TMS570LC4357 Hercules™ Microcontroller Based on the ARM® Cortex®-R Core

1 Device Overview

1.1 Features

- High-Performance Automotive-Grade Microcontroller for Safety-Critical Applications
  - Dual-Core Lockstep CPUs With ECC-Protected Caches
  - ECC on Flash and RAM Interfaces
  - Built-In Self-Test (BIST) for CPU, High-End Timers, and On-Chip RAMs
  - Error Signaling Module (ESM) With Error Pin
  - Voltage and Clock Monitoring
- ARM® Cortex® - R5F 32-Bit RISC CPU
  - 1.66 DMIPS/MHz With 8-Stage Pipeline
  - FPU With Single- and Double-Precision
  - 16-Region Memory Protection Unit (MPU)
  - 32KB of Instruction and 32KB of Data Caches With ECC
  - Open Architecture With Third-Party Support
- Operating Conditions
  - Up to 300-MHz CPU Clock
  - Core Supply Voltage (VCC): 1.14 to 1.32 V
  - I/O Supply Voltage (VCCIO): 3.0 to 3.6 V
- Integrated Memory
  - 4MB of Program Flash With ECC
  - 512KB of RAM With ECC
  - 128KB of Data Flash for Emulated EEPROM With ECC
- 16-Bit External Memory Interface (EMIF)
- Hercules™ Common Platform Architecture
  - Consistent Memory Map Across Family
  - Real-Time Interrupt (RTI) Timer (OS Timer)
  - Two 128-Channel Vectored Interrupt Modules (VIMs) With ECC Protection on Vector Table
    - VIM1 and VIM2 in Safety Lockstep Mode
  - Two 2-Channel Cyclic Redundancy Checker (CRC) Modules
- Direct Memory Access (DMA) Controller
  - 32 Channels and 48 Peripheral Requests
  - ECC Protection for Control Packet RAM
  - DMA Accesses Protected by Dedicated MPU
- Frequency-Modulated Phase-Locked Loop (FMPLL) With Built-In Slip Detector
- Separate Nonmodulating PLL
- IEEE 1149.1 JTAG, Boundary Scan, and ARM CoreSight™ Components
- Advanced JTAG Security Module (AJSM)
- Trace and Calibration Capabilities
  - ETM™, RTP, DMM, POM
- Multiple Communication Interfaces
  - 10/100 Mbps Ethernet MAC (EMAC)
    - IEEE 802.3 Compliant (3.3-V I/O Only)
    - Supports MII, RMII, and MDIO
  - FlexRay Controller With 2 Channels
    - 8KB of Message RAM With ECC Protection
    - Dedicated FlexRay Transfer Unit (FTU)
  - Four CAN Controller (DCAN) Modules
    - 64 Mailboxes, Each With ECC Protection
    - Compliant to CAN Protocol Version 2.0B
  - Two Inter-Integrated Circuit (I^2C) Modules
  - Five Multibuffered Serial Peripheral Interface (MibSPI) Modules
    - MibSPI1: 256 Words With ECC Protection
    - Other MibSPIs: 128 Words With ECC Protection
  - Four UART (SCI) Interfaces, Two With Local Interconnect Network (LIN 2.1) Interface Support
- Two Next Generation High-End Timer (N2HET) Modules
  - 32 Programmable Channels Each
  - 256-Word Instruction RAM With Parity
  - Hardware Angle Generator for Each N2HET
  - Dedicated High-End Timer Transfer Unit (HTU) for Each N2HET
- Two 12-Bit Multibuffered Analog-to-Digital Converter (MibADC) Modules
  - MibADC1: 32 Channels Plus Control for up to 1024 Off-Chip Channels
  - MibADC2: 25 Channels
  - 16 Shared Channels
  - 64 Result Buffers Each With Parity Protection
- Enhanced Timing Peripherals
  - 7 Enhanced Pulse Width Modulator (ePWM) Modules
  - 6 Enhanced Capture (eCAP) Modules
  - 2 Enhanced Quadrature Encoder Pulse (eQEP) Modules
- Three On-Die Temperature Sensors
- Up to 145 Pins Available for General-Purpose I/O (GPIO)
- 16 Dedicated GPIO Pins With External Interrupt Capability
- Packages
  - 337-Ball Grid Array (ZWT) [Green]

1.2 Applications

- Braking Systems (Antilock Brake Systems and Electronic Stability Control)
- Electric Power Steering (EPS)
- HEV and EV Inverter Systems
- Battery-Management Systems
- Active Driver Assistance Systems
- Aerospace and Avionics
- Railway Communications
- Off-road Vehicles

1.3 Description

The TMS570LC4357 device is part of the Hercules TMS570 series of high-performance automotive-grade ARM® Cortex®-R-based MCUs. Comprehensive documentation, tools, and software are available to assist in the development of ISO 26262 and IEC 61508 functional safety applications. Start evaluating today with the Hercules TMS570LC43x LaunchPad Development Kit. The TMS570LC4357 device has on-chip diagnostic features including: dual CPUs in lockstep, Built-In Self-Test (BIST) logic for CPU, the N2HET coprocessors, and for on-chip SRAMs; ECC protection on the L1 caches, L2 flash, and SRAM memories. The device also supports ECC or parity protection on peripheral memories and loopback capability on peripheral I/Os.

The TMS570LC4357 device integrates two ARM Cortex-R5F floating-point CPUs, operating in lockstep, which offer an efficient 1.66 DMIPS/MHz, and can run up to 300 MHz providing up to 498 DMIPS. The device supports the big-endian [BE32] format.

The TMS570LC4357 device has 4MB of integrated flash and 512KB of data RAM with single-bit error correction and double-bit error detection. The flash memory on this device is a nonvolatile, electrically erasable and programmable memory, implemented with a 64-bit-wide data bus interface. The flash operates on a 3.3-V supply input (the same level as the I/O supply) for all read, program, and erase operations. The SRAM supports read and write accesses in byte, halfword, and word modes.

The TMS570LC4357 device features peripherals for real-time control-based applications, including two Next Generation High-End Timer (N2HET) timing coprocessors with up to 64 total I/O terminals.

The N2HET is an advanced intelligent timer that provides sophisticated timing functions for real-time applications. The timer is software-controlled, with a specialized timer micromachine and an attached I/O port. The N2HET can be used for pulse-width-modulated outputs, capture or compare inputs, or GPIO. The N2HET is especially well suited for applications requiring multiple sensor information or drive actuators with complex and accurate time pulses. The High-End Timer Transfer Unit (HTU) can perform DMA-type transactions to transfer N2HET data to or from main memory. A Memory Protection Unit (MPU) is built into the HTU.

The Enhanced Pulse Width Modulator (ePWM) module can generate complex pulse width waveforms with minimal CPU overhead or intervention. The ePWM is easy to use and supports both high-side and low-side PWM and deadband generation. With integrated trip zone protection and synchronization with the on-chip MibADC, the ePWM is ideal for digital motor control applications.

The Enhanced Capture (eCAP) module is essential in systems where the accurately timed capture of external events is important. The eCAP can also be used to monitor the ePWM outputs or for simple PWM generation when not needed for capture applications.

The Enhanced Quadrature Encoder Pulse (eQEP) module directly interfaces with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine as used in high-performance motion and position-control systems.

The device has two 12-bit-resolution MibADCs with 41 total channels and 64 words of parity-protected buffer RAM. The MibADC channels can be converted individually or by group for special conversion sequences. Sixteen channels are shared between the two MibADCs. Each MibADC supports three separate groupings. Each sequence can be converted once when triggered or configured for continuous conversion mode. The MibADC has a 10-bit mode for use when compatibility with older devices or faster conversion time is desired. One of the channels in MibADC1 and two of the channels in MibADC2 can be used to convert temperature measurements from the three on-chip temperature sensors.

The device has multiple communication interfaces: Five MibSPIs; four UART (SCI) interfaces, two with LIN support; four CANs; two I2C modules;one Ethernet Controller; and one FlexRay controller. The SPI provides a convenient method of serial interaction for high-speed communications between similar shift-register type devices. The LIN supports the Local Interconnect standard (LIN 2.1) and can be used as a UART in full-duplex mode using the standard Non-Return-to-Zero (NRZ) format. The DCAN supports the CAN 2.0B protocol standard and uses a serial, multimaster communication protocol that efficiently supports distributed real-time control with robust communication rates of up to 1 Mbps. The DCAN is ideal for applications operating in noisy and harsh environments (for example, automotive and industrial fields) that require reliable serial communication or multiplexed wiring. The FlexRay controller uses a dual-channel serial, fixed time base multimaster communication protocol with communication rates of 10 Mbps per channel. A FlexRay Transfer Unit (FTU) enables autonomous transfers of FlexRay data to and from main CPU memory. HTU transfers are protected by a dedicated, built-in MPU. The Ethernet module supports MII, RMII, and Management Data I/O (MDIO) interfaces. The I2C module is a multimaster communication module providing an interface between the microcontroller and an I^2C-compatible device through the I^2C serial bus. The I2C module supports speeds of 100 and 400 kbps.

The Frequency-Modulated Phase-Locked Loop (FMPLL) clock module multiplies the external frequency reference to a higher frequency for internal use. The Global Clock Module (GCM) manages the mapping between the available clock sources and the internal device clock domains.

The device also has two External Clock Prescaler (ECP) modules. When enabled, the ECPs output a continuous external clock on the ECLK1 and ECLK2 balls. The ECLK frequency is a user-programmable ratio of the peripheral interface clock (VCLK) frequency. This low-frequency output can be monitored externally as an indicator of the device operating frequency.

The Direct Memory Access (DMA) controller has 32 channels, 48 peripheral requests, and ECC protection on its memory. An MPU is built into the DMA to protect memory against erroneous transfers.

The Error Signaling Module (ESM) monitors on-chip device errors and determines whether an interrupt or external Error pin/ball (nERROR) is triggered when a fault is detected. The nERROR signal can be monitored externally as an indicator of a fault condition in the microcontroller.

The External Memory Interface (EMIF) provides a memory extension to asynchronous and synchronous memories or other slave devices.

A Parameter Overlay Module (POM) is included to enhance the debugging capabilities of application code. The POM can reroute flash accesses to internal RAM or to the EMIF, thus avoiding the reprogramming steps necessary for parameter updates in flash. This capability is particularly helpful during real-time system calibration cycles.

Several interfaces are implemented to enhance the debugging capabilities of application code. In addition to the built-in ARM Cortex-R5F CoreSight debug features, the Embedded Cross Trigger (ECT) supports the interaction and synchronization of multiple triggering events within the SoC. An External Trace Macrocell (ETM) provides instruction and data trace of program execution. For instrumentation purposes, a RAM Trace Port (RTP) module is implemented to support high-speed tracing of RAM and peripheral accesses by the CPU or any other master. A Data Modification Module (DMM) gives the ability to write external data into the device memory. Both the RTP and DMM have no or minimal impact on the program execution time of the application code.

With integrated safety features and a wide choice of communication and control peripherals, the TMS570LC4357 device is an ideal solution for high-performance real-time control applications with safety-critical requirements.

Device Information (1)

PART NUMBER	PACKAGE	BODY SIZE
TMS570LC4357ZWT	NFBGA (337)	16.00 mm × 16.00 mm

(1) For more information on these devices, see Section 10, *Mechanical Packaging and Orderable Information*.

1.4 Functional Block Diagram

Figure 1-1 shows the functional block diagram of the device.

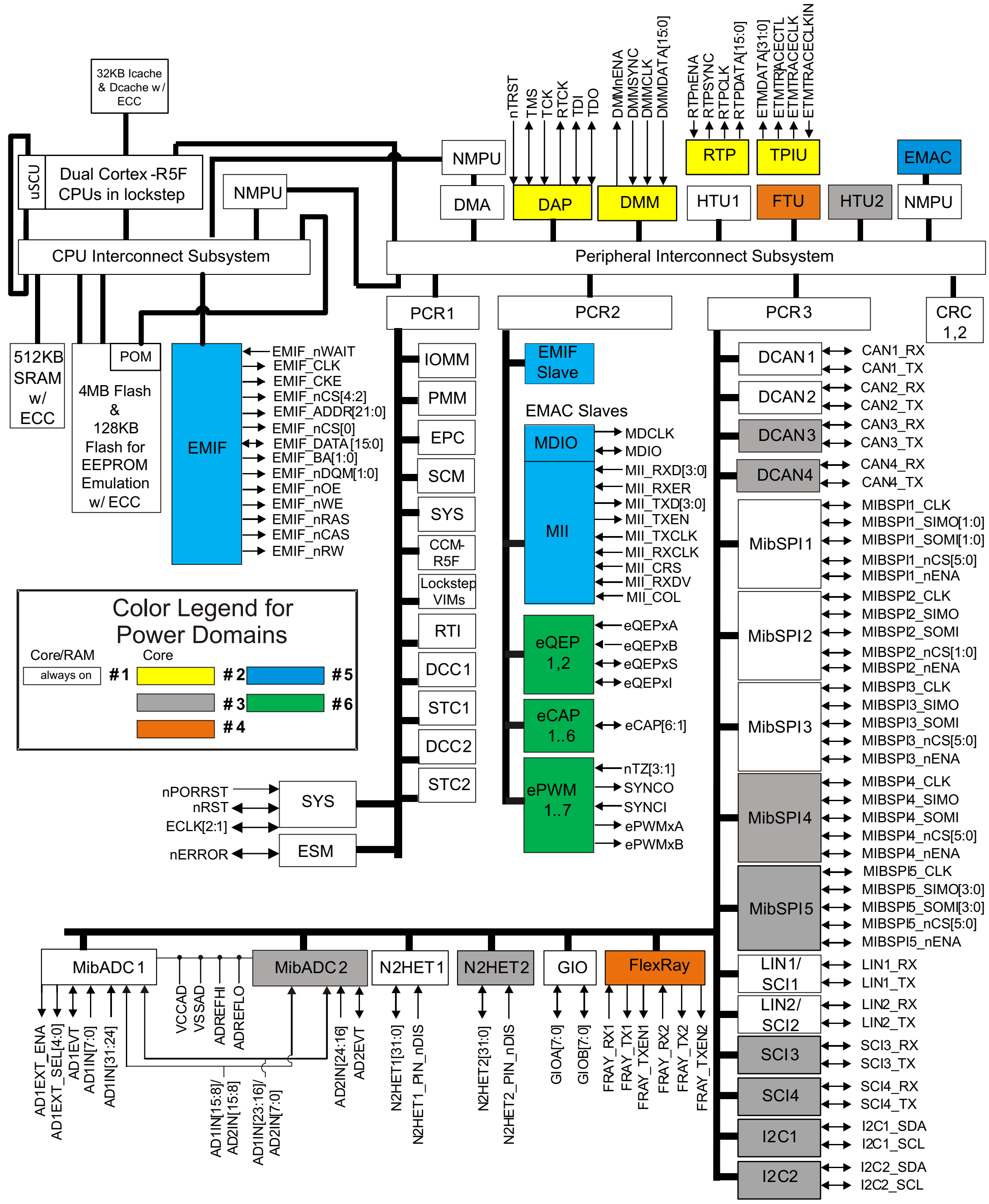


Figure 1-1. Functional Block Diagram

Table of Contents

2 Revision History

This data manual revision history highlights the technical changes made to the SPNS195B device-specific data manual to make it an SPNS195C revision. These devices are now in the Production Data (PD) stage of development.

Changes from January 31, 2016 to June 25, 2016 (from B Revision (January 2016) to C Revision) **Page**

- Global: Updated/Changed the product status from Product Preview to Production Data. 1
- Table 6-13 (LPO Specifications): Updated/Changed LPO - HF oscillator, Untrimmed frequency TYP value from "9.6" to "9" MHz 78
- Section 6.9.3 (Special Consideration for CPU Access Errors Resulting in Imprecise Aborts): Add missing subsection 96
- Section 6.14.1 (External Memory Interface (EMIF), Features): Updated/Changed the EMIF asynchronous memory maximum addressable size from "32KB" to "16MB" each 109
- Section 6.14 (External Memory Interface (EMIF)): Added 32-bit access note using a 16-bit EMIF interface. 109
- Added "Commonly caused by ..." statement for clarification 131
- Table 6-56 (ETMTRACECLK Timing): Restructured timing table formatting to standards.............................. 149
- Table 7-7, (eCAPx Clock Enable Control): Updated/Changed "ePWM" to "eCAP" in the MODULE INSTANCE column.. 162
- Table 7-11, (eQEPx Clock Enable Control): Updated/Changed "ePWM" to "eQEP" in the MODULE INSTANCE column.. 164
- Table 7-16 (MibADC1 Event Trigger Selection): Added lead-in paragraph referencing the able 166
- (MibADC1 Event Trigger Hookup): NOTE: Added new paragraph 168
- Table 7-17 (MibADC2 Event Trigger Selection): Added lead-in paragraph referencing the able 168
- Section 7.4.2.3 (Controlling ADC1 and ADC2 Event Trigger Options Using SOC Output from ePWM Modules): Updated/Changed the names of the four ePWM signals that event trigger the ADC 170
- Table 7-22 (MibADC Operating Characteristics Over 3.0 V to 3.6 V Operating Conditions): Updated/Changed the 10- and 12-bit mode formulas to be superscript power of 2 values 175
- Table 7-23 (MibADC Operating Characteristics Over 3.6 V to 5.25 V Operating Conditions): Updated/Changed the 10- and 12-bit mode formulas to be superscript power of 2 values 175

3 Device Comparison

Table 3-1 lists the features of the TMS570LC4357 devices.

Table 3-1. TMS570LC4357 Device Comparison(1)(2)

FEATURES	DEVICES			
Generic Part Number	TMS570LC4357ZWT(3)	TMS570LS3137ZWT(3)	TMS570LS3135ZWT	TMS570LS1227ZWT(3)
Package	337 BGA	337 BGA	337 BGA	337 BGA
CPU	ARM Cortex-R5F	ARM Cortex-R4F	ARM Cortex-R4F	ARM Cortex-R4F
Frequency (MHz)	300	180	180	180
Cache (KB)	32 I 32 D	–	–	–
Flash (KB)	4096	3072	3072	1280
RAM (KB)	512	256	256	192
Data Flash [EEPROM] (KB)	128	64	64	64
EMAC	10/100	10/100	–	10/100
FlexRay	2-ch	2-ch	2-ch	2-ch
CAN	4	3	3	3
MibADC 12-bit (Ch)	2 (41ch)	2 (24ch)	2 (24ch)	2 (24ch)
N2HET (Ch)	2 (64)	2 (44)	2 (44)	2 (44)
ePWM Channels	14	–	–	14
eCAP Channels	6	–	–	6
eQEP Channels	2	–	–	2
MibSPI (CS)	5 (4 x 6 + 2)	3 (6 + 6 + 4)	3 (6 + 6 + 4)	3 (6 + 6 + 4)
SPI (CS)	–	2 (2 + 1)	2 (2 + 1)	2 (2 + 1)
SCI (LIN)	4 (2 with LIN)	2 (1 with LIN)	2 (1 with LIN)	2 (1 with LIN)
I^2C	2	1	1	1
GPIO (INT)(4)	168 (with 16 interrupt capable)	144 (with 16 interrupt capable)	144 (with 16 interrupt capable)	101 (with 16 interrupt capable)
EMIF	16-bit data	16-bit data	16-bit data	16-bit data
ETM (Trace)	32-bit	32-bit	32-bit	–
RTP/DMM	16/16	16/16	16/16	–
Operating Temperature	–40ºC to 125ºC	–40ºC to 125ºC	–40ºC to 125ºC	–40ºC to 125ºC
Core Supply (V)	1.14 V – 1.32 V	1.14 V – 1.32 V	1.14 V – 1.32 V	1.14 V – 1.32 V
I/O Supply (V)	3.0 V – 3.6 V	3.0 V – 3.6 V	3.0 V – 3.6 V	3.0 V – 3.6 V

(1) For additional device variants, see www.ti.com/tms570.
(2) This table reflects the maximum configuration for each peripheral. Some functions are multiplexed and not all pins are available at the same time.
(3) Superset device
(4) Total number of pins that can be used as general-purpose input or output when not used as part of a peripheral.

4 Terminal Configuration and Functions

4.1 ZWT BGA Package Ball-Map (337 Terminal Grid Array)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	
19	VSS	VSS	TMS	N2HET1 [10]	MIBSPI5 NCS[0]	MIBSPI1 SIMO[0]	MIBSPI1 NENA	MIBSPI5 CLK	MIBSPI5 SIMO[0]	N2HET1 [28]	DMM_ DATA[0]	DCAN3RX	AD1EVT	AD1IN[15] / AD2IN[15]	AD1IN[22] / AD2IN[06]	AD1IN [06]	AD1IN[11] / AD2IN[11]	AD2IN[24]	VSSAD	19
18	VSS	TCK	TDO	nTRST	N2HET1 [08]	MIBSPI1 CLK	MIBSPI1 SOMI[0]	MIBSPI5 NENA	MIBSPI5 SOMI[0]	N2HET1 [0]	DMM_ DATA[1]	DCAN3TX	AD1IN[24]	AD1IN[08] / AD2IN[08]	AD1IN[14] / AD2IN[14]	AD1IN[13] / AD2IN[13]	AD1IN [04]	AD1IN [02]	AD2IN[24]	18
17	TDI	nRST	EMIF_ ADDR[21]	EMIF_ nWE	MIBSPI5 SOMI[1]	DMM_ CLK	MIBSPI5 SIMO[3]	MIBSPI5 SIMO[2]	N2HET1 [31]	EMIF_ nCS[3]	EMIF_ nCS[2]	EMIF_ nCS[4]	EMIF_ nCS[0]	AD1IN[25]	AD1IN [05]	AD1IN [03]	AD1IN[10] / AD2IN[10]	AD1IN [01]	AD1IN[09] / AD2IN[09]	17
16	RTCK	FRAY TXEN1	EMIF_ ADDR[20]	EMIF_ BA[1]	MIBSPI5 SIMO[1]	DMM_ nENA	MIBSPI5 SOMI[3]	MIBSPI5 SOMI[2]	DMM_ SYNC	N2HET2 [08]	N2HET2 [09]	N2HET2 [10]	N2HET2 [11]	AD1IN[26]	AD1IN[23] / AD2IN[07]	AD1IN[12] / AD2IN[12]	AD1IN[19] / AD2IN[03]	ADREFLO	VSSAD	16
15	FRAYRX1	FRAYTX1	EMIF_ ADDR[19]	EMIF_ ADDR[18]	ETM DATA[06]	ETM DATA[05]	ETM DATA[04]	ETM DATA[03]	ETM DATA[02]	ETM DATA[16] / EMIF_ DATA[0]	ETM DATA[17] / EMIF_ DATA[1]	ETM DATA[18] / EMIF_ DATA[2]	ETM DATA[19] / EMIF_ DATA[3]	AD1IN[27]	AD1IN[28]	AD1IN[21] / AD2IN[05]	AD1IN[20] / AD2IN[04]	ADREFHI	VCCAD	15
14	N2HET1 [26]	nERROR	EMIF_ ADDR[17]	EMIF_ ADDR[16]	ETM DATA[07]	VCCIO	VCCIO	VCCIO	VCC	VCC	VCCIO	VCCIO	VCCIO	VCCIO	AD1IN[29]	AD1IN[30]	AD1IN[18] / AD2IN[02]	AD1IN [07]	AD1IN [0]	14
13	N2HET1 [17]	N2HET1 [19]	EMIF_ ADDR[15]	N2HET2 [04]	ETM DATA[12] / EMIF_BA[0]	VCCIO								VCCIO	ETM DATA[01]	AD1IN[31]	AD1IN[17] / AD2IN[01]	AD1IN[16] / AD2IN[0]	AD2IN[16]	13
12	ECLK	N2HET1 [04]	EMIF_ ADDR[14]	N2HET2 [05]	ETM DATA[13] / EMIF_nOE	VCCIO		VSS	VSS	VCC	VSS	VSS		VCCIO	ETM DATA[0]	MIBSPI5 NCS[3]	AD2IN[19]	AD2IN[18]	AD2IN[17]	12
11	N2HET1 [14]	N2HET1 [30]	EMIF_ ADDR[13]	N2HET2 [06]	ETM DATA[14] / EMIF_ nDQM[1]	VCCIO		VSS	VSS	VSS	VSS	VSS		VCCPLL	ETM TRACE CTL	AD2IN[20]	AD2IN[21]	AD2IN[22]	AD2IN[23]	11
10	DCAN1TX	DCAN1RX	EMIF_ ADDR[12]	ePWM1B	ETM DATA[15] / EMIF_ nDQM[0]	VCC		VCC	VSS	VSS	VSS	VCC		VCC	ETM TRACE CLKOUT	AD2EVT	MIBSPI1 NCS[4]	MIBSPI3 NCS[0]	GIOB[3]	10
9	N2HET1 [27]	FRAY TXEN2	EMIF_ ADDR[11]	ePWM1A	ETM DATA[08] / EMIF_ ADDR[5]	VCC		VSS	VSS	VSS	VSS	VSS		VCCIO	ETM TRACE CLKIN	MDCLK	MIBSPI1 NCS[5]	MIBSPI3 CLK	MIBSPI3 NENA	9
8	FRAYRX2	FRAYTX2	EMIF_ ADDR[10]	N2HET2[1]	ETM DATA[09] / EMIF_ ADDR[4]	VCCP		VSS	VSS	VCC	VSS	VSS		VCCIO	ETM DATA[31] / EMIF_ DATA[15]	N2HET2 [23]	MII_TXD [0]	MIBSPI3 SOMI	MIBSPI3 SIMO	8
7	LIN1RX	LIN1TX	EMIF_ ADDR[9]	N2HET2 [2]	ETM DATA[10] / EMIF_ ADDR[3]	VCCIO								VCCIO	ETM DATA[30] / EMIF_ DATA[14]	N2HET2 [22]	MII_TX_ CLK	N2HET1 [09]	nPORRST	7
6	GIOA[4]	MIBSPI5 NCS[1]	EMIF_ ADDR[8]	N2HET2 [0]	ETM DATA[11] / EMIF_ ADDR[2]	VCCIO	VCCIO	VCCIO	VCCIO	VCC	VCC	VCCIO	VCCIO	VCCIO	ETM DATA[29] / EMIF_ DATA[13]	N2HET2 [21]	MII_RX_ DV	N2HET1 [05]	MIBSPI5 NCS[2]	6
5	GIOA[0]	GIOA[5]	EMIF_ ADDR[7]	EMIF_ ADDR[1]	ETM DATA[20] / EMIF_ DATA[4]	ETM DATA[21] / EMIF_ DATA[5]	ETM DATA[22] / EMIF_ DATA[6]	FLTP2	FLTP1	ETM DATA[23] / EMIF_ DATA[7]	ETM DATA[24] / EMIF_ DATA[8]	ETM DATA[25] / EMIF_ DATA[9]	ETM DATA[26] / EMIF_ DATA[10]	ETM DATA[27] / EMIF_ DATA[11]	ETM DATA[28] / EMIF_ DATA[12]	N2HET2 [20]	MII_RX_ ER	MIBSPI3 NCS[1]	N2HET1 [02]	5
4	N2HET1 [16]	N2HET1 [12]	EMIF_ ADDR[6]	EMIF_ ADDR[0]	MII_TXEN	MDIO	MII_TXD [3]	N2HET1 [21]	N2HET1 [23]	N2HET2 [15]	N2HET2 [16]	N2HET2 [17]	N2HET2 [18]	N2HET2 [19]	EMIF_ nCAS	MII_ RXCLK	MII_RXD [0]	MII_CRS	MII_COL	4
3	N2HET1 [29]	N2HET1 [22]	MIBSPI3 NCS[3]	N2HET2 [12]	N2HET1 [11]	MIBSPI1 NCS[1]	MIBSPI1 NCS[2]	GIOA[6]	MIBSPI1 NCS[3]	EMIF_ CLK	EMIF_ CKE	N2HET1 [25]	N2HET2 [7]	EMIF_ nWAIT	EMIF_ nRAS	MII_RXD [1]	MII_RXD [2]	MII_RXD [3]	N2HET1 [06]	3
2	VSS	MIBSPI3 NCS[2]	GIOA[1]	N2HET2 [13]	N2HET2 [3]	GIOB[2]	GIOB[5]	DCAN2TX	GIOB[6]	GIOB[1]	KELVIN_ GND	GIOB[0]	N2HET1 [13]	N2HET1 [20]	MIBSPI1 NCS[0]	MII_TXD [2]	TEST	N2HET1 [1]	VSS	2
1	VSS	VSS	GIOA[2]	N2HET2 [14]	GIOA[3]	GIOB[7]	GIOB[4]	DCAN2RX	N2HET1 [18]	OSCIN	OSCOUT	GIOA[7]	N2HET1 [15]	N2HET1 [24]	MII_TXD [1]	N2HET1 [7]	NHET1 [03]	VSS	VSS	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	

Figure 4-1. ZWT Package Pinout. Top View

Note: Balls can have multiplexed functions. See Section 4.2.2 for detailed information.

4.2 Terminal Functions

Table 4-1 through Table 4-27 identify the external signal names, the associated terminal numbers along with the mechanical package designator, the terminal type (Input, Output, I/O, Power, or Ground), whether the terminal has any internal pullup/pulldown, whether the terminal can be configured as a GIO, and a functional terminal description. The first signal name listed is the primary function for that terminal. The signal name in **Bold** is the function being described. For information on how to select between different multiplexed functions, see the Section 4.2.2, *Multiplexing* of this data manual along with the I/O Multiplexing Module (IOMM) chapter in the Technical Reference Manual (TRM) (SPNU563).

NOTE

In the Terminal Functions tables below, the "Default Pull State" is the state of the pull applied to the terminal while nPORRST is low and immediately after nPORRST goes High. The default pull direction may change when software configures the pin for an alternate function. The "Pull Type" is the type of pull asserted when the signal name in bold is enabled for the given terminal by the IOMM control registers.

All I/O signals except nRST are configured as inputs while nPORRST is low and immediately after nPORRST goes High. While nPORRST is low, the input buffers are disabled, and the output buffers are disabled with the default pulls enabled.

All output-only signals have the output buffer disabled and the default pull enabled while nPORRST is low, and are configured as outputs with the pulls disabled immediately after nPORRST goes High.

4.2.1 ZWT Package

4.2.1.1 Multibuffered Analog-to-Digital Converters (MibADC)

Table 4-1. ZWT Multibuffered Analog-to-Digital Converters (MibADC1, MibADC2)

TERMINAL		SIGNAL TYPE	DEFAULT PULL STATE	PULL TYPE	OUTPUT BUFFER DRIVE STRENGTH	DESCRIPTION
SIGNAL NAME	337 ZWT					
AD1EVT/MII_RX_ER/RMII_RX_ER/nTZ1_1	N19	I/O	Pulldown	Programmable, 20 µA	2mA ZD	ADC1 event trigger input, or GIO
AD1IN[0]	W14	Input	-	-	-	ADC1 Input
AD1IN[1]	V17	Input	-	-	-	ADC1 Input
AD1IN[2]	V18	Input	-	-	-	ADC1 Input
AD1IN[3]	T17	Input	-	-	-	ADC1 Input
AD1IN[4]	U18	Input	-	-	-	ADC1 Input
AD1IN[5]	R17	Input	-	-	-	ADC1 Input
AD1IN[6]	T19	Input	-	-	-	ADC1 Input
AD1IN[7]	V14	Input	-	-	-	ADC1 Input
AD1IN[8]/AD2IN[8]	P18	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[9]/AD2IN[9]	W17	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[10]/AD2IN[10]	U17	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[11]/AD2IN[11]	U19	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[12]/AD2IN[12]	T16	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[13]/AD2IN[13]	T18	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[14]/AD2IN[14]	R18	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[15]/AD2IN[15]	P19	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[16]/AD2IN[0]	V13	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[17]/AD2IN[1]	U13	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[18]/AD2IN[2]	U14	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[19]/AD2IN[3]	U16	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[20]/AD2IN[4]	U15	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[21]/AD2IN[5]	T15	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[22]/AD2IN[6]	R19	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[23]/AD2IN[7]	R16	Input	-	-	-	ADC1/ADC2 shared Input
AD1IN[24]	N18	Input	-	-	-	ADC1 Input

Table 4-1. ZWT Multibuffered Analog-to-Digital Converters (MibADC1, MibADC2) (continued)

TERMINAL		SIGNAL TYPE	DEFAULT PULL STATE	PULL TYPE	OUTPUT BUFFER DRIVE STRENGTH	DESCRIPTION
SIGNAL NAME	337 ZWT					
AD1IN[25]	P17	Input	-	-	-	ADC1 Input
AD1IN[26]	P16	Input	-	-	-	ADC1 Input
AD1IN[27]	P15	Input	-	-	-	ADC1 Input
AD1IN[28]	R15	Input	-	-	-	ADC1 Input
AD1IN[29]	R14	Input	-	-	-	ADC1 Input
AD1IN[30]	T14	Input	-	-	-	ADC1 Input
AD1IN[31]	T13	Input	-	-	-	ADC1 Input(1)
AD2EVT	T10	I/O	Pulldown	Programmable, 20 µA	2mA ZD	ADC2 event trigger input, or GIO
MIBSPI3NCS[0]/**AD2EVT**/eQEP1I	V10(2)					
AD2IN[16]	W13	Input	-	-	-	ADC2 Input
AD2IN[17]	W12	Input	-	-	-	ADC2 Input
AD2IN[18]	V12	Input	-	-	-	ADC2 Input
AD2IN[19]	U12	Input	-	-	-	ADC2 Input
AD2IN[20]	T11	Input	-	-	-	ADC2 Input
AD2IN[21]	U11	Input	-	-	-	ADC2 Input
AD2IN[22]	V11	Input	-	-	-	ADC2 Input
AD2IN[23]	W11	Input	-	-	-	ADC2 Input
AD2IN[24]	V19	Input	-	-	-	ADC2 Input
AD2IN[24]	W18					
ADREFHI	V15(3)	Input	-	-	-	ADC high reference supply
ADREFLO	V16(3)	Input	-	-	-	ADC low reference supply
MIBSPI3SOMI/**AD1EXT_ENA**/ECAP2	V8	Output	Pullup	20 µA	2mA ZD	External Mux ENA
MIBSPI5SOMI[3]/DMM_DATA[15]/I2C2_SCL/**AD1EXT_ENA**	G16					
MIBSPI3SIMO/**AD1EXT_SEL[0]**/ECAP3	W8	Output	Pullup	20 µA	2mA ZD	External Mux Select 0
MIBSPI5SIMO[1]/DMM_DATA[9]/**AD1EXT_SEL[0]**	E16					
MIBSPI3CLK/**AD1EXT_SEL[1]**/eQEP1A	V9	Output	Pullup	20 µA	2mA ZD	External Mux Select 1
MIBSPI5SIMO[2]/DMM_DATA[10]/**AD1EXT_SEL[1]**	H17					
MIBSPI5SIMO[3]/DMM_DATA[11]/I2C2_SDA/**AD1EXT_SEL[2]**	G17	Output	Pullup	20 µA	2mA ZD	External Mux Select 2
MIBSPI5SOMI[1]/DMM_DATA[13]/**AD1EXT_SEL[3]**	E17	Output	Pullup	20 µA	2mA ZD	External Mux Select 3
MIBSPI5SOMI[2]/DMM_DATA[14]/**AD1EXT_SEL[4]**	H16	Output	Pullup	20 µA	2mA ZD	External Mux Select 4

Table 4-1. ZWT Multibuffered Analog-to-Digital Converters (MibADC1, MibADC2) (continued)

TERMINAL		SIGNAL TYPE	DEFAULT PULL STATE	PULL TYPE	OUTPUT BUFFER DRIVE STRENGTH	DESCRIPTION
SIGNAL NAME	337 ZWT					
VCCAD	W15(3)	Input	-	-	-	Operating supply for ADC
VSSAD	W16(3)	Input	-	-	-	ADC supply ground
VSSAD	W19(3)	Input	-	-	-	ADC supply ground

(1) This ADC channel is also multiplexed with an internal temperature sensor.

