GPIO77	MII_TXCK	SSI0RX	–	–
–	PK6_GPIO78	MII_TXER	SSI0CLK	–	–
–	PK7_GPIO79	MII_CRS	SSI0FSS	–	–

ADVANCE INFORMATION

Table 2-28. GPIO_MUX1 Pin Assignments (M3 Alternate Modes)(1) (continued)

Analog Mode (USB Pins)	Device Pin Name	M3 Alternate Mode 12	M3 Alternate Mode 13	M3 Alternate Mode 14	M3 Alternate Mode 15
–	PL0_GPIO80	MII_RXD3	–	–	SSI1TX
–	PL1_GPIO81	MII_RXD2	–	–	SSI1RX
–	PL2_GPIO82	MII_RXD1	–	–	SSI1CLK
–	PL3_GPIO83	MII_RXD0	–	–	SSI1FSS
–	PL4_GPIO84	MII_COL	SSI3TX	–	–
–	PL5_GPIO85	$\overline{\text{MII\_PHYRST}}$	SSI3RX	–	–
–	PL6_GPIO86	$\overline{\text{MII\_PHYINTR}}$	SSI3CLK	–	–
–	PL7_GPIO87	MII_MDC	SSI3FSS	–	–
–	PM0_GPIO88	MII_MDIO	SSI2TX	–	–
–	PM1_GPIO89	MII_TXD3	SSI2RX	–	–
–	PM2_GPIO90	MII_TXD2	SSI2CLK	–	–
–	PM3_GPIO91	MII_TXD1	SSI2FSS	–	–
–	PM4_GPIO92	MII_TXD0	–	–	–
–	PM5_GPIO93	MII_RXDV	–	–	–
–	PM6_GPIO94	MII_RXER	–	–	–
–	PM7_GPIO95	MII_RXCK	–	–	–
–	PN0_GPIO96	–	I2C0SCL	–	–
–	PN1_GPIO97	–	I2C0SDA	–	–
–	PN2_GPIO98	–	U1RX	–	–
–	PN3_GPIO99	–	U1TX	–	–
–	PN4_GPIO100	–	U3TX	–	–
–	PN5_GPIO101	–	U3RX	–	–
–	PN6_GPIO102	–	U4RX	EPI0S42(2)	USB0EPEN
–	PN7_GPIO103	–	U4TX	EPI0S43(2)	USB0PFLT
–	PP0_GPIO104	–	I2C1SCL	–	–
–	PP1_GPIO105	–	I2C1SDA	–	–
–	PP2_GPIO106	–	I2C0SCL	–	–
–	PP3_GPIO107	–	I2C0SDA	–	–
–	PP4_GPIO108	–	I2C1SCL	–	–
–	PP5_GPIO109	–	I2C1SDA	–	–
–	PP6_GPIO110	–	–	–	–
–	PP7_GPIO111	–	–	–	–
–	PQ0_GPIO112	–	–	–	–
–	PQ1_GPIO113	–	–	–	–
–	PQ2_GPIO114	–	–	U0RX	–
–	PQ3_GPIO115	–	–	U0TX	–
–	PQ4_GPIO116	–	SSI1TX	–	–
–	PQ5_GPIO117	–	SSI1RX	–	–
–	PQ6_GPIO118	–	–	–	–
–	PQ7_GPIO119	–	–	–	–

Table 2-28. GPIO_MUX1 Pin Assignments (M3 Alternate Modes)(1) (continued)

Analog Mode (USB Pins)	Device Pin Name	M3 Alternate Mode 12	M3 Alternate Mode 13	M3 Alternate Mode 14	M3 Alternate Mode 15
–	PR0_GPIO120	–	SSI3TX	–	–
–	PR1_GPIO121	–	SSI3RX	–	–
–	PR2_GPIO122	–	SSI3CLK	–	–
–	PR3_GPIO123	–	SSI3FSS	–	–
–	PR4_GPIO124	–	–	–	–
–	PR5_GPIO125	–	–	–	–
–	PR6_GPIO126	–	–	–	–
–	PR7_GPIO127	–	–	–	–
–	PS0_GPIO128	–	–	–	–
–	PS1_GPIO129	–	–	–	–
–	PS2_GPIO130	–	–	–	–
–	PS3_GPIO131	–	–	–	–
–	PS4_GPIO132	–	–	–	–
–	PS5_GPIO133	–	–	–	–
–	PS6_GPIO134	–	–	–	–
–	PS7_GPIO135	–	–	–	–

ADVANCE INFORMATION

Table 2-29. GPIO_MUX1 Pin Assignments (C28x Peripheral Modes)(1)

Analog Mode (USB Pins)	Device Pin Name	C28x Peripheral Mode 0	C28x Peripheral Mode 1	C28x Peripheral Mode 2	C28x Peripheral Mode 3
–	PA0_GPIO0	GPIO0	EPWM1A	–	–
–	PA1_GPIO1	GPIO1	EPWM1B	ECAP6	–
–	PA2_GPIO2	GPIO2	EPWM2A	–	–
–	PA3_GPIO3	GPIO3	EPWM2B	ECAP5	–
–	PA4_GPIO4	GPIO4	EPWM3A	–	–
–	PA5_GPIO5	GPIO5	EPWM3B	MFSRA	ECAP1
–	PA6_GPIO6	GPIO6	EPWM4A	–	EPWMSYNCO
–	PA7_GPIO7	GPIO7	EPWM4B	MCLKRA	ECAP2
–	PB0_GPIO8	GPIO8	EPWM5A	–	$\overline{\text{ADCSOCAO}}$
–	PB1_GPIO9	GPIO9	EPWM5B	–	ECAP3
–	PB2_GPIO10	GPIO10	EPWM6A	–	$\overline{\text{ADCSOCBO}}$
–	PB3_GPIO11	GPIO11	EPWM6B	–	ECAP4
–	PB4_GPIO12	GPIO12	EPWM7A	–	–
–	PB5_GPIO13	GPIO13	EPWM7B	–	–
–	PB6_GPIO14	GPIO14	EPWM8A	–	–
–	PB7_GPIO15	GPIO15	EPWM8B	–	–
–	PD0_GPIO16	GPIO16	SPISIMOA	–	–
–	PD1_GPIO17	GPIO17	SPISOMIA	–	–
–	PD2_GPIO18	GPIO18	SPICLKA	–	–
–	PD3_GPIO19	GPIO19	SPISTEA	–	–
–	PD4_GPIO20	GPIO20	EQEP1A	MDXA	–
–	PD5_GPIO21	GPIO21	EQEP1B	MDRA	–
–	PD6_GPIO22	GPIO22	EQEP1S	MCLKXA	–
–	PD7_GPIO23	GPIO23	EQEP1I	MFSXA	–
–	PE0_GPIO24	GPIO24	ECAP1	EQEP2A	–
–	PE1_GPIO25	GPIO25	ECAP2	EQEP2B	–
–	PE2_GPIO26	GPIO26	ECAP3	EQEP2I	–
–	PE3_GPIO27	GPIO27	ECAP4	EQEP2S	–
–	PE4_GPIO28	GPIO28	SCIRXDA	–	–
–	PE5_GPIO29	GPIO29	SCITXDA	–	–
–	PE6_GPIO30	GPIO30	–	–	EPWM9A
–	PE7_GPIO31	GPIO31	–	–	EPWM9B
–	PF0_GPIO32	GPIO32	I2CASDA	SCIRXDA	$\overline{\text{ADCSOCAO}}$
–	PF1_GPIO33	GPIO33	I2CASCL	EPWMSYNCO	$\overline{\text{ADCSOCBO}}$
–	PF2_GPIO34	GPIO34	ECAP1	SCIRXDA	XCLKOUT
–	PF3_GPIO35	GPIO35	SCITXDA	–	–
–	PF4_GPIO36	GPIO36	SCIRXDA	–	–
–	PF5_GPIO37	GPIO37	ECAP2	–	–
USB0VBUS	PF6_GPIO38	GPIO38	–	–	–
–	PF7_GPIO39	GPIO39	–	–	–

(1) Blank fields represent Reserved functions.

Table 2-29. GPIO_MUX1 Pin Assignments (C28x Peripheral Modes)(1) (continued)

Analog Mode (USB Pins)	Device Pin Name	C28x Peripheral Mode 0	C28x Peripheral Mode 1	C28x Peripheral Mode 2	C28x Peripheral Mode 3
–	PG0_GPIO40	GPIO40	–	–	–
–	PG1_GPIO41	GPIO41	–	–	–
USB0DM	PG2_GPIO42	GPIO42	–	–	–
–	PG3_GPIO43	GPIO43	–	–	–
–	PG4_GPIO44	GPIO44	–	–	–
USB0DP	PG5_GPIO45	GPIO45	–	–	–
USB0ID	PG6_GPIO46	GPIO46	–	–	–
–	PG7_GPIO47	GPIO47	–	–	–
–	PH0_GPIO48	GPIO48	ECAP5	–	–
–	PH1_GPIO49	GPIO49	ECAP6	–	–
–	PH2_GPIO50	GPIO50	EQEP1A	–	–
–	PH3_GPIO51	GPIO51	EQEP1B	–	–
–	PH4_GPIO52	GPIO52	EQEP1S	–	–
–	PH5_GPIO53	GPIO53	EQEP1I	–	–
–	PH6_GPIO54	GPIO54	SPISIMOA	–	EQEP3A
–	PH7_GPIO55	GPIO55	SPISOMIA	–	EQEP3B
–	PJ0_GPIO56	GPIO56	SPICLKA	–	EQEP3S
–	PJ1_GPIO57	GPIO57	SPISTEA	–	EQEP3I
–	PJ2_GPIO58	GPIO58	MCLKRA	–	EPWM7A
–	PJ3_GPIO59	GPIO59	MFSRA	–	EPWM7B
–	PJ4_GPIO60	GPIO60	–	–	EPWM8A
–	PJ5_GPIO61	GPIO61	–	–	EPWM8B
–	PJ6_GPIO62	GPIO62	–	–	EPWM9A
–	PJ7_GPIO63/ XCLKIN	GPIO63	–	–	EPWM9B
–	PC0_GPIO64	GPIO64	EQEP1A	EQEP2I	–
–	PC1_GPIO65	GPIO65	EQEP1B	EQEP2S	–
–	PC2_GPIO66	GPIO66	EQEP1S	EQEP2A	–
–	PC3_GPIO67	GPIO67	EQEP1I	EQEP2B	–
–	PC4_GPIO68	GPIO68	–	–	–
–	PC5_GPIO69	GPIO69	–	–	–
–	PC6_GPIO70	GPIO70	–	–	–
–	PC7_GPIO71	GPIO71	–	–	–
–	PK0_GPIO72	GPIO72	SPISIMOA	–	–
–	PK1_GPIO73	GPIO73	SPISOMIA	–	–
–	PK2_GPIO74	GPIO74	SPICLKA	–	–
–	PK3_GPIO75	GPIO75	SPISTEA	–	–
–	PK4_GPIO76	GPIO76	–	–	–
–	PK5_GPIO77	GPIO77	–	–	–
–	PK6_GPIO78	GPIO78	–	–	–
–	PK7_GPIO79	GPIO79	–	–	–

Table 2-29. GPIO_MUX1 Pin Assignments (C28x Peripheral Modes)(1) (continued)

Analog Mode (USB Pins)	Device Pin Name	C28x Peripheral Mode 0	C28x Peripheral Mode 1	C28x Peripheral Mode 2	C28x Peripheral Mode 3
–	PL0_GPIO80	GPIO80	–	–	–
–	PL1_GPIO81	GPIO81	–	–	–
–	PL2_GPIO82	GPIO82	–	–	–
–	PL3_GPIO83	GPIO83	–	–	–
–	PL4_GPIO84	GPIO84	–	–	–
–	PL5_GPIO85	GPIO85	–	–	–
–	PL6_GPIO86	GPIO86	–	–	–
–	PL7_GPIO87	GPIO87	–	–	–
–	PM0_GPIO88	GPIO88	–	–	–
–	PM1_GPIO89	GPIO89	–	–	–
–	PM2_GPIO90	GPIO90	–	–	–
–	PM3_GPIO91	GPIO91	–	–	–
–	PM4_GPIO92	GPIO92	–	MDXA	–
–	PM5_GPIO93	GPIO93	–	MDRA	–
–	PM6_GPIO94	GPIO94	–	MCLKXA	–
–	PM7_GPIO95	GPIO95	–	MFSXA	–
–	PN0_GPIO96	GPIO96	–	MCLKRA	–
–	PN1_GPIO97	GPIO97	–	MFSRA	–
–	PN2_GPIO98	GPIO98	–	–	–
–	PN3_GPIO99	GPIO99	–	–	–
–	PN4_GPIO100	GPIO100	–	–	–
–	PN5_GPIO101	GPIO101	–	–	–
–	PN6_GPIO102	GPIO102	–	–	–
–	PN7_GPIO103	GPIO103	–	–	–
–	PP0_GPIO104	GPIO104	I2CSDAA	–	–
–	PP1_GPIO105	GPIO105	I2CSCLA	–	–
–	PP2_GPIO106	GPIO106	EQEP1A	–	–
–	PP3_GPIO107	GPIO107	EQEP1B	–	–
–	PP4_GPIO108	GPIO108	EQEP1S	–	–
–	PP5_GPIO109	GPIO109	EQEP1I	–	–
–	PP6_GPIO110	GPIO110	–	EQEP2A	EQEP3S
–	PP7_GPIO111	GPIO111	–	EQEP2B	EQEP3I
–	PQ0_GPIO112	GPIO112	–	EQEP2I	EQEP3A
–	PQ1_GPIO113	GPIO113	–	EQEP2S	EQEP3B
–	PQ2_GPIO114	GPIO114	–	–	–
–	PQ3_GPIO115	GPIO115	–	–	–
–	PQ4_GPIO116	GPIO116	–	–	–
–	PQ5_GPIO117	GPIO117	–	–	–
–	PQ6_GPIO118	GPIO118	–	SCITXDA	–
–	PQ7_GPIO119	GPIO119	–	SCIRXDA	–

Table 2-29. GPIO_MUX1 Pin Assignments (C28x Peripheral Modes)[1] (continued)

Analog Mode (USB Pins)	Device Pin Name	C28x Peripheral Mode 0	C28x Peripheral Mode 1	C28x Peripheral Mode 2	C28x Peripheral Mode 3
–	PR0_GPIO120	GPIO120	–	–	–
–	PR1_GPIO121	GPIO121	–	–	–
–	PR2_GPIO122	GPIO122	–	–	–
–	PR3_GPIO123	GPIO123	–	–	–
–	PR4_GPIO124	GPIO124	EPWM7A	–	–
–	PR5_GPIO125	GPIO125	EPWM7B	–	–
–	PR6_GPIO126	GPIO126	EPWM8A	–	–
–	PR7_GPIO127	GPIO127	EPWM8B	–	–
–	PS0_GPIO128	GPIO128	EPWM9A	–	–
–	PS1_GPIO129	GPIO129	EPWM9B	–	–
–	PS2_GPIO130	GPIO130	EPWM10A	–	–
–	PS3_GPIO131	GPIO131	EPWM10B	–	–
–	PS4_GPIO132	GPIO132	EPWM11A	–	–
–	PS5_GPIO133	GPIO133	EPWM11B	–	–
–	PS6_GPIO134	GPIO134	EPWM12A	–	–
–	PS7_GPIO135	GPIO135	EPWM12B	–	–

ADVANCE INFORMATION

2.16.2 GPIO_MUX2

The eight pins of the GPIO_MUX2 block can be selectively mapped to eight General-Purpose Inputs, eight General-Purpose Outputs, or six COMPOUT outputs from the Analog Comparator peripheral. Each GPIO_MUX2 pin can have a pullup enabled or disabled. On reset, all pins of the GPIO_MUX2 block are configured as analog inputs, and the GPIO function is disabled. The GPIO_MUX2 block is programmed through a separate set of registers from those used to program GPIO_MUX1.

The multiple registers responsible for configuring the GPIO_MUX2 pins are organized in register set G. They are accessible by the C28x CPU only. The middle portion of Figure 2-16 shows set G of Control Subsystem registers, plus muxing logic for the associated eight GPIO pins. The GPGMUX1 register selects one of six possible digital output signals from analog comparators, or one of eight general-purpose GPIO digital outputs. The GPGPUD register disables pullups for the GPIO_MUX2 pins when a corresponding bit of that register is set to “1”. Other registers of set G allow reading and writing of the eight GPIO bits, as well as setting the direction for each of the bits (read or write). See Table 2-30 for the mapping of comparator outputs and GPIO to the eight pins of GPIO_MUX2.

Peripheral Modes 0, 1, 2, and 3 are chosen by setting selected bit pairs of GPGMUX1 register to “00”, “01”, “10”, and “11”, respectively. For example, setting bits 5–4 of the GPGMUX1 register to “00” (Peripheral Mode 0) assigns pin GPIO194 to internal signal GPIO194 (digital GPIO). Setting bits 5–4 of the GPGMUX1 register to “11” (Peripheral Mode 3) assigns pin GPIO194 to internal signal COMP6OUT coming from Analog Comparator 6. Peripheral Modes 1 and 2 are reserved and are not currently available.

Table 2-30. GPIO_MUX2 Pin Assignments (C28x Peripheral Modes)(1)

Device Pin Name	C28x Peripheral Mode 0	C28x Peripheral Mode 1	C28x Peripheral Mode 2	C28x Peripheral Mode 3
GPIO192	GPIO192	–	–	–
GPIO193	GPIO193	–	–	COMP1OUT
GPIO194	GPIO194	–	–	COMP6OUT
GPIO195	GPIO195	–	–	COMP2OUT
GPIO196	GPIO196	–	–	COMP3OUT
GPIO197	GPIO197	–	–	COMP4OUT
GPIO198	GPIO198	–	–	–
GPIO199	GPIO199	–	–	COMP5OUT

(1) Blank fields represent Reserved functions.

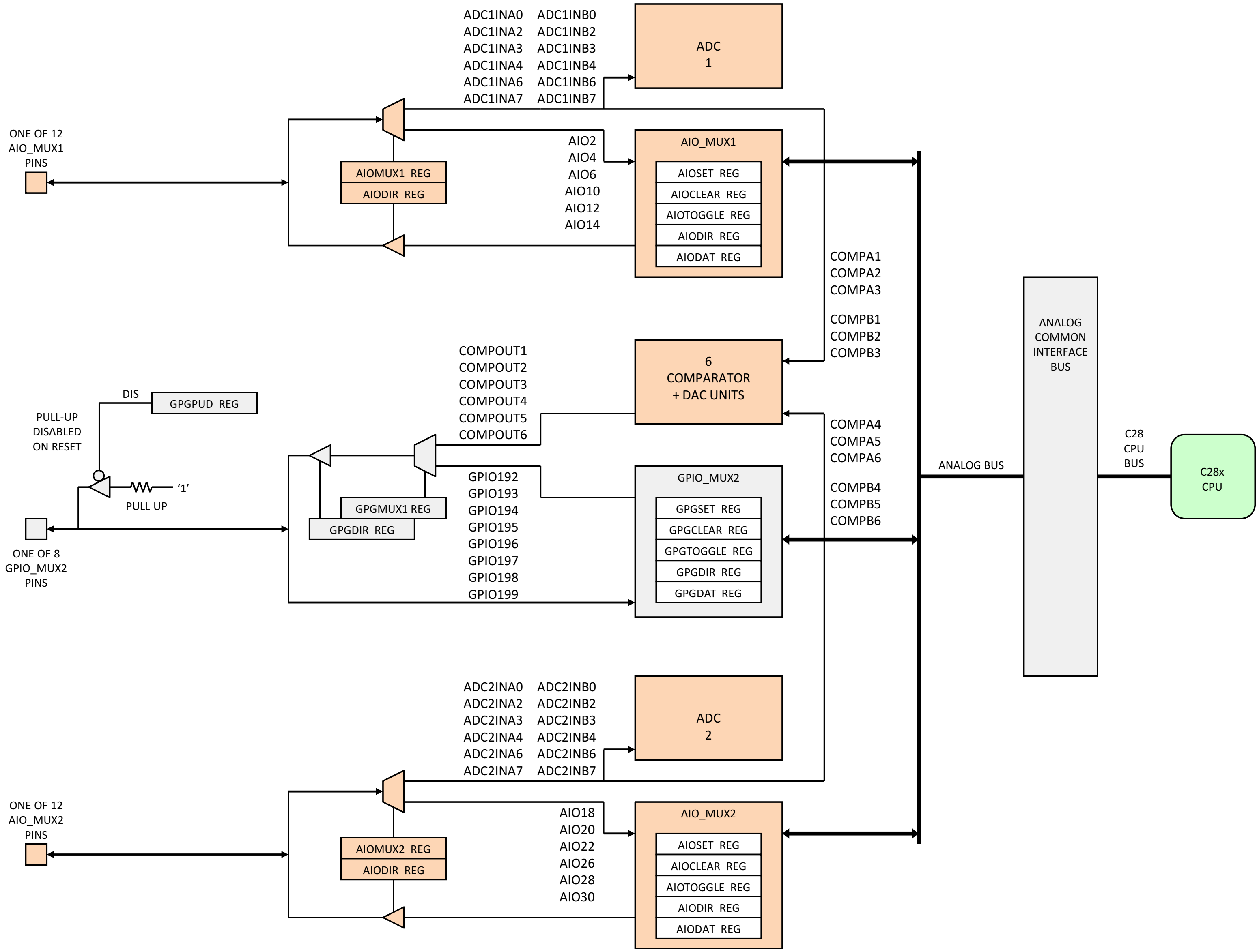


Figure 2-16. Pin Muxing on AIO_MUX1, AIO_MUX2, and GPIO_MUX2

2.16.3 AIO_MUX1

The 12 pins of AIO_MUX1 can be selectively mapped through a dedicated set of registers to 12 analog inputs for ADC1 peripheral, six analog inputs for Comparator peripherals, four General-Purpose Inputs, or four General-Purpose Outputs. Note that while AIO_MUX1 has been named after the analog signals passing through it, the GPIOs (here called AIOs) are still digital, although with fewer features than those in the GPIO_MUX1 and GPIO_MUX2 blocks—for example, they do not offer pullups. On reset, all pins of the AIO_MUX1 block are configured as analog inputs and the GPIO function is disabled. The AIO_MUX1 block is programmed through a separate set of registers from those used to program AIO_MUX2.

The multiple registers responsible for configuring the AIO_MUX1 pins are accessible by the C28x CPU only. The top portion of Figure 2-16 shows Control Subsystem registers and muxing logic for the associated 12 AIO pins. The AIOMUX1 register selects 1 of 12 possible analog input signals or 1 of 6 general-purpose AIO inputs. Other registers allow reading and writing of the 6 AIO bits, as well as setting the direction for each of the bits (read or write). See Table 2-31 for the mapping of analog inputs and AIOs to the 12 pins of AIO_MUX1.

AIO Mode 0 is chosen by setting selected odd bits of the AIOMUX1 register to ‘0’. AIO Mode 1 is chosen by setting selected odd bits of the AIOMUX1 register to ‘1’. For example, setting bit 5 of the AIOMUX1 register to ‘0’ assigns pin ADC1INA2 to internal signal AIO2 (digital GPIO). Setting bit 5 of the AIOMUX1 register to ‘1’ assigns pin ADC1INA2 to analog inputs ADC1INA2 or COMPA1 (only one should be enabled at a time in the respective analog module). Currently, all even bits of the AIOMUX1 register are “don’t cares”.

Table 2-31. AIO_MUX1 Pin Assignments (C28x AIO Modes)(1)(2)

Device Pin Name	C28x AIO Mode 0(3)	C28x AIO Mode 1(4)
ADC1INA0	–	ADC1INA0
ADC1INA2	AIO2	ADC1INA2, COMPA1
ADC1INA3	–	ADC1INA3
ADC1INA4	AIO4	ADC1INA4, COMPA2
ADC1INA6	AIO6	ADC1INA6, COMPA3
ADC1INA7	–	ADC1INA7
ADC1INB0	–	ADC1INB0
ADC2INB2	AIO10	ADC2INB2, COMPB1
ADC1INB3	–	ADC1INB3
ADC1INB4	AIO12	ADC1INB4, COMPB2
ADC2INB6	AIO14	ADC2INB6, COMPB3
ADC1INB7	–	ADC1INB7

(1) Blank fields represent Reserved functions.
(2) For each field with two pins (for example, ADC1INA2, COMPA1), only one pin should be enabled at a time; the other pin should be disabled. Use registers inside the respective destination analog peripherals to enable or disable these inputs.
(3) AIO Mode 0 represents digital general-purpose inputs or outputs.
(4) AIO Mode 1 represents analog inputs for ADC1 or the Comparator module.

2.16.4 AIO_MUX2

The 12 pins of AIO_MUX2 can be selectively mapped through a dedicated set of registers to 12 analog inputs for ADC2 peripheral, six analog inputs for Comparator peripherals, four General-Purpose Inputs, or four General-Purpose Outputs. Note that while AIO_MUX2 has been named after the analog signals passing through it, the GPIOs (here called AIOs) are still digital, although with fewer features than those in the GPIO_MUX1 and GPIO_MUX2 blocks—for example, they do not offer pullups. On reset, all pins of the AIO_MUX2 block are configured as analog inputs and the GPIO function is disabled. The AIO_MUX2 block is programmed through a separate set of registers from those used to program AIO_MUX1.

The multiple registers responsible for configuring the AIO_MUX2 pins are accessible by the C28x CPU only. The bottom portion of Figure 2-16 shows Control Subsystem registers and muxing logic for the associated 12 AIO pins. The AIOMUX2 register selects 1 of 12 possible analog input signals or 1 of 6 general-purpose AIO inputs. Other registers allow reading and writing of the 6 AIO bits, as well as setting the direction for each of the bits (read or write). See Table 2-32 for the mapping of analog inputs and AIOs to the 12 pins of AIO_MUX2. Peripheral Modes 1 and 2 are currently not available.

AIO Mode 0 is chosen by setting selected odd bits of the AIOMUX2 register to ‘0’. AIO Mode 1 is chosen by setting selected odd bits of the AIOMUX2 register to ‘1’. For example, setting bit 9 of the AIOMUX2 register to ‘0’ assigns pin ADC2INA4 to internal signal AIO20 (digital GPIO). Setting bit 9 of the AIOMUX2 register to ‘1’ assigns pin ADC2INA4 to analog inputs ADC2INA4 or COMPA5 (only one should be enabled at a time in the respective analog module). Currently, all even bits of the AIOMUX2 register are “don’t cares”.

Table 2-32. AIO_MUX2 Pin Assignments (C28x AIO Modes)(1)(2)

Device Pin Name	C28x AIO Mode 0(3)	C28x AIO Mode 1(4)
ADC2INA0	–	ADC2INA0
ADC2INA2	AIO18	ADC2INA2, COMPA4
ADC2INA3	–	ADC2INA3
ADC2INA4	AIO20	ADC2INA4, COMPA5
ADC2INA6	AIO22	ADC2INA6, COMPA6
ADC2INA7	–	ADC2INA7
ADC2INB0	–	ADC2INB0
ADC2INB2	AIO26	ADC2INB2, COMPB4
ADC2INB3	–	ADC2INB3
ADC2INB4	AIO28	ADC2INB4, COMPB5
ADC2INB6	AIO30	ADC2INB6, COMPB6
ADC2INB7	–	ADC2INB7

(1) Blank fields represent Reserved functions.
(2) For each field with two pins (for example, ADC2INA6, COMPA6), only one pin should be enabled at a time; the other pin should be disabled. Use registers inside the respective destination analog peripherals to enable or disable these inputs.
(3) AIO Mode 0 represents digital general-purpose inputs or outputs.
(4) AIO Mode 1 represents analog inputs for ADC2 or the Comparator module.

2.17 Emulation/JTAG

Concerto devices have two types of emulation ports to support debug operations: the 7-pin TI JTAG port and the 5-pin Cortex™-M3 Instrumentation Trace Macrocell (ITM) port. The 7-pin TI JTAG port can be used to connect to debug tools via the TI 14-pin JTAG header or the TI 20-pin JTAG header. The 5-pin Cortex™-M3 ITM port can only be accessed through the TI 20-pin JTAG header.

The JTAG port has seven dedicated pins: $\overline{\text{TRST}}$, TMS, TDI, TDO, TCK, EMU0, and EMU1. The $\overline{\text{TRST}}$ signal should always be pulled down via a 2.2-kΩ pulldown resistor on the board. EMU0 and EMU1 signals should be pulled up through a pair of pullups ranging from 2.2 kΩ to 4.7 kΩ (depending on the drive strength of the debugger ports). The JTAG port is TI's standard debug port.

The ITM port uses five GPIO pins that can be mapped to internal Cortex™-M3 ITM trace signals: TRACE0, TRACE1, TRACE2, TRACE3, and TRACECLK. This port is typically used for advanced software debug.

TI emulators, and those from other manufacturers, can connect to Concerto devices via TI's 14-pin JTAG header or 20-pin JTAG header. See Figure 2-17 to see how the 14-pin JTAG header connects to Concerto's JTAG port signals. Note that the 14-pin header does not support the ITM debug mode.

Figure 2-18 shows two possible ways to connect the 20-pin header to Concerto's emulation pins. The left side of the drawing shows all seven JTAG signals connecting to the 20-pin header similar to the way the 14-pin header was connected. Note that the JTAG EMU0 and EMU1 signals are mapped to the corresponding terminals on the 20-pin header. In this mode, header terminals EMU2, EMU3, and EMU4 are left unconnected and the ITM trace mode is not available.

The right side of the drawing shows the same 20-pin header now connected to five ITM signals and five of seven JTAG signals. Note that Concerto's EMU0 and EMU1 signals are left unconnected in this mode; thus, the emulation functions associated with these two signals are not available when debugging with ITM trace.

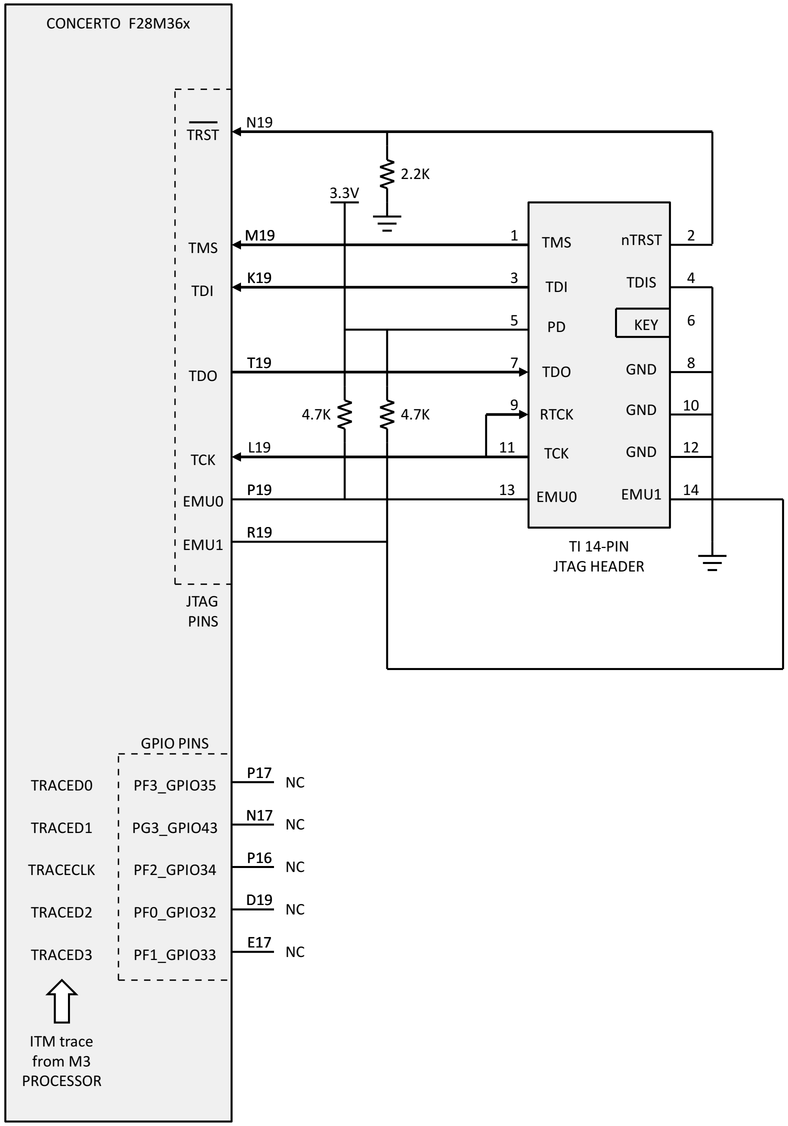


Figure 2-17. Connecting to TI 14-Pin JTAG Emulator Header

ADVANCE INFORMATION

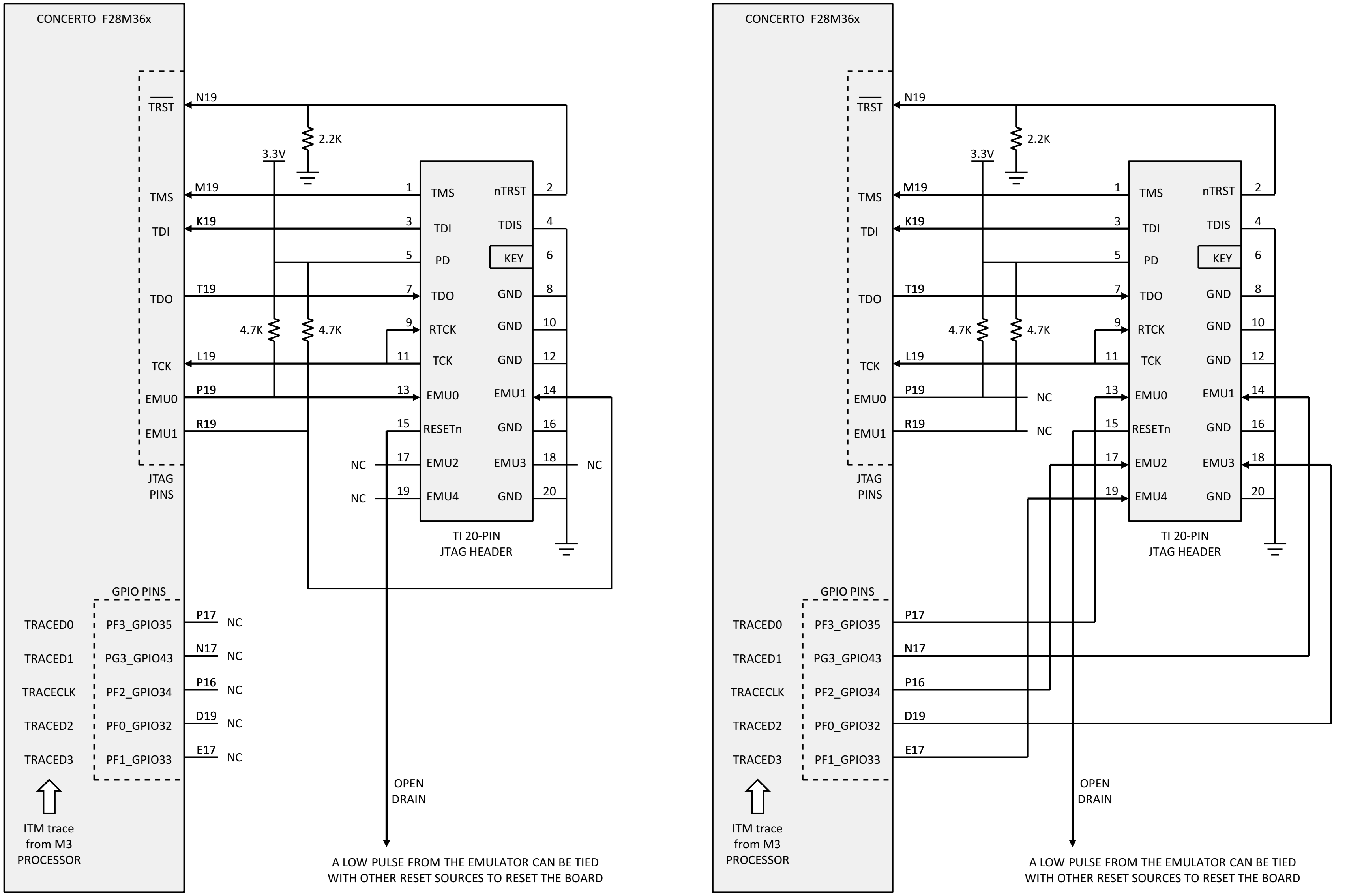


Figure 2-18. Connecting to TI 20-Pin JTAG Emulator Header

2.18 Code Security Module (CSM)

The Code Security Module (CSM) is a security feature incorporated in Concerto™ devices. The CSM prevents access and visibility to on-chip secure memories by unauthorized persons—that is, the CSM prevents duplication and reverse-engineering of proprietary code. The word "secure" means that access to on-chip secure memories is protected. The word "unsecure" means that access to on-chip secure memory is not protected—that is, the contents of the memory could be read by any means (for example, by using a debugging tool such as Code Composer Studio™).

2.18.1 Functional Description

The security module restricts the CPU access to on-chip secure memory without interrupting or stalling CPU execution. When a read occurs to a protected memory location, the read returns a zero value and CPU execution continues with the next instruction. This process, in effect, blocks read and write access to various memories through the JTAG port or external peripherals. Security is defined with respect to the access of on-chip secure memories and prevents unauthorized copying of proprietary code or data.

The zone is secure when CPU access to the on-chip secure memories associated with that zone is restricted. When secure, two levels of protection are possible, depending on where the program counter is currently pointing. If code is currently running from inside secure memory, only an access through JTAG is blocked (that is, through the emulator). This process allows secure code to access secure data. Conversely, if code is running from unsecure memory, all accesses to secure memories are blocked. User code can dynamically jump in and out of secure memory, thereby allowing secure function calls from unsecure memory. Similarly, interrupt service routines can be placed in secure memory, even if the main program loop is run from unsecure memory.

The code security mechanism present in this device offers dual-zone security for the Cortex™-M3 code and single-zone security for the C28x code. In case of dual-zone security on the master subsystem, the different secure memories (RAMs and flash sectors) can be assigned to different security zones by configuring the GRABRAM and GRABSECT registers associated with each zone. Flash Sector N and Flash Sector A are dedicated to Zone1 and Zone2, respectively, and cannot be allocated to any other zone by configuration. Similarly, flash sectors get assigned to different zones based on the setting in the GRABSECT registers.

Security is provided by a CSM password of 128 bits of data (four 32-bit words) that is used to secure or unsecure the zones. Each zone has its own 128-bit CSM password. The zone can be unsecured by executing the password match flow (PMF).

The CSM password for each zone is stored in its dedicated flash sector. The password storage locations in the flash sector store the CSM password. The password is selected by the system designer. If the password locations of a zone have all 128 bits as ones, the zone is considered "unsecure". Since new flash devices have erased flash (all ones), only a read of the password locations is required to bring any zone into unsecure mode. If the password locations of a zone have all 128 bits as zeros, the zone is considered "secure", regardless of the contents of the CSMKEY registers. The user should not use all zeros as a password or reset the device during an erase of the flash. Resetting the device during an erase routine can result in either an all-zero or unknown password. If a device is reset when the password locations are all zeros, the device cannot be unlocked by the password match flow. Using a password of all zeros will seriously limit the user's ability to debug secure code or reprogram the flash.

NOTE

If a device is reset while the password locations of a zone contain all zeros or an unknown value, that zone will be permanently locked unless a method to run the flash erase routine from secure SARAM is embedded into the flash or OTP. Care must be taken when implementing this procedure to avoid introducing a security hole.

2.19 µCRC Module

The µCRC module is part of the master subsystem. This module can be used by Cortex™-M3 software to compute CRC on data and program, which are stored at memory locations that are addressable by Cortex™-M3. On this device, the Cortex™-M3 Flash Bank and ROM are mapped to the code space that is only accessed by the ICODE/DCODE bus of Cortex™-M3; and RAMs are mapped on the SRAM space that is accessible by the SYSTEM bus. Hence, the µCRC module snoops both the DCODE and SYSTEM buses to support CRC calculation for data and program.

2.19.1 Functional Description

The µCRC module snoops both the DCODE and SYSTEM buses to support CRC calculation for data and program. To allow interrupts execution in between CRC calculations for a block of data and to discard the Cortex™-M3 literal pool accesses in between executions of the program (which reads data for CRC calculation), the Cortex™-M3 ROM, Flash, and RAMs are mapped to a mirrored memory location. The µCRC module grabs data from the bus to calculate CRC only if the address of the read data belongs to mirrored memory space. After grabbing, the µCRC module performs the CRC calculation on the grabbed data and updates the µCRC Result Register (µCRCRES). This register can be read at any time to get the calculated CRC for all the previous read data. The µCRC module only supports CRC calculation for byte accesses. So, in order to calculate the CRC on a block of data, software must perform byte accesses to all the data. For half-word and word accesses, the µCRC module discards the data and does not update the µCRCRES register.

NOTE

If a read to a mirrored address space is thrown from the debugger (Code Composer Studio or any other debug platform), the µCRC module ignores the read data and does not update the CRC result for that particular read.

2.19.2 CRC Polynomials

The following are the CRC polynomials that are supported by the µCRC module:

- CRC8 Polynomial = 0x07
- CRC16 Polynomial-1 = 0x8005
- CRC16 Polynomial-2 = 0x1021
- CRC32 Polynomial = 0x04C11DB7

2.19.3 CRC Calculation Procedure

The software procedure for calculating CRC for a set of data that is stored in Cortex™-M3 addressable memory space is as follows:

1. Save the current value of the µCRC Result Register (µCRCRES) into the stack to allow calculation of CRC in nested interrupt
2. Clear the µCRC Result Register (µCRCRES) by setting the CLEAR field of the µCRC Control Register (µCRCCONTROL) to "1"
3. Configure the µCRC polynomials (CRC8, CRC16-P1, CRC16-P2, or CRC32) in the µCRC Configuration Register (µCRCCONFIG)
4. Read the data from memory locations for which CRC needs to be calculated using mirrored address
5. Read the µCRCRES register to get the calculated CRC value. Pop the last saved value of the CRC from the stack and store this value into the µCRC Result Register (uCRCRES)

2.19.4 CRC Calculation for Data Stored In Secure Memory

This device has dual-zone security for the Cortex™-M3 subsystem. Since ZoneX (X → 1/2) software does not have access to program/data in ZoneY (Y → 2/1), code running from ZoneX cannot calculate CRC on data stored in ZoneY memory. Similarly, in the case of Exe-Only flash sectors, even though software is running from same secure zone, the software cannot read the data stored in Exe-Only sectors. However, hardware does allow CRC computation on data stored in Exe-Only flash sectors as long as the read access for this data is initiated by code running from same secure zone. These reads are just dummy reads and, in this case, read data only goes to the µCRC module, not to the CPU.

3 Device Pins

3.1 Pin Assignments

Figure 3-1 illustrates the ball locations for the 289-ball ZWT plastic ball grid array (PBGA) package and is used in conjunction with Figure 3-2, Figure 3-3, Figure 3-4, and Figure 3-5 to locate signal names and ball grid numbers.

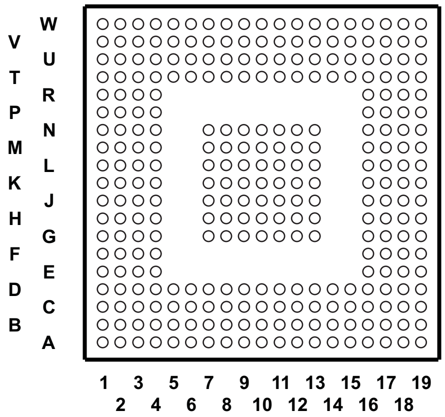
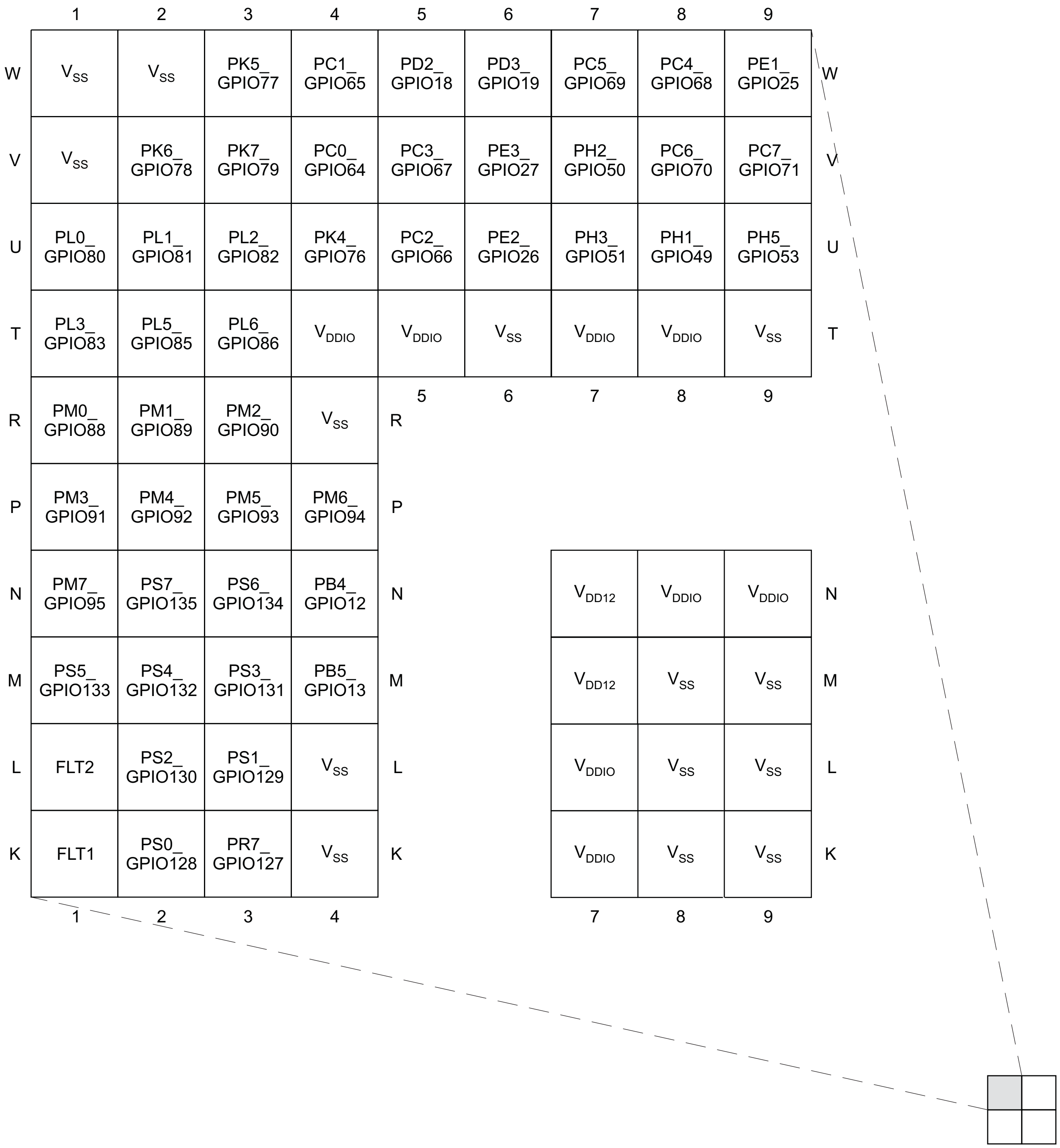


Figure 3-1. 289-Ball ZWT Ball Grid Array (Bottom View)

3.1.1 Pin Map (Bottom View)

Figure 3-2 through Figure 3-5 show the pin assignments on the 289-ball ZWT package in four quadrants (A, B, C, and D). See Table 3-1, Terminal Functions, for the complete multiplexed signal names.