(2) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

(3) The ADREFHI, ADREFLO, VCCAD, and VSSAD connections are common for both ADC cores.

4.2.1.2 Enhanced High-End Timer Modules (N2HET)

Table 4-2. ZWT Enhanced High-End Timer Modules (N2HET)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
N2HET1[0]/MIBSPI4CLK/ePWM2B	K18	I/O	Pulldown	Programmable, 20 µA	2 mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[1]/MIBSPI4NENA/N2HET2[8]/eQEP2A	V2	I/O	Pulldown	Programmable, 20 µA	2 mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[2]/MIBSPI4SIMO/ePWM3A	W5	I/O	Pulldown	Programmable, 20 µA	2 mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[3]/MIBSPI4NCS[0]/N2HET2[10]/eQEP2B	U1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[4]/MIBSPI4NCS[1]/ePWM4B	B12	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[5]/MIBSPI4SOMI/N2HET2[12]/ePWM3B	V6	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[6]/SCI3RX/ePWM5A	W3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[7]/MIBSPI4NCS[2]/N2HET2[14]/ePWM7B	T1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[8]/MIBSPI1SIMO[1]/MII_TXD[3]	E18	I/O	Pulldown	Programmable, 20 µA	8mA	N2HET1 time input capture or output compare, or GIO
N2HET1[9]/MIBSPI4NCS[3]/N2HET2[16]/ePWM7A	V7	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[10]/MIBSPI4NCS[4]/MII_TX_CLK/nTZ1_3	D19	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[11]/MIBSPI3NCS[4]/N2HET2[18]/ePWM1SYNCO	E3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[12]/MIBSPI4NCS[5]/MII_CRS/RMII_CRS_DV	B4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[13]/SCI3TX/N2HET2[20]/ePWM5B	N2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[14]	A11	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[15]/MIBSPI1NCS[4]/N2HET2[22]/ECAP1	N1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[16]/ePWM1SYNCI/ePWM1SYNCO	A4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO

Table 4-2. ZWT Enhanced High-End Timer Modules (N2HET) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
N2HET1[17]/EMIF_nOE/SCI4RX	A13	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
MIBSPI1NCS[1]/MII_COL/**N2HET1[17]**/eQEP1S	F3(1)					
N2HET1[18]/EMIF_RNW/ePWM6A	J1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[19]/EMIF_nDQM[0]/SCI4TX	B13	I/O	Pulldown	Programmable, 20 µA	2m A ZD	N2HET1 time input capture or output compare, or GIO
MIBSPI1NCS[2]/MDIO/**N2HET1[19]**	G3(1)					
N2HET1[20]/EMIF_nDQM[1]/ePWM6B	P2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[21]/EMIF_nDQM[2]	H4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
MIBSPI1NCS[3]/**N2HET1[21]**/nTZ1_3	J3(1)					
N2HET1[22]/EMIF_nDQM[3]	B3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[23]/EMIF_BA[0]	J4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
MIBSPI1NENA/MII_RXD[2]/**N2HET1[23]**/ECAP4	G19(1)					
N2HET1[24]/MIBSPI1NCS[5]/MII_RXD[0]/RMII_RXD[0]	P1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[25]	M3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
MIBSPI3NCS[1]/MDCLK/**N2HET1[25]**	V5(1)					
N2HET1[26]/MII_RXD[1]/RMII_RXD[1]	A14	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[27]	A9	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
MIBSPI3NCS[2]/I2C1_SDA/**N2HET1[27]**/nTZ1_2	B2(1)					
N2HET1[28]/MII_RXCLK/RMII_REFCLK	K19	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[29]	A3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
MIBSPI3NCS[3]/I2C1_SCL/**N2HET1[29]**/nTZ1_1	C3(1)					
N2HET1[30]/MII_RX_DV/eQEP2S	B11	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
N2HET1[31]	J17	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET1 time input capture or output compare, or GIO
MIBSPI3NENA/MIBSPI3NCS[5]/**N2HET1[31]**/eQEP1B	W9(1)					
N2HET2[0]	D6	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
GIOA[2]/**N2HET2[0]**/eQEP2I	C1(1)					
N2HET2[1]/N2HET1_NDIS	D8	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
EMIF_ADDR[0]/**N2HET2[1]**	D4(1)					

Table 4-2. ZWT Enhanced High-End Timer Modules (N2HET) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
N2HET2[2]/N2HET2_NDIS	D7	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
GIOA[3]/**N2HET2[2]**	E1(1)					
N2HET2[3]/MIBSPI2CLK	E2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
EMIF_ADDR[1]/**N2HET2[3]**	D5(1)					
N2HET2[4]	D13	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
GIOA[6]/**N2HET2[4]**/ePWM1B	H3(1)					
N2HET2[5]	D12	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
EMIF_BA[1]/**N2HET2[5]**	D16(1)					
N2HET2[6]	D11	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
GIOA[7]/**N2HET2[6]**/ePWM2A	M1(1)					
N2HET2[7]/MIBSPI2NCS[0]	N3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
EMIF_nCS[0]/RTP_DATA[15]/**N2HET2[7]**	N17(1)					
N2HET2[8]	K16	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET1[1]/MIBSPI4NENA/**N2HET2[8]**/eQEP2A	V2(1)					
N2HET2[9]	L16	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
EMIF_nCS[3]/RTP_DATA[14]/**N2HET2[9]**	K17(1)					
N2HET2[10]	M16	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET1[3]/MIBSPI4NCS[0]/**N2HET2[10]**/eQEP2B	U1(1)					
N2HET2[11]	N16	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
EMIF_ADDR[6]/RTP_DATA[13]/**N2HET2[11]**	C4(1)					
N2HET2[12]/MIBSPI2NENA/MIBSPI2NCS[1]	D3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET1[5]/MIBSPI4SOMI/**N2HET2[12]**/ePWM3B	V6(1)					
N2HET2[13]/MIBSPI2SOMI	D2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
EMIF_ADDR[7]/RTP_DATA[12]/**N2HET2[13]**	C5(1)					
N2HET2[14]/MIBSPI2SIMO	D1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET1[7]/MIBSPI4NCS[2]/**N2HET2[14]**/ePWM7B	T1(1)					
N2HET2[15]	K4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
EMIF_ADDR[8]/RTP_DATA[11]/**N2HET2[15]**	C6(1)					
N2HET2[16]	L4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET1[9]/MIBSPI4NCS[3]/**N2HET2[16]**/ePWM7A	V7(1)					
N2HET2[17]	M4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO

Table 4-2. ZWT Enhanced High-End Timer Modules (N2HET) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
N2HET2[18]	N4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET1[11]/MIBSPI3NCS[4]/**N2HET2[18]**/ePWM1SYNCO	E3(1)					
N2HET2[19]/LIN2RX	P4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET2[20]/LIN2TX	T5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET1[13]/SCI3TX/**N2HET2[20]**/ePWM5B	N2(1)					
N2HET2[21]	T6	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET2[22]	T7	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET1[15]/MIBSPI1NCS[4]/**N2HET2[22]**/ECAP1	N1(1)					
N2HET2[23]	T8	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
ETMDATA[24]/EMIF_DATA[8]/**N2HET2[24]**/MIBSPI5NCS[4]	L5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
ETMDATA[25]/EMIF_DATA[9]/**N2HET2[25]**/MIBSPI5NCS[5]	M5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
ETMDATA[26]/EMIF_DATA[10]/**N2HET2[26]**	N5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
ETMDATA[27]/EMIF_DATA[11]/**N2HET2[27]**	P5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
ETMDATA[28]/EMIF_DATA[12]/**N2HET2[28]**/GIOA[0]	R5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
ETMDATA[29]/EMIF_DATA[13]/**N2HET2[29]**/GIOA[1]	R6	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
ETMDATA[30]/EMIF_DATA[14]/**N2HET2[30]**/GIOA[3]	R7	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
ETMDATA[31]/EMIF_DATA[15]/**N2HET2[31]**/GIOA[4]	R8	I/O	Pulldown	Programmable, 20 µA	2mA ZD	N2HET2 time input capture or output compare, or GIO
N2HET2[1]/**N2HET1_NDIS**	D8	Input	Pulldown	Fixed, 20 µA	2mA ZD	N2HET1 Disable
N2HET2[2]/**N2HET2_NDIS**	D7	Input	Pulldown	Fixed, 20 µA	2mA ZD	N2HET2 Disable

(1) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

4.2.1.3 RAM Trace Port (RTP)

Table 4-3. ZWT RAM Trace Port (RTP)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
EMIF_ADDR[21]/**RTP_CLK**	C17	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet clock, or GIO
EMIF_ADDR[18]/**RTP_DATA[0]**	D15	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[17]/**RTP_DATA[1]**	C14	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[16]/**RTP_DATA[2]**	D14	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[15]/**RTP_DATA[3]**	C13	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[14]/**RTP_DATA[4]**	C12	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[13]/**RTP_DATA[5]**	C11	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[12]/**RTP_DATA[6]**	C10	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_nCS[4]/**RTP_DATA[7]**/GIOB[5]	M17	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[11]/**RTP_DATA[8]**	C9	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[10]/**RTP_DATA[9]**	C8	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[9]/**RTP_DATA[10]**	C7	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[8]/**RTP_DATA[11]**/N2HET2[15]	C6	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[7]/**RTP_DATA[12]**/N2HET2[13]	C5	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[6]/**RTP_DATA[13]**/N2HET2[11]	C4	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_nCS[3]/**RTP_DATA[14]**/N2HET2[9]	K17	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_nCS[0]/**RTP_DATA[15]**/N2HET2[7]	N17	I/O	Pulldown	Programmable, 20 µA	8mA	RTP packet data, or GIO
EMIF_ADDR[19]/**RTP_nENA**	C15	I/O	Pullup	Programmable, 20 µA	8mA	RTP packet handshake, or GIO
EMIF_ADDR[20]/**RTP_nSYNC**	C16	I/O	Pullup	Programmable, 20 µA	8mA	RTP synchronization, or GIO

(1) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

4.2.1.4 Enhanced Capture Modules (eCAP)

Table 4-4. ZWT Enhanced Capture Modules (eCAP)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
N2HET1[15]/MIBSPI1NCS[4]/N2HET2[22]/**ECAP1**	N1	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced Capture Module 1 I/O
MIBSPI3SOMI/AD1EXT_ENA/**ECAP2**	V8	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced Capture Module 2 I/O
MIBSPI3SIMO/AD1EXT_SEL[0]/**ECAP3**	W8	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced Capture Module 3 I/O
MIBSPI1NENA/MII_RXD[2]/N2HET1[23]/**ECAP4**	G19	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced Capture Module 4 I/O
MIBSPI5NENA/DMM_DATA[7]/MII_RXD[3]/**ECAP5**	H18	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced Capture Module 5 I/O
MIBSPI1NCS[0]/MIBSPI1SOMI[1]/MII_TXD[2]/**ECAP6**	R2	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced Capture Module 6 I/O

4.2.1.5 Enhanced Quadrature Encoder Pulse Modules (eQEP)

Table 4-5. ZWT Enhanced Quadrature Encoder Pulse Modules (eQEP)(1)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
MIBSPI3CLK/AD1EXT_SEL[1]/**eQEP1A**	V9	Input	Pullup	Fixed, 20 µA	-	Enhanced QEP1 Input A
MIBSPI3NENA/MIBSPI3NCS[5]/N2HET1[31]/**eQEP1B**	W9	Input	Pullup	Fixed, 20 µA	-	Enhanced QEP1 Input B
MIBSPI3NCS[0]/AD2EVT/**eQEP1I**	V10	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced QEP1 Index
MIBSPI1NCS[1]/MII_COL/N2HET1[17]/**eQEP1S**	F3	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced QEP1 Strobe
N2HET1[1]/MIBSPI4NENA/N2HET2[8]/**eQEP2A**	V2	Input	Pullup	Fixed, 20 µA	-	Enhanced QEP2 Input A
N2HET1[3]/MIBSPI4NCS[0]/N2HET2[10]/**eQEP2B**	U1	Input	Pullup	Fixed, 20 µA	-	Enhanced QEP2 Input B
GIOA[2]/N2HET2[0]/**eQEP2I**	C1	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced QEP2 Index
N2HET1[30]/MII_RX_DV/**eQEP2S**	B11	I/O	Pullup	Fixed, 20 µA	8mA	Enhanced QEP2 Strobe

(1) These signals are double-synchronized and then optionally filtered with a 6-cycle VCLK4-based counter.

4.2.1.6 Enhanced Pulse-Width Modulator Modules (ePWM)

Table 4-6. ZWT Enhanced Pulse-Width Modulator Modules (ePWM)

TERMINAL		SIGNAL TYPE	DEFAULT PULL STATE	PULL TYPE	OUTPUT BUFFER DRIVE STRENGTH	DESCRIPTION
SIGNAL NAME	337 ZWT					
ePWM1A	D9	Output	–	–	8 mA	Enhanced PWM1 Output A
GIOA[5]/EXTCLKIN1/**ePWM1A**	B5[1]					
ePWM1B	D10	Output	–	–	8 mA	Enhanced PWM1 Output B
GIOA[6]/N2HET2[4]/**ePWM1B**	H3[1]					
N2HET1[16]/**ePWM1SYNCI**/ePWM1SYNCO	A4	Input	Pulldown	Fixed, 20 µA	–	External ePWM Sync Pulse Input
N2HET1[11]/MIBSPI3NCS[4]/N2HET2[18]/ **ePWM1SYNCO**	E3	Output	Pulldown	20 µA	2mA ZD	External ePWM Sync Pulse Output
N2HET1[16]/ePWM1SYNCI/**ePWM1SYNCO**	A4[1]					
GIOA[7]/N2HET2[6]/**ePWM2A**	M1	Output	Pulldown	20 µA	8 mA	Enhanced PWM2 Output A
N2HET1[0]/MIBSPI4CLK/**ePWM2B**	K18	Output	Pulldown	20 µA	8 mA	Enhanced PWM2 Output B
N2HET1[2]/MIBSPI4SIMO/**ePWM3A**	W5	Output	Pulldown	20 µA	8 mA	Enhanced PWM3 Output A
N2HET1[5]/MIBSPI4SOMI/N2HET2[12]/**ePWM3B**	V6	Output	Pulldown	20 µA	8 mA	Enhanced PWM3 Output B
MIBSPI5NCS[0]/DMM_DATA[5]/**ePWM4A**	E19	Output	Pulldown	20 µA	8 mA	Enhanced PWM4 Output A
N2HET1[4]/MIBSPI4NCS[1]/**ePWM4B**	B12	Output	Pulldown	20 µA	8 mA	Enhanced PWM4 Output B
N2HET1[6]/SCI3RX/**ePWM5A**	W3	Output	Pulldown	20 µA	8 mA	Enhanced PWM5 Output A
N2HET1[13]/SCI3TX/N2HET2[20]/**ePWM5B**	N2	Output	Pulldown	20 µA	8 mA	Enhanced PWM5 Output B
N2HET1[18]/EMIF_RNW/**ePWM6A**	J1	Output	–	–	8 mA	Enhanced PWM6 Output A
N2HET1[20]/EMIF_nDQM[1]/**ePWM6B**	P2	Output	–	–	8 mA	Enhanced PWM6 Output B
N2HET1[9]/MIBSPI4NCS[3]/N2HET2[16]/**ePWM7A**	V7	Output	–	–	8 mA	Enhanced PWM7 Output A
N2HET1[7]/MIBSPI4NCS[2]/N2HET2[14]/**ePWM7B**	T1	Output	–	–	8 mA	Enhanced PWM7 Output B
AD1EVT/MII_RX_ER/RMII_RX_ER/**nTZ1_1**	N19	Input	Pulldown	Fixed, 20 µA	–	Trip Zone 1 Input 1
MIBSPI3NCS[3]/I2C1_SCL/N2HET1[29]/**nTZ1_1**	C3[1]					
GIOB[7]/**nTZ1_2**	F1	Input	Pulldown	Fixed, 20 µA	–	Trip Zone 1 Input 2
MIBSPI3NCS[2]/I2C1_SDA/N2HET1[27]/**nTZ1_2**	B2[1]					
MIBSPI1NCS[3]/N2HET1[21]/**nTZ1_3**	J3	Input	Pullup	Fixed, 20 µA	–	Trip Zone 1 Input 3
N2HET1[10]/MIBSPI4NCS[4]/MII_TX_CLK/**nTZ1_3**	D19[1]					

(1) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

4.2.1.7 Data Modification Module (DMM)

Table 4-7. ZWT Data Modification Module (DMM)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
DMM_CLK	F17	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM clock, or GIO
DMM_DATA[0]	L19	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
DMM_DATA[1]	L18	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5NCS[2]/**DMM_DATA[2]**	W6	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5NCS[3]/**DMM_DATA[3]**	T12	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5CLK/**DMM_DATA[4]**/MII_TXEN/RMII_TXEN	H19	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5NCS[0]/**DMM_DATA[5]**/ePWM4A	E19	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5NCS[1]/**DMM_DATA[6]**	B6	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5NENA/**DMM_DATA[7]**/MII_RXD[3]/ECAP5	H18	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5SIMO[0]/**DMM_DATA[8]**/MII_TXD[1]/RMII_TXD[1]	J19	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5SIMO[1]/**DMM_DATA[9]**/AD1EXT_SEL[0]	E16	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5SIMO[2]/**DMM_DATA[10]**/AD1EXT_SEL[1]	H17	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5SIMO[3]/**DMM_DATA[11]**/I2C2_SDA/AD1EXT_SEL[2]	G17	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5SOMI[0]/**DMM_DATA[12]**/MII_TXD[0]/RMII_TXD[0]	J18	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5SOMI[1]/**DMM_DATA[13]**/AD1EXT_SEL[3]	E17	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5SOMI[2]/**DMM_DATA[14]**/AD1EXT_SEL[4]	H16	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
MIBSPI5SOMI[3]/**DMM_DATA[15]**/I2C2_SCL/AD1EXT_ENA	G16	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM data, or GIO
DMM_nENA	F16	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM handshake, or GIO
DMM_SYNC	J16	I/O	Pullup	Programmable, 20 µA	2mA ZD	DMM synchronization, or GIO

4.2.1.8 General-Purpose Input / Output (GIO)

Table 4-8. ZWT General-Purpose Input / Output (GIO)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
GIOA[0]	A5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
ETMDATA[28]/EMIF_DATA[12]/N2HET2[28]/**GIOA[0]**	R5(1)					
GIOA[1]	C2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
ETMDATA[29]/EMIF_DATA[13]/N2HET2[29]/**GIOA[1]**	R6(1)					
GIOA[2]/N2HET2[0]/eQEP2I	C1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
FRAYTX1/**GIOA[2]**	B15(1)					
GIOA[3]/N2HET2[2]	E1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
ETMDATA[30]/EMIF_DATA[14]/N2HET2[30]/**GIOA[3]**	R7(1)					
GIOA[4]	A6	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
ETMDATA[31]/EMIF_DATA[15]/N2HET2[31]/**GIOA[4]**	R8(1)					
GIOA[5]/EXTCLKIN1/ePWM1A	B5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
ETMTRACECLKIN/EXTCLKIN2/**GIOA[5]**	R9(1)					
GIOA[6]/N2HET2[4]/ePWM1B	H3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
ETMTRACECLKOUT/**GIOA[6]**	R10(1)					
GIOA[7]/N2HET2[6]/ePWM2A	M1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
ETMTRACECTL/**GIOA[7]**	R11(1)					
GIOB[0]	M2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
FRAYTX2/**GIOB[0]**	B8(1)					
GIOB[1]	K2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
FRAYTXEN1/**GIOB[1]**	B16(1)					
GIOB[2]/DCAN4TX	F2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
FRAYTXEN2/**GIOB[2]**	B9(1)					
GIOB[3]/DCAN4RX	W10	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
EMIF_nCAS/**GIOB[3]**	R4(1)					
GIOB[4]	G1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
EMIF_nCS[2]/**GIOB[4]**	L17(1)					
GIOB[5]	G2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
EMIF_nCS[4]/RTP_DATA[7]/**GIOB[5]**	M17(1)					

Table 4-8. ZWT General-Purpose Input / Output (GIO) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
GIOB[6]/nERROR	J2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
EMIF_nRAS/**GIOB[6]**	R3(1)					
GIOB[7]/nTZ1_2	F1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	General-purpose I/O, external interrupt capable
EMIF_nWAIT/**GIOB[7]**	P3(1)					

(1) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

4.2.1.9 FlexRay Interface Controller (FlexRay)

Table 4-9. FlexRay Interface Controller (FlexRay)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
FRAYRX1	A15	Input	Pullup	Fixed, 100 µA	–	FlexRay data receive (channel 1)
FRAYRX2	A8	Input	Pullup	Fixed, 100 µA	–	FlexRay data receive (channel 2)
FRAYTX1/GIOA[2]	B15	Output	Pulldown	20 µA	8mA	FlexRay data transmit (channel 1)
FRAYTX2/GIOB[0]	B8	Output	Pulldown	20 µA	8mA	FlexRay data transmit (channel 2)
FRAYTXEN1/GIOB[1]	B16	Output	Pulldown	20 µA	8mA	FlexRay transmit enable (channel 1)
FRAYTXEN2/GIOB[2]	B9	Output	Pulldown	20 µA	8mA	FlexRay transmit enable (channel 2)

4.2.1.10 Controller Area Network Controllers (DCAN)

Table 4-10. ZWT Controller Area Network Controllers (DCAN)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
DCAN1RX	B10	I/O	Pullup	Programmable, 20 µA	2mA ZD	CAN1 receive, or GIO
DCAN1TX	A10	I/O	Pullup	Programmable, 20 µA	2mA ZD	CAN1 transmit, or GIO
DCAN2RX	H1	I/O	Pullup	Programmable, 20 µA	2mA ZD	CAN2 receive, or GIO
DCAN2TX	H2	I/O	Pullup	Programmable, 20 µA	2mA ZD	CAN2 transmit, or GIO
DCAN3RX	M19	I/O	Pullup	Programmable, 20 µA	2mA ZD	CAN3 receive, or GIO
DCAN3TX	M18	I/O	Pullup	Programmable, 20 µA	2mA ZD	CAN3 transmit, or GIO
GIOB[3]/**DCAN4RX**	W10	I/O	Pulldown	Programmable, 20 µA	2mA ZD	CAN4 receive, or GIO
GIOB[2]/**DCAN4TX**	F2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	CAN4 transmit, or GIO

4.2.1.11 Local Interconnect Network Interface Module (LIN)

Table 4-11. ZWT Local Interconnect Network Interface Module (LIN)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
LIN1RX	A7	I/O	Pullup	Programmable, 20 µA	2mA ZD	LIN receive, or GIO
LIN1TX	B7	I/O	Pullup	Programmable, 20 µA	2mA ZD	LIN transmit, or GIO
N2HET2[19]/**LIN2RX**	P4	I/O	Pulldown	Programmable, 20 µA	2mA ZD	LIN receive, or GIO
N2HET2[20]/**LIN2TX**	T5	I/O	Pulldown	Programmable, 20 µA	2mA ZD	LIN transmit, or GIO

4.2.1.12 Standard Serial Communication Interface (SCI)

Table 4-12. ZWT Standard Serial Communication Interface (SCI)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
N2HET1[6]/**SCI3RX**/ePWM5A	W3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	SCI receive, or GIO
N2HET1[13]/**SCI3TX**/N2HET2[20]/ePWM5B	N2	I/O	Pulldown	Programmable, 20 µA	2mA ZD	SCI transmit, or GIO
N2HET1[17]/EMIF_nOE/**SCI4RX**	A13	I/O	Pulldown	Programmable, 20 µA	2mA ZD	SCI receive, or GIO
N2HET1[19]/EMIF_nDQM[0]/**SCI4TX**	B13	I/O	Pulldown	Programmable, 20 µA	2mA ZD	SCI transmit, or GIO

4.2.1.13 Inter-Integrated Circuit Interface Module (I2C)

Table 4-13. ZWT Inter-Integrated Circuit Interface Module (I2C)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
MIBSPI3NCS[3]/**I2C1_SCL**/N2HET1[29]/nTZ1_1	C3	I/O	Pullup	Programmable, 20 µA	2mA ZD	I2C serial clock, or GIO
MIBSPI3NCS[2]/**I2C1_SDA**/N2HET1[27]/nTZ1_2	B2	I/O	Pullup	Programmable, 20uA	2mA ZD	I2C serial data, or GIO
MIBSPI5SOMI[3]/DMM_DATA[15]/**I2C2_SCL**/AD1EXT_ENA	G16	I/O	Pullup	Programmable, 20uA	2mA ZD	I2C serial clock, or GIO
MIBSPI5SIMO[3]/DMM_DATA[11]/**I2C2_SDA**/AD1EXT_SEL[2]	G17	I/O	Pullup	Programmable, 20uA	2mA ZD	I2C serial data, or GIO

4.2.1.14 Multibuffered Serial Peripheral Interface Modules (MibSPI)

Table 4-14. ZWT Multibuffered Serial Peripheral Interface Modules (MibSPI)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
MIBSPI1CLK	F18	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI1 clock, or GIO
MIBSPI1NCS[0]/MIBSPI1SOMI[1]/MII_TXD[2]/ECAP6	R2	I/O	Pullup	Programmable, 20 >µA	8mA	MibSPI1 chip select, or GIO
MIBSPI1NCS[1]/MII_COL/N2HET1[17]/eQEP1S	F3	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI1 chip select, or GIO
MIBSPI1NCS[2]/MDIO /N2HET1[19]	G3	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI1 chip select, or GIO
MIBSPI1NCS[3]/N2HET1[21]/nTZ1_3	J3	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI1 chip select, or GIO
MIBSPI1NCS[4]	U10	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI1 chip select, or GIO
N2HET1[15]/**MIBSPI1NCS[4]**/N2HET2[22]/ECAP1	N1(1)					
MIBSPI1NCS[5]	U9	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI1 chip select, or GIO
N2HET1[24]/**MIBSPI1NCS[5]**/MII_RXD[0]/RMII_RXD[0]	P1(1)					
MIBSPI1NENA/MII_RXD[2]/N2HET1[23]/ECAP4	G19	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI1 enable, or GIO
MIBSPI1SIMO[0]	F19	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI1 slave-in master-out, or GIO
N2HET1[8]/**MIBSPI1SIMO[1]**/MII_TXD[3]	E18	I/O	Pulldown	Programmable, 20 µA	8mA	MibSPI1 slave-in master-out, or GIO
MIBSPI1SOMI[0]	G18	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI1 slave-out master-in, or GIO
MIBSPI1NCS[0]/**MIBSPI1SOMI[1]**/MII_TXD[2]/ECAP6	R2	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI1 slave-out master-in, or GIO
N2HET2[3]/**MIBSPI2CLK**	E2	I/O	Pulldown	Programmable, 20 µA	8mA	MibSPI2 clock, or GIO
N2HET2[7]/**MIBSPI2NCS[0]**	N3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI2 chip select, or GIO
N2HET2[12]/MIBSPI2NENA/**MIBSPI2NCS[1]**	D3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI2 chip select, or GIO
N2HET2[12]/**MIBSPI2NENA**/MIBSPI2NCS[1]	D3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI2 enable, or GIO
N2HET2[14]/**MIBSPI2SIMO**	D1	I/O	Pulldown	Programmable, 20 µA	8mA	MibSPI2 slave-in master-out, or GIO
N2HET2[13]/**MIBSPI2SOMI**	D2	I/O	Pulldown	Programmable, 20 µA	8mA	MibSPI2 slave-out master-in, or GIO
MIBSPI3CLK/AD1EXT_SEL[1]/eQEP1A	V9	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI3 clock, or GIO
MIBSPI3NCS[0]/AD2EVT/eQEP1I	V10	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI3 chip select, or GIO
MIBSPI3NCS[1]/MDCLK/N2HET1[25]	V5	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI3 chip select, or GIO
MIBSPI3NCS[2]/I2C1_SDA/N2HET1[27] /nTZ1_2	B2	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI3 chip select, or GIO
MIBSPI3NCS[3]/I2C1_SCL/N2HET1[29] /nTZ1_1	C3	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI3 chip select, or GIO
N2HET1[11]/**MIBSPI3NCS[4]**/N2HET2[18]/ePWM1SYNCO	E3	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI3 chip select, or GIO
MIBSPI3NENA/**MIBSPI3NCS[5]**/N2HET1[31]/eQEP1B	W9	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI3 chip select, or GIO
MIBSPI3NENA/MIBSPI3NCS[5]/N2HET1[31]/eQEP1B	W9	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI3 enable, or GIO
MIBSPI3SIMO/AD1EXT_SEL[0]/ECAP3	W8	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI3 slave-in master-out, or GIO
MIBSPI3SOMI/AD1EXT_ENA/ECAP2	V8	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI3 slave-out master-in, or GIO
N2HET1[0]/**MIBSPI4CLK**/ePWM2B	K18	I/O	Pulldown	Programmable, 20 µA	8mA	MibSPI4 clock, or GIO
N2HET1[3]/**MIBSPI4NCS[0]**/N2HET2[10]/eQEP2B	U1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI4 chip select, or GIO

Table 4-14. ZWT Multibuffered Serial Peripheral Interface Modules (MibSPI) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
N2HET1[4]/**MIBSPI4NCS[1]**/ePWM4B	B12	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI4 chip select, or GIO
N2HET1[7]/**MIBSPI4NCS[2]**/N2HET2[14]/ePWM7B	T1	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI4 chip select, or GIO
N2HET1[9]/**MIBSPI4NCS[3]**/N2HET2[16]/ePWM7A	V7	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI4 chip select, or GIO
N2HET1[10]/**MIBSPI4NCS[4]**/MII_TX_CLK/nTZ1_3	D19	I/O	Pulldown	Programmable, 20 µA	2mA ZD	MibSPI4 chip select, or GIO
N2HET1[12]/**MIBSPI4NCS[5]**/MII_CRS/RMII_CRS_DV	B4	I/O	Pulldown	Programmable, 20 µA	4mA	MibSPI4 chip select, or GIO
N2HET1[1]/**MIBSPI4NENA**/N2HET2[8]/eQEP2A	V2	I/O	Pulldown	Programmable, 20 µA	8mA	MibSPI4 enable, or GIO
N2HET1[2]/**MIBSPI4SIMO**/ePWM3A	W5	I/O	Pulldown	Programmable, 20 µA	8mA	MibSPI4 slave-in master-out, or GIO
N2HET1[5]/**MIBSPI4SOMI**/N2HET2[12]/ePWM3B	V6	I/O	Pulldown	Programmable, 20 µA	8mA	MibSPI4 slave-out master-in, or GIO
MIBSPI5CLK/DMM_DATA[4]/MII_TXEN/RMII_TXEN	H19	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 clock, or GIO
MIBSPI5NCS[0]/DMM_DATA[5]/ePWM4A	E19	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI5 chip select, or GIO
MIBSPI5NCS[1]/DMM_DATA[6]	B6	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI5 chip select, or GIO
MIBSPI5NCS[2]/DMM_DATA[2]	W6	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI5 chip select, or GIO
MIBSPI5NCS[3]/DMM_DATA[3]	T12	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI5 chip select, or GIO
ETMDATA[24]/EMIF_DATA[8]/N2HET2[24]/**MIBSPI5NCS[4]**	L5	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI5 chip select, or GIO
ETMDATA[25]/EMIF_DATA[9]/N2HET2[25]/**MIBSPI5NCS[5]**	M5	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI5 chip select, or GIO
MIBSPI5NENA/DMM_DATA[7] /MII_RXD[3]/ECAP5	H18	I/O	Pullup	Programmable, 20 µA	2mA ZD	MibSPI5 enable, or GIO
MIBSPI5SIMO[0]/DMM_DATA[8]/MII_TXD[1]/RMII_TXD[1]	J19	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 slave-in master-out, or GIO
MIBSPI5SIMO[1]/DMM_DATA[9]/AD1EXT_SEL[0]	E16	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 slave-in master-out, or GIO
MIBSPI5SIMO[2]/DMM_DATA[10]/AD1EXT_SEL[1]	H17	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 slave-in master-out, or GIO
MIBSPI5SIMO[3]/DMM_DATA[11]/I2C2_SDA/AD1EXT_SEL[2]	G17	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 slave-in master-out, or GIO
MIBSPI5SOMI[0]/DMM_DATA[12]/MII_TXD[0]/RMII_TXD[0]	J18	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 slave-out master-in, or GIO
MIBSPI5SOMI[1]/DMM_DATA[13]/AD1EXT_SEL[3]	E17	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 slave-out master-in, or GIO
MIBSPI5SOMI[2]/DMM_DATA[14]/AD1EXT_SEL[4]	H16	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 slave-out master-in, or GIO
MIBSPI5SOMI[3]/DMM_DATA[15]/I2C2_SCL/AD1EXT_ENA	G16	I/O	Pullup	Programmable, 20 µA	8mA	MibSPI5 slave-out master-in, or GIO

(1) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

4.2.1.15 Ethernet Controller

Table 4-15. ZWT Ethernet Controller: MDIO Interface

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
MDCLK	T9	Output	-	-	8mA	Serial clock output
MIBSPI3NCS[1]/**MDCLK**/N2HET1[25]	V5(1)					
MDIO	F4	I/O	Pulldown	Fixed, 20 µA	8mA	Serial data input/output
MIBSPI1NCS[2]/**MDIO**/N2HET1[19]	G3(1)					

(1) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

Table 4-16. ZWT Ethernet Controller: Reduced Media Independent Interface (RMII)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
N2HET1[12]/MIBSPI4NCS[5]/MII_CRS/**RMII_CRS_DV**	B4	Input	Pulldown	Fixed, 20 µA	-	RMII carrier sense and data valid
N2HET1[28]/MII_RXCLK/**RMII_REFCLK**	K19	Input	Pulldown	Fixed, 20 µA	8mA	EMII synchronous reference clock for receive, transmit and control interface
AD1EVT/MII_RX_ER/**RMII_RX_ER**/nTZ1_1	N19	Input	Pulldown	Fixed, 20 µA	-	RMII receive error
N2HET1[24]/MIBSPI1NCS[5]/MII_RXD[0]/**RMII_RXD[0]**	P1	Input	Pulldown	Fixed, 20 µA	-	RMII receive data
N2HET1[26]/MII_RXD[1]/**RMII_RXD[1]**	A14	Input	Pulldown	Fixed, 20 µA	-	RMII receive data
MIBSPI5SOMI[0]/DMM_DATA[12]/MII_TXD[0]/**RMII_TXD[0]**	J18	Output	Pullup	20 µA	8mA	RMII transmit data
MIBSPI5SIMO[0]/DMM_DATA[8]/MII_TXD[1]/**RMII_TXD[1]**	J19	Output	Pullup	20 µA	8mA	RMII transmit data
MIBSPI5CLK/DMM_DATA[4]/MII_TXEN/**RMII_TXEN**	H19	Output	Pullup	20 µA	8mA	RMII transmit enable