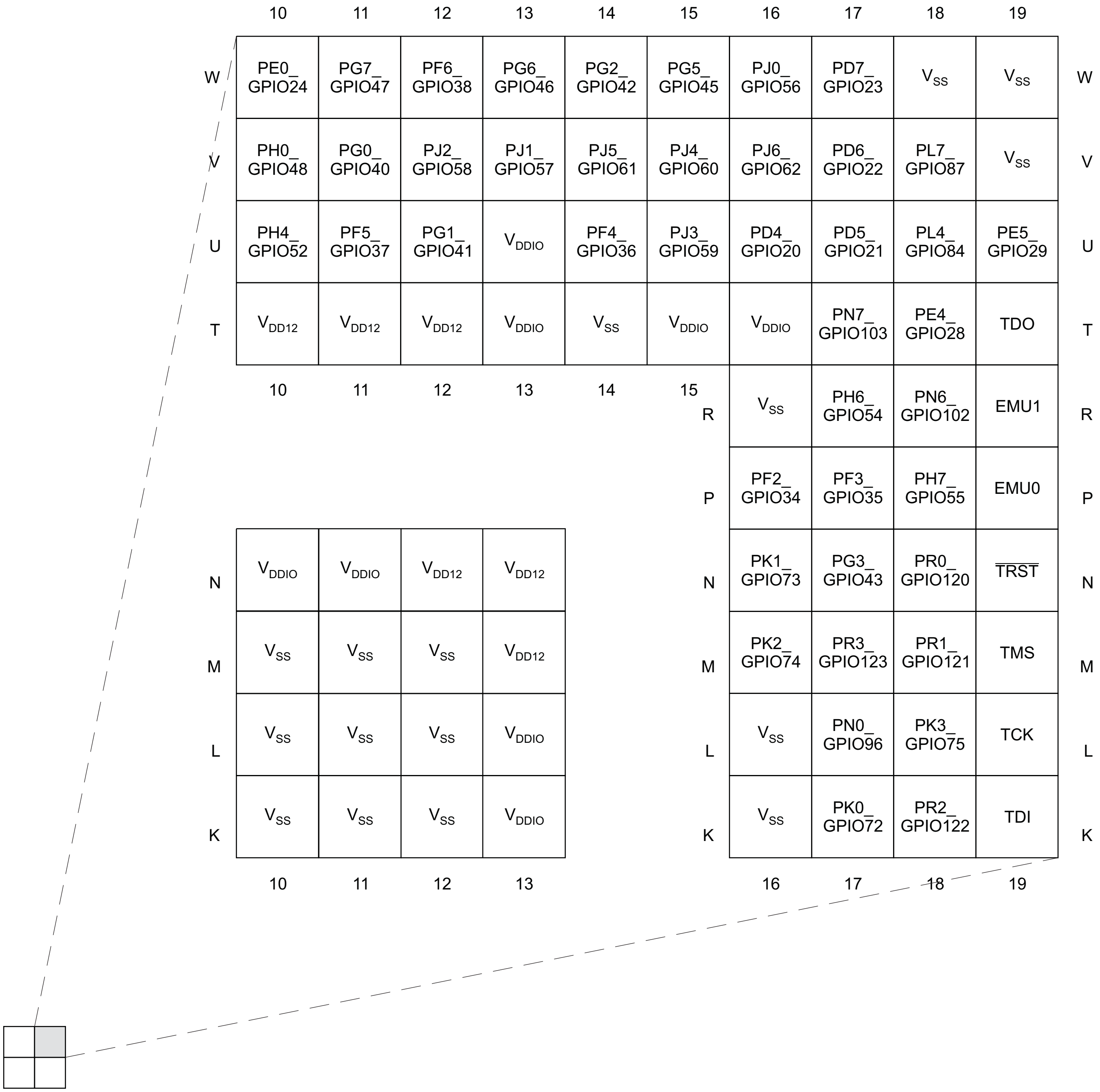


	1	2	3	4	5	6	7	8	9	
W	V_{SS}	V_{SS}	PK5_ GPIO77	PC1_ GPIO65	PD2_ GPIO18	PD3_ GPIO19	PC5_ GPIO69	PC4_ GPIO68	PE1_ GPIO25	W
V	V_{SS}	PK6_ GPIO78	PK7_ GPIO79	PC0_ GPIO64	PC3_ GPIO67	PE3_ GPIO27	PH2_ GPIO50	PC6_ GPIO70	PC7_ GPIO71	V
U	PL0_ GPIO80	PL1_ GPIO81	PL2_ GPIO82	PK4_ GPIO76	PC2_ GPIO66	PE2_ GPIO26	PH3_ GPIO51	PH1_ GPIO49	PH5_ GPIO53	U
T	PL3_ GPIO83	PL5_ GPIO85	PL6_ GPIO86	V_{DDIO}	V_{DDIO}	V_{SS}	V_{DDIO}	V_{DDIO}	V_{SS}	T
R	PM0_ GPIO88	PM1_ GPIO89	PM2_ GPIO90	V_{SS}						R
P	PM3_ GPIO91	PM4_ GPIO92	PM5_ GPIO93	PM6_ GPIO94						P
N	PM7_ GPIO95	PS7_ GPIO135	PS6_ GPIO134	PB4_ GPIO12			V_{DD12}	V_{DDIO}	V_{DDIO}	N
M	PS5_ GPIO133	PS4_ GPIO132	PS3_ GPIO131	PB5_ GPIO13			V_{DD12}	V_{SS}	V_{SS}	M
L	FLT2	PS2_ GPIO130	PS1_ GPIO129	V_{SS}			V_{DDIO}	V_{SS}	V_{SS}	L
K	FLT1	PS0_ GPIO128	PR7_ GPIO127	V_{SS}			V_{DDIO}	V_{SS}	V_{SS}	K
	1	2	3	4			7	8	9	

A. See Table 3-1, Terminal Functions, for the complete multiplexed signal names.

Figure 3-2. 289-Ball ZWT Pin Map [Quadrant A]

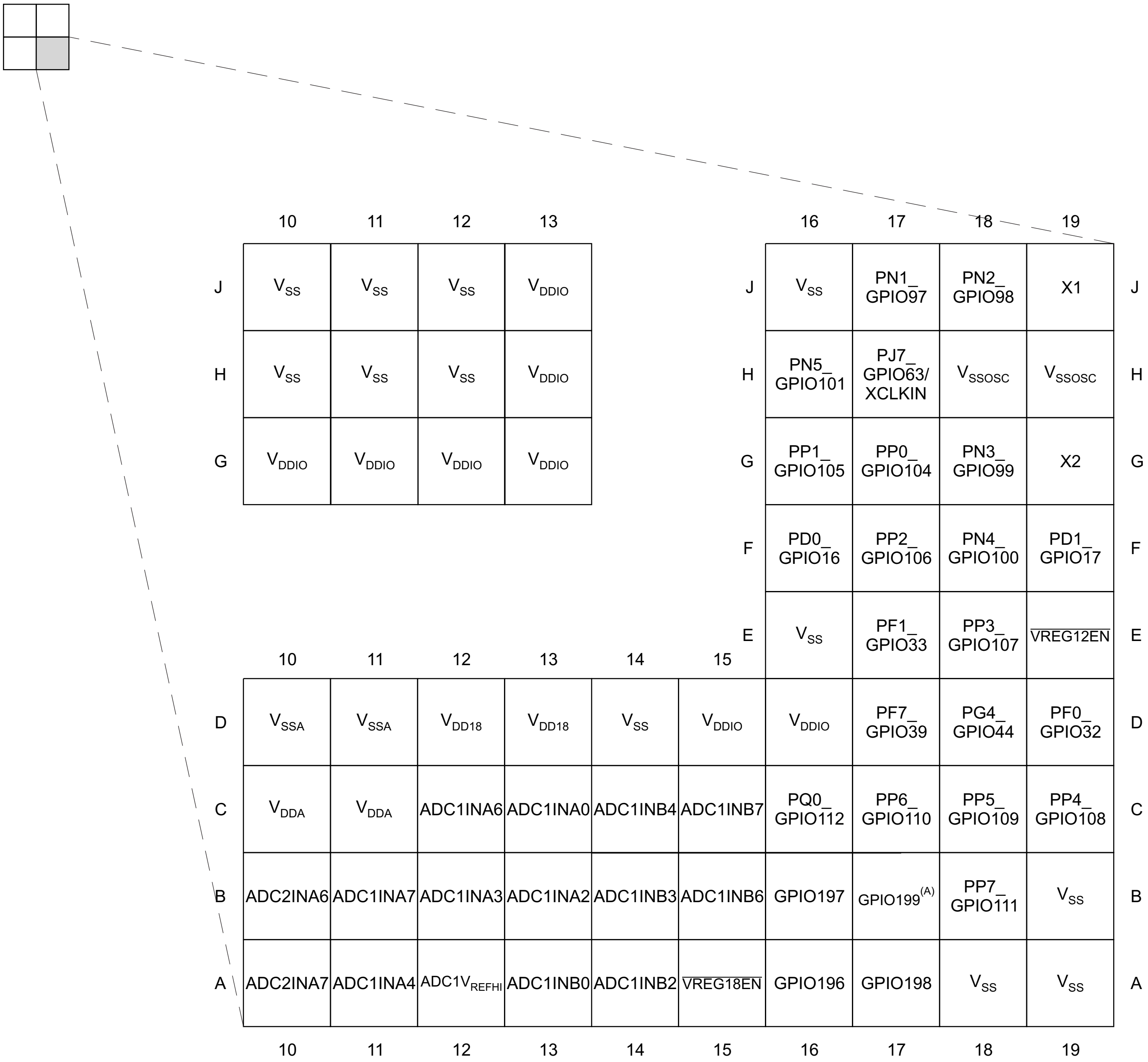
ADVANCE INFORMATION



	10	11	12	13	14	15	16	17	18	19
W	PE0_GPIO24	PG7_GPIO47	PF6_GPIO38	PG6_GPIO46	PG2_GPIO42	PG5_GPIO45	PJ0_GPIO56	PD7_GPIO23	V_{SS}	V_{SS}
V	PH0_GPIO48	PG0_GPIO40	PJ2_GPIO58	PJ1_GPIO57	PJ5_GPIO61	PJ4_GPIO60	PJ6_GPIO62	PD6_GPIO22	PL7_GPIO87	V_{SS}
U	PH4_GPIO52	PF5_GPIO37	PG1_GPIO41	V_{DDIO}	PF4_GPIO36	PJ3_GPIO59	PD4_GPIO20	PD5_GPIO21	PL4_GPIO84	PE5_GPIO29
T	V_{DD12}	V_{DD12}	V_{DD12}	V_{DDIO}	V_{SS}	V_{DDIO}	V_{DDIO}	PN7_GPIO103	PE4_GPIO28	TDO
R							V_{SS}	PH6_GPIO54	PN6_GPIO102	EMU1
P							PF2_GPIO34	PF3_GPIO35	PH7_GPIO55	EMU0
N	V_{DDIO}	V_{DDIO}	V_{DD12}	V_{DD12}			PK1_GPIO73	PG3_GPIO43	PR0_GPIO120	$\overline{TRST}$
M	V_{SS}	V_{SS}	V_{SS}	V_{DD12}			PK2_GPIO74	PR3_GPIO123	PR1_GPIO121	TMS
L	V_{SS}	V_{SS}	V_{SS}	V_{DDIO}			V_{SS}	PN0_GPIO96	PK3_GPIO75	TCK
K	V_{SS}	V_{SS}	V_{SS}	V_{DDIO}			V_{SS}	PK0_GPIO72	PR2_GPIO122	TDI

A. See Table 3-1, Terminal Functions, for the complete multiplexed signal names.

Figure 3-3. 289-Ball ZWT Pin Map [Quadrant B]

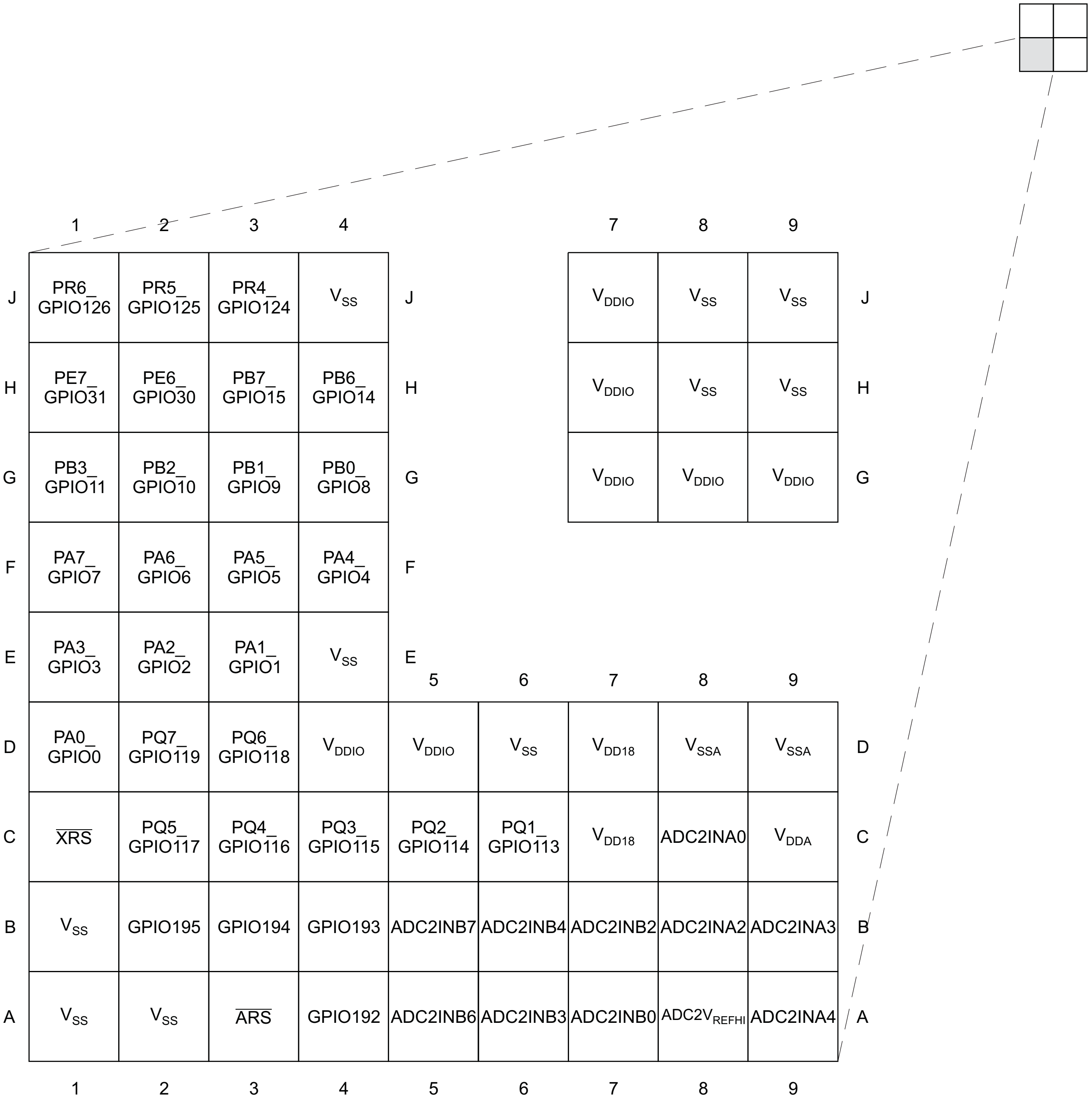


	10	11	12	13	14	15	16	17	18	19	
J	V_{SS}	V_{SS}	V_{SS}	V_{DDIO}			V_{SS}	PN1_ GPIO97	PN2_ GPIO98	X1	J
H	V_{SS}	V_{SS}	V_{SS}	V_{DDIO}			PN5_ GPIO101	PJ7_ GPIO63/ XCLKIN	V_{SSOSC}	V_{SSOSC}	H
G	V_{DDIO}	V_{DDIO}	V_{DDIO}	V_{DDIO}			PP1_ GPIO105	PP0_ GPIO104	PN3_ GPIO99	X2	G
F							PD0_ GPIO16	PP2_ GPIO106	PN4_ GPIO100	PD1_ GPIO17	F
E							V_{SS}	PF1_ GPIO33	PP3_ GPIO107	$\overline{VREG12EN}$	E
D	V_{SSA}	V_{SSA}	V_{DD18}	V_{DD18}	V_{SS}	V_{DDIO}	V_{DDIO}	PF7_ GPIO39	PG4_ GPIO44	PF0_ GPIO32	D
C	V_{DDA}	V_{DDA}	ADC1INA6	ADC1INA0	ADC1INB4	ADC1INB7	PQ0_ GPIO112	PP6_ GPIO110	PP5_ GPIO109	PP4_ GPIO108	C
B	ADC2INA6	ADC1INA7	ADC1INA3	ADC1INA2	ADC1INB3	ADC1INB6	GPIO197	GPIO199(A)	PP7_ GPIO111	V_{SS}	B
A	ADC2INA7	ADC1INA4	ADC1V_{REFHI}	ADC1INB0	ADC1INB2	$\overline{VREG18EN}$	GPIO196	GPIO198	V_{SS}	V_{SS}	A
	10	11	12	13	14	15	16	17	18	19	

A. All I/Os, except for GPIO199, are glitch-free during power up and power down. See Section 2.11.

B. See Table 3-1, Terminal Functions, for the complete multiplexed signal names.

Figure 3-4. 289-Ball ZWT Pin Map [Quadrant C]



	1	2	3	4	5	6	7	8	9
J	PR6_ GPIO126	PR5_ GPIO125	PR4_ GPIO124	V_{SS}			V_{DDIO}	V_{SS}	V_{SS}
H	PE7_ GPIO31	PE6_ GPIO30	PB7_ GPIO15	PB6_ GPIO14			V_{DDIO}	V_{SS}	V_{SS}
G	PB3_ GPIO11	PB2_ GPIO10	PB1_ GPIO9	PB0_ GPIO8			V_{DDIO}	V_{DDIO}	V_{DDIO}
F	PA7_ GPIO7	PA6_ GPIO6	PA5_ GPIO5	PA4_ GPIO4					
E	PA3_ GPIO3	PA2_ GPIO2	PA1_ GPIO1	V_{SS}					
D	PA0_ GPIO0	PQ7_ GPIO119	PQ6_ GPIO118	V_{DDIO}	V_{DDIO}	V_{SS}	V_{DD18}	V_{SSA}	V_{SSA}
C	$\overline{XRS}$	PQ5_ GPIO117	PQ4_ GPIO116	PQ3_ GPIO115	PQ2_ GPIO114	PQ1_ GPIO113	V_{DD18}	ADC2INA0	V_{DDA}
B	V_{SS}	GPIO195	GPIO194	GPIO193	ADC2INB7	ADC2INB4	ADC2INB2	ADC2INA2	ADC2INA3
A	V_{SS}	V_{SS}	$\overline{ARS}$	GPIO192	ADC2INB6	ADC2INB3	ADC2INB0	ADC2V_{REFHI}	ADC2INA4

A. See Table 3-1, Terminal Functions, for the complete multiplexed signal names.

Figure 3-5. 289-Ball ZWT Pin Map [Quadrant D]

3.2 Terminal Functions

Table 3-1 describes the signals.

Table 3-1. Terminal Functions[1]

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
ADC 1 Reference Inputs, Analog Comparator Inputs, DAC Inputs, AIO Group 1					
ADC1V$_{REFHI}$	A12	I	ADC1 External High Reference – used only when in ADC external reference mode.		
ADC1V$_{REFLO}$	see V$_{SSA}$	I	ADC1 External Low Reference – used only when in ADC external reference mode.		
ADC1INA0	C13	I	ADC1 Group A, Channel 0 input		
ADC1INA2 COMPA1 AIO2	B13	I I I/O	ADC1 Group A, Channel 2 input Comparator Input A1 Digital AIO2		4 mA
ADC1INA3	B12	I	ADC1 Group A, Channel 3 input		
ADC1INA4 COMPA2 AIO4	A11	I I I/O	ADC1 Group A, Channel 4 input Comparator Input A2 Digital AIO4		4 mA
ADC1INA6 COMPA3 AIO6	C12	I I I/O	ADC1 Group A, Channel 6 input Comparator Input A3 Digital AIO6		4 mA
ADC1INA7	B11	I	ADC1 Group A, Channel 7 input		
ADC1INB0	A13	I	ADC1 Group B, Channel 0 input		
ADC1INB2 COMPB1 AIO10	A14	I I I/O	ADC1 Group B, Channel 2 input Comparator Input B1 Digital AIO10		4 mA
ADC1INB3	B14	I	ADC1 Group B, Channel 3 input		
ADC1INB4 COMPB2 AIO12	C14	I I I/O	ADC1 Group B, Channel 4 input Comparator Input B2 Digital AIO12		4 mA
ADC1INB6 COMPB3 AIO14	B15	I I I/O	ADC1 Group B, Channel 6 input Comparator Input B3 Digital AIO14		4 mA
ADC1INB7	C15	I	ADC1 Group B, Channel 7 input		
ADC 2 Reference Inputs, Analog Comparator Inputs, DAC Inputs, AIO Group 2					
ADC2V$_{REFHI}$	A8	I	ADC2 External High Reference – used only when in ADC external reference mode.		
ADC2V$_{REFLO}$	see V$_{SSA}$	I	ADC2 External Low Reference – used only when in ADC external reference mode.		
ADC2INA0	C8	I	ADC2 Group A, Channel 0 input		
ADC2INA2 COMPA4 AIO18	B8	I I I/O	ADC2 Group A, Channel 2 input Comparator Input A4 Digital AIO18		4 mA
ADC2INA3	B9	I	ADC2 Group A, Channel 3 input		
ADC2INA4 COMPA5 AIO20	A9	I I I/O	ADC2 Group A, Channel 4 input Comparator Input A5 Digital AIO20		4 mA

Table 3-1. Terminal Functions[(1)] (continued)

TERMINAL		I/O/Z[(2)]	DESCRIPTION	PU or PD[(3)]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
ADC2INA6 COMPA6 AIO22	B10	I I I/O	ADC2 Group A, Channel 6 input Comparator Input A6 Digital AIO22		4 mA
ADC2INA7	A10	I	ADC2 Group A, Channel 7 input		
ADC2INB0	A7	I	ADC2 Group B, Channel 0 input		
ADC2INB2 COMPB4 AIO26	B7	I I I/O	ADC2 Group B, Channel 2 input Comparator Input B4 Digital AIO26		4 mA
ADC2INB3	A6	I	ADC2 Group B, Channel 3 input		
ADC2INB4 COMPB5 AIO28	B6	I I I/O	ADC2 Group B, Channel 4 input Comparator Input B5 Digital AIO28		4 mA
ADC2INB6 COMPB6 AIO30	A5	I I I/O	ADC2 Group B, Channel 6 input Comparator Input B6 Digital AIO30		4 mA
ADC2INB7	B5	I	ADC2 Group B, Channel 7 input		
ADC Modules Analog Power and Ground					
V_{DDA}	C9		3.3-V Analog Module Power Pin. Tie with a 2.2-µF capacitor (typical) close to the pin.		
V_{DDA}	C10		3.3-V Analog Module Power Pin. Tie with a 2.2-µF capacitor (typical) close to the pin.		
V_{DDA}	C11		3.3-V Analog Module Power Pin. Tie with a 2.2-µF capacitor (typical) close to the pin.		
V_{SSA}	D8		Analog ground for ADC1, ADC2, ADC1V_{REFLO}, ADC2V_{REFLO}, COMP1–6, and DAC1–3		
V_{SSA}	D9		Analog ground for ADC1, ADC2, ADC1V_{REFLO}, ADC2V_{REFLO}, COMP1–6, and DAC1–3		
V_{SSA}	D10		Analog ground for ADC1, ADC2, ADC1V_{REFLO}, ADC2V_{REFLO}, COMP1–6, and DAC1–3		
V_{SSA}	D11		Analog ground for ADC1, ADC2, ADC1V_{REFLO}, ADC2V_{REFLO}, COMP1–6, and DAC1–3		
Analog Comparator Results (Digital) and GPIO Group 2 (C28x Access Only)					
GPIO192	A4	I/O	General-purpose input/output 192	PU	4 mA
GPIO193 COMP1OUT	B4	I/O O	General-purpose input/output 193 Compare result from Analog Comparator 1	PU	4 mA
GPIO194 COMP6OUT	B3	I/O O	General-purpose input/output 194 Compare result from Analog Comparator 6	PU	4 mA
GPIO195 COMP2OUT	B2	I/O O	General-purpose input/output 195 Compare result from Analog Comparator 2	PU	4 mA
GPIO196 COMP3OUT	A16	I/O O	General-purpose input/output 196 Compare result from Analog Comparator 3	PU	8 mA
GPIO197 COMP4OUT	B16	I/O O	General-purpose input/output 197 Compare result from Analog Comparator 4	PU	4 mA
GPIO198	A17	I/O	General-purpose input/output 198	PU	4 mA
GPIO199[(4)] COMP5OUT	B17	I/O O	General-purpose input/output 199 Compare result from Analog Comparator 5	PU	8 mA

Table 3-1. Terminal Functions[(1)] (continued)

TERMINAL		I/O/Z[(2)]	DESCRIPTION	PU or PD[(3)]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
GPIO Group 1 and Peripheral Signals					
PA0_GPIO0	D1	I/O/Z	General-purpose input/output 0	PU	4 mA
M_U0RX		I	UART-0 receive data		
M_I2C1SCL		I/OD	I2C-1 clock open-drain bidirectional port		
M_U1RX		I	UART-1 receive data		
C_EPWM1A		O	Enhanced PWM-1 output A		
PA1_GPIO1	E3	I/O/Z	General-purpose input/output 1	PU	4 mA
M_U0TX		O	UART-0 transmit data		
M_I2C1SDA		I/OD	I2C-1 data open-drain bidirectional port		
M_U1TX		O	UART-1 data transmit		
M_SSI1FSS		I/O	SSI-1 frame		
C_EPWM1B		O	Enhanced PWM-1 output B		
C_ECAP6		I/O	Enhanced Capture-6 input/output		
PA2_GPIO2	E2	I/O/Z	General-purpose input/output 2	PU	4 mA
M_SSI0CLK		I/O	SSI-0 clock		
M_MIITXD2		O	EMAC MII transmit data bit 2		
M_U1CTS		I	UART-1 clear-to-send modem status		
C_EPWM2A		O	Enhanced PWM-2 output A		
PA3_GPIO3	E1	I/O/Z	General-purpose input/output 3	PU	4 mA
M_SSI0FSS		I/O	SSI-0 frame		
M_MIITXD1		O	EMAC MII transmit data bit 1		
M_U1DCD		I	UART-1 data carrier detect		
M_SSI1CLK		I/O	SSI-1 clock		
C_EPWM2B		O	Enhanced PWM-2 output B		
C_ECAP5		I/O	Enhanced Capture-5 input/output		
PA4_GPIO4	F4	I/O/Z	General-purpose input/output 4	PU	4 mA
M_SSI0RX		I	SSI-0 receive data		
M_MIITXD0		O	EMAC MII transmit data bit 0		
M_CAN0RX		I	CAN-0 receive data		
M_U1DSR		I	UART-1 data set ready		
C_EPWM3A		O	Enhanced PWM-3 output A		
PA5_GPIO5	F3	I/O/Z	General-purpose input/output 5	PU	4 mA
M_SSI0TX		O	SSI-0 transmit data		
M_MIIRXDV		I	EMAC MII receive data valid		
M_CAN0TX		O	CAN-0 transmit data		
M_U1RTS		O	UART-1 request-to-send		
C_EPWM3B		O	Enhanced PWM-3 output B		
C_MFSRA		I	McBSP-A receive frame sync		
C_ECAP1		I/O	Enhanced Capture-1 input/output		

ADVANCE INFORMATION

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PA6_GPIO6	F2	I/O/Z	General-purpose input/output 6	PU	4 mA
M_I2C1SCL		I/OD	I2C-1 clock open-drain bidirectional port		
M_CCP1		I/O	Capture/Compare/PWM-1 (General-purpose Timer)		
M_MIIRXCK		I	EMAC MII receive clock		
M_CAN0RX		I	CAN-0 receive data		
M_USB0EPEN		O	USB-0 external power enable (optionally used in host mode)		
M_U1CTS		I	UART-1 clear-to-send modem status		
M_U1DTR		O	UART-1 data terminal ready		
C_EPWM4A		O	Enhanced PWM-4 output A		
C_EPWMSYNCO		O	Enhanced PWM-4 external sync pulse		
PA7_GPIO7	F1	I/O/Z	General-purpose input/output 7	PU	4 mA
M_I2C1SDA		I/OD	I2C-1 data open-drain bidirectional port		
M_CCP4		I/O	Capture/Compare/PWM-4 (General-purpose Timer)		
M_MIIRXER		I	EMAC MII receive error		
M_CAN0TX		O	CAN-0 transmit data		
M_CCP3		I/O	Capture/Compare/PWM-3 (General-purpose Timer)		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
M_U1DCD		I	UART-1 data carrier detect		
M_MIIRXD1		I	EMAC MII receive data 1		
M_U1RI		I	UART-1 ring indicator modem status		
C_EPWM4B		O	Enhanced PWM-4 output B		
C_MCLKRA		I	McBSP-A receive clock		
C_ECAP2		I/O	Enhanced Capture-1 input/output		
PB0_GPIO8	G4	I/O/Z	General-purpose input/output 8	PU	4 mA
M_CCP0		I/O	Capture/Compare/PWM-0 (General-purpose Timer)		
M_U1RX		I	UART-1 data receive data		
M_SSI2TX		O	SSI-2 transmit data		
M_CAN1TX		O	CAN-1 transmit data		
M_U4TX		O	UART-4 transmit data		
C_EPWM5A		O	Enhanced PWM-5 output A		
$\overline{\text{C\_ADCSOCAO}}$		O	ADC start-of-conversion A		
PB1_GPIO9	G3	I/O/Z	General-purpose input/output 9	PU	4 mA
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_CCP1		I/O	Capture/Compare/PWM-1 (General-purpose Timer)		
M_U1TX		O	UART-1 transmit data		
M_SSI2RX		I	SSI-2 receive data		
C_EPWM5B		O	Enhanced PWM-5 output B		
C_ECAP3		I/O	Enhanced Capture-3 input/output		

Table 3-1. Terminal Functions[(1)] (continued)

TERMINAL		I/O/Z[(2)]	DESCRIPTION	PU or PD[(3)]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PB2_GPIO10	G2	I/O/Z	General-purpose input/output 10	PU	4 mA
M_I2C0SCL		I/OD	I2C-0 clock open-drain bidirectional port		
M_CCP3		I/O	Capture/Compare/PWM-3 (General-purpose Timer)		
M_CCP0		I/O	Capture/Compare/PWM-0 (General-purpose Timer)		
M_USB0EPEN		O	USB-0 external power enable (optionally used in the host mode)		
M_SSI2CLK		I/O	SSI-2 clock		
M_CAN1RX		I	CAN-1 receive data		
M_U4RX		I	UART-4 receive data		
C_EPWM6A		O	Enhanced PWM-6 output A		
$\overline{\text{C\_ADCSOCBO}}$		O	ADC start-of-conversion B		
PB3_GPIO11	G1	I/O/Z	General-purpose input/output 11	PU	4 mA
M_I2C0SDA		I/OD	I2C-0 data open-drain bidirectional port		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
M_SSI2FSS		I/O	SSI-2 frame		
M_U1RX		I	UART-1 receive data		
C_EPWM6B		O	Enhanced PWM-6 output B		
C_ECAP4		I/O	Enhanced Capture-4 input/output		
PB4_GPIO12	N4	I/O/Z	General-purpose input/output 12	PU	4 mA
M_U2RX		I	UART-2 receive data		
M_CAN0RX		I	CAN-0 receive data		
M_U1RX		I	UART-1 receive data		
M_EPI0S23		I/O	EPI-0 signal 23		
M_CAN1TX		O	CAN-1 transmit data		
M_SSI1TX		O	SSI-1 transmit data		
C_EPWM7A		O	Enhanced PWM-7 output A		
PB5_GPIO13	M4	I/O/Z	General-purpose input/output 13	PU	4 mA
M_CCP5		I/O	Capture/Compare/PWM-5 (General-purpose Timer)		
M_CCP6		I/O	Capture/Compare/PWM-6 (General-purpose Timer)		
M_CCP0		I/O	Capture/Compare/PWM-0 (General-purpose Timer)		
M_CAN0TX		O	CAN-0 transmit data		
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_U1TX		O	UART-1 transmit data		
M_EPI0S22		I/O	EPI-0 signal 22		
M_CAN1RX		I	CAN-1 receive data		
M_SSI1RX		I	SSI-1 receive data		
C_EPWM7B		O	Enhanced PWM-7 output B		

ADVANCE INFORMATION

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PB6_GPIO14	H4	I/O/Z	General-purpose input/output 14	PU	4 mA
M_CCP1		I/O	Capture/Compare/PWM-1 (General-purpose Timer)		
M_CCP7		I/O	Capture/Compare/PWM-7 (General-purpose Timer)		
M_CCP5		I/O	Capture/Compare/PWM-5 (General-purpose Timer)		
M_EPI0S37[5]		I/O	EPI-0 signal 37		
M_MIICRS		I	EMAC MII carrier sense		
M_I2C0SDA		I/OD	I2C-0 data open-drain bidirectional port		
M_U1TX		O	UART-1 transmit data		
M_SSI1CLK		I/O	SSI-1 clock		
C_EPWM8A		O	Enhanced PWM-8 output A		
PB7_GPIO15	H3	I/O/Z	General-purpose input/output 15	PU	4 mA
M_EXTNMI		I	Cortex™-M3 external non-maskable interrupt		
M_MIIRXD1		I	EMAC MII receive data 1		
M_EPI0S36[5]		I/O	EPI-0 signal 36		
M_I2C0SCL		I/OD	I2C-0 clock open-drain bidirectional port		
M_U1RX		I	UART-1 receive data		
M_SSI1FSS		I/O	SSI-1 frame		
C_EPWM8B		O	Enhanced PWM-8 output B		
PD0_GPIO16	F16	I/O/Z	General-purpose input/output 16	PU	4 mA
M_CAN0RX		I	CAN-0 receive data		
M_U2RX		I	UART-2 receive data		
M_U1RX		I	UART-1 receive data		
M_CCP6		I/O	Capture/Compare/PWM-6 (General-purpose Timer)		
M_MIIRXDV		I	EMAC MII receive data valid		
M_U1CTS		I	UART-1 clear-to-send modem status		
M_MIIRXD2		I	EMAC MII receive data 2		
M_SSI0TX		O	SSI-0 transmit data		
M_CAN1TX		O	CAN-1 transmit data		
M_USB0EPEN		O	USB-0 external power enable (optionally used in the host mode)		
C_SPISIMOA		I/O	SPI-A slave in, master out		

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PD1_GPIO17	F19	I/O/Z	General-purpose input/output 17	PU	4 mA
M_CAN0TX		O	CAN-0 transmit data		
M_U2TX		O	UART-2 transmit data		
M_U1TX		O	UART-1 transmit data		
M_CCP7		I/O	Capture/Compare/PWM-7 (General-purpose Timer)		
M_MIITXER		O	EMAC MII transmit error		
M_U1DCD		I	UART-1 data carrier detect		
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_MIICOL		I	EMAC MII collision detect		
M_SSI0RX		I	SSI-0 receive data		
M_CAN1RX		I	CAN-1 receive data		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
C_SPISOMIA		I/O	SPI-A master in, slave out		
PD2_GPIO18	W5	I/O/Z	General-purpose input/output 18	PU	4 mA
M_U1RX		I	UART-1 receive data		
M_CCP6		I/O	Capture/Compare/PWM-6 (General-purpose Timer)		
M_CCP5		I/O	Capture/Compare/PWM-5 (General-purpose Timer)		
M_EPI0S20		I/O	EPI-0 signal 20		
M_SSI0CLK		I/O	SSI-0 clock		
M_U1TX		O	UART-1 transmit data		
M_CAN0RX		I	CAN-0 receive data		
C_SPICLKA		I/O	SPI-A clock		
PD3_GPIO19	W6	I/O/Z	General-purpose input/output 19	PU	4 mA
M_U1TX		O	UART-1 transmit data		
M_CCP7		I/O	Capture/Compare/PWM-7 (General-purpose Timer)		
M_CCP0		I/O	Capture/Compare/PWM-0 (General-purpose Timer)		
M_EPI0S21		I/O	EPI-0 signal 21		
M_SSI0FSS		I/O	SSI-0 frame		
M_U1RX		I	UART-1 receive data		
M_CAN0TX		O	CAN-0 transmit data		
C_SPISTEA		I/O	SPI-A slave transmit enable		

ADVANCE INFORMATION

Table 3-1. Terminal Functions[(1)] (continued)

TERMINAL		I/O/Z[(2)]	DESCRIPTION	PU or PD[(3)]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PD4_GPIO20	U16	I/O/Z	General-purpose input/output 20	PU	4 mA
M_CCP0		I/O	Capture/Compare/PWM-0 (General-purpose Timer)		
M_CCP3		I/O	Capture/Compare/PWM-3 (General-purpose Timer)		
M_MIITXD3		O	EMAC MII transmit data 3		
M_U1RI		I	UART-1 ring indicator modem status		
M_EPI0S19		I/O	EPI-0 signal 19		
M_U3TX		O	UART-3 transmit data		
M_CAN1TX		O	CAN-1 transmit data		
C_EQEP1A		I	Enhanced QEP-1 input A		
C_MDXA		O	McBSP-A transmit data		
PD5_GPIO21	U17	I/O/Z	General-purpose input/output 21	PU	6 mA
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_CCP4		I/O	Capture/Compare/PWM-4 (General-purpose Timer)		
M_MIITXD2		O	EMAC MII transmit data 2		
M_U2RX		I	UART-2 receive data		
M_EPI0S28		I/O	EPI-0 signal 28		
M_U3RX		I	UART-3 receive data		
M_CAN1RX		I	CAN-1 receive data		
C_EQEP1B		I	Enhanced QEP-1 input B		
C_MDRA		I	McBSP-A receive data		
PD6_GPIO22	V17	I/O/Z	General-purpose input/output 22	PU	6 mA
M_MIITXD1		O	EMAC MII transmit data 1		
M_U2TX		O	UART-2 transmit data		
M_EPI0S29		I/O	EPI-0 signal 29		
M_I2C1SDA		I/OD	I2C-0 data open-drain bidirectional port		
M_U1TX		O	UART-1 transmit data		
C_EQEP1S		I/O	Enhanced QEP-1 strobe		
C_MCLKXA		O	McBSP-A transmit clock		
PD7_GPIO23	W17	I/O/Z	General-purpose input/output 23	PU	6 mA
M_CCP1		I/O	Capture/Compare/PWM-1 (General-purpose Timer)		
M_MIITXD0		O	EMAC MII transmit data 0		
M_U1DTR		O	UART-1 data terminal ready		
M_EPI0S30		I/O	EPI-0 signal 30		
M_I2C1SCL		I/OD	I2C-1 clock open-drain bidirectional port		
M_U1RX		I	UART-1 receive data		
C_EQEP1I		I/O	Enhanced QEP-1 index		
C_MFSXA		O	McBSP-A transmit frame sync		

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PE0_GPIO24	W10	I/O/Z	General-purpose input/output 24	PU	4 mA
M_SSI1CLK		I/O	SSI-1 clock		
M_CCP3		I/O	Capture/Compare/PWM-3 (General-purpose Timer)		
M_EPI0S8		I/O	EPI-0 signal 8		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
M_SSI3TX		O	SSI-3 transmit data		
M_CAN0RX		I	CAN-1 receive data		
M_SSI1TX		O	SSI-1 transmit data		
C_ECAP1		I/O	Enhanced Capture-1 input/output		
C_EQEP2A		I	Enhanced QEP-2 input A		
PE1_GPIO25	W9	I/O/Z	General-purpose input/output 25	PU	4 mA
M_SSI1FSS		I/O	SSI-1 frame		
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_CCP6		I/O	Capture/Compare/PWM-6 (General-purpose Timer)		
M_EPI0S9		I/O	EPI-0 signal 9		
M_SSI3RX		I	SSI-3 receive data		
M_CAN0TX		O	CAN-1 transmit data		
M_SSI1RX		O	SSI-1 receive data		
C_ECAP2		I/O	Enhanced Capture-2 input/output		
C_EQEP2B		I	Enhanced QEP-2 input B		
PE2_GPIO26	U6	I/O/Z	General-purpose input/output 26	PU	4 mA
M_CCP4		I/O	Capture/Compare/PWM-4 (General-purpose Timer)		
M_SSI1RX		I	SSI-1 receive data		
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_EPI0S24		I/O	EPI-0 signal 24		
M_SSI3CLK		I/O	SSI-3 clock		
M_U2RX		I	UART-2 receive data		
M_SSI1CLK		I/O	SSI-1 clock		
C_ECAP3		I/O	Enhanced Capture-3 input/output		
C_EQEP2I		I/O	Enhanced QEP-2 index		
PE3_GPIO27	V6	I/O/Z	General-purpose input/output 27	PU	4 mA
M_CCP1		I/O	Capture/Compare/PWM-1 (General-purpose Timer)		
M_SSI1TX		O	SSI-1 transmit data		
M_CCP7		I/O	Capture/Compare/PWM-7 (General-purpose Timer)		
M_EPI0S25		I/O	EPI-0 signal 25		
M_SSI3FSS		I/O	SSI-3 frame		
M_U2TX		O	UART-2 transmit data		
M_SSI1FSS		I/O	SSI-1 frame		
C_ECAP4		I/O	Enhanced Capture-4 input/output		
C_EQEP2S		I/O	Enhanced QEP-2 strobe		

Table 3-1. Terminal Functions(1) (continued)

TERMINAL		I/O/Z(2)	DESCRIPTION	PU or PD(3)	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PE4_GPIO28	T18	I/O/Z	General-purpose input/output 28	PU	4 mA
M_CCP3		I/O	Capture/Compare/PWM-3 (General-purpose Timer)		
M_U2TX		O	UART-2 transmit data		
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_MIIRXD0		I	EMAC MII receive data 0		
M_EPI0S34(5)		I/O	EPI-0 signal 34		
M_U0RX		I	UART-0 receive data		
M_EPI0S38(5)		I/O	EPI-0 signal 38		
M_USB0EPEN		O	USB-0 external power enable (optionally used in the host mode)		
C_SCIRXDA		I	SCI-A receive data		
PE5_GPIO29	U19	I/O/Z	General-purpose input/output 29	PU	4 mA
M_CCP5		I/O	Capture/Compare/PWM-5 (General-purpose Timer)		
M_EPI0S35(5)		I/O	EPI-0 signal 35		
M_MIITXER		O	EMAC MII transmit error		
M_U0TX		O	UART-0 transmit data		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
C_SCITXDA		O	SCI-A transmit data		
PE6_GPIO30	H2	I/O/Z	General-purpose input/output 30	PU	4 mA
M_U1CTS		I	UART-1 clear-to-send modem status		
M_MDIOD		I/O	EMAC management data input/output		
M_CAN0RX		I	CAN-0 receive data		
C_EPWM9A		O	Enhanced PWM-9 output A		
PE7_GPIO31	H1	I/O/Z	General-purpose input/output 31	PU	4 mA
M_U1DCD		I	UART-1 data carrier detect		
M_MIIRXD3		I	EMAC MII receive data 3		
M_CAN0TX		O	CAN-0 transmit data		
C_EPWM9B		O	Enhanced PWM-9 output B		
PF0_GPIO32	D19	I/O/Z	General-purpose input/output 32	PU	4 mA
M_CAN1RX		I	CAN-1 receive data		
M_MIIRXCK		I	EMAC MII receive clock		
M_U1DSR		I	UART-1 data set ready		
M_I2C0SDA		I/OD	I2C-0 data open-drain bidirectional port		
M_TRACED2		O	Trace data 2		
C_I2CASDA		I/OD	I2C-A data open-drain bidirectional port		
C_SCIRXDA		I	SCI-A receive data		
$\overline{\text{C\_ADCSOCAO}}$		O	ADC start-of-conversion A(6)		

Table 3-1. Terminal Functions(1) (continued)

TERMINAL		I/O/Z(2)	DESCRIPTION	PU or PD(3)	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PF1_GPIO33	E17	I/O/Z	General-purpose input/output 33	PU	4 mA
M_CAN1TX		O	CAN-1 transmit data		
M_MIIRXER		I	EMAC MII receive error		
M_U1RTS		O	UART-1 request-to-send		
M_CCP3		I/O	Capture/Compare/PWM-3 (General-purpose Timer)		
M_I2C0SCL		I/OD	I2C-0 clock open-drain bidirectional port		
M_TRACED3		O	Trace data 3		
C_I2CASCL		I/OD	I2C-A clock open-drain bidirectional port		
C_EPWMSYNCO		O	Enhanced PWM sync out		
C_ADCSOCBO		O	ADC start-of-conversion B(6)		
PF2_GPIO34	P16	I/O/Z	General-purpose input/output 34	PU	4 mA
M_MIIPHYINTR		I	EMAC PHY MII interrupt		
M_EPI0S32(5)		I/O	EPI-0 signal 32		
M_SSI1CLK		I/O	SSI-1 clock		
M_TRACECLK		O	Trace clock		
M_XCLKOUT		O	External output clock		
C_ECAP1		I/O	Enhanced Capture-1 input/output		
C_SCIRXDA		I	SCI-A receive data		
C_XCLKOUT		O	External output clock		
BOOT_3		I	Boot pin 3		
PF3_GPIO35	P17	I/O/Z	General-purpose input/output 35	PU	4 mA
M_MDIOCK		I	EMAC management data clock		
M_EPI0S33(5)		I/O	EPI-0 signal 33		
M_SSI1FSS		I/O	SSI-1 frame		
M_U0TX		O	UART-0 transmit data		
M_TRACED0		O	Trace data 0		
C_SCITXDA		O	SCI-A transmit data		
BOOT_2		I	Boot pin 2		
PF4_GPIO36	U14	I/O/Z	General-purpose input/output 36	PU	4 mA
M_CCP0		I/O	Capture/Compare/PWM-0 (General-purpose Timer)		
M_MDIOD		I/O	EMAC management data input/output		
M_EPI0S12		I/O	EPI-0 signal 12		
M_SSI1RX		I	SSI-1 receive data		
M_U0RX		I	UART-0 receive data		
C_SCIRXDA		I	SCI-A receive data		
PF5_GPIO37	U11	I/O/Z	General-purpose input/output 37	PU	4 mA
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_MIIRXD3		I	EMAC MII receive data 3		
M_EPI0S15		I/O	EPI-0 signal 15		
M_SSI1TX		O	SSI-1 transmit data		
M_MIITXEN		O	EMAC MII transmit enable		
C_ECAP2		I/O	Enhanced Capture-2 input/output		

Table 3-1. Terminal Functions(1) (continued)

TERMINAL		I/O/Z(2)	DESCRIPTION	PU or PD(3)	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PF6_GPIO38	W12	I/O/Z	General-purpose input/output 38. **NOTE:** For this pin, only the USB0VBUS function is available on silicon revision 0 devices (GPIO and the four other functions listed are not available).	PU	4 mA
M_USB0VBUS		Analog	USB0 VBUS power (5-V tolerant)		
M_CCP1		I/O	Capture/Compare/PWM-1 (General-purpose Timer)		
M_MIIRXD2		I	EMAC MII receive data 2		
M_EPI0S38(5)		I/O	EPI-0 signal 38		
M_U1RTS		O	UART-1 request-to-send		
PF7_GPIO39	D17	I/O/Z	General-purpose input/output 39	PU	4 mA
M_CAN1TX		O	CAN-1 transmit data		
PG0_GPIO40	V11	I/O/Z	General-purpose input/output 40	PU	4 mA
M_U2RX		I	UART-2 receive data		
M_I2C1SCL		I/OD	I2C-1 clock open-drain bidirectional port		
M_USB0EPEN		O	USB-0 external power enable (optionally used in the host mode)		
M_EPI0S13		I/O	EPI-0 signal 13		
M_MIIRXD2		I	EMAC MII receive data 2		
M_U4RX		I	UART-4 receive data		
M_MIITXCK		I	EMAC MII transmit clock		
PG1_GPIO41	U12	I/O/Z	General-purpose input/output 41	PU	4 mA
M_U2TX		O	UART-2 transmit data		
M_I2C1SDA		I/OD	I2C-1 data open-drain bidirectional port		
M_EPI0S14		I/O	EPI-0 signal 14		
M_MIIRXD1		I	EMAC MII receive data 1		
M_U4TX		O	UART-4 transmit data		
M_MIITXER		O	EMAC MII transmit error		
PG2_GPIO42	W14	I/O/Z	General-purpose input/output 42	PU	4 mA
M_USB0DM		Analog	USB0 data minus		
M_MIICOL		I	EMAC MII collision detect		
M_EPI0S39(5)		I/O	EPI-0 signal 39		
PG3_GPIO43	N17	I/O/Z	General-purpose input/output 43	PU	4 mA
M_MIICRS		I	EMAC MII carrier sense		
M_MIIRXDV		I	EMAC MII receive data valid		
M_TRACED1		O	Trace data 1		
BOOT_0		I	Boot pin 0		
PG4_GPIO44	D18	I/O/Z	General-purpose input/output 44	PU	4 mA
M_CAN1RX		I	CAN-1 receive data		
PG5_GPIO45	W15	I/O/Z	General-purpose input/output 45	PU	4 mA
M_USB0DP		Analog	USB0 data plus		
M_CCP5		I/O	Capture/Compare/PWM-5 (General-purpose Timer)		
M_MIITXEN		O	EMAC MII transmit enable		
M_EPI0S40(5)		I/O	EPI-0 signal 40		
M_U1DTR		O	UART-1 data terminal ready		

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PG6_GPIO46	W13	I/O/Z	General-purpose input/output 46. **NOTE:** For this pin, only the USB0ID function is available on silicon revision 0 devices (GPIO and the three other functions listed are not available).	PU	4 mA
M_USB0ID		Analog	USB0 ID (5-V tolerant)		
M_MIITCK		I	EMAC MII transmit clock		
M_EPI0S41[5]		I/O	EPI-0 signal 41		
M_U1RI		I	UART-1 receive data		
PG7_GPIO47	W11	I/O/Z	General-purpose input/output 47	PU	6 mA
M_MIITXER		O	EMAC MII transmit error		
M_CCP5		I/O	Capture/Compare/PWM-5 (General-purpose Timer)		
M_EPI0S31		I/O	EPI-0 signal 31		
M_MIICRS		I	EMAC MII carrier sense		
BOOT_1		I	Boot pin 1		
PH0_GPIO48	V10	I/O/Z	General-purpose input/output 48	PU	4 mA
M_CCP6		I/O	Capture/Compare/PWM-6 (General-purpose Timer)		
$\overline{\text{M\_MIIPHYRST}}$		O	EMAC PHY MII reset		
M_EPI0S6		I/O	EPI-0 signal 6		
M_SSI3TX		O	SSI-3 transmit data		
M_MIITXD3		O	EMAC MII transmit data 3		
C_ECAP5		I/O	Enhanced Capture-5 input/output		
PH1_GPIO49	U8	I/O/Z	General-purpose input/output 49	PU	4 mA
M_CCP7		I/O	Capture/Compare/PWM-7 (General-purpose Timer)		
M_EPI0S7		I/O	EPI-0 signal 7		
M_MIIRXD0		I	EMAC MII receive data 0		
M_SSI3RX		I	SSI-3 receive data		
M_MIITXD2		O	EMAC MII transmit data 2		
C_ECAP6		I/O	Enhanced Capture-6 input/output		
PH2_GPIO50	V7	I/O/Z	General-purpose input/output 50	PU	4 mA
M_EPI0S1		I/O	EPI-0 signal 1		
M_MIITXD3		O	EMAC MII transmit data 3		
M_SSI3CLK		I/O	SSI-3 clock		
M_MIITXD1		O	EMAC MII transmit data 1		
C_EQEP1A		I	Enhanced QEP-1 input A		
PH3_GPIO51	U7	I/O/Z	General-purpose input/output 51	PU	4 mA
M_USB0EPEN		O	USB-0 external power enable (optionally used in the host mode)		
M_EPI0S0		I/O	EPI-0 signal 0		
M_MIITXD2		O	EMAC MII transmit data 2		
M_SSI3FSS		I/O	SSI-3 frame		
M_MIITXD0		O	EMAC MII transmit data 0		
C_EQEP1B		I	Enhanced QEP-1 input B		

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PH4_GPIO52	U10	I/O/Z	General-purpose input/output 52	PU	4 mA
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
M_EPI0S10		I/O	EPI-0 signal 10		
M_MIITXD1		O	EMAC MII transmit data 1		
M_SSI1CLK		I/O	SSI-1 clock		
M_U3TX		O	UART-3 transmit data		
M_MIICOL		I	EMAC MII collision detect		
C_EQEP1S		I/O	Enhanced QEP-1 strobe		
PH5_GPIO53	U9	I/O/Z	General-purpose input/output 53	PU	4 mA
M_EPI0S11		I/O	EPI-0 signal 11		
M_MIITXD0		O	EMAC MII transmit data 0		
M_SSI1FSS		I/O	SSI-1 frame		
M_U3RX		I	UART-3 receive data		
M_MIIPHYRST		O	EMAC PHY MII reset		
C_EQEP1I		I/O	Enhanced QEP-1 index		
PH6_GPIO54	R17	I/O/Z	General-purpose input/output 54	PU	4 mA
M_EPI0S26		I/O	EPI-0 signal 26		
M_MIIRXDV		I	EMAC MII receive data valid		
M_SSI1RX		I	SSI-1 receive data		
M_MIITXEN		O	EMAC MII transmit enable		
M_SSI0TX		O	SSI-0 transmit data		
M_MIIPHYINTR		I	EMAC PHY MII interrupt		
C_SPISIMOA		I/O	SPI-A slave in, master out		
C_EQEP3A		I	Enhanced QEP-1 input A		
PH7_GPIO55	P18	I/O/Z	General-purpose input/output 55	PU	4 mA
M_MIIRXCK		I	EMAC MII receive clock		
M_EPI0S27		I/O	EPI-0 signal 27		
M_SSI1TX		O	SSI-1 transmit data		
M_MIITXCK		I	EMAC MII transmit clock		
M_SSI0RX		I	SSI-0 receive data		
M_MDIOCK		O	EMAC management data clock		
C_SPISOMIA		I/O	SPI-A master in, slave out		
C_EQEP3B		I	Enhanced QEP-3 input B		
PJ0_GPIO56	W16	I/O/Z	General-purpose input/output 56	PU	4 mA
M_MIIRXER		I	EMAC MII receive error		
M_EPI016		I/O	EPI-0 signal 16		
M_I2C1SCL		I/OD	I2C-1 clock open-drain bidirectional port		
M_SSI0CLK		I/O	SSI-0 clock		
M_MDIOD		I/O	EMAC management data input/output		
C_SPICLKA		I/O	SPI-A clock		
C_EQEP3S		I/O	Enhanced QEP-3 strobe		

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PJ1_GPIO57	V13	I/O/Z	General-purpose input/output 57	PU	4 mA
M_EPI0S17		I/O	EPI-0 signal 17		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
M_I2C1SDA		I/OD	I2C-1 data open-drain bidirectional port		
M_MIIRXDV		I	EMAC MII receive data valid		
M_SSI0FSS		I/O	SSI-0 frame		
M_MIIRXD3		I	EMAC MII receive data 3		
C_SPISTEA		I/O	SPI-A slave transmit enable		
C_EQEP3I		I/O	Enhanced QEP-3 index		
PJ2_GPIO58	V12	I/O/Z	General-purpose input/output 58	PU	4 mA
M_EPI0S18		I/O	EPI-0 signal 18		
M_CCP0		I/O	Capture/Compare/PWM-0 (General-purpose Timer)		
M_MIIRXCK		I	EMAC MII receive clock		
M_SSI0CLK		I/O	SSI-0 clock		
M_U0TX		O	UART-0 transmit data		
M_MIIRXD2		I	EMAC MII receive data 2		
C_MCLKRA		I	McBSP-A receive clock		
C_EPWM7A		O	Enhanced PWM-7 output A		
PJ3_GPIO59	U15	I/O/Z	General-purpose input/output 59	PU	4 mA
M_EPI0S19		I/O	EPI-0 signal 19		
M_U1CTS		I	UART-1 clear-to-send		
M_CCP6		I/O	Capture/Compare/PWM-6 (General-purpose Timer)		
M_MDIOCK		O	EMAC management data clock		
M_SSI0FSS		I/O	SSI-0 frame		
M_U0RX		I	UART-0 receive data		
M_MIIRXD1		I	EMAC MII receive data 1		
C_MFSRA		I	McBSP-A receive frame sync		
C_EPWM7B		O	Enhanced PWM-7 output B		
PJ4_GPIO60	V15	I/O/Z	General-purpose input/output 60	PU	6 mA
M_EPI0S28		I/O	EPI-0 signal 28		
M_U1DCD		I	UART-1 data carrier detect		
M_CCP4		I/O	Capture/Compare/PWM-4 (General-purpose Timer)		
M_MIICOL		I	EMAC MII collision detect		
M_SSI1CLK		I/O	SSI-1 clock		
M_MIIRXD0		I	EMAC MII receive data 0		
C_EPWM8A		O	Enhanced PWM-8 output A		

ADVANCE INFORMATION

Table 3-1. Terminal Functions(1) (continued)

TERMINAL		I/O/Z(2)	DESCRIPTION	PU or PD(3)	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PJ5_GPIO61	V14	I/O/Z	General-purpose input/output 61	PU	6 mA
M_EPI0S29		I/O	EPI-0 signal 29		
M_U1DSR		I	UART-1 data set ready		
M_CCP2		I/O	Capture/Compare/PWM-2 (General-purpose Timer)		
M_MIICRS		I	EMAC MII carrier sense		
M_SSI1FSS		I/O	SSI-1 frame		
M_MIIRXDV		I	EMAC MII receive data valid		
C_EPWM8B		O	Enhanced PWM-8 output B		
PJ6_GPIO62	V16	I/O/Z	General-purpose input/output 62	PU	6 mA
M_EPI0S30		I/O	EPI-0 signal 30		
M_U1RTS		O	UART-1 request-to-send		
M_CCP1		I/O	Capture/Compare/PWM-1 (General-purpose Timer)		
$\overline{\text{M\_MIIPHYINTR}}$		I	EMAC PHY MII interrupt		
M_U2RX		I	UART-2 receive data		
M_MIIRXER		I	EMAC MII receive error		
C_EPWM9A		O	Enhanced PWM-9 output A		
PJ7_GPIO63	H17	I/O/Z	General-purpose input/output 63	PU	4 mA
M_U1DTR		O	UART-1 data terminal ready		
M_CCP0		I/O	Capture/Compare/PWM-0 (General-purpose Timer)		
$\overline{\text{M\_MIIPHYRST}}$		O	EMAC PHY MII reset		
M_U2TX		O	UART-2 transmit data		
M_MIIRXCK		I	EMAC MII receive clock		
M_XCLKIN		I	External oscillator input for USB PLL and CAN (always available, see Figure 2-15)		
C_EPWM9B		O	Enhanced PWM-9 output B		
PC0_GPIO64	V4	I/O/Z	General-purpose input/output 64	PU	4 mA
M_EPI0S32(5)		I/O	EPI-0 signal 32		
M_MIIRXD2		I	EMAC MII receive data 2		
C_EQEP1A		I	Enhanced QEP-1 input A		
C_EQEP2I		I/O	Enhanced QEP-2 index		
PC1_GPIO65	W4	I/O/Z	General-purpose input/output 65	PU	4 mA
M_EPI0S33(5)		I/O	EPI-0 signal 33		
M_MIICOL		I	EMAC MII collision detect		
C_EQEP1B		I	Enhanced QEP-1 input B		
C_EQEP2S		I/O	Enhanced QEP-2 strobe		
PC2_GPIO66	U5	I/O/Z	General-purpose input/output 66	PU	4 mA
M_EPI0S37(5)		I/O	EPI-0 signal 37		
M_MIITXEN		O	EMAC MII transmit enable		
C_EQEP1S		I/O	Enhanced QEP-1 strobe		
C_EQEP2A		I	Enhanced QEP-2 input A		

Table 3-1. Terminal Functions[(1)] (continued)

TERMINAL		I/O/Z[(2)]	DESCRIPTION	PU or PD[(3)]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PC3_GPIO67	V5	I/O/Z	General-purpose input/output 67	PU	4 mA
M_EPI0S36[(5)]		I/O	EPI-0 signal 36		
M_MIITXCK		I	EMAC MII transmit clock		
C_EQEP1I		I/O	Enhanced QEP-1 index		
C_EQEP2B		I	Enhanced QEP-2 input B		
PC4_GPIO68	W8	I/O/Z	General-purpose input/output 68	PU	4 mA
M_CCP5		I	Capture/Compare/PWM-5 (General-purpose Timer)		
M_MIITXD3		O	EMAC MII transmit data 3		
M_CCP2		I	Capture/Compare/PWM-2 (General-purpose Timer)		
M_CCP4		I	Capture/Compare/PWM-4 (General-purpose Timer)		
M_EPI0S2		I/O	EPI-0 signal 2		
M_CCP1		I	Capture/Compare/PWM-1 (General-purpose Timer)		
PC5_GPIO69	W7	I/O/Z	General-purpose input/output 69	PU	4 mA
M_CCP1		I	Capture/Compare/PWM-1 (General-purpose Timer)		
M_CCP3		I	Capture/Compare/PWM-3 (General-purpose Timer)		
M_USB0EPEN		O	USB-0 external power enable (optionally used in the host mode)		
M_EPI0S3		I/O	EPI-0 signal 3		
PC6_GPIO70	V8	I/O/Z	General-purpose input/output 70	PU	4 mA
M_CCP3		I	Capture/Compare/PWM-3 (General-purpose Timer)		
M_U1RX		I	UART-1 receive data		
M_CCP0		I	Capture/Compare/PWM-0 (General-purpose Timer)		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
M_EPI0S4		I/O	EPI-0 signal 4		
PC7_GPIO71	V9	I/O/Z	General-purpose input/output 71	PU	4 mA
M_CCP4		I	Capture/Compare/PWM-4 (General-purpose Timer)		
M_CCP0		I	Capture/Compare/PWM-0 (General-purpose Timer)		
M_U1TX		O	UART-1 transmit data		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		
M_EPI0S5		I/O	EPI-0 signal 5		
PK0_GPIO72	K17	I/O/Z	General-purpose input/output 72	PU	4 mA
M_SSI0TX		O	SSI-0 transmit data		
C_SPISIMOA		I/O	SPI-A slave in, master out		
PK1_GPIO73	N16	I/O/Z	General-purpose input/output 73	PU	4 mA
M_SSI0RX		I/O	SSI-0 receive data		
C_SPISOMIA		I/O	SPI-A master in, slave out		

ADVANCE INFORMATION

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PK2_GPIO74	M16	I/O/Z	General-purpose input/output 74	PU	4 mA
M_SSI0CLK		I/O	SSI-0 clock		
C_SPICLKA		I/O	SPI-A clock		
PK3_GPIO75	L18	I/O/Z	General-purpose input/output 75	PU	4 mA
M_SSI0FSS		I/O	SSI-0 frame		
C_SPISTEA		I/O	SPI-A slave transmit enable		
PK4_GPIO76	U4	I/O/Z	General-purpose input/output 76	PU	4 mA
M_MIITXEN		O	EMAC MII transmit enable		
M_SSI0TX		O	SSI-0 transmit data		
PK5_GPIO77	W3	I/O/Z	General-purpose input/output 77	PU	4 mA
M_MIITXCK		I	EMAC MII transmit clock		
M_SSI0RX		I/O	SSI-0 receive data		
PK6_GPIO78	V2	I/O/Z	General-purpose input/output 78	PU	4 mA
M_MIITXER		O	EMAC MII transmit error		
M_SSI0CLK		I/O	SSI-0 clock		
PK7_GPIO79	V3	I/O/Z	General-purpose input/output 79	PU	4 mA
M_MIICRS		I	EMAC MII carrier sense		
M_SSI0FSS		I/O	SSI-0 frame		
PL0_GPIO80	U1	I/O/Z	General-purpose input/output 80	PU	4 mA
M_MIIRXD3		I	EMAC MII receive data 3		
M_SSI1TX		O	SSI-1 transmit data		
PL1_GPIO81	U2	I/O/Z	General-purpose input/output 81	PU	4 mA
M_MIIRXD2		I	EMAC MII receive data 2		
M_SSI1RX		I/O	SSI-1 receive data		
PL2_GPIO82	U3	I/O/Z	General-purpose input/output 82	PU	4 mA
M_MIIRXD1		I	EMAC MII receive data 1		
M_SSI1CLK		I/O	SSI-1 clock		
PL3_GPIO83	T1	I/O/Z	General-purpose input/output 83	PU	4 mA
M_MIIRXD0		I	EMAC MII receive data 0		
M_SSI1FSS		I/O	SSI-1 frame		
PL4_GPIO84	U18	I/O/Z	General-purpose input/output 84	PU	4 mA
M_MIICOL		I	EMAC MII collision detect		
M_SSI3TX		O	SSI-3 transmit data		
PL5_GPIO85	T2	I/O/Z	General-purpose input/output 85	PU	4 mA
$\overline{\text{M\_MIIPHYRST}}$		O	EMAC PHY MII reset		
M_SSI3RX		I/O	SSI-3 receive data		
PL6_GPIO86	T3	I/O/Z	General-purpose input/output 86	PU	4 mA
$\overline{\text{M\_MIIPHYINTR}}$		O	EMAC PHY MII interrupt		
M_SSI3CLK		I/O	SSI-3 clock		
PL7_GPIO87	V18	I/O/Z	General-purpose input/output 87	PU	4 mA
M_MDIOCK		O	EMAC management data clock		
M_SSI3FSS		I/O	SSI-3 frame		
PM0_GPIO88	R1	I/O/Z	General-purpose input/output 88	PU	4 mA
M_MDIOD		I/O	EMAC management data input/output		
M_SSI2TX		O	SSI-2 transmit data		

Table 3-1. Terminal Functions(1) (continued)

TERMINAL		I/O/Z(2)	DESCRIPTION	PU or PD(3)	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PM1_GPIO89	R2	I/O/Z	General-purpose input/output 89	PU	4 mA
M_MIITXD3		O	EMAC MII transmit data 3		
M_SSI2RX		I/O	SSI-2 receive data		
PM2_GPIO90	R3	I/O/Z	General-purpose input/output 90	PU	4 mA
M_MIITXD2		O	EMAC MII transmit data 2		
M_SSI2CLK		I/O	SSI-2 clock		
PM3_GPIO91	P1	I/O/Z	General-purpose input/output 91	PU	4 mA
M_MIITXD1		O	EMAC MII transmit data 1		
M_SSI2FSS		I/O	SSI-2 frame		
PM4_GPIO92	P2	I/O/Z	General-purpose input/output 92	PU	4 mA
M_MIITXD0		O	EMAC MII transmit data 0		
C_MDXA		O	McBSP-A transmit data		
PM5_GPIO93	P3	I/O/Z	General-purpose input/output 93	PU	4 mA
M_MIIRXDV		I	EMAC MII receive data valid		
C_MDRA		I	McBSP-A receive data		
PM6_GPIO94	P4	I/O/Z	General-purpose input/output 94	PU	4 mA
M_MIIRXER		I	EMAC MII receive error		
C_MCLKXA		O	McBSP-A transmit clock		
PM7_GPIO95	N1	I/O/Z	General-purpose input/output 95	PU	4 mA
M_MIIRXCK		I	EMAC MII receive clock		
C_MFSXA		O	McBSP-A transmit frame sync		
PN0_GPIO96	L17	I/O/Z	General-purpose input/output 96	PU	4 mA
M_I2C0SCL		I/OD	I2C-0 clock open-drain bidirectional port		
C_MCLKRA		I	McBSP-A receive clock		
PN1_GPIO97	J17	I/O/Z	General-purpose input/output 97	PU	4 mA
M_I2C0SDA		I/OD	I2C-0 data open-drain bidirectional port		
C_MFSRA		I	McBSP-A receive frame sync		
PN2_GPIO98	J18	I/O/Z	General-purpose input/output 98	PU	4 mA
M_U1RX		I	UART-1 receive data		
PN3_GPIO99	G18	I/O/Z	General-purpose input/output 99	PU	4 mA
M_U1TX		O	UART-1 transmit data		
PN4_GPIO100	F18	I/O/Z	General-purpose input/output 100	PU	4 mA
M_U3TX		O	UART-3 transmit data		
PN5_GPIO101	H16	I/O/Z	General-purpose input/output 101	PU	4 mA
M_U3RX		I	UART-3 receive data		
PN6_GPIO102	R18	I/O/Z	General-purpose input/output 102	PU	4 mA
M_U4RX		I	UART-4 receive data		
M_EPI0S42(5)		I/O	EPI-0 signal 42		
M_USB0EPEN		O	USB-0 external power enable (optionally used in the host mode)		
PN7_GPIO103	T17	I/O/Z	General-purpose input/output 103	PU	4 mA
M_U4TX		O	UART-4 transmit data		
M_EPI0S43(5)		I/O	EPI-0 signal 43		
M_USB0PFLT		I	USB-0 external power error state (optionally used in the host mode)		

Table 3-1. Terminal Functions[(1)] (continued)

TERMINAL		I/O/Z[(2)]	DESCRIPTION	PU or PD[(3)]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PP0_GPIO104	G17	I/O/Z	General-purpose input/output 104	PU	4 mA
M_I2C1SCL		I/OD	I2C-1 clock open-drain bidirectional port		
C_I2CSDAA		I/OD	I2C-A data open-drain bidirectional port		
PP1_GPIO105	G16	I/O/Z	General-purpose input/output 105	PU	4 mA
M_I2C1SDA		I/OD	I2C-1 data open-drain bidirectional port		
C_I2CSCLA		I/OD	I2C-A clock open-drain bidirectional port		
PP2_GPIO106	F17	I/O/Z	General-purpose input/output 106	PU	4 mA
M_I2C0SCL		I/OD	I2C-0 clock open-drain bidirectional port		
C_EQEP1A		I	Enhanced QEP-1 input A		
PP3_GPIO107	E18	I/O/Z	General-purpose input/output 107	PU	4 mA
M_I2C0SDA		I/OD	I2C-0 data open-drain bidirectional port		
C_EQEP1B		I	Enhanced QEP-1 input B		
PP4_GPIO108	C19	I/O/Z	General-purpose input/output 108	PU	4 mA
M_I2C1SCL		I/OD	I2C-1 clock open-drain bidirectional port		
C_EQEP1S		I/O	Enhanced QEP-1 strobe		
PP5_GPIO109	C18	I/O/Z	General-purpose input/output 109	PU	4 mA
M_I2C1SDA		I/OD	I2C-1 data open-drain bidirectional port		
C_EQEP1I		I/O	Enhanced QEP-1 index		
PP6_GPIO110	C17	I/O/Z	General-purpose input/output 110	PU	4 mA
C_EQEP3S		I/O	Enhanced QEP-3 strobe		
C_EQEP2A		I	Enhanced QEP-2 input A		
PP7_GPIO111	B18	I/O/Z	General-purpose input/output 111	PU	4 mA
C_EQEP3I		I/O	Enhanced QEP-3 index		
C_EQEP2B		I	Enhanced QEP-2 input B		
PQ0_GPIO112	C16	I/O/Z	General-purpose input/output 112	PU	4 mA
C_EQEP3A		I	Enhanced QEP-1 input A		
C_EQEP2I		I/O	Enhanced QEP-2 index		
PQ1_GPIO113	C6	I/O/Z	General-purpose input/output 113	PU	4 mA
C_EQEP3B		I	Enhanced QEP-3 input B		
C_EQEP2S		I/O	Enhanced QEP-2 strobe		
PQ2_GPIO114	C5	I/O/Z	General-purpose input/output 114	PU	4 mA
M_U0RX		I	UART-0 receive data		
PQ3_GPIO115	C4	I/O/Z	General-purpose input/output 115	PU	4 mA
M_U0TX		O	UART-0 transmit data		
PQ4_GPIO116	C3	I/O/Z	General-purpose input/output 116	PU	4 mA
M_SSI1TX		O	SSI-1 transmit data		
PQ5_GPIO117	C2	I/O/Z	General-purpose input/output 117	PU	4 mA
M_SSI1RX		I	SSI-1 receive data		
PQ6_GPIO118	D3	I/O/Z	General-purpose input/output 118	PU	4 mA
C_SCITXDA		O	SCI-A transmit data		
PQ7_GPIO119	D2	I/O/Z	General-purpose input/output 119	PU	4 mA
C_SCIRXDA		I	SCI-A receive data		
PR0_GPIO120	N18	I/O/Z	General-purpose input/output 120	PU	4 mA
M_SSI3TX		O	SSI-3 transmit data		
PR1_GPIO121	M18	I/O/Z	General-purpose input/output 121	PU	4 mA
M_SSI3RX		I	SSI-3 receive data		

Table 3-1. Terminal Functions[(1)] (continued)

TERMINAL		I/O/Z[(2)]	DESCRIPTION	PU or PD[(3)]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
PR2_GPIO122 M_SSI3CLK	K18	I/O/Z I/O	General-purpose input/output 122 SSI-3 clock	PU	4 mA
PR3_GPIO123 M_SSI3FSS	M17	I/O/Z I/O	General-purpose input/output 123 SSI-3 frame	PU	4 mA
PR4_GPIO124 C_EPWM7A	J3	I/O/Z O	General-purpose input/output 124 Enhanced PWM-7 output A	PU	4 mA
PR5_GPIO125 C_EPWM7B	J2	I/O/Z O	General-purpose input/output 125 Enhanced PWM-7 output B	PU	4 mA
PR6_GPIO126 C_EPWM8A	J1	I/O/Z O	General-purpose input/output 126 Enhanced PWM-8 output A	PU	4 mA
PR7_GPIO127 C_EPWM8B	K3	I/O/Z O	General-purpose input/output 127 Enhanced PWM-8 output B	PU	4 mA
PS0_GPIO128 C_EPWM9A	K2	I/O/Z O	General-purpose input/output 128 Enhanced PWM-9 output A	PU	4 mA
PS1_GPIO129 C_EPWM9B	L3	I/O/Z O	General-purpose input/output 129 Enhanced PWM-9 output B	PU	4 mA
PS2_GPIO130 C_EPWM10A	L2	I/O/Z O	General-purpose input/output 130 Enhanced PWM-10 output A	PU	4 mA
PS3_GPIO131 C_EPWM10B	M3	I/O/Z O	General-purpose input/output 131 Enhanced PWM-10 output B	PU	4 mA
PS4_GPIO132 C_EPWM11A	M2	I/O/Z O	General-purpose input/output 132 Enhanced PWM-11 output A	PU	4 mA
PS5_GPIO133 C_EPWM11B	M1	I/O/Z O	General-purpose input/output 133 Enhanced PWM-11 output B	PU	4 mA
PS6_GPIO134 C_EPWM12A	N3	I/O/Z O	General-purpose input/output 134 Enhanced PWM-12 output A	PU	4 mA
PS7_GPIO135 C_EPWM12B	N2	I/O/Z O	General-purpose input/output 135 Enhanced PWM-12 output B	PU	4 mA

ADVANCE INFORMATION

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
Resets					
$\overline{\text{XRS}}$	C1	I/OD	Digital Subsystem Reset (in) and Watchdog/Power-on Reset (out). In most applications, TI recommends that the $\overline{\text{XRS}}$ pin be tied with the $\overline{\text{ARS}}$ pin. The Digital Subsystem has a built-in power-on-reset (POR) circuit, and during a power-on condition, this pin is driven low by the Digital Subsystem. This pin is also driven low by the Digital Subsystem when a watchdog reset occurs. During watchdog reset, the $\overline{\text{XRS}}$ pin is driven low for the watchdog reset duration of 512 OSCCLK cycles. If need be, an external circuitry may also drive this pin to assert device reset. In this case, TI recommends that this pin be driven by an open-drain device. An R-C circuit must be connected to this pin for noise immunity reasons. Regardless of the source, a device reset causes the Digital Subsystem to terminate execution. The Cortex™-M3 program counter points to the address contained at the location 0x00000004. The C28 program counter points to the address contained at the location 0x3FFFC0. When reset is deactivated, execution begins at the location designated by the program counter. The output buffer of this pin is an open-drain with an internal pullup.	PU	4 mA
$\overline{\text{ARS}}$	A3	I/OD	Analog Subsystem Reset (in) and Power-on Reset (out).In most applications, TI recommends that the $\overline{\text{ARS}}$ pin be tied with the $\overline{\text{XRS}}$ pin. The Analog Subsystem has a built-in power-on-reset (POR) circuit, and during a power-on condition, this pin is driven low by the Analog Subsystem. If need be, external circuitry may also drive this pin to assert a device reset. In this case, TI recommends that this pin be driven by an open-drain device. An R-C circuit must be connected to this pin for noise immunity reasons. Regardless of the source, the Analog Subsystem reset causes the digital logic associated with the Analog Subsystem, to enter reset state. The output buffer of this pin is an open-drain with an internal pullup.	PU	4 mA

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
Clocks					
X1	J19	I	External oscillator input or on-chip crystal-oscillator input. To use the on-chip oscillator, a quartz crystal or a ceramic resonator must be connected across X1 and X2. See Figure 2-7.		
X2	G19	O	On-chip crystal-oscillator output. A quartz crystal or a ceramic resonator must be connected across X1 and X2. If X2 is not used, it must be left unconnected. See Figure 2-7.		
V_{SSOSC}	H18		Clock Oscillator Ground Pin. Use this pin to connect the GND of external crystal load capacitors or the ground pin of 3-terminal ceramic resonators with built-in capacitors. ***Do not*** connect to board ground. See Figure 2-7.		
V_{SSOSC}	H19		Clock Oscillator Ground Pin. Use this pin to connect the GND of external crystal load capacitors or the ground pin of 3-terminal ceramic resonators with built-in capacitors. ***Do not*** connect to board ground. See Figure 2-7.		
XCLKIN	see PJ7_GPIO63	I	External oscillator input. This pin feeds a clock from an external 3.3-V oscillator to internal USB PLL module and to the CAN peripherals.		
XCLKOUT	see PF2_GPIO34	O/Z	External oscillator output. This pin outputs a clock divided-down from the internal PLL System Clock. The divide ratio is defined in the XCLKCFG register.		
Boot Pins					
BOOT_0	see PG3_GPIO43	I	One of four boot mode pins. BOOT_0 selects a specific configuration source from which the Concerto device boots on start-up.	PU	
BOOT_1	see PG7_GPIO47	I	One of four boot mode pins. BOOT_1 selects a specific configuration source from which the Concerto device boots on start-up.	PU	
BOOT_2	see PF3_GPIO35	I	One of four boot mode pins. BOOT_2 selects a specific configuration source from which the Concerto device boots on start-up.	PU	
BOOT_3	see PF2_GPIO34	I	One of four boot mode pins. BOOT_3 selects a specific configuration source from which the Concerto device boots on start-up.	PU	
JTAG					
$\overline{TRST}$	N19	I	JTAG test reset with internal pulldown. $\overline{TRST}$, when driven high, gives the scan system control of the operations of the device. If this signal is not connected or driven low, the device operates in its functional mode, and the test reset signals are ignored. **NOTE:** $\overline{TRST}$ is an active-low test pin and must be maintained low during normal device operation. An external pull-down resistor is required on this pin. The value of this resistor should be based on drive strength of the debugger pods applicable to the design. A 2.2-kΩ resistor generally offers adequate protection. Since the value of the resistor is application-specific, TI recommends that each target board be validated for proper operation of the debugger and the application.	PD	
TCK	L19	I	JTAG test clock		
TMS	M19	I	JTAG test-mode select (TMS) with internal pullup. This serial control input is clocked into the TAP controller on the rising edge of TCK.	PU	

Table 3-1. Terminal Functions(1) (continued)

TERMINAL		I/O/Z(2)	DESCRIPTION	PU or PD(3)	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
TDI	K19	I	JTAG test data input (TDI) with internal pullup. TDI is clocked into the selected register (instruction or data) on a rising edge of TCK.	PU	
TDO	T19	O	JTAG scan out, test data output (TDO). The contents of the selected register (instruction or data) are shifted out of TDO on the falling edge of TCK.		4 mA
EMU0	P19	I/O/Z	Emulator pin 0. When $\overline{\text{TRST}}$ is driven high, this pin is used as an interrupt to or from the emulator system and is defined as input/output through the JTAG scan. This pin is also used to put the device into boundary-scan mode. With the EMU0 pin at a logic-high state and the EMU1 pin at a logic-low state, a rising edge on the $\overline{\text{TRST}}$ pin would latch the device into boundary-scan mode. **NOTE:** An external pullup resistor is required on this pin. The value of this resistor should be based on the drive strength of the debugger pods applicable to the design. A 2.2-kΩ to 4.7-kΩ resistor is generally adequate. Since the value of the resistor is application-specific, TI recommends that each target board be validated for proper operation of the debugger and the application. **NOTE:** If EMU0 is 0 and EMU1 is 1 when coming out of reset, the device enters Wait-in-Reset mode. WIR suspends bootloader execution, allowing the Emulator to connect to the device and to modify FLASH contents.	PU	4 mA
EMU1	R19	I/O/Z	Emulator pin 1. When $\overline{\text{TRST}}$ is driven high, this pin is used as an interrupt to or from the emulator system and is defined as input/output through the JTAG scan. This pin is also used to put the device into boundary-scan mode. With the EMU0 pin at a logic-high state and the EMU1 pin at a logic-low state, a rising edge on the $\overline{\text{TRST}}$ pin would latch the device into boundary-scan mode. **NOTE:** An external pullup resistor is required on this pin. The value of this resistor should be based on the drive strength of the debugger pods applicable to the design. A 2.2-kΩ to 4.7-kΩ resistor is generally adequate. Since the value of the resistor is application-specific, TI recommends that each target board be validated for proper operation of the debugger and the application. **NOTE:** If EMU0 is 0 and EMU1 is 1 when coming out of reset, the device enters Wait-in-Reset mode. WIR suspends bootloader execution, allowing the Emulator to connect to the device and to modify FLASH contents.	PU	4 mA

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
ITM Trace (ARM® Instrumentation Trace Macrocell)					
TRACED0	see PF3_GPIO35	O	ITM Trace data 0		4 mA
TRACED1	see PG3_GPIO43	O	ITM Trace data 1		4 mA
TRACED2	see PF0_GPIO32	O	ITM Trace data 2		4 mA
TRACED3	see PF1_GPIO33	O	ITM Trace data 3		4 mA
TRACECLK	see PF2_GPIO34	O	ITM Trace clock		4 mA
Test Pins					
FLT1	K1	I/O	FLASH Test Pin 1. Reserved for TI. Must be left unconnected.		
FLT2	L1	I/O	FLASH Test Pin 2. Reserved for TI. Must be left unconnected.		
Internal Voltage Regulator Control					
$\overline{\text{VREG18EN}}$	A15		Internal 1.8-V VREG Enable/Disable for V_{DD18}. Pull low to enable the internal 1.8-V voltage regulator (VREG18), pull high to disable VREG18.	PD	
$\overline{\text{VREG12EN}}$	E19		Internal 1.2-V VREG Enable/Disable for V_{DD12}. Pull low to enable the internal 1.2-V voltage regulator (VREG12), pull high to disable VREG12.	PD	

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
Digital Logic Power Pins for I/Os, Flash, USB, and Internal Oscillators					
V_{DDIO}	D4		3.3-V Digital I/O and FLASH Power Pin. Tie with a 0.1-µF capacitor (typical) close to the pin.		
V_{DDIO}	D5				
V_{DDIO}	D15				
V_{DDIO}	D16				
V_{DDIO}	G7				
V_{DDIO}	G13				
V_{DDIO}	G8				
V_{DDIO}	G9				
V_{DDIO}	G10				
V_{DDIO}	G11				
V_{DDIO}	G12				
V_{DDIO}	H7				
V_{DDIO}	H13				
V_{DDIO}	J7				
V_{DDIO}	J13				
V_{DDIO}	N8				
V_{DDIO}	N9				
V_{DDIO}	N10				
V_{DDIO}	N11				
V_{DDIO}	K7				
V_{DDIO}	L7				
V_{DDIO}	K13				
V_{DDIO}	L13				
V_{DDIO}	T4				
V_{DDIO}	T5				
V_{DDIO}	T7				
V_{DDIO}	T8				
V_{DDIO}	T15				
V_{DDIO}	T16				
V_{DDIO}	T13				
V_{DDIO}	U13				
Digital Logic Power Pins (Analog Subsystem)					
V_{DD18}	C7		1.8-V Digital Logic Power Pins (associated with the Analog Subsystem) - no supply needed when using internal VREG18. Tie with 1.2-µF (minimum) ceramic capacitor (10% tolerance) to ground when using internal VREG. Higher value capacitors may be used but could impact supply-rail ramp-up time.		
V_{DD18}	D7				
V_{DD18}	D12				
V_{DD18}	D13				

Table 3-1. Terminal Functions[1] (continued)

TERMINAL		I/O/Z[2]	DESCRIPTION	PU or PD[3]	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
Digital Logic Power Pins (Master and Control Subsystems)					
V_{DD12}	M7		1.2-V Digital Logic Power Pins - no supply needed when using internal VREG12. Tie with 250-nF (minimum) to 750-nF (maximum) ceramic capacitor (10% tolerance) to ground when using internal VREG. Higher value capacitors may be used but could impact supply-rail ramp-up time.		
V_{DD12}	M13				
V_{DD12}	N7				
V_{DD12}	N12				
V_{DD12}	N13				
V_{DD12}	T10				
V_{DD12}	T11				
V_{DD12}	T12				
Digital Logic Ground (Analog, Master, and Control Subsystems)					
V_{SS}	A1		Digital Ground		
V_{SS}	A2				
V_{SS}	A18				
V_{SS}	A19				
V_{SS}	B1				
V_{SS}	B19				
V_{SS}	D6				
V_{SS}	D14				
V_{SS}	E4				
V_{SS}	E16				
V_{SS}	H8				
V_{SS}	H9				
V_{SS}	H10				
V_{SS}	H11				
V_{SS}	H12				
V_{SS}	J4				
V_{SS}	J8				
V_{SS}	J9				
V_{SS}	J10				
V_{SS}	J11				
V_{SS}	J12				
V_{SS}	J16				
V_{SS}	K4				
V_{SS}	K8				
V_{SS}	K9				
V_{SS}	K10				
V_{SS}	K11				
V_{SS}	K12				
V_{SS}	K16				

ADVANCE INFORMATION

Table 3-1. Terminal Functions(1) (continued)

TERMINAL		I/O/Z(2)	DESCRIPTION	PU or PD(3)	OUTPUT BUFFER STRENGTH
NAME	ZWT BALL NO.				
V_{SS}	L4		Digital Ground		
V_{SS}	L8				
V_{SS}	L9				
V_{SS}	L10				
V_{SS}	L11				
V_{SS}	L12				
V_{SS}	L16				
V_{SS}	M8				
V_{SS}	M9				
V_{SS}	M10				
V_{SS}	M11				
V_{SS}	M12				
V_{SS}	R4				
V_{SS}	R16				
V_{SS}	T6				
V_{SS}	T9				
V_{SS}	T14				
V_{SS}	V1				
V_{SS}	V19				
V_{SS}	W1				
V_{SS}	W2				
V_{SS}	W18				
V_{SS}	W19				

(1) Throughout this table, Master Subsystem signals are denoted by the color "blue"; Control Subsystem signals are denoted by the color "green"; and Analog Subsystem signals are denoted by the color "orange".

(2) I = Input, O = Output, Z = High Impedance, OD = Open Drain

(3) PU = Pullup, PD = Pulldown
- GPIO_MUX1 pullups can be enabled or disabled by Cortex™-M3 software (disabled on reset).
- GPIO_MUX2 pullups can be enabled or disabled by C28x software (disabled on reset).
- AIO_MUX1 and AIO_MUX2 terminals do not have pullups or pulldowns.
- All other pullups are always enabled ($\overline{\text{XRS}}$, $\overline{\text{ARS}}$, TMS, TDI, $\overline{\text{EMU0}}$, EMU1).
- All pulldowns are always enabled ($\overline{\text{VREG18EN}}$, $\overline{\text{VREG12EN}}$, $\overline{\text{TRST}}$).

(4) All I/Os, except for GPIO199, are glitch-free during power up and power down. See Section 2.11.

(5) This muxing option is only available on silicon Revision A devices; this muxing option is not available on silicon Revision 0 devices.

(6) Output from the Concerto ePWM is meant for the external ADC (if present).

4 Device Operating Conditions

4.1 Absolute Maximum Ratings(1) (2)

Supply voltage range, V_{DDIO} (I/O and Flash)	with respect to V_{SS}	–0.3 V to 4.6 V
Supply voltage range, V_{DD18}	with respect to V_{SS}	–0.3 V to 2.5 V
Supply voltage range, V_{DD12}	with respect to V_{SS}	–0.3 V to 1.5 V
Analog voltage range, V_{DDA}	with respect to V_{SSA}	–0.3 V to 4.6 V
Input voltage range, V_{IN} (3.3 V)		–0.3 V to 4.6 V
Output voltage range, V_O		–0.3 V to 4.6 V
Input clamp current, I_{IK} ($V_{IN} < 0$ or $V_{IN} > V_{DDIO}$)(3)		±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{DDIO}$)		±20 mA
Free-Air temperature, T_A		–40°C to 125°C
Junction temperature range, T_J (4)		–40°C to 150°C
Storage temperature range, T_{stg} (4)		–65°C to 150°C

(1) Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under Section 4.2 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) All voltage values are with respect to V_{SS}, unless otherwise noted.
(3) Continuous clamp current per pin is ± 2 mA.
(4) Long-term high-temperature storage or extended use at maximum temperature conditions may result in a reduction of overall device life. For additional information, see *IC Package Thermal Metrics Application Report* (literature number SPRA953) and *Reliability Data for TMS320LF24xx and TMS320F28xx Devices Application Report* (literature number SPRA963).

4.2 Recommended Operating Conditions

		MIN	NOM	MAX	UNIT
Device supply voltage, I/O, V_{DDIO}(1)		2.97	3.3	3.63	V
Device supply voltage, Analog Subsystem, V_{DD18} (when internal VREG is disabled and 1.8 V is supplied externally)		1.71	1.8	1.995	V
Device supply voltage, Master and Control Subsystems, V_{DD12} (when internal VREG is disabled and 1.2 V is supplied externally)		1.14	1.2	1.32	V
Supply ground, V_{SS}			0		V
Analog supply voltage, V_{DDA}(1)		2.97	3.3	3.63	V
Analog ground, V_{SSA}			0		V
Device clock frequency (system clock)		2		60	MHz
High-level input voltage, V_{IH} (3.3 V)		V_{DDIO} * 0.7		V_{DDIO} + 0.3	V
Low-level input voltage, V_{IL} (3.3 V)		V_{SS} – 0.3		V_{DDIO} * 0.3	V
High-level output source current, $V_{OH} = V_{OH(MIN)}$, I_{OH}	All GPIO/AIO pins			–4	mA
	Group 2(2)			–8	mA
Low-level output sink current, $V_{OL} = V_{OL(MAX)}$, I_{OL}	All GPIO/AIO pins			4	mA
	Group 2(2)			8	mA
Free-Air temperature, T_A	T version	–40		105	°C
	S version	–40		125	
	Q version (Q100 qualification)	–40		125	
Junction temperature, T_J	T version	–40		125	°C
	S version	–40		150	
	Q version (Q100 qualification)	–40		150	

(1) V_{DDIO} and V_{DDA} should be maintained within approximately 0.3 V of each other.
(2) Group 2 pins are as follows: PD3_GPIO19, PE2_GPIO26, PE3_GPIO27, PH6_GPIO54, PH7_GPIO55, EMU0, TDO, EMU1, PD0_GPIO16, AIO7, AIO4.

4.3 Electrical Characteristics

over recommended operating conditions (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage		$I_{OH} = I_{OH}$ MAX		V_{DDIO} * 0.8			V
			I_{OH} = 50 µA		V_{DDIO} – 0.2			
V_{OL}	Low-level output voltage		$I_{OL} = I_{OL}$ MAX				V_{DDIO} * 0.2	V
I_{IL}	Input current (low level)	Pin with pullup enabled	V_{DDIO} = 3.3 V, V_{IN} = 0 V	All GPIO/AIO		–140		µA
				$\overline{XRS}$ pin and $\overline{ARS}$ pin		–300		
		Pin with pulldown enabled	V_{DDIO} = 3.3 V, V_{IN} = 0 V			±2		
I_{IH}	Input current (high level)	Pin with pullup enabled	V_{DDIO} = 3.3 V, $V_{IN} = V_{DDIO}$			±2		µA
		Pin with pulldown enabled	V_{DDIO} = 3.3 V, $V_{IN} = V_{DDIO}$			50		
I_{OZ}	Output current, pullup or pulldown disabled		$V_O = V_{DDIO}$ or 0 V			±2		µA
C_I	Input capacitance					2		pF
	Digital Subsystem POR reset release delay time		Time after POR event is removed to $\overline{XRS}$ release			50		µs
	Analog Subsystem POR reset release delay time		Time after POR event is removed to $\overline{ARS}$ release		400		800	µs
	VREG V_{DD18} output		Internal VREG18 on		1.77		1.935	V
	VREG V_{DD12} output		Internal VREG12 on			1.2		V

5 Electrical Specifications

5.1 Current Consumption

Table 5-1. Current Consumption at 150-MHz C28x SYSCLKOUT and 75-MHz M3SSCLK(1)(2)

MODE	TEST CONDITIONS(3)	VREG ENABLED				VREG DISABLED							
		I_{DDIO}(4)		I_{DDA}		I_{DD18}		I_{DD12}		I_{DDIO}(4)		I_{DDA}	
		TYP(5)	MAX	TYP(5)	MAX	TYP(5)	MAX	TYP(5)	MAX	TYP(5)	MAX	TYP(5)	MAX
Operational (RAM)	The following Cortex™-M3 peripherals are exercised: • I2C1 • SSI1, SSI2 • UART0, UART1, UART2 • CAN0 • USB • µDMA • Timer0, Timer1 • µCRC • WDOG0, WDOG1 • Flash • Internal Oscillator 1, Internal Oscillator 2 The following C28x peripherals are exercised: • McBSP • eQEP1, eQEP2 • eCAP1, eCAP2, eCAP3, eCAP4 • SCI-A • SPI-A • I2C • DMA • VCU • FPU • Flash The following Analog peripherals are exercised: • ADC1, ADC2 • Comparator 1, Comparator 2, Comparator 3, Comparator 4, Comparator 5, Comparator 6	445 mA	–	32 mA	–	20 mA	–	367 mA	–	33 mA	–	32 mA	–

(1) Currently only typical current consumption data is available, maximum numbers will come in another release of this data sheet.
(2) The numbers in Table 5-1 are not assured at this time, and are subject to change.
(3) The following is done in a loop:
- Code is running out of RAM.
- All I/O pins are left unconnected.
- All the communication peripherals are exercised in loop-back mode.
- USB – Only logic is exercised by loading and unloading FIFO.
- µDMA does memory-to-memory transfer.
- DMA does memory-to-memory transfer.
- VCU – CRC calculated and checked.
- FPU – Float operations performed.
- ePWM – 6 enabled and generates 150-kHz PWM output on 12 pins, HRPWM clock enabled.
- Timers and Watchdog serviced.
- eCAP in APWM mode generates 36.6-kHz output on 4 pins.
- ADC performs continuous conversion.
- FLASH is continuously read and in active state.
- XCLKOUT is turned off.

(4) I_{DDIO} current is dependent on the electrical loading on the I/O pins.
(5) The TYP numbers are applicable over room temperature and nominal voltage.

Table 5-1. Current Consumption at 150-MHz C28x SYSCLKOUT and 75-MHz M3SSCLK[1][2] (continued)

MODE	TEST CONDITIONS[3]	VREG ENABLED				VREG DISABLED							
		I_{DDIO}[4]		I_{DDA}		I_{DD18}		I_{DD12}		I_{DDIO}[4]		I_{DDA}	
		TYP[5]	MAX	TYP[5]	MAX	TYP[5]	MAX	TYP[5]	MAX	TYP[5]	MAX	TYP[5]	MAX
SLEEP IDLE	• PLL is on. • Cortex™-M3 CPU is not executing. • M3SSCLK is on. • C28CLKIN is on. • C28x™ CPU is not executing. • C28CPUCLK is off. • C28SYSCLK is on.	80 mA	–	315 µA	–	14 mA	–	51 mA	–	15 mA	–	315 µA	–
SLEEP STANDBY	• PLL is on. • Cortex™-M3 CPU is not executing. • M3SSCLK is on. • C28CLKIN is off. • C28x™ CPU is not executing. • C28CPUCLK is off. • C28SYSCLK is off.	71 mA	–	210 µA	–	15 mA	–	42 mA	–	14 mA	–	205 µA	–
DEEP SLEEP STANDBY	• PLL is off. • Cortex™-M3 CPU is not executing. • M3SSCLK is 32 kHz. • C28CLKIN is off. • C28x™ CPU is not executing. • C28CPUCLK is off. • C28SYSCLK is off.	24 mA	–	200 µA	–	2 mA	–	19 mA	–	3 mA	–	195 µA	–

NOTE

The peripheral-I/O multiplexing implemented in the device prevents all available peripherals from being used at the same time because more than one peripheral function may share an I/O pin. It is, however, possible to turn on the clocks to all the peripherals at the same time, although such a configuration is not useful. If the clocks to all the peripherals are turned on at the same time, the current drawn by the device will be more than the numbers specified in the current consumption table.

5.2 Thermal Design Considerations

Based on the end-application design and operational profile, the I_{DD12}, I_{DD18}, and I_{DDIO} currents could vary. Systems that exceed the recommended maximum power dissipation in the end product may require additional thermal enhancements. Ambient temperature (T_A) varies with the end application and product design. The critical factor that affects reliability and functionality is T_J, the junction temperature, not the ambient temperature. Hence, care should be taken to keep T_J within the specified limits. T_{case} should be measured to estimate the operating junction temperature T_J. T_{case} is normally measured at the center of the package top-side surface. For more details about thermal metrics and definitions, see the *Semiconductor and IC Package Thermal Metrics Application Report* (literature number SPRA953) and the *Reliability Data for TMS320LF24xx and TMS320F28xx Devices Application Report* (literature number SPRA963).

5.3 Timing Parameter Symbology

Timing parameter symbols used are created in accordance with JEDEC Standard 100. To shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows:

Lowercase subscripts and their meanings:		**Letters and symbols and their meanings:**	
a	access time	H	High
c	cycle time (period)	L	Low
d	delay time	V	Valid
f	fall time	X	Unknown, changing, or don't care level
h	hold time	Z	High impedance
r	rise time		
su	setup time		
t	transition time		
v	valid time		
w	pulse duration (width)		

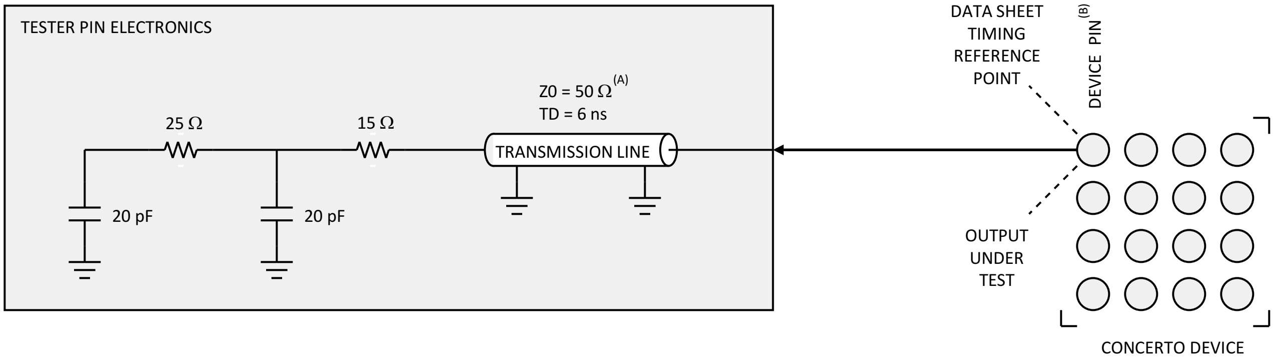
5.3.1 General Notes on Timing Parameters

All output signals from the 28x devices (including XCLKOUT) are derived from an internal clock such that all output transitions for a given half-cycle occur with a minimum of skewing relative to each other.

The signal combinations shown in the following timing diagrams may not necessarily represent actual cycles. For actual cycle examples, see the appropriate cycle description section of this document.

5.3.2 Test Load Circuit

This test load circuit is used to measure all switching characteristics provided in this document.



A. Input requirements in this data sheet are tested with an input slew rate of < 4 Volts per nanosecond (4 V/ns) at the device pin.

B. The data sheet provides timing at the device pin. For output timing analysis, the tester pin electronics and its transmission line effects must be taken into account. A transmission line with a delay of 2 ns or longer can be used to produce the desired transmission line effect. The transmission line is intended as a load only. It is not necessary to add or subtract the transmission line delay (2 ns or longer) from the data sheet timing.

Figure 5-1. 3.3-V Test Load Circuit

ADVANCE INFORMATION

5.4 Clock Frequencies, Requirements, and Characteristics

This section provides the frequencies and timing requirements of the input clocks; PLL lock times; frequencies of the internal clocks; and the frequency and switching characteristics of the output clock.

5.4.1 Input Clock Frequency and Timing Requirements, PLL Lock Times

Table 5-2 shows the frequency requirements for the input clocks to the F28M36x devices. Table 5-3 shows the input clock cycle time. Table 5-4, Table 5-5, Table 5-6, and Table 5-7 show the timing requirements for the input clocks to the F28M36x devices. Table 5-8 shows the PLL lock times for the Main PLL and the USB PLL. The Main PLL operates from the X1 or X1/X2 input clock pins, and the USB PLL operates from the XCLKIN input clock pin.