Table 4-17. ZWT Ethernet Controller: Media Independent Interface (MII)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
MII_COL	W4	Input	Pullup	Fixed, 20 µA	-	Collision detect
MIBSPI1NCS[1]/**MII_COL/**N2HET1[17]/eQEP1S	F3(1)					
MII_CRS	V4	Input	Pulldown	Fixed, 20 µA	-	Carrier sense and receive valid
N2HET1[12]/MIBSPI4NCS[5]/**MII_CRS**/RMII_CRS_DV	B4(1)					
MII_RX_DV	U6	Input	Pulldown	Fixed, 20 µA	-	Received data valid
N2HET1[30]/**MII_RX_DV**/eQEP2S	B11(1)					
MII_RX_ER	U5	Input	Pulldown	Fixed, 20 µA	-	Receive error
AD1EVT/**MII_RX_ER**/RMII_RX_ER/nTZ1_1	N19(1)					
MII_RXCLK	T4	Input	Pulldown	Fixed, 20 µA	-	Receive clock
N2HET1[28]/**MII_RXCLK**/RMII_REFCLK	K19(1)					
MII_RXD[0]	U4	Input	Pulldown	Fixed, 20 µA	-	Receive data
N2HET1[24]/MIBSPI1NCS[5]/**MII_RXD[0]**/RMII_RXD[0]	P1(1)					
MII_RXD[1]	T3	Input	Pulldown	Fixed, 20 µA	-	Receive data
N2HET1[26]/**MII_RXD[1]**/RMII_RXD[1]	A14(1)					
MII_RXD[2]	U3	Input	Pulldown	Fixed, 20 µA	-	Receive data
MIBSPI1NENA/**MII_RXD[2]**/N2HET1[23]/ECAP4	G19(1)					
MII_RXD[3]	V3	Input	Pulldown	Fixed, 20 µA	-	Receive data
MIBSPI5NENA/DMM_DATA[7]/**MII_RXD[3]**/ECAP5	H18(1)					
MII_TX_CLK	U7	Input	Pulldown	Fixed, 20 µA	-	Transmit clock
N2HET1[10]/MIBSPI4NCS[4]/**MII_TX_CLK**/nTZ1_3	D19(1)					
MII_TXD[0]	U8	Output	-	-	8mA	Transmit data
MIBSPI5SOMI[0]/DMM_DATA[12]/**MII_TXD[0]**/RMII_TXD[0]	J18(1)					
MII_TXD[1]	R1	Output	-	-	8mA	Transmit data
MIBSPI5SIMO[0]/DMM_DATA[8]/**MII_TXD[1]**/RMII_TXD[1]	J19(1)					
MII_TXD[2]	T2	Output	-	-	8mA	Transmit data
MIBSPI1NCS[0]/MIBSPI1SOMI[1]/**MII_TXD[2]**/ECAP6	R2(1)					
MII_TXD[3]	G4	Output	-	-	8mA	Transmit data
N2HET1[8]/MIBSPI1SIMO[1]/**MII_TXD[3]**	E18(1)					

Table 4-17. ZWT Ethernet Controller: Media Independent Interface (MII) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
MII_TXEN	E4	Output	-	-	8mA	Transmit enable
MIBSPI5CLK/DMM_DATA[4]/**MII_TXEN**/RMII_TXEN	H19(1)					

(1) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

4.2.1.16 External Memory Interface (EMIF)

Table 4-18. External Memory Interface (EMIF)(2)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
EMIF_ADDR[0]/N2HET2[1]	D4	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[1]/N2HET2[3]	D5	Output	Pulldown	20 µA	8mA	EMIF address
ETMDATA[11]/**EMIF_ADDR[2]**	E6	Output	-	-	8mA	EMIF address
ETMDATA[10]/**EMIF_ADDR[3]**	E7	Output	-	-	8mA	EMIF address
ETMDATA[9]/**EMIF_ADDR[4]**	E8	Output	-	-	8mA	EMIF address
ETMDATA[8]/**EMIF_ADDR[5]**	E9	Output	-	-	8mA	EMIF address
EMIF_ADDR[6]/RTP_DATA[13]/N2HET2[11]	C4	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[7]/RTP_DATA[12]/N2HET2[13]	C5	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[8]/RTP_DATA[11]/N2HET2[15]	C6	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[9]/RTP_DATA[10]	C7	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[10]/RTP_DATA[9]	C8	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[11]/RTP_DATA[8]	C9	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[12]/RTP_DATA[6]	C10	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[13]/RTP_DATA[5]	C11	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[14]/RTP_DATA[4]	C12	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[15]/RTP_DATA[3]	C13	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[16]/RTP_DATA[2]	D14	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[17]/RTP_DATA[1]	C14	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[18]/RTP_DATA[0]	D15	Output	Pulldown	20 µA	8mA	EMIF address
EMIF_ADDR[19]/RTP_nENA	C15	Output	Pullup	20 µA	8mA	EMIF address
EMIF_ADDR[20]/RTP_nSYNC	C16	Output	Pullup	20 µA	8mA	EMIF address
EMIF_ADDR[21]/RTP_CLK	C17	Output	Pulldown	20 µA	8mA	EMIF address
ETMDATA[12]/**EMIF_BA[0]**	E13	Output	Pulldown	20 µA	8mA	EMIF bank address or address line
N2HET1[23]/**EMIF_BA[0]**	J4(1)					
EMIF_BA[1]/N2HET2[5]	D16	Output	Pulldown	20 µA	8mA	EMIF bank address or address line

Table 4-18. External Memory Interface (EMIF)(2) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
EMIF_CKE	L3	Output	-	-	8mA	EMIF clock enable
EMIF_CLK/ECLK2	K3	Output	Pulldown	20 µA	8mA	EMIF clock
ETMDATA[16]/**EMIF_DATA[0]**	K15	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[17]/**EMIF_DATA[1]**	L15	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[18]/**EMIF_DATA[2]**	M15	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[19]/**EMIF_DATA[3]**	N15	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[20]/**EMIF_DATA[4]**	E5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[21]/**EMIF_DATA[5]**	F5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[22]/**EMIF_DATA[6]**	G5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[23]/**EMIF_DATA[7]**	K5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[24]/**EMIF_DATA[8]**/N2HET2[24]/MIBSPI5NCS[4]	L5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[25]/**EMIF_DATA[9]**/N2HET2[25]/MIBSPI5NCS[5]	M5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[26]/**EMIF_DATA[10]**/N2HET2[26]	N5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[27]/**EMIF_DATA[11]**/N2HET2[27]	P5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[28]/**EMIF_DATA[12]**/N2HET2[28]/GIOA[0]	R5	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[29]/**EMIF_DATA[13]**/N2HET2[29]/GIOA[1]	R6	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[30]/**EMIF_DATA[14]**/N2HET2[30]/GIOA[3]	R7	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
ETMDATA[31]/**EMIF_DATA[15]**/N2HET2[31]/GIOA[4]	R8	I/O	Pulldown	Fixed, 20 µA	8mA	EMIF data
EMIF_nCAS/GIOB[3]	R4	Output	Pulldown	20 µA	8mA	EMIF column address strobe
EMIF_nCS[0]/RTP_DATA[15]/N2HET2[7]	N17	Output	Pulldown	20 µA	8mA	EMIF chip select, synchronous
EMIF_nCS[2]/GIOB[4]	L17	Output	Pulldown	20 µA	8mA	EMIF chip select, asynchronous
EMIF_nCS[3]/RTP_DATA[14]/N2HET2[9]	K17	Output	Pulldown	20 µA	8mA	EMIF chip select, asynchronous
EMIF_nCS[4]/RTP_DATA[7]/GIOB[5]	M17	Output	Pulldown	20 µA	8mA	EMIF chip select, asynchronous
ETMDATA[15]/**EMIF_nDQM[0]**	E10	Output	Pulldown	20 µA	8mA	EMIF byte enable
N2HET1[19]/**EMIF_nDQM[0]**/SCI4TX	B13(1)					
ETMDATA[14]/**EMIF_nDQM[1]**	E11	Output	Pulldown	20 µA	8mA	EMIF byte enable
N2HET1[20]/**EMIF_nDQM[1]**/ePWM6B	P2(1)					
N2HET1[21]/**EMIF_nDQM[2]**	H4	Output	Pulldown	20 µA	8mA	EMIF byte enable
N2HET1[22]/**EMIF_nDQM[3]**	B3	Output	Pulldown	20 µA	8mA	EMIF byte enable

Table 4-18. External Memory Interface (EMIF)(2) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
ETMDATA[13]/**EMIF_nOE**	E12	Output	Pulldown	20 µA	8mA	EMIF output enable
N2HET1[17]/**EMIF_nOE**/SCI4RX	A13(1)					
EMIF_nRAS/GIOB[6]	R3	Output	Pulldown	20 µA	8mA	EMIF row address strobe
EMIF_nWAIT/GIOB[7]	P3	Input	Pullup	Fixed, 20 µA	-	EMIF wait
EMIF_nWE/EMIF_RNW	D17	Output	-	-	8mA	EMIF write enable
EMIF_nWE/**EMIF_RNW**	D17	Output	-	-	8mA	EMIF read-not-write
N2HET1[18]/**EMIF_RNW**/ePWM6A	J1(1)					

(1) This is the secondary terminal at which the signal is also available. See Section 4.2.2.2 for more detail on how to select between the available terminals for input functionality.

(2) By default, the EMIF interface pins are the primary pins before configurating the IOMM (IO Muxing Module). The output buffers of these pins are forced to tri-state until enabled by setting PINMMR174[8] = 0 and PINMMR174[9] = 1."

4.2.1.17 Embedded Trace Macrocell Interface for Cortex-R5F (ETM-R5)

Table 4-19. ZWT Embedded Trace Macrocell Interface for Cortex-R5F (ETM-R5)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
ETMDATA[0]	R12	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[1]	R13	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[2]	J15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[3]	H15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[4]	G15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[5]	F15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[6]	E15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[7]	E14	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[8]/EMIF_ADDR[5]	E9	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[9]/EMIF_ADDR[4]	E8	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[10]/EMIF_ADDR[3]	E7	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[11]/EMIF_ADDR[2]	E6	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[12]/EMIF_BA[0]	E13	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[13]/EMIF_nOE	E12	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[14]/EMIF_nDQM[1]	E11	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[15]/EMIF_nDQM[0]	E10	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[16]/EMIF_DATA[0]	K15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[17]/EMIF_DATA[1]	L15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[18]/EMIF_DATA[2]	M15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[19]/EMIF_DATA[3]	N15	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[20]/EMIF_DATA[4]	E5	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[21]/EMIF_DATA[5]	F5	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[22]/EMIF_DATA[6]	G5	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[23]/EMIF_DATA[7]	K5	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[24]/EMIF_DATA[8]/N2HET2[24]/MIBSPI5NCS[4]	L5	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[25]/EMIF_DATA[9]/N2HET2[25]/MIBSPI5NCS[5]	M5	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[26]/EMIF_DATA[10]/N2HET2[26]	N5	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[27]/EMIF_DATA[11]/N2HET2[27]	P5	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[28]/EMIF_DATA[12]/N2HET2[28]/GIOA[0]	R5	Output	Pulldown	20 µA	8mA	ETM data

Table 4-19. ZWT Embedded Trace Macrocell Interface for Cortex-R5F (ETM-R5) (continued)

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
ETMDATA[29]/EMIF_DATA[13]/N2HET2[29]/GIOA[1]	R6	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[30]/EMIF_DATA[14]/N2HET2[30]/GIOA[3]	R7	Output	Pulldown	20 µA	8mA	ETM data
ETMDATA[31]/EMIF_DATA[15]/N2HET2[31]/GIOA[4]	R8	Output	Pulldown	20 µA	8mA	ETM data
ETMTRACECLKIN/EXTCLKIN2/GIOA[5]	R9	Input	Pullup	Fixed, 20 µA	-	ETM trace clock input
ETMTRACECLKOUT/GIOA[6]	R10	Output	Pulldown	20 µA	8mA	ETM trace clock output
ETMTRACECTL/GIOA[7]	R11	Output	Pulldown	20 µA	8mA	ETM trace control

4.2.1.18 System Module Interface

Table 4-20. ZWT System Module Interface

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
nERROR	B14	Output	Pulldown	20 µA	8mA	ESM error (And of Error 1 and Error 2)
GIOB[6]/**nERROR**	J2	Output	Pulldown	20 µA	8mA	ESM error 1
nPORRST	W7	Input	Pulldown	100 µA	-	Power-on reset, cold reset
nRST	B17	I/O	Pullup	100 µA	4mA	System reset, warm reset

4.2.1.19 Clock Inputs and Outputs

Table 4-21. ZWT Clock Inputs and Outputs

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	**337 ZWT**					
ECLK1	A12	I/O	Pulldown	Programmable, 20 µA	2mA ZD/8mA	External clock output, or GIO
EMIF_CLK/**ECLK2**	K3	I/O	Pulldown	Programmable, 20 µA	2mA ZD/8mA	External clock output, or GIO
GIOA[5]/**EXTCLKIN1**/ePWM1A	B5	Input	Pulldown	Fixed, 20 µA	-	External clock input
ETMTRACECLKIN/**EXTCLKIN2**/GIOA[5]	R9	Input	Pullup	Fixed, 20 µA	-	External clock input # 2
KELVIN_GND	L2	Input	-	-	-	Kelvin ground for oscillator
OSCIN	K1	Input	-	-	-	From external crystal/resonator, or external clock input
OSCOUT	L1	Output	-	-	-	To external crystal/resonator

4.2.1.20 Test and Debug Modules Interface

Table 4-22. ZWT Test and Debug Modules Interface

TERMINAL		SIGNAL TYPE	DEFAULT PULL STATE	PULL TYPE	OUTPUT BUFFER DRIVE STRENGTH	DESCRIPTION
SIGNAL NAME	337 ZWT					
nTRST	D18	Input	Pulldown	100 µA	-	JTAG test hardware reset
TCK	B18	Input	Pulldown	Fixed, 100 µA	-	JTAG test clock
TDI	A17	Input	Pullup	Fixed, 100 µA	-	JTAG test data in
TDO	C18	Output	Pulldown	Fixed, 100 µA	8mA	JTAG test data out
TEST	U2	Input	Pulldown	Fixed, 100 µA	-	Test mode enable. This terminal must be connected to ground directly or through a pulldown resistor.
TMS	C19	Input	Pullup	Fixed, 100 µA	-	JTAG test mode select
RTCK	A16	Output	-	-	8mA	JTAG return test clock

4.2.1.21 Flash Supply and Test Pads

Table 4-23. ZWT Flash Supply and Test Pads

TERMINAL		SIGNAL TYPE	DEFAULT PULL STATE	PULL TYPE	OUTPUT BUFFER DRIVE STRENGTH	DESCRIPTION
SIGNAL NAME	337 ZWT					
VCCP	F8	3.3-V Power	–	–	–	Flash pump supply
FLTP1	J5	Input	–	–	–	Flash test pads. These terminals are reserved for TI use only. For proper operation these terminals must connect only to a test pad or not be connected at all [no connect (NC)].
FLTP2	H5	Input	–	–	–	

4.2.1.22 Supply for Core Logic: 1.2-V Nominal

Table 4-24. ZWT Supply for Core Logic: 1.2-V Nominal

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
VCC	P10	1.2-V Power	-	-	-	Core supply
VCC	L6		-	-	-	Core supply
VCC	K6		-	-	-	Core supply
VCC	F9		-	-	-	Core supply
VCC	F10		-	-	-	Core supply
VCC	J14		-	-	-	Core supply
VCC	K14		-	-	-	Core supply
VCC	M10		-	-	-	Core supply
VCC	K8		-	-	-	Core supply
VCC	H10		-	-	-	Core supply
VCC	K12		-	-	-	Core supply

4.2.1.23 Supply for I/O Cells: 3.3-V Nominal

Table 4-25. ZWT Supply for I/O Cells: 3.3-V Nominal

Terminal		Signal Type	Default Pull State	Pull Type	Output Buffer Drive Strength	Description
Signal Name	337 ZWT					
VCCIO	F11	3.3-V Power	–	–	–	Operating supply for I/Os
VCCIO	F12		–	–	–	Operating supply for I/Os
VCCIO	F13		–	–	–	Operating supply for I/Os
VCCIO	F14		–	–	–	Operating supply for I/Os
VCCIO	G14		–	–	–	Operating supply for I/Os
VCCIO	H14		–	–	–	Operating supply for I/Os
VCCIO	L14		–	–	–	Operating supply for I/Os
VCCIO	M14		–	–	–	Operating supply for I/Os
VCCIO	N14		–	–	–	Operating supply for I/Os
VCCIO	P14		–	–	–	Operating supply for I/Os
VCCIO	P13		–	–	–	Operating supply for I/Os
VCCIO	P12		–	–	–	Operating supply for I/Os
VCCIO	P9		–	–	–	Operating supply for I/Os
VCCIO	P8		–	–	–	Operating supply for I/Os
VCCIO	P7		–	–	–	Operating supply for I/Os
VCCIO	P6		–	–	–	Operating supply for I/Os
VCCIO	N6		–	–	–	Operating supply for I/Os
VCCIO	M6		–	–	–	Operating supply for I/Os
VCCIO	J6		–	–	–	Operating supply for I/Os
VCCIO	H6		–	–	–	Operating supply for I/Os
VCCIO	G6		–	–	–	Operating supply for I/Os
VCCIO	F6		–	–	–	Operating supply for I/Os
VCCIO	F7		–	–	–	Operating supply for I/Os

4.2.1.24 Ground Reference for All Supplies Except VCCAD

Table 4-26. ZWT Ground Reference for All Supplies Except VCCAD

TERMINAL		SIGNAL TYPE	DEFAULT PULL STATE	PULL TYPE	OUTPUT BUFFER DRIVE STRENGTH	DESCRIPTION
SIGNAL NAME	337 ZWT					
VSS	W1	Ground	–	–	–	Ground reference
VSS	V1		–	–	–	Ground reference
VSS	W2		–	–	–	Ground reference
VSS	B1		–	–	–	Ground reference
VSS	A1		–	–	–	Ground reference
VSS	A2		–	–	–	Ground reference
VSS	A18		–	–	–	Ground reference
VSS	A19		–	–	–	Ground reference
VSS	B19		–	–	–	Ground reference
VSS	M8		–	–	–	Ground reference
VSS	M9		–	–	–	Ground reference
VSS	M11		–	–	–	Ground reference
VSS	M12		–	–	–	Ground reference
VSS	L8		–	–	–	Ground reference
VSS	L9		–	–	–	Ground reference
VSS	L10		–	–	–	Ground reference
VSS	L11		–	–	–	Ground reference
VSS	L12		–	–	–	Ground reference
VSS	K9		–	–	–	Ground reference
VSS	K10		–	–	–	Ground reference
VSS	K11		–	–	–	Ground reference
VSS	J8		–	–	–	Ground reference
VSS	J9		–	–	–	Ground reference
VSS	J10		–	–	–	Ground reference
VSS	J11		–	–	–	Ground reference
VSS	J12		–	–	–	Ground reference
VSS	H8		–	–	–	Ground reference
VSS	H9		–	–	–	Ground reference
VSS	H11		–	–	–	Ground reference
VSS	H12		–	–	–	Ground reference

4.2.1.25 Other Supplies

Table 4-27. Other Supplies

TERMINAL		SIGNAL TYPE	DEFAULT PULL STATE	PULL TYPE	OUTPUT BUFFER DRIVE STRENGTH	DESCRIPTION
SIGNAL NAME	337 ZWT					
Supply for PLL: 1.2-V nominal						
VCCPLL	P11	1.2-V Power	–	–	–	Core supply for PLL's

4.2.2 Multiplexing

This microcontroller has several interfaces and uses extensive multiplexing to bring out the functions as required by the target application. The multiplexing is mostly on the output signals. A few inputs are also multiplexed to allow the same input signal to be driven in from an alternative terminal. For more information on multiplexing, refer to the IOMM chapter of the device specific technical reference manual.

4.2.2.1 Output Multiplexing

Table 4-28. Output Multiplexing

Address Offset	337 ZWT BALL	DEFAULT FUNCTION	Select Bit	Alternate Function 1	Select Bit	Alternate Function 2	Select Bit	Alternate Function 3	Select Bit	Alternate Function 4	Select Bit	Alternate Function 5	Select Bit
0x110	N19	AD1EVT	0[0]			MII_RX_ER	0[2]	RMII_RX_ER	0[3]			nTZ1_1	0[5]
	D4	EMIF_ADDR[0]	0[8]			N2HET2[1]	0[10]						
	D5	EMIF_ADDR[1]	0[16]			N2HET2[3]	0[18]						
	C4	EMIF_ADDR[6]	0[24]	RTP_DATA[13]	0[25]	N2HET2[11]	0[26]						
0x114	C5	EMIF_ADDR[7]	1[0]	RTP_DATA[12]	1[1]	N2HET2[13]	1[2]						
	C6	EMIF_ADDR[8]	1[8]	RTP_DATA[11]	1[9]	N2HET2[15]	1[10]						
	C7	EMIF_ADDR[9]	1[16]	RTP_DATA[10]	1[17]								
	C8	EMIF_ADDR[10]	1[24]	RTP_DATA[9]	1[25]								
0x118	C9	EMIF_ADDR[11]	2[0]	RTP_DATA[8]	2[1]								
	C10	EMIF_ADDR[12]	2[8]	RTP_DATA[6]	2[9]								
	C11	EMIF_ADDR[13]	2[16]	RTP_DATA[5]	2[17]								
	C12	EMIF_ADDR[14]	2[24]	RTP_DATA[4]	2[25]								
0x11C	C13	EMIF_ADDR[15]	3[0]	RTP_DATA[3]	3[1]								
	D14	EMIF_ADDR[16]	3[8]	RTP_DATA[2]	3[9]								
	C14	EMIF_ADDR[17]	3[16]	RTP_DATA[1]	3[17]								
	D15	EMIF_ADDR[18]	3[24]	RTP_DATA[0]	3[25]								
0x120	C15	EMIF_ADDR[19]	4[0]	RTP_nENA	4[1]								
	C16	EMIF_ADDR[20]	4[8]	RTP_nSYNC	4[9]								
	C17	EMIF_ADDR[21]	4[16]	RTP_CLK	4[17]								
0x124 - 0x12C	Reserved												
0x130	PINMMR8[23:0] are reserved												
	D16	EMIF_BA[1]	8[24]		8[25]	N2HET2[5]	8[26]						
0x134	K3	RESERVED	9[0]	EMIF_CLK	9[1]	ECLK2	9[2]						
	R4	EMIF_nCAS	9[8]			GIOB[3]	9[10]						
	N17	EMIF_nCS[0]	9[16]	RTP_DATA[15]	9[17]	N2HET2[7]	9[18]						
	L17	EMIF_nCS[2]	9[24]			GIOB[4]	9[26]						

Table 4-28. Output Multiplexing (continued)

Address Offset	337 ZWT BALL	DEFAULT FUNCTION	Select Bit	Alternate Function 1	Select Bit	Alternate Function 2	Select Bit	Alternate Function 3	Select Bit	Alternate Function 4	Select Bit	Alternate Function 5	Select Bit
0x138	K17	EMIF_nCS[3]	10[0]	RTP_DATA[14]	10[1]	N2HET2[9]	10[2]						
	M17	EMIF_nCSl[4]	10[8]	RTP_DATA[7]	10[9]	GIOB[5]	10[10]						
	R3	EMIF_nRAS	10[16]			GIOB[6]	10[18]						
	P3	EMIF_nWAIT	10[24]			GIOB[7]	10[26]						
0x13C	D17	EMIF_nWE	11[0]	EMIF_RNW	11[1]								
	E9	ETMDATA[8]	11[8]	EMIF_ADDR[5]	11[9]								
	E8	ETMDATA[9]	11[16]	EMIF_ADDR[4]	11[17]								
	E7	ETMDATA[10]	11[24]	EMIF_ADDR[3]	11[25]								
0x140	E6	ETMDATA[11]	12[0]	EMIF_ADDR[2]	12[1]								
	E13	ETMDATA[12]	12[8]	EMIF_BA[0]	12[9]								
	E12	ETMDATA[13]	12[16]	EMIF_nOE	12[17]								
	E11	ETMDATA[14]	12[24]	EMIF_nDQM[1]	12[25]								
0x144	E10	ETMDATA[15]	13[0]	EMIF_nDQM[0]	13[1]								
	K15	ETMDATA[16]	13[8]	EMIF_DATA[0]	13[9]								
	L15	ETMDATA[17]	13[16]	EMIF_DATA[1]	13[17]								
	M15	ETMDATA[18]	13[24]	EMIF_DATA[2]	13[25]								
0x148	N15	ETMDATA[19]	14[0]	EMIF_DATA[3]	14[1]								
	E5	ETMDATA[20]	14[8]	EMIF_DATA[4]	14[9]								
	F5	ETMDATA[21]	14[16]	EMIF_DATA[5]	14[17]								
	G5	ETMDATA[22]	14[24]	EMIF_DATA[6]	14[25]								
0x14C	K5	ETMDATA[23]	15[0]	EMIF_DATA[7]	15[1]								
	L5	ETMDATA[24]	15[8]	EMIF_DATA[8]	15[9]	N2HET2[24]	15[10]	MIBSPI5NCS[4]	15[11]				
	M5	ETMDATA[25]	15[16]	EMIF_DATA[9]	15[17]	N2HET2[25]	15[18]	MIBSPI5NCS[5]	15[19]				
	N5	ETMDATA[26]	15[24]	EMIF_DATA[10]	15[25]	N2HET2[26]	15[26]						
0x150	P5	ETMDATA[27]	16[0]	EMIF_DATA[11]	16[1]	N2HET2[27]	16[2]						
	R5	ETMDATA[28]	16[8]	EMIF_DATA[12]	16[9]	N2HET2[28]	16[10]	GIOA[0]	16[11]				
	R6	ETMDATA[29]	16[16]	EMIF_DATA[13]	16[17]	N2HET2[29]	16[18]	GIOA[1]	16[19]				
	R7	ETMDATA[30]	16[24]	EMIF_DATA[14]	16[25]	N2HET2[30]	16[26]	GIOA[3]	16[27]				
0x154	R8	ETMDATA[31]	17[0]	EMIF_DATA[15]	17[1]	N2HET2[31]	17[2]	GIOA[4]	17[3]				
	R9	ETMTRACECLKIN	17[8]	EXTCLKIN2	17[9]			GIOA[5]	17[11]				
	R10	ETMTRACECLKOUT	17[16]					GIOA[6]	17[19]				
	R11	ETMTRACECTL	17[24]					GIOA[7]	17[27]				

Table 4-28. Output Multiplexing (continued)

Address Offset	337 ZWT BALL	DEFAULT FUNCTION	Select Bit	Alternate Function 1	Select Bit	Alternate Function 2	Select Bit	Alternate Function 3	Select Bit	Alternate Function 4	Select Bit	Alternate Function 5	Select Bit
0x158	B15	FRAYTX1	18[0]					GIOA[2]	18[3]				
	B8	FRAYTX2	18[8]					GIOB[0]	18[11]				
	B16	FRAYTXEN1	18[16]					GIOB[1]	18[19]				
	B9	FRAYTXEN2	18[24]					GIOB[2]	18[27]				
0x15C	C1	GIOA[2]	19[0]			N2HET2[0]	19[2]					eQEP2I	19[5]
	E1	GIOA[3]	19[8]			N2HET2[2]	19[10]						
	B5	GIOA[5]	19[16]					EXTCLKIN1	19[19]			ePWM1A	19[21]
	H3	GIOA[6]	19[24]			N2HET2[4]	19[26]					ePWM1B	19[29]
0x160	M1	GIOA[7]	20[0]			N2HET2[6]	20[2]					ePWM2A	20[5]
	F2	GIOB[2]	20[8]					DCAN4TX	20[11]				
	W10	GIOB[3]	20[16]					DCAN4RX	20[19]				
	J2	GIOB[6]	20[24]	nERROR	20[25]								
0x164	F1	GIOB[7]	21[0]	RESERVED	21[1]							nTZ1_2	21[5]
	R2	MIBSPI1NCS[0]	21[8]	MIBSPI1SOMI[1]	21[9]	MII_TXD[2]	21[10]					ECAP6	21[13]
	F3	MIBSPI1NCS[1]	21[16]			MII_COL	21[18]	N2HET1[17]	21[19]			eQEP1S	21[21]
	G3	MIBSPI1NCS[2]	21[24]			MDIO	21[26]	N2HET1[19]	21[27]				
0x168	J3	MIBSPI1NCS[3]	22[0]					N2HET1[21]	22[3]			nTZ1_3	22[5]
	G19	MIBSPI1NENA	22[8]			MII_RXD[2]	22[10]	N2HET1[23]	22[11]			ECAP4	22[13]
	V9	MIBSPI3CLK	22[16]	AD1EXT_SEL[1]	22[17]							eQEP1A	22[21]
	V10	MIBSPI3NCS[0]	22[24]	AD2EVT	22[25]							eQEP1I	22[29]
0x16C	V5	MIBSPI3NCS[1]	23[0]			MDCLK	23[2]	N2HET1[25]	23[3]				
	B2	MIBSPI3NCS[2]	23[8]	I2C1_SDA	23[9]			N2HET1[27]	23[11]			nTZ1_2	23[13]
	C3	MIBSPI3NCS[3]	23[16]	I2C1_SCL	23[17]			N2HET1[29]	23[19]			nTZ1_1	23[21]
	W9	MIBSPI3NENA	23[24]	MIBSPI3NCS[5]	23[25]			N2HET1[31]	23[27]			eQEP1B	23[29]
0x170	W8	MIBSPI3SIMO	24[0]	AD1EXT_SEL[0]	24[1]							ECAP3	24[5]
	V8	MIBSPI3SOMI	24[8]	AD1EXT_ENA	24[9]							ECAP2	24[13]
	H19	MIBSPI5CLK	24[16]	DMM_DATA[4]	24[17]	MII_TXEN	24[18]	RMII_TXEN	24[19]				
	E19	MIBSPI5NCS[0]	24[24]	DMM_DATA[5]	24[25]							ePWM4A	24[29]
0x174	B6	MIBSPI5NCS[1]	25[0]	DMM_DATA[6]	25[1]								
	W6	MIBSPI5NCS[2]	25[8]	DMM_DATA[2]	25[9]								
	T12	MIBSPI5NCS[3]	25[16]	DMM_DATA[3]	25[17]								
	H18	MIBSPI5NENA	25[24]	DMM_DATA[7]	25[25]	MII_RXD[3]	25[26]					ECAP5	25[29]

Table 4-28. Output Multiplexing (continued)

Address Offset	337 ZWT BALL	DEFAULT FUNCTION	Select Bit	Alternate Function 1	Select Bit	Alternate Function 2	Select Bit	Alternate Function 3	Select Bit	Alternate Function 4	Select Bit	Alternate Function 5	Select Bit
0x178	J19	MIBSPI5SIMO[0]	26[0]	DMM_DATA[8]	26[1]	MII_TXD[1]	26[2]	RMII_TXD[1]	26[3]				
	E16	MIBSPI5SIMO[1]	26[8]	DMM_DATA[9]	26[9]					AD1EXT_SEL[0]	26[12]		
	H17	MIBSPI5SIMO[2]	26[16]	DMM_DATA[10]	26[17]					AD1EXT_SEL[1]	26[20]		
	G17	MIBSPI5SIMO[3]	26[24]	DMM_DATA[11]	26[25]	I2C2_SDA	26[26]			AD1EXT_SEL[2]	26[28]		
0x17C	J18	MIBSPI5SOMI[0]	27[0]	DMM_DATA[12]	27[1]	MII_TXD[0]	27[2]	RMII_TXD[0]	27[3]				
	E17	MIBSPI5SOMI[1]	27[8]	DMM_DATA[13]	27[9]					AD1EXT_SEL[3]	27[12]		
	H16	MIBSPI5SOMI[2]	27[16]	DMM_DATA[14]	27[17]					AD1EXT_SEL[4]	27[20]		
	G16	MIBSPI5SOMI[3]	27[24]	DMM_DATA[15]	27[25]	I2C2_SCL	27[26]			AD1EXT_ENA	27[28]		
0x180	K18	N2HET1[0]	28[0]	MIBSPI4CLK	28[1]							ePWM2B	28[5]
	V2	N2HET1[1]	28[8]	MIBSPI4NENA	28[9]			N2HET2[8]	28[11]			eQEP2A	28[13]
	W5	N2HET1[2]	28[16]	MIBSPI4SIMO	28[17]							ePWM3A	28[21]
	U1	N2HET1[3]	28[24]	MIBSPI4NCS[0]	28[25]			N2HET2[10]	28[27]			eQEP2B	28[29]
0x184	B12	N2HET1[4]	29[0]	MIBSPI4NCS[1]	29[1]							ePWM4B	29[5]
	V6	N2HET1[5]	29[8]	MIBSPI4SOMI	29[9]			N2HET2[12]	29[11]			ePWM3B	29[13]
	W3	N2HET1[6]	29[16]	SCI3RX	29[17]							ePWM5A	29[21]
	T1	N2HET1[7]	29[24]	MIBSPI4NCS[2]	29[25]			N2HET2[14]	29[27]			ePWM7B	29[29]
0x188	E18	N2HET1[8]	30[0]	MIBSPI1SIMO[1]	30[1]	MII_TXD[3]	30[2]						
	V7	N2HET1[9]	30[8]	MIBSPI4NCS[3]	30[9]			N2HET2[16]	30[11]			ePWM7A	30[13]
	D19	N2HET1[10]	30[16]	MIBSPI4NCS[4]	30[17]	MII_TX_CLK	30[18]	RESERVED	30[19]			nTZ1_3	30[21]
	E3	N2HET1[11]	30[24]	MIBSPI3NCS[4]	30[25]			N2HET2[18]	30[27]			ePWM1SYNCO	30[29]
0x18C	B4	N2HET1[12]	31[0]	MIBSPI4NCS[5]	31[1]	MII_CRS	31[2]	RMII_CRS_DV	31[3]				
	N2	N2HET1[13]	31[8]	SCI3TX	31[9]			N2HET2[20]	31[11]			ePWM5B	31[13]
	N1	N2HET1[15]	31[16]	MIBSPI1NCS[4]	31[17]			N2HET2[22]	31[19]			ECAP1	31[21]
	A4	N2HET1[16]	31[24]					ePWM1SYNCI	31[27]			ePWM1SYNCO	31[29]
0x190	A13	N2HET1[17]	32[0]	EMIF_nOE	32[1]	SCI4RX	32[2]						
	J1	N2HET1[18]	32[8]	EMIF_RNW	32[9]							ePWM6A	32[13]
	B13	N2HET1[19]	32[16]	EMIF_nDQM[0]	32[17]	SCI4TX	32[18]						
	P2	N2HET1[20]	32[24]	EMIF_nDQM[1]	32[25]							ePWM6B	32[29]
0x194	H4	N2HET1[21]	33[0]	EMIF_nDQM[2]	33[1]								
	B3	N2HET1[22]	33[8]	EMIF_nDQM[3]	33[9]								
	J4	N2HET1[23]	33[16]	EMIF_BA[0]	33[17]								
	P1	N2HET1[24]	33[24]	MIBSPI1NCS[5]	33[25]	MII_RXD[0]	33[26]	RMII_RXD[0]	33[27]				

Table 4-28. Output Multiplexing (continued)

Address Offset	337 ZWT BALL	DEFAULT FUNCTION	Select Bit	Alternate Function 1	Select Bit	Alternate Function 2	Select Bit	Alternate Function 3	Select Bit	Alternate Function 4	Select Bit	Alternate Function 5	Select Bit
0x198	A14	N2HET1[26]	34[0]			MII_RXD[1]	34[2]	RMII_RXD[1]	34[3]				
	K19	N2HET1[28]	34[8]			MII_RXCLK	34[10]	RMII_REFCLK	34[11]	RESERVED	34[12]		
	B11	N2HET1[30]	34[16]			MII_RX_DV	34[18]					eQEP2S	34[21]
	D8	N2HET2[1]	34[24]	N2HET1_NDIS	34[25]								
0x19C	D7	N2HET2[2]	35[0]	N2HET2_NDIS	35[1]								
	D3	N2HET2[12]	35[8]							MIBSPI2NENA	35[12]	MIBSPI2NCS[1]	35[13]
	D2	N2HET2[13]	35[16]							MIBSPI2SOMI	35[20]		
	D1	N2HET2[14]	35[24]							MIBSPI2SIMO	35[28]		
0x1A0	P4	N2HET2[19]	36[0]	LIN2RX	36[1]								
	T5	N2HET2[20]	36[8]	LIN2TX	36[9]								
	T4	MII_RXCLK	36[16]							RESERVED	36[20]		
	U7	MII_TX_CLK	36[24]							RESERVED	36[28]		
0x1A4	E2	N2HET2[3]	37[0]							MIBSPI2CLK	37[4]		
	N3	N2HET2[7]	37[8]							MIBSPI2NCS[0]	37[12]		

4.2.2.1.1 Notes on Output Multiplexing

Table 4-28 lists the output signal multiplexing and control signals for selecting the desired functionality for each pad.