Table 5-2. Input Clock Frequency

		MIN	MAX	UNIT
$f_{(OSC)}$	Frequency, X1/X2, from external crystal or resonator	2	20	MHz
$f_{(OCI)}$	Frequency, X1, from external oscillator ***(PLL enabled)***	2	30	MHz
$f_{(OCI)}$	Frequency, X1, from external oscillator ***(PLL disabled)***	2	100	MHz
$f_{(XCI)}$	Frequency, XCLKIN, from external oscillator	2	60	MHz

Table 5-3. Input Clock Cycle Time

NO.			MAX	MIN	UNIT
C1	$t_{c(OSC)}$	Cycle time, X1/X2, from external crystal or resonator	500	50	ns
C2	$t_{c(OCI)}$	Cycle time, X1, from external oscillator ***(PLL enabled)***	500	33.3	ns
C2	$t_{c(OCI)}$	Cycle time, X1, from external oscillator ***(PLL disabled)***	500	10	ns
C3	$t_{c(XCI)}$	Cycle time, XCLKIN, from external oscillator	500	16.6	ns

Table 5-4. X1 Timing Requirements - PLL Enabled(1)

NO.			MIN	MAX	UNIT
C4	$t_{f(OCI)}$	Fall time, X1		6	ns
C5	$t_{r(OCI)}$	Rise time, X1		6	ns
C6	$t_{w(OCL)}$	Pulse duration, X1 low as a percentage of $t_{c(OCI)}$	45	55	%
C7	$t_{w(OCH)}$	Pulse duration, X1 high as a percentage of $t_{c(OCI)}$	45	55	%

(1) The possible Main PLL configuration modes are shown in Table 2-19 to Table 2-22.

Table 5-5. X1 Timing Requirements - PLL Disabled

NO.				MIN	MAX	UNIT
C4	$t_{f(OCI)}$	Fall time, X1	Up to 20 MHz		6	ns
			20 MHz to 100 MHz		2	
C5	$t_{r(OCI)}$	Rise time, X1	Up to 20 MHz		6	ns
			20 MHz to 100 MHz		2	
C6	$t_{w(OCL)}$	Pulse duration, X1 low as a percentage of $t_{c(OCI)}$		45	55	%
C7	$t_{w(OCH)}$	Pulse duration, X1 high as a percentage of $t_{c(OCI)}$		45	55	%

Table 5-6. XCLKIN Timing Requirements - PLL Enabled(1)

NO.			MIN	MAX	UNIT
C8	$t_{f(XCI)}$	Fall time, XCLKIN		6	ns
C9	$t_{r(XCI)}$	Rise time, XCLKIN		6	ns
C10	$t_{w(XCL)}$	Pulse duration, XCLKIN low as a percentage of $t_{c(XCI)}$	45	55	%
C11	$t_{w(XCH)}$	Pulse duration, XCLKIN high as a percentage of $t_{c(XCI)}$	45	55	%

(1) The possible USB PLL configuration modes are shown in Table 2-23 and Table 2-24.

Table 5-7. XCLKIN Timing Requirements - PLL Disabled

NO.				MIN	MAX	UNIT
C8	$t_{f(XCI)}$	Fall time, XCLKIN	Up to 20 MHz		6	ns
			20 MHz to 100 MHz		2	
C9	$t_{r(XCI)}$	Rise time, XCLKIN	Up to 20 MHz		6	ns
			20 MHz to 100 MHz		2	
C10	$t_{w(XCL)}$	Pulse duration, XCLKIN low as a percentage of $t_{c(XCI)}$		45	55	%
C11	$t_{w(XCH)}$	Pulse duration, XCLKIN high as a percentage of $t_{c(XCI)}$		45	55	%

Table 5-8. PLL Lock Times

		MIN	NOM	MAX	UNIT
$t_{(PLL)}$	Lock time, Main PLL (X1, from external oscillator)		2000(1)		input clock cycles
$t_{(USB)}$	Lock time, USB PLL (XCLKIN, from external oscillator)		2000(1)		input clock cycles

(1) For example, if the input clock to the PLL is 10 MHz, then the PLL lock time is 100 ns x 2000 = 200 µs.

5.4.2 Internal Clock Frequencies

Table 5-9 provides the clock frequencies for the internal clocks of the F28M36x devices.

Table 5-9. Internal Clock Frequencies (150-MHz Devices)

		MIN	NOM	MAX	UNIT
$f_{(USB)}$	Frequency, USBPLLCLK		60		MHz
$f_{(PLL)}$	Frequency, PLLSYSCLK	2		150	MHz
$f_{(OCK)}$	Frequency, OSCCLK	2		100	MHz
$f_{(M3C)}$	Frequency, M3SSCLK	2		100(1)	MHz
$f_{(ADC)}$	Frequency, ASYSCLK	2		37.5	MHz
$f_{(SYS)}$	Frequency, C28SYSCLK	2		150(1)	MHz
$f_{(HSP)}$	Frequency, C28HSPCLK	2		150(1)	MHz
$f_{(LSP)}$	Frequency, C28LSPCLK(2)	2	37.5(3)	150(1)	MHz
$f_{(10M)}$	Frequency, 10MHZCLK		10		MHz
$f_{(32K)}$	Frequency, 32KHZCLK		32		kHz

(1) An integer divide ratio must be maintained between the C28x and Cortex™-M3 clock frequencies. For example, when the C28x is configured to run at a maximum frequency of 150 MHz, the fastest allowable frequency for the Cortex™-M3 will be 75 MHz. See Figure 2-10 and Figure 2-11 to see the internal clocks and clock divider options.
(2) Lower LSPCLK will reduce device power consumption.
(3) This is the default reset value if C28SYSCLK = 150 MHz.

5.4.3 Output Clock Frequency and Switching Characteristics

Table 5-10 provides the frequency of the output clock from the F28M36x devices. Table 5-11 shows the switching characteristics of the output clock from the F28M36x devices, XCLKOUT.

Table 5-10. Output Clock Frequency

NO.			MIN	MAX	UNIT
C14	$f_{(XCO)}$	Frequency, XCLKOUT	2	37.5	MHz

Table 5-11. XCLKOUT Switching Characteristics (PLL Bypassed or Enabled)[1][2]

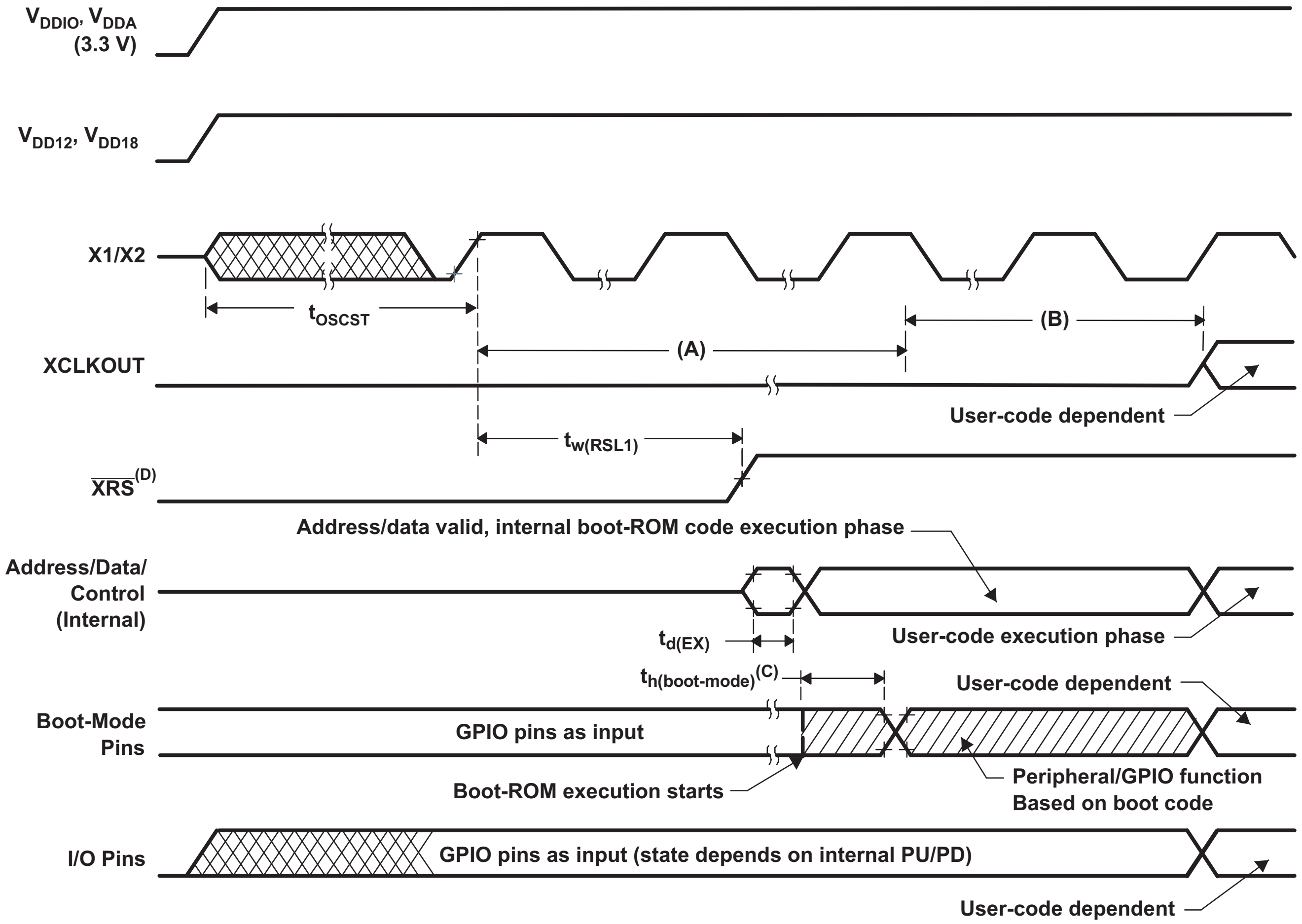
over recommended operating conditions (unless otherwise noted)

NO.	PARAMETER		MIN	MAX	UNIT
C15	$t_{f(XCO)}$	Fall time, XCLKOUT		5	ns
C16	$t_{r(XCO)}$	Rise time, XCLKOUT		5	ns
C17	$t_{w(XCOL)}$	Pulse duration, XCLKOUT low	H – 2	H + 2	ns
C18	$t_{w(XCOH)}$	Pulse duration, XCLKOUT high	H – 2	H + 2	ns

(1) A load of 40 pF is assumed for these parameters.
(2) $H = 0.5t_{c(XCO)}$

5.5 Power Sequencing

There is no power sequencing requirement needed to ensure the device is in the proper state after reset or to prevent the I/Os from glitching during power up and power down. (All I/Os, except for GPIO199, are glitch-free during power up and power down.) No voltage larger than a diode drop (0.7 V) above V_{DDIO} should be applied to any digital pin (for analog pins, this value is 0.7 V above V_{DDA}) prior to powering up the device. Voltages applied to pins on an unpowered device can bias internal p-n junctions in unintended ways and produce unpredictable results.



A. Upon power up, PLLSYSCLK is OSCCLK/8. Since the XCLKOUTDIV bits in the XCLK register come up with a reset state of 0, PLLSYSCLK is further divided by 4 before PLLSYSCLK appears at XCLKOUT. XCLKOUT = OSCCLK/32 during this phase.

B. Boot ROM configures the SYSDIVSEL bits for /1 operation. XCLKOUT = OSCCLK/4 during this phase. Note that XCLKOUT will not be visible at the pin until explicitly configured by user code.

C. After reset, the boot ROM code samples Boot Mode pins. Based on the status of the Boot Mode pin, the boot code branches to destination memory or boot code function. If boot ROM code executes after power-on conditions (in debugger environment), the boot code execution time is based on the current M3SSCLK speed. The M3SSCLK will be based on user environment and could be with or without PLL enabled.

D. Using the $\overline{XRS}$ pin is optional due to the on-chip power-on reset (POR) circuitry.

Figure 5-2. Power-On Reset

ADVANCE INFORMATION

Table 5-12. Reset ($\overline{XRS}$) Timing Requirements

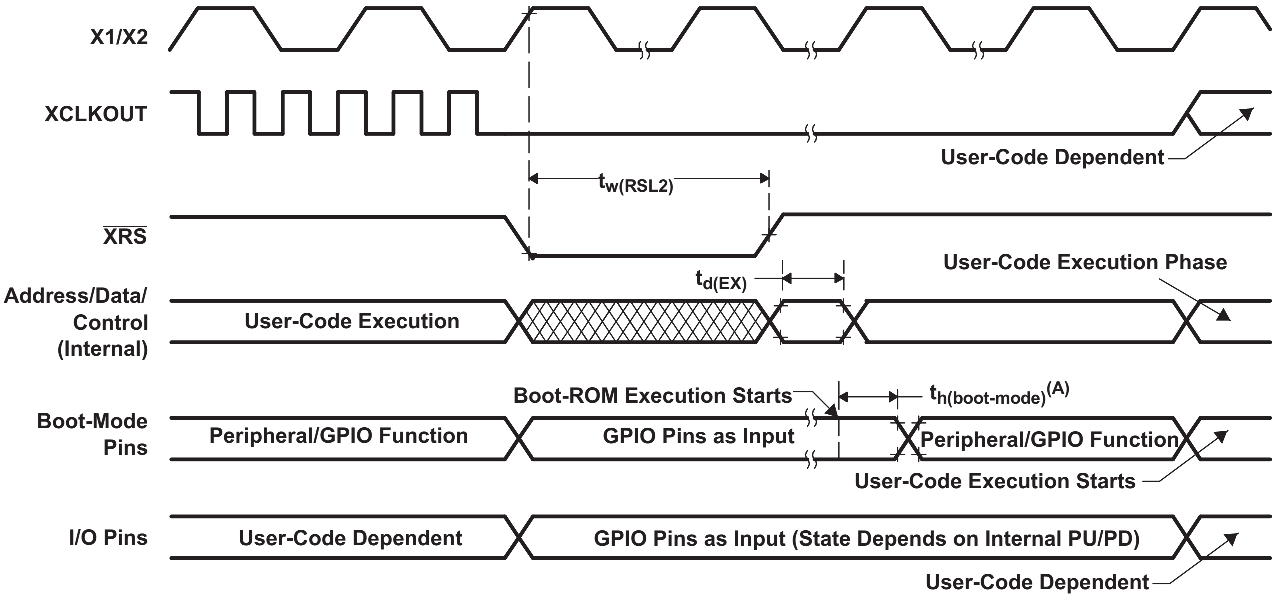
		MIN	MAX	UNIT
$t_{h(boot-mode)}$	Hold time for boot-mode pins	$14000t_{c(M3C)}$		cycles
$t_{w(RSL2)}$	Pulse duration, $\overline{XRS}$ low on warm reset	$32t_{c(OCK)}$		cycles

Table 5-13. Reset ($\overline{XRS}$) Switching Characteristics

over recommended operating conditions (unless otherwise noted)

PARAMETER		MIN	TYP	MAX	UNIT
$t_{w(RSL1)}$	Pulse duration, $\overline{XRS}$ driven by device		600		μs
$t_{w(WDRS)}$	Pulse duration, reset pulse generated by watchdog		$512t_{c(OCK)}$		cycles
$t_{d(EX)}$	Delay time, address/data valid after $\overline{XRS}$ high		$32t_{c(OCK)}$		cycles
$t_{INTOSCST}$	Start up time, internal zero-pin oscillator		3		μs
t_{OSCST} (1)	On-chip crystal-oscillator start-up time	1	10		ms

(1) Dependent on crystal/resonator and board design.



A. After reset, the Boot ROM code samples BOOT Mode pins. Based on the status of the Boot Mode pin, the boot code branches to destination memory or boot code function. If Boot ROM code executes after power-on conditions (in debugger environment), the Boot code execution time is based on the current M3SSCLK speed. The M3SSCLK will be based on user environment and could be with or without PLL enabled.

Figure 5-3. Warm Reset

5.5.1 *Changing the Frequency of the Main PLL*

Figure 5-4 shows how to change the frequency of the Main PLL. The three steps are described below:

1. The PLL must first be placed in bypass mode (by writing to the SYSPLLCTL register) before any changes are made to the SPLLIMULT and SPLLFMULT fields of the SYSPLLMULT Register. Figure 5-4 shows that before being placed in bypass mode, the internal PLLSYSCLK clock was operating at 100 MHz. After entering the bypass mode, the PLLSYSCLK becomes 10 MHz, which is the frequency of OSCCLK, the input clock to the PLL
2. Once the PLL is placed in bypass mode, the SYSPLLMULT register can be modified to increase the PLLSYSCLK frequency to 150 MHz. See Figure 5-4 for the settings of the SPLLIMULT (integer) and SPLLFMULT (fractional) multiply fields of the SYSPLLMULT register for this step, and see Figure 2-8 for the functional description of the Main PLL. The PLL bypass mode must be maintained for at least 2000 OSCCLK cycles in order for the PLL to properly lock to the new frequency.
3. Finally, the SYSPLLCTL register is written to again, this time to take the PLL out of the bypass mode. Following this step, the PLLSYSCLK switches over from 10 MHz to the new frequency of 150 MHz.

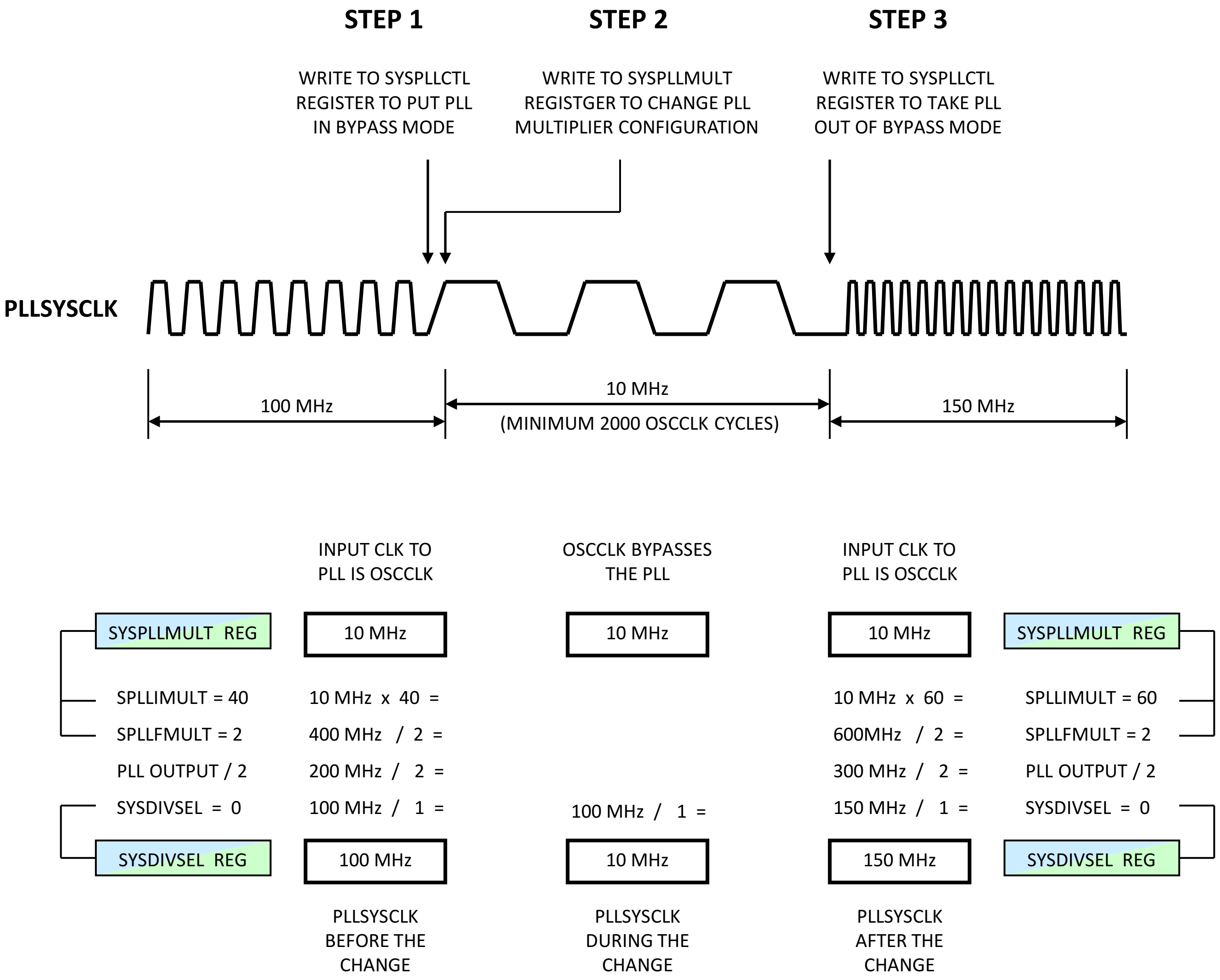


Figure 5-4. Changing the Frequency of the Main PLL

5.5.2 Power Management and Supervisory Circuit Solutions

Table 5-14 lists the power management and supervisory circuit solutions for F28M36x devices. LDO selection depends on the total power consumed in the end application. Go to www.ti.com and click on *Power Management* for a complete list of TI power ICs or select the *Power Management Selection Guide* link for specific power reference designs.

Table 5-14. Power Management and Supervisory Circuit Solutions

SUPPLIER	TYPE	PART	DESCRIPTION
Texas Instruments	DC/DC	TPS62160/170	1/0.5-A, 3–17-V input, step-down converter in 2x2 QFN package
Texas Instruments	DC/DC	TPS62140/150	2/1-A, 3–17-V input, step-down converter in 3x3 QFN package
Texas Instruments	LDO	TPS7A8001	Low-noise, high-bandwidth PSRR, 1A low-dropout linear regulator
Texas Instruments	LDO	TPS7A7001	2A, single-output, very-low input, adjustable low-dropout linear regulator
Texas Instruments	LDO/SVS	TPS75005	Dual, 500-mA, low-dropout regulators and triple-voltage rail monitor
Texas Instruments	DC/DC	LM22672/1	1/0.5-A, 4.5–42-V input SIMPLE SWITCHER, step-down voltage regulator with features
Texas Instruments	DC/DC	TPS54160/060	3.5-V to 60-V input, 1.5/0.5-A step-down converter with Eco-Mode
Texas Instruments	Module	LMZ10501	1A SIMPLE SWITCHER Nano Module with 5.5-V maximum input voltage
Texas Instruments	SVS	TPS386000/040	Quad supply voltage supervisors with programmable delay and watchdog timer
Texas Instruments	LDO	TPS73719	Single-output LDO, 1-A, fixed (1.9-V), reverse-current protection
Texas Instruments	LDO	TPS73534	Single-output LDO, 500-mA, fixed (3.4-V), low-quiescent current, low-noise, high PSRR

5.6 Flash Timing – Master Subsystem

Table 5-15. Flash/OTP Endurance for T Temperature Material(1)

		ERASE/PROGRAM TEMPERATURE	MIN	TYP	MAX	UNIT
N_f	Flash endurance for the array (write/erase cycles)	0°C to 105°C (ambient)	20000(2)	TBD		cycles
N_{OTP}	OTP endurance for the array (write cycles)	0°C to 105°C (ambient)			1	write

(1) Write/erase operations outside of the temperature ranges indicated are not specified and may affect the endurance numbers.
(2) Pending qualification completion.

Table 5-16. Flash/OTP Endurance for S Temperature Material(1)

		ERASE/PROGRAM TEMPERATURE	MIN	TYP	MAX	UNIT
N_f	Flash endurance for the array (write/erase cycles)	0°C to 125°C (ambient)	20000(2)	TBD		cycles
N_{OTP}	OTP endurance for the array (write cycles)	0°C to 125°C (ambient)			1	write

(1) Write/erase operations outside of the temperature ranges indicated are not specified and may affect the endurance numbers.
(2) Pending qualification completion.

Table 5-17. Flash/OTP Endurance for Q Temperature Material(1)

		ERASE/PROGRAM TEMPERATURE	MIN	TYP	MAX	UNIT
N_f	Flash endurance for the array (write/erase cycles)	–40°C to 125°C (ambient)	20000(2)	TBD		cycles
N_{OTP}	OTP endurance for the array (write cycles)	–40°C to 125°C (ambient)			1	write

(1) Write/erase operations outside of the temperature ranges indicated are not specified and may affect the endurance numbers.
(2) Pending qualification completion.

Table 5-18. Flash Parameters at 75 MHz

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Program Time(1)	128 bits			280	TBD	µs
	32K Sector			580	TBD	ms
	128K Sector			2320	TBD	ms
Erase Time(2)(3)	32K Sector			60	TBD	ms
	128K Sector			60	TBD	ms
I_{DDP} (4)	V_{DD} current consumption during Erase/Program cycle	VREG disabled		TBD		mA
I_{DDIOP} (4)	V_{DDIO} current consumption during Erase/Program cycle			TBD		
I_{DDIOP} (4)	V_{DDIO} current consumption during Erase/Program cycle	VREG enabled		TBD		mA

(1) Program time includes overhead of state machine but does not include data transfer time. Program time assumes programming 144 bits at a time. Program time includes Program verify by the CPU.
(2) The on-chip flash memory is in an erased state when the device is shipped from TI. As such, erasing the flash memory is not required prior to programming, when programming the device for the first time. However, the erase operation is needed on all subsequent programming operations.
(3) Erase time includes Erase verify by the CPU.
(4) Typical parameters as seen at room temperature including function call overhead, with all peripherals off.

Table 5-19. Flash Parameters at 125 MHz

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Program Time(1)	128 bits			240	TBD	μs
	32K Sector			500	TBD	ms
	128K Sector			2000	TBD	ms
Erase Time(2)(3)	32K Sector			50	TBD	ms
	128K Sector			50	TBD	ms
I_{DDP} (4)	V_{DD} current consumption during Erase/Program cycle	VREG disabled		TBD		mA
I_{DDIOP} (4)	V_{DDIO} current consumption during Erase/Program cycle			TBD		
I_{DDIOP} (4)	V_{DDIO} current consumption during Erase/Program cycle	VREG enabled		TBD		mA

(1) Program time includes overhead of state machine but does not include data transfer time. Program time assumes programming 144 bits at a time. Program time includes Program verify by the CPU.
(2) The on-chip flash memory is in an erased state when the device is shipped from TI. As such, erasing the flash memory is not required prior to programming, when programming the device for the first time. However, the erase operation is needed on all subsequent programming operations.
(3) Erase time includes Erase verify by the CPU.
(4) Typical parameters as seen at room temperature including function call overhead, with all peripherals off.

Table 5-20. Flash/OTP Access Timing(1)

PARAMETER		MIN	MAX	UNIT
$t_{a(f)}$	Flash access time	25		ns
$t_{a(OTP)}$	OTP access time	50		ns

(1) Access time numbers shown in this table are prior to device characterization. Final numbers will be published in the datasheet for the fully qualified production device.

Table 5-21. Flash Data Retention Duration

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{retention}$	Data retention duration	TBD	TBD		years

Table 5-22. Minimum Required Flash/OTP Wait-States at Different Frequencies

SYSCLKOUT (MHz)	SYSCLKOUT (ns)	WAIT-STATE
125	8	3
120	8.33	2
110	9.1	2
100	10	2
90	11.11	2
80	12.5	1
70	14.29	1
60	16.67	1
50	20	1
40	25	0
30	33.33	0
20	50	0
10	100	0

The equation to compute the Flash wait-state in Table 5-22 is as follows:

$$\text{RWAIT} = \left[\frac{\text{SYSCLK (MHz)}}{\text{40 (MHz)}} \right] - 1 \tag{1}$$

round up to the next integer, or 1, whichever is larger.

5.7 Flash Timing – Control Subsystem

Table 5-23. Flash/OTP Endurance for T Temperature Material(1)

		ERASE/PROGRAM TEMPERATURE	MIN	TYP	MAX	UNIT
N_f	Flash endurance for the array (write/erase cycles)	0°C to 105°C (ambient)	20000(2)	TBD		cycles
N_{OTP}	OTP endurance for the array (write cycles)	0°C to 105°C (ambient)			1	write

(1) Write/erase operations outside of the temperature ranges indicated are not specified and may affect the endurance numbers.
(2) Pending qualification completion.

Table 5-24. Flash/OTP Endurance for S Temperature Material(1)

		ERASE/PROGRAM TEMPERATURE	MIN	TYP	MAX	UNIT
N_f	Flash endurance for the array (write/erase cycles)	0°C to 125°C (ambient)	20000(2)	TBD		cycles
N_{OTP}	OTP endurance for the array (write cycles)	0°C to 125°C (ambient)			1	write

(1) Write/erase operations outside of the temperature ranges indicated are not specified and may affect the endurance numbers.
(2) Pending qualification completion.

Table 5-25. Flash/OTP Endurance for Q Temperature Material(1)

		ERASE/PROGRAM TEMPERATURE	MIN	TYP	MAX	UNIT
N_f	Flash endurance for the array (write/erase cycles)	–40°C to 125°C (ambient)	20000(2)	TBD		cycles
N_{OTP}	OTP endurance for the array (write cycles)	–40°C to 125°C (ambient)			1	write

(1) Write/erase operations outside of the temperature ranges indicated are not specified and may affect the endurance numbers.
(2) Pending qualification completion.

Table 5-26. Flash Parameters at 100 MHz

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Program Time(1)	128 bits			200	TBD	μs
	16K Sector			205	TBD	ms
	64K Sector			820	TBD	ms
Erase Time(2)(3)	16K Sector			50	TBD	ms
	64K Sector			50	TBD	ms
I_{DDP} (4)	V_{DD} current consumption during Erase/Program cycle	VREG disabled		TBD		mA
I_{DDIOP} (4)	V_{DDIO} current consumption during Erase/Program cycle			TBD		
I_{DDIOP} (4)	V_{DDIO} current consumption during Erase/Program cycle	VREG enabled		TBD		mA

(1) Program time includes overhead of state machine but does not include data transfer time. Program time assumes programming 144 bits at a time. Program time includes Program verify by the CPU.
(2) The on-chip flash memory is in an erased state when the device is shipped from TI. As such, erasing the flash memory is not required prior to programming, when programming the device for the first time. However, the erase operation is needed on all subsequent programming operations.
(3) Erase time includes Erase verify by the CPU.
(4) Typical parameters as seen at room temperature including function call overhead, with all peripherals off.

Table 5-27. Flash Parameters at 150 MHz

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Program Time(1)	128 bits			180	TBD	μs
	16K Sector			190	TBD	ms
	64K Sector			760	TBD	ms
Erase Time(2)(3)	16K Sector			40	TBD	ms
	64K Sector			40	TBD	ms
I_{DDP} (4)	V_{DD} current consumption during Erase/Program cycle	VREG disabled		TBD		mA
I_{DDIOP} (4)	V_{DDIO} current consumption during Erase/Program cycle			TBD		
I_{DDIOP} (4)	V_{DDIO} current consumption during Erase/Program cycle	VREG enabled		TBD		mA

(1) Program time includes overhead of state machine but does not include data transfer time. Program time assumes programming 144 bits at a time. Program time includes Program verify by the CPU.
(2) The on-chip flash memory is in an erased state when the device is shipped from TI. As such, erasing the flash memory is not required prior to programming, when programming the device for the first time. However, the erase operation is needed on all subsequent programming operations.
(3) Erase time includes Erase verify by the CPU.
(4) Typical parameters as seen at room temperature including function call overhead, with all peripherals off.

Table 5-28. Flash/OTP Access Timing(1)

PARAMETER		MIN	MAX	UNIT
$t_{a(f)}$	Flash access time	25		ns
$t_{a(OTP)}$	OTP access time	50		ns

(1) Access time numbers shown in this table are prior to device characterization. Final numbers will be published in the datasheet for the fully qualified production device.

Table 5-29. Flash Data Retention Duration

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{retention}$	Data retention duration	TBD	TBD		years

Table 5-30. Minimum Required Flash/OTP Wait-States at Different Frequencies

SYSCLKOUT (MHz)	SYSCLKOUT (ns)	WAIT-STATE
150	6.7	3
140	7.14	3
130	7.7	3
120	8.33	2
110	9.1	2
100	10	2
90	11.11	2
80	12.5	1
70	14.29	1
60	16.67	1
50	20	1
40	25	0
30	33.33	0
20	50	0
10	100	0

The equation to compute the Flash wait-state in Table 5-30 is as follows:

$$\text{RWAIT} = \left[\frac{\text{SYSCLK (MHz)}}{\text{40 (MHz)}} \right] - 1 \tag{2}$$

round up to the next integer, or 1, whichever is larger.

6 Peripheral Information and Timings

6.1 Analog and Shared Peripherals

Concerto Shared Peripherals are accessible from both the Master Subsystem and the Control Subsystem. The Analog Shared Peripherals include two 12-bit ADCs (Analog-to-Digital Converters), and six Comparator + DAC (10-bit) modules. The ADC Result Registers are accessible by CPUs and DMAs of the Master and Control Subsystems. All other analog registers, such as the ADC Configuration and Comparator Registers, are accessible by the C28x CPU only. The Digital Shared Peripherals include the Inter-Processor Communications (IPC) peripheral and the External Peripheral Interface (EPI). IPC is accessible by both CPUs; EPI is accessible by both CPUs and both DMAs.

IPC is used for sending and receiving synchronization events between Master and Control subsystems to coordinate execution of software running on both processors, or exchanging of data between the two processors. EPI is used by this device to communicate with external memory and other devices.

For detailed information on the processor peripherals, see the *Concerto F28M36x Technical Reference Manual* (literature number SPRUHE8).

6.1.1 Analog-to-Digital Converter (ADC)

Figure 6-1 shows the internal structure of each of the two ADC peripherals that are present on Concerto. Each ADC has 16 channels that can be programmed to select analog inputs, select start-of-conversion trigger, set the sampling window, and select end-of-conversion interrupt to prompt a CPU or DMA to read 16 result registers. The 16 ADC channels can be used independently or in pairs, based on the assignments inside the SAMPLEMODE register. Pairing up the channels allows two analog inputs to be sampled simultaneously—thereby, increasing the overall conversion performance.

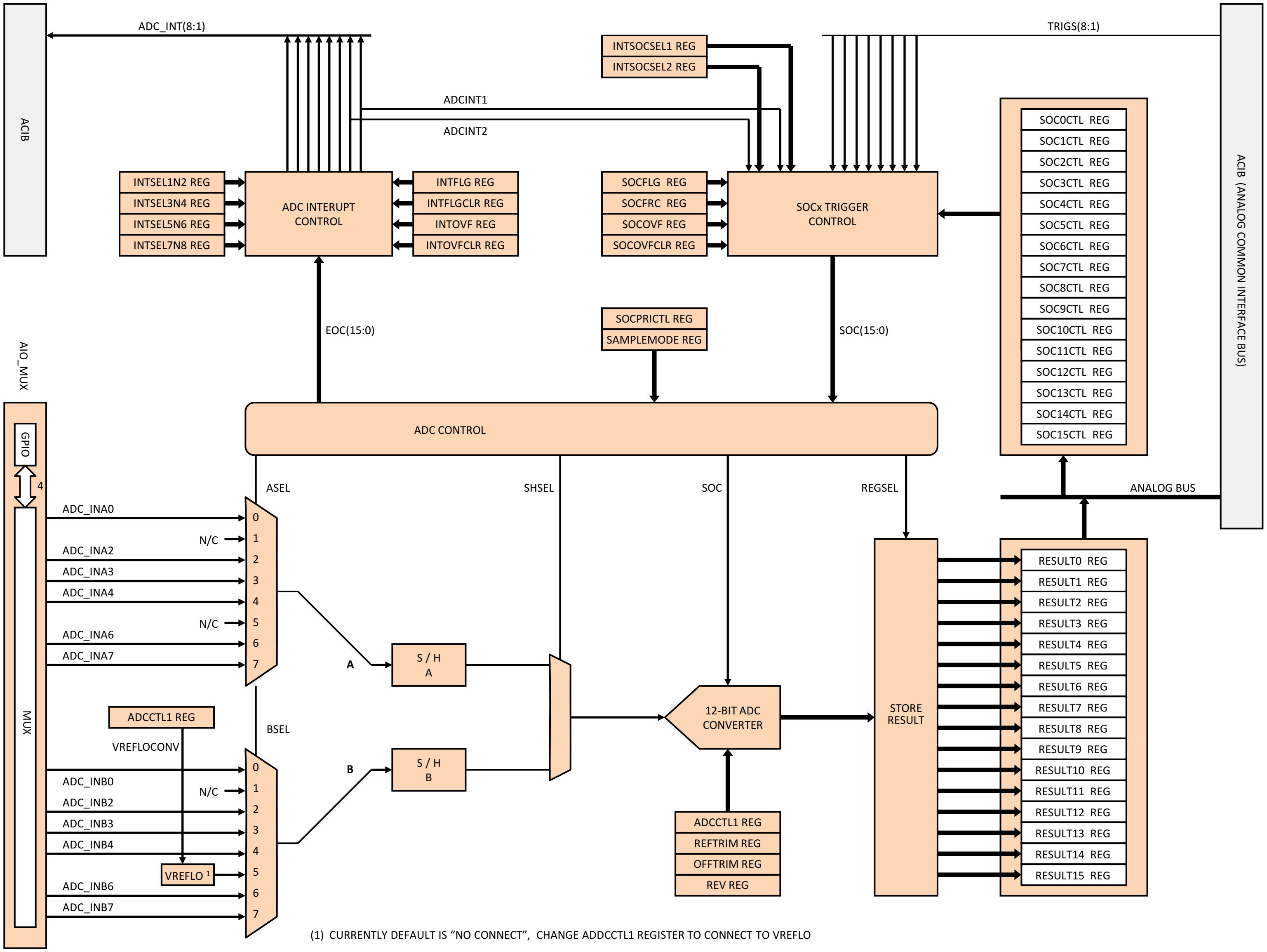
6.1.1.1 Sample Mode

Each ADC has 16 programmable channels that can be independently programmed for analog-to-digital conversion when corresponding bits in the SAMPLEMODE register are set to Sequential Mode. For example, if bit 2 in the SAMPLEMODE register is set to 0, ADC channels 4 and 5 are set to sequential mode. Both the SOC4CTL and SOC5CTL registers can then be programmed to configure channels 4 and 5 to independently perform analog-to-digital conversions with results being stored in the RESULT4 and RESULT5 registers. "Independently" means that channel 4 may use a different Start-Of-Conversion (SOC) trigger, different analog input, and different sampling window than the trigger, input, and window assigned to channel 5.

The 16 programmable channels for each ADC may also be grouped in 8 channel pairs when corresponding bits in the SAMPLEMODE register are set to Simultaneous Mode. For example, if bit 2 in the SAMPLEMODE register is set to 1, ADC channels 4 and 5 are set to Simultaneous Mode. The SOC4CTL register now contains configuration parameters for both channel 4 and channel 5, and the SOC5CTL register is ignored. While channel 4 and channel 5 are still using dedicated analog inputs (now selected as pairs in the CHSEL field of SOC4CTL), they both share the same SOC trigger and Sampling Window, with the results being stored in the RESULT4 and RESULT5 registers.

The Simultaneous mode is made possible by two sample-and-hold units present in each ADC. Each sample-and-hold unit has its own mux for selecting analog inputs (see Figure 6-1). By programming the SAMPLEMODE register, the 16 available channels can be configured as 16 independent channels, 8 channel pairs, or any combination thereof (for example, 10 sequential channels and 3 simultaneous pairs).

ADVANCE INFORMATION



(1) CURRENTLY DEFAULT IS "NO CONNECT", CHANGE ADDCCTL1 REGISTER TO CONNECT TO VREFLO

Figure 6-1. ADC

6.1.1.2 Start-of-Conversion (SOC) Triggers

There are eight external SOC triggers that go to each of the two ADC modules (from the Control Subsystem). In addition to the eight external SOC triggers, there are also two internal SOC triggers derived from End-Of-Conversion (EOC) interrupts inside each ADC module (ADCINT1 and ADCINT2). Registers INTSOCSEL1 and 2 are used to configure each of the 16 ADC channels for internal or external SOC sources. If internal SOC is chosen for a given channel, the INTSOCSEL1 and 2 registers also select whether the internal source is ADCINT1 or ADCINT2. If external SOC is chosen for a given ADC channel, the TRIGSEL field of the corresponding SOCxCTL register selects which of the eight external triggers is used for SOC in that channel. One analog-to-digital conversion can be performed at a time by the 12-bit ADC. The analog-to-digital conversion priority is managed according to the state of the PRICTL register.

6.1.1.3 Analog Inputs

Analog inputs to each of the two ADC modules are organized in two groups—A and B, with each group having a dedicated mux and sample-and-hold unit (see Figure 6-1). Mux A selects one of six possible analog inputs via AIO MUX. Mux B selects one of seven possible analog inputs—six external inputs via AIO MUX, and one from the internal VREFLO signal, which is currently tied to the Analog Ground. The Mux A and Mux B inputs can be simultaneously or sequentially sampled by the two sample-and-hold units according to the sampling window chosen in the SOCxCTL register for the corresponding channel.

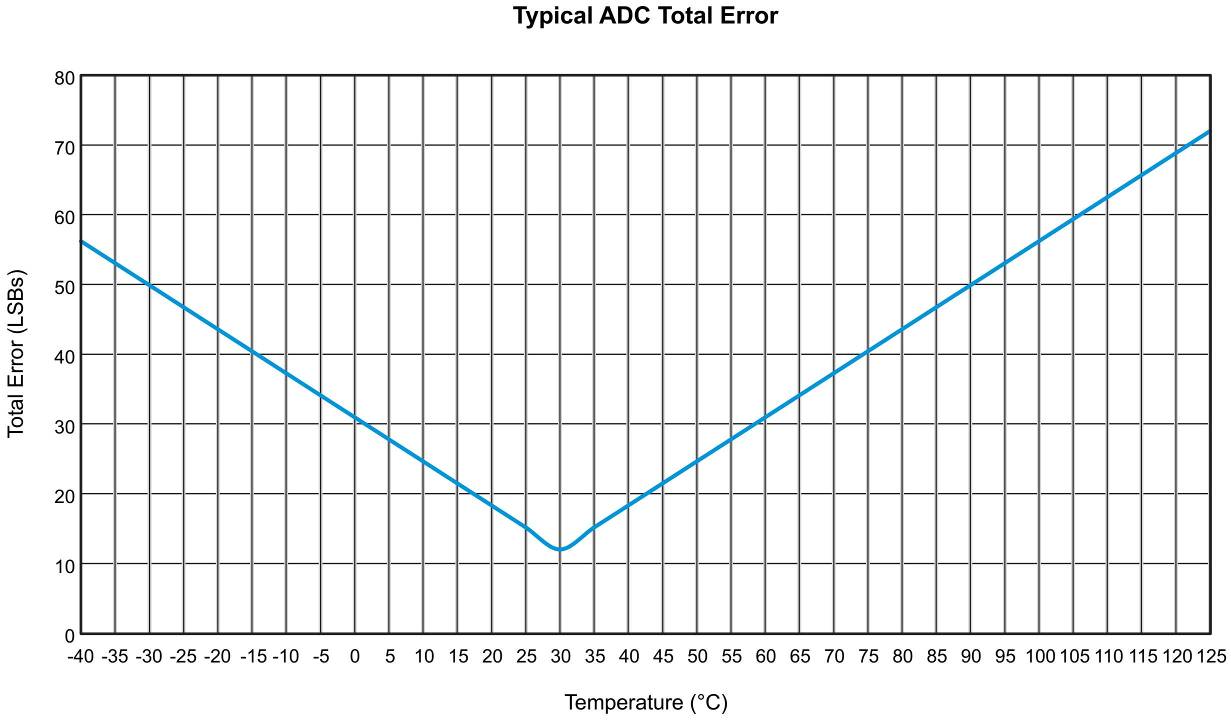
6.1.1.4 ADC Result Registers and EOC Interrupts

Concerto analog-to-digital conversion results are stored in 32 Results Registers (16 for ADC1 and 16 for ADC2). The 16 ADCx channels can be programmed via the INTSELxNy registers to trigger up to eight ADCINT interrupts per ADC module, when their results are ready to be read. The eight ADCINT interrupts from ADC1 and the eight ADCINT interrupts from ADC2 are AND-ed together before propagating to both the Master Subsystem and the Control Subsystem, announcing that the Result Registers are ready to be read by a CPU or DMA (see Figure 2-3).

6.1.1.5 ADC Electrical Data and Timing

Table 6-1. ADC Electrical Characteristics

PARAMETER		MIN	TYP	MAX	UNIT
DC SPECIFICATIONS					
Resolution			12		Bits
ADC clock		2		37.5	MHz
Sample Window		7		64	ADC Clocks
ACCURACY					
INL (Integral nonlinearity)		–4		4	LSB
DNL (Differential nonlinearity)		–1		1.5	LSB
Offset error	Executing a single self-recalibration	–20	0	20	LSB
	Executing periodic self-recalibration	–4	0	4	
Overall gain error with internal reference		–60		60	LSB
Overall gain error with external reference		–40		40	LSB
Channel-to-channel offset variation		–4		4	LSB
Channel-to-channel gain variation		–4		4	LSB
V_{REFLO} input current			–100		µA
V_{REFHI} input current			100		µA
ANALOG INPUT					
Analog input voltage with internal reference		0		3.3	V
Analog input voltage with external reference		V_{REFLO}		V_{REFHI}	V
V_{REFLO} input voltage		V_{SSA}		0.66	V
V_{REFHI} input voltage		2.64		V_{DDA}	V
Input capacitance			5		pF
Input leakage current			±2		µA
ADDITIONAL					
ADC SNR			65		dB
ADC SINAD			62		dB
ADC THD (50 kHz)			–65		dB
ENOB (SNR)			10.1		Bits
SFDR			66		dB



A. Gain error contribution is based on sampling of full-scale voltage using internal reference mode.

B. Periodic ADC offset re-cal is assumed.

C. Total error shown represents the absolute value of possible error.

Figure 6-2. Typical ADC Total Error [Temperature (°C) versus Total Error (LSBs)]

6.1.2 Comparator + DAC Units

Figure 6-3 shows the internal structure of the six analog Comparator + DAC units present in Concerto devices. Each unit compares two analog inputs (A and B) and assigns a value of '1' when the voltage of the A input is greater than that of the B input, or a value of '0' when the opposite is true. The six A inputs and six B inputs come from AIO_MUX1 and AIO_MUX2. All six B inputs can also be provided by the 10-bit digital-to-analog units that are present in each comparator DAC. The 10-bit value for each DAC unit is programmed in the respective DACVAL register. Another comparator register, COMPCTL, can be programmed to select the source of the B input, to enable or disable the comparator circuit, to invert comparator output, to synchronize comparator output to C28x SYSCLK, and to select the qualification period (number of clock cycles). All six output signals from the six comparators can be routed out to the device pins via GPIO_MUX2 pin mux.

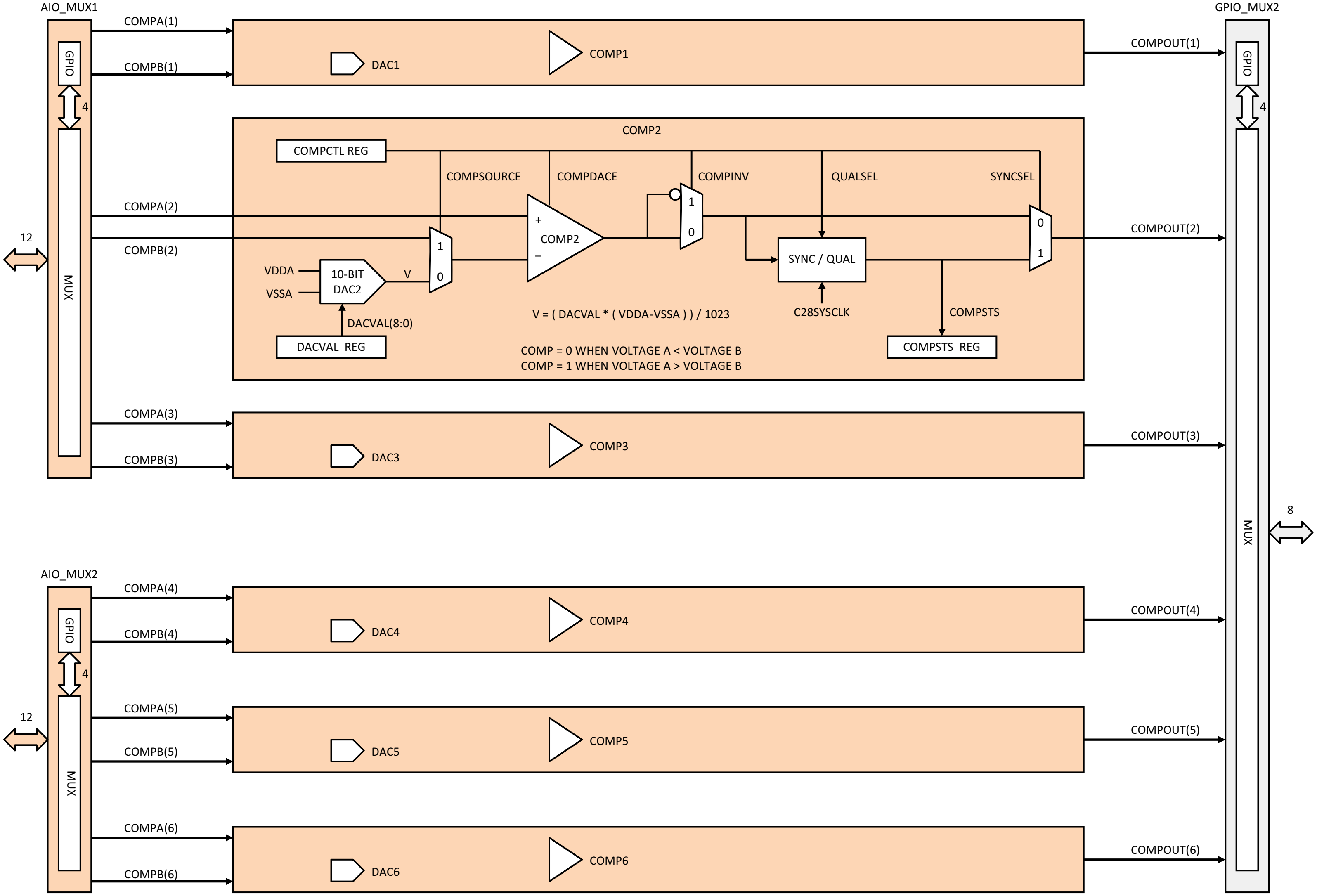


Figure 6-3. Comparator + DAC Units

6.1.2.1 On-Chip Comparator and DAC Electrical Data and Timing

Table 6-2. Electrical Characteristics of the Comparator/DAC

CHARACTERISTIC	MIN	TYP	MAX	UNITS
Comparator				
Comparator Input Range		V_{SSA} – V_{DDA}		V
Comparator response time to PWM Trip Zone (Async)		30		ns
Input Offset		±5		mV
Input Hysteresis(1)		35		mV
DAC				
DAC Output Range		V_{SSA} – V_{DDA}		V
DAC resolution		10		bits
DAC settling time		See Figure 6-4		
DAC Gain		–1.5		%
DAC Offset		10		mV
Monotonic		Yes		
INL		±3		LSB

(1) Hysteresis on the comparator inputs is achieved with a Schmidt trigger configuration. This results in an effective 100-kΩ feedback resistance between the output of the comparator and the non-inverting input of the comparator.

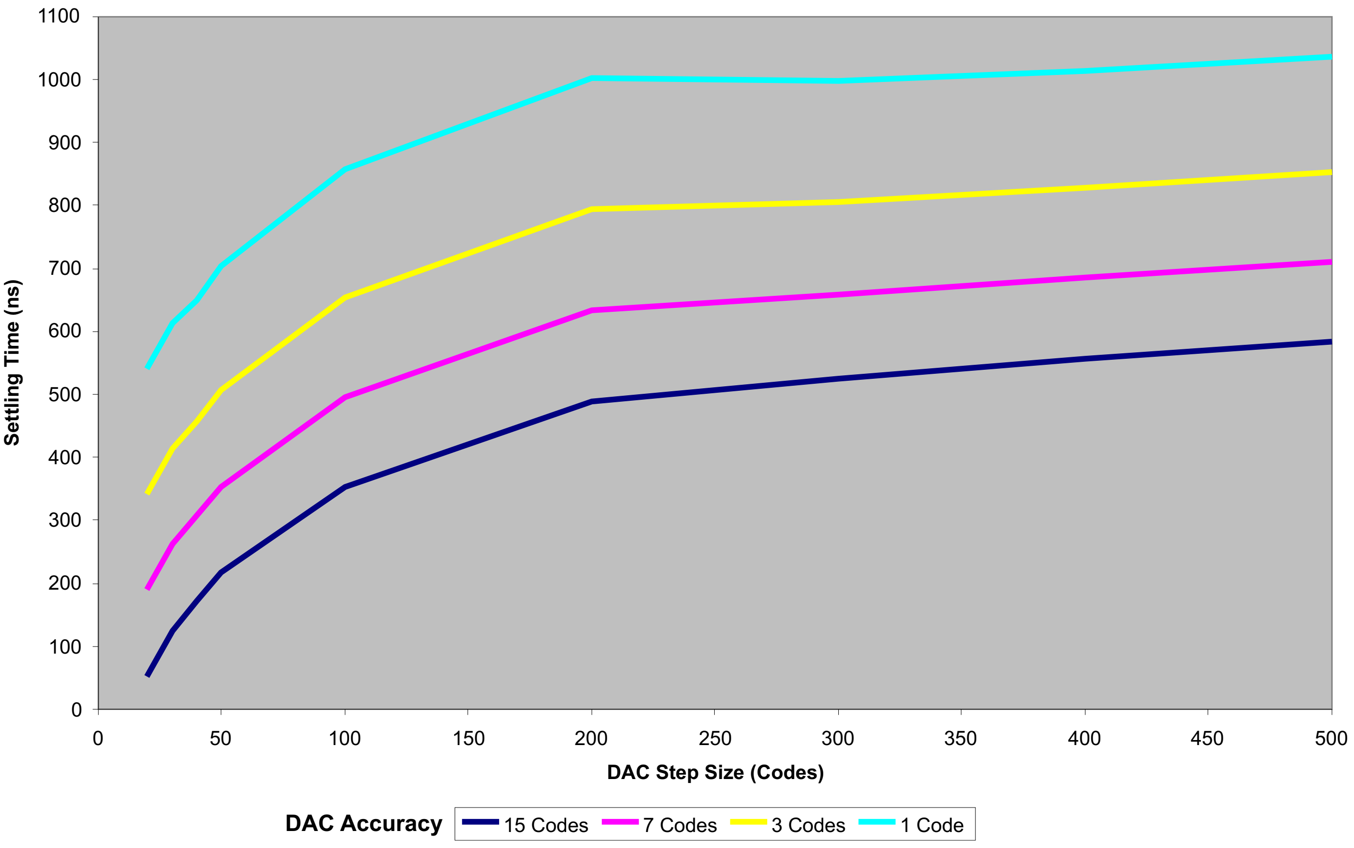


Figure 6-4. DAC Settling Time

6.1.3 Inter-Processor Communications (IPC)

Figure 6-5 shows the internal structure of the IPC peripheral used to synchronize program execution and exchange of data between the Cortex™-M3 and the C28x CPU. IPC can be used by itself when synchronizing program execution or it can be used in conjunction with Message RAMs when coordinating data transfers between processors. In either case, the operation of the IPC is the same. There are two independent sides to the IPC peripheral—MTOC (Master to Control) and CTOM (Control to Master).

The MTOC IPC is used by the Master Subsystem to send events to the Control Subsystem. The MTOC IPC typically sends events to the Control Subsystem by using the following registers: MTOCIPCSET, MTOCIPCFLG/MTOCIPCSTS (2), and MTOCIPCACK. Each of the 32 bits of these registers represents 32 independent channels through which the Cortex™-M3 CPU can send up to 32 events to the C28x CPU via software handshaking. Additionally, the first 4 bits of the MTOCIPC registers are supplemented with interrupts. To send an event via channel 2 from Cortex™-M3 to C28x, for example, the Cortex™-M3 and C28x CPUs use bit 2 of the MTOCIPCSET, MTOCIPCFLG/MTOCIPCSTS, MTOCIPCACK registers. The handshake starts with the Cortex™-M3 polling bit 2 of the MTOCIPCFLG register to make sure bit 2 is ‘0’. Next, the Cortex™-M3 writes a ‘1’ into bit 2 of the MTOCIPCSET register to start the handshake. In the mean time, the C28x is continually polling the MTOCIPCSTS register while waiting for the message. As soon as the Cortex™-M3 writes ‘1’ to bit 2 of the MTOCIPCSET register, bit 2 of MTOCIPCFLG/MTOCIPCSTS also turns ‘1’, thus announcing the event to the C28x. As soon as the C28x CPU reads a ‘1’ from the MTOCIPCSTS register, the C28x CPU should acknowledge by writing a ‘1’ to bit 2 of the MTOCIPCACK register, which in turn, clears bit 2 of the MTOCIPCFLG/MTOCIPCSTS register, enabling the Cortex™-M3 to send another message. Since the first four channels (bits 0, 1, 2, 3) are backed up by interrupts, both processors in the above example can use IPC interrupt 2 instead of polling to increase performance.

A similar handshake is also used when sending data (not just event) from the Master Subsystem to the Control Subsystem, but with two additional steps. Before setting a bit in the MTOCIPCSET register, the Cortex™-M3 should first load the MTOC Message RAM with a block of data that is to be made available to the C28x. In the second additional step, the C28x should read the data before setting a bit in the MTOCIPCACK register. This way, no data gets lost during multiple data transfers through a given block of the message RAM.

The CTOM IPC is used by the Control Subsystem to send events to the Master Subsystem. The CTOM IPC typically sends events to the Master Subsystem by using the following three registers: CTOMIPCSET, CTOMIPCFLG/CTOMIPCSTS, and CTOMIPCACK. The process is exactly the same as that for the MTOC IPC communication above.

---

(2) Note that physically MTOCIPCFLG/MTOCIPCSTS is one register, but it is referred to as the MTOCIPCFLG register when the Cortex™-M3 CPU reads it, and as the MTOCIPCSTS register when the C28x CPU reads it.

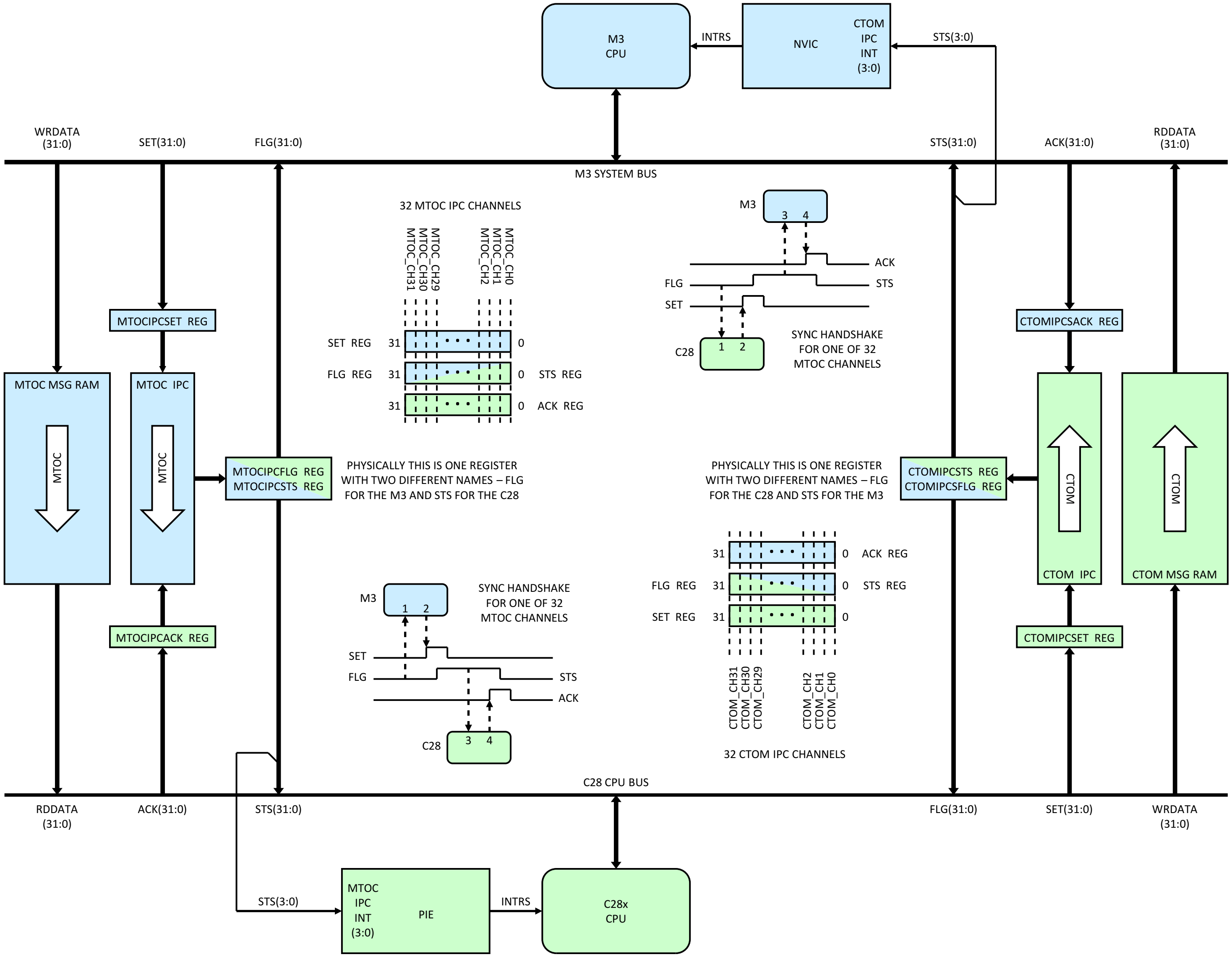


Figure 6-5. Interprocessor Communications (IPC)

6.1.4 External Peripheral Interface (EPI)

The External Peripheral Interface (EPI) provides a high-speed parallel bus for interfacing external peripherals and memory. EPI is accessible from both the Master Subsystem and the Control Subsystem. EPI has several modes of operation to enable glueless connectivity to most types of external devices. Some EPI modes of operation conform to standard microprocessor address/data bus protocols, while others are tailored to support a variety of fast custom interfaces, such as those communicating with field-programmable gate arrays (FPGAs) and complex programmable logic devices (CPLDs).

The EPI peripheral can be accessed by the Cortex™-M3 CPU, the Cortex™-M3 DMA, the C28x CPU, and the C28x DMA over the high-performance AHB bus. The Cortex™-M3 CPU and the µDMA drive AHB bus cycles directly through the Cortex™-M3 Bus Matrix. The C28x CPU and DMA also connect to the Cortex™-M3 Bus Matrix, but not directly. Before entering the Cortex™-M3 Bus Matrix, the native C28x CPU and DMA bus cycles are first converted to AHB protocol inside the MEM32-to-AHB Bus Bridge. After that, they pass through the Frequency Gasket to reduce the bus frequency by a factor of 2 or 4. Inside the Cortex™-M3 Bus Matrix, the Cortex™-M3 bus cycles may have to compete with C28x bus cycles for access to the AHB bus on the way to the EPI peripheral. See Figure 6-6 to see how EPI interfaces to the Concerto Master Subsystem, the Concerto Control Subsystem, Resets, Clocks, and Interrupts.

NOTE

The Control Subsystem has no direct access to EPI in silicon revision 0 devices.

Depending on how the Real-Time Window registers are configured inside the Bus Matrix, the arbitration between the Cortex™-M3 and C28x bus cycles is fixed-priority with Cortex™-M3 having higher priority than C28x, or the C28x having the option to own the Bus Matrix for a fixed period of time (window)—effectively stalling all Cortex™-M3 accesses during that time. Another EPI register inside the Cortex™-M3 Bus Matrix is the Memory Protection Register, which enables assignments of chip-select spaces to Cortex™-M3 or C28x EPI accesses (or both). The assignments of chip-select spaces prevent a bus cycle (from any processor) that does not own a given chip-select space, from getting through to EPI. The Real-time Window registers are the only EPI-related registers that are configurable by the C28x. The Memory Protection Register is configurable only by the Cortex™-M3 CPU, as are all configuration registers inside the EPI peripheral. Figure 6-6 shows the EPI registers and how they relate to individual blocks within the EPI.

Once a bus cycle arrives at the AHB bus interface inside the EPI peripheral, the bus cycle is routed to the General-Purpose Block, SDRAM Block, or the Host Bus Module, depending on the operating mode chosen through the EPI Configuration Register. Write cycles are buffered in a 4-word-deep Write FIFO; therefore, in most cases, the write cycles do not stall the CPU or DMA unless the Write FIFO becomes full. Read cycles can be handled in two different ways: blocking read cycles and non-blocking read cycles. Blocking read cycles are implemented when the content of a Read Data Register is 0. Blocking reads stall the CPU or DMA until the bus transaction completes. Non-blocking read cycles are triggered when a non-zero value is written into a Read Data Register. A non-zero value being written into a Read Data register triggers EPI to autonomously perform multiple data reads in the background (without involving CPU or DMA) according to values stored inside the Read Address Register and the Read Size Register. The incoming data is then temporarily stored in the Non-Blocking Read (NBR) FIFO until an EPI interrupt is generated to prompt the CPU or DMA to read the FIFO without risk of stalling. Furthermore, EPI has actually two sets of Data/Address/Size registers (set 0 and set 1) to enable ping-pong operation of non-blocking reads. In a ping-pong operation, while the previously fetched data is being read by the CPU or DMA from one end of the NBR FIFO, the next set of data words is simultaneously being deposited into the other end of the NBR FIFO.

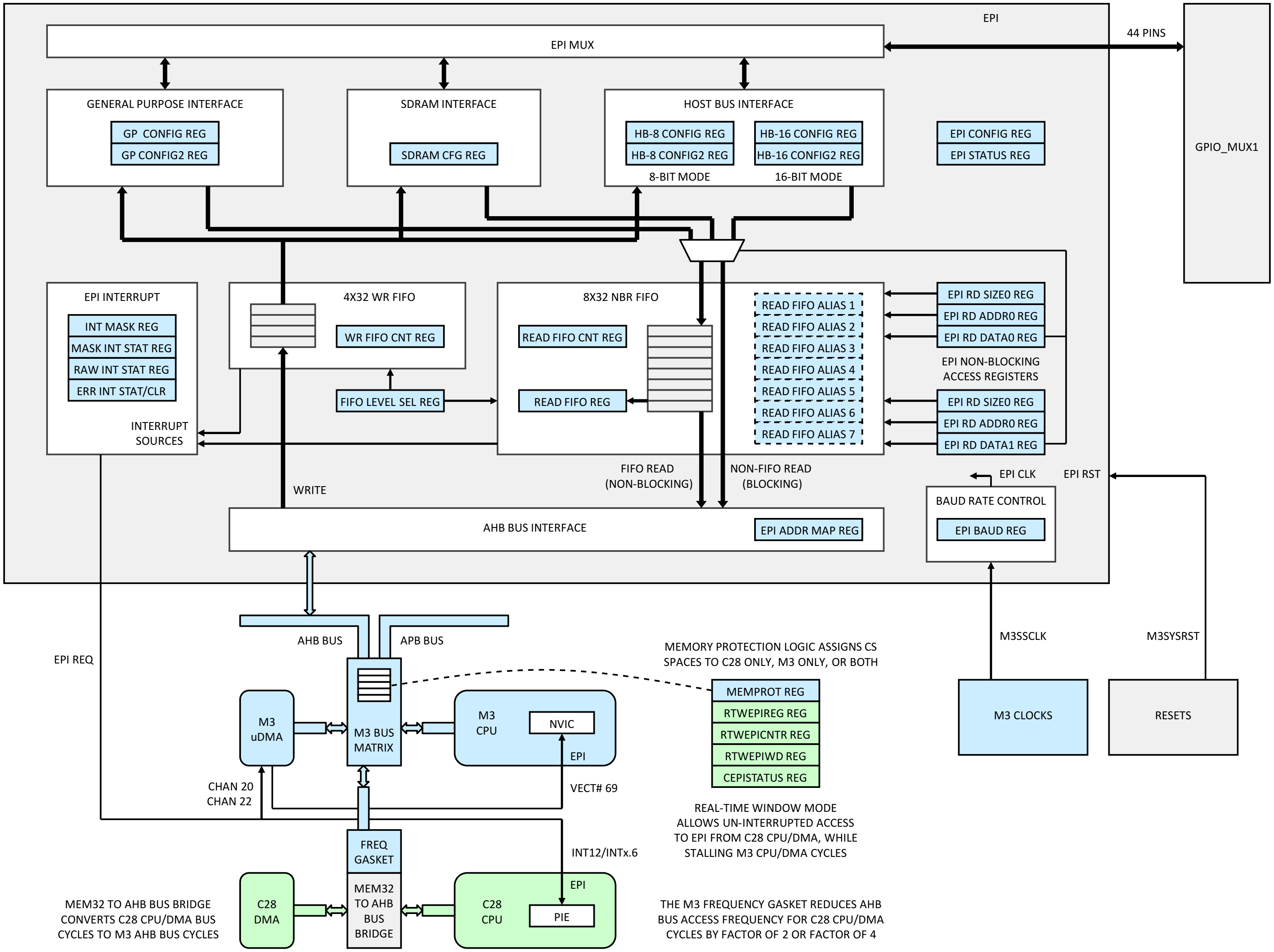


Figure 6-6. External Peripheral Interface (EPI)

EPI can directly interrupt the Cortex™-M3 CPU, the Cortex™-M3 uDMA, and the C28x CPU (but not the C28x DMA) via the EPI interrupt. Typically, EPI interrupts are used to prompt the CPU or DMA to move data to and from EPI. There are four EPI Interrupt registers that control various facets of interrupt generation, clearing, and masking. The EPI Interrupt can trigger µDMA to perform reads and writes through DMA Channels 20 and 22. If a CPU is the intended recipient, the Cortex™-M3 CPU is interrupted by Nested Vectored Interrupt Controller (NVIC) vector 69, and the C28x CPU is interrupted through the INT12/INTx6 vector to the PIE.

During EPI bus cycles, addresses entering the EPI module can propagate unchanged to the pins, or be remapped to different addresses according to values stored in the EPI Address Map Register in conjunction with the most significant bit of the incoming address.

The EPI's three primary operating modes are: the General-Purpose Mode, the SDRAM Mode, and the Host Bus Mode (including 8-bit and 16-bit versions).

6.1.4.1 EPI General-Purpose Mode

The EPI General-Purpose Mode is designed for high-speed clocked interfaces such as ones communicating with FPGAs and CPLDs. The high-speed clocked interfaces are different from the slower Host Bus interfaces, which have more relaxed timings that are compatible with established protocols like ones used to communicate with 8051 devices. Support of bus cycle framing and precisely controlled clocking are the additional features of the General-Purpose Mode that differentiate the General-Purpose Mode from the 8-bit and 16-bit Host Bus Modes.

Framing allows multiple bus transactions to be grouped together with an output signal called FRAME. The slave device responding to the bus cycles may use this signal to recognize related words of data and to speed up their transfers. The frame lengths are programmable and may vary from 1 to 30 clocks, depending on the clocking mode used.

Precise clocking is accomplished with a dedicated clock output pin (CLK). Devices responding the bus cycles can synchronize to CLK for faster transfers. The clock frequency can be precisely controlled through the Baud Rate Control block. This output clock can be gated or free-running. A gated approach uses a setup-time model in which the EPI clock controls when bus transactions are starting and stopping. A free-running EPI clock requires another method for determining when data is live, such as the frame pin or RD/WR strobes.

These and numerous other aspects of the General-Purpose Mode are controlled through the General-Purpose Configuration Register and the General-Purpose Configuration2 Register. The clocking for the General-Purpose Mode is configured through the EPI Baud Register of the EPI Baud Rate Control block.

See Figure 6-7 for a snapshot of the General-Purpose Mode registers, modes, and features. For more detailed maps of the General-Purpose Mode, see Table 6-3.

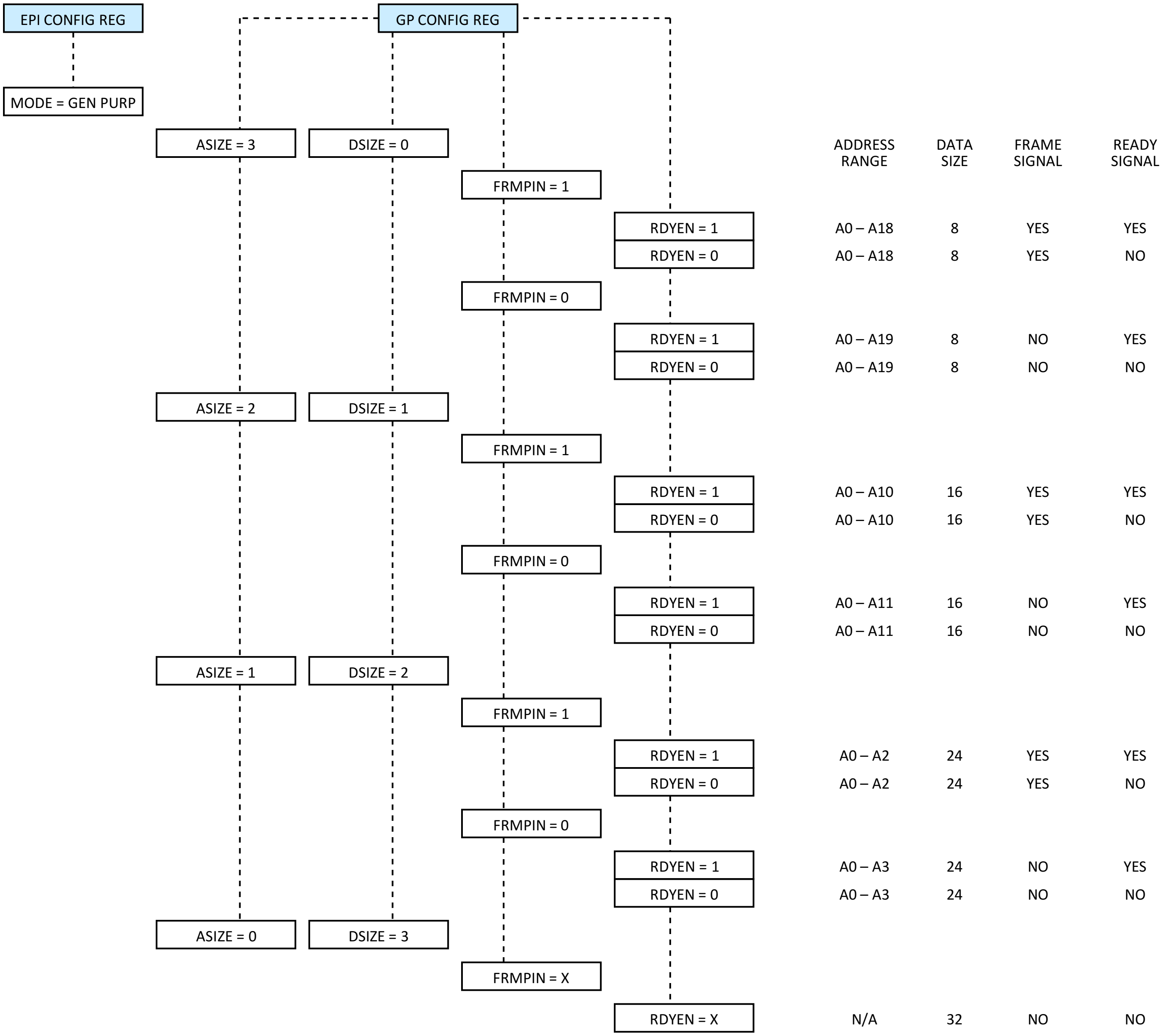


Figure 6-7. EPI General-Purpose Modes

Table 6-3. EPI MODES – General-Purpose Mode (EPICFG/MODE = 0x0)

EPI PORT NAME		EPI SIGNAL FUNCTION				DEVICE PIN	
Accessible by Cortex™-M3	Accessible by C28x	General-Purpose Signal (D8, A20)	General-Purpose Signal (D16, A12)	General-Purpose Signal (D24, A4)	General-Purpose Signal (D30, No Addr)	(Available GPIOMUX_1 Muxing Choices for EPI)	
EPI0S0		D0	D0	D0	D0	PH3_GPIO51	
EPI0S1		D1	D1	D1	D1	PH2_GPIO50	
EPI0S2		D2	D2	D2	D2	PC4_GPIO68	
EPI0S3		D3	D3	D3	D3	PC5_GPIO69	
EPI0S4		D4	D4	D4	D4	PC6_GPIO70	
EPI0S5		D5	D5	D5	D5	PC7_GPIO71	
EPI0S6		D6	D6	D6	D6	PH0_GPIO48	
EPI0S7		D7	D7	D7	D7	PH1_GPIO49	
EPI0S8		A0	D8	D8	D8	PE0_GPIO24	
EPI0S9		A1	D9	D9	D9	PE1_GPIO25	
EPI0S10		A2	D10	D10	D10	PH4_GPIO52	
EPI0S11		A3	D11	D11	D11	PH5_GPIO53	
EPI0S12		A4	D12	D12	D12	PF4_GPIO36	
EPI0S13		A5	D13	D13	D13	PG0_GPIO40	
EPI0S14		A6	D14	D14	D14	PG1_GPIO41	
EPI0S15		A7	D15	D15	D15	PF5_GPIO37	
EPI0S16		A8	A0	D16	D16	PJ0_GPIO56	
EPI0S17		A9	A1	D17	D17	PJ1_GPIO57	
EPI0S18		A10	A2	D18	D18	PJ2_GPIO58	
EPI0S19		A11	A3	D19	D19	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A12	A4	D29	D29	PD2_GPIO18	
EPI0S21		A13	A5	D21	D21	PD3_GPIO19	
EPI0S22		A14	A6	D22	D22	PB5_GPIO13	
EPI0S23		A15	A7	D23	D23	PB4_GPIO12	
EPI0S24		A16	A8	A0	D24	PE2_GPIO26	
EPI0S25		A17	A9	A1	D25	PE3_GPIO27	
EPI0S26		A18	A10	A2	D26	PH6_GPIO54	
EPI0S27		A19/RDY	A11/RDY	A3/RDY	D27	PH7_GPIO55	
EPI0S28		WR	WR	WR	D28	PD5_GPIO21	PJ4_GPIO60
EPI0S29		RD	RD	RD	D29	PD6_GPIO22	PJ5_GPIO61
EPI0S30		FRAME	FRAME	FRAME	D30	PD7_GPIO23	PJ6_GPIO62
EPI0S31		CLK	CLK	CLK	D31	PG7_GPIO47	
EPI0S32		x	x	x	x	PF2_GPIO34	PC0_GPIO64
EPI0S33		x	x	x	x	PF3_GPIO35	PC1_GPIO65
EPI0S34		x	x	x	x	PE4_GPIO28	
EPI0S35		x	x	x	x	PE5_GPIO29	
EPI0S36		x	x	x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	x	x	PG2_GPIO42	
EPI0S40		x	x	x	x	PG5_GPIO45	
EPI0S41		x	x	x	x	PG6_GPIO46	
EPI0S42		x	x	x	x	PN6_GPIO102	
EPI0S43		x	x	x	x	PN7_GPIO103	

6.1.4.2 EPI SDRAM Mode

The EPI SDRAM Mode combines high performance, low cost, and low pin utilization to access up to 512 megabits (Mb) of external memory. Main features of the EPI SDRAM interface are:

- Supports x16 (single data rate) SDRAM
- Supports low-cost SDRAMs up to 64 megabytes (MB) [or 512Mb]
- Includes automatic refresh and access to all banks, rows
- Includes Sleep/Standby Mode to keep contents active with minimal power drain
- Multiplexed address/data interface for reduced pin count

See Figure 6-8 for a snapshot of the SDRAM Mode registers and supported memory sizes. For more detailed maps of the SDRAM Mode, see Table 6-4.

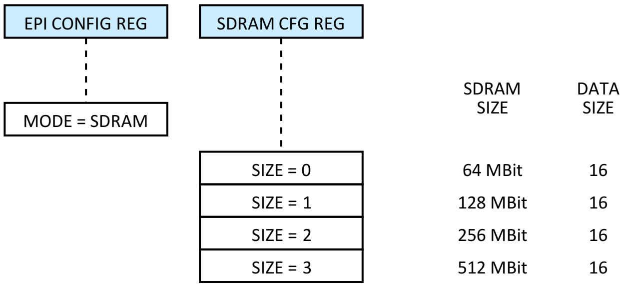


Figure 6-8. EPI SDRAM Mode

Table 6-4. EPI MODES – SDRAM Mode (EPICFG/MODE = 0x1)

EPI PORT NAME		EPI SIGNAL FUNCTION		DEVICE PIN	
Accessible by Cortex™-M3	Accessible by C28x	Column/Row Address	Data	(Available GPIOMUX_1 Muxing Choices for EPI)	
EPI0S0		A0	D0	PH3_GPIO51	
EPI0S1		A1	D1	PH2_GPIO50	
EPI0S2		A2	D2	PC4_GPIO68	
EPI0S3		A3	D3	PC5_GPIO69	
EPI0S4		A4	D4	PC6_GPIO70	
EPI0S5		A5	D5	PC7_GPIO71	
EPI0S6		A6	D6	PH0_GPIO48	
EPI0S7		A7	D7	PH1_GPIO49	
EPI0S8		A8	D8	PE0_GPIO24	
EPI0S9		A9	D9	PE1_GPIO25	
EPI0S10		A10	D10	PH4_GPIO52	
EPI0S11		A11	D11	PH5_GPIO53	
EPI0S12		A12	D12	PF4_GPIO36	
EPI0S13		BA0	D13	PG0_GPIO40	
EPI0S14		BA1	D14	PG1_GPIO41	
EPI0S15		D15		PF5_GPIO37	
EPI0S16		DQML		PJ0_GPIO56	
EPI0S17		DQMH		PJ1_GPIO57	
EPI0S18		$\overline{\text{CAS}}$		PJ2_GPIO58	
EPI0S19		$\overline{\text{RAS}}$		PD4_GPIO20	PJ3_GPIO59
EPI0S28		$\overline{\text{WE}}$		PD5_GPIO21	PJ4_GPIO60
EPI0S29		$\overline{\text{CS}}$		PD6_GPIO22	PJ5_GPIO61
EPI0S30		CKE		PD7_GPIO23	PJ6_GPIO62
EPI0S31		CLK		PG7_GPIO47	
EPI0S20		x		PD2_GPIO18	
EPI0S21		x		PD3_GPIO19	
EPI0S22		x		PB5_GPIO13	
EPI0S23		x		PB4_GPIO12	
EPI0S24		x		PE2_GPIO26	
EPI0S25		x		PE3_GPIO27	
EPI0S26		x		PH6_GPIO54	
EPI0S27		x		PH7_GPIO55	
EPI0S32		x		PF2_GPIO34	PC0_GPIO64
EPI0S33		x		PF3_GPIO35	PC1_GPIO65
EPI0S34		x		PE4_GPIO28	
EPI0S35		x		PE5_GPIO29	
EPI0S36		x		PB7_GPIO15	PC3_GPIO67
EPI0S37		x		PB6_GPIO14	PC2_GPIO66
EPI0S38		x		PF6_GPIO38	PE4_GPIO28
EPI0S39		x		PG2_GPIO42	
EPI0S40		x		PG5_GPIO45	
EPI0S41		x		PG6_GPIO46	
EPI0S42		x		PN6_GPIO102	
EPI0S43		x		PN7_GPIO103	

6.1.4.3 EPI Host Bus Mode

There are two versions of the EPI Host Bus Mode: an 8-bit version (HB-8) and a 16-bit version (HB-16). Section 6.1.4.3.1 discusses the EPI 8-Bit Host Bus Mode. Section 6.1.4.3.2 discusses the EPI 16-Bit Host Bus Mode.

6.1.4.3.1 EPI 8-Bit Host Bus (HB-8) Mode

The 8-Bit Host Bus (HB-8) Mode uses fewer data pins than the 16-Bit Host Bus (HB-16) Mode; hence, more pins are available for address. The HB-8 Mode is also slower than the General-Purpose Mode in order to accommodate older logic. The HB-8 Mode is selected with the MODE field of EPI Configuration Register. Within the HB-8 Mode, two additional registers are used to select address/data muxing, chip selects, and other options. These registers are the HB-8 Configuration Register and the HB-8 Configuration2 Register. See Figure 6-9 for a snapshot of HB-8 registers, modes, and features.

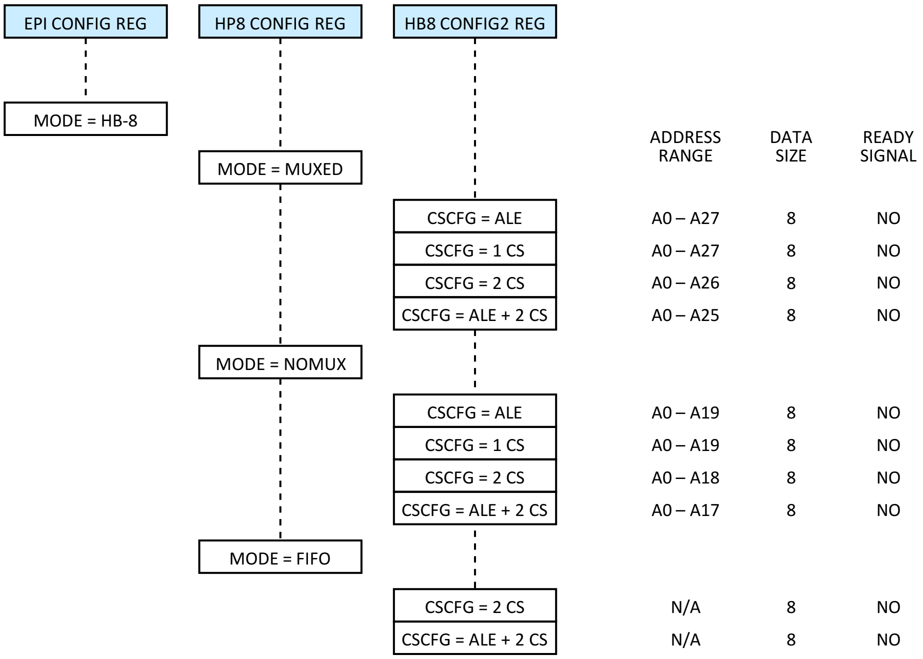


Figure 6-9. EPI 8-Bit Host Bus Mode

6.1.4.3.1.1 HB-8 Muxed Address/Data Mode

The HB-8 Muxed Mode multiplexes address signals with low-order data signals. For this reason, the Muxed Mode allows for a larger address space as compared to the Non-Muxed Mode. The HB-8 Muxed Mode is selected with the MODE field of the HB-8 Configuration Register. In addition to data and address signals, the HB-8 Muxed Mode also features the ALE signal (indicating to an external latch to capture address and hold the address until the data phase); RD and WR data strobes; and 1–4 Chip Select (CS) signals to enable one of four external peripherals. The ALE and CS options are chosen with the CSCFG field of the HB-8 Configuration2 Register. For more detailed maps of the HB-8 Muxed Mode, see Table 6-5.

Table 6-5. EPI MODES – 8-Bit Host-Bus Mode (EPICFG/MODE = 0x2), Muxed (EPIHB16CFG/MODE = 0x0)

EPI PORT NAME		EPI SIGNAL FUNCTION				DEVICE PIN	
Accessible by Cortex™-M3	**Accessible by C28x**	**With Address Latch Enable (CSCFG = 0x0)**	**With One Chip Select (CSCFG = 0x1)**	**With Two Chip Selects (CSCFG = 0x2)**	**With ALE and Two Chip Selects (CSCFG = 0x3)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S0		AD0	AD0	AD0	AD0	PH3_GPIO51	
EPI0S1		AD1	AD1	AD1	AD1	PH2_GPIO50	
EPI0S2		AD2	AD2	AD2	AD2	PC4_GPIO68	
EPI0S3		AD3	AD3	AD3	AD3	PC5_GPIO69	
EPI0S4		AD4	AD4	AD4	AD4	PC6_GPIO70	
EPI0S5		AD5	AD5	AD5	AD5	PC7_GPIO71	
EPI0S6		AD6	AD6	AD6	AD6	PH0_GPIO48	
EPI0S7		AD7	AD7	AD7	AD7	PH1_GPIO49	
EPI0S8		A8	A8	A8	A8	PE0_GPIO24	
EPI0S9		A9	A9	A9	A9	PE1_GPIO25	
EPI0S10		A10	A10	A10	A10	PH4_GPIO52	
EPI0S11		A11	A11	A11	A11	PH5_GPIO53	
EPI0S12		A12	A12	A12	A12	PF4_GPIO36	
EPI0S13		A13	A13	A13	A13	PG0_GPIO40	
EPI0S14		A14	A14	A14	A14	PG1_GPIO41	
EPI0S15		A15	A15	A15	A15	PF5_GPIO37	
EPI0S16		A16	A16	A16	A16	PJ0_GPIO56	
EPI0S17		A17	A17	A17	A17	PJ1_GPIO57	
EPI0S18		A18	A18	A18	A18	PJ2_GPIO58	
EPI0S19		A19	A19	A19	A19	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A20	A20	A20	A20	PD2_GPIO18	
EPI0S21		A21	A21	A21	A21	PD3_GPIO19	
EPI0S22		A22	A22	A22	A22	PB5_GPIO13	
EPI0S23		A23	A23	A23	A23	PB4_GPIO12	
EPI0S24		A24	A24	A24	A24	PE2_GPIO26	
EPI0S25		A25	A25	A25	A25	PE3_GPIO27	
EPI0S26		A26	A26	A26	$\overline{CS0}$	PH6_GPIO54	
EPI0S27		A27	A27	$\overline{CS1}$	$\overline{CS1}$	PH7_GPIO55	
EPI0S30		ALE	$\overline{CS0}$	$\overline{CS0}$	ALE	PD7_GPIO23	PJ6_GPIO62
EPI0S29		WR	WR	WR	WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	RD	RD	RD	PD5_GPIO21	PJ4_GPIO60
EPI0S31		x	x	x	x	PG7_GPIO47	
EPI0S32		x	x	x	x	PF2_GPIO34	PC0_GPIO64
EPI0S33		x	x	x	x	PF3_GPIO35	PC1_GPIO65
EPI0S34		x	x	x	x	PE4_GPIO28	
EPI0S35		x	x	x	x	PE5_GPIO29	
EPI0S36		x	x	x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	x	x	PG2_GPIO42	
EPI0S40		x	x	x	x	PG5_GPIO45	
EPI0S41		x	x	x	x	PG6_GPIO46	
EPI0S42		x	x	x	x	PN6_GPIO102	
EPI0S43		x	x	x	x	PN7_GPIO103	

6.1.4.3.1.2 HB-8 Non-Muxed Address/Data Mode

The HB-8 Non-Muxed Mode uses dedicated pins for address and data signals. For this reason, the Non-Muxed Mode has reduced address reach as compared to the Muxed Mode. The HB-8 Non-Muxed Mode is selected with the MODE field of the HB-8 Configuration Register. In addition to data and address signals, the HB-8 Non-Muxed Mode also features the ALE signal (indicating to an external latch to capture address and hold the address until the data phase); RD and WR data strobes; and 1–4 Chip Select (CS) signals to enable one of four external peripherals. The ALE and CS options are chosen with the CSCFG field of the HB-8 Configuration2 Register. For more detailed maps of the HB-8 Non-Muxed Mode, see Table 6-6.

Table 6-6. EPI MODES – 8-Bit Host-Bus Mode (EPICFG/MODE = 0x2), Non-Muxed (EPIHB16CFG/MODE = 0x1)

EPI PORT NAME		EPI SIGNAL FUNCTION				DEVICE PIN
Accessible by Cortex™-M3	Accessible by C28x	With Address Latch Enable (CSCFG = 0x0)	With One Chip Select (CSCFG = 0x1)	With Two Chip Selects (CSCFG = 0x2)	With ALE and Two Chip Selects (CSCFG = 0x3)	(Available GPIOMUX_1 Muxing Choices for EPI)
EPI0S0		D0	D0	D0	D0	PH3_GPIO51
EPI0S1		D1	D1	D1	D1	PH2_GPIO50
EPI0S2		D2	D2	D2	D2	PC4_GPIO68
EPI0S3		D3	D3	D3	D3	PC5_GPIO69
EPI0S4		D4	D4	D4	D4	PC6_GPIO70
EPI0S5		D5	D5	D5	D5	PC7_GPIO71
EPI0S6		D6	D6	D6	D6	PH0_GPIO48
EPI0S7		D7	D7	D7	D7	PH1_GPIO49
EPI0S8		A0	A0	A0	A0	PE0_GPIO24
EPI0S9		A1	A1	A1	A1	PE1_GPIO25
EPI0S10		A2	A2	A2	A2	PH4_GPIO52
EPI0S11		A3	A3	A3	A3	PH5_GPIO53
EPI0S12		A4	A4	A4	A4	PF4_GPIO36
EPI0S13		A5	A5	A5	A5	PG0_GPIO40
EPI0S14		A6	A6	A6	A6	PG1_GPIO41
EPI0S15		A7	A7	A7	A7	PF5_GPIO37
EPI0S16		A8	A8	A8	A8	PJ0_GPIO56
EPI0S17		A9	A9	A9	A9	PJ1_GPIO57
EPI0S18		A10	A10	A10	A10	PJ2_GPIO58
EPI0S19		A11	A11	A11	A11	PD4_GPIO20 PJ3_GPIO59
EPI0S20		A12	A12	A12	A12	PD2_GPIO18
EPI0S21		A13	A13	A13	A13	PD3_GPIO19
EPI0S22		A14	A14	A14	A14	PB5_GPIO13
EPI0S23		A15	A15	A15	A15	PB4_GPIO12
EPI0S24		A16	A16	A16	A16	PE2_GPIO26
EPI0S25		A17	A17	A17	A17	PE3_GPIO27
EPI0S26		A18	A18	A18	$\overline{CS0}$	PH6_GPIO54
EPI0S27		A19	A19	$\overline{CS1}$	$\overline{CS1}$	PH7_GPIO55
EPI0S30		ALE	$\overline{CS0}$	$\overline{CS0}$	ALE	PD7_GPIO23 PJ6_GPIO62
EPI0S29		WR	WR	WR	WR	PD6_GPIO22 PJ5_GPIO61
EPI0S28		RD	RD	RD	RD	PD5_GPIO21 PJ4_GPIO60
EPI0S31		x	x	x	x	PG7_GPIO47
EPI0S32		x	x	x	x	PF2_GPIO34 PC0_GPIO64
EPI0S33		x	x	x	x	PF3_GPIO35 PC1_GPIO65

Table 6-6. EPI MODES – 8-Bit Host-Bus Mode (EPICFG/MODE = 0x2), Non-Muxed (EPIHB16CFG/MODE = 0x1) (continued)

EPI PORT NAME		**EPI SIGNAL FUNCTION**				**DEVICE PIN**	
Accessible by Cortex™-M3	**Accessible by C28x**	**With Address Latch Enable (CSCFG = 0x0)**	**With One Chip Select (CSCFG = 0x1)**	**With Two Chip Selects (CSCFG = 0x2)**	**With ALE and Two Chip Selects (CSCFG = 0x3)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S34		x	x	x	x	PE4_GPIO28	
EPI0S35		x	x	x	x	PE5_GPIO29	
EPI0S36		x	x	x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	x	x	PG2_GPIO42	
EPI0S40		x	x	x	x	PG5_GPIO45	
EPI0S41		x	x	x	x	PG6_GPIO46	
EPI0S42		x	x	x	x	PN6_GPIO102	
EPI0S43		x	x	x	x	PN7_GPIO103	

6.1.4.3.1.3 HB-8 FIFO Mode

The HB-8 FIFO Mode uses 8 bits of data, removes ALE and address pins, and optionally adds external FIFO Full/Empty flag inputs. This scheme is used by many devices, such as radios, communication devices (including USB2 devices), and some FPGA configuration (FIFO throughblock RAM). This FIFO Mode presents the data side of the normal Host-Bus interface, but is paced by FIFO control signals. It is important to consider that the FIFO Full/Empty control inputs may stall the EPI interface and can potentially block other CPU or DMA accesses. For more detailed maps of the HB-8 FIFO Mode, see Table 6-7.

Table 6-7. EPI MODES – 8-Bit Host-Bus Mode (EPICFG/MODE = 0x2), FIFO Mode (EPIHB16CFG/MODE = 0x3)

EPI PORT NAME		EPI SIGNAL FUNCTION		DEVICE PIN	
Accessible by Cortex™-M3	**Accessible by C28x**	**With One Chip Select (CSCFG = 0x1)**	**With Two Chip Selects (CSCFG = 0x2)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S0		D0	D0	PH3_GPIO51	
EPI0S1		D1	D1	PH2_GPIO50	
EPI0S2		D2	D2	PC4_GPIO68	
EPI0S3		D3	D3	PC5_GPIO69	
EPI0S4		D4	D4	PC6_GPIO70	
EPI0S5		D5	D5	PC7_GPIO71	
EPI0S6		D6	D6	PH0_GPIO48	
EPI0S7		D7	D7	PH1_GPIO49	
EPI0S25		x	$\overline{\text{CS1}}$	PE3_GPIO27	
EPI0S30		$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	PD7_GPIO23	PJ6_GPIO62
EPI0S27		FFULL	FFULL	PH7_GPIO55	
EPI0S26		FEMPTY	FEMPTY	PH6_GPIO54	
EPI0S29		WR	WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	RD	PD5_GPIO21	PJ4_GPIO60
EPI0S8		x	x	PE0_GPIO24	
EPI0S9		x	x	PE1_GPIO25	
EPI0S10		x	x	PH4_GPIO52	
EPI0S11		x	x	PH5_GPIO53	
EPI0S12		x	x	PF4_GPIO36	
EPI0S13		x	x	PG0_GPIO40	
EPI0S14		x	x	PG1_GPIO41	
EPI0S15		x	x	PF5_GPIO37	
EPI0S16		x	x	PJ0_GPIO56	
EPI0S17		x	x	PJ1_GPIO57	
EPI0S18		x	x	PJ2_GPIO58	
EPI0S19		x	x	PD4_GPIO20	PJ3_GPIO59
EPI0S20		x	x	PD2_GPIO18	
EPI0S21		x	x	PD3_GPIO19	
EPI0S22		x	x	PB5_GPIO13	
EPI0S23		x	x	PB4_GPIO12	
EPI0S24		x	x	PE2_GPIO26	
EPI0S32		x	x	PF2_GPIO34	PC0_GPIO64
EPI0S31		x	x	PG7_GPIO47	
EPI0S33		x	x	PF3_GPIO35	PC1_GPIO65
EPI0S34		x	x	PE4_GPIO28	
EPI0S35		x	x	PE5_GPIO29	
EPI0S36		x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	PG2_GPIO42	
EPI0S40		x	x	PG5_GPIO45	
EPI0S41		x	x	PG6_GPIO46	
EPI0S42		x	x	PN6_GPIO102	
EPI0S43		x	x	PN7_GPIO103	

6.1.4.3.2 EPI 16-Bit Host Bus (HB-16) Mode

The 16-Bit Host Bus (HB-16) Mode uses fewer address pins than the 8-Bit Host Bus (HB-8) Mode; hence, more pins are available for data. The HB-16 Mode is also slower than the General-Purpose Mode in order to accommodate older logic. The HB-16 Mode is selected with the MODE field of EPI Configuration Register. Within the HB-16 Mode, two additional registers are used to select address/data muxing, byte selects, chip selects, and other options. These registers are the HB-16 Configuration Register and the HB-16 Configuration2 Register. See Figure 6-10 for a snapshot of HB-16 registers, modes, and features.