- The pads default to the signal defined by the "Default Function" in Table 4-28.
- The CTRL x columns in Table 4-28 contain a value of type x[y] which indicates the control register PINMMRx, bit y. It indicates the multiplexing control register and the bit that must be set in order to select the corresponding functionality to be output on any particular pad.
  – For example, consider the multiplexing on pin H3 for the 337-ZWT package:

337 ZWT BALL	DEFAULT FUNCTION	CTRL1	OPTION 2	CTRL2	OPTION 3	CTRL3	OPTION 4	CTRL4	OPTION 5	CTRL5	OPTION 6	CTRL6
H3	GIOA[6]	19[24]			N2HET2[4]	19[26]					ePWM1B	19[29]

  – When GIOA[6] is configured as an output pin in the GIO module control register, then the programmed output level appears on pin H3 by default. The PINMMR19[24] is set by default to indicate that the GIOA[6] signal is selected to be output.
  – If the application must output the N2HET2[4] signal on pin H3, it must clear PINMMR19[24] and set PINMMR19[26].
  – Note that the pin is connected as input to both the GIO and N2HET2 modules. That is, there is no input multiplexing on this pin.
- The base address of the IOMM module starts at 0xFFFF_1C00. The Output mux control registers with the first register PINMMR0 starts at the offset address 0x110 within the IOMM module.

4.2.2.2 Input Multiplexing

Some signals are connected to more than one terminals, so that the inputs for these signals can come from either of these terminals. A multiplexor is implemented to let the application choose the terminal that will be used for providing the input signal from among the available options. The input path selection is done based on two bits in the PINMMR control registers as listed in Table 4-29.

Table 4-29. Input Multiplexing

Address Offset	Signal Name	Default Terminal	Terminal 1 Input Multiplex Control	Alternate Terminal	Terminal 2 Input Multiplex Control
250h	AD2EVT	T10	PINMMR80[0]	V10	PINMMR80[1]
25Ch	GIOA[0]	A5	PINMMR83[24]	R5	PINMMR83[25]
260h	GIOA[1]	C2	PINMMR84[0]	R6	PINMMR84[1]
	GIOA[2]	C1	PINMMR84[8]	B15	PINMMR84[9]
	GIOA[3]	E1	PINMMR84[16]	R7	PINMMR84[17]
	GIOA[4]	A6	PINMMR84[24]	R8	PINMMR84[25]
264h	GIOA[5]	B5	PINMMR85[0]	R9	PINMMR85[1]
	GIOA[6]	H3	PINMMR85[8]	R10	PINMMR85[9]
	GIOA[7]	M1	PINMMR85[16]	R11	PINMMR85[17]
	GIOB[0]	M2	PINMMR85[24]	B8	PINMMR85[25]
268h	GIOB[1]	K2	PINMMR86[0]	B16	PINMMR86[1]
	GIOB[2]	F2	PINMMR86[8]	B9	PINMMR86[9]
	GIOB[3]	W10	PINMMR86[16]	R4	PINMMR86[17]
	GIOB[4]	G1	PINMMR86[24]	L17	PINMMR86[25]
26Ch	GIOB[5]	G2	PINMMR87[0]	M17	PINMMR87[1]
	GIOB[6]	J2	PINMMR87[8]	R3	PINMMR87[9]
	GIOB[7]	F1	PINMMR87[16]	P3	PINMMR87[17]
	MDIO	F4	PINMMR87[24]	G3	PINMMR87[25]
270h	MIBSPI1NCS[4]	U10	PINMMR88[0]	N1	PINMMR88[1]
	MIBSPI1NCS[5]	U9	PINMMR88[8]	P1	PINMMR88[9]
274h	MII_COL	W4	PINMMR89[16]	F3	PINMMR89[17]
	MII_CRS	V4	PINMMR89[24]	B4	PINMMR89[25]
278h	MII_RX_DV	U6	PINMMR90[0]	B11	PINMMR90[1]
	MII_RX_ER	U5	PINMMR90[8]	N19	PINMMR90[9]
	MII_RXCLK	T4	PINMMR90[16]	K19	PINMMR90[17]
	MII_RXD[0]	U4	PINMMR90[24]	P1	PINMMR90[25]
27Ch	MII_RXD[1]	T3	PINMMR91[0]	A14	PINMMR91[1]
	MII_RXD[2]	U3	PINMMR91[8]	G19	PINMMR91[9]
	MII_RXD[3]	V3	PINMMR91[16]	H18	PINMMR91[17]
	MII_TX_CLK	U7	PINMMR91[24]	D19	PINMMR91[25]
280h	N2HET1[17]	A13	PINMMR92[0]	F3	PINMMR92[1]
	N2HET1[19]	B13	PINMMR92[8]	G3	PINMMR92[9]
	N2HET1[21]	H4	PINMMR92[16]	J3	PINMMR92[17]
	N2HET1[23]	J4	PINMMR92[24]	G19	PINMMR92[25]
284h	N2HET1[25]	M3	PINMMR93[0]	V5	PINMMR93[1]
	N2HET1[27]	A9	PINMMR93[8]	B2	PINMMR93[9]
	N2HET1[29]	A3	PINMMR93[16]	C3	PINMMR93[17]
	N2HET1[31]	J17	PINMMR93[24]	W9	PINMMR93[25]

Table 4-29. Input Multiplexing (continued)

Address Offset	Signal Name	Default Terminal	Terminal 1 Input Multiplex Control	Alternate Terminal	Terminal 2 Input Multiplex Control
288h	N2HET2[0]	D6	PINMMR94[0]	C1	PINMMR94[1]
	N2HET2[1]	D8	PINMMR94[8]	D4	PINMMR94[9]
	N2HET2[2]	D7	PINMMR94[16]	E1	PINMMR94[17]
	N2HET2[3]	E2	PINMMR94[24]	D5	PINMMR94[25]
28Ch	N2HET2[4]	D13	PINMMR95[0]	H3	PINMMR95[1]
	N2HET2[5]	D12	PINMMR95[8]	D16	PINMMR95[9]
	N2HET2[6]	D11	PINMMR95[16]	M1	PINMMR95[17]
	N2HET2[7]	N3	PINMMR95[24]	N17	PINMMR95[25]
290h	N2HET2[8]	K16	PINMMR96[0]	V2	PINMMR96[1]
	N2HET2[9]	L16	PINMMR96[8]	K17	PINMMR96[9]
	N2HET2[10]	M16	PINMMR96[16]	U1	PINMMR96[17]
	N2HET2[11]	N16	PINMMR96[24]	C4	PINMMR96[25]
294h	N2HET2[12]	D3	PINMMR97[0]	V6	PINMMR97[1]
	N2HET2[13]	D2	PINMMR97[8]	C5	PINMMR97[9]
	N2HET2[14]	D1	PINMMR97[16]	T1	PINMMR97[17]
	N2HET2[15]	K4	PINMMR97[24]	C6	PINMMR97[25]
298h	N2HET2[16]	L4	PINMMR98[0]	V7	PINMMR98[1]
	N2HET2[18]	N4	PINMMR98[8]	E3	PINMMR98[9]
	N2HET2[20]	T5	PINMMR98[16]	N2	PINMMR98[17]
	N2HET2[22]	T7	PINMMR98[24]	N1	PINMMR98[25]
29Ch	nTZ1_1	N19	PINMMR99[0]	C3	PINMMR99[1]
	nTZ1_2	F1	PINMMR99[8]	B2	PINMMR99[9]
	nTZ1_3	J3	PINMMR99[16]	D19	PINMMR99[17]

4.2.2.2.1 Notes on Input Multiplexing

- The Terminal x Input Multiplex Control column in Table 4-29 lists the multiplexing control register and the bit that must be set in order to select the terminal for providing the input signal to the system. For example, N2HET2[22] can appears on two different terminals at terminal number T7 and N1. By default PINMMR98[24] is set and PINMMR98[25] is cleared to select T7 for providing N2HET2[22] to the system. If the application chooses to use N1 for providing N2HET2[22] then PINMMR98[24] must be cleared and PINMMR98[25] must be set.
- Base address of the IOMM module starts at 0xFFFF_1C00. Input mux control registers with the first register PINMMR80 starts at the offset address 0x250 within the IOMM module.

4.2.2.2.2 General Rules for Multiplexing Control Registers

- The PINMMR control registers can only be written in privileged mode. A write in a nonprivileged mode will generate an error response.
- If the application writes all 9's to any PINMMR control register, then the default functions are selected for the affected pads.
- Each byte in a PINMMR control register is used to select the functionality for a given pad. If the application sets more than one bit within a byte for any pad, then the default function is selected for this pad.
- Several bits in the PINMMR control registers are reserved and are not used to enable any functions. If the application sets only these bits and clears the other bits, then the default functions are selected for the affected pads.

5 Specifications

5.1 Absolute Maximum Ratings[(1)]

Over Operating Free-Air Temperature Range

		MIN	MAX	UNIT
Supply voltage	V_{CC} [(2)]	–0.3	1.43	V
	V_{CCIO}, V_{CCP} [(2)]	–0.3	4.6	
	V_{CCAD}	–0.3	6.25	
Input voltage	All input pins, with exception of ADC pins	–0.3	4.6	V
	ADC input pins	–0.3	6.25	
Input clamp current:	I_{IK} (V_I < 0 or V_I > V_{CCIO}) All pins, except AD1IN[31:0] and AD2IN[24:0]	–20	20	mA
	I_{IK} (V_I < 0 or V_I > V_{CCAD}) AD1IN[31:0] and AD2IN[24:0]	–10	10	
	Total	–40	40	
Operating free-air temperature (T_A)		–40	125	°C
Operating junction temperature (T_J)		–40	150	°C
Storage temperature (T_{stg})		–65	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 *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) Maximum-rated conditions for extended periods may affect device reliability. All voltage values are with respect to their associated grounds.

5.2 ESD Ratings

				MIN	MAX	UNIT
V_{ESD}	Electrostatic discharge (ESD) performance:	Human Body Model (HBM), per AEC Q100-002D[(1)]		–2	2	kV
		Charged Device Model (CDM), per AEC Q100-011	All pins except corner balls	–500	500	V
			Corner balls	–750	750	V

(1) AEC Q100-002D indicates HBM stressing is done in accordance with the ANSI/ESDA/JEDEC JS-001-2011 specification.

5.3 Power-On Hours (POH)

POH is a function of voltage and temperature. Usage at higher voltages and temperatures will result in a reduction in POH to achieve the same reliability performance. The POH information in Table 5-1 is provided solely for convenience and does not extend or modify the warranty provided under TI's standard terms and conditions for TI Semiconductor Products. To avoid significant device degradation, the device POH must be limited to those listed in Table 5-1. To convert to equivalent POH for a specific temperature profile, see the *Calculating Equivalent Power-on-Hours for Hercules Safety MCUs Application Report* (SPNA207).

Table 5-1. Power-On Hours Limits

NOMINAL V_{CC} VOLTAGE (V)	JUNCTION TEMPERATURE (T_J)	LIFETIME POH[(1)]
1.2 V	105 ºC	100K

(1) POH represent device operation under the specified nominal conditions continuously for the duration of the calculated lifetime.

5.4 Device Recommended Operating Conditions[1]

		MIN	NOM	MAX	UNIT
V_{CC}	Digital logic supply voltage (Core)	1.14	1.2	1.32	V
V_{CCPLL}	PLL supply voltage	1.14	1.2	1.32	V
V_{CCIO}	Digital logic supply voltage (I/O)	3	3.3	3.6	V
V_{CCAD}	MibADC supply voltage	3		5.25	V
V_{CCP}	Flash pump supply voltage	3	3.3	3.6	V
V_{SS}	Digital logic supply ground		0		V
V_{SSAD}	MibADC supply ground	–0.1		0.1	V
$V_{ADREFHI}$	Analog-to-Digital (A-to-D) high-voltage reference source	V_{SSAD}		V_{CCAD}	V
$V_{ADREFLO}$	A-to-D low-voltage reference source	V_{SSAD}		V_{CCAD}	V
T_A	Operating free-air temperature	–40		125	°C
T_J	Operating junction temperature	–40		150	°C

(1) All voltages are with respect to V_{SS}, except V_{CCAD}, which is with respect to V_{SSAD}.

5.5 Switching Characteristics over Recommended Operating Conditions for Clock Domains

Table 5-2. Clock Domain Timing Specifications

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
f_{OSC}	OSC - oscillator clock frequency using an external crystal		5	20	MHz
f_{GCLK1}	GCLK - R5F CPU clock frequency			300	MHz
f_{GCLK2}	GCLK - R5F CPU clock frequency			300	MHz
f_{HCLK}	HCLK - System clock frequency			150	MHz
f_{VCLK}	VCLK - Primary peripheral clock frequency			110	MHz
f_{VCLK2}	VCLK2 - Secondary peripheral clock frequency			110	MHz
f_{VCLK3}	VCLK3 - Secondary peripheral clock frequency			150	MHz
f_{VCLKA1}	VCLKA1 - Primary asynchronous peripheral clock frequency			110	MHz
f_{VCLKA2}	VCLKA2 - Secondary asynchronous peripheral clock frequency			110	MHz
f_{VCLKA4}	VCLKA4 - Secondary asynchronous peripheral clock frequency			110	MHz
$f_{RTICLK1}$	RTICLK1 - clock frequency			f_{VCLK}	MHz
$f_{PROG/ERASE}$	System clock frequency - flash programming/erase			f_{HCLK}	MHz
f_{ECLK1}	External Clock 1			110	MHz
f_{ECLK2}	External Clock 2			110	MHz
$f_{ETMCLKOUT}$	ETM trace clock output			55	MHz
$f_{ETMCLKIN}$	ETM trace clock input			110	MHz
$f_{EXTCLKIN1}$	External input clock 1			110	MHz
$f_{EXTCLKIN2}$	External input clock 2			110	MHz

Table 5-2 lists the maximum frequency of the CPU (GLKx), the level-2 memory (HCLK) and the peripheral clocks (VCLKx). It is not always possible to run each clock at its maximum frequency as GCLK must be an integral multiple of HCLK and HCLK must be an integral multiple of VCLKx. Depending on the system, the optimum performance may be obtained by maximizing either the CPU frequency, the level-two RAM interface, the level-two flash interface, or the peripherals.

5.6 Wait States Required - L2 Memories

Wait states are cycles the CPU must wait in order to retrieve data from the memories which have access times longer than a CPU clock. Memory wrapper, SCR interconnect and the CPU itself may introduce additional cycles of latency due to logic pipelining and synchronization. Therefore, the total latency cycles as seen by the CPU can be more than the number of wait states to cover the memory access time.

Figure 5-1 shows only the number of programmable wait states needed for L2 flash memory at different frequencies. The number of wait states is correlated to HCLK frequency. The clock ratio between CPU clock (GCLKx) and HCLK can vary. Therefore, the total number of wait states in terms of GCLKx can be obtained by taking the programmed wait states multiplied by the clock ratio.

There is no user programmable wait state for L2 SRAM access. L2 SRAM is clocked by HCLK and is limited to maximum 150 MHz.

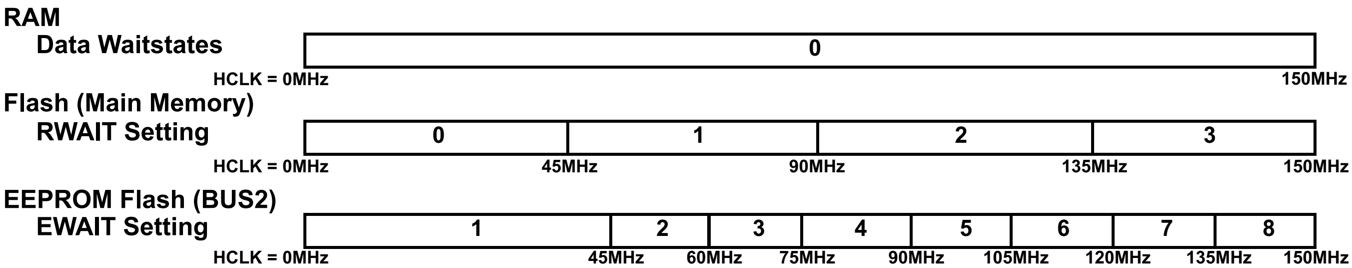


Figure 5-1. Wait States Scheme

L2 flash is clocked by HCLK and is limited to maximum 150 MHz. The L2 flash can support zero data wait state up to 45 MHz.

5.7 Power Consumption Summary

Over Recommended Operating Conditions

PARAMETER		TEST CONDITIONS	MIN	TYP(1)	MAX	UNIT
I_{CC}	V_{CC} digital supply and PLL current (operating mode)	f_{GCLK} = 300 MHz, f_{HCLK} = 150 MHz, f_{VCLK} = 75 MHz, f_{VCLK2} = 75 MHz, f_{VCLK3} = 150 MHz		510	990 (2)	mA
	V_{CC} digital supply and PLL current (LBIST mode, or PBIST mode)	LBIST clock rate = 75 MHz PBIST ROM clock frequency = 75 MHz		880	1375	(3)(4) mA
I_{CCIO}	V_{CCIO} digital supply current (operating mode)	No DC load, V_{CCmax}			15	mA
I_{CCAD}	V_{CCAD} supply current (operating mode)	Single ADC operational, $V_{CCADmax}$			15	mA
		Single ADC power down, $V_{CCADmax}$			5	µA
		Both ADCs operational, $V_{CCADmax}$			30	mA
$I_{CCREFHI}$	AD_{REFHI} supply current (operating mode)	Single ADC operational, $AD_{REFHImax}$			5	mA
		Both ADCs operational, $AD_{REFHImax}$			10	mA
I_{CCP}	V_{CCP} pump supply current	Read operation of two banks in parallel, V_{CCPmax}			70	mA
		Read from two banks and program or erase another bank, V_{CCPmax}			93	mA

(1) The typical value is the average current for the nominal process corner and junction temperature of 25ºC.
(2) The maximum $I_{CC,}$ value can be derated
- linearly with voltage
- by 1.8 mA/MHz for lower GCLK frequency when f_{GCLK}= 2 * f_{HCLK}= 4 * f_{VCLK}
- for lower junction temperature by the equation below where T_{JK} is the junction temperature in Kelvin and the result is in milliamperes.
  $405 - 0.2\, e^{0.018\, T_{JK}}$

(3) The maximum $I_{CC,}$ value can be derated
- linearly with voltage
- by 3.2 mA/MHz for lower GCLK frequency
- for lower junction temperature by the equation below where T_{JK} is the junction temperature in Kelvin and the result is in milliamperes.
  $405 - 0.2\, e^{0.018\, T_{JK}}$

(4) LBIST and PBIST currents are for a short duration, typically less than 10 ms. They are usually ignored for thermal calculations for the device and the voltage regulator.

5.8 Input/Output Electrical Characteristics Over Recommended Operating Conditions[1]

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{hys}	Input hysteresis	All inputs (except FRAYRX1, FRAYRX2)		180			mV
		FRAYRX1, FRAYRX2		100			mV
V_{IL}	Low-level input voltage	All inputs[2] (except FRAYRX1, FRAYRX2)		–0.3		0.8	V
		FRAYRX1, FRAYRX2				0.4 * V_{CCIO}	V
V_{IH}	High-level input voltage	All inputs[2] (except FRAYRX1, FRAYRX2)		2		V_{CCIO} + 0.3	V
		FRAYRX1, FRAYRX2		0.6 * V_{CCIO}			V
V_{OL}	Low-level output voltage		$I_{OL} = I_{OLmax}$			0.2 * V_{CCIO}	V
			I_{OL} = 50 µA, standard output mode			0.2	
V_{OH}	High-level output voltage		$I_{OH} = I_{OHmax}$	0.8 * V_{CCIO}			V
			I_{OH} = 50 µA, standard output mode	V_{CCIO} – 0.3			
I_{IC}	Input clamp current (I/O pins)		$V_I < V_{SSIO}$ – 0.3 or $V_I > V_{CCIO}$ + 0.3	–3.5		3.5	mA
I_I	Input current (I/O pins)	I_{IH} Pulldown 20 µA	$V_I = V_{CCIO}$	5		40	µA
		I_{IH} Pulldown 100 µA	$V_I = V_{CCIO}$	40		195	
		I_{IL} Pullup 20 µA	$V_I = V_{SS}$	-40		–5	
		I_{IL} Pullup 100 µA	$V_I = V_{SS}$	–195		–40	
		All other pins	No pullup or pulldown	–1		1	
I_{OL}	Low-level output current	Pins with output buffers of 8 mA drive strength	V_{OLmax}			8	mA
		Pins with output buffers of 4 mA drive strength				4	
		Pins with output buffers of 2 mA drive strength				2	
I_{OH}	High-level output current	Pins with output buffers of 8 mA drive strength	V_{OLmin}	–8			mA
		Pins with output buffers of 4 mA drive strength		–4			
		Pins with output buffers of 2 mA drive strength		–2			
C_I	Input capacitance				2		pF
C_O	Output capacitance				3		pF

(1) Source currents (out of the device) are negative while sink currents (into the device) are positive.
(2) This does not apply to the nPORRST pin.

5.9 Thermal Resistance Characteristics for the BGA Package (ZWT)

Over operating free-air temperature range (unless otherwise noted) (1)

		°C / W
$R\Theta_{JA}$	Junction-to-free air thermal resistance, still air (includes 5×5 thermal via cluster in 2s2p PCB connected to 1st ground plane)	14.3
$R\Theta_{JB}$	Junction-to-board thermal resistance (includes 5×5 thermal via cluster in 2s2p PCB connected to 1st ground plane)	5.49
$R\Theta_{JC}$	Junction-to-case thermal resistance (2s0p PCB)	5.02
Ψ_{JT}	Junction-to-package top, still air (includes 5×5 thermal via cluster in 2s2p PCB connected to 1st ground plane)	0.29
Ψ_{JB}	Junction-to-board, still air (includes 5×5 thermal via cluster in 2s2p PCB connected to 1st ground plane)	6.41

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report SPRA953

5.10 Timing and Switching Characteristics

5.10.1 Input Timings

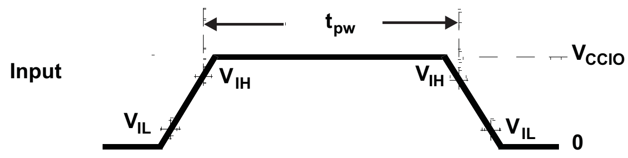


Figure 5-2. TTL-Level Inputs

Table 5-3. Timing Requirements for Inputs(1)

		MIN	MAX	UNIT
t_{pw}	Input minimum pulse width	$t_{c(VCLK)} + 10$(2)		ns
t_{in_slew}	Time for input signal to go from V_{IL} to V_{IH} or from V_{IH} to V_{IL}		1	ns

(1) $t_{c(VCLK)}$ = peripheral VBUS clock cycle time = 1 / $f_{(VCLK)}$
(2) The timing shown above is only valid for pin used in general-purpose input mode.

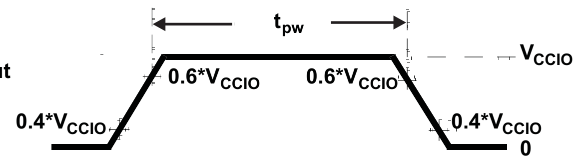


Figure 5-3. FlexRay Inputs

Table 5-4. Timing Requirements for FlexRay Inputs(1)

		MIN	MAX	UNIT
t_{pw}	Input minimum pulse width to meet the FlexRay sampling requirement	$t_{c(VCLKA2)}$ + 2.5		ns

(1) $t_{c(VCLKA2)}$ = sample clock cycle time for FlexRay = 1 / $f_{(VCLKA2)}$

5.10.2 Output Timings

Table 5-5. Switching Characteristics for Output Timings versus Load Capacitance (CL)

PARAMETER				MIN	MAX	UNIT
Rise time, t_r	8 mA low EMI pins		CL = 15 pF		2.5	ns
			CL = 50 pF		4	
			CL = 100 pF		7.2	
			CL = 150 pF		12.5	
Fall time, t_f			CL = 15 pF		2.5	ns
			CL = 50 pF		4	
			CL = 100 pF		7.2	
			CL = 150 pF		12.5	
Rise time, t_r	4 mA low EMI pins		CL = 15 pF		5.6	ns
			CL = 50 pF		10.4	
			CL = 100 pF		16.8	
			CL = 150 pF		23.2	
Fall time, t_f			CL = 15 pF		5.6	ns
			CL= 50 pF		10.4	
			CL = 100 pF		16.8	
			CL = 150 pF		23.2	
Rise time, t_r	2 mA-z low EMI pins		CL = 15 pF		8	ns
			CL = 50 pF		15	
			CL = 100 pF		23	
			CL = 150 pF		33	
Fall time, t_f			CL = 15 pF		8	ns
			CL = 50 pF		15	
			CL = 100 pF		23	
			CL = 150 pF		33	
Rise time, t_r	Selectable 8mA / 2mA-z pins	8 mA mode	CL = 15 pF		2.5	ns
			CL = 50 pF		4	
			CL = 100 pF		7.2	
			CL = 150 pF		12.5	
Fall time, t_f			CL = 15 pF		2.5	ns
			CL = 50 pF		4	
			CL = 100 pF		7.2	
			CL = 150 pF		12.5	
Rise time, t_r		2 mA-z mode	CL = 15 pF		8	ns
			CL = 50 pF		15	
			CL = 100 pF		23	
			CL = 150 pF		33	
Fall time, t_f			CL = 15 pF		8	ns
			CL = 50 pF		15	
			CL = 100 pF		23	
			CL = 150 pF		33	

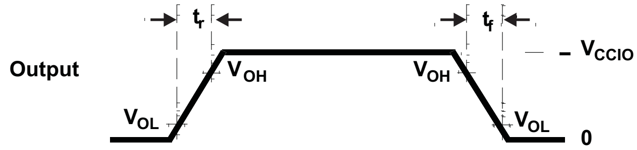


Figure 5-4. CMOS-Level Outputs

Table 5-6. Timing Requirements for Outputs(1)

		MIN	MAX	UNIT
$t_{d(parallel\_out)}$	Delay between low to high, or high to low transition of general-purpose output signals that can be configured by an application in parallel, for example, all signals in a GIOA port, or all N2HET1 signals, and so forth.		6	ns

(1) This specification does not account for any output buffer drive strength differences or any external capacitive loading differences. Check for output buffer drive strength information on each signal.

6 System Information and Electrical Specifications

6.1 Device Power Domains

The device core logic is split up into multiple virtual power domains to optimize the power for a given application use case.

This device has six logic power domains: PD1, PD2, PD3, PD4, PD5, and PD6. PD1 is a domain which cannot turn off of its clocks at once through the Power-Management Module (PMM). However, individual clock domain operating in PD1 can be individually enabled or disabled through the SYS.CDDIS register. Each of the other power domains can be turned ON, IDLE or OFF as per the application requirement through the PMM module.

In this device, a power domain can operate in one of the three possible power states: ON, IDLE and OFF. ON state is the normal operating state where clocks are actively running in the power domain. When clocks are turned off, the dynamic current is removed from the power domain. In this device, both the IDLE and OFF states have the same power characteristic. When put into either the IDLE or the OFF state, only clocks are turned off from the power domain. Leakage current from the power domain still remains. Note that putting a power domain in the OFF state will not remove any leakage current in this device. In changing the power domain states, the user must poll the system status register to check the completion of the transition. From a programmer model perspective, all three power states are available from the PMM module.

The actual management of the power domains and the hand-shaking mechanism is managed by the PMM. Refer to the Power Management Module (PMM) chapter of the device technical reference manual for more details.

6.2 Voltage Monitor Characteristics

A voltage monitor is implemented on this device. The purpose of this voltage monitor is to eliminate the requirement for a specific sequence when powering up the core and I/O voltage supplies.

6.2.1 Important Considerations

- The voltage monitor does not eliminate the need of a voltage supervisor circuit to ensure that the device is held in reset when the voltage supplies are out of range.
- The voltage monitor only monitors the core supply (VCC) and the I/O supply (VCCIO). The other supplies are not monitored by the VMON. For example, if the VCCAD or VCCP are supplied from a source different from that for VCCIO, then there is no internal voltage monitor for the VCCAD and VCCP supplies.

6.2.2 Voltage Monitor Operation

The voltage monitor generates the Power Good MCU signal (PGMCU) as well as the I/Os Power Good IO signal (PGIO) on the device. During power-up or power-down, the PGMCU and PGIO are driven low when the core or I/O supplies are lower than the specified minimum monitoring thresholds. The PGIO and PGMCU being low isolates the core logic as well as the I/O controls during power up or power down of the supplies. This allows the core and I/O supplies to be powered up or down in any order.

When the voltage monitor detects a low voltage on the I/O supply, it will assert a power-on reset. When the voltage monitor detects an out-of-range voltage on the core supply, it asynchronously makes all output pins high impedance, and asserts a power-on reset. The I/O supply must be above the threshold for monitoring the core supply. The voltage monitor is disabled when the device enters a low power mode.

The VMON also incorporates a glitch filter for the nPORRST input. Refer to Section 6.3.3.1 for the timing information on this glitch filter.

Table 6-1. Voltage Monitoring Specifications

PARAMETER			MIN	TYP	MAX	UNIT
V_{MON}	Voltage monitoring thresholds	VCC low - VCC level below this threshold is detected as too low.	0.75	0.9	1.13	V
		VCC high - VCC level above this threshold is detected as too high.	1.40	1.7	2.1	
		VCCIO low - VCCIO level below this threshold is detected as too low.	1.85	2.4	2.99	

6.2.3 Supply Filtering

The VMON has the capability to filter glitches on the VCC and VCCIO supplies.

Table 6-2 lists the characteristics of the supply filtering. Glitches in the supply larger than the maximum specification cannot be filtered.

Table 6-2. VMON Supply Glitch Filtering Capability

PARAMETER	MIN	MAX	UNIT
Width of glitch on VCC that can be filtered	250	1000	ns
Width of glitch on VCCIO that can be filtered	250	1000	ns

6.3 Power Sequencing and Power-On Reset

6.3.1 Power-Up Sequence

There is no timing dependency between the ramp of the VCCIO and the VCC supply voltage. The power-up sequence starts with the I/O voltage rising above the minimum I/O supply threshold, (for more details, see Table 6-3), core voltage rising above the minimum core supply threshold and the release of power-on reset. The high-frequency oscillator will start up first and its amplitude will grow to an acceptable level. The oscillator start-up time is dependent on the type of oscillator and is provided by the oscillator vendor. The different supplies to the device can be powered up in any order.

The device goes through the following sequential phases during power up.

Table 6-3. Power-Up Phases

Oscillator start-up and validity check	1024 oscillator cycles
eFuse autoload	3650 oscillator cycles
Flash pump power-up	250 oscillator cycles
Flash bank power-up	1460 oscillator cycles
Total	**6384 oscillator cycles**

The CPU reset is released at the end of the above sequence and fetches the first instruction from address 0x00000000.

6.3.2 Power-Down Sequence

The different supplies to the device can be powered down in any order.

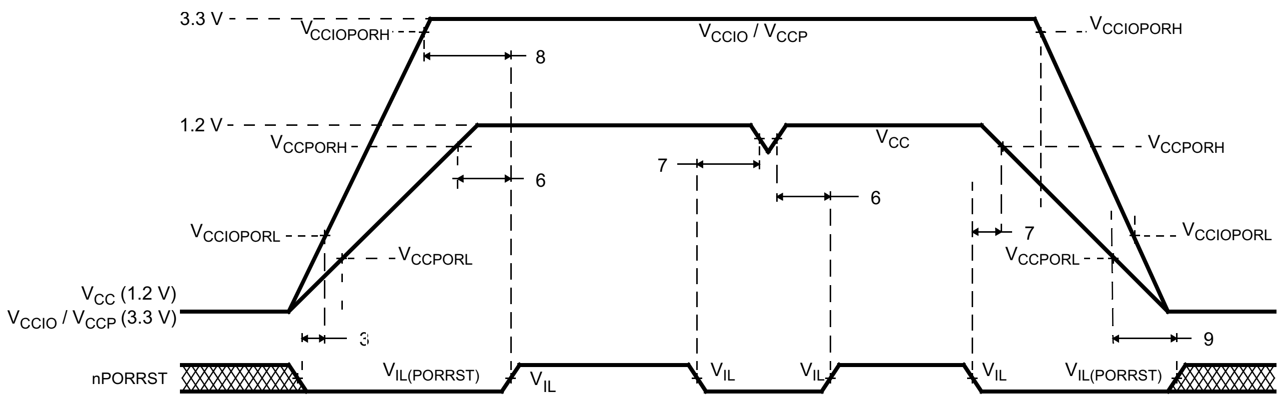
6.3.3 Power-On Reset: nPORRST

This is the power-on reset. This reset must be asserted by an external circuitry whenever the I/O or core supplies are outside the specified recommended range. This signal has a glitch filter on it. It also has an internal pulldown.

6.3.3.1 nPORRST Electrical and Timing Requirements

Table 6-4. Electrical Requirements for nPORRST

NO.			MIN	MAX	UNIT
	V_{CCPORL}	V_{CC} low supply level when nPORRST must be active during power up		0.5	V
	V_{CCPORH}	V_{CC} high supply level when nPORRST must remain active during power up and become active during power down	1.14		V
	$V_{CCIOPORL}$	V_{CCIO} / V_{CCP} low supply level when nPORRST must be active during power up		1.1	V
	$V_{CCIOPORH}$	V_{CCIO} / V_{CCP} high supply level when nPORRST must remain active during power up and become active during power down	3.0		V
	$V_{IL(PORRST)}$	Low-level input voltage of nPORRST V_{CCIO} > 2.5 V		0.2 * V_{CCIO}	V
		Low-level input voltage of nPORRST V_{CCIO} < 2.5 V		0.5	
3	$t_{su(PORRST)}$	Setup time, nPORRST active before V_{CCIO} and V_{CCP} > $V_{CCIOPORL}$ during power up	0		ms
6	$t_{h(PORRST)}$	Hold time, nPORRST active after V_{CC} > V_{CCPORH}	1		ms
7	$t_{su(PORRST)}$	Setup time, nPORRST active before V_{CC} < V_{CCPORH} during power down	2		µs
8	$t_{h(PORRST)}$	Hold time, nPORRST active after V_{CCIO} and V_{CCP} > $V_{CCIOPORH}$	1		ms
9	$t_{h(PORRST)}$	Hold time, nPORRST active after V_{CC} < V_{CCPORL}	0		ms
	$t_{f(nPORRST)}$	Filter time nPORRST terminal; pulses less than MIN will be filtered out, pulses greater than MAX will generate a reset. Pulses greater than MIN but less than MAX may or may not generate a reset.	475	2000	ns



A. Figure 6-1 shows that there is no timing dependency between the ramp of the V_{CCIO} and the V_{CC} supply voltages.

Figure 6-1. nPORRST Timing Diagram(A)

6.4 Warm Reset (nRST)

This is a bidirectional reset signal. The internal circuitry drives the signal low on detecting any device reset condition. An external circuit can assert a device reset by forcing the signal low. On this terminal, the output buffer is implemented as an open drain (drives low only). To ensure an external reset is not arbitrarily generated, TI recommends that an external pullup resistor is connected to this terminal.

This terminal has a glitch filter. It also has an internal pullup

6.4.1 Causes of Warm Reset

Table 6-5. Causes of Warm Reset

DEVICE EVENT	SYSTEM STATUS FLAG
Power-Up Reset	Exception Status Register, bit 15
Oscillator fail	Global Status Register, bit 0
PLL slip	Global Status Register, bits 8 and 9
Watchdog exception	Exception Status Register, bit 13
Debugger reset	Exception Status Register, bit 11
CPU Reset (driven by the CPU STC)	Exception Status Register, bit 5
Software Reset	Exception Status Register, bit 4
External Reset	Exception Status Register, bit 3

6.4.2 nRST Timing Requirements

Table 6-6. nRST Timing Requirements(1)

		MIN	MAX	UNIT
$t_{v(RST)}$	Valid time, nRST active after nPORRST inactive	$5032t_{c(OSC)}$		ns
	Valid time, nRST active (all other System reset conditions)	$32t_{c(VCLK)}$		
$t_{f(nRST)}$	Filter time nRST terminal; pulses less than MIN will be filtered out, pulses greater than MAX will generate a reset	475	2000	ns

(1) Specified values ***do not*** include rise/fall times. For rise and fall timings, see Table 5-5.

6.5 ARM Cortex-R5F CPU Information

6.5.1 Summary of ARM Cortex-R5F CPU Features

The features of the ARM Cortex-R5F CPU include:

- An integer unit with integral Embedded ICE-RT logic.
- High-speed Advanced Microprocessor Bus Architecture (AMBA) Advanced eXtensible Interfaces (AXI) for Level two (L2) master and slave interfaces.
- Floating-Point Coprocessor
- Dynamic branch prediction with a global history buffer, and a 4-entry return stack
- Low interrupt latency.
- Nonmaskable interrupt.
- Harvard Level one (L1) memory system with:
  - 32KB of instruction cache and 32KB of data cache implemented. Both Instruction and data cache have ECC support.
  - ARMv7-R architecture Memory Protection Unit (MPU) with 16 regions
- Dual core logic for fault detection in safety-critical applications.
- L2 memory interface:
  - Single 64-bit master AXI interface
  - 64-bit slave AXI interface to cache memories
  - 32-bit AXI_Peri ports to support low latency peripheral ports
- Debug interface to a CoreSight Debug Access Port (DAP).
- Performance Monitoring Unit (PMU).
- Vectored Interrupt Controller (VIC) port.
- AXI accelerator coherency port (ACP) supporting IO coherency with write-through cacheable regions
- Ability to generate ECC on L2 data buses and parity of all control channels
- Both CPU cores in lock-step
- Eight hardware breakpoints
- Eight watchpoints

6.5.2 Dual Core Implementation

The device has two Cortex-R5F cores, where the output signals of both CPUs are compared in the CCM-R5F unit. To avoid common mode impacts the signals of the CPUs to be compared are delayed by two clock cycles as shown in Figure 6-2.

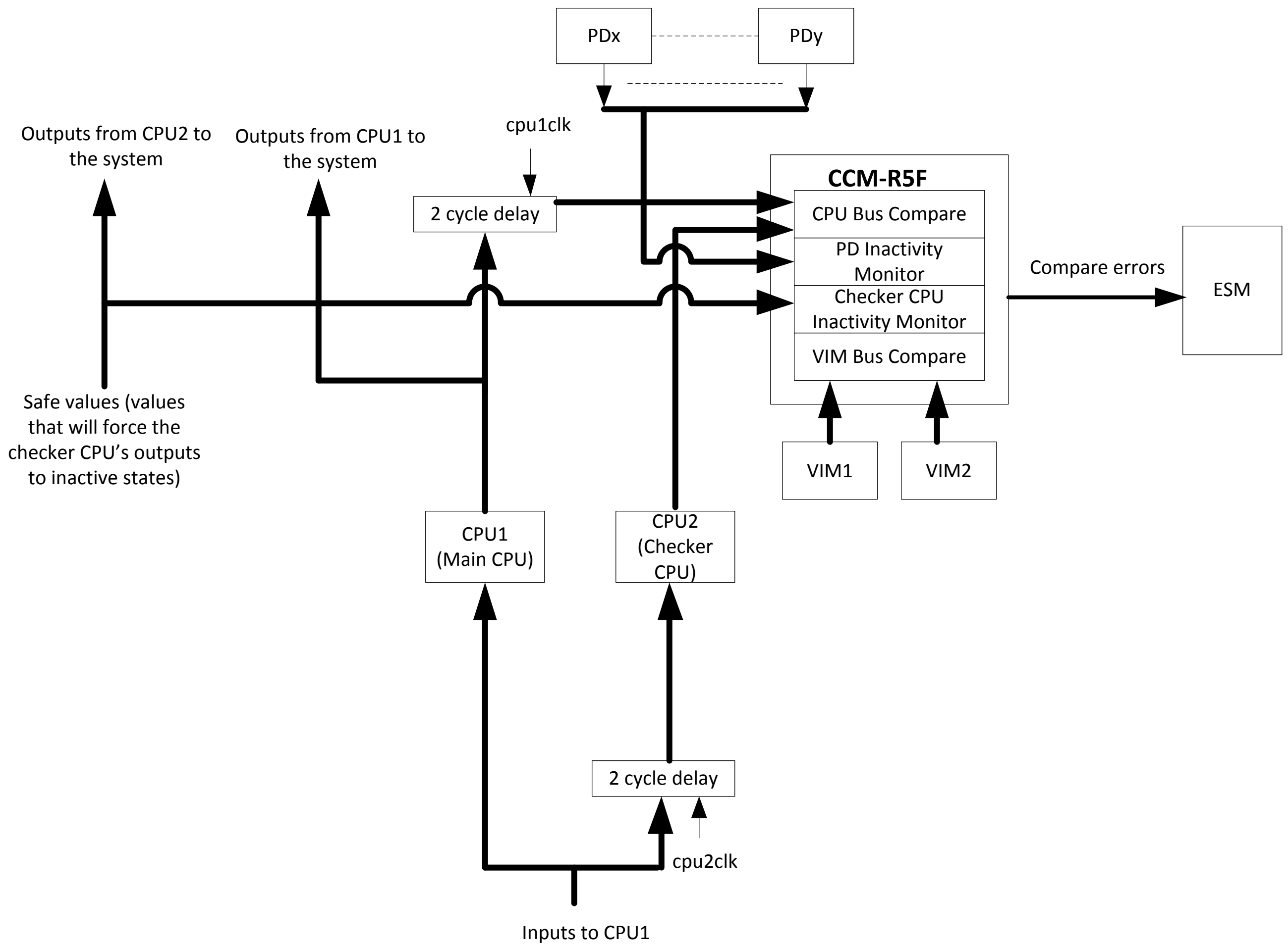


Figure 6-2. Dual Core Implementation

6.5.3 Duplicate Clock Tree After GCLK

The CPU clock domain is split into two clock trees, one for each CPU, with the clock of the second CPU running at the same frequency and in phase to the clock of CPU1. See Figure 6-2.

6.5.4 ARM Cortex-R5F CPU Compare Module (CCM) for Safety

CCM-R5F has two major functions. One is to compare the outputs of two Cortex-R5F processor cores and the VIM modules. The second function is inactivity monitoring, to detect any faulted transaction initiated by the checker core.

6.5.4.1 Signal Compare Operating Modes

The CCM-R5F module run in one of four operating modes - active compare lockstep, self-test, error forcing, and self-test error forcing mode. To select an operating mode, a dedicated key must be written to the key register. CPU compare block and VIM compare block have separate key registers to select their operating modes. Status registers are also separate for these blocks.

6.5.4.1.1 Active Compare Lockstep Mode

In this mode the output signals of both CPUs and both VIMs are compared, and a difference in the outputs is indicated by the compare_error terminal. For more details about CPU and VIM lockstep comparison, refer to the device technical reference manual.

CCM-R5F also produces a signal to ESM GP1.92 to indicate its current status whether it is out of lockstep or is in self-test mode. This ensures that any lock step fault is reported to the CPU.

6.5.4.1.2 Self-Test Mode

In self-test mode the CCM-R5F is checked for faults, by applying internally generated, series of test patterns to look for any hardware faults inside the module. During self-test the compare error signal is deactivated. If a fault on the CCM-R5F module is detected, an error is shown on the selftest_error pin.

6.5.4.1.3 Error Forcing Mode

In error forcing mode a test pattern is applied to the CPU and VIM related inputs of the compare logic to force an error at the compare error signal of the compare unit. Error forcing mode is done separately for VIM signal compare block and CPU signal compare block. For each block, this mode is enabled by writing the key in corresponding block's key register.

6.5.4.1.4 Self-Test Error Forcing Mode

In self-test error forcing mode an error is forced at the self-test error signal. The compare block is still running in lockstep mode and the key is switched to lockstep after one clock cycle.

Table 6-7. CPU Compare Self-Test Cycles

MODE	NUMBER OF GCLK CYCLES
Self-Test Mode	4947
Self-Test Error Forcing Mode	1
Error Forcing Mode	1

Table 6-8. VIM Compare Self-Test Cycles

MODE	NUMBER OF VCLK CYCLES
Self-Test Mode	151
Self-Test Error Forcing Mode	1
Error Forcing Mode	1

6.5.4.2 Bus Inactivity Monitor

CCM-R5F also monitors the inputs to the interconnect coming from the checker Cortex-R5F core. The input signals to the interconnect are compared against their default clamped values. The checker core must not generate any bus transaction to the interconnect system as all bus transactions are carried out through the main CPU core. If any signal value is different from its clamped value, an error signal is generated. The error response in case of a detected transaction is sent to ESM.

In addition to bus monitoring the checker CPU core, the CCM-R5F will also monitor several other critical signals from other masters residing in other power domains. This is to ensure an inadvertent bus transaction from an unused power domain can be detected. To enable detection of unwanted transaction from an unused master, the power domain in which the master to be monitored will need to be configured in OFF power state through the PMM module.

6.5.4.3 CPU Registers Initialization

To avoid an erroneous CCM-R5F compare error, the application software must ensure that the CPU registers of both CPUs are initialized with the same values before the registers are used, including function calls where the register values are pushed onto the stack.

Example routine for CPU register initialization:

```
     .text
     .state32
     .global __clearRegisters_
     .asmfunc
 __clearRegisters_:
        mov   r0,  lr
        mov   r1,  #0x0000
        mov   r2,  #0x0000
        mov   r3,  #0x0000
        mov   r4,  #0x0000
        mov   r5,  #0x0000
        mov   r6,  #0x0000
        mov   r7,  #0x0000
        mov   r8,  #0x0000
        mov   r9,  #0x0000
        mov   r10, #0x0000
        mov   r11, #0x0000
        mov   r12, #0x0000
        mov   r1,  #0x11                            ; FIQ Mode = 10001
        msr   cpsr, r1
        msr   spsr, r1
        mov   lr,  r0
        mov   r8,  #0x0000                          ; Registers R8 to R12 are also
banked in FIQ mode
        mov   r9,  #0x0000
        mov   r10, #0x0000
        mov   r11, #0x0000
        mov   r12, #0x0000
        mov   r1, #0x13                             ; SVC Mode = 10011
        msr   cpsr, r1
        msr   spsr, r1
        mov   lr,  r0
        mov   r1, #0x17                             ; ABT Mode = 10111
        msr   cpsr, r13
        msr   spsr, r13
        mov   lr,  r0
        mov   r1, #0x12                             ; IRQ Mode = 10010
        msr   cpsr, r13
        msr   spsr, r13
        mov   lr,  r0
        mov   r1, #0x1B                             ; UDEF Mode = 11011
        msr   cpsr, r13
        msr   spsr, r13
        mov   lr,  r0
        mov   r1, #0xDF                             ; System Mode = 11011111
        msr   cpsr, r13
        msr   spsr, r13

; Floating Point Co-Processor Initialization. FPU needs to be enabled first.

        mrc   p15,    #0x00,      r2,      c1, c0, #0x02
        orr   r2,     r2,         #0xF00000
        mcr   p15,    #0x00,      r2,      c1, c0, #0x02
        mov   r2,     #0x40000000
        fmxr  fpexc,   r2

        fmdrr d0,  r1,    r1
        fmdrr d1,  r1,    r1
        fmdrr d2,  r1,    r1
        fmdrr d3,  r1,    r1
        fmdrr d4,  r1,    r1
        fmdrr d5,  r1,    r1
        fmdrr d6,  r1,    r1
        fmdrr d7,  r1,    r1
        fmdrr d8,  r1,    r1
        fmdrr d9,  r1,    r1
        fmdrr d10, r1,    r1
        fmdrr d11, r1,    r1
        fmdrr d12, r1,    r1
        fmdrr d13, r1,    r1
        fmdrr d14, r1,    r1
        fmdrr d15, r1,    r1
        bl    $+4
        bl    $+4
        bl    $+4
        bl    $+4
        bx    r0

.endasmfunc
```

6.5.5 CPU Self-Test

The CPU STC (Self-Test Controller) is used to test the two Cortex-R5F CPU Cores using the Deterministic Logic BIST Controller as the test engine.

The main features of the self-test controller are:

- Ability to divide the complete test run into independent test intervals
- Capable of running the complete test as well as running few intervals at a time
- Ability to continue from the last executed interval (test set) as well as ability to restart from the beginning (First test set)
- Complete isolation of the self-tested CPU core from rest of the system during the self-test run
- Ability to capture the Failure interval number
- Time-out counter for the CPU self-test run as a fail-safe feature

6.5.5.1 Application Sequence for CPU Self-Test

1. Configure clock domain frequencies.
2. Select number of test intervals to be run.
3. Configure the time-out period for the self-test run.
4. Enable self-test.
5. Wait for CPU reset.
6. In the reset handler, read CPU self-test status to identify any failures.
7. Retrieve CPU state if required.

For more information see the device technical reference manual.

6.5.5.2 CPU Self-Test Clock Configuration

The maximum clock rate for the self-test is 110 MHz. The STCCLK is divided down from the CPU clock. This divider is configured by the STCCLKDIV register at address 0xFFFFE644.

For more information see the device-specific Technical Reference Manual.

6.5.5.3 CPU Self-Test Coverage

The self-test, if enabled, is automatically applied to the entire processor group. Self-test will only start when nCLKSTOPPEDm is asserted which indicates the CPU cores and the ACP interface are in quiescent state. While the processor group is in self-test, other masters can still function normally including accesses to the system memory such as the L2 SRAM. Because uSCU is part of the processor group under self-test, the cache coherency checking will be bypassed.

When the self-test is completed, reset is asserted to all logic subjected to self-test. After self-test is complete, software must invalidate the cache accordingly.

The default value of the CPU LBIST clock prescaler is' divide-by-1'. A prescalar in the STC module can be used to configure the CPU LBIST frequency with respect to the CPU GCLK frequency.

Table 6-9 lists the CPU test coverage achieved for each self-test interval. It also lists the cumulative test cycles. The test time can be calculated by multiplying the number of test cycles with the STC clock period.

Table 6-9. CPU Self-Test Coverage

INTERVALS	TEST COVERAGE, %	TEST CYCLES
0	0	0
1	56.85	1629
2	64.19	3258
3	68.76	4887
4	71.99	6516
5	75	8145
6	76.61	9774
7	78.08	11403
8	79.2	13032
9	80.18	14661
10	81.03	16290
11	81.9	17919
12	82.58	19548
13	83.24	21177
14	83.73	22806
15	84.15	24435
16	84.52	26064
17	84.9	27693
18	85.26	29322
19	85.68	30951
20	86.05	32580
21	86.4	34209
22	86.68	35838
23	86.94	37467
24	87.21	39096
25	87.48	40725
26	87.74	42354
27	87.98	43983
28	88.18	45612
29	88.38	47241
30	88.56	48870
31	88.75	50499
32	88.93	52128
33	89.1	53757
34	89.23	55386
35	89.41	57015
36	89.55	58644
37	89.7	60273
38	89.83	61902
39	89.96	63531
40	90.1	65160

6.5.6 N2HET STC / LBIST Self-Test Coverage

Logic BIST self-test capability for N2HETs is available in this device. The STC2 can be configured to perform self-test for both N2HETs at the same time or one at the time. The default value of the N2HET LBIST clock prescaler is divide-by-1. However, the maximum clock rate for the N2HET STC / LBIST is VCLK/2. N2HET STC test should not be executed concurrently with CPU STC test.

Table 6-10. N2HET Self-Test Coverage

INTERVALS	TEST COVERAGE, %	TEST CYCLES
0	0	0
1	70.01	1365
2	77.89	2730
3	81.73	4095
4	84.11	5460
5	86.05	6825
6	87.78	8190
7	88.96	9555
8	89.95	10920
9	90.63	12285

6.6 Clocks

6.6.1 Clock Sources

Table 6-11 lists the available clock sources on the device. Each clock source can be enabled or disabled using the CSDISx registers in the system module. The clock source number in the table corresponds to the control bit in the CSDISx register for that clock source.

Table 6-11 also lists the default state of each clock source.

Table 6-11. Available Clock Sources

CLOCK SOURCE NO.	NAME	DESCRIPTION	DEFAULT STATE
0	OSCIN	Main Oscillator	Enabled
1	PLL1	Output From PLL1	Disabled
2	Reserved	Reserved	Disabled
3	EXTCLKIN1	External Clock Input 1	Disabled
4	CLK80K	Low-Frequency Output of Internal Reference Oscillator	Enabled
5	CLK10M	High-Frequency Output of Internal Reference Oscillator	Enabled
6	PLL2	Output From PLL2	Disabled
7	EXTCLKIN2	External Clock Input 2	Disabled

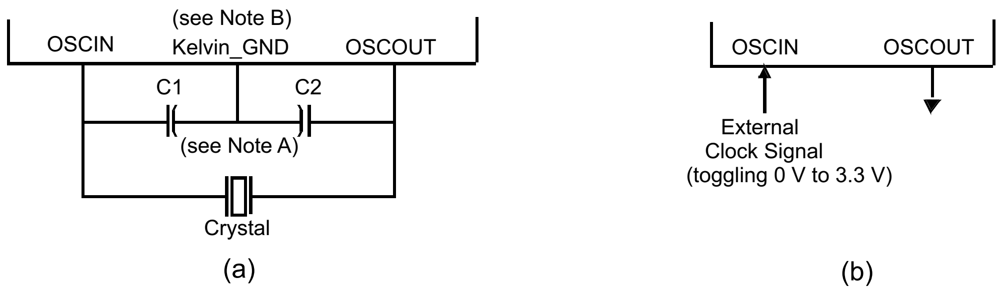
6.6.1.1 Main Oscillator

The oscillator is enabled by connecting the appropriate fundamental resonator/crystal and load capacitors across the external OSCIN and OSCOUT pins as shown in Figure 6-3. The oscillator is a single-stage inverter held in bias by an integrated bias resistor. This resistor is disabled during leakage test measurement and low power modes.

NOTE

TI strongly encourages each customer to submit samples of the device to the resonator/crystal vendors for validation. The vendors are equipped to determine which load capacitors will best tune their resonator/crystal to the microcontroller device for optimum start-up and operation over temperature and voltage extremes.

An external oscillator source can be used by connecting a 3.3-V clock signal to the OSCIN terminal and leaving the OSCOUT terminal unconnected (open) as shown in Figure 6-3.



Note A: The values of C1 and C2 should be provided by the resonator/crystal vendor.
Note B: Kelvin_GND should not be connected to any other GND.

Figure 6-3. Recommended Crystal/Clock Connection

6.6.1.1.1 Timing Requirements for Main Oscillator

Table 6-12. Timing Requirements for Main Oscillator

		MIN	NOM	MAX	UNIT
$t_{c(OSC)}$	Cycle time, OSCIN (when using a sine-wave input)	50		200	ns
$t_{c(OSC\_SQR)}$	Cycle time, OSCIN, (when input to the OSCIN is a square wave)	50		200	ns
$t_{w(OSCIL)}$	Pulse duration, OSCIN low (when input to the OSCIN is a square wave)	15			ns
$t_{w(OSCIH)}$	Pulse duration, OSCIN high (when input to the OSCIN is a square wave)	15			ns

6.6.1.2 Low-Power Oscillator

The Low-Power Oscillator (LPO) is comprised of two oscillators — HF LPO and LF LPO, in a single macro.

6.6.1.2.1 Features

The main features of the LPO are:

- Supplies a clock at extremely low power to reduce power consumption. This is connected as clock source 4 of the Global Clock Module (GCM).
- Supplies a high-frequency clock for nontiming-critical systems. This is connected as clock source 5 of the GCM.
- Provides a comparison clock for the crystal oscillator failure detection circuit.

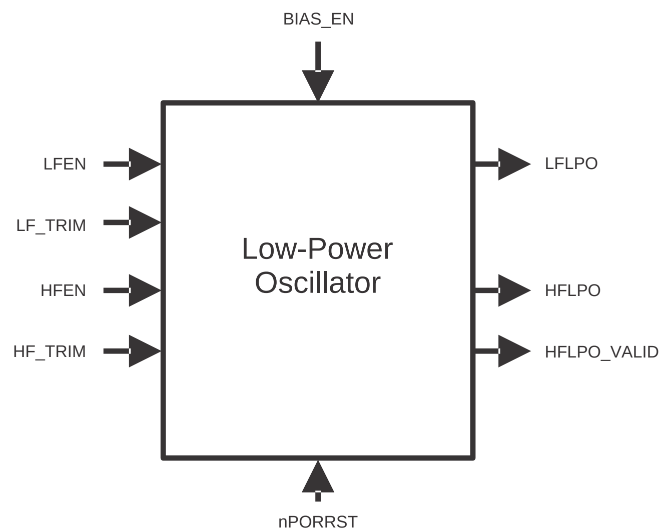


Figure 6-4. LPO Block Diagram

Figure 6-4 shows a block diagram of the internal reference oscillator. This is a low-power oscillator (LPO) and provides two clock sources: one nominally 80 kHz and one nominally 10 MHz.

6.6.1.2.2 LPO Electrical and Timing Specifications

Table 6-13. LPO Specifications

PARAMETER		MIN	TYP	MAX	UNIT
Clock detection	Oscillator fail frequency - lower threshold, using untrimmed LPO output	1.375	2.4	4.875	MHz
	Oscillator fail frequency - higher threshold, using untrimmed LPO output	22	38.4	78	MHz
LPO - HF oscillator	Untrimmed frequency	5.5	9	19.5	MHz
	Trimmed frequency	8.0	9.6	11.0	MHz
	Start-up time from STANDBY (LPO BIAS_EN high for at least 900 µs)			10	µs
	Cold start-up time			900	µs
LPO - LF oscillator	Untrimmed frequency	36	85	180	kHz
	Start-up time from STANDBY (LPO BIAS_EN high for at least 900 µs)			100	µs
	Cold start-up time			2000	µs

6.6.1.3 Phase-Locked Loop (PLL) Clock Modules

The PLL is used to multiply the input frequency to some higher frequency.

The main features of the PLL are:

- Frequency modulation can be optionally superimposed on the synthesized frequency of PLL1. The frequency modulation capability of PLL2 is permanently disabled.
- Configurable frequency multipliers and dividers
- Built-in PLL Slip monitoring circuit
- Option to reset the device on a PLL slip detection

6.6.1.3.1 Block Diagram

Figure 6-5 shows a high-level block diagram of the two PLL macros on this microcontroller. PLLCTL1 and PLLCTL2 are used to configure the multiplier and dividers for the PLL1. PLLCTL3 is used to configure the multiplier and dividers for PLL2.

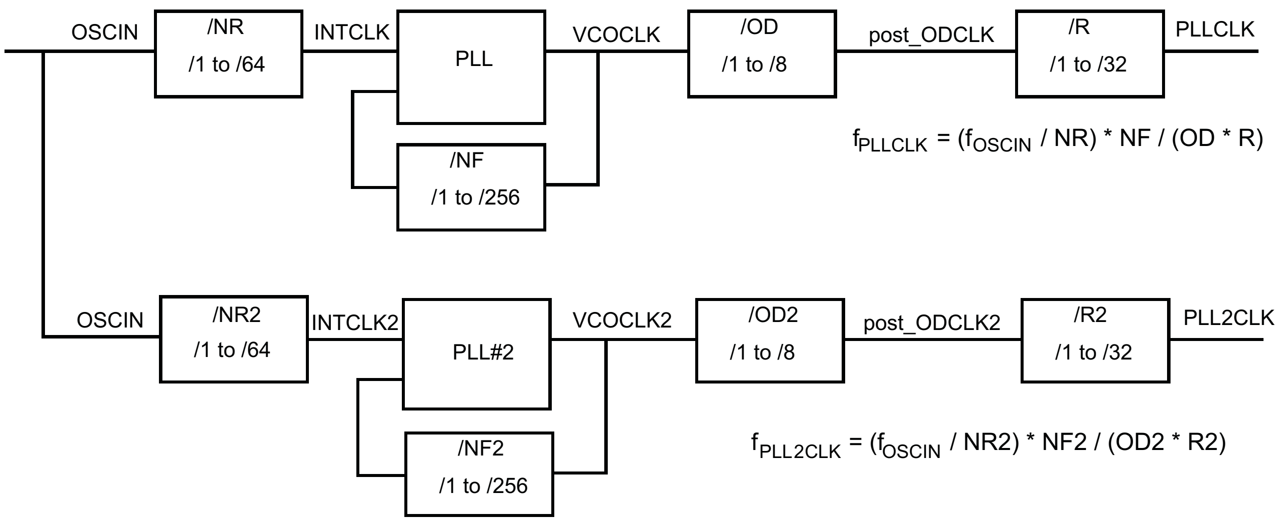


Figure 6-5. ZWT PLLx Block Diagram

6.6.1.3.2 PLL Timing Specifications

Table 6-14. PLL Timing Specifications

PARAMETER		MIN	MAX	UNIT
f_{INTCLK}	PLL1 Reference Clock frequency	1	20	MHz
f_{post_ODCLK}	Post-ODCLK – PLL1 Post-divider input clock frequency		400	MHz
f_{VCOCLK}	VCOCLK – PLL1 Output Divider (OD) input clock frequency		550	MHz
$f_{INTCLK2}$	PLL2 Reference Clock frequency	1	20	MHz
f_{post_ODCLK2}	Post-ODCLK – PLL2 Post-divider input clock frequency		400	MHz
$f_{VCOCLK2}$	VCOCLK – PLL2 Output Divider (OD) input clock frequency		550	MHz

6.6.1.4 External Clock Inputs

The device supports up to two external clock inputs. This clock input must be a square-wave input. Table 6-15 specifies the electrical and timing requirements for these clock inputs.

Table 6-15. External Clock Timing and Electrical Specifications

PARAMETER		MIN	MAX	UNIT
$f_{EXTCLKx}$	External clock input frequency		80	MHz
$t_{w(EXTCLKIN)H}$	EXTCLK high-pulse duration	6		ns
$t_{w(EXTCLKIN)L}$	EXTCLK low-pulse duration	6		ns
$v_{iL(EXTCLKIN)}$	Low-level input voltage	–0.3	0.8	V
$v_{iH(EXTCLKIN)}$	High-level input voltage	2	VCCIO + 0.3	V

6.6.2 Clock Domains

6.6.2.1 Clock Domain Descriptions

Table 6-16 lists the device clock domains and their default clock sources. Table 6-16 also lists the system module control register that is used to select an available clock source for each clock domain.

Table 6-16. Clock Domain Descriptions

CLOCK DOMAIN	CLOCK DISABLE BIT	DEFAULT SOURCE	SOURCE SELECTION REGISTER	SPECIAL CONSIDERATIONS
GCLK1	SYS.CDDIS.0	OSCIN	SYS.GHVSRC[3:0]	• This the main clock from which HCLK is divided down • In phase with HCLK • Is disabled separately from HCLK through the CDDISx registers bit 0 • Can be divided-by-1 up to 8 when running CPU self-test (LBIST) using the CLKDIV field of the STCCLKDIV register at address 0xFFFFE108
GCLK2	SYS.CDDIS.0	OSCIN	SYS.GHVSRC[3:0]	• Always the same frequency as GCLK1 • 2 cycles delayed from GCLK1 • Is disabled along with GCLK1 • Gets divided by the same divider setting as that for GCLK1 when running CPU self-test (LBIST)
HCLK	SYS.CDDIS.1	OSCIN	SYS.GHVSRC[3:0]	• Divided from GCLK1 through HCLKCNTLregister • Allowable clock ratio from 1:1 to 4:1 • Is disabled through the CDDISx registers bit 1
VCLK	SYS.CDDIS.2	OSCIN	SYS.GHVSRC[3:0]	• Divided down from HCLK through CLKCNTL register • Can be HCLK/1, HCLK/2,... or HCLK/16 • Is disabled separately from HCLK through the CDDISx registers bit 2 • HCLK:VCLK2:VCLK must be integer ratios of each other
VCLK2	SYS.CDDIS.3	OSCIN	SYS.GHVSRC[3:0]	• Divided down from HCLK • Can be HCLK/1, HCLK/2,... or HCLK/16 • Frequency must be an integer multiple of VCLK frequency • Is disabled separately from HCLK through the CDDISx registers bit 3
VCLK3	SYS.CDDIS.8	OSCIN	SYS.GHVSRC[3:0]	• Divided down from HCLK • Can be HCLK/1, HCLK/2,... or HCLK/16 • Is disabled separately from HCLK through the CDDISx registers bit 8
VCLKA1	SYS.CDDIS.4	VCLK	SYS.VCLKASRC[3:0]	• Defaults to VCLK as the source • Is disabled through the CDDISx registers bit 4
VCLKA2	SYS.CDDIS.5	VCLK	SYS.VCLKASRC[3:0]	• Defaults to VCLK as the source • Is disabled through the CDDISx registers bit 5
VCLKA4	SYS.CDDIS.11	VCLK	SYS.VCLKACON1[19:16]	• Defaults to VCLK as the source • Is disabled through the CDDISx registers bit 11

Table 6-16. Clock Domain Descriptions (continued)

CLOCK DOMAIN	CLOCK DISABLE BIT	DEFAULT SOURCE	SOURCE SELECTION REGISTER	SPECIAL CONSIDERATIONS
VCLKA4_DIVR	SYS.VCLKACON1.20	VCLK	SYS.VCLKACON1[19:16]	• Divided down from VCLKA4 using the VCLKA4R field of the VCLKACON1 register • Frequency can be VCLKA4/1, VCLKA4/2, ..., or VCLKA4/8 • Default frequency is VCLKA4/2 • Is disabled separately through the VCLKA4_DIV_CDDIS bit in the VCLKACON1 register, if the VCLKA4 is not already disabled
RTICLK1	SYS.CDDIS.6	VCLK	SYS.RCLKSRC[3:0]	• Defaults to VCLK as the source • If a clock source other than VCLK is selected for RTICLK1, then the RTICLK1 frequency must be less than or equal to VCLK/3 • Application can ensure this by programming the RTI1DIV field of the RCLKSRC register, if necessary • Is disabled through the CDDISx registers bit 6

6.6.2.2 Mapping of Clock Domains to Device Modules

Each clock domain has a dedicated functionality as shown in Figure 6-6.

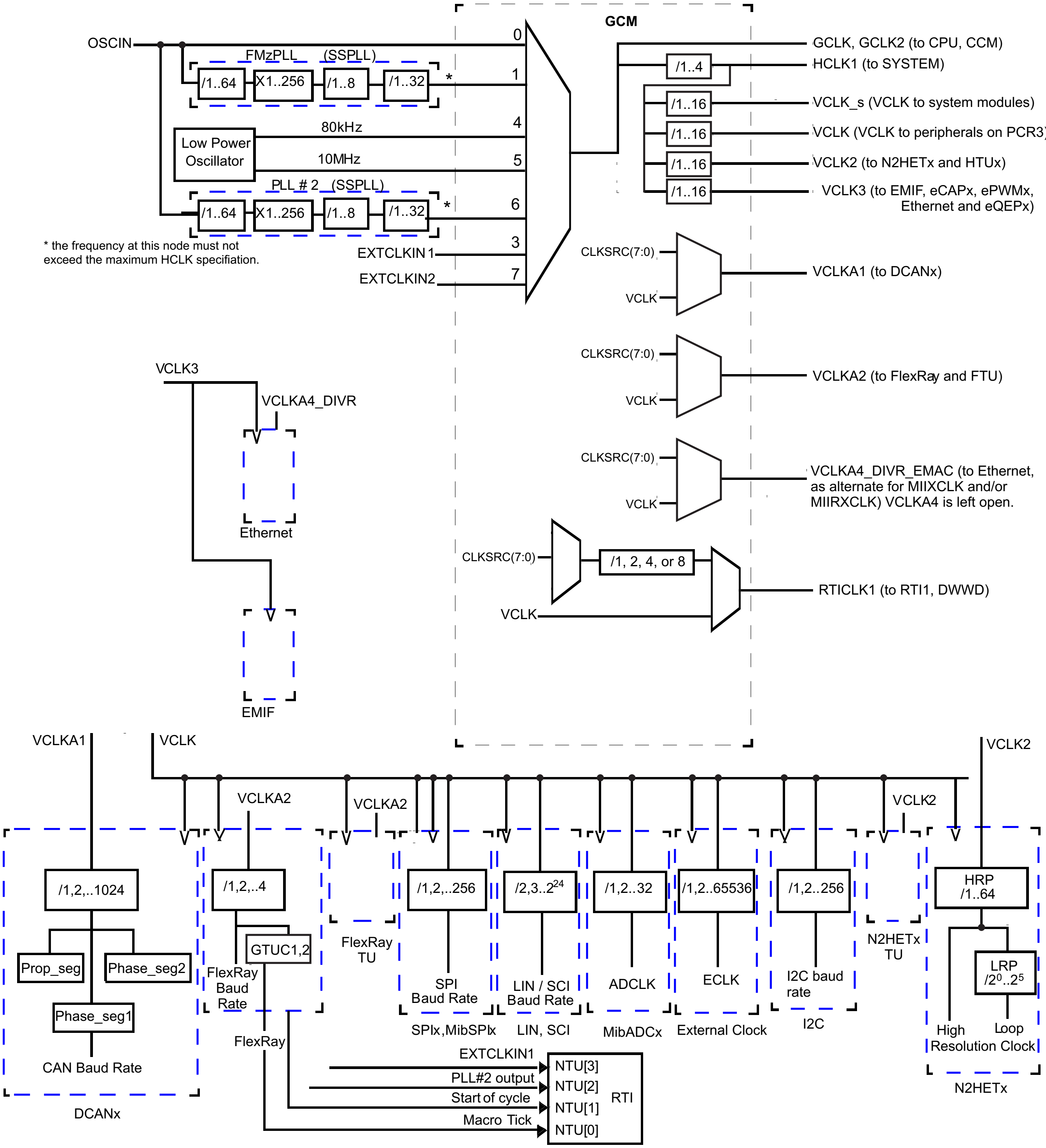


Figure 6-6. Device Clock Domains

6.6.3 Special Clock Source Selection Scheme for VCLKA4_DIVR_EMAC

Some applications may need to use both the FlexRay and the Ethernet interfaces. The FlexRay controller requires the VCLKA2 frequency to be 80 MHz, while the MII interface requires VCLKA4_DIVR_EMAC to be 25 MHz and the RMII requires VCLKA4_DIVR_EAMC to be 50 MHz.