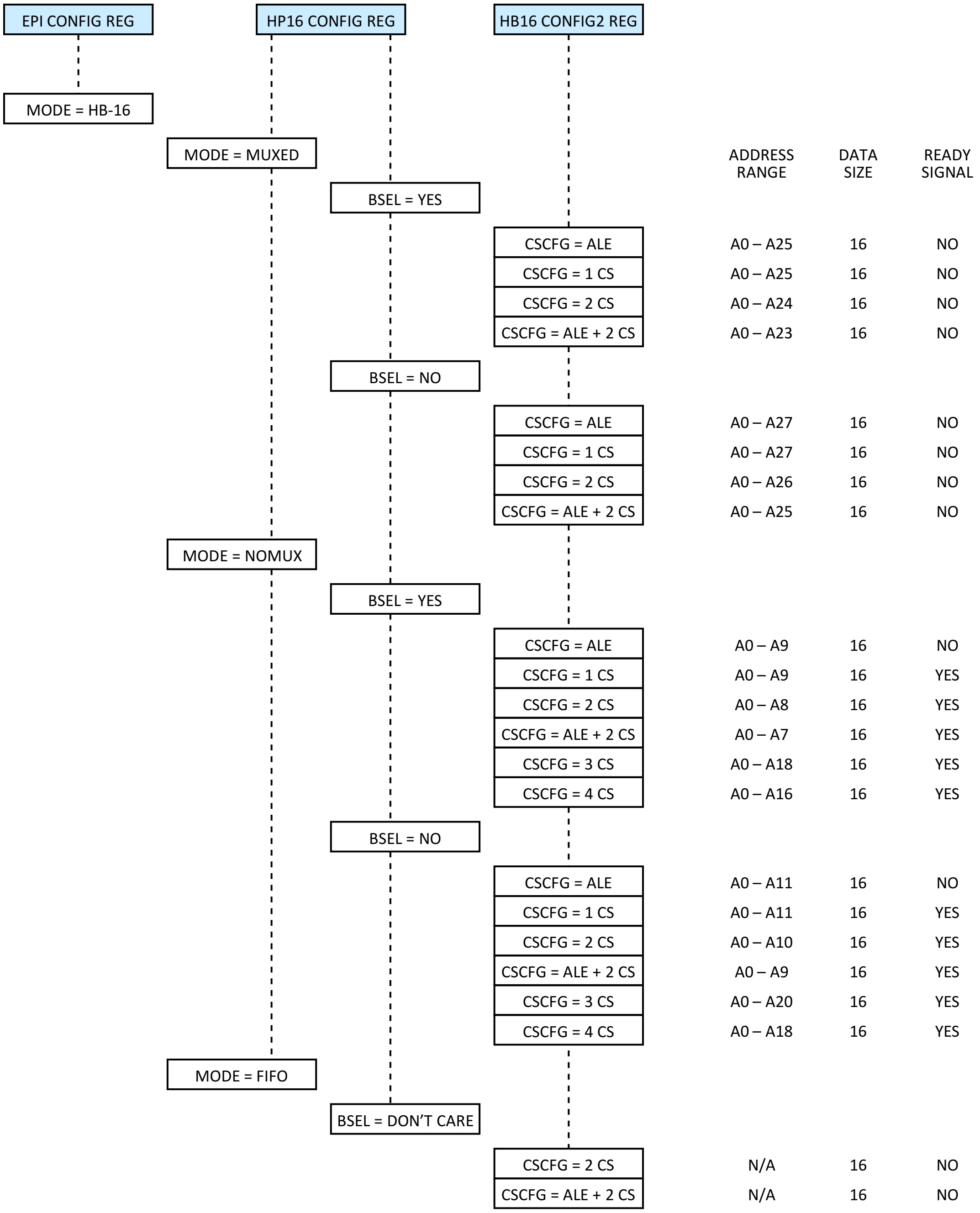


Figure 6-10. EPI 16-Bit Host Bus Mode

6.1.4.3.2.1 *HB-16 Muxed Address/Data Mode*

The HB-16 Muxed Mode multiplexes address signals with low-order data signals. For this reason, the Muxed Mode allows for a larger address space as compared to the Non-Muxed Mode. The HB-16 Muxed Mode is selected with the MODE field of the HB-16 Configuration Register. In addition to data and address signals, the HB-16 Muxed Mode also features the ALE signal (indicating to an external latch to capture address and hold the address until the data phase); RD and WR data strobes; 1–4 Chip Select (CS) signals to enable one of four external peripherals; and two Byte Select (BSEL) signals to accommodate byte accesses to lower or upper half of 16-bit data. The Byte Selects are chosen with the BSEL field of the HB-16 Configuration Register. The ALE and CS options are chosen with the CSCFG field of the HB-16 Configuration2 Register. For more detailed maps of the HB-16 Muxed Mode ***without*** Byte Selects, see Table 6-8. For more detailed maps of the HB-16 Muxed Mode ***with*** Byte Selects, see Table 6-9.

Table 6-8. EPI MODES – 16-Bit Host-Bus Mode (EPICFG/MODE = 0x3), *Muxed* (EPIHB16CFG/MODE = 0x0), *Without Byte Selects* (EPIHB16CFG/BSEL = 0x1), and With Chip Selects (EPIHB16CFG2/CSCFG = 0x0,1,2,3)

EPI PORT NAME		EPI SIGNAL FUNCTION				DEVICE PIN	
Accessible by Cortex™-M3	**Accessible by C28x**	**With Address Latch Enable (CSCFG = 0x0)**	**With One Chip Select (CSCFG = 0x1)**	**With Two Chip Selects (CSCFG = 0x2)**	**With ALE and Two Chip Selects (CSCFG = 0x3)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S0		AD0	AD0	AD0	AD0	PH3_GPIO51	
EPI0S1		AD1	AD1	AD1	AD1	PH2_GPIO50	
EPI0S2		AD2	AD2	AD2	AD2	PC4_GPIO68	
EPI0S3		AD3	AD3	AD3	AD3	PC5_GPIO69	
EPI0S4		AD4	AD4	AD4	AD4	PC6_GPIO70	
EPI0S5		AD5	AD5	AD5	AD5	PC7_GPIO71	
EPI0S6		AD6	AD6	AD6	AD6	PH0_GPIO48	
EPI0S7		AD7	AD7	AD7	AD7	PH1_GPIO49	
EPI0S8		AD8	AD8	AD8	AD8	PE0_GPIO24	
EPI0S9		AD9	AD9	AD9	AD9	PE1_GPIO25	
EPI0S10		AD10	AD10	AD10	AD10	PH4_GPIO52	
EPI0S11		AD11	AD11	AD11	AD11	PH5_GPIO53	
EPI0S12		AD12	AD12	AD12	AD12	PF4_GPIO36	
EPI0S13		AD13	AD13	AD13	AD13	PG0_GPIO40	
EPI0S14		AD14	AD14	AD14	AD14	PG1_GPIO41	
EPI0S15		AD15	AD15	AD15	AD15	PF5_GPIO37	
EPI0S16		A16	A16	A16	A16	PJ0_GPIO56	
EPI0S17		A17	A17	A17	A17	PJ1_GPIO57	
EPI0S18		A18	A18	A18	A18	PJ2_GPIO58	
EPI0S19		A19	A19	A19	A19	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A20	A20	A20	A20	PD2_GPIO18	
EPI0S21		A21	A21	A21	A21	PD3_GPIO19	
EPI0S22		A22	A22	A22	A22	PB5_GPIO13	
EPI0S23		A23	A23	A23	A23	PB4_GPIO12	
EPI0S24		A24	A24	A24	A24	PE2_GPIO26	
EPI0S25		A25	A25	A25	A25	PE3_GPIO27	
EPI0S26		A26	A26	A26	$\overline{CS0}$	PH6_GPIO54	
EPI0S27		A27	A27	$\overline{CS1}$	$\overline{CS1}$	PH7_GPIO55	
EPI0S30		ALE	$\overline{CS0}$	$\overline{CS0}$	ALE	PD7_GPIO23	PJ6_GPIO62
EPI0S29		WR	WR	WR	WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	RD	RD	RD	PD5_GPIO21	PJ4_GPIO60

Table 6-8. EPI MODES – 16-Bit Host-Bus Mode (EPICFG/MODE = 0x3), *Muxed* (EPIHB16CFG/MODE = 0x0), *Without Byte Selects* (EPIHB16CFG/BSEL = 0x1), and With Chip Selects (EPIHB16CFG2/CSCFG = 0x0,1,2,3) (continued)

EPI PORT NAME		EPI SIGNAL FUNCTION				DEVICE PIN	
Accessible by Cortex™-M3	**Accessible by C28x**	**With Address Latch Enable (CSCFG = 0x0)**	**With One Chip Select (CSCFG = 0x1)**	**With Two Chip Selects (CSCFG = 0x2)**	**With ALE and Two Chip Selects (CSCFG = 0x3)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S31		x	x	x	x	PG7_GPIO47	
EPI0S32		x	x	x	x	PF2_GPIO34	PC0_GPIO64
EPI0S33		x	x	x	x	PF3_GPIO35	PC1_GPIO65
EPI0S34		x	x	x	x	PE4_GPIO28	
EPI0S35		x	x	x	x	PE5_GPIO29	
EPI0S36		x	x	x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	x	x	PG2_GPIO42	
EPI0S40		x	x	x	x	PG5_GPIO45	
EPI0S41		x	x	x	x	PG6_GPIO46	
EPI0S42		x	x	x	x	PN6_GPIO102	
EPI0S43		x	x	x	x	PN7_GPIO103	

Table 6-9. EPI MODES – 16-Bit Host-Bus (EPICFG/MODE = 0x3), *Muxed* (EPIHB16CFG/MODE = 0x0), *With Byte Selects* (EPIHB16CFG/BSEL = 0x0), and With Chip Selects (EPIHB16CFG2/CSCFG=0x0,1,2,3)

EPI PORT NAME		EPI SIGNAL FUNCTION				DEVICE PIN	
Accessible by Cortex™-M3	**Accessible by C28x**	**With Address Latch Enable (CSCFG = 0x0)**	**With One Chip Select (CSCFG = 0x1)**	**With Two Chip Selects (CSCFG = 0x2)**	**With ALE and Two Chip Selects (CSCFG = 0x3)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S0		AD0	AD0	AD0	AD0	PH3_GPIO51	
EPI0S1		AD1	AD1	AD1	AD1	PH2_GPIO50	
EPI0S2		AD2	AD2	AD2	AD2	PC4_GPIO68	
EPI0S3		AD3	AD3	AD3	AD3	PC5_GPIO69	
EPI0S4		AD4	AD4	AD4	AD4	PC6_GPIO70	
EPI0S5		AD5	AD5	AD5	AD5	PC7_GPIO71	
EPI0S6		AD6	AD6	AD6	AD6	PH0_GPIO48	
EPI0S7		AD7	AD7	AD7	AD7	PH1_GPIO49	
EPI0S8		AD8	AD8	AD8	AD8	PE0_GPIO24	
EPI0S9		AD9	AD9	AD9	AD9	PE1_GPIO25	
EPI0S10		AD10	AD10	AD10	AD10	PH4_GPIO52	
EPI0S11		AD11	AD11	AD11	AD11	PH5_GPIO53	
EPI0S12		AD12	AD12	AD12	AD12	PF4_GPIO36	
EPI0S13		AD13	AD13	AD13	AD13	PG0_GPIO40	
EPI0S14		AD14	AD14	AD14	AD14	PG1_GPIO41	
EPI0S15		AD15	AD15	AD15	AD15	PF5_GPIO37	
EPI0S16		A16	A16	A16	A16	PJ0_GPIO56	
EPI0S17		A17	A17	A17	A17	PJ1_GPIO57	
EPI0S18		A18	A18	A18	A18	PJ2_GPIO58	
EPI0S19		A19	A19	A19	A19	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A20	A20	A20	A20	PD2_GPIO18	
EPI0S21		A21	A21	A21	A21	PD3_GPIO19	
EPI0S22		A22	A22	A22	A22	PB5_GPIO13	
EPI0S23		A23	A23	A23	A23	PB4_GPIO12	

Table 6-9. EPI MODES – 16-Bit Host-Bus (EPICFG/MODE = 0x3), *Muxed* (EPIHB16CFG/MODE = 0x0), *With Byte Selects* (EPIHB16CFG/BSEL = 0x0), and With Chip Selects (EPIHB16CFG2/CSCFG=0x0,1,2,3) (continued)

EPI PORT NAME		**EPI SIGNAL FUNCTION**				**DEVICE PIN**	
Accessible by Cortex™-M3	**Accessible by C28x**	**With Address Latch Enable (CSCFG = 0x0)**	**With One Chip Select (CSCFG = 0x1)**	**With Two Chip Selects (CSCFG = 0x2)**	**With ALE and Two Chip Selects (CSCFG = 0x3)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S24		A24	A24	A24	$\overline{\text{BSEL0}}$	PE2_GPIO26	
EPI0S25		A25	A25	$\overline{\text{BSEL0}}$	$\overline{\text{BSEL1}}$	PE3_GPIO27	
EPI0S26		$\overline{\text{BSEL0}}$	$\overline{\text{BSEL0}}$	$\overline{\text{BSEL1}}$	$\overline{\text{CS0}}$	PH6_GPIO54	
EPI0S27		$\overline{\text{BSEL1}}$	$\overline{\text{BSEL1}}$	$\overline{\text{CS1}}$	$\overline{\text{CS1}}$	PH7_GPIO55	
EPI0S30		ALE	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	ALE	PD7_GPIO23	PJ6_GPIO62
EPI0S29		WR	WR	WR	WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	RD	RD	RD	PD5_GPIO21	PJ4_GPIO60
EPI0S31		x	x	x	x	PG7_GPIO47	
EPI0S32		x	x	x	x	PF2_GPIO34	PC0_GPIO64
EPI0S33		x	x	x	x	PF3_GPIO35	PC1_GPIO65
EPI0S34		x	x	x	x	PE4_GPIO28	
EPI0S35		x	x	x	x	PE5_GPIO29	
EPI0S36		x	x	x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	x	x	PG2_GPIO42	
EPI0S40		x	x	x	x	PG5_GPIO45	
EPI0S41		x	x	x	x	PG6_GPIO46	
EPI0S42		x	x	x	x	PN6_GPIO102	
EPI0S43		x	x	x	x	PN7_GPIO103	

6.1.4.3.2.2 *HB-16 Non-Muxed Address/Data Mode*

The HB-16 Non-Muxed Mode uses dedicated pins for address and data signals. For this reason, the Non-Muxed Mode has reduced address reach as compared to the Muxed Mode. The HB-16 Non-Muxed Mode is selected with the MODE field of the HB-16 Configuration Register. In addition to data and address signals, the HB-16 Non-Muxed Mode also features the ALE signal (indicating to an external latch to capture address and hold the address until the data phase); RD and WR data strobes; 1–4 Chip Select (CS) signals to enable one of four external peripherals; and two Byte Select (BSEL) signals to accommodate byte accesses to lower or upper half of 16-bit data. The Byte Selects are chosen with the BSEL field of the HB-16 Configuration Register. The ALE and CS options are chosen with the CSCFG field of the HB-16 Configuration2 Register. For Non-Muxed bus cycles, most of the CSCFG modes also support a RDY signal. The RDY input to EPI is used by an external peripheral to extend bus cycles when the peripheral needs more time to complete reading or writing of data. While most EPI modes use up to 32 pins, the Non-Muxed CSCFG modes with 3 and 4 Chip Selects use 12 additional pins to extend the address reach and the number of CS signals. For detailed maps of HB-16 Non-Muxed Modes ***without*** Byte Selects, see Table 6-10 and Table 6-11. For detailed maps of HB-16 Non-Muxed Modes ***with*** Byte Selects, see Table 6-12 and Table 6-13.

Table 6-10. EPI MODES – 16-Bit Host-Bus Mode (EPICFG/MODE = 0x3), *Non-Muxed* (EPIHB16CFG/MODE = 0x1), *Without Byte Selects* (EPIHB16CFG/BSEL = 0x1), and With Chip Selects (EPIHB16CFG2/CSCFG = 0x0,1,2,3)

EPI PORT NAME		EPI SIGNAL FUNCTION				DEVICE PIN	
Accessible by Cortex™-M3	Accessible by C28x	With Address Latch Enable (CSCFG = 0x0)	With One Chip Select (CSCFG = 0x1)	With Two Chip Selects (CSCFG = 0x2)	With ALE and Two Chip Selects (CSCFG = 0x3)	(Available GPIOMUX_1 Muxing Choices for EPI)	
EPI0S0		D0	D0	D0	D0	PH3_GPIO51	
EPI0S1		D1	D1	D1	D1	PH2_GPIO50	
EPI0S2		D2	D2	D2	D2	PC4_GPIO68	
EPI0S3		D3	D3	D3	D3	PC5_GPIO69	
EPI0S4		D4	D4	D4	D4	PC6_GPIO70	
EPI0S5		D5	D5	D5	D5	PC7_GPIO71	
EPI0S6		D6	D6	D6	D6	PH0_GPIO48	
EPI0S7		D7	D7	D7	D7	PH1_GPIO49	
EPI0S8		D8	D8	D8	D8	PE0_GPIO24	
EPI0S9		D9	D9	D9	D9	PE1_GPIO25	
EPI0S10		D10	D10	D10	D10	PH4_GPIO52	
EPI0S11		D11	D11	D11	D11	PH5_GPIO53	
EPI0S12		D12	D12	D12	D12	PF4_GPIO36	
EPI0S13		D13	D13	D13	D13	PG0_GPIO40	
EPI0S14		D14	D14	D14	D14	PG1_GPIO41	
EPI0S15		D15	D15	D15	D15	PF5_GPIO37	
EPI0S16		A0	A0	A0	A0	PJ0_GPIO56	
EPI0S17		A1	A1	A1	A1	PJ1_GPIO57	
EPI0S18		A2	A2	A2	A2	PJ2_GPIO58	
EPI0S19		A3	A3	A3	A3	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A4	A4	A4	A4	PD2_GPIO18	
EPI0S21		A5	A5	A5	A5	PD3_GPIO19	
EPI0S22		A6	A6	A6	A6	PB5_GPIO13	
EPI0S23		A7	A7	A7	A7	PB4_GPIO12	
EPI0S24		A8	A8	A8	A8	PE2_GPIO26	
EPI0S25		A9	A9	A9	A9	PE3_GPIO27	
EPI0S26		A10	A10	A10	$\overline{CS0}$	PH6_GPIO54	
EPI0S27		A11	A11	$\overline{CS1}$	$\overline{CS1}$	PH7_GPIO55	
EPI0S30		ALE	$\overline{CS0}$	$\overline{CS0}$	ALE	PD7_GPIO23	PJ6_GPIO62
EPI0S29		WR	WR	WR	WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	RD	RD	RD	PD5_GPIO21	PJ4_GPIO60
EPI0S32		x	RDY	RDY	RDY	PF2_GPIO34	PC0_GPIO64
EPI0S31		x	x	x	x	PG7_GPIO47	
EPI0S33		x	x	x	x	PF3_GPIO35	PC1_GPIO65
EPI0S34		x	x	x	x	PE4_GPIO28	
EPI0S35		x	x	x	x	PE5_GPIO29	
EPI0S36		x	x	x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	x	x	PG2_GPIO42	
EPI0S40		x	x	x	x	PG5_GPIO45	
EPI0S41		x	x	x	x	PG6_GPIO46	
EPI0S42		x	x	x	x	PN6_GPIO102	
EPI0S43		x	x	x	x	PN7_GPIO103	

Table 6-11. EPI MODES – 16-Bit Host-Bus Mode (EPICFG/MODE=0x3), *Non-Muxed* (EPIHB16CFG/MODE = 0x1), *Without Byte Selects* (EPIHB16CFG/BSEL = 0x1), and *With Additional Chip Selects* (EPIHB16CFG2/CSCFG = 0x5,7)

EPI PORT NAME		EPI SIGNAL FUNCTION	DEVICE PIN	
Accessible by Cortex™-M3	Accessible by C28x	With *Three* Chip Selects (CSCFG = 0x7)	(Available GPIOMUX_1 Muxing Choices for EPI)	
EPI0S0		D0	PH3_GPIO51	
EPI0S1		D1	PH2_GPIO50	
EPI0S2		D2	PC4_GPIO68	
EPI0S3		D3	PC5_GPIO69	
EPI0S4		D4	PC6_GPIO70	
EPI0S5		D5	PC7_GPIO71	
EPI0S6		D6	PH0_GPIO48	
EPI0S7		D7	PH1_GPIO49	
EPI0S8		D8	PE0_GPIO24	
EPI0S9		D9	PE1_GPIO25	
EPI0S10		D10	PH4_GPIO52	
EPI0S11		D11	PH5_GPIO53	
EPI0S12		D12	PF4_GPIO36	
EPI0S13		D13	PG0_GPIO40	
EPI0S14		D14	PG1_GPIO41	
EPI0S15		D15	PF5_GPIO37	
EPI0S16		A0	PJ0_GPIO56	
EPI0S17		A1	PJ1_GPIO57	
EPI0S18		A2	PJ2_GPIO58	
EPI0S19		A3	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A4	PD2_GPIO18	
EPI0S21		A5	PD3_GPIO19	
EPI0S22		A6	PB5_GPIO13	
EPI0S23		A7	PB4_GPIO12	
EPI0S24		A8	PE2_GPIO26	
EPI0S25		A9	PE3_GPIO27	
EPI0S26		A10	PH6_GPIO54	
EPI0S36		A11	PB7_GPIO15	PC3_GPIO67
EPI0S37		A12	PB6_GPIO14	PC2_GPIO66
EPI0S38		A13	PF6_GPIO38	PE4_GPIO28
EPI0S39		A14	PG2_GPIO42	
EPI0S27		A15	PH7_GPIO55	
EPI0S35		A16	PE5_GPIO29	
EPI0S40		A17	PG5_GPIO45	
EPI0S41		A18	PG6_GPIO46	
EPI0S42		A19	PN6_GPIO102	
EPI0S43		A20	PN7_GPIO103	
EPI0S30		$\overline{CS0}$	PD7_GPIO23	PJ6_GPIO62
EPI0S34		$\overline{CS2}$	PE4_GPIO28	
EPI0S33		$\overline{CS3}$	PF3_GPIO35	PC1_GPIO65
EPI0S29		WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	PD5_GPIO21	PJ4_GPIO60
EPI0S32		RDY	PF2_GPIO34	PC0_GPIO64
EPI0S31		x	PG7_GPIO47	

EPI PORT NAME		EPI SIGNAL FUNCTION	DEVICE PIN	
Accessible by Cortex™-M3	Accessible by C28x	With *Four* Chip Selects (CSCFG = 0x5)	(Available GPIOMUX_1 Muxing Choices for EPI)	
EPI0S0		D0	PH3_GPIO51	
EPI0S1		D1	PH2_GPIO50	
EPI0S2		D2	PC4_GPIO68	
EPI0S3		D3	PC5_GPIO69	
EPI0S4		D4	PC6_GPIO70	
EPI0S5		D5	PC7_GPIO71	
EPI0S6		D6	PH0_GPIO48	
EPI0S7		D7	PH1_GPIO49	
EPI0S8		D8	PE0_GPIO24	
EPI0S9		D9	PE1_GPIO25	
EPI0S10		D10	PH4_GPIO52	
EPI0S11		D11	PH5_GPIO53	
EPI0S12		D12	PF4_GPIO36	
EPI0S13		D13	PG0_GPIO40	
EPI0S14		D14	PG1_GPIO41	
EPI0S15		D15	PF5_GPIO37	
EPI0S16		A0	PJ0_GPIO56	
EPI0S17		A1	PJ1_GPIO57	
EPI0S18		A2	PJ2_GPIO58	
EPI0S19		A3	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A4	PD2_GPIO18	
EPI0S21		A5	PD3_GPIO19	
EPI0S22		A6	PB5_GPIO13	
EPI0S23		A7	PB4_GPIO12	
EPI0S24		A8	PE2_GPIO26	
EPI0S25		A9	PE3_GPIO27	
EPI0S26		A10	PH6_GPIO54	
EPI0S36		A11	PB7_GPIO15	PC3_GPIO67
EPI0S37		A12	PB6_GPIO14	PC2_GPIO66
EPI0S38		A13	PF6_GPIO38	PE4_GPIO28
EPI0S39		A14	PG2_GPIO42	
EPI0S40		A15	PG5_GPIO45	
EPI0S41		A16	PG6_GPIO46	
EPI0S42		A17	PN6_GPIO102	
EPI0S43		A18	PN7_GPIO103	
EPI0S30		$\overline{CS0}$	PD7_GPIO23	PJ6_GPIO62
EPI0S27		$\overline{CS1}$	PH7_GPIO55	
EPI0S34		$\overline{CS2}$	PE4_GPIO28	
EPI0S33		$\overline{CS3}$	PF3_GPIO35	PC1_GPIO65
EPI0S29		WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	PD5_GPIO21	PJ4_GPIO60
EPI0S32		RDY	PF2_GPIO34	PC0_GPIO64
EPI0S31		x	PG7_GPIO47	
EPI0S35		x	PE5_GPIO29	

Table 6-12. EPI MODES – 16-Bit Host-Bus (EPICFG/MODE = 0x3), *Non-Muxed* (EPIHB16CFG/MODE = 0x1), *With Byte Selects* (EPIHB16CFG/BSEL = 0x0), and With Chip Selects (EPIHB16CFG2/CSCFG = 0x0,1,2,3)

EPI PORT NAME		EPI SIGNAL FUNCTION				DEVICE PIN	
Accessible by Cortex™-M3	Accessible by C28x	With Address Latch Enable (CSCFG = 0x0)	With One Chip Select (CSCFG = 0x1)	With Two Chip Selects (CSCFG = 0x2)	With ALE and Two Chip Selects (CSCFG = 0x3)	(Available GPIOMUX_1 Muxing Choices for EPI)	
EPI0S0		D0	D0	D0	D0	PH3_GPIO51	
EPI0S1		D1	D1	D1	D1	PH2_GPIO50	
EPI0S2		D2	D2	D2	D2	PC4_GPIO68	
EPI0S3		D3	D3	D3	D3	PC5_GPIO69	
EPI0S4		D4	D4	D4	D4	PC6_GPIO70	
EPI0S5		D5	D5	D5	D5	PC7_GPIO71	
EPI0S6		D6	D6	D6	D6	PH0_GPIO48	
EPI0S7		D7	D7	D7	D7	PH1_GPIO49	
EPI0S8		D8	D8	D8	D8	PE0_GPIO24	
EPI0S9		D9	D9	D9	D9	PE1_GPIO25	
EPI0S10		D10	D10	D10	D10	PH4_GPIO52	
EPI0S11		D11	D11	D11	D11	PH5_GPIO53	
EPI0S12		D12	D12	D12	D12	PF4_GPIO36	
EPI0S13		D13	D13	D13	D13	PG0_GPIO40	
EPI0S14		D14	D14	D14	D14	PG1_GPIO41	
EPI0S15		D15	D15	D15	D15	PF5_GPIO37	
EPI0S16		A0	A0	A0	A0	PJ0_GPIO56	
EPI0S17		A1	A1	A1	A1	PJ1_GPIO57	
EPI0S18		A2	A2	A2	A2	PJ2_GPIO58	
EPI0S19		A3	A3	A3	A3	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A4	A4	A4	A4	PD2_GPIO18	
EPI0S21		A5	A5	A5	A5	PD3_GPIO19	
EPI0S22		A6	A6	A6	A6	PB5_GPIO13	
EPI0S23		A7	A7	A7	A7	PB4_GPIO12	
EPI0S24		A8	A8	A8	$\overline{\text{BSEL0}}$	PE2_GPIO26	
EPI0S25		A9	A9	$\overline{\text{BSEL0}}$	$\overline{\text{BSEL1}}$	PE3_GPIO27	
EPI0S26		$\overline{\text{BSEL0}}$	$\overline{\text{BSEL0}}$	$\overline{\text{BSEL1}}$	$\overline{\text{CS0}}$	PH6_GPIO54	
EPI0S27		$\overline{\text{BSEL1}}$	$\overline{\text{BSEL1}}$	$\overline{\text{CS1}}$	$\overline{\text{CS1}}$	PH7_GPIO55	
EPI0S30		ALE	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	ALE	PD7_GPIO23	PJ6_GPIO62
EPI0S29		WR	WR	WR	WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	RD	RD	RD	PD5_GPIO21	PJ4_GPIO60
EPI0S32		x	RDY	RDY	RDY	PF2_GPIO34	PC0_GPIO64
EPI0S31		x	x	x	x	PG7_GPIO47	
EPI0S33		x	x	x	x	PF3_GPIO35	PC1_GPIO65
EPI0S34		x	x	x	x	PE4_GPIO28	
EPI0S35		x	x	x	x	PE5_GPIO29	
EPI0S36		x	x	x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	x	x	PG2_GPIO42	
EPI0S40		x	x	x	x	PG5_GPIO45	
EPI0S41		x	x	x	x	PG6_GPIO46	
EPI0S42		x	x	x	x	PN6_GPIO102	
EPI0S43		x	x	x	x	PN7_GPIO103	

Table 6-13. EPI MODES – 16-Bit Host-Bus (EPICFG/MODE = 0x3), *Non-Muxed* (EPIHB16CFG/MODE = 0x1), *With Byte Selects* (EPIHB16CFG/BSEL = 0x0), and *With Additional Chip Selects* (EPIHB16CFG2/CSCFG = 0x5,7)

EPI PORT NAME		EPI SIGNAL FUNCTION	DEVICE PIN	
Accessible by Cortex™-M3	**Accessible by C28x**	**With *Three* Chip Selects (CSCFG = 0x7)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S0		D0	PH3_GPIO51	
EPI0S1		D1	PH2_GPIO50	
EPI0S2		D2	PC4_GPIO68	
EPI0S3		D3	PC5_GPIO69	
EPI0S4		D4	PC6_GPIO70	
EPI0S5		D5	PC7_GPIO71	
EPI0S6		D6	PH0_GPIO48	
EPI0S7		D7	PH1_GPIO49	
EPI0S8		D8	PE0_GPIO24	
EPI0S9		D9	PE1_GPIO25	
EPI0S10		D10	PH4_GPIO52	
EPI0S11		D11	PH5_GPIO53	
EPI0S12		D12	PF4_GPIO36	
EPI0S13		D13	PG0_GPIO40	
EPI0S14		D14	PG1_GPIO41	
EPI0S15		D15	PF5_GPIO37	
EPI0S16		A0	PJ0_GPIO56	
EPI0S17		A1	PJ1_GPIO57	
EPI0S18		A2	PJ2_GPIO58	
EPI0S19		A3	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A4	PD2_GPIO18	
EPI0S21		A5	PD3_GPIO19	
EPI0S22		A6	PB5_GPIO13	
EPI0S23		A7	PB4_GPIO12	
EPI0S24		A8	PE2_GPIO26	
EPI0S40		A9	PG5_GPIO45	
EPI0S41		A10	PG6_GPIO46	
EPI0S36		A11	PB7_GPIO15	PC3_GPIO67
EPI0S37		A12	PB6_GPIO14	PC2_GPIO66
EPI0S38		A13	PF6_GPIO38	PE4_GPIO28
EPI0S39		A14	PG2_GPIO42	
EPI0S27		A15	PH7_GPIO55	
EPI0S35		A16	PE5_GPIO29	
EPI0S42		A17	PN6_GPIO102	
EPI0S43		A18	PN7_GPIO103	
EPI0S25		$\overline{\text{BSEL0}}$	PE3_GPIO27	
EPI0S26		$\overline{\text{BSEL1}}$	PH6_GPIO54	
EPI0S30		$\overline{\text{CS0}}$	PD7_GPIO23	PJ6_GPIO62
EPI0S34		$\overline{\text{CS2}}$	PE4_GPIO28	
EPI0S33		$\overline{\text{CS3}}$	PF3_GPIO35	PC1_GPIO65
EPI0S29		WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	PD5_GPIO21	PJ4_GPIO60
EPI0S32		RDY	PF2_GPIO34	PC0_GPIO64
EPI0S31		x	PG7_GPIO47	

EPI PORT NAME		EPI SIGNAL FUNCTION	DEVICE PIN	
Accessible by Cortex™-M3	**Accessible by C28x**	**With *Four* Chip Selects (CSCFG = 0x5)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S0		D0	PH3_GPIO51	
EPI0S1		D1	PH2_GPIO50	
EPI0S2		D2	PC4_GPIO68	
EPI0S3		D3	PC5_GPIO69	
EPI0S4		D4	PC6_GPIO70	
EPI0S5		D5	PC7_GPIO71	
EPI0S6		D6	PH0_GPIO48	
EPI0S7		D7	PH1_GPIO49	
EPI0S8		D8	PE0_GPIO24	
EPI0S9		D9	PE1_GPIO25	
EPI0S10		D10	PH4_GPIO52	
EPI0S11		D11	PH5_GPIO53	
EPI0S12		D12	PF4_GPIO36	
EPI0S13		D13	PG0_GPIO40	
EPI0S14		D14	PG1_GPIO41	
EPI0S15		D15	PF5_GPIO37	
EPI0S16		A0	PJ0_GPIO56	
EPI0S17		A1	PJ1_GPIO57	
EPI0S18		A2	PJ2_GPIO58	
EPI0S19		A3	PD4_GPIO20	PJ3_GPIO59
EPI0S20		A4	PD2_GPIO18	
EPI0S21		A5	PD3_GPIO19	
EPI0S22		A6	PB5_GPIO13	
EPI0S23		A7	PB4_GPIO12	
EPI0S24		A8	PE2_GPIO26	
EPI0S40		A9	PG5_GPIO45	
EPI0S41		A10	PG6_GPIO46	
EPI0S36		A11	PB7_GPIO15	PC3_GPIO67
EPI0S37		A12	PB6_GPIO14	PC2_GPIO66
EPI0S38		A13	PF6_GPIO38	PE4_GPIO28
EPI0S39		A14	PG2_GPIO42	
EPI0S42		A15	PN6_GPIO102	
EPI0S43		A16	PN7_GPIO103	
EPI0S25		$\overline{\text{BSEL0}}$	PE3_GPIO27	
EPI0S26		$\overline{\text{BSEL1}}$	PH6_GPIO54	
EPI0S30		$\overline{\text{CS0}}$	PD7_GPIO23	PJ6_GPIO62
EPI0S27		$\overline{\text{CS1}}$	PH7_GPIO55	
EPI0S34		$\overline{\text{CS2}}$	PE4_GPIO28	
EPI0S33		$\overline{\text{CS3}}$	PF3_GPIO35	PC1_GPIO65
EPI0S29		WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	PD5_GPIO21	PJ4_GPIO60
EPI0S32		RDY	PF2_GPIO34	PC0_GPIO64
EPI0S31		x	PG7_GPIO47	
EPI0S35		x	PE5_GPIO29	

6.1.4.3.2.3 HB-16 FIFO Mode

The HB-16 FIFO Mode uses 16 bits of data, removes ALE and address pins, and optionally adds external FIFO Full/Empty flag inputs. This scheme is used by many devices, such as radios, communication devices (including USB2 devices), and some FPGA configuration (FIFO throughblock RAM). This FIFO Mode presents the data side of the normal Host-Bus interface, but is paced by FIFO control signals. It is important to consider that the FIFO Full/Empty control inputs may stall the EPI interface and can potentially block other CPU or DMA accesses. For detailed maps of the HB-16 FIFO Mode, see Table 6-14.

Table 6-14. EPI MODES – 16-Bit Host-Bus Mode (EPICFG/MODE = 0x3), FIFO Mode (EPIHB16CFG/MODE = 0x3)

EPI PORT NAME		EPI SIGNAL FUNCTION		DEVICE PIN	
Accessible by Cortex™-M3	Accessible by C28x	With One Chip Select (CSCFG = 0x1)	With Two Chip Selects (CSCFG = 0x2)	(Available GPIOMUX_1 Muxing Choices for EPI)	
EPI0S0		D0	D0	PH3_GPIO51	
EPI0S1		D1	D1	PH2_GPIO50	
EPI0S2		D2	D2	PC4_GPIO68	
EPI0S3		D3	D3	PC5_GPIO69	
EPI0S4		D4	D4	PC6_GPIO70	
EPI0S5		D5	D5	PC7_GPIO71	
EPI0S6		D6	D6	PH0_GPIO48	
EPI0S7		D7	D7	PH1_GPIO49	
EPI0S8		D8	D8	PE0_GPIO24	
EPI0S9		D9	D9	PE1_GPIO25	
EPI0S10		D10	D10	PH4_GPIO52	
EPI0S11		D11	D11	PH5_GPIO53	
EPI0S12		D12	D12	PF4_GPIO36	
EPI0S13		D13	D13	PG0_GPIO40	
EPI0S14		D14	D14	PG1_GPIO41	
EPI0S15		D15	D15	PF5_GPIO37	
EPI0S25		x	$\overline{CS1}$	PE3_GPIO27	
EPI0S30		$\overline{CS0}$	$\overline{CS0}$	PD7_GPIO23	PJ6_GPIO62
EPI0S27		FFULL	FFULL	PH7_GPIO55	
EPI0S26		FEMPTY	FEMPTY	PH6_GPIO54	
EPI0S29		WR	WR	PD6_GPIO22	PJ5_GPIO61
EPI0S28		RD	RD	PD5_GPIO21	PJ4_GPIO60
EPI0S32		x	x	PF2_GPIO34	PC0_GPIO64
EPI0S16		x	x	PJ0_GPIO56	
EPI0S17		x	x	PJ1_GPIO57	
EPI0S18		x	x	PJ2_GPIO58	
EPI0S19		x	x	PD4_GPIO20	PJ3_GPIO59
EPI0S20		x	x	PD2_GPIO18	
EPI0S21		x	x	PD3_GPIO19	
EPI0S22		x	x	PB5_GPIO13	
EPI0S23		x	x	PB4_GPIO12	
EPI0S24		x	x	PE2_GPIO26	
EPI0S31		x	x	PG7_GPIO47	

Table 6-14. EPI MODES – 16-Bit Host-Bus Mode (EPICFG/MODE = 0x3), FIFO Mode (EPIHB16CFG/MODE = 0x3) (continued)

EPI PORT NAME		EPI SIGNAL FUNCTION		DEVICE PIN	
Accessible by Cortex™-M3	**Accessible by C28x**	**With One Chip Select (CSCFG = 0x1)**	**With Two Chip Selects (CSCFG = 0x2)**	**(Available GPIOMUX_1 Muxing Choices for EPI)**	
EPI0S33		x	x	PF3_GPIO35	PC1_GPIO65
EPI0S34		x	x	PE4_GPIO28	
EPI0S35		x	x	PE5_GPIO29	
EPI0S36		x	x	PB7_GPIO15	PC3_GPIO67
EPI0S37		x	x	PB6_GPIO14	PC2_GPIO66
EPI0S38		x	x	PF6_GPIO38	PE4_GPIO28
EPI0S39		x	x	PG2_GPIO42	
EPI0S40		x	x	PG5_GPIO45	
EPI0S41		x	x	PG6_GPIO46	
EPI0S42		x	x	PN6_GPIO102	
EPI0S43		x	x	PN7_GPIO103	

6.2 Master Subsystem Peripherals

Master Subsystem peripherals are located on the APB Bus and AHB Bus, and are accessible from the Cortex™-M3 CPU/µDMA. The AHB peripherals include EPI, USB, and two CAN modules. The APB peripherals include EMAC, two I2Cs, five UARTs, four SSIs, four GPTIMERs, two WDOGs, NMI WDOG, and a µCRC module (Cyclic Redundancy Check). The Cortex™-M3 CPU/µDMA also have access to Analog (Result Registers only) and Shared peripherals (see Section 6.1).

For detailed information on the processor peripherals, see the *Concerto F28M36x Technical Reference Manual* (literature number SPRUHE8).

6.2.1 Synchronous Serial Interface (SSI)

This device has four Synchronous Serial Interface (SSI) modules. Each SSI has a Master or Slave interface for synchronous serial communication with peripheral devices that have Texas Instruments™ Synchronous Serial interfaces, SPI, MICROWIRE, or Freescale™ serial format.

The SSI peripheral performs serial-to-parallel conversion on data received from a peripheral device. The CPU accesses data, control, and status information. The transmit and receive paths are buffered with internal FIFO memories, allowing up to eight 16-bit values to be stored independently in both transmit and receive modes. The SSI also supports µDMA transfers. The transmit and receive FIFOs can be programmed as destination/source addresses in the µDMA module. An µDMA operation is enabled by setting the appropriate bit or bits in the SSIDMACTL register.

Figure 6-11 shows the SSI peripheral.

6.2.1.1 Bit Rate Generation

The SSI includes a programmable bit-rate clock divider and prescaler to generate the serial output clock. Bit rates are supported to 2 MHz and higher, although maximum bit rate is determined by peripheral devices. The serial bit rate is derived by dividing-down the input clock (SysClk). The clock is first divided by an even prescale value CPSDVSR from 2 to 254, which is programmed in the SSI Clock Prescale (SSICPSR) register. The clock is further divided by a value from 1 to 256, which is 1 + SCR, where SCR is the value programmed in the SSI Control 0 (SSICR0) register. The frequency of the output clock SSIClk is defined by:

SSIClk = SysClk / [CPSDVSR * (1 + SCR)]

NOTE

For master mode, the system clock must be at least four times faster than SSIClk, with the restriction that SSIClk cannot be faster than 25 MHz. For slave mode, the system clock must be at least 12 times faster than SSIClk.

6.2.1.2 Transmit FIFO

The transmit FIFO is a 16-bit-wide, 8-location-deep, first-in, first-out memory buffer. The CPU writes data to the FIFO through the SSI Data (SSIDR) register, and data is stored in the FIFO until the data is read out by the transmission logic. When configured as a master or a slave, parallel data is written into the transmit FIFO prior to serial conversion and transmission to the attached slave or master, respectively, through the SSITx pin.

In slave mode, the SSI transmits data each time the master initiates a transaction. If the transmit FIFO is empty and the master initiates a transaction, the slave transmits the 8th most recent value in the transmit FIFO. If less than eight values have been written to the transmit FIFO since the SSI module clock was enabled using the SSI bit in the RGCG1 register, then "0" is transmitted. Care should be taken to ensure that valid data is in the FIFO as needed. The SSI can be configured to generate an interrupt or an µDMA request when the FIFO is empty.

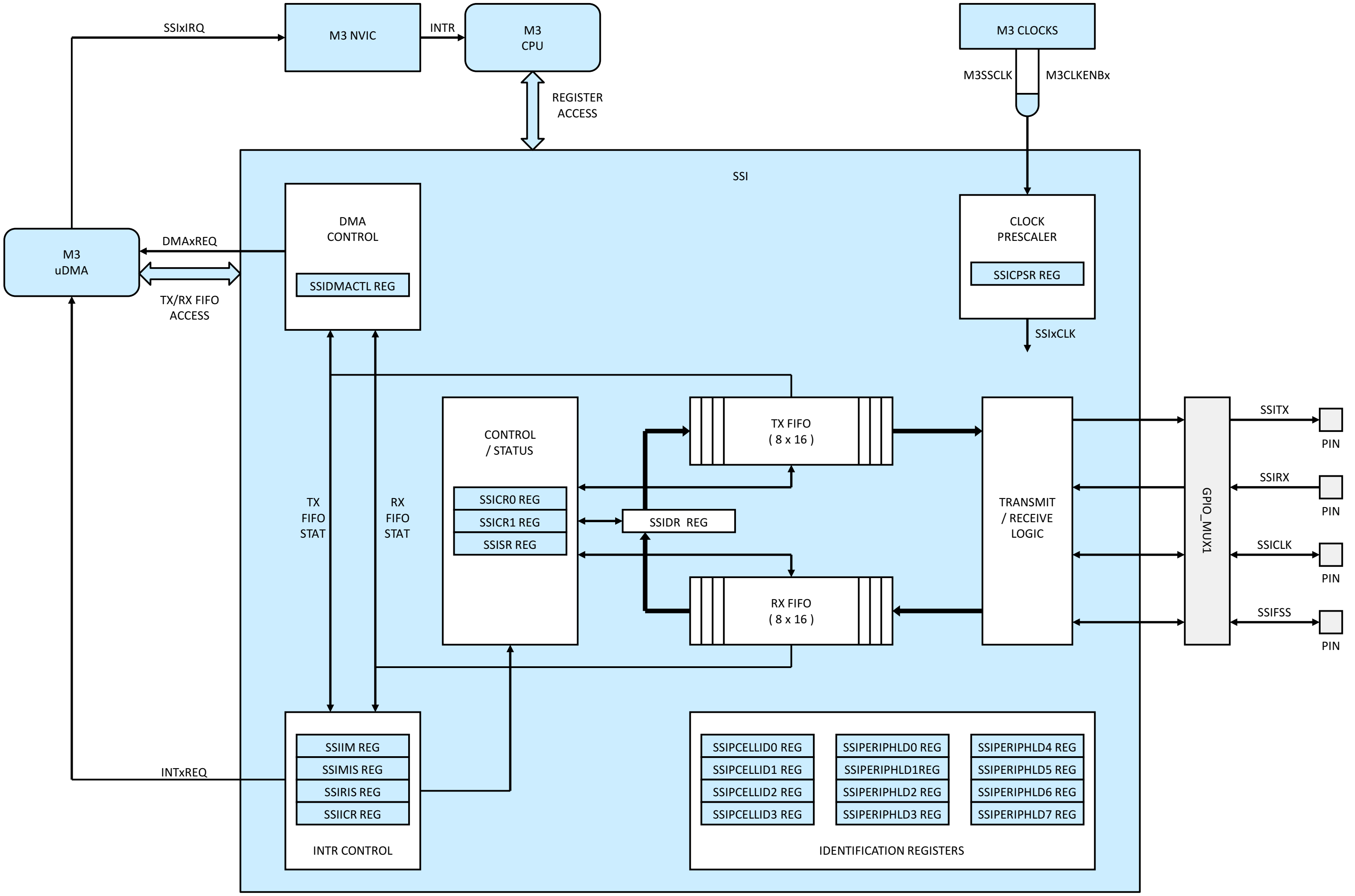


Figure 6-11. SSI

6.2.1.3 Receive FIFO

The receive FIFO is a 16-bit-wide, 8-location-deep, first-in, first-out memory buffer. Received data from the serial interface is stored in the buffer until read out by the CPU, which accesses the read FIFO by reading the SSIDR register. When configured as a master or slave, serial data received through the SSIRx pin is registered prior to parallel loading into the attached slave or master receive FIFO, respectively.

6.2.1.4 Interrupts

The SSI can generate interrupts when the following conditions are observed:

- Transmit FIFO service (when the transmit FIFO is half full or less)
- Receive FIFO service (when the receive FIFO is half full or more)
- Receive FIFO time-out
- Receive FIFO overrun
- End of transmission

All of the interrupt events are ORed together before being sent to the interrupt controller, so the SSI generates a single interrupt request to the controller regardless of the number of active interrupts. Each of the four individual maskable interrupts can be masked by clearing the appropriate bit in the SSI Interrupt Mask (SSIIM) register. Setting the appropriate mask bit enables the interrupt.

The individual outputs, along with a combined interrupt output, allow the use of either a global interrupt service routine or modular device drivers to handle interrupts. The transmit and receive dynamic data-flow interrupts have been separated from the status interrupts so that data can be read or written in response to the FIFO trigger levels. The status of the individual interrupt sources can be read from the SSI Raw Interrupt Status (SSIRIS) and SSI Masked Interrupt Status (SSIMIS) registers.

The receive FIFO has a time-out period that is 32 periods at the rate of SSIClk (whether or not SSIClk is currently active) and is started when the RX FIFO goes from EMPTY to not-EMPTY. If the RX FIFO is emptied before 32 clocks have passed, the time-out period is reset. As a result, the ISR should clear the Receive FIFO Time-out Interrupt just after reading out the RX FIFO by writing a "1" to the RTIC bit in the SSI Interrupt Clear (SSIICR) register. The interrupt should not be cleared so late that the ISR returns before the interrupt is actually cleared, or the ISR may be reactivated unnecessarily.

The End-of-Transmission (EOT) interrupt indicates that the data has been transmitted completely. This interrupt can be used to indicate when it is safe to turn off the SSI module clock or enter sleep mode. In addition, because transmitted data and received data complete at exactly the same time, the interrupt can also indicate that read data is ready immediately, without waiting for the receive FIFO time-out period to complete.

6.2.1.5 Frame Formats

Each data frame is between 4 bits and 16 bits long, depending on the size of data programmed, and is transmitted starting with the MSB. Three basic frame types can be selected:

- Texas Instruments™ Synchronous Serial
- Freescale™ SPI
- MICROWIRE

For all three formats, the serial clock (SSIClk) is held inactive while the SSI is idle, and SSIClk transitions at the programmed frequency only during active transmission or reception of data. The idle state of SSIClk is utilized to provide a receive time-out indication that occurs when the receive FIFO still contains data after a time-out period.

6.2.2 *Universal Asynchronous Receiver/Transmitter (UART)*

This device has five Universal Asynchronous Receiver/Transmitter (UART) modules. The CPU accesses data, control, and status information. The UART also supports µDMA transfers. Each UART performs functions of parallel-to-serial and serial-to-parallel conversions. Each of the five UART modules is similar in functionality to a 16C550 UART, but is not register-compatible.

The UART is configured for transmit and receive via the TXE bit and the RXE bit, respectively, of the UART Control (UARTCTL) register. Transmit and receive are both enabled out of reset. Before any control registers are programmed, the UART must be disabled by clearing the UARTEN bit in UARTCTL. If the UART is disabled during a TX or RX operation, the current transaction is completed prior to the UART stopping.

The UART module also includes a serial IR (SIR) encoder/decoder block that can be connected to an infrared transceiver to implement an IrDA SIR physical layer. The SIR function is programmed using the UARTCTL register.

Figure 6-12 shows the UART peripheral.

6.2.2.1 Baud-Rate Generation

The baud-rate divisor is a 22-bit number consisting of a 16-bit integer and a 6-bit fractional part. The number formed by these two values is used by the baud-rate generator to determine the bit period. Having a fractional baud-rate divider allows the UART to generate all the standard baud rates.

The 16-bit integer is loaded through the UART Integer Baud-Rate Divisor (UARTIBRD) register, and the 6-bit fractional part is loaded with the UART Fractional Baud-Rate Divisor (UARTFBRD) register. The baud rate divisor (BRD) has the following relationship to the system clock (where BRDI is the integer part of the BRD, and BRDF is the fractional part, separated by a decimal place).

$$\text{BRD} = \text{BRDI} + \text{BRDF} = \text{UARTSysClk} / (\text{ClkDiv} * \text{Baud Rate})$$

where UARTSysClk is the system clock connected to the UART, and ClkDiv is either 16 (if HSE in UARTCTL is clear) or 8 (if HSE is set).

The 6-bit fractional number (that is to be loaded into the DIVFRAC bit field in the UARTFBRD register) can be calculated by taking the fractional part of the baud-rate divisor, multiplying this fractional part by 64, and adding 0.5 to account for rounding errors:

$$\text{UARTFBRD[DIVFRAC]} = \text{integer}(\text{BRDF} * 64 + 0.5)$$

The UART generates an internal baud-rate reference clock at 8x or 16x the baud rate [referred to as Baud8 and Baud16, depending on the setting of the HSE bit (bit 5 in UARTCTL)]. This reference clock is divided by 8 or 16 to generate the transmit clock, and is used for error detection during receive operations.