These different frequencies are supported by adding special dedicated clock source selection options for the VCLKA4_DIVR_EMAC clock domain. This logic is shown in Figure 6-7.

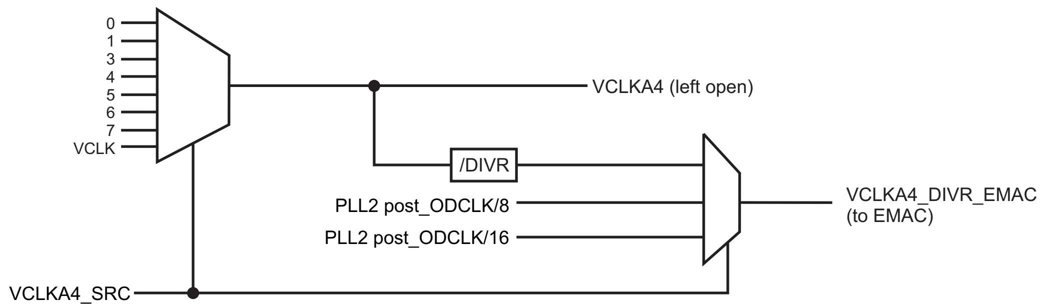


Figure 6-7. VCLKA4_DIVR Source Selection Options

The PLL2 post_ODCLK is brought out as a separate output from the PLL wrapper module. There are two additional dividers implemented at the device-level to divide this PLL2 post_ODCLK by 8 and by 16.

As shown in Figure 6-7, the VCLKA4_SRC configured through the system module VCLKACON1 control register is used to determine the clock source for the VCLKA4 and VCLKA4_DIVR. An additional multiplexor is implemented to select between the VCLKA4_DIVR and the two additional clock sources – PLL2 post_ODCLK/8 and post_ODCLK/16.

Table 6-17 lists the VCLKA4_DIVR_EMAC clock source selections.

Table 6-17. VCLKA4_DIVR_EMAC Clock Source Selection

VCLKA4_SRC FROM VCLKACON1[19–16]	CLOCK SOURCE FOR VCLKA4_DIVR_EMAC
0x0	OSCIN / VCLKA4R
0x1	PLL1CLK / VCLKA4R
0x2	Reserved
0x3	EXTCLKIN1 / VCLKA4R
0x4	LF LPO / VCLKA4R
0x5	HF LPO / VCLKA4R
0x6	PLL2CLK / VCLKA4R
0x7	EXTCLKIN2 / VCLKA4R
0x8–0xD	VCLK
0xE	PLL2 post_ODCLK/8
0xF	PLL2 post_ODCLK/16

6.6.4 Clock Test Mode

The TMS570 platform architecture defines a special mode that allows various clock signals to be selected and output on the ECLK1 terminal and N2HET1[12] device outputs. This special mode, Clock Test Mode, is very useful for debugging purposes and can be configured through the CLKTEST register in the system module. See Table 6-18 and Table 6-19 for the CLKTEST bits value and signal selection.

Table 6-18. Clock Test Mode Options for Signals on ECLK1

SEL_ECP_PIN = CLKTEST[4-0]	SIGNAL ON ECLK1
00000	Oscillator Clock
00001	PLL1 Clock Output
00010	Reserved
00011	EXTCLKIN1
00100	Low-Frequency Low-Power Oscillator (LFLPO) Clock [CLK80K]
00101	High-Frequency Low-Power Oscillator (HFLPO) Clock [CLK10M]
00110	PLL2 Clock Output
00111	EXTCLKIN2
01000	GCLK1
01001	RTI1 Base
01010	Reserved
01011	VCLKA1
01100	VCLKA2
01101	Reserved
01110	VCLKA4_DIVR
01111	Flash HD Pump Oscillator
10000	Reserved
10001	HCLK
10010	VCLK
10011	VCLK2
10100	VCLK3
10101	Reserved
10110	Reserved
10111	EMAC Clock Output
11000	Reserved
11001	Reserved
11010	Reserved
11011	Reserved
11100	Reserved
11101	Reserved
11110	Reserved
11111	Reserved

Table 6-19. Clock Test Mode Options for Signals on N2HET1[12]

SEL_GIO_PIN = CLKTEST[11-8]	SIGNAL ON N2HET1[12]
0000	Oscillator Valid Status
0001	PLL1 Valid Status
0010	Reserved
0011	Reserved
0100	Reserved
0101	HFLPO Clock Output Valid Status [CLK10M]
0110	PLL2 Valid Status
0111	Reserved
1000	LFLPO Clock Output Valid Status [CLK80K]
1001	Oscillator Valid status
1010	Oscillator Valid status
1011	Oscillator Valid status
1100	Oscillator Valid status
1101	Reserved
1110	VCLKA4
1111	Oscillator Valid status

6.7 Clock Monitoring

The LPO Clock Detect (LPOCLKDET) module consists of a clock monitor (CLKDET) and an internal LPO.

The LPO provides two different clock sources – a low frequency (CLK80K) and a high frequency (CLK10M).

The CLKDET is a supervisor circuit for an externally supplied clock signal (OSCIN). In case the OSCIN frequency falls out of a frequency window, the CLKDET flags this condition in the global status register (GLBSTAT bit 0: OSC FAIL) and switches all clock domains sourced by OSCIN to the CLK10M clock (limp mode clock).

The valid OSCIN frequency range is defined as: $f_{CLK10M} / 4 < f_{OSCIN} < f_{CLK10M} * 4$.

6.7.1 Clock Monitor Timings

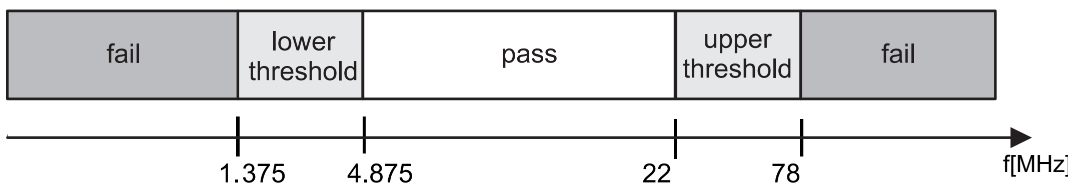


Figure 6-8. LPO and Clock Detection, Untrimmed CLK10M

6.7.2 External Clock (ECLK) Output Functionality

The ECLK1/ECLK2 terminal can be configured to output a prescaled clock signal indicative of an internal device clock. This output can be externally monitored as a safety diagnostic.

6.7.3 Dual Clock Comparators

The Dual Clock Comparator (DCC) module determines the accuracy of selectable clock sources by counting the pulses of two independent clock sources (counter 0 and counter 1). If one clock is out of spec, an error signal is generated. For example, the DCC1 can be configured to use CLK10M as the reference clock (for counter 0) and VCLK as the "clock under test" (for counter 1). This configuration allows the DCC1 to monitor the PLL output clock when VCLK is using the PLL output as its source.

An additional use of this module is to measure the frequency of a selectable clock source. For example, the reference clock is connected to Counter 0 and the signal to be measured is connected to Counter 1. Counter 0 is programmed with a start value of known time duration (measurement time) from the reference clock. Counter 1 is programmed with a maximum start value. Start both counter simultaneously. When Counter 0 decrements to zero, both counter will stop and an error signal is generated if Counter 1 does not reach zero. The frequency of the input signals can be calculated from the count value of Counter 1 and the measurement time.

6.7.3.1 Features

- Takes two different clock sources as input to two independent counter blocks.
- One of the clock sources is the known-good, or reference clock; the second clock source is the "clock under test."
- Each counter block is programmable with initial, or seed values.
- The counter blocks start counting down from their seed values at the same time; a mismatch from the expected frequency for the clock under test generates an error signal which is used to interrupt the CPU.

6.7.3.2 Mapping of DCC Clock Source Inputs

Table 6-20. DCC1 Counter 0 Clock Sources

CLOCK SOURCE[3:0]	CLOCK NAME
Others	Oscillator (OSCIN)
0x5	High-frequency LPO
0xA	Test clock (TCK)

Table 6-21. DCC1 Counter 1 Clock Sources

KEY[3:0]	CLOCK SOURCE[3:0]	CLOCK NAME
Others	–	N2HET1[31]
	0x0	Main PLL free-running clock output
	0x1	PLL #2 free-running clock output
	0x2	Low-frequency LPO
0xA	0x3	High-frequency LPO
	0x4	Reserved
	0x5	EXTCLKIN1
	0x6	EXTCLKIN2
	0x7	Reserved
	0x8 - 0xF	VCLK

Table 6-22. DCC2 Counter 0 Clock Sources

CLOCK SOURCE[3:0]	CLOCK NAME
Others	Oscillator (OSCIN)
0xA	Test clock (TCK)

Table 6-23. DCC2 Counter 1 Clock Sources

KEY[3:0]	CLOCK SOURCE[3:0]	CLOCK NAME
Others	–	N2HET2[0]
0xA	0x1	PLL2_post_ODCLK/8
	0x2	PLL2_post_ODCLK/16
	0x3 - 0x7	Reserved
	0x8 - 0xF	VCLK

6.8 Glitch Filters

Table 6-24 lists the signals with glitch filters present .

Table 6-24. Glitch Filter Timing Specifications

TERMINAL	PARAMETER		MIN	MAX	UNIT
nPORRST	$t_{f(nPORRST)}$	Filter time nPORRST terminal; pulses less than MIN will be filtered out, pulses greater than MAX will generate a reset[1]	475	2000	ns
nRST	$t_{f(nRST)}$	Filter time nRST terminal; pulses less than MIN will be filtered out, pulses greater than MAX will generate a reset	475	2000	ns
TEST	$t_{f(TEST)}$	Filter time TEST terminal; pulses less than MIN will be filtered out, pulses greater than MAX will pass through	475	2000	ns

(1) The glitch filter design on the nPORRST signal is designed such that no size pulse will reset any part of the microcontroller (flash pump, I/O pins, forth) without also generating a valid reset signal to the CPU.

6.9 Device Memory Map

6.9.1 Memory Map Diagram

Figure 6-9 shows the device memory map.

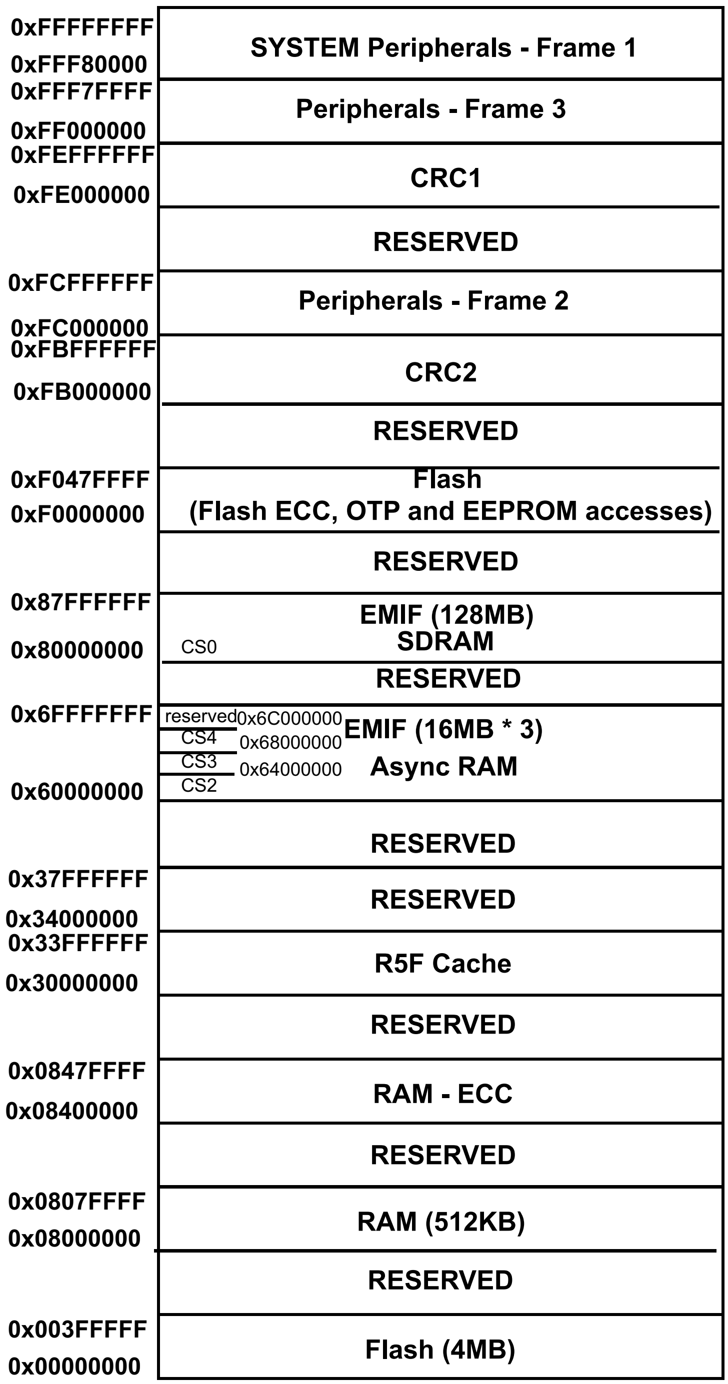


Figure 6-9. Memory Map

6.9.2 Memory Map Table

Table 6-25. Module Registers / Memories Memory Map

TARGET NAME	MEMORY SELECT	ADDRESS RANGE START	ADDRESS RANGE END	FRAME SIZE	ACTUAL SIZE	RESPONSE FOR ACCESS TO UNIMPLEMENTED LOCATIONS IN FRAME
Level 2 Memories						
Level 2 Flash Data Space		0x0000_0000	0x003F_FFFF	4MB	4MB	Abort
Level 2 RAM		0x0800_0000	0x083F_FFFF	4MB	512KB	Abort
Level 2 RAM ECC		0x0840_0000	0x087F_FFFF	4MB	512KB	
Accelerator Coherency Port						
Accelerator Coherency Port		0x0800_0000	0x087F_FFFF	8MB	512KB	Abort
Level 1 Cache Memories						
Cortex-R5F Data Cache Memory		0x3000_0000	0x30FF_FFFF	16MB	32KB	Abort
Cortex-R5F Instruction Cache Memory		0x3100_0000	0x31FF_FFFF	16MB	32KB	
External Memory Accesses						
EMIF Chip Select 2 (asynchronous)		0x6000_0000	0x63FF_FFFF	64MB	16MB	Access to "Reserved" space will generate Abort
EMIF Chip Select 3 (asynchronous)		0x6400_0000	0x67FF_FFFF	64MB	16MB	
EMIF Chip Select 4 (asynchronous)		0x6800_0000	0x6BFF_FFFF	64MB	16MB	
EMIF Chip Select 0 (synchronous)		0x8000_0000	0x87FF_FFFF	128MB	128MB	
Flash OTP, ECC, EEPROM Bank						
Customer OTP, Bank0		0xF000_0000	0xF000_1FFF	8KB	4KB	Abort
Customer OTP, Bank1		0xF000_2000	0xF000_3FFF	8KB	4KB	
Customer OTP, EEPROM Bank		0xF000_E000	0xF000_FFFF	8KB	1KB	
Customer OTP-ECC, Bank0		0xF004_0000	0xF004_03FF	1KB	512B	
Customer OTP-ECC, Bank1		0xF004_0400	0xF004_07FF	1KB	512B	
Customer OTP-ECC, EEPROM Bank		0xF004_1C00	0xF004_1FFF	1KB	128B	
TI OTP, Bank0		0xF008_0000	0xF008_1FFF	8KB	4KB	
TI OTP, Bank1		0xF008_2000	0xF008_3FFF	8KB	4KB	
TI OTP, EEPROM Bank		0xF008_E000	0xF008_FFFF	8KB	1KB	
TI OTP-ECC, Bank0		0xF00C_0000	0xF00C_03FF	1KB	512B	
TI OTP-ECC, Bank1		0xF00C_0400	0xF00C_07FF	1KB	512B	Abort
TI OTP-ECC, EEPROM Bank		0xF00C_1C00	0xF00C_1FFF	1KB	128B	
EEPROM Bank-ECC		0xF010_0000	0xF01F_FFFF	1MB	16KB	
EEPROM Bank		0xF020_0000	0xF03F_FFFF	2MB	128KB	
Flash Data Space ECC		0xF040_0000	0xF05F_FFFF	2MB	512KB	
Interconnect SDC MMR						
Interconnect SDC MMR		0xFA00_0000	0xFAFF_FFFF	16MB	16MB	
Registers/Memories under PCR2 (Peripheral Segment 2)						
CPPI Memory Slave (Ethernet RAM)	PCS[41]	0xFC52_0000	0xFC52_1FFF	8KB	8KB	Abort
CPGMAC Slave (Ethernet Slave)	PS[30]-PS[31]	0xFCF7_8000	0xFCF7_87FF	2KB	2KB	No Error
CPGMACSS Wrapper (Ethernet Wrapper)	PS[29]	0xFCF7_8800	0xFCF7_88FF	256B	256B	No Error
Ethernet MDIO Interface	PS[29]	0xFCF7_8900	0xFCF7_89FF	256B	256B	No Error

Table 6-25. Module Registers / Memories Memory Map (continued)

TARGET NAME	MEMORY SELECT	ADDRESS RANGE START	ADDRESS RANGE END	FRAME SIZE	ACTUAL SIZE	RESPONSE FOR ACCESS TO UNIMPLEMENTED LOCATIONS IN FRAME
ePWM1	PS[28]	0xFCF7_8C00	0xFCF7_8CFF	256B	256B	Abort
ePWM2		0xFCF7_8D00	0xFCF7_8DFF	256B	256B	Abort
ePWM3		0xFCF7_8E00	0xFCF7_8EFF	256B	256B	Abort
ePWM4		0xFCF7_8F00	0xFCF7_8FFF	256B	256B	Abort
ePWM5	PS[27]	0xFCF7_9000	0xFCF7_90FF	256B	256B	Abort
ePWM6		0xFCF7_9100	0xFCF7_91FF	256B	256B	Abort
ePWM7		0xFCF7_9200	0xFCF7_92FF	256B	256B	Abort
eCAP1		0xFCF7_9300	0xFCF7_93FF	256B	256B	Abort
eCAP2	PS[26]	0xFCF7_9400	0xFCF7_94FF	256B	256B	Abort
eCAP3		0xFCF7_9500	0xFCF7_95FF	256B	256B	Abort
eCAP4		0xFCF7_9600	0xFCF7_96FF	256B	256B	Abort
eCAP5		0xFCF7_9700	0xFCF7_97FF	256B	256B	Abort
eCAP6	PS[25]	0xFCF7_9800	0xFCF7_98FF	256B	256B	Abort
eQEP1		0xFCF7_9900	0xFCF7_99FF	256B	256B	Abort
eQEP2		0xFCF7_9A00	0xFCF7_9AFF	256B	256B	Abort
PCR2 registers	PPSE[4]–PPSE[5]	0xFCFF_1000	0xFCFF_17FF	2KB	2KB	Reads return zeros, writes have no effect
NMPU (EMAC)	PPSE[6]	0xFCFF_1800	0xFCFF_18FF	512B	512B	Abort
EMIF Registers	PPS[2]	0xFCFF_E800	0xFCFF_E8FF	256B	256B	Abort
Cyclic Redundancy Checker (CRC) Module Register Frame						
CRC1		0xFE00_0000	0xFEFF_FFFF	16MB	512KB	Accesses above 0xFE000200 generate abort.
CRC2		0xFB00_0000	0xFBFF_FFFF	16MB	512KB	Accesses above 0xFB000200 generate abort.

Table 6-25. Module Registers / Memories Memory Map (continued)

TARGET NAME	MEMORY SELECT	ADDRESS RANGE START	ADDRESS RANGE END	FRAME SIZE	ACTUAL SIZE	RESPONSE FOR ACCESS TO UNIMPLEMENTED LOCATIONS IN FRAME
Memories under User PCR3 (Peripheral Segment 3)						
MIBSPI5 RAM	PCS[5]	0xFF0A_0000	0xFF0B_FFFF	128KB	2KB	Abort for accesses above 2KB
MIBSPI4 RAM	PCS[3]	0xFF06_0000	0xFF07_FFFF	128KB	2KB	Abort for accesses above 2KB
MIBSPI3 RAM	PCS[6]	0xFF0C_0000	0xFF0D_FFFF	128KB	2KB	Abort for accesses above 2KB
MIBSPI2 RAM	PCS[4]	0xFF08_0000	0xFF09_FFFF	128KB	2KB	Abort for accesses above 2KB
MIBSPI1 RAM	PCS[7]	0xFF0E_0000	0xFF0F_FFFF	128KB	4KB	Abort for accesses above 4KB
DCAN4 RAM	PCS[12]	0xFF18_0000	0xFF19_FFFF	128KB	8KB	Abort generated for accesses beyond offset 0x2000
DCAN3 RAM	PCS[13]	0xFF1A_0000	0xFF1B_FFFF	128KB	8KB	Abort generated for accesses beyond offset 0x2000
DCAN2 RAM	PCS[14]	0xFF1C_0000	0xFF1D_FFFF	128KB	8KB	Abort generated for accesses beyond offset 0x2000
DCAN1 RAM	PCS[15]	0xFF1E_0000	0xFF1F_FFFF	128KB	8KB	Abort generated for accesses beyond offset 0x2000.
MIBADC2 RAM	PCS[29]	0xFF3A_0000	0xFF3B_FFFF	128KB	8KB	Wrap around for accesses to unimplemented address offsets lower than 0x1FFF.
MIBADC1 RAM	PCS[31]	0xFF3E_0000	0xFF3F_FFFF	128KB	8KB	Wrap around for accesses to unimplemented address offsets lower than 0x1FFF.
MIBADC1 Look-UP Table					384 bytes	Look-Up Table for ADC1 wrapper. Starts at address offset 0x2000 and ends at address offset 0x217F. Wrap around for accesses between offsets 0x0180 and 0x3FFF. Abort generation for accesses beyond offset 0x4000.
NHET2 RAM	PCS[34]	0xFF44_0000	0xFF45_FFFF	128KB	16KB	Wrap around for accesses to unimplemented address offsets lower than 0x3FFF. Abort generated for accesses beyond 0x3FFF.
NHET1 RAM	PCS[35]	0xFF46_0000	0xFF47_FFFF	128KB	16KB	Wrap around for accesses to unimplemented address offsets lower than 0x3FFF. Abort generated for accesses beyond 0x3FFF.
HET TU2 RAM	PCS[38]	0xFF4C_0000	0xFF4D_FFFF	128KB	1KB	Abort
HET TU1 RAM	PCS[39]	0xFF4E_0000	0xFF4F_FFFF	128KB	1KB	Abort
FlexRay TU RAM	PCS[40]	0xFF50_0000	0xFF51_FFFF	128KB	1KB	Abort

Table 6-25. Module Registers / Memories Memory Map (continued)

TARGET NAME	MEMORY SELECT	ADDRESS RANGE		FRAME SIZE	ACTUAL SIZE	RESPONSE FOR ACCESS TO UNIMPLEMENTED LOCATIONS IN FRAME
		START	END			
CoreSight Debug Components						
CoreSight Debug ROM	CSCS[0]	0xFFA0_0000	0xFFA0_0FFF	4KB	4KB	Reads return zeros, writes have no effect
Cortex-R5F Debug	CSCS[1]	0xFFA0_1000	0xFFA0_1FFF	4KB	4KB	Reads return zeros, writes have no effect
ETM-R5	CSCS[2]	0xFFA0_2000	0xFFA0_2FFF	4KB	4KB	Reads return zeros, writes have no effect
CoreSight TPIU	CSCS[3]	0xFFA0_3000	0xFFA0_3FFF	4KB	4KB	Reads return zeros, writes have no effect
POM	CSCS[4]	0xFFA0_4000	0xFFA0_4FFF	4KB	4KB	Reads return zeros, writes have no effect
CTI1	CSCS[7]	0xFFA0_7000	0xFFA0_7FFF	4KB	4KB	Reads return zeros, writes have no effect
CTI3	CSCS[9]	0xFFA0_9000	0xFFA0_9FFF	4KB	4KB	Reads return zeros, writes have no effect
CTI4	CSCS[10]	0xFFA0_A000	0xFFA0_AFFF	4KB	4KB	Reads return zeros, writes have no effect
CSTF	CSCS[11]	0xFFA0_B000	0xFFA0_BFFF	4KB	4KB	Reads return zeros, writes have no effect
Registers under PCR3 (Peripheral Segment 3)						
PCR3 registers	PS[31:30]	0xFFF7_8000	0xFFF7_87FF	2KB	2KB	Reads return zeros, writes have no effect
FTU	PS[23]	0xFFF7_A000	0xFFF7_A1FF	512B	512B	Reads return zeros, writes have no effect
HTU1	PS[22]	0xFFF7_A400	0xFFF7_A4FF	256B	256B	Abort
HTU2	PS[22]	0xFFF7_A500	0xFFF7_A5FF	256B	256B	Abort
NHET1	PS[17]	0xFFF7_B800	0xFFF7_B8FF	256B	256B	Reads return zeros, writes have no effect
NHET2	PS[17]	0xFFF7_B900	0xFFF7_B9FF	256B	256B	Reads return zeros, writes have no effect
GIO	PS[16]	0xFFF7_BC00	0xFFF7_BCFF	256B	256B	Reads return zeros, writes have no effect
MIBADC1	PS[15]	0xFFF7_C000	0xFFF7_C1FF	512B	512B	Reads return zeros, writes have no effect
MIBADC2	PS[15]	0xFFF7_C200	0xFFF7_C3FF	512B	512B	Reads return zeros, writes have no effect
FlexRay	PS[12]+PS[13]	0xFFF7_C800	0xFFF7_CFFF	2KB	2KB	Reads return zeros, writes have no effect
I2C1	PS[10]	0xFFF7_D400	0xFFF7_D4FF	256B	256B	Reads return zeros, writes have no effect
I2C2	PS[10]	0xFFF7_D500	0xFFF7_D5FF	256B	256B	Reads return zeros, writes have no effect
DCAN1	PS[8]	0xFFF7_DC00	0xFFF7_DDFF	512B	512B	Reads return zeros, writes have no effect
DCAN2	PS[8]	0xFFF7_DE00	0xFFF7_DFFF	512B	512B	Reads return zeros, writes have no effect
DCAN3	PS[7]	0xFFF7_E000	0xFFF7_E1FF	512B	512B	Reads return zeros, writes have no effect
DCAN4	PS[7]	0xFFF7_E200	0xFFF7_E3FF	512B	512B	Reads return zeros, writes have no effect
LIN1	PS[6]	0xFFF7_E400	0xFFF7_E4FF	256B	256B	Reads return zeros, writes have no effect
SCI3	PS[6]	0xFFF7_E500	0xFFF7_E5FF	256B	256B	Reads return zeros, writes have no effect
LIN2	PS[6]	0xFFF7_E600	0xFFF7_E6FF	256B	256B	Reads return zeros, writes have no effect
SCI4	PS[6]	0xFFF7_E700	0xFFF7_E7FF	256B	256B	Reads return zeros, writes have no effect

Table 6-25. Module Registers / Memories Memory Map (continued)

TARGET NAME	MEMORY SELECT	ADDRESS RANGE		FRAME SIZE	ACTUAL SIZE	RESPONSE FOR ACCESS TO UNIMPLEMENTED LOCATIONS IN FRAME
		START	END			
MibSPI1	PS[2]	0xFFF7_F400	0xFFF7_F5FF	512B	512B	Reads return zeros, writes have no effect
MibSPI2	PS[2]	0xFFF7_F600	0xFFF7_F7FF	512B	512B	Reads return zeros, writes have no effect
MibSPI3	PS[1]	0xFFF7_F800	0xFFF7_F9FF	512B	512B	Reads return zeros, writes have no effect
MibSPI4	PS[1]	0xFFF7_FA00	0xFFF7_FBFF	512B	512B	Reads return zeros, writes have no effect
MibSPI5	PS[0]	0xFFF7_FC00	0xFFF7_FDFF	512B	512B	Reads return zeros, writes have no effect
System Modules Control Registers and Memories under PCR1 (Peripheral Segment 1)						
DMA RAM	PPCS[0]	0xFFF8_0000	0xFFF8_0FFF	4KB	4KB	Abort
VIM RAM	PPCS[2]	0xFFF8_2000	0xFFF8_2FFF	4KB	4KB	Wrap around for accesses to unimplemented address offsets lower than 0x2FFF.
RTP RAM	PPCS[3]	0xFFF8_3000	0xFFF8_3FFF	4KB	4KB	Abort
Flash Wrapper	PPCS[7]	0xFFF8_7000	0xFFF8_7FFF	4KB	4KB	Abort
eFuse Farm Controller	PPCS[12]	0xFFF8_C000	0xFFF8_CFFF	4KB	4KB	Abort
Power Domain Control (PMM)	PPSE[0]	0xFFFF_0000	0xFFFF_01FF	512B	512B	Abort
FMTM Note: This module is only used by TI during test	PPSE[1]	0xFFFF_0400	0xFFFF_05FF	512B	512B	Reads return zeros, writes have no effect
STC2 (NHET1/2)	PPSE[2]	0xFFFF_0800	0xFFFF_08FF	256B	256B	Reads return zeros, writes have no effect
SCM	PPSE[2]	0xFFFF_0A00	0xFFFF_0AFF	256B	256B	Abort
EPC	PPSE[3]	0xFFFF_0C00	0xFFFF_0FFF	1KB	1KB	Abort
PCR1 registers	PPSE[4]–PPSE[5]	0xFFFF_1000	0xFFFF_17FF	2KB	2KB	Reads return zeros, writes have no effect
NMPU (PS_SCR_S)	PPSE[6]	0xFFFF_1800	0xFFFF_19FF	512B	512B	Abort
NMPU (DMA Port A)	PPSE[6]	0xFFFF_1A00	0xFFFF_1BFF	512B	512B	Abort
Pin Mux Control (IOMM)	PPSE[7]	0xFFFF_1C00	0xFFFF_1FFF	2KB	1KB	Reads return zeros, writes have no effect
System Module - Frame 2 (see the TRM SPNU563)	PPS[0]	0xFFFF_E100	0xFFFF_E1FF	256B	256B	Reads return zeros, writes have no effect
PBIST	PPS[1]	0xFFFF_E400	0xFFFF_E5FF	512B	512B	Reads return zeros, writes have no effect
STC1 (Cortex-R5F)	PPS[1]	0xFFFF_E600	0xFFFF_E6FF	256B	256B	Reads return zeros, writes have no effect
DCC1	PPS[3]	0xFFFF_EC00	0xFFFF_ECFF	256B	256B	Reads return zeros, writes have no effect
DMA	PPS[4]	0xFFFF_F000	0xFFFF_F3FF	1KB	1KB	Abort
DCC2	PPS[5]	0xFFFF_F400	0xFFFF_F4FF	256B	256B	Reads return zeros, writes have no effect
ESM register	PPS[5]	0xFFFF_F500	0xFFFF_F5FF	256B	256B	Reads return zeros, writes have no effect
CCM-R5F	PPS[5]	0xFFFF_F600	0xFFFF_F6FF	256B	256B	Reads return zeros, writes have no effect
DMM	PPS[5]	0xFFFF_F700	0xFFFF_F7FF	256B	256B	Reads return zeros, writes have no effect
L2RAMW	PPS[6]	0xFFFF_F900	0xFFFF_F9FF	256B	256B	Abort
RTP	PPS[6]	0xFFFF_FA00	0xFFFF_FAFF	256B	256B	Reads return zeros, writes have no effect
RTI + DWWD	PPS[7]	0xFFFF_FC00	0xFFFF_FCFF	256B	256B	Reads return zeros, writes have no effect

Table 6-25. Module Registers / Memories Memory Map (continued)

TARGET NAME	MEMORY SELECT	ADDRESS RANGE		FRAME SIZE	ACTUAL SIZE	RESPONSE FOR ACCESS TO UNIMPLEMENTED LOCATIONS IN FRAME
		START	END			
VIM	PPS[7]	0xFFFF_FD00	0xFFFF_FEFF	512B	512B	Reads return zeros, writes have no effect
System Module - Frame 1 (see the TRM SPNU563)	PPS[7]	0xFFFF_FF00	0xFFFF_FFFF	256B	256B	Reads return zeros, writes have no effect

6.9.3 Special Consideration for CPU Access Errors Resulting in Imprecise Aborts

Any CPU write access to a Normal or Device type memory, which generates a fault, will generate an imprecise abort. The imprecise abort exception is disabled by default and must be enabled for the CPU to handle this exception. The imprecise abort handling is enabled by clearing the "A" bit in the CPU program status register (CPSR).

6.9.4 Master/Slave Access Privileges

Table 6-26 and Table 6-27 list the access permissions for each bus master on the device. A bus master is a module that can initiate a read or a write transaction on the device.

Each slave module on either the CPU Interconnect Subsystem or the Peripheral Interconnect Subsystem is listed in Table 6-27. Allowed indicates that the module listed in the MASTERS column can access that slave module.

Table 6-26. Bus Master / Slave Access Matrix for CPU Interconnect Subsystem

MASTERS	SLAVES ON CPU INTERCONNECT SUBSYSTEM				
	L2 Flash OTP, ECC, Bank 7 (EEPROM)	L2 FLASH	L2 SRAM	CACHE MEMORY	EMIF
CPU Read	Allowed	Allowed	Allowed	Allowed	Allowed
CPU Write	Not allowed	Not allowed	Allowed	Allowed	Allowed
DMA PortA	Allowed	Allowed	Allowed	Not allowed	Allowed
POM	Not allowed	Not allowed	Allowed	Not allowed	Allowed
PS_SCR_M	Allowed	Allowed	Allowed	Not allowed	Allowed
ACP_M	Not allowed	Not Allowed	Allowed	Not allowed	Not allowed

Table 6-27. Bus Master / Slave Access Matrix for Peripheral Interconnect Subsystem

MASTER ID TO PCRx	MASTERS	SLAVES ON PERIPHERAL INTERCONNECT SUBSYSTEM			
		CRC1/CRC2	Resources Under PCR2 and PCR3	Resources Under PCR1	CPU Interconnect Subsystem SDC MMR Port (see Section 6.9.6)
0	CPU Read	Allowed	Allowed	Allowed	Allowed
	CPU Write	Allowed	Allowed	Allowed	Allowed
1	Reserved	–	–	–	–
2	DMA PortB	Allowed	Allowed	Allowed	Not allowed
3	HTU1	Not allowed	Not allowed	Not allowed	Not allowed
4	HTU2	Not allowed	Not allowed	Not allowed	Not allowed
5	FTU	Not allowed	Not allowed	Not allowed	Not allowed
7	DMM	Allowed	Allowed	Allowed	Allowed
9	DAP	Allowed	Allowed	Allowed	Allowed
10	EMAC	Not allowed	Allowed	Not allowed	Not allowed

6.9.4.1 Special Notes on Accesses to Certain Slaves

By design only the CPU and debugger can have privileged write access to peripherals under the PCR1 segment. The other masters can only read from these registers.

The master-id filtering check is implemented inside each PCR module of each peripheral segment and can be used to block certain masters from write accesses to certain peripherals. An unauthorized master write access detected by the PCR will result in the transaction being discarded and an error being generated to the ESM module.

The device contains dedicated logic to generate a bus error response on any access to a module that is in a power domain that has been turned off.

6.9.5 MasterID to PCRx

The MasterID associated with each master port on the Peripheral Interconnect Subsystem contains a 4-bit value. The MasterID is passed along with the address and control signals to three PCR modules. PCR decodes the address and control signals to select the peripheral. In addition, it decodes this 4-bit MasterID value to perform memory protection. With 4-bit of MasterID, it allows the PCR to distinguish among 16 different masters to allow or disallow access to a given peripheral. Associated with each peripheral a 16-bit MasterID access protection register is defined. Each bit grants or denies the permission of the corresponding binary coded decimal MasterID. For example, if bit 5 of the access permission register is set, it grants MasterID 5 to access the peripheral. If bit 7 is clear, it denies MasterID 7 to access the peripheral. Figure 6-10 shows the MasterID filtering scheme. Table 6-27 lists the MasterID of each master, which can access the PCRx.

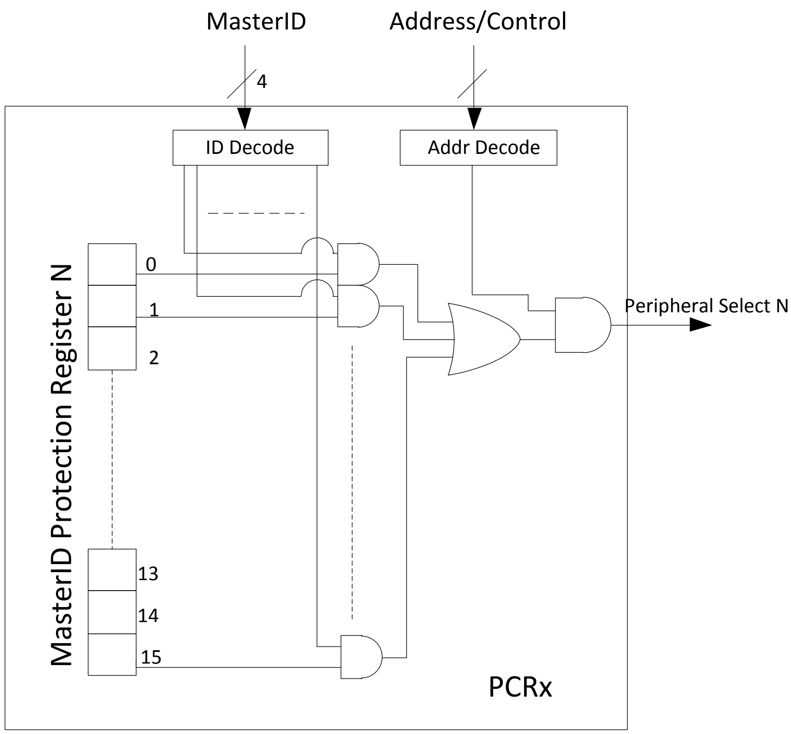


Figure 6-10. PCR MasterID Filtering

6.9.6 CPU Interconnect Subsystem SDC MMR Port

The CPU Interconnect Subsystem SDC MMR Port is a special slave to the Peripheral Interconnect Subsystem. It is memory mapped at starting address of 0xFA00_0000. Various status registers pertaining to the diagnostics of the CPU Interconnect Subsystem can be access through this slave port. The CPU Interconnect Subsystem contains built-in hardware diagnostic checkers which will constantly watch transactions flowing through the interconnect. There is a checker for each master and slave attached to the CPU Interconnect Subsystem. The checker checks the expected behavior against the generated behavior by the interconnect. For example, if the CPU issues a burst read request to the flash, the checker will ensure that the expected behavior is indeed a burst read request to the proper slave module. If the interconnects generates a transaction which is not a read, or not a burst or not to the flash as the destination, then the checker will flag it one of the registers. The detected error will also be signaled to the ESM module. Refer to the Interconnect chapter of the TRM SPNU563 for details on the registers.

Table 6-28. CPU Interconnect Subsystem SDC Register Bit Field Mapping

Register name	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	Remark
ERR_GENERIC_PARITY	PS_SCR_M	POM	DMA_PORTA	Reserved	CPU AXI-M	ACP-M	Reserved	• Each bit indicates the transaction processing block inside the interconnect corresponding to the master that is detected by the interconnect checker to have a fault. • error related to parity mismatch in the incoming address
ERR_UNEXPECTED_TRANS	PS_SCR_M	POM	DMA_PORTA	Reserved	CPU AXI-M	ACP-M	Reserved	• error related to unexpected transaction sent by the master
ERR_TRANS_ID	PS_SCR_M	POM	DMA_PORTA	Reserved	CPU AXI-M	ACP-M	Reserved	• error related to mismatch on the transaction ID
ERR_TRANS_SIGNATURE	PS_SCR_M	POM	DMA_PORTA	Reserved	CPU AXI-M	ACP-M	Reserved	• error related to mismatch on the transaction signature
ERR_TRANS_TYPE	PS_SCR_M	POM	DMA_PORTA	Reserved	CPU AXI-M	ACP-M	Reserved	• error related to mismatch on the transaction type
ERR_USER_PARITY	PS_SCR_M	POM	DMA_PORTA	Reserved	CPU AXI-M	ACP-M	Reserved	• error related to mismatch on the parity
SERR_UNEXPECTED_MID	L2 RAM Wrapper	L2 Flash Wrapper Port A	L2 Flash Wrapper Port B	EMIF	Reserved	CPU AXi-S	ACP-S	• Each bit indicates the transaction processing block inside the interconnect corresponding to the slave that is detected by the interconnect checker to have a fault. • error related to mismatch on the master ID
SERR_ADDR_DECODE	L2 RAM Wrapper	L2 Flash Wrapper Port A	L2 Flash Wrapper Port B	EMIF	Reserved	CPU AXi-S	ACP-S	• error related to mismatch on the most significant address bits
SERR_USER_PARITY	L2 RAM Wrapper	L2 Flash Wrapper Port A	L2 Flash Wrapper Port B	EMIF	Reserved	CPU AXi-S	ACP-S	• error related to mismatch on the parity of the most significant address bits

6.9.7 Parameter Overlay Module (POM) Considerations

The Parameter Overlay Module (POM) is implemented as part of the L2FMC module. It is used to redirect flash memory accesses to external memory interfaces or internal SRAM. The POM has an OCP master port to redirect accesses. The POM MMRs are located in a separate block and read/writes will happen through the Debug APB port on the L2FMC. The POM master port is capable of read accesses only. Inside the CPU Subsystem SCR, the POM master port is connected to both the L2RAMW and EMIF slaves. The primary roles of the POM are:

- The POM snoops the access on the two flash slave ports to determine if access should be remapped or not. It supports 32 regions among the two slave ports.
- If access is to be remapped, then the POM kills the access to the flash bank, and instead redirects the access through its own master.
- Upon obtaining response, the POM populates the response FIFO of the respective port so that the response is delivered back to the original requester.
- The access is unaffected if the request is not mapped to any region, or if the POM is disabled.
- The POM does not add any latency to the flash access when it is turned off.
- The POM does not add any latency to the remapped access (except the latency, if any, associated with the getting the response from the an alternate slave)

6.10 Flash Memory

6.10.1 Flash Memory Configuration

Flash Bank: A separate block of logic consisting of 1 to 16 sectors. Each flash bank normally has a customer-OTP and a TI-OTP area. These flash sectors share input/output buffers, data paths, sense amplifiers, and control logic.

Flash Sector: A contiguous region of flash memory which must be erased simultaneously due to physical construction constraints.

Flash Pump: A charge pump which generates all the voltages required for reading, programming, or erasing the flash banks.

Flash Module: Interface circuitry required between the host CPU and the flash banks and pump module.

Table 6-29. Flash Memory Banks and Sectors

MEMORY ARRAYS (OR BANKS)	SECTOR NO.	SEGMENT	LOW ADDRESS	HIGH ADDRESS
BANK0 (2.0MB)	0	16KB	0x0000_0000	0x0000_3FFF
	1	16KB	0x0000_4000	0x0000_7FFF
	2	16KB	0x0000_8000	0x0000_BFFF
	3	16KB	0x0000_C000	0x0000_FFFF
	4	16KB	0x0001_0000	0x0001_3FFF
	5	16KB	0x0001_4000	0x0001_7FFF
	6	32KB	0x0001_8000	0x0001_FFFF
	7	128KB	0x0002_0000	0x0003_FFFF
	8	128KB	0x0004_0000	0x0005_FFFF
	9	128KB	0x0006_0000	0x0007_FFFF
	10	256KB	0x0008_0000	0x000B_FFFF
	11	256KB	0x000C_0000	0x000F_FFFF
	12	256KB	0x0010_0000	0x0013_FFFF
	13	256KB	0x0014_0000	0x0017_FFFF
	14	256KB	0x0018_0000	0x001B_FFFF
	15	256KB	0x001C_0000	0x001F_FFFF
BANK1 (2.0MB)	0	128KB	0x0020_0000	0x0021_FFFF
	1	128KB	0x0022_0000	0x0023_FFFF
	2	128KB	0x0024_0000	0x0025_FFFF
	3	128KB	0x0026_0000	0x0027_FFFF
	4	128KB	0x0028_0000	0x0029_FFFF
	5	128KB	0x002A_0000	0x002B_FFFF
	6	128KB	0x002C_0000	0x002D_FFFF
	7	128KB	0x002E_0000	0x002F_FFFF
	8	128KB	0x0030_0000	0x0031_FFFF
	9	128KB	0x0032_0000	0x0033_FFFF
	10	128KB	0x0034_0000	0x0035_FFFF
	11	128KB	0x0036_0000	0x0037_FFFF
	12	128KB	0x0038_0000	0x0039_FFFF
	13	128KB	0x003A_0000	0x003B_FFFF
	14	128KB	0x003C_0000	0x003D_FFFF
	15	128KB	0x003E_0000	0x003F_FFFF

Table 6-30. EEPROM Flash Bank

MEMORY ARRAYS (OR BANKS)	SECTOR NO.	SEGMENT	LOW ADDRESS	HIGH ADDRESS
BANK7 (128KB) for EEPROM emulation	0	4KB	0xF020_0000	0xF020_0FFF
	" " "	" " "	" " "	" " "
	31	4KB	0xF021_F000	0xF021_FFFF

6.10.2 Main Features of Flash Module

- Support for multiple flash banks for program and/or data storage
- Simultaneous read accesses on two banks while performing program or erase operation on any other bank
- Integrated state machines to automate flash erase and program operations
- Software interface for flash program and erase operations
- Pipelined mode operation to improve instruction access interface bandwidth
- Support for Single Error Correction Double Error Detection (SECDED) block inside Cortex-R5F CPU
- Support for a rich set of diagnostic features

6.10.3 ECC Protection for Flash Accesses

All accesses to the L2 program flash memory are protected by SECDED logic embedded inside the CPU. The flash module provides 8 bits of ECC code for 64 bits of instructions or data fetched from the flash memory. The CPU calculates the expected ECC code based on the 64 bits data received and compares it with the ECC code returned by the flash module. A single-bit error is corrected and flagged by the CPU, while a multibit error is only flagged. The CPU signals an ECC error through its Event bus. This signaling mechanism is not enabled by default and must be enabled by setting the 'X' bit of the Performance Monitor Control Register, c9.

```
MRC p15,#0,r1,c9,c12,#0     ;Enabling Event monitor states
ORR r1, r1, #0x00000010
MCR p15,#0,r1,c9,c12,#0     ;Set 4th bit ('X') of PMNC register
MRC p15,#0,r1,c9,c12,#0
```

NOTE

ECC is permanently enabled in the CPU L2 interface.

6.10.4 Flash Access Speeds

For information on flash memory access speeds and the relevant wait states required, refer to Section 5.6.

6.10.5 Flash Program and Erase Timings

6.10.5.1 Flash Program and Erase Timings for Program Flash

Table 6-31. Timing Requirements for Program Flash

			MIN	NOM	MAX	UNIT
$t_{prog(288bits)}$	Wide Word (288-bits) programming time			40	300	µs
$t_{prog(Total)}$	4.0MB programming time[(1)]	–40°C to 125°C			21.3	s
		0°C to 60°C, for first 25 cycles		5.3	10.6	s
t_{erase}	Sector/Bank erase time	–40°C to 125°C		0.3	4	s
		0°C to 60°C, for first 25 cycles			100	ms
t_{wec}	Write/erase cycles with 15-year Data Retention requirement	–40°C to 125°C			1000	cycles

(1) This programming time includes overhead of state machine, but does not include data transfer time. The programming time assumes programming 288 bits at a time at the maximum specified operating frequency.

6.10.5.2 Flash Program and Erase Timings for Data Flash

Table 6-32. Timing Requirements for Data Flash

			MIN	NOM	MAX	UNIT
$t_{prog(72bits)}$	Wide Word (72-bits) programming time			47	300	µs
$t_{prog(Total)}$	EEPROM Emulation (bank 7) 128KB programming time[(1)]	–40°C to 125°C			2.6	s
		0°C to 60°C, for first 25 cycles		775	1320	ms
EEPROM Emulation (bank 7) Sector/Bank erase time $t_{erase(bank7)}$		–40°C to 125°C		0.2	8	s
		0°C to 60°C, for first 25 cycles		14	100	ms
t_{wec}	Write/erase cycles with 15-year Data Retention requirement	–40°C to 125°C			100000	cycles

(1) This programming time includes overhead of state machine, but does not include data transfer time. The programming time assumes programming 72 bits at a time at the maximum specified operating frequency.

6.11 L2RAMW (Level 2 RAM Interface Module)

L2RAMW is the TMS570 level two RAM wrapper. Major features implemented in this device include:

- Supports 512KB of L2 SRAMs
- One 64-bit OCP interface
- Built-in ECC generation and evaluation logic
  - The ECC logic is enabled by default.
  - When enabled, automatic ECC correction on write data from masters on any write sizes (8-,16-,32-,or 64-bit)
  - Less than 64-bit write forces built in read-modify-write
  - When enabled, reads due to read-modify-write go through ECC correction before data merging with the incoming write data
- Redundant address decoding. Same address decode logic block is duplicated and compared to each other
- Data Trace
  - Support tracing of both read and write accesses through RTP module
- Auto initialization of memory banks to known values for both data and their corresponding ECC checksum

6.11.1 L2 SRAM Initialization

The entire L2 SRAM can be globally initialized by setting the corresponding bit in SYS.MSINENA register. When initialized, the memory arrays are written with all zeros for the 64-bit data and the corresponding 8-bit ECC checksum. Hardware memory initialization eliminates ECC error when the CPU reads from an un-initialized memory location which can cause an ECC error. For more information, see the device-specific Technical Reference Manual.

6.12 ECC / Parity Protection for Accesses to Peripheral RAMs

Accesses to some peripheral RAMs are protected by either odd/even parity checking or ECC checking. During a read access the parity or ECC is calculated based on the data read from the peripheral RAM and compared with the good parity or ECC value stored in the peripheral RAM for that peripheral. If any word fails the parity or ECC check, the module generates a ECC/parity error signal that is mapped to the Error Signaling Module. The module also captures the peripheral RAM address that caused the parity error.

The parity or ECC protection for peripheral RAMs is not enabled by default and must be enabled by the application. Each individual peripheral contains control registers to enable the parity or ECC protection for accesses to its RAM.

NOTE

For peripherals with parity protection the CPU read access gets the actual data from the peripheral. The application can choose to generate an interrupt whenever a peripheral RAM parity error is detected.

6.13 On-Chip SRAM Initialization and Testing

6.13.1 On-Chip SRAM Self-Test Using PBIST

6.13.1.1 Features

- Extensive instruction set to support various memory test algorithms
- ROM-based algorithms allow application to run TI production-level memory tests
- Independent testing of all on-chip SRAM

6.13.1.2 PBIST RAM Groups

Table 6-33. PBIST RAM Grouping

MEMORY	RAM GROUP	TEST CLOCK	RGS	RDS	MEM TYPE	NO. BANKS	TEST PATTERN (ALGORITHM)			
							TRIPLE READ SLOW READ	TRIPLE READ FAST READ	March 13N(1) TWO PORT (cycles)	March 13N(1) SINGLE PORT (cycles)
							ALGO MASK 0x1	ALGO MASK 0x2	ALGO MASK 0x4	ALGO MASK 0x8
PBIST_ROM	1	GCM_PBIST_ROM	1	1	ROM	1	24578	8194		
STC1_1_ROM_R5	2	GCM_PBIST_ROM	14	1	ROM	1	49154	16386		
STC1_2_ROM_R5	3	GCM_PBIST_ROM	14	2	ROM	1	49154	16386		
STC2_ROM_NHET	4	GCM_PBIST_ROM	15	1	ROM	1	46082	15362		
AWM1	5	GCM_VCLKP	2	1	2P	1			4210	
DCAN1	6	GCM_VCLKP	3	1..6	2P	2			25260	
DCAN2	7	GCM_VCLKP	4	1..6	2P	2			25260	
DMA	8	GCM_HCLK	5	1..6	2P	2			37740	
HTU1	9	GCM_VCLK2	6	1..6	2P	2			6540	
MIBSPI1	10	GCM_VCLKP	8	1..4	2P	2			66760	
MIBSPI2	11	GCM_VCLKP	9	1..4	2P	2			33480	
MIBSPI3	12	GCM_VCLKP	10	1..4	2P	2			33480	
NHET1	13	GCM_VCLK2	11	1..12	2P	4			50520	
VIM	14	GCM_VCLK	12	1..2	2P	1			16740	
Reserved	15	-	-	-	-	-			-	
RTP	16	GCM_HCLK	16	1..12	2P	4			50520	
ATB(2)	17	GCM_GCLK1	17	1..16	2P	8			133920	
AWM2	18	GCM_VCLKP	18	1	2P	1			4210	
DCAN3	19	GCM_VCLKP	19	1..6	2P	2			25260	
DCAN4	20	GCM_VCLKP	20	1..6	2P	2			25260	
HTU2	21	GCM_VCLK2	21	1..6	2P	2			6540	
MIBSPI4	22	GCM_VCLKP	22	1..4	2P	2			33480	
MIBSPI5	23	GCM_VCLKP	23	1..4	2P	2			33480	
NHET2	24	GCM_VCLK2	24	1..12	2P	4			50520	
FTU	25	GCM_VCLKP	25	1	2P	1			8370	
FRAY_INBUF_OUTBUF	26	GCM_VCLKP	26	1..8	2P	4			33680	
CPGMAC_STATE_RXADDR	27	GCM_VCLK3	27	1..3	2P	2			6390	
CPGMAC_STAT_FIFO	28	GCM_VCLK3	27	4..6	2P	3			8730	

(1) March13N is the only algorithm recommended for application testing of RAM.
(2) ATB RAM is part of the ETM module. PBIST testing of this RAM is limited to 85ºC or lower.

Table 6-33. PBIST RAM Grouping (continued)

MEMORY	RAM GROUP	TEST CLOCK	RGS	RDS	MEM TYPE	NO. BANKS	TEST PATTERN (ALGORITHM)			
							TRIPLE READ SLOW READ	TRIPLE READ FAST READ	March 13N[1] TWO PORT (cycles)	March 13N[1] SINGLE PORT (cycles)
							ALGO MASK 0x1	ALGO MASK 0x2	ALGO MASK 0x4	ALGO MASK 0x8
L2RAMW	29	GCM_HCLK	7	1	SP	4				532580
				6	SP	4				
L2RAMW	30	GCM_HCLK	32	1	SP	4				1597740
				6	SP	4				
				11	SP	4				
				16	SP	4				
				21	SP	4				
				26	SP	4				
R5_ICACHE	31	GCM_GCLK1	40	1	SP	4				166600
				6	SP	4				
				11	SP	4				
				16	SP	4				
R5_DCACHE	32	GCM_GCLK1	41	1	SP	4				299820
				6	SP	4				
				11	SP	4				
				16	SP	4				
				21	SP	4				
				26	SP	4				
Reserved	33	GCM_GCLK2	43	1	SP	4				166600
				6	SP	4				
				11	SP	4				
				16	SP	4				
Reserved	34	GCM_GCLK2	44	1	SP	4				299820
				6	SP	4				
				11	SP	4				
				16	SP	4				
				21	SP	4				
				26	SP	4				
FRAY_TRBUF_MSG RAM	35	GCM_VCLKP	26	9..11	SP	3				149910
CPGMAC_CPPI	36	GCM_VCLK3	27	7	SP	1				133170
R5_DCACHE_Dirty	37	GCM_GCLK1	42	2	SP	1				16690
Reserved	38	-	-	-	-	-				-

Several memory testing algorithms are stored in the PBIST ROM. However, TI only recommends the March13N algorithm for application testing of RAM.

The PBIST ROM clock frequency is limited to the maximum frequency of 82.5 MHz.

The PBIST ROM clock is divided down from HCLK. The divider is selected by programming the ROM_DIV field of the Memory Self-Test Global Control Register (MSTGCR) at address 0xFFFFFF58.

6.13.2 On-Chip SRAM Auto Initialization

This microcontroller allows some of the on-chip memories to be initialized through the Memory Hardware Initialization mechanism in the system module. This hardware mechanism allows an application to program the memory arrays with error detection capability to a known state based on their error detection scheme (odd/even parity or ECC).

The MINITGCR register enables the memory initialization sequence, and the MSINENA register selects the memories that are to be initialized.

For more information on these registers, see the device-specific Technical Reference Manual.

The mapping of the different on-chip memories to the specific bits of the MSINENA registers is provided in Table 6-34.

Table 6-34. Memory Initialization[(1)(2)]

CONNECTING MODULE	ADDRESS RANGE		SYS.MSINENA Register Bit #	L2RAMW.MEMINT_ENA Register Bit #[(3)]
	BASE ADDRESS	ENDING ADDRESS		
L2 SRAM	0x08000000	0x0800FFFF	0	0
L2 SRAM	0x08010000	0x0801FFFF	0	1
L2 SRAM	0x08020000	0x0802FFFF	0	2
L2 SRAM	0x08030000	0x0803FFFF	0	3
L2 SRAM	0x08040000	0x0804FFFF	0	4
L2 SRAM	0x08050000	0x0805FFFF	0	5
L2 SRAM	0x08060000	0x0806FFFF	0	6
L2 SRAM	0x08070000	0x0807FFFF	0	7
MIBSPI5 RAM[(4)]	0xFF0A0000	0xFF0BFFFF	12	n/a
MIBSPI4 RAM[(4)]	0xFF060000	0xFF07FFFF	19	n/a
MIBSPI3 RAM[(4)]	0xFF0C0000	0xFF0DFFFF	11	n/a
MIBSPI2 RAM[(4)]	0xFF080000	0xFF09FFFF	18	n/a
MIBSPI1 RAM[(4)]	0xFF0E0000	0xFF0FFFFF	7	n/a
DCAN4 RAM	0xFF180000	0xFF19FFFF	20	n/a
DCAN3 RAM	0xFF1A0000	0xFF1BFFFF	10	n/a
DCAN2 RAM	0xFF1C0000	0xFF1DFFFF	6	n/a
DCAN1 RAM	0xFF1E0000	0xFF1FFFFF	5	n/a
MIBADC2 RAM	0xFF3A0000	0xFF3BFFFF	14	n/a
MIBADC1 RAM	0xFF3E0000	0xFF3FFFFF	8	n/a
NHET2 RAM	0xFF440000	0xFF45FFFF	15	n/a
NHET1 RAM	0xFF460000	0xFF47FFFF	3	n/a
HET TU2 RAM	0xFF4C0000	0xFF4DFFFF	16	n/a
HET TU1 RAM	0xFF4E0000	0xFF4FFFFF	4	n/a
DMA RAM	0xFFF80000	0xFFF80FFF	1	n/a
VIM RAM	0xFFF82000	0xFFF82FFF	2	n/a
FlexRay TU RAM	0xFF500000	0xFF51FFFF	13	n/a

(1) If parity protection is enabled for the peripheral SRAM modules, then the parity bits will also be initialized along with the SRAM modules.
(2) If ECC protection is enabled for the CPU data RAM or peripheral SRAM modules, then the auto-initialization process also initializes the corresponding ECC space.
(3) The L2 SRAM from range 128KB to 512KB is divided into 8 memory regions. Each region has an associated control bit to enable auto-initialization.
(4) The MibSPIx modules perform an initialization of the transmit and receive RAMs as soon as the multibuffered mode is enabled. This is independent of whether the application has already initialized these RAMs using the auto-initialization method or not. The MibSPIx modules must be released from reset by writing a 1 to the SPIGCR0 registers before starting auto-initialization on the respective RAMs.

NOTE

Peripheral memories not listed in the table either do not support auto-initialization or have implemented auto-initialization controlled directly by their respective peripherals.

6.14 External Memory Interface (EMIF)

6.14.1 Features

The EMIF includes many features to enhance the ease and flexibility of connecting to external asynchronous memories or SDRAM devices. The EMIF features includes support for:

- 3 addressable chip select for asynchronous memories of up to 16MB each
- 1 addressable chip select space for SDRAMs up to 128MB
- 8 or 16-bit data bus width
- Programmable cycle timings such as setup, strobe, and hold times as well as turnaround time
- Select strobe mode
- Extended Wait mode
- Data bus parking

NOTE

The EMIF is inherently BE8, or byte invariant big endian. This device is BE32, or word invariant big endian. There is no difference when interfacing to RAM or using an 8-bit wide data bus. However, there is an impact when reading from external ROMs or interfacing to hardware registers with a 16-bit wide data bus. The EMIF can be made BE32 by connecting EMIF_DATA[7:0] to the ROM or ASIC DATA[15:8] and EMIF_DATA[15:8] to the ROM or ASIC DATA[7:0].

Alternatively, the code stored in the ROM can be linked as -be8 instead of -be32.

NOTE

For a 32-bit access on the 16-bit EMIF interface, the lower 16-bits (the EMIF_BA[1] will be low) will be put out first followed by the upper 16-bits (EMIF_BA[1] will be high).

6.14.2 Electrical and Timing Specifications

6.14.2.1 Read Timing (Asynchronous RAM)

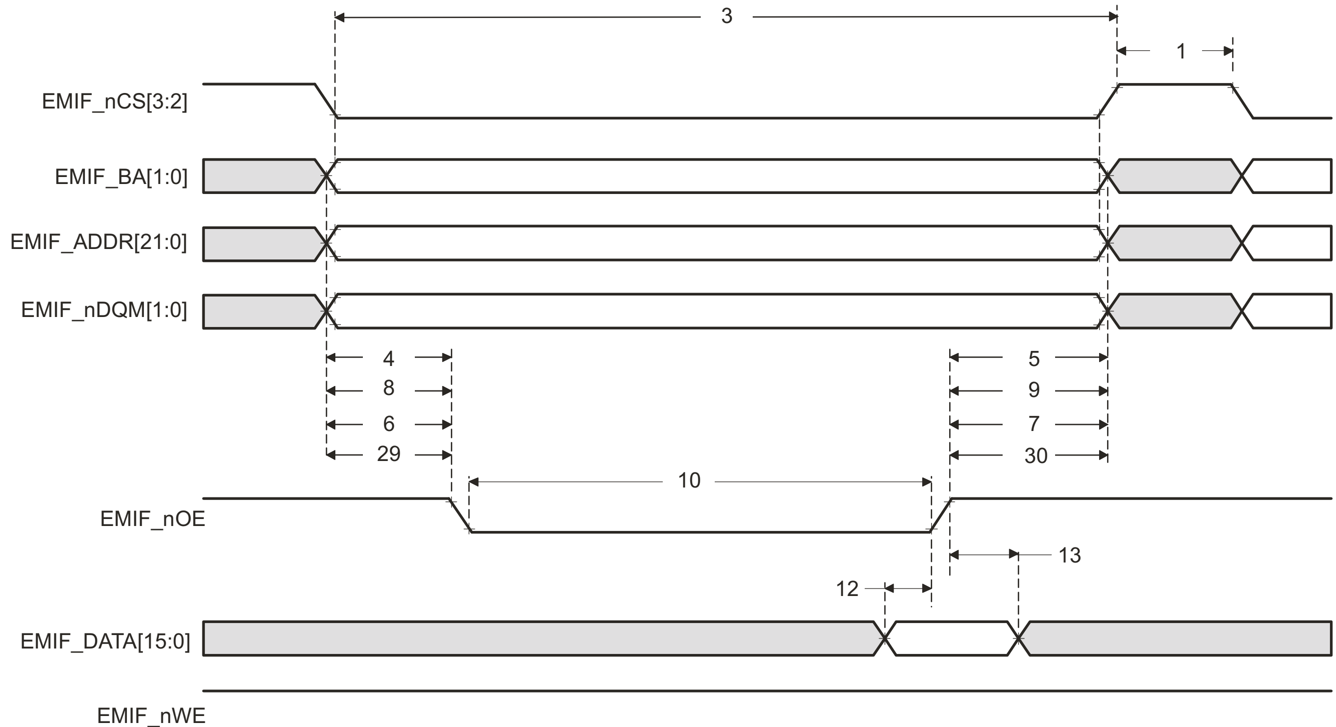


Figure 6-11. Asynchronous Memory Read Timing

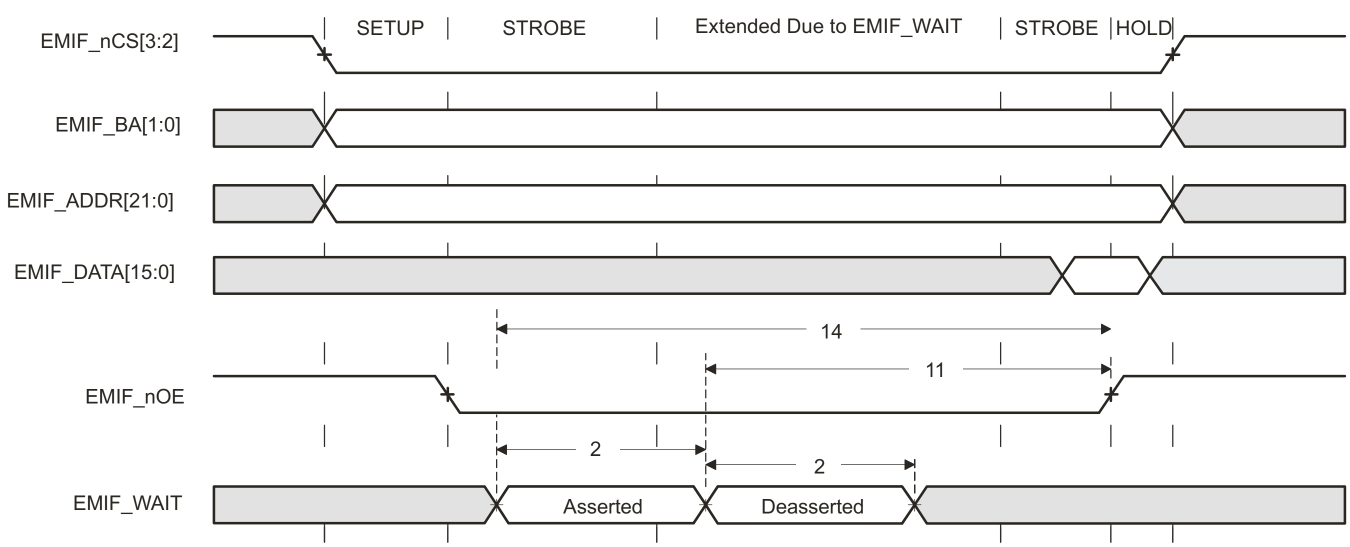


Figure 6-12. EMIFnWAIT Read Timing Requirements

6.14.2.2 Write Timing (Asynchronous RAM)

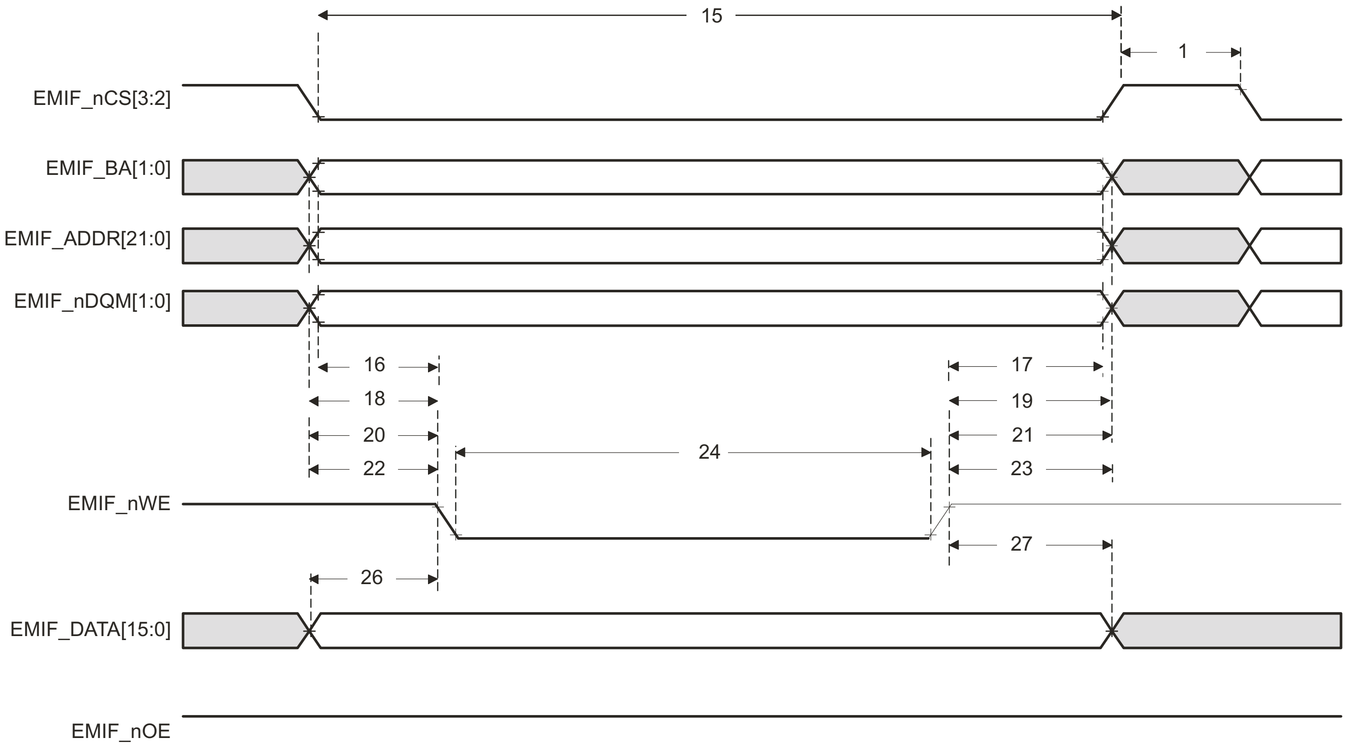


Figure 6-13. Asynchronous Memory Write Timing

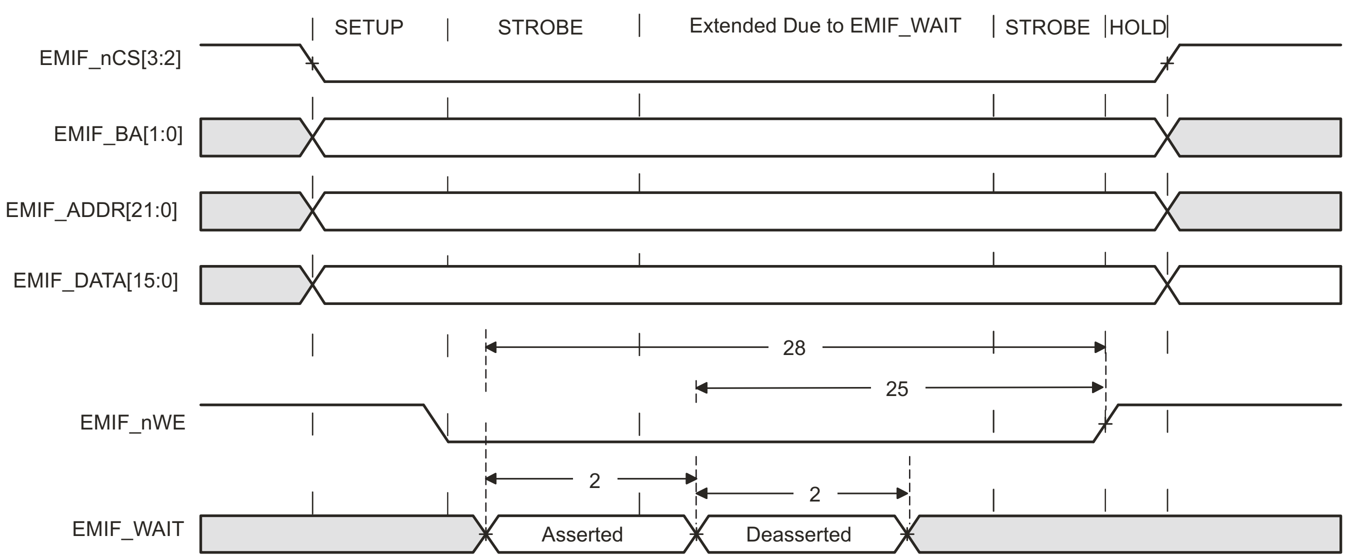


Figure 6-14. EMIFnWAIT Write Timing Requirements

6.14.2.3 EMIF Asynchronous Memory Timing

Table 6-35. EMIF Asynchronous Memory Timing Requirements(1)

NO.		MIN	NOM	MAX	UNIT
Reads and Writes					

(1) E = EMIF_CLK period in ns.