Along with the UART Line Control, High Byte (UARTLCRH) register, the UARTIBRD and UARTFBRD registers form an internal 30-bit register. This internal register is only updated when a write operation to UARTLCRH is performed, so any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register for the changes to take effect.

6.2.2.2 Transmit and Receive Logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. The control logic outputs the serial bit stream beginning with a start bit and followed by the data bits (LSB first), parity bit, and the stop bits according to the programmed configuration in the control registers.

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line-break detection are also performed, and their status accompanies the data that is written to the receive FIFO.

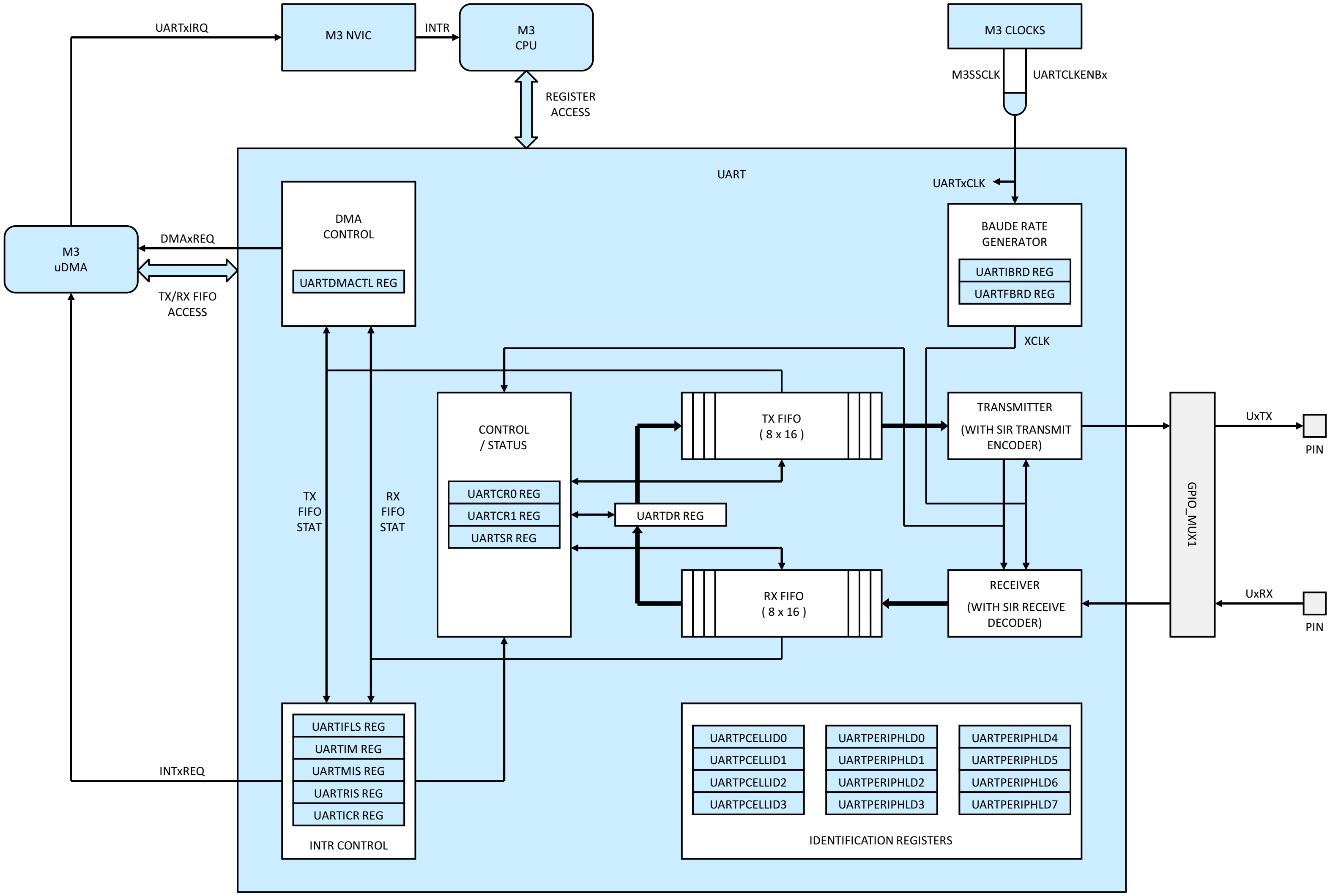


Figure 6-12. UART

6.2.2.3 Data Transmission and Reception

Data received or transmitted is stored in two 16-byte FIFOs, though the receive FIFO has an extra four bits per character for status information. For transmission, data is written into the transmit FIFO. If the UART is enabled, a data frame starts transmitting with the parameters indicated in the UARTLCRH register. Data continues to be transmitted until there is no data left in the transmit FIFO. The BUSY bit in the UART Flag (UARTFR) register is asserted as soon as data is written to the transmit FIFO (that is, if the FIFO is non-empty) and remains asserted while data is being transmitted. The BUSY bit is negated only when the transmit FIFO is empty, and the last character has been transmitted from the shift register, including the stop bits. The UART can indicate that it is busy even though the UART may no longer be enabled.

When the receiver is idle (the UnRx signal is continuously "1"), and the data input goes Low (a start bit has been received), the receive counter begins running and data is sampled on the eighth cycle of Baud16 or the fourth cycle of Baud8, depending on the setting of the HSE bit (bit 5 in UARTCTL).

The start bit is valid and recognized if the UnRx signal is still low on the eighth cycle of Baud16 (HSE clear) or the fourth cycle of Baud 8 (HSE set), otherwise the start bit is ignored. After a valid start bit is detected, successive data bits are sampled on every 16th cycle of Baud16 or 8th cycle of Baud8 (that is, one bit period later), according to the programmed length of the data characters and value of the HSE bit in UARTCTL. The parity bit is then checked if parity mode is enabled. Data length and parity are defined in the UARTLCRH register.

Lastly, a valid stop bit is confirmed if the UnRx signal is High, otherwise a framing error has occurred. When a full word is received, the data is stored in the receive FIFO along with any error bits associated with that word.

6.2.2.4 Interrupts

The UART can generate interrupts when the following conditions are observed:

- Overrun Error
- Break Error
- Parity Error
- Framing Error
- Receive Time-out
- Transmit (when the condition defined in the TXIFLSEL bit in the UARTIFLS register is met, or if the EOT bit in UARTCTL is set, when the last bit of all transmitted data leaves the serializer)
- Receive (when the condition defined in the RXIFLSEL bit in the UARTIFLS register is met)

All of the interrupt events are ORed together before being sent to the interrupt controller, so the UART can only generate a single interrupt request to the controller at any given time. Software can service multiple interrupt events in a single interrupt service routine by reading the UART Masked Interrupt Status (UARTMIS) register.

The interrupt events that can trigger a controller-level interrupt are defined in the UART Interrupt Mask (UARTIM) register by setting the corresponding IM bits. If interrupts are not used, the raw interrupt status is always visible via the UART Raw Interrupt Status (UARTRIS) register.

Interrupts are always cleared (for both the UARTMIS and UARTRIS registers) by writing a "1" to the corresponding bit in the UART Interrupt Clear (UARTICR) register.

The receive time-out interrupt is asserted when the receive FIFO is not empty, and no further data is received over a 32-bit period. The receive time-out interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), or when a "1" is written to the corresponding bit in the UARTICR register.

6.2.3 Cortex™-M3 Inter-Integrated Circut (I2C)

This device has two Cortex™-M3 I2C peripherals. The Cortex™-M3 Inter-Integrated Circuit (I2C) bus provides bidirectional data transfer through a two-wire design (a serial data line SDA and a serial clock line SCL), and interfaces to external I2C devices such as serial memory (RAMs and ROMs), networking devices, LCDs, tone generators, and so on. The I2C bus may also be used for system testing and diagnostic purposes in product development and manufacture. The microcontroller includes two I2C modules, providing the ability to interact (both transmit and receive) with other I2C devices on the bus.

The two Cortex™-M3 I2C modules include the following features:

- Devices on the I2C bus can be designated as either a master or a slave
  - Supports both transmitting and receiving data as either a master or a slave
  - Supports simultaneous master and slave operation
- Four I2C modes
  - Master transmit
  - Master receive
  - Slave transmit
  - Slave receive
- Two transmission speeds: Standard (100 Kbps) and Fast (400 Kbps)
- Master and slave interrupt generation
  - Master generates interrupts when a transmit or receive operation completes (or aborts due to an error)
  - Slave generates interrupts when data has been transferred or requested by a master or when a START or STOP condition is detected
- Master with arbitration and clock synchronization, multimaster support, and 7-bit addressing mode

Figure 6-13 shows the Cortex™-M3 I2C peripheral.

6.2.3.1 Functional Overview

Each I2C module comprises both master and slave functions. For proper operation, the SDA and SCL pins must be configured as open-drain signals.

The I2C bus uses only two signals: SDA and SCL, named I2CSDA and I2CSCL. SDA is the bidirectional serial data line and SCL is the bidirectional serial clock line. The bus is considered idle when both lines are high.

Every transaction on the I2C bus is nine bits long, consisting of eight data bits and a single acknowledge bit. The number of bytes per transfer (defined as the time between a valid START and STOP condition) is unrestricted, but each byte has to be followed by an acknowledge bit, and data must be transferred MSB first. When a receiver cannot receive another complete byte, the receiver can hold the clock line SCL Low and force the transmitter into a wait state. The data transfer continues when the receiver releases the clock SCL.

6.2.3.2 Available Speed Modes

The I2C bus can run in either standard mode (100 Kbps) or fast mode (400 Kbps). The selected mode should match the speed of the other I2C devices on the bus.

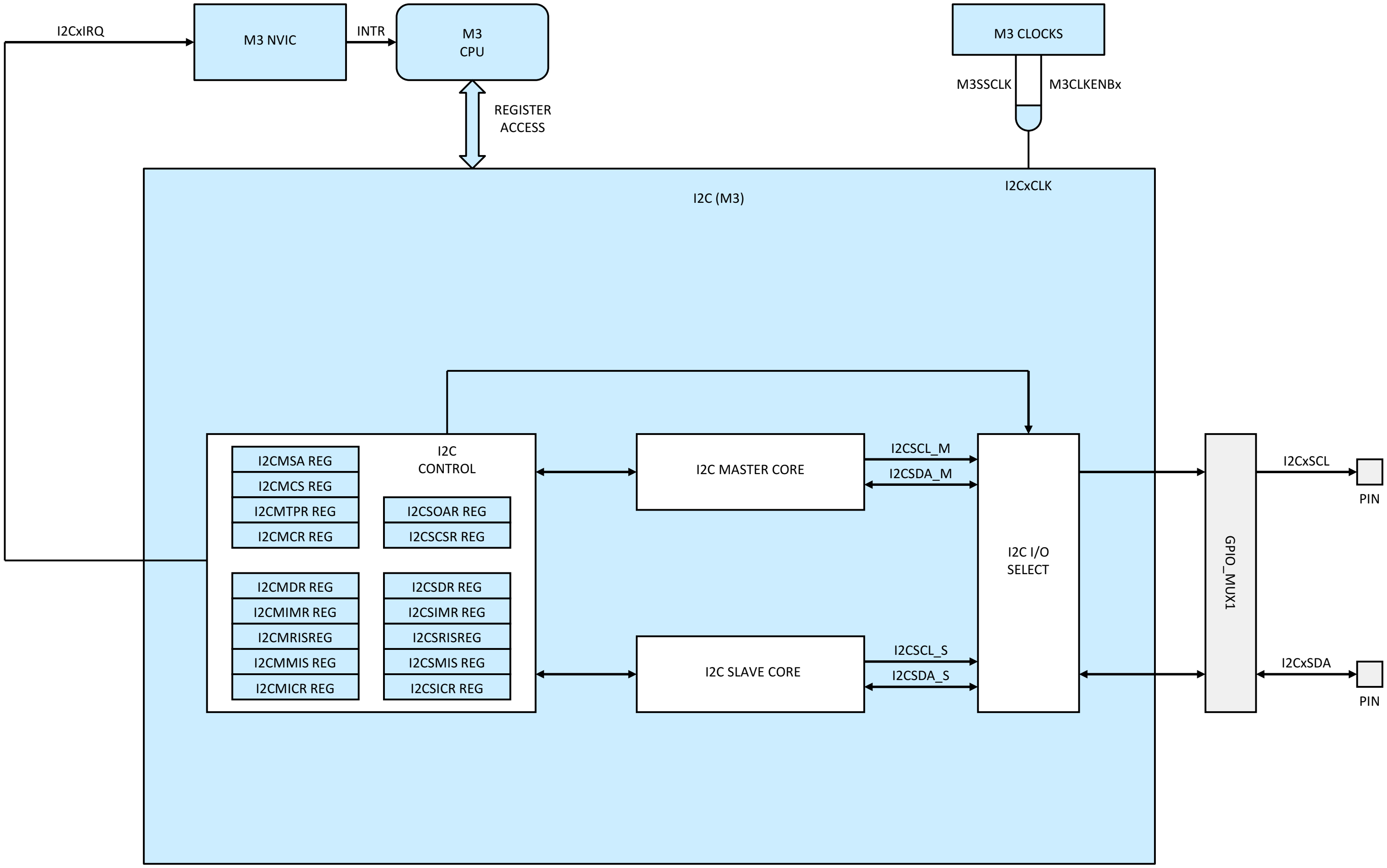


Figure 6-13. I2C (Cortex™-M3)

6.2.4 Cortex™-M3 Controller Area Network (CAN)

This device has two Cortex™-M3 Controller Area Network (CAN) peripherals. CAN is a serial communications protocol that efficiently supports distributed real-time control with a high level of security. The CAN module supports bit rates up to 1 Mbit/s and is compliant with the CAN 2.0B protocol specification.

CAN implements the following features:

- CAN protocol version 2.0 part A, B
- Bit rates up to 1 Mbit/s
- Multiple clock sources
- 32 message objects
- Individual identifier mask for each message object
- Programmable FIFO mode for message objects
- Programmable loop-back modes for self-test operation
- Suspend mode for debug support
- Software module reset
- Automatic bus on after Bus-Off state by a programmable 32-bit timer
- Message RAM parity check mechanism
- Two interrupt lines
- Global power down and wakeup support

Figure 6-14 shows the Cortex™-M3 CAN peripheral.

6.2.4.1 Functional Overview

CAN performs CAN protocol communication according to ISO 11898-1 (identical to Bosch® CAN protocol specification 2.0 A, B). The bit rate can be programmed to values up to 1 Mbit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

For communication on a CAN network, individual message objects can be configured. The message objects and identifier masks are stored in the Message RAM. All functions concerning the handling of messages are implemented in the message handler. Those functions are: acceptance filtering, the transfer of messages between the CAN Core and the Message RAM, and the handling of transmission requests.

The register set of the CAN is accessible directly by the CPU via the module interface. These registers are used to control/configure the CAN Core and the message handler, and to access the message RAM.

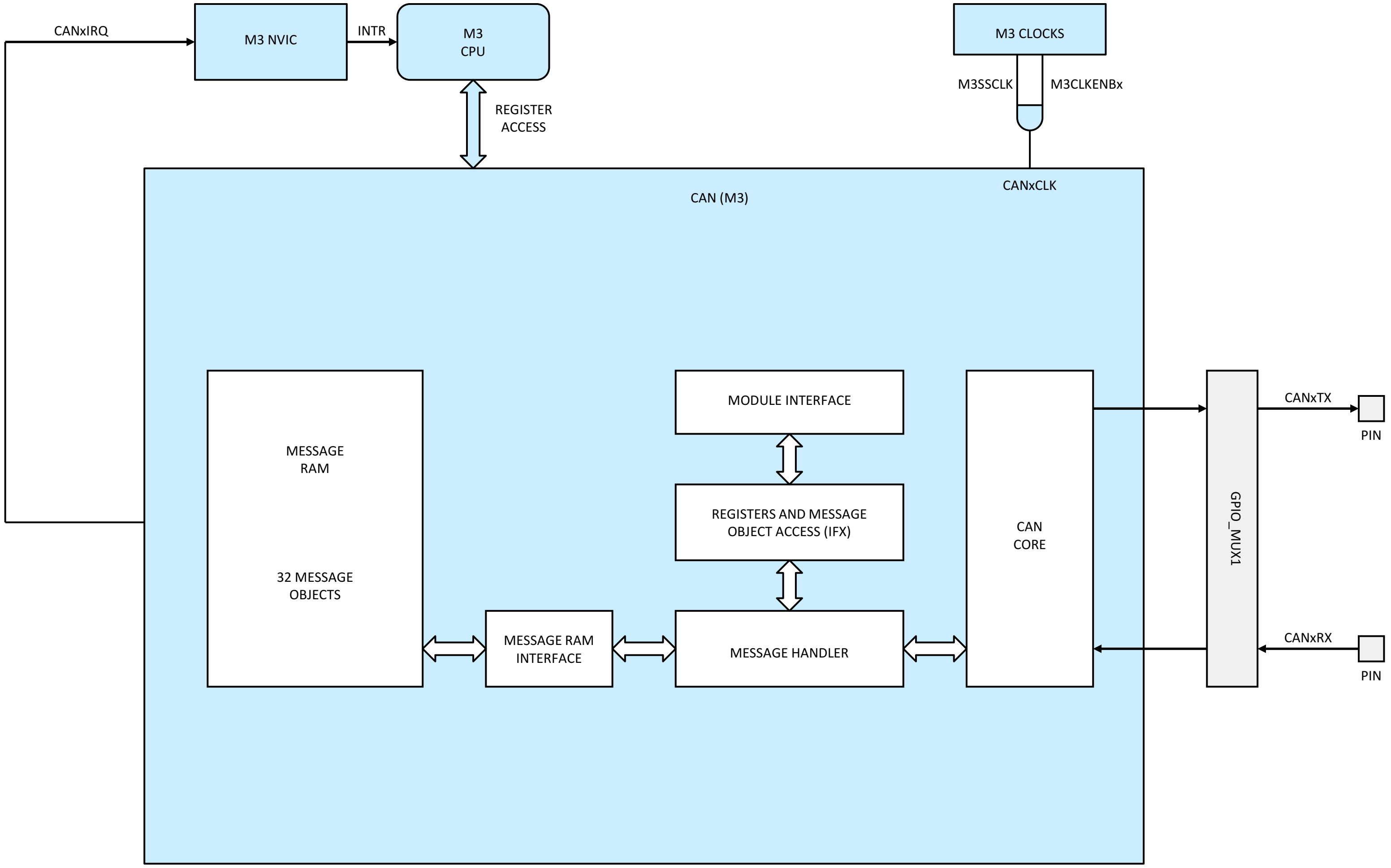


Figure 6-14. CAN (Cortex™-M3)

ADVANCE INFORMATION

6.2.5 Cortex™-M3 Universal Serial Bus (USB) Controller

This device has one Cortex™-M3 USB controller. The USB controller operates as a full-speed or low-speed function controller during point-to-point communications with the USB Host, Device, or OTG functions. The controller complies with the USB 2.0 standard, which includes SUSPEND and RESUME signaling. Thirty-two endpoints, which comprised of 2 hardwired endpoints for control transfers (one endpoint for IN and one endpoint for OUT) and 30 endpoints defined by firmware, along with a dynamic sizable FIFO, support multiple packet queuing. DMA access to the FIFO allows minimal interference from system software. Software-controlled connect and disconnect allow flexibility during USB device start-up. The controller complies with the OTG standard's Session Request Protocol (SRP) and Host Negotiation Protocol (HNP).

The USB controller includes the following features:

- Complies with USB-IF certification standards
- USB 2.0 full-speed (12-Mbps) and low-speed (1.5-Mbps) operation
- Integrated PHY
- Four transfer types: Control, Interrupt, Bulk, and Isochronous
- 32 endpoints:
  - One dedicated control IN endpoint and one dedicated control OUT endpoint
  - 15 configurable IN endpoints and 15 configurable OUT endpoints
- 4KB dedicated endpoint memory: one endpoint may be defined for double-buffered 1023-byte isochronous packet size
- VBUS droop and valid ID detection and interrupt
- Efficient transfers using direct memory access controller (DMA):
  - Separate channels for transmit and receive for up to three IN endpoints and three OUT endpoints
  - Channel requests asserted when FIFO contains required amount of data

Figure 6-15 shows the USB peripheral.

6.2.5.1 Functional Description

The USB controller provides full OTG negotiation by supporting both the Session Request Protocol (SRP) and the Host Negotiation Protocol (HNP). The SRP allows devices on the B side of a cable to request the A-side devices' turn on VBUS. The HNP is used after the initial session request protocol has powered the bus and provides a method to determine which end of the cable will act as the Host controller. When the device is connected to non-OTG peripherals or devices, the controller can detect which cable end was used and provides a register to indicate if the controller should act as the Host controller or the Device controller. This indication and the mode of operation are handled automatically by the USB controller. This autodetection allows the system to use a single A/B connector instead of having both A and B connectors in the system, and supports full OTG negotiations with other OTG devices.

In addition, the USB controller provides support for connecting to non-OTG peripherals or Host controllers. The USB controller can be configured to act as either a dedicated Host or Device, in which case, the USB0VBUS and USB0ID signals can be used as GPIOs. However, when the USB controller is acting as a self-powered Device, a GPIO input must be connected to VBUS and configured to generate an interrupt when the VBUS level drops. This interrupt is used to disable the pullup resistor on the USB0DP signal.

NOTE

When the USB is used in the system, the minimum system frequency is 20 MHz.

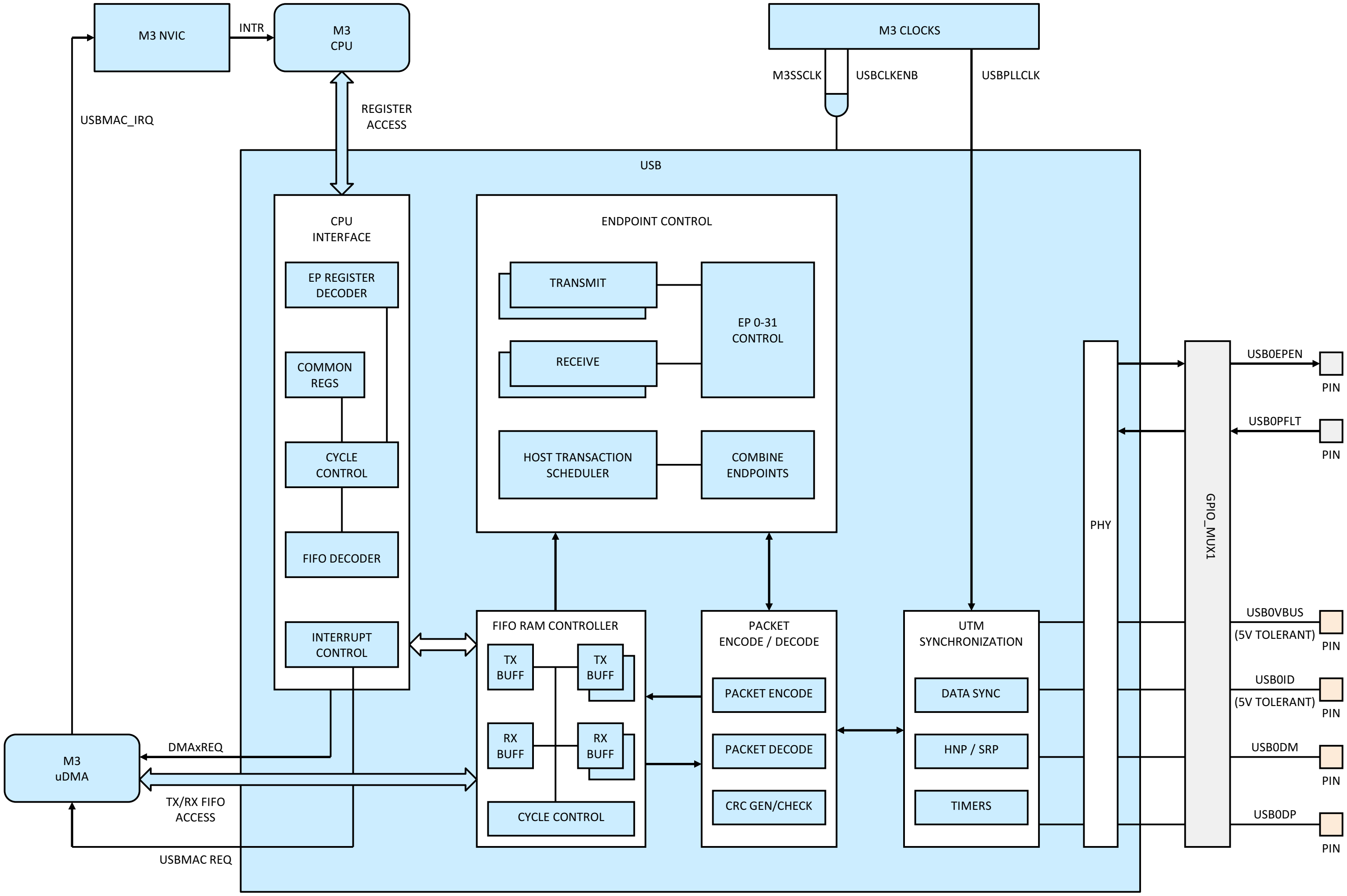


Figure 6-15. USB

ADVANCE INFORMATION

6.2.6 Cortex™-M3 Ethernet Media Access Controller (EMAC)

The Cortex™-M3 Ethernet Media Access Controller (EMAC) conforms to IEEE 802.3 specifications and fully supports 10BASE-T and 100BASE-TX standards. This device has one Ethernet Media Access Controller.

The EMAC module has the following features:

- Conforms to the IEEE 802.3-2002 specification
  - 10BASE-T/100BASE-TX IEEE-802.3 compliant
- Multiple operational modes
  - Full- and half-duplex 100-Mbps
  - Full- and half-duplex 10-Mbps
  - Power-saving and power-down modes
- Highly configurable:
  - Programmable MAC address
  - Promiscuous mode support
  - CRC error-rejection control
  - User-configurable interrupts
- IEEE 1588 Precision Time Protocol: Provides highly accurate time stamps for individual packets
- Efficient transfers using the Micro Direct Memory Access Controller (µDMA)
  - Separate channels for transmit and receive
  - Receive channel request asserted on packet receipt
  - Transmit channel request asserted on empty transmit FIFO

Figure 6-16 shows the EMAC peripheral.

6.2.6.1 Functional Overview

The Ethernet Controller is functionally divided into two layers: the Media Access Controller (MAC) layer and the Network Physical (PHY) layer. The MAC resides inside the device, and the PHY outside of the device. These layers correspond to the OSI model layers 2 and 1, respectively. The CPU accesses the Ethernet Controller via the MAC layer. The MAC layer provides transmit and receive processing for Ethernet frames. The MAC layer also provides the interface to the external PHY layer via an internal Media Independent Interface (MII). The PHY layer communicates with the Ethernet bus.

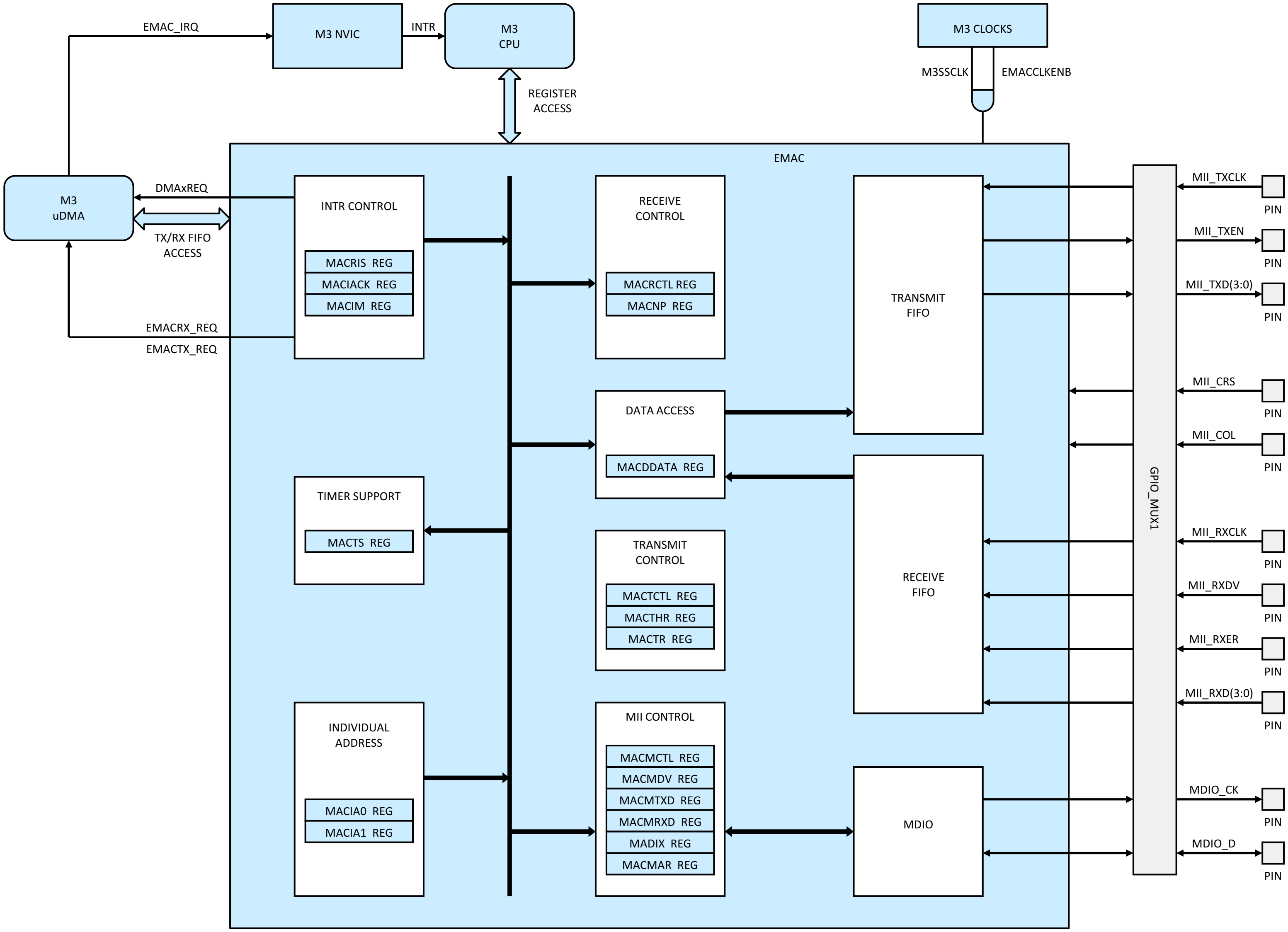


Figure 6-16. EMAC

6.2.6.2 MII Signals

The individual EMAC and MDIO signals for the MII interface are summarized in Table 6-15.

Table 6-15. EMAC and MDIO Signals for MII Interface

SIGNAL	TYPE[1]	DESCRIPTION
MII_TXCK	I	Transmit clock. The transmit clock is a continuous clock that provides the timing reference for transmit operations. The MII_TXD and MII_TXEN signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10-Mbps operation and 25 MHz at 100-Mbps operation.
MII_TXD[3-0]	O	Transmit data. The transmit data pins are a collection of four data signals comprising 4 bits of data. MTDX0 is the least-significant bit (LSB). The signals are synchronized by MII_TXCLK and are valid only when MII_TXEN is asserted.
MII_TXEN	O	Transmit enable. The transmit enable signal indicates that the MII_TXD pins are generating nibble data for use by the PHY. MII_TXEN is driven synchronously to MII_TXCLK.
MII_COL	I	Collision detected. In half-duplex operation, the MII_COL pin is asserted by the PHY when the PHY detects a collision on the network. The MII_COL pin remains asserted while the collision condition persists. This signal is not necessarily synchronous to MII_TXCLK or MII_RXCLK. In full-duplex operation, the MII_COL pin is used for hardware transmit flow control. Asserting the MII_COL pin will stop packet transmissions; packets in the process of being transmitted when MII_COL is asserted will complete transmission. The MII_COL pin should be held low if hardware transmit flow control is not used.
MII_CRS	I	Carrier sense. In half-duplex operation, the MII_CRS pin is asserted by the PHY when the network is not idle in either transmit or receive. The pin is deasserted when both transmit and receive are idle. This signal is not necessarily synchronous to MII_TXCLK or MII_RXCLK. In full-duplex operation, the MII_CRS pin should be held low.
MII_RXCK	I	Receive clock. The receive clock is a continuous clock that provides the timing reference for receive operations. The MII_RXD, MII_RXDV, and MII_RXER signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10-Mbps operation and 25 MHz at 100-Mbps operation.
MII_RXD[3-0]	I	Receive data. The receive data pins are a collection of four data signals comprising 4 bits of data. MRDX0 is the least-significant bit (LSB). The signals are synchronized by MII_RXCLK and are valid only when MII_RXDV is asserted.
MII_RXDV	I	Receive data valid. The receive data valid signal indicates that the MII_RXD pins are generating nibble data for use by the EMAC. MII_RXDV is driven synchronously to MII_RXCLK.
MII_RXER	I	Receive error. The receive error signal is asserted for one or more MII_RXCLK periods to indicate that an error was detected in the received frame. The MII_RXER signal being asserted is meaningful only during data reception when MII_RXDV is active.
MDIO_CK	O	Management data clock. The MDIO data clock is sourced by the MDIO module on the system. MDIO_CK is used to synchronize MDIO data access operations done on the MDIO pin. The frequency of this clock is controlled by the CLKDIV bits in the MDIO Control Register (CONTROL).
MDIO_D	I/O	Management data input output. The MDIO data pin drives PHY management data into and out of the PHY by way of an access frame that consists of start-of-frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles, at which time the pin is an input for read operations.

(1) I = Input, O = Output, I/O = Input/Output

6.3 Control Subsystem Peripherals

Control Subsystem peripherals are accessible from the C28x CPU via the C28x Memory Bus, and from the C28x DMA via the C28x DMA Bus. They include one NMI Watchdog, three Timers, four Serial Port Peripherals (SCI, SPI, McBSP, I2C), and three types of Control Peripherals (ePWM, eQEP, eCAP). Additionally, the C28x CPU/DMA also have access to the External Peripheral Interface (EPI), and to Analog and Shared peripherals (see Section 6.1).

For detailed information on the processor peripherals, see the *Concerto F28M36x Technical Reference Manual* (literature number SPRUHE8).

6.3.1 High-Resolution PWM (HRPWM) and Enhanced PWM (ePWM) Modules

There are 12 PWM modules in the Concerto device. Eight of these are of the High-Resolution PWM (HRPWM) type with high-resolution control on both A and B signal outputs, and four are of the Enhanced PWM (ePWM) type. The HRPWM modules have all the features of the ePWM plus they offer significantly higher PWM resolution (time granularity on the order of 150 ps). Figure 6-17 shows the eight HRPWM modules (PWM 1–8) and four ePWM modules (PWM 9–PWM12).

The synchronization inputs to the PWM modules include the SYNCI signal from the GPTRIP1 output of GPIO_MUX1, and the TBCLKSYNC signal from the CPCLKCR0 register. Synchronization output SYNCO1 comes from the ePWM1 module and is stretched by 8 HSPCLK cycles before entering GPIO_MUX1. There are two groups of trip signal inputs to PWM modules. TRIP1–15 inputs come from GPTRIP1–12 (from GPIO_MUX1), ECCDBLERR signal (from C28x Local and Shared RAM), and PIEERR signal from the C28x CPU. TZ1–6 (Trip Zone) inputs come from GPTRIP 1–3 (from GPIO_MUX1), EQEPERR (from the eQEP peripheral), CLOCKFAIL (from M3 CLOCKS), and EMUSTOP (from the C28x CPU).

There are 12 SOCA PWM outputs and 12 SOCB PWM outputs—a pair from each PWM module. The 12 SOCA outputs are OR-ed together and stretched by 32 HSPCLK cycles before entering GPIO_MUX1 as a single SOCAO signal. The 12 SOCB outputs are OR-ed together and stretched by 32 HSPCLK cycles before entering GPIO_MUX1 as a single SOCBO signal. The 18 SOCA/B outputs from PWM1–PWM9 also go to the Analog Subsystem, where they can be selected to become conversion triggers to ADC modules.

The 12 PWM modules also drive two other sets of outputs which can interrupt the C28x CPU via the C28x PIE block. These are 12 EPWMINT interrupts and 12 EPWMTZINT trip-zone interrupts. See Figure 6-18 for the internal structure of the HRPWM and ePWM modules. The green-colored blocks are common to both ePWM and HRPWM modules, but only the HRPWMs have the grey-colored hi-resolution blocks.

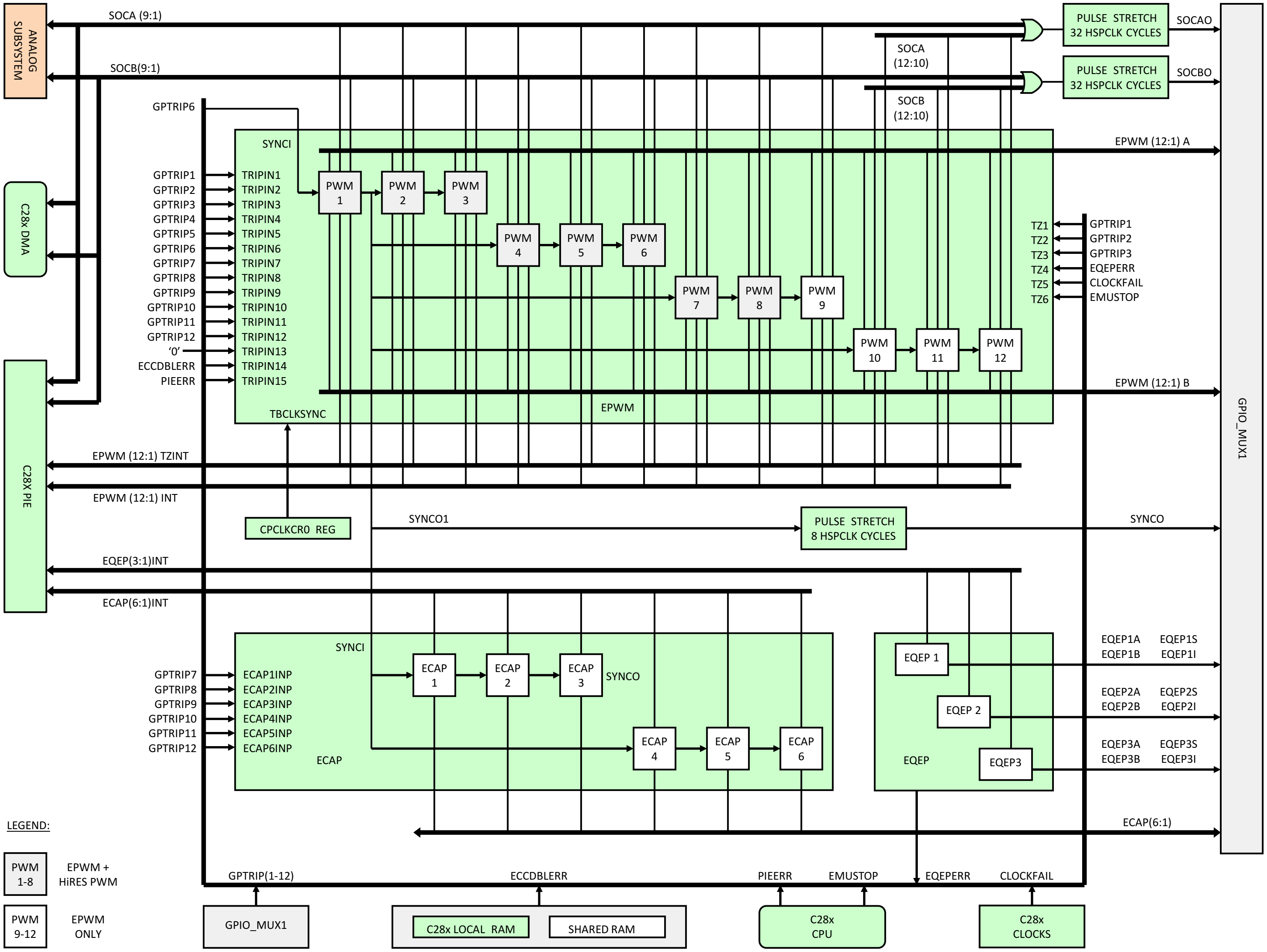
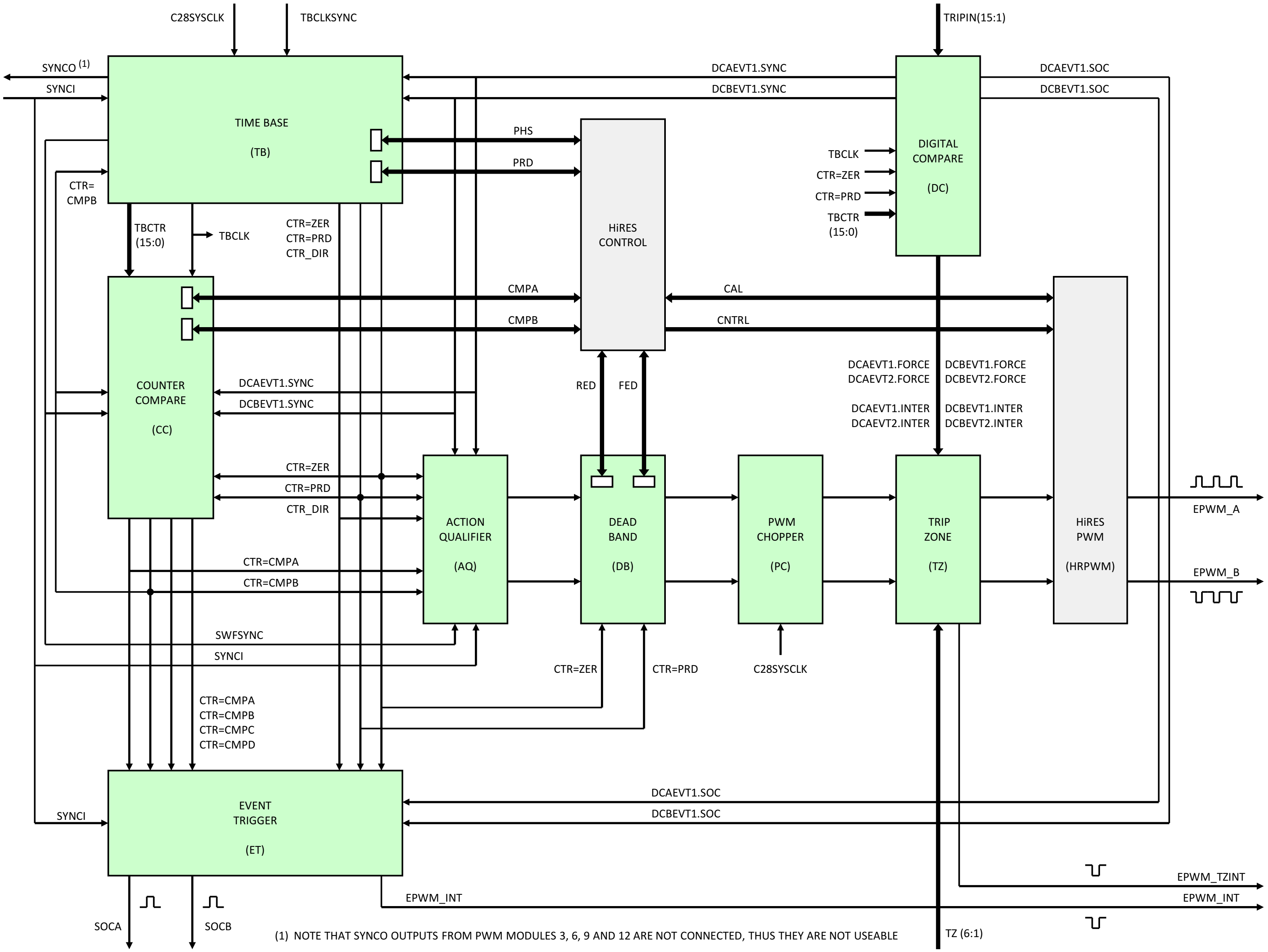


Figure 6-17. PWM, eCAP, eQEP



(1) NOTE THAT SYNCO OUTPUTS FROM PWM MODULES 3, 6, 9 AND 12 ARE NOT CONNECTED, THUS THEY ARE NOT USEABLE

Figure 6-18. Internal Structure of PWM

6.3.2 Enhanced Capture (eCAP) Module

There are six identical eCAP modules in Concerto devices: eCAP1, 2, 3, 4, 5, and 6. Each eCAP module represents one complete capture channel. Its main function is to accurately capture the timings of external events. One can also use eCAP modules for PWM, when they are not being used for input captures. This secondary function is selected by flipping the CAP/APWM bit of the ECCTL2 Register. For PWM function, the counter operates in count-up mode, providing a time base for asymmetrical pulse width (PWM) waveforms. The CAP1 and CAP2 registers become the period and compare registers, respectively; while the CAP3 and CAP4 registers become the shadow registers of the main period and capture registers, respectively.

The left side of Figure 6-19 shows internal components associated with the capture block, and the right side depicts the PWM block. The two blocks share a set of four registers that are used in both Capture and PWM modes. Other components include the Counter block that uses the SYNCIN and SYNCOUT ports to synchronize with other modules; and the Interrupt Trigger and Flag Control block that sends Capture, PWM, and Counter events to the C28x PIE block via the ECAPxINT output. There are six ECAPxINT interrupts—one for each eCAP module.

The eCAP peripherals are clocked by C28SYSCLK, and its registers are accessible by the C28x CPU. This peripheral clock can be enabled or disabled by flipping a bit in one of the system control registers.

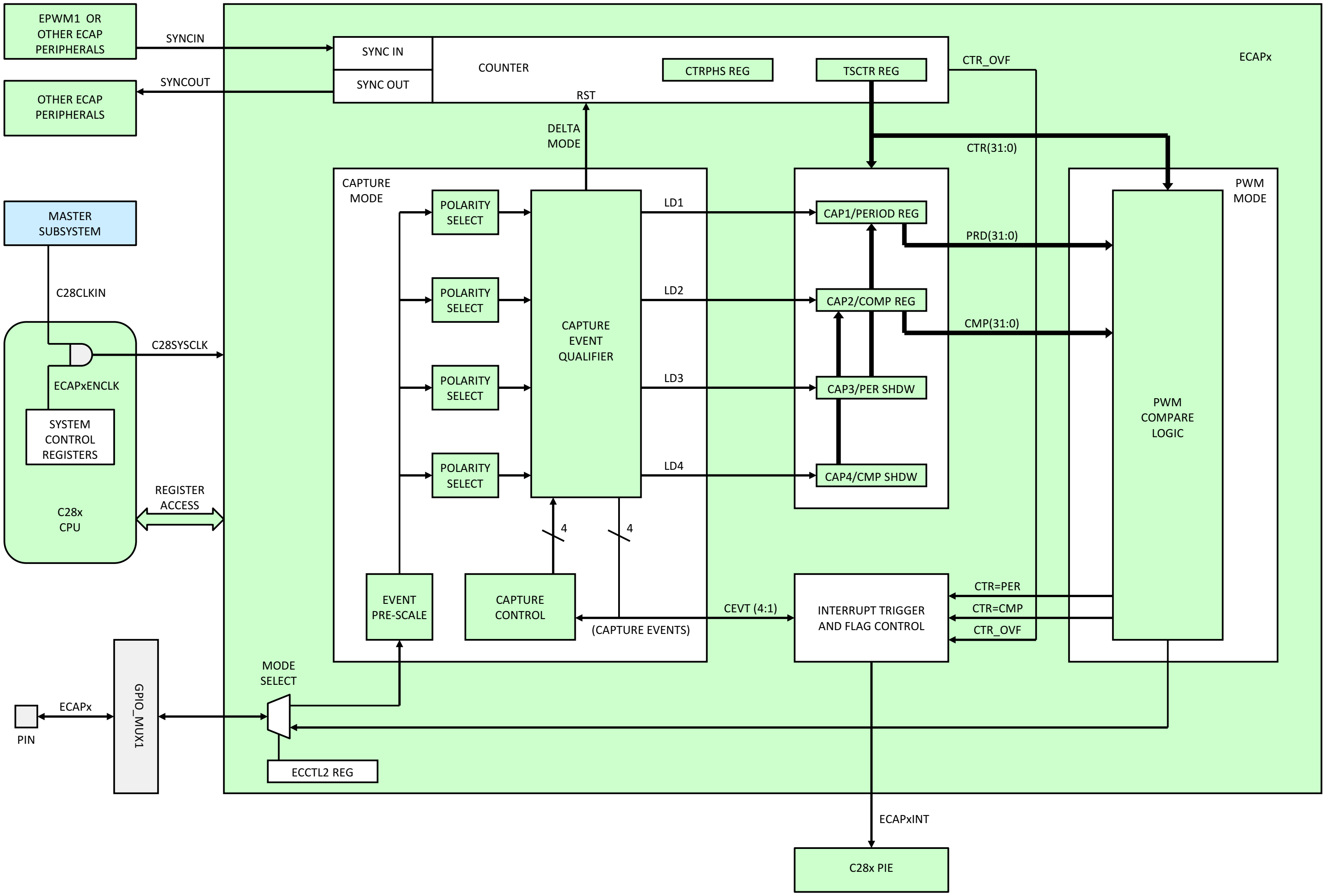


Figure 6-19. eCAP

6.3.3 Enhanced Quadrature Encoder Pulse (eQEP) Module

The Enhanced Quadrature Encoder Pulse (eQEP) module interfaces directly with linear or rotary incremental encoders to obtain position, direction, and speed information from rotating machines used in high-performance motion and position-control systems. There are three Type 0 eQEP modules in each Concerto device.

Each eQEP peripheral comprises five major functional blocks: Quadrature Capture Unit (QCAP), Position Counter/Control Unit (PCCU), Quadrature Decoder (QDU), Unit Time Base for speed and frequency measurement (UTIME), and Watchdog timer for detecting stalls (QWDOG). The C28x CPU controls and communicates with these modules through a set of associated registers (see Figure 6-20). The eQEP peripherals are clocked by C28SYSCLK, and its registers are accessible by the C28x CPU. This peripheral clock can be enabled or disabled by flipping a bit in one of the system control registers.

Each eQEP peripheral connects through the GPIO_MUX1 block to four device pins. Two of the four pins are always inputs, while the other two can be inputs or outputs, depending on the operating mode. The PCCU block of each eQEP also drives one interrupt to the C28x PIE. There is a total of three EQEPxINT interrupts—one from each of the three eQEP modules.

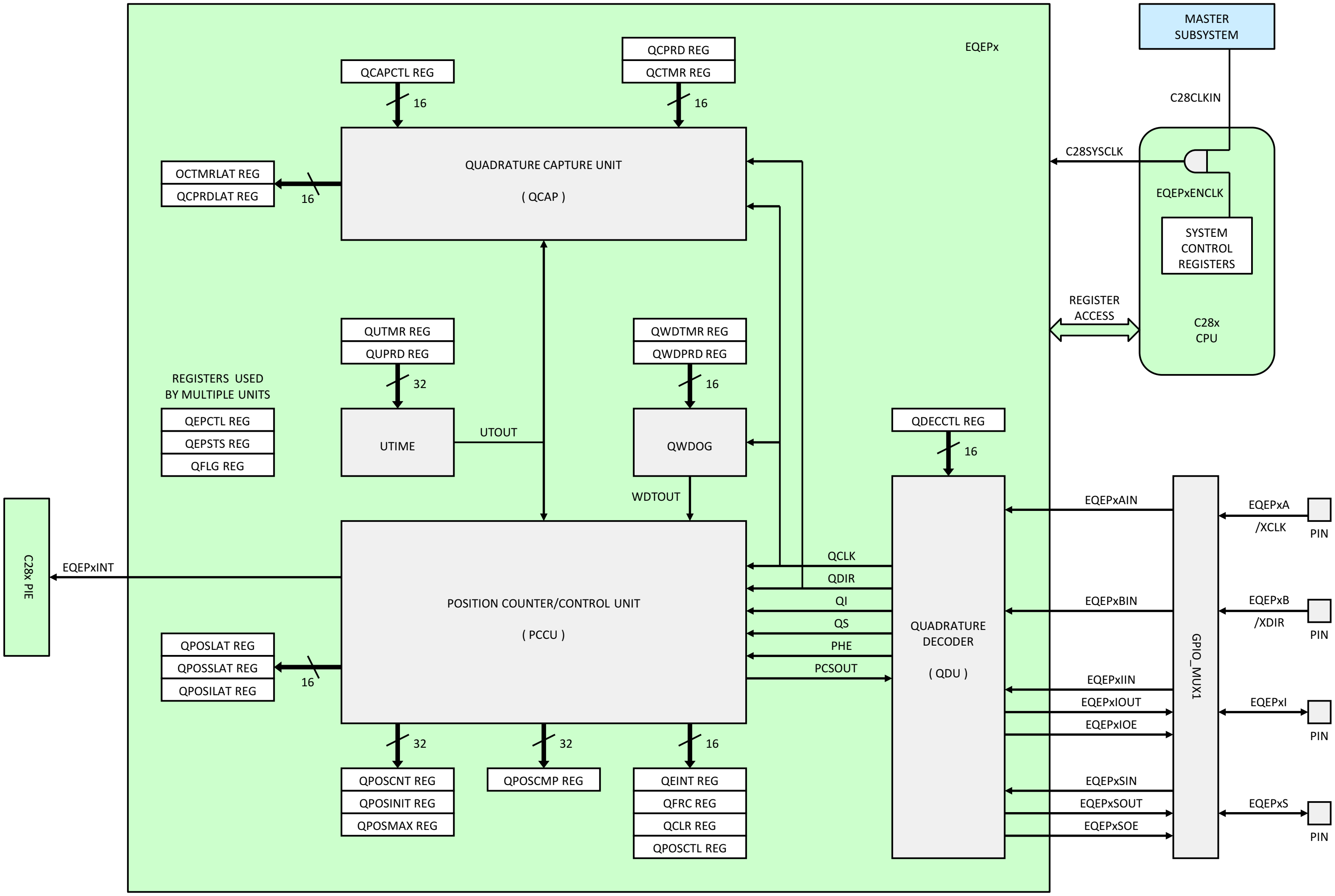


Figure 6-20. eQEP

6.3.4 C28x Inter-Integrated Circuit Module (I2C)

This device has one C28x inter-integrated circuit (I2C) peripheral. The I2C provides an interface between a Concerto device and devices compliant with the Philips® *I^{2}C-Bus Specification Version 2.1* and connected by way of an I^{2}C Bus®. External components attached to this 2-wire serial bus can transmit 1-bit to 8-bit data to and receive 1-bit to 8-bit data from the device through the I2C module.

NOTE

A unit of data transmitted or received by the I2C module can have fewer than 8 bits; however, for convenience, a unit of data is called a ***data byte*** in this section. The number of bits in a data byte is selectable via the BC bits of the mode register, I2CMDR.

The I2C module has the following features:

- Compliance with the Philips® *I^{2}C-Bus Specification Version 2.1*:
  - Support for 1-bit to 8-bit format transfers
  - 7-bit and 10-bit addressing modes
  - General call
  - START byte mode
  - Support for multiple master-transmitters and slave-receivers
  - Support for multiple slave-transmitters and master-receivers
  - Combined master transmit-and-receive and receive-and-transmit mode
  - Data transfer rate of from 10 Kbps up to 400 Kbps (I2C Fast-mode rate)
- One 4-word receive FIFO and one 4-word transmit FIFO
- One interrupt that can be used by the CPU. This interrupt can be generated as a result of one of the following conditions:
  - Transmit-data ready
  - Receive-data ready
  - Register-access ready
  - No-acknowledgment received
  - Arbitration lost
  - Stop condition detected
  - Addressed as slave
- An additional interrupt that can be used by the CPU when in FIFO mode
- Module enable or disable capability
- Free data format mode

The I2C module does not support:

- High-speed mode (Hs-mode)
- CBUS-compatibility mode

Figure 6-21 shows the C28x I2C peripheral.

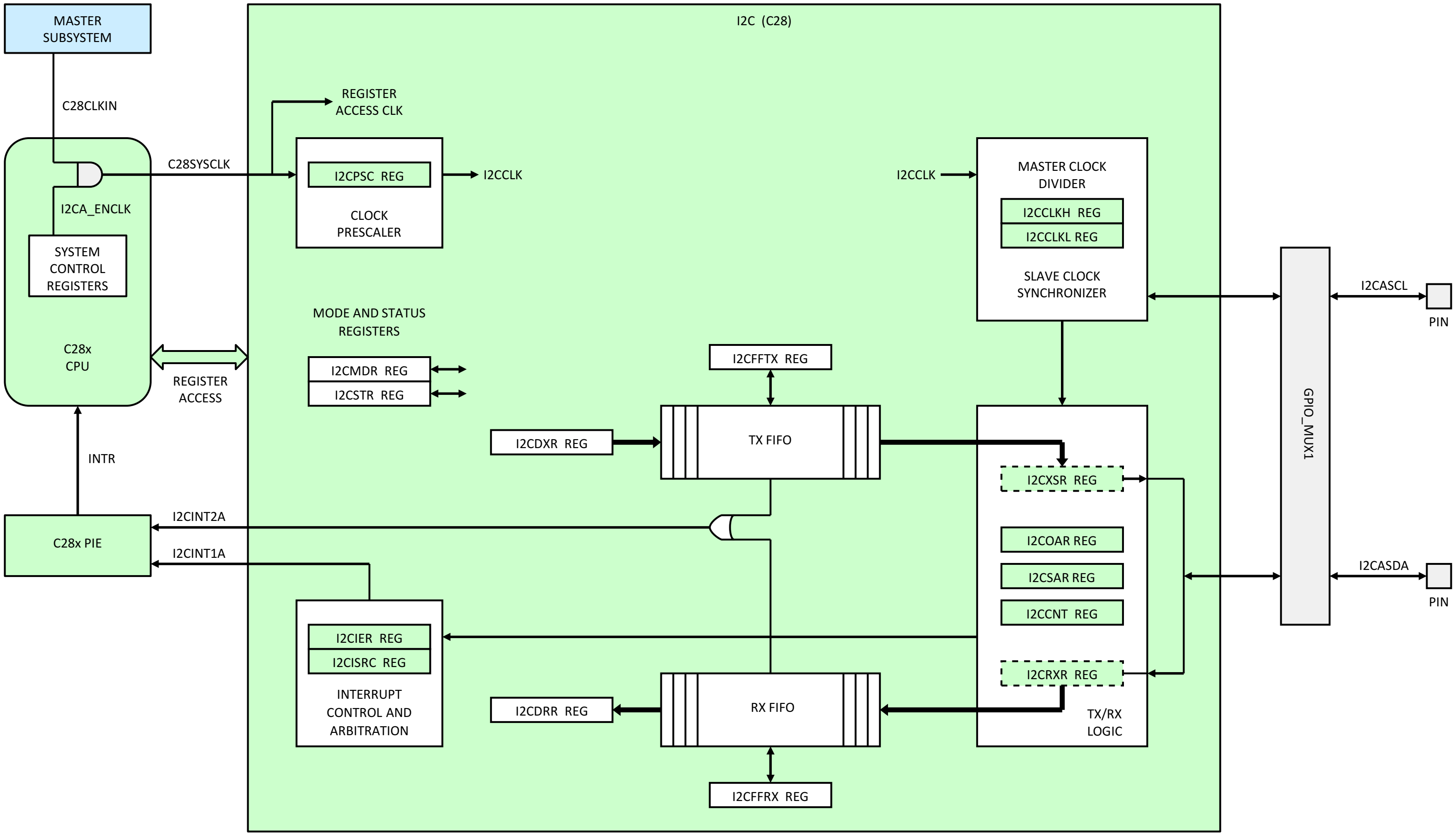


Figure 6-21. I2C (C28x)

ADVANCE INFORMATION

6.3.4.1 Functional Overview

Each device connected to an I^{2}C Bus is recognized by a unique address. Each device can operate as either a transmitter or a receiver, depending on the function of the device. A device connected to the I^{2}C Bus can also be considered as the master or the slave when performing data transfers. A master device is the device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. During this transfer, any device addressed by this master is considered a slave. The I2C module supports the multi-master mode, in which one or more devices capable of controlling an I^{2}C Bus can be connected to the same I^{2}C Bus.

For data communication, the I2C module has a serial data pin (SDA) and a serial clock pin (SCL). These two pins carry information between the C28x device and other devices connected to the I^{2}C Bus. The SDA and SCL pins both are bidirectional. They each must be connected to a positive supply voltage using a pullup resistor. When the bus is free, both pins are high. The driver of these two pins has an open-drain configuration to perform the required wired-AND function. There are two major transfer techniques:

1. Standard Mode: Send exactly n data values, where n is a value you program in an I2C module register.
2. Repeat Mode: Keep sending data values until you use software to initiate a STOP condition or a new START condition.

The I2C module consists of the following primary blocks:

- A serial interface: one data pin (SDA) and one clock pin (SCL)
- Data registers and FIFOs to temporarily hold receive data and transmit data traveling between the SDA pin and the CPU
- Control and status registers
- A peripheral bus interface to enable the CPU to access the I2C module registers and FIFOs.

6.3.4.2 Clock Generation

The device clock generator receives a signal from an external clock source and produces an I2C input clock with a programmed frequency. The I2C input clock is equivalent to the CPU clock and is then divided twice more inside the I2C module to produce the module clock and the master clock.

6.3.5 C28x Serial Communications Interface (SCI)

This device has one serial communication interface (SCI) peripheral. SCI is a two-wire asynchronous serial port, commonly known as a UART. The SCI module supports digital communications between the CPU and other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format

The SCI receiver and transmitter each have a 16-level-deep FIFO for reducing servicing overhead, and each has its own separate enable and interrupt bits. Both can be operated independently for half-duplex communication, or simultaneously for full-duplex communication. To specify data integrity, the SCI checks received data for break detection, parity, overrun, and framing errors. The bit rate is programmable to different speeds through a 16-bit baud-select register.

Features of the SCI module include:

- Two external pins:
  - SCITXD: SCI transmit-output pin
  - SCIRXD: SCI receive-input pin

    NOTE: Both pins can be used as GPIO if not used for SCI.
  - Baud rate programmable to 64K different rates
- Data-word format
  - One start bit
  - Data-word length programmable from one to eight bits
  - Optional even/odd/no parity bit
  - One or two stop bits
- Four error-detection flags: parity, overrun, framing, and break detection
- Two wake-up multiprocessor modes: idle-line and address bit
- Half- or full-duplex operation
- Double-buffered receive and transmit functions
- Transmitter and receiver operations can be accomplished through interrupt-driven or polled algorithms with status flags.
  - Transmitter: TXRDY flag (transmitter-buffer register is ready to receive another character) and TX EMPTY flag (transmitter-shift register is empty)
  - Receiver: RXRDY flag (receiver-buffer register is ready to receive another character), BRKDT flag (break condition occurred), and RX ERROR flag (monitoring four interrupt conditions)
- Separate enable bits for transmitter and receiver interrupts (except BRKDT)
- NRZ (non-return-to-zero) format

NOTE

All registers in this module are 8-bit registers that are connected to Peripheral Frame 2. When a register is accessed, the register data is in the lower byte (bits 7–0), and the upper byte (bits 15–8) is read as zeros. Writing to the upper byte has no effect.

- Auto baud-detect hardware logic
- 4-level transmit and receive FIFO

Figure 6-22 shows the C28x SCI peripheral.

ADVANCE INFORMATION

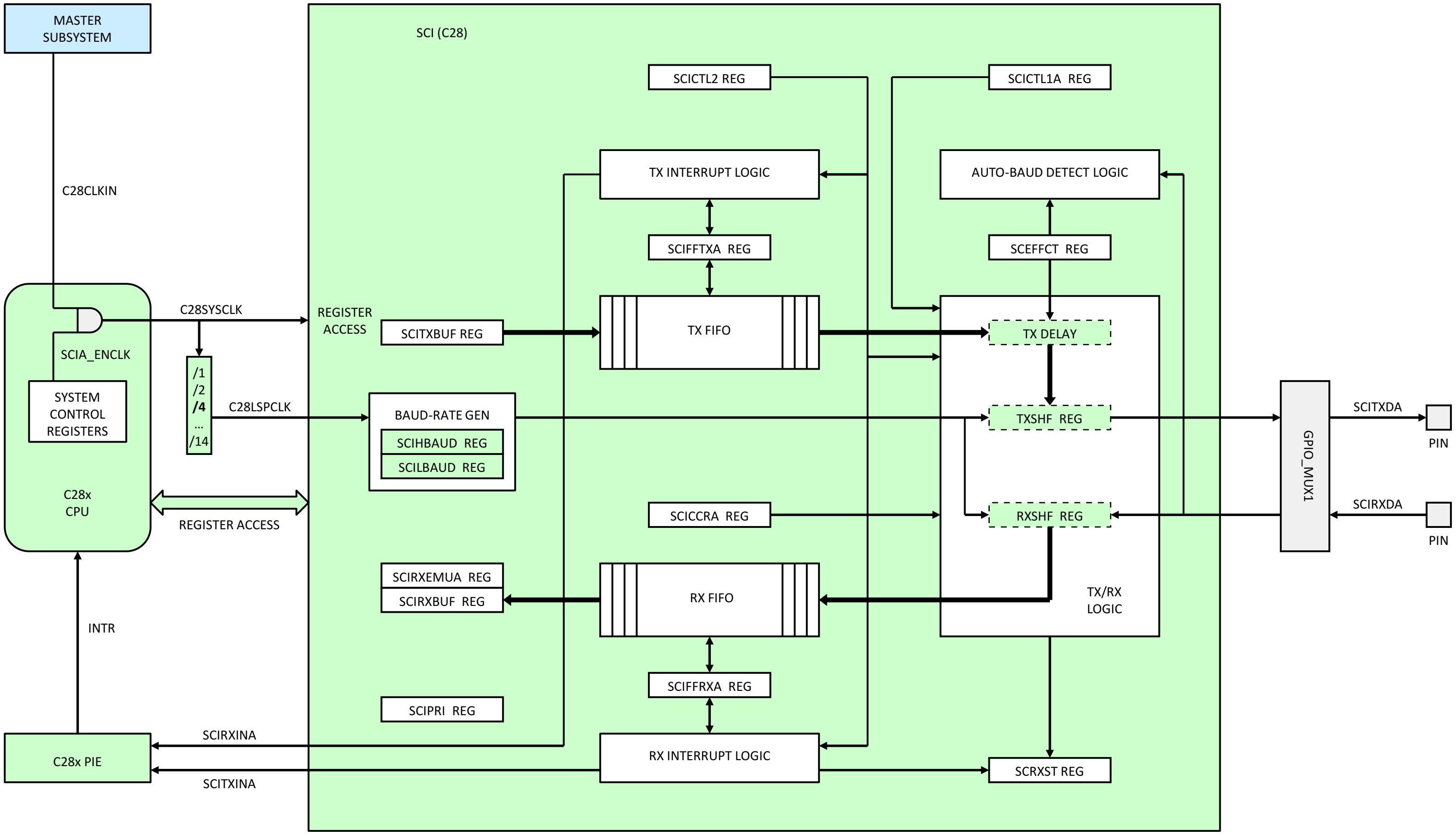


Figure 6-22. SCI (C28x)

6.3.5.1 Architecture

The major elements used in full-duplex operation include:

- A transmitter (TX) and its major registers:
  - SCITXBUF register – Transmitter Data Buffer register. Contains data (loaded by the CPU) to be transmitted
  - TXSHF register – Transmitter Shift register. Accepts data from the SCITXBUF register and shifts data onto the SCITXD pin, one bit at a time
- A receiver (RX) and its major registers:
  - RXSHF register – Receiver Shift register. Shifts data in from the SCIRXD pin, one bit at a time
  - SCIRXBUF register – Receiver Data Buffer register. Contains data to be read by the CPU. Data from a remote processor is loaded into the RXSHF register and then into the SCIRXBUF and SCIRXEMU registers
- A programmable baud generator
- Data-memory-mapped control and status registers enable the CPU to access the I2C module registers and FIFOs.

The SCI receiver and transmitter can operate either independently or simultaneously.

6.3.5.2 Multiprocessor and Asynchronous Communication Modes

The SCI has two multiprocessor protocols: the idle-line multiprocessor mode and the address-bit multiprocessor mode. These protocols allow efficient data transfer between multiple processors.

The SCI offers the universal asynchronous receiver/transmitter (UART) communications mode for interfacing with many popular peripherals. The asynchronous mode requires two lines to interface with many standard devices such as terminals and printers that use RS-232-C formats.

Data transmission characteristics include:

- One start bit
- One to eight data bits
- An even/odd parity bit or no parity bit
- One or two stop bits with a programmed frequency. The I2C input clock is equivalent to the CPU clock and is then divided twice more inside the I2C module to produce the module clock and the master clock.

6.3.6 C28x Serial Peripheral Interface (SPI)

This device has one C28x serial peripheral interface (SPI). The serial peripheral interface (SPI) is a high-speed synchronous serial input/output (I/O) port that allows a serial bit stream of programmed length (1 to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communications between the DSP controller and external peripherals or another controller. Typical applications include external I/O or peripheral expansion via devices such as shift registers, display drivers, and analog-to-digital converters (ADCs). Multi-device communications are supported by the master/slave operation of the SPI. The port supports a 16-level, receive-and-transmit FIFO for reducing CPU servicing overhead.

The SPI module features include:

- SPISOMI: SPI slave-output/master-input pin
- SPISIMO: SPI slave-input/master-output pin
- SPISTE: SPI slave transmit-enable pin
- SPICLK: SPI serial-clock pin

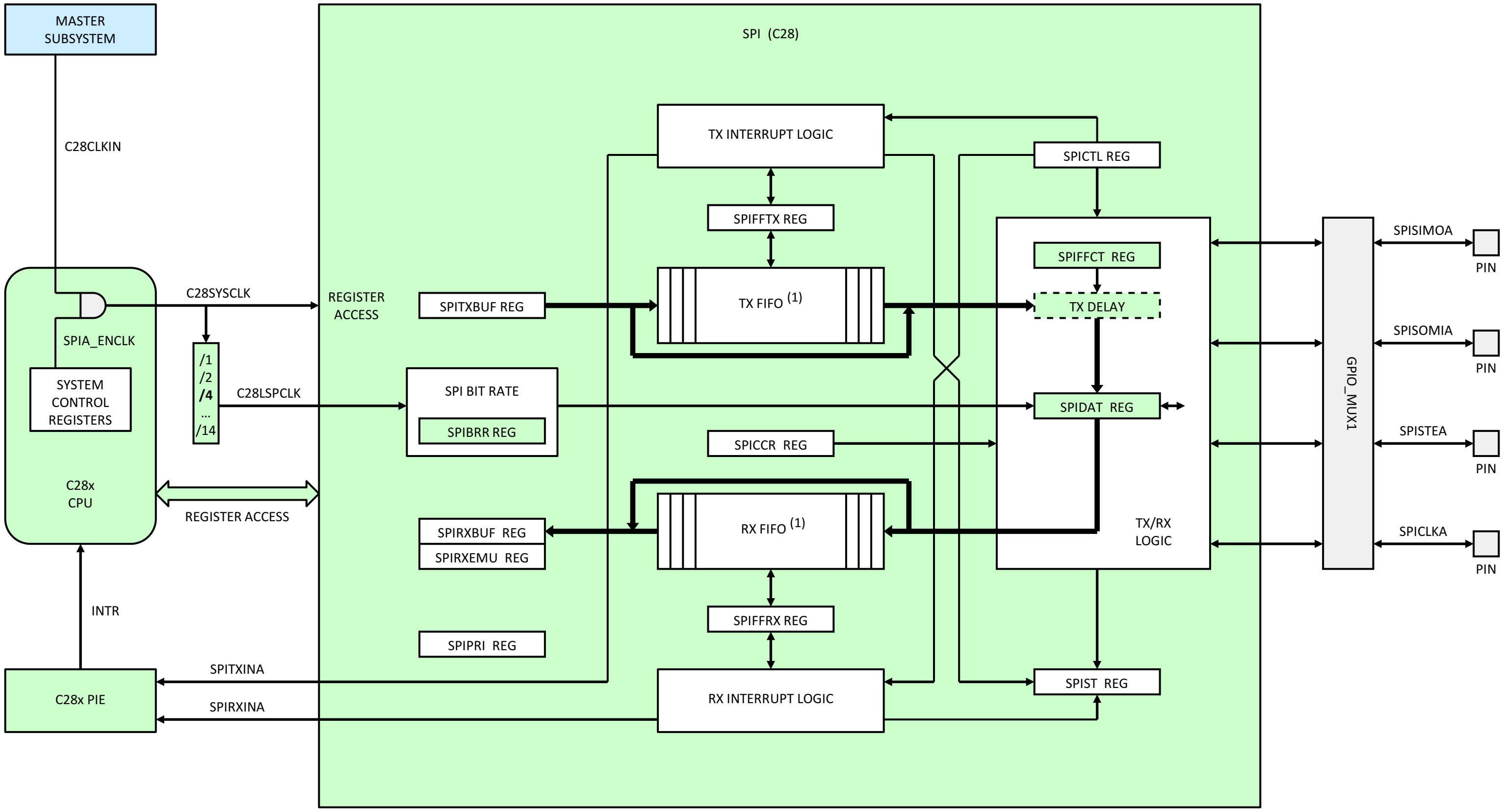
  NOTE: All four pins can be used as GPIO, if the SPI module is not used.
- Two operational modes: master and slave
- Baud rate: 125 different programmable rates. The maximum baud rate that can be employed is limited by the maximum speed of the I/O buffers used on the SPI pins.
- Data word length: 1 to 16 data bits
- Four clocking schemes (controlled by clock polarity and clock phase bits) include:
  - Falling edge without phase delay: SPICLK active-high. SPI transmits data on the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
  - Falling edge with phase delay: SPICLK active-high. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge without phase delay: SPICLK inactive-low. SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge with phase delay: SPICLK inactive-low. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
- Simultaneous receive-and-transmit operation (transmit function can be disabled in software)
- Transmitter and receiver operations are accomplished through either interrupt-driven or polled algorithms.
- Twelve SPI module control registers: Located in control register frame beginning at address 7040h.

NOTE

All registers in this module are 16-bit registers that are connected to Peripheral Frame 2. When a register is accessed, the register data is in the lower byte (bits 7−0), and the upper byte (bits 15−8) is read as zeros. Writing to the upper byte has no effect.

- 16-level transmit and receive FIFO
- Delayed transmit control

Figure 6-23 shows the C28x SPI peripheral.



(1) RX FIFO AND TX FIFO CAN BE BYPASSED BY CONFIGURING BIT SPIFFENA OF THE SPIFFTX REGISTER

Figure 6-23. SPI (C28x)

6.3.6.1 Functional Overview

The SPI operates in master or slave mode. The master initiates data transfer by sending the SPICLK signal. For both the slave and the master, data is shifted out of the shift registers on one edge of the SPICLK and latched into the shift register on the opposite SPICLK clock edge. If the CLOCK PHASE bit (SPICTL.3) is high, data is transmitted and received a half-cycle before the SPICLK transition. As a result, both controllers send and receive data simultaneously. The application software determines whether the data is meaningful or dummy data. There are three possible methods for data transmission:

- Master sends data; slave sends dummy data
- Master sends data; slave sends data
- Master sends dummy data; slave sends data

The master can initiate a data transfer at any time because it controls the SPICLK signal. The software, however, determines how the master detects when the slave is ready to broadcast data.

6.3.7 C28x Multichannel Buffered Serial Port (McBSP)

This device provides one high-speed multichannel buffered serial port (McBSP) that allows direct interface to codecs and other devices. The CPU accesses data, control, and status information. The MCBSP also supports µDMA transfers.

The McBSP consists of a data-flow path and a control path connected to external devices by six pins. Data is communicated to devices interfaced with the McBSP via the data transmit (DX) pin for transmission and via the data receive (DR) pin for reception. Control information in the form of clocking and frame synchronization is communicated via the following pins: CLKX (transmit clock), CLKR (receive clock), FSX (transmit frame synchronization), and FSR (receive frame synchronization).

The CPU and the DMA controller communicate with the McBSP through 16-bit-wide registers accessible via the internal peripheral bus. The CPU or the DMA controller writes the data to be transmitted to the data transmit registers (DXR1, DXR2). Data written to the DXRs is shifted out to DX via the transmit shift registers (XSR1, XSR2). Similarly, receive data on the DR pin is shifted into the receive shift registers (RSR1, RSR2) and copied into the receive buffer registers (RBR1, RBR2). The contents of the RBRs is then copied to the DRRs, which can be read by the CPU or the DMA controller. This method allows simultaneous movement of internal and external data communications.

DRR2, RBR2, RSR2, DXR2, and XSR2 are not used (written, read, or shifted) if the serial word length is 8 bits, 12 bits, or 16 bits. For larger word lengths, these registers are needed to hold the most significant bits.

The frame and clock loop-back is implemented at chip level to enable CLKX and FSX to drive CLKR and FSR. If the loop-back is enabled, the CLKR and FSR get their signals from the CLKX and FSX pads instead of the CLKR and FSR pins.

McBSP features include:

- Full-duplex communication
- Double-buffered transmission and triple-buffered reception, allowing a continuous data stream
- Independent clocking and framing for reception and transmission
- The capability to send interrupts to the CPU and to send DMA events to the DMA controller
- 128 channels for transmission and reception
- Multichannel selection modes that enable or disable block transfers in each of the channels
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected A/D and D/A devices
- Support for external generation of clock signals and frame-synchronization signals
- A programmable sample rate generator for internal generation and control of clock signals and frame synchronization signals
- Programmable polarity for frame-synchronization pulses and clock signals
- Direct interface to:
  - T1/E1 framers
  - IOM-2 compliant devices
  - AC97-compliant devices (the necessary multi-phase frame capability is provided)
  - I2S compliant devices
  - SPI devices
- A wide selection of data sizes: 8, 12, 16, 20, 24, and 32 bits

NOTE

A value of the chosen data size is referred to as a ***serial word*** or ***word*** in this section. Elsewhere, ***word*** is used to describe a 16-bit value.

- µ-law and A-law companding
- The option of transmitting/receiving 8-bit data with the LSB first
- Status bits for flagging exception/error conditions
- ABIS mode is not supported

Figure 6-24 shows the C28x McBSP peripheral.

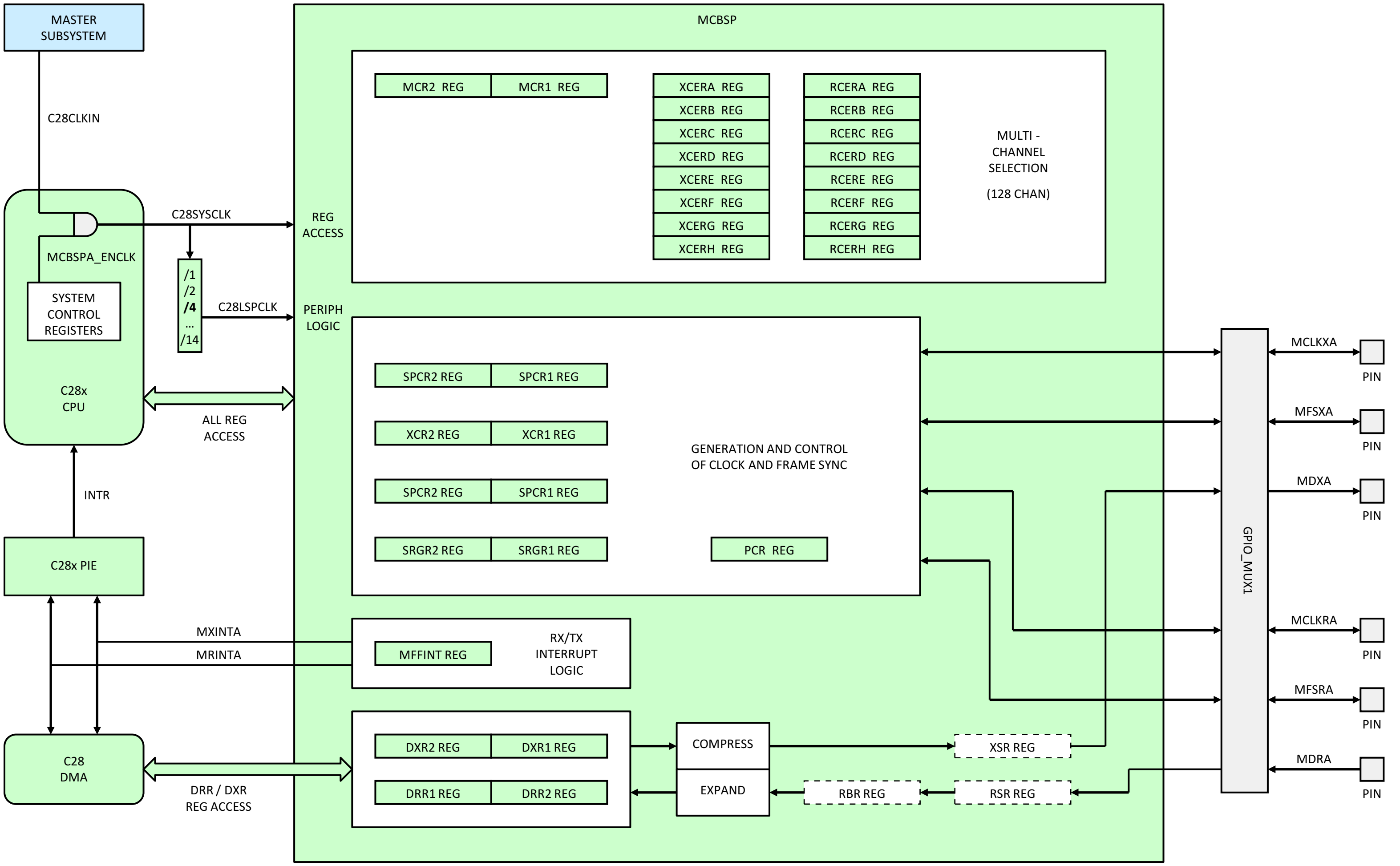


Figure 6-24. McBSP (C28x)

ADVANCE INFORMATION

7 Device and Documentation Support

7.1 Device Support

7.1.1 Development Support

TI offers an extensive line of development tools, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules. The tool's support documentation is electronically available within the Code Composer Studio™ Integrated Development Environment (IDE).

The following products support development of processor applications:

Software Development Tools: Code Composer Studio™ Integrated Development Environment (IDE): including Editor C/C++/Assembly Code Generation, and Debug plus additional development tools Scalable, Real-Time Foundation Software (SYS/BIOS), which provides the basic run-time target software needed to support any processor application.

Hardware Development Tools: Extended Development System (XDS™) Emulator

For a complete listing of development-support tools for the processor platform, visit the Texas Instruments website at www.ti.com. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

7.1.2 Device Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all Concerto™ MCU devices and support tools. Each Concerto™ MCU commercial family member has one of three prefixes: x, p, or no prefix (for example, **x**F28M36P63C2ZWTT). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (with prefix **x** for devices and TMDX for tools) through fully qualified production devices/tools (with no prefix for devices and TMDS, instead of TMDX, for tools).

xF28M36...	Experimental device that is not necessarily representative of the final device's electrical specifications
pF28M36...	Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification
F28M36...	Fully qualified production device

Support tool development evolutionary flow:

TMDX	Development-support product that has not yet completed Texas Instruments internal qualification testing
TMDS	Fully qualified development-support product

Devices with prefix **x** or **p** and TMDX development-support tools are shipped against the following disclaimer:
"Developmental product is intended for internal evaluation purposes."

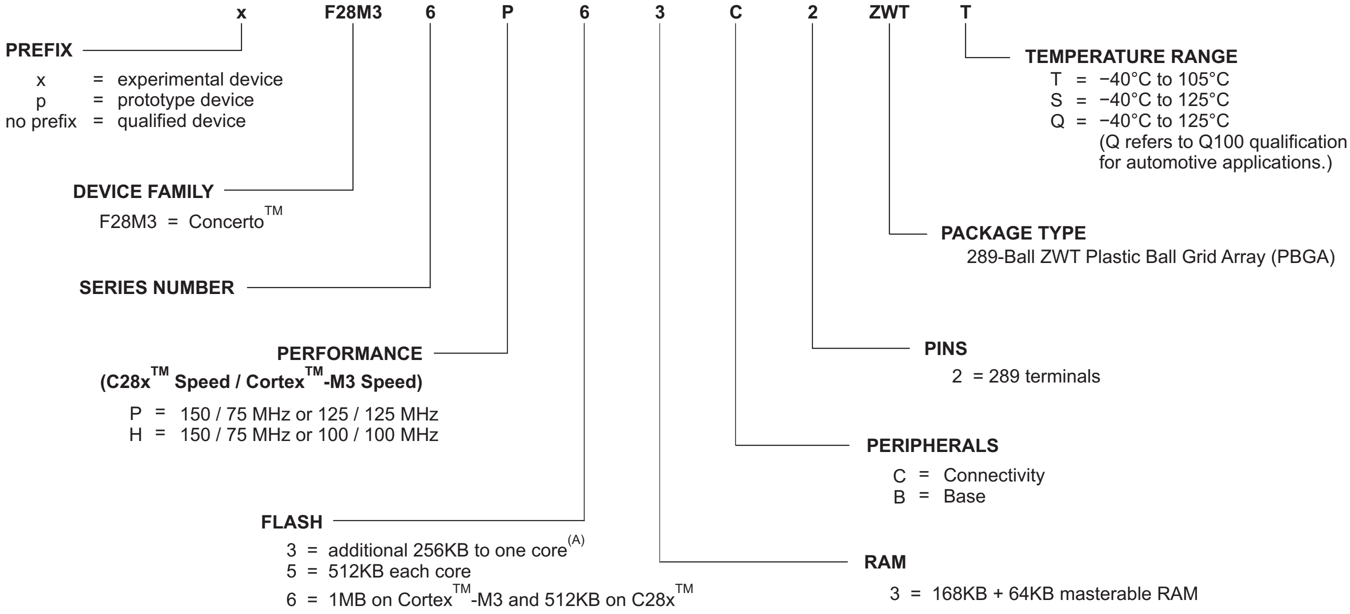
Production devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices with prefix of **x** or **p** have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, ZWT) and temperature range (for example, T).

For device part numbers and further ordering information of F28M36x devices in the ZWT package type, see the TI website (www.ti.com) or contact your TI sales representative.

For additional description of the device nomenclature markings on the die, see the *F28M36P63C, F28M36P53C, F28M36H53C, F28M36H53B, F28M36H33C, F28M36H33B Concerto MCU Silicon Errata* (literature number SPRZ375).



A. The additional 256KB is added to the Cortex™-M3 core (Connectivity Devices) or to the C28x™ core (Base Devices).

Figure 7-1. Device Nomenclature

7.2 Documentation Support

The following documents describe the MCU. Copies of these documents are available on the Internet at www.ti.com. Tip: Enter the literature number in the search box.

SPRUHE8	***Concerto F28M36x Technical Reference Manual*** details the integration, the environment, the functional description, and the programming models for each peripheral and subsystem in the F28M36x Microcontroller Processors.
SPRZ375	***F28M36P63C, F28M36P53C, F28M36H53C, F28M36H53B, F28M36H33C, F28M36H33B Concerto MCU Silicon Errata*** describes known advisories on silicon and provides workarounds.

7.3 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

TI E2E Community ***TI's Engineer-to-Engineer (E2E) Community.*** Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

TI Embedded Processors Wiki ***Texas Instruments Embedded Processors Wiki.*** Established to help developers get started with Embedded Processors from Texas Instruments and to foster innovation and growth of general knowledge about the hardware and software surrounding these devices.

7.4 Trademarks

Concerto, TMS320C28x, C28x, C2000, Piccolo, Delfino, TMS320C2000, Texas Instruments, XDS, Code Composer Studio are trademarks of Texas Instruments.

Cortex is a trademark of ARM Limited.

ARM is a registered trademark of ARM Ltd or its subsidiaries.

Freescale is a trademark of Freescale Semiconductor, Inc.

Philips and I^2C Bus are registered trademarks of Koninklijke Philips Electronics N.V.

Bosch is a registered trademark of Robert Bosch GmbH Corporation.

All trademarks are the property of their respective owners.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

This data sheet revision history highlights the technical changes made to the SPRS825 device-specific data sheet to make it an SPRS825A revision.

Scope: Changed title of document from *F28M36P63B2, F28M36P63C2, F28M36P53B2, F28M36P53C2, F28M36P33B2, F28M36P33C2 Concerto Microcontrollers* to *F28M36P63C, F28M36P53C, F28M36H53C, F28M36H53B, F28M36H33C, F28M36H33B Concerto Microcontrollers*.

Information/data on the F28M36x devices is **ADVANCE INFORMATION**.

ADVANCE INFORMATION concerns new products in the sampling or preproduction phase of development. Characteristic data and other specifications are subject to change without notice.

See table below.

LOCATION	ADDITIONS, DELETIONS, AND MODIFICATIONS
Global	• Changed title of document from *F28M36P63B2, F28M36P63C2, F28M36P53B2, F28M36P53C2, F28M36P33B2, F28M36P33C2 Concerto Microcontrollers* to *F28M36P63C, F28M36P53C, F28M36H53C, F28M36H53B, F28M36H33C, F28M36H33B Concerto Microcontrollers*. • Changed title of SPRZ375 from *F28M36P63B2, F28M36P63C2, F28M36P53B2, F28M36P53C2, F28M36P33B2, F28M36P33C2 Concerto MCU Silicon Errata* to *F28M36P63C, F28M36P53C, F28M36H53C, F28M36H53B, F28M36H33C, F28M36H33B Concerto MCU Silicon Errata*.
Figure 1-1	Functional Block Diagram: • Removed "1.8V VMON" block • Removed "1.2V VMON" block
Table 2-1	Updated "Hardware Features" table and footnotes
Table 2-11	Master Subsystem Peripherals: • Added "PBIST Control Registers" at 400F B000 – 400F B1FF • 400F BB00 – 400F BBFF, "Master Subsystem Peripherals" column: – Changed from "M Hardware Logic BIST Registers" to "M HWBIST Registers"
Table 2-14	Interrupts from NVIC to Cortex™-M3: • Interrupt Number 91, "Description" column: – Changed from "Reserved" to "PBIST Done"
Figure 2-3	Updated "Analog Subsystem" figure
Section 2.8	Resets: • Updated section to include POR reset
Section 2.8.1	Cortex™-M3 Resets: • Removed VMON/BOR functionality
Figure 2-5	Resets: • Changed "VOLTAGE REGULATION AND MONITORING" block to "VOLTAGE REGULATION AND POWER-ON-RESET"
Section 2.9	Internal Voltage Regulation and Power-On-Reset Functionality: • Updated section to support POR functions and to remove VMON and BOR functions
Section 2.9.1	Analog Subsystem's Internal 1.8-V VREG: • Added 10% tolerance to capacitor value
Figure 2-6	Voltage Regulation and Monitoring: • Updated figure to support both POR functions and to remove support for BOR functions • Removed "1.2V BOW" block
Section 2.9.2	Digital Subsystem's Internal 1.2-V VREG: • Updated capacitance range for each pin
Section 2.10.4	Main PLL: • Updated "The Main PLL uses the reference clock from pins X1 ..." paragraph
Figure 2-8	Updated "Main PLL" figure
Section 2.10.5	USB PLL: • Updated "The USB PLL uses the reference clock selectable between the input clock ..." paragraph

LOCATION	ADDITIONS, DELETIONS, AND MODIFICATIONS
Figure 2-9	Updated "USB PLL" figure
Table 3-1	Terminal Functions: • $\overline{XRS}$: Updated DESCRIPTION to remove brown-out-reset (BOR) functionality • $\overline{ARS}$: Updated DESCRIPTION to remove brown-out-reset (BOR) functionality • V_{DD18}: Updated capacitor value to 1.2 µF • V_{DD12}: Updated capacitor value to a range of 250 nF–750 nF
Section 4.2	Recommended Operating Conditions: • Device supply voltage, I/O, V_{DDIO}: – Changed MIN value from 3.14 V to 2.97 V – Changed MAX value from 3.46 V to 3.63 V • Device supply voltage, Master and Control Subsystems, V_{DD12}: – Changed MAX value from 1.26 V to 1.32 V • Analog supply voltage, V_{DDA}: – Changed MIN value from 3.14 V to 2.97 V – Changed MAX value from 3.47 V to 3.63 V
Section 4.3	Electrical Characteristics: • Removed "V_{DDIO} BOR trip point" parameter • Removed "V_{DDIO} BOR hysteresis" parameter • Removed "Supervisor reset release delay time" parameter • Added "Digital Subsystem POR reset release delay time" parameter • Added "Analog Subsystem POR reset release delay time" parameter • VREG V_{DD18} output: – Added MIN value of 1.77 V – Removed TYP value of 1.8 V – Added MAX value of 1.935 V • Removed footnote about on-chip VREG output being monitored by POR/BOR circuits
Table 5-1	Current Consumption at 150-MHz C28x SYSCLKOUT and 75-MHz M3SSCLK: • Added TYP values
Figure 5-2	Power-On Reset: • Removed "The internal pullup or pulldown will take effect when BOR is driven high" footnote
Section 5.6	Added "Flash Timing – Master Subsystem" section
Section 5.7	Added "Flash Timing – Control Subsystem" section
Section 6.1	Analog and Shared Peripherals: • Added reference to the *Concerto F28M36x Technical Reference Manual* (literature number SPRUHE8)
Section 6.1.1.5	Added "ADC Electrical Data and Timing" section
Section 6.1.2.1	Added "On-Chip Comparator and DAC Electrical Data and Timing" section
Figure 6-7	EPI General-Purpose Modes: • Changed values of FRMPIN so that the values alternate between 1 and 0
Figure 6-8	EPI SDRAM Mode: • Changed SDRAM SIZE from "16 MBit" to "64 MBit"
Section 6.2	Master Subsystem Peripherals: • Added reference to the *Concerto F28M36x Technical Reference Manual* (literature number SPRUHE8)
Section 6.3	Control Subsystem Peripherals: • Added reference to the *Concerto F28M36x Technical Reference Manual* (literature number SPRUHE8)
Section 7.2	Added "Documentation Support" section

8 Mechanical Packaging and Orderable Information

8.1 Thermal Data for Package

Table 8-1 and Table 8-2 show the thermal data. See Section 5.2 for more information on thermal design considerations.

Table 8-1. Thermal Model 289-Ball ZWT Results (Revision 0 Silicon)

	AIR FLOW			
PARAMETER	**0 lfm**	**150 lfm**	**250 lfm**	**500 lfm**
θ_{JA} [°C/W] High k PCB	23.0	20.5	19.5	18.5
Ψ_{JT} [°C/W]	0.5	0.6	0.8	1.0
Ψ_{JB}	12.9	12.9	12.8	12.7
θ_{JC}	10.5			
θ_{JB}	12.8			

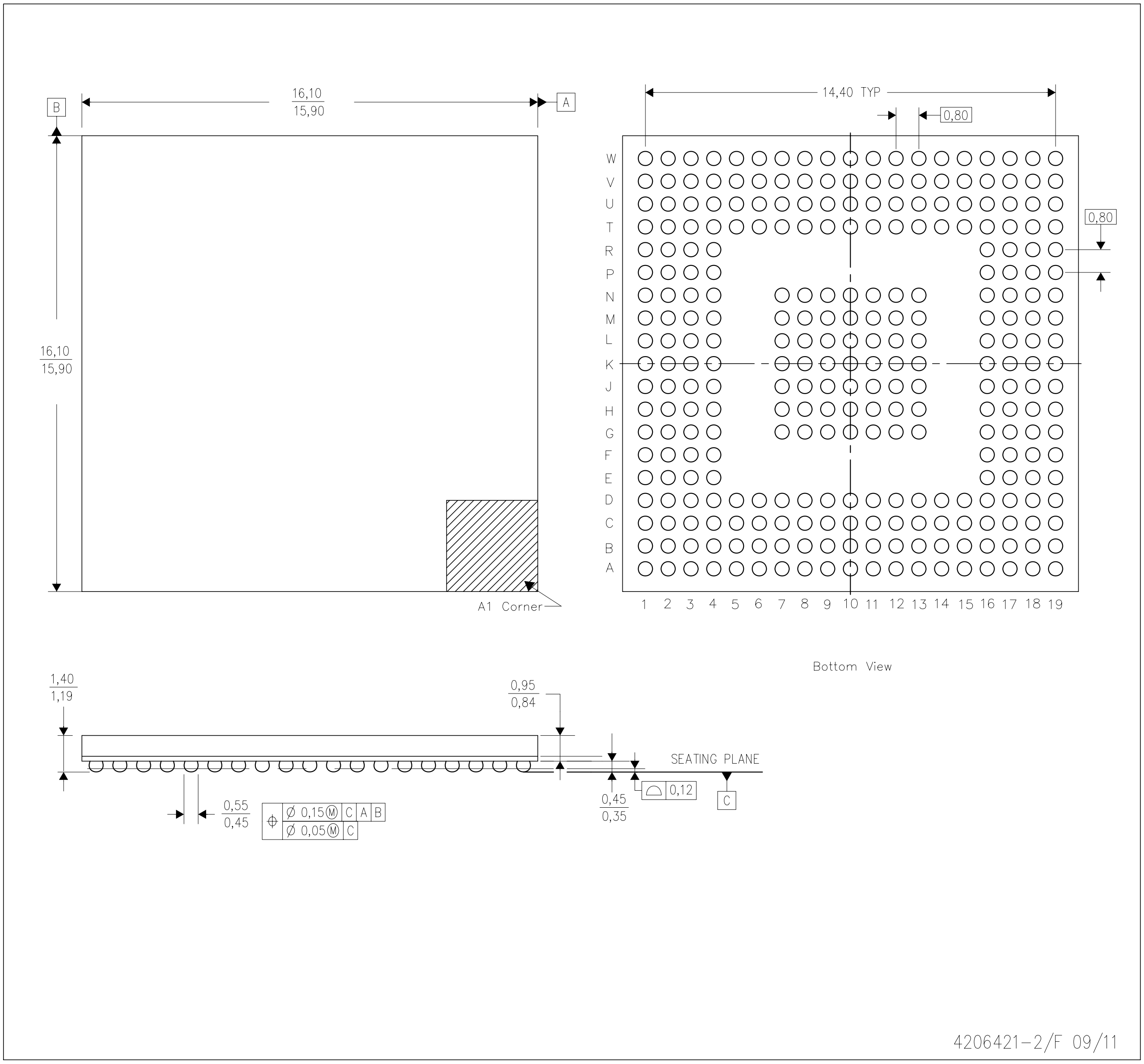
Table 8-2. Thermal Model 289-Ball ZWT Results (Revision A Silicon)

	AIR FLOW			
PARAMETER	**0 lfm**	**150 lfm**	**250 lfm**	**500 lfm**
θ_{JA} [°C/W] High k PCB	20.6	17.9	16.8	15.6
Ψ_{JT} [°C/W]	0.25	0.35	0.42	0.53
Ψ_{JB}	10.4	10.5	10.4	10.3
θ_{JC}	7.5			
θ_{JB}	10.5			

8.2 Packaging Information

The following packaging information and addendum reflect the most current data available for the designated devices. This data is subject to change without notice and without revision of this document.

ZWT (S–PBGA–N289) PLASTIC BALL GRID ARRAY



NOTES:
A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
B. This drawing is subject to change without notice.
C. This is a Pb–free solder ball design.
D. Falls within JEDEC MO–275.

PACKAGE OPTION ADDENDUM

11-Apr-2013

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)	Op Temp (°C)	Top-Side Markings (4)	Samples
XF28M36P63C2ZWTT	ACTIVE	NFBGA	ZWT	289	1	TBD	Call TI	Call TI	-40 to 85		Samples

[1] The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

[2] Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

[3] MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

[4] Multiple Top-Side Markings will be inside parentheses. Only one Top-Side Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Top-Side Marking for that device.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2013, Texas Instruments Incorporated