Table 6-35. EMIF Asynchronous Memory Timing Requirements[(1)] (continued)

NO.			MIN	NOM	MAX	UNIT
2	$t_{w(EM\_WAIT)}$	Pulse duration, EMIFnWAIT assertion and deassertion	2E			ns
		Reads				
12	$t_{su(EMDV-EMOEH)}$	Setup time, EMIFDATA[15:0] valid before EMIFnOE high	11			ns
13	$t_{h(EMOEH-EMDIV)}$	Hold time, EMIFDATA[15:0] valid after EMIFnOE high	0.5			ns
14	$t_{su(EMOEL-EMWAIT)}$	Setup Time, EMIFnWAIT asserted before end of Strobe Phase[(2)]	4E+14			ns
		Writes				
28	$t_{su(EMWEL-EMWAIT)}$	Setup Time, EMIFnWAIT asserted before end of Strobe Phase[(2)]	4E+14			ns

(2) Setup before end of STROBE phase (if no extended wait states are inserted) by which EMIFnWAIT must be asserted to add extended wait states. Figure 6-12 and Figure 6-14 describe EMIF transactions that include extended wait states inserted during the STROBE phase. However, cycles inserted as part of this extended wait period should not be counted; the 4E requirement is to the start of where the HOLD phase would begin if there were no extended wait cycles.

Table 6-36. EMIF Asynchronous Memory Switching Characteristics[(1)(2)(3)]

NO.	PARAMETER		MIN	TYP	MAX	UNIT
		Reads and Writes				
1	$t_{d(TURNAROUND)}$	Turn around time	(TA)*E -3	(TA)*E	(TA)*E + 3	ns
		Reads				
3	$t_{c(EMRCYCLE)}$	EMIF read cycle time (EW = 0)	(RS+RST+RH)*E-3	(RS+RST+RH)*E	(RS+RST+RH)*E + 3	ns
		EMIF read cycle time (EW = 1)	(RS+RST+RH+ EWC)*E -3	(RS+RST+RH+ EWC)*E	(RS+RST+RH+ EWC)*E + 3	ns
4	$t_{su(EMCEL-EMOEL)}$	Output setup time, EMIF_nCS[4:2] low to EMIF_nOE low (SS = 0)	(RS)*E-3	(RS)*E	(RS)*E+3	ns
		Output setup time, EMIFnCS[4:2] low to EMIF_nOE low (SS = 1)	–3	0	3	ns
5	$t_{h(EMOEH-EMCEH)}$	Output hold time, EMIF_nOE high to EMIF_nCS[4:2] high (SS = 0)	(RH)*E -4	(RH)*E	(RH)*E + 3	ns
		Output hold time, EMIF_nOE high to EMIF_nCS[4:2] high (SS = 1)	–4	0	3	ns
6	$t_{su(EMBAV-EMOEL)}$	Output setup time, EMIF_BA[1:0] valid to EMIF_nOE low	(RS)*E-3	(RS)*E	(RS)*E+3	ns
7	$t_{h(EMOEH-EMBAIV)}$	Output hold time, EMIF_nOE high to EMIF_BA[1:0] invalid	(RH)*E-4	(RH)*E	(RH)*E+3	ns
8	$t_{su(EMBAV-EMOEL)}$	Output setup time, EMIF_ADDR[21:0] valid to EMIF_nOE low	(RS)*E-3	(RS)*E	(RS)*E+3	ns
9	$t_{h(EMOEH-EMAIV)}$	Output hold time, EMIF_nOE high to EMIF_ADDR[21:0] invalid	(RH)*E-4	(RH)*E	(RH)*E+3	ns
10	$t_{w(EMOEL)}$	EMIF_nOE active low width (EW = 0)	(RST)*E-3	(RST)*E	(RST)*E+3	ns
		EMIF_nOE active low width (EW = 1)	(RST+EWC) *E-3	(RST+EWC)*E	(RST+EWC) *E+3	ns
11	$t_{d(EMWAITH-EMOEH)}$	Delay time from EMIF_nWAIT deasserted to EMIF_nOE high	3E-3	4E	4E+5	ns
29	$t_{su(EMDQMV-EMOEL)}$	Output setup time, EMIF_nDQM[1:0] valid to EMIF_nOE low	(RS)*E-5	(RS)*E	(RS)*E+3	ns
30	$t_{h(EMOEH-EMDQMIV)}$	Output hold time, EMIF_nOE high to EMIF_nDQM[1:0] invalid	(RH)*E-4	(RH)*E	(RH)*E+5	ns
		Writes				
15	$t_{c(EMWCYCLE)}$	EMIF write cycle time (EW = 0)	(WS+WST+WH)* E-3	(WS+WST+WH)*E	(WS+WST+WH)* E+3	ns
		EMIF write cycle time (EW = 1)	(WS+WST+WH+ EWC)*E -3	(WS+WST+WH+ EWC)*E	(WS+WST+WH+ EWC)*E + 3	ns

(1) TA = Turn around, RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold, MEWC = Maximum external wait cycles. These parameters are programmed through the Asynchronous Bank and Asynchronous Wait Cycle Configuration Registers. These support the following ranges of values: TA[4–1], RS[16–1], RST[64–1], RH[8–1], WS[16–1], WST[64–1], WH[8–1], and MEWC[1–256]. See the EMIF chapter of the TRM SPNU563 for more information.
(2) E = EMIF_CLK period in ns.
(3) EWC = external wait cycles determined by EMIFnWAIT input signal. EWC supports the following range of values. EWC[256–1]. Note that the maximum wait time before time-out is specified by bit field MEWC in the Asynchronous Wait Cycle Configuration Register. See the EMIF chapter of the TRM SPNU563 for more information.

Table 6-36. EMIF Asynchronous Memory Switching Characteristics[(1)(2)(3)] (continued)

NO.	PARAMETER		MIN	TYP	MAX	UNIT
16	$t_{su(EMCEL-EMWEL)}$	Output setup time, EMIF_nCS[4:2] low to EMIF_nWE low (SS = 0)	(WS)*E -3	(WS)*E	(WS)*E + 3	ns
		Output setup time, EMIF_nCS[4:2] low to EMIF_nWE low (SS = 1)	–3	0	3	ns
17	$t_{h(EMWEH-EMCEH)}$	Output hold time, EMIF_nWE high to EMIF_nCS[4:2] high (SS = 0)	(WH)*E-3	(WH)*E	(WH)*E+3	ns
		Output hold time, EMIF_nWE high to EMIF_CS[4:2] high (SS = 1)	–3	0	3	ns
18	$t_{su(EMDQMV-EMWEL)}$	Output setup time, EMIF_nDQM[1:0] valid to EMIF_nWE low	(WS)*E-3	(WS)*E	(WS)*E+3	ns
19	$t_{h(EMWEH-EMDQMIV)}$	Output hold time, EMIF_nWE high to EMIF_nDQM[1:0] invalid	(WH)*E-3	(WH)*E	(WH)*E+3	ns
20	$t_{su(EMBAV-EMWEL)}$	Output setup time, EMIF_BA[1:0] valid to EMIF_nWE low	(WS)*E-3	(WS)*E	(WS)*E+3	ns
21	$t_{h(EMWEH-EMBAIV)}$	Output hold time, EMIF_nWE high to EMIF_BA[1:0] invalid	(WH)*E-3	(WH)*E	(WH)*E+3	ns
22	$t_{su(EMAV-EMWEL)}$	Output setup time, EMIF_ADDR[21:0] valid to EMIF_nWE low	(WS)*E-3	(WS)*E	(WS)*E+3	ns
23	$t_{h(EMWEH-EMAIV)}$	Output hold time, EMIF_nWE high to EMIF_ADDR[21:0] invalid	(WH)*E-3	(WH)*E	(WH)*E+3	ns
24	$t_{w(EMWEL)}$	EMIF_nWE active low width (EW = 0)	(WST)*E-3	(WST)*E	(WST)*E+3	ns
		EMIF_nWE active low width (EW = 1)	(WST+EWC) *E-3	(WST+EWC)*E	(WST+EWC) *E+3	ns
25	$t_{d(EMWAITH-EMWEH)}$	Delay time from EMIF_nWAIT deasserted to EMIF_nWE high	3E+3	4E	4E+14	ns
26	$t_{su(EMDV-EMWEL)}$	Output setup time, EMIF_DATA[15:0] valid to EMIF_nWE low	(WS)*E-3	(WS)*E	(WS)*E+3	ns
27	$t_{h(EMWEH-EMDIV)}$	Output hold time, EMIF_nWE high to EMIF_DATA[15:0] invalid	(WH)*E-3	(WH)*E	(WH)*E+3	ns

6.14.2.4 Read Timing (Synchronous RAM)

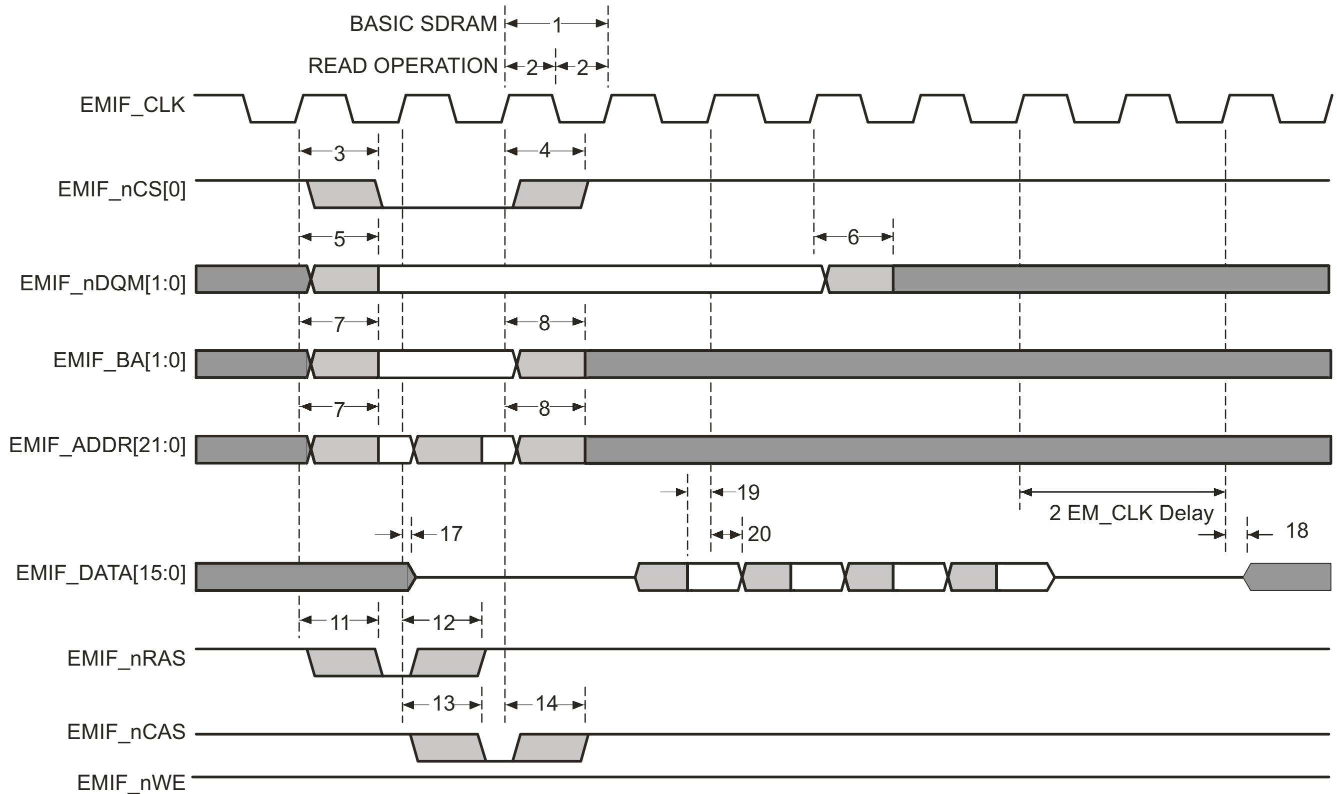


Figure 6-15. Basic SDRAM Read Operation

6.14.2.5 Write Timing (Synchronous RAM)

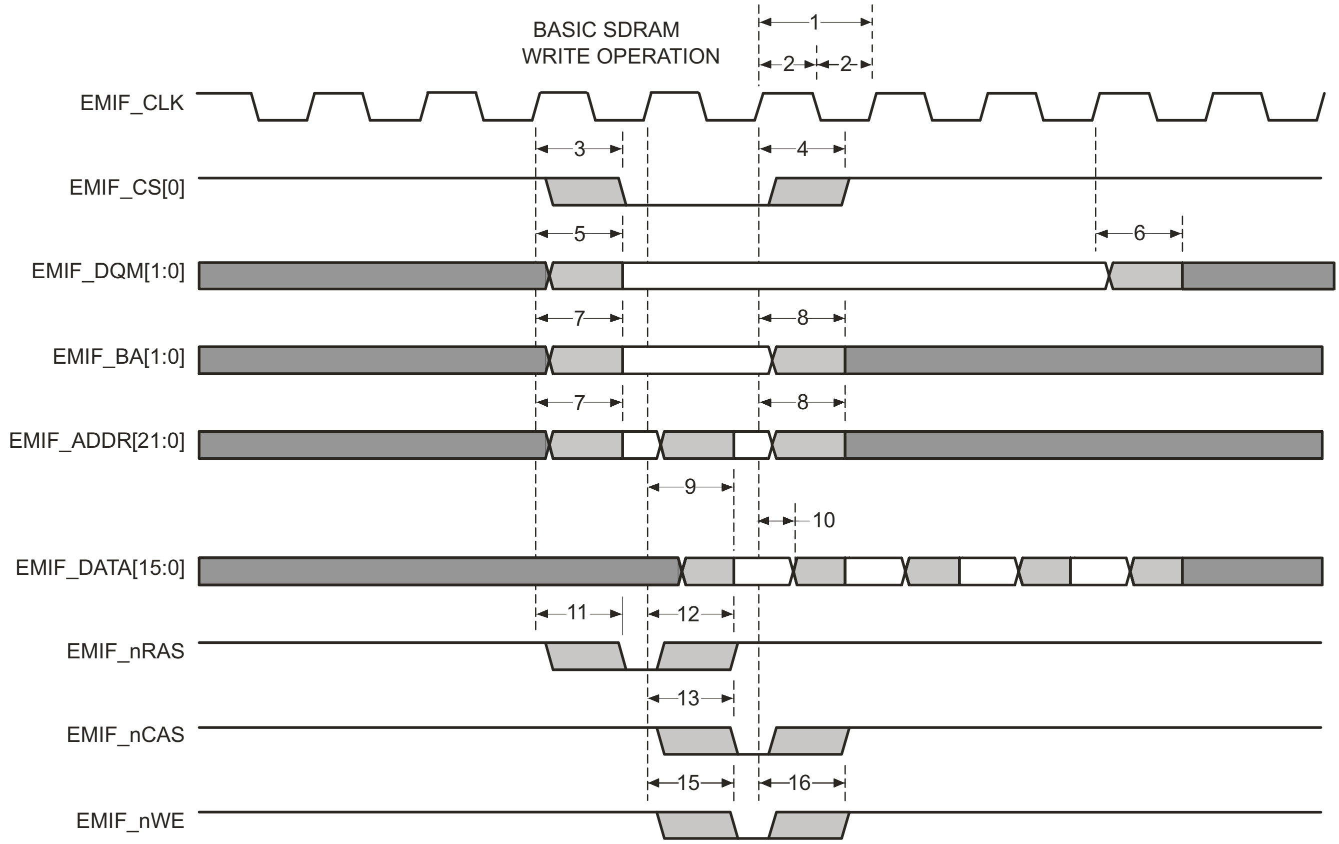


Figure 6-16. Basic SDRAM Write Operation

EMIF Synchronous Memory Timing

Table 6-37. EMIF Synchronous Memory Timing Requirements

NO.			MIN	MAX	UNIT
19	$t_{su(EMIFDV\text{-}EM\_CLKH)}$	Input setup time, read data valid on EMIF_DATA[15:0] before EMIF_CLK rising	1		ns
20	$t_{h(CLKH\text{-}DIV)}$	Input hold time, read data valid on EMIF_DATA[15:0] after EMIF_CLK rising	2.2		ns

Table 6-38. EMIF Synchronous Memory Switching Characteristics

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{c(CLK)}$	Cycle time, EMIF clock EMIF_CLK	10		ns
2	$t_{w(CLK)}$	Pulse width, EMIF clock EMIF_CLK high or low	3		ns
3	$t_{d(CLKH\text{-}CSV)}$	Delay time, EMIF_CLK rising to EMIF_nCS[0] valid		7	ns
4	$t_{oh(CLKH\text{-}CSIV)}$	Output hold time, EMIF_CLK rising to EMIF_nCS[0] invalid	1		ns
5	$t_{d(CLKH\text{-}DQMV)}$	Delay time, EMIF_CLK rising to EMIF_nDQM[1:0] valid		7	ns
6	$t_{oh(CLKH\text{-}DQMIV)}$	Output hold time, EMIF_CLK rising to EMIF_nDQM[1:0] invalid	1		ns
7	$t_{d(CLKH\text{-}AV)}$	Delay time, EMIF_CLK rising to EMIF_ADDR[21:0] and EMIF_BA[1:0] valid		7	ns
8	$t_{oh(CLKH\text{-}AIV)}$	Output hold time, EMIF_CLK rising to EMIF_ADDR[21:0] and EMIF_BA[1:0] invalid	1		ns
9	$t_{d(CLKH\text{-}DV)}$	Delay time, EMIF_CLK rising to EMIF_DATA[15:0] valid		7	ns

Table 6-38. EMIF Synchronous Memory Switching Characteristics (continued)

NO.	PARAMETER		MIN	MAX	UNIT
10	$t_{oh(CLKH-DIV)}$	Output hold time, EMIF_CLK rising to EMIF_DATA[15:0] invalid	1		ns
11	$t_{d(CLKH-RASV)}$	Delay time, EMIF_CLK rising to EMIF_nRAS valid		7	ns
12	$t_{oh(CLKH-RASIV)}$	Output hold time, EMIF_CLK rising to EMIF_nRAS invalid	1		ns
13	$t_{d(CLKH-CASV)}$	Delay time, EMIF_CLK rising to EMIF_nCAS valid		7	ns
14	$t_{oh(CLKH-CASIV)}$	Output hold time, EMIF_CLK rising to EMIF_nCAS invalid	1		ns
15	$t_{d(CLKH-WEV)}$	Delay time, EMIF_CLK rising to EMIF_nWE valid		7	ns
16	$t_{oh(CLKH-WEIV)}$	Output hold time, EMIF_CLK rising to EMIF_nWE invalid	1		ns
17	$t_{dis(CLKH-DHZ)}$	Delay time, EMIF_CLK rising to EMIF_DATA[15:0] tri-stated		7	ns
18	$t_{ena(CLKH-DLZ)}$	Output hold time, EMIF_CLK rising to EMIF_DATA[15:0] driving	1		ns

6.15 Vectored Interrupt Manager

There are two on-chip Vector Interrupt Manager (VIM) modules. The VIM module provides hardware assistance for prioritizing and controlling the many interrupt sources present on a device. Interrupts are caused by events outside of the normal flow of program execution. Normally, these events require a timely response from the CPU; therefore, when an interrupt occurs, the CPU switches execution from the normal program flow to an interrupt service routine (ISR).

6.15.1 VIM Features

The VIM module has the following features:

- Supports 128 interrupt channels
- Provides programmable priority for the request lines
- Manages interrupt channels through masking
- Prioritizes interrupt channels to the CPU
- Provides the CPU with the address of the interrupt service routine (ISR) for each interrupt

The two VIM modules are in lockstep. These two VIM modules are memory mapped to the same address space. From a programmer's model point of view it is only one VIM module. Writes to VIM1 registers and memory will be broadcasted to both VIM1 and VIM2. Reads from VIM1 will only read the VIM1 registers and memory. All interrupt requests which go to the VIM1 module will also go to the VIM2 module. Because the VIM1 and VIM2 have the identical setup, both will result in the same output behavior responding to the same interrupt requests. The second VIM module acts as a diagnostic checker module against the first VIM module. The output signals of the two VIM modules are routed to CCM-R5F module and are compared constantly. Mis-compare detected will be signaled as an error to the ESM module. The lockstep VIM pair takes care of the interrupt generation to the lockstep R5F pair.

6.15.2 Interrupt Generation

To avoid common mode failures the input and output signals of the two VIMs are delayed in a different way as shown in Figure 6-17.

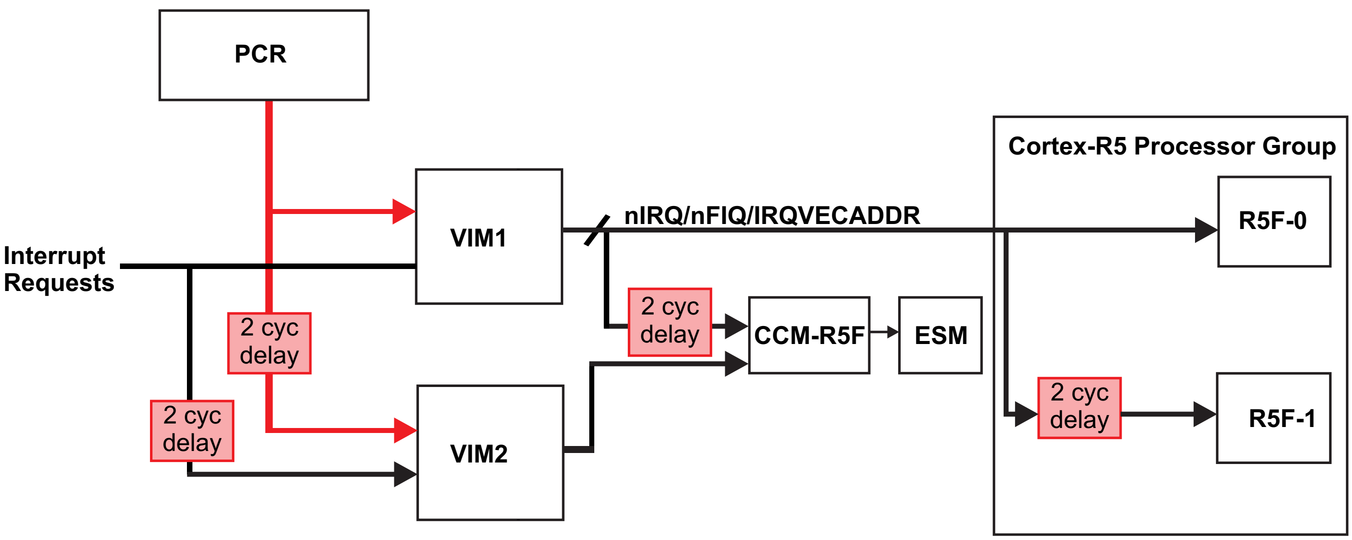


Figure 6-17. Interrupt Generation

6.15.3 Interrupt Request Assignments

Table 6-39. Interrupt Request Assignments

MODULES	VIM INTERRUPT SOURCES	DEFAULT VIM INTERRUPT CHANNEL
ESM	ESM high-level interrupt (NMI)	0
Reserved	Reserved	1
RTI	RTI1 compare interrupt 0	2
RTI	RTI1 compare interrupt 1	3
RTI	RTI1 compare interrupt 2	4
RTI	RTI1 compare interrupt 3	5
RTI	RTI1 overflow interrupt 0	6
RTI	RTI1 overflow interrupt 1	7
RTI	RTI1 time-base	8
GIO	GIO high level interrupt	9
NHET1	NHET1 high-level interrupt (priority level 1)	10
HET TU1	HET TU1 level 0 interrupt	11
MIBSPI1	MIBSPI1 level 0 interrupt	12
LIN1	LIN1 level 0 interrupt	13
MIBADC1	MIBADC1 event group interrupt	14
MIBADC1	MIBADC1 software group 1 interrupt	15
DCAN1	DCAN1 level 0 interrupt	16
MIBSPI2	MIBSPI2 level 0 interrupt	17
FlexRay	FlexRay level 0 interrupt (CC_int0)	18
CRC1	CRC1 Interrupt	19
ESM	ESM low-level interrupt	20
SYSTEM	Software interrupt for Cortex-R5F (SSI)	21
CPU	Cortex-R5F PMU Interrupt	22
GIO	GIO low level interrupt	23
NHET1	NHET1 low level interrupt (priority level 2)	24
HET TU1	HET TU1 level 1 interrupt	25
MIBSPI1	MIBSPI1 level 1 interrupt	26
LIN1	LIN1 level 1 interrupt	27
MIBADC1	MIBADC1 software group 2 interrupt	28
DCAN1	DCAN1 level 1 interrupt	29
MIBSPI2	MIBSPI2 level 1 interrupt	30
MIBADC1	MIBADC1 magnitude compare interrupt	31
FlexRay	FlexRay level 1 interrupt (CC_int1)	32
DMA	FTCA interrupt	33
DMA	LFSA interrupt	34
DCAN2	DCAN2 level 0 interrupt	35
DMM	DMM level 0 interrupt	36
MIBSPI3	MIBSPI3 level 0 interrupt	37
MIBSPI3	MIBSPI3 level 1 interrupt	38
DMA	HBCA interrupt	39
DMA	BTCA interrupt	40
EMIF	AEMIFINT	41
DCAN2	DCAN2 level 1 interrupt	42
DMM	DMM level 1 interrupt	43
DCAN1	DCAN1 IF3 interrupt	44

Table 6-39. Interrupt Request Assignments (continued)

MODULES	VIM INTERRUPT SOURCES	DEFAULT VIM INTERRUPT CHANNEL
DCAN3	DCAN3 level 0 interrupt	45
DCAN2	DCAN2 IF3 interrupt	46
FPU	FPU interrupt of Cortex-R5F	47
FlexRay TU	FlexRay TU Transfer Status interrupt (TU_Int0)	48
MIBSPI4	MIBSPI4 level 0 interrupt	49
MIBADC2	MibADC2 event group interrupt	50
MIBADC2	MibADC2 software group1 interrupt	51
FlexRay	FlexRay T0C interrupt (CC_tint0)	52
MIBSPI5	MIBSPI5 level 0 interrupt	53
MIBSPI4	MIBSPI4 level 1 interrupt	54
DCAN3	DCAN3 level 1 interrupt	55
MIBSPI5	MIBSPI5 level 1 interrupt	56
MIBADC2	MibADC2 software group2 interrupt	57
FlexRay TU	FlexRay TU Error interrupt (TU_Int1)	58
MIBADC2	MibADC2 magnitude compare interrupt	59
DCAN3	DCAN3 IF3 interrupt	60
L2FMC	FSM_DONE interrupt	61
FlexRay	FlexRay T1C interrupt (CC_tint1)	62
NHET2	NHET2 level 0 interrupt	63
SCI3	SCI3 level 0 interrupt	64
NHET TU2	NHET TU2 level 0 interrupt	65
I2C1	I2C level 0 interrupt	66
Reserved	Reserved	67–72
NHET2	NHET2 level 1 interrupt	73
SCI3	SCI3 level 1 interrupt	74
NHET TU2	NHET TU2 level 1 interrupt	75
Ethernet	C0_MISC_PULSE	76
Ethernet	C0_TX_PULSE	77
Ethernet	C0_THRESH_PULSE	78
Ethernet	C0_RX_PULSE	79
HWAG1	HWA_INT_REQ_H	80
HWAG2	HWA_INT_REQ_H	81
DCC1	DCC1 done interrupt	82
DCC2	DCC2 done interrupt	83
SYSTEM	Reserved	84
PBIST	PBIST Done	85
Reserved	Reserved	86–87
HWAG1	HWA_INT_REQ_L	88
HWAG2	HWA_INT_REQ_L	89
ePWM1INTn	ePWM1 Interrupt	90
ePWM1TZINTn	ePWM1 Trip Zone Interrupt	91
ePWM2INTn	ePWM2 Interrupt	92
ePWM2TZINTn	ePWM2 Trip Zone Interrupt	93
ePWM3INTn	ePWM3 Interrupt	94
ePWM3TZINTn	ePWM3 Trip Zone Interrupt	95
ePWM4INTn	ePWM4 Interrupt	96
ePWM4TZINTn	ePWM4 Trip Zone Interrupt	97

Table 6-39. Interrupt Request Assignments (continued)

MODULES	VIM INTERRUPT SOURCES	DEFAULT VIM INTERRUPT CHANNEL
ePWM5INTn	ePWM5 Interrupt	98
ePWM5TZINTn	ePWM5 Trip Zone Interrupt	99
ePWM6INTn	ePWM6 Interrupt	100
ePWM6TZINTn	ePWM6 Trip Zone Interrupt	101
ePWM7INTn	ePWM7 Interrupt	102
ePWM7TZINTn	ePWM7 Trip Zone Interrupt	103
eCAP1INTn	eCAP1 Interrupt	104
eCAP2INTn	eCAP2 Interrupt	105
eCAP3INTn	eCAP3 Interrupt	106
eCAP4INTn	eCAP4 Interrupt	107
eCAP5INTn	eCAP5 Interrupt	108
eCAP6INTn	eCAP6 Interrupt	109
eQEP1INTn	eQEP1 Interrupt	110
eQEP2INTn	eQEP2 Interrupt	111
Reserved	Reserved	112
DCAN4	DCAN4 Level 0 interrupt	113
I2C2	I2C2 interrupt	114
LIN2	LIN2 level 0 interrupt	115
SCI4	SCI4 level 0 interrupt	116
DCAN4	DCAN4 Level 1 interrupt	117
LIN2	LIN2 level 1 interrupt	118
SCI4	SCI4 level 1 interrupt	119
DCAN4	DCAN4 IF3 Interrupt	120
CRC2	CRC2 Interrupt	121
Reserved	Reserved	122
Reserved	Reserved	123
EPC	EPC FIFO FULL or CAM FULL interrupt	124
Reserved	Reserved	125-127

NOTE

Address location 0x00000000 in the VIM RAM is reserved for the phantom interrupt ISR entry; therefore only request channels 0..126 can be used and are offset by one address in the VIM RAM.

NOTE

The EMIF_nWAIT signal has a pull-up on it. The EMIF module generates a "Wait Rise" interrupt whenever it detects a rising edge on the EMIF_nWAIT signal. This interrupt condition is indicated as soon as the device is powered up. This can be ignored if the EMIF_nWAIT signal is not used in the application. If the EMIF_nWAIT signal is actually used in the application, then the external slave memory must always drive the EMIF_nWAIT signal such that an interrupt is not caused due to the default pull-up on this signal.

NOTE

The lower-order interrupt channels are higher priority channels than the higher-order interrupt channels.

NOTE

The application can change the mapping of interrupt sources to the interrupt channels through the interrupt channel control registers (CHANCTRLx) inside the VIM module.

6.16 ECC Error Event Monitoring and Profiling

This device includes an Error Profiling Controller (EPC) module. The main goal of this module is to enable the system to tolerate a certain amount of ECC correctable errors on the same address repeatedly in the memory system with minimal runtime overhead. Main features implemented in this device are described below.

- Capture the address of correctable ECC faults from different sources (for example, CPU, L2RAM, Interconnect) into a 16-entry Content Addressable Memory (CAM).
- For correctable faults, the error handling depends on the below conditions:
  – if the incoming address is already in the 16-entry CAM, discard the fail. No error generated to ESM
  – if the address is not in the CAM list, and the CAM has empty entries, add the address into the CAM list. In addition, raise the error signal to the ESM group 1 if enabled.
  – if the address is not in the CAM list, and the CAM has no empty entries, always raise a signal to the ESM group 1.
- A 4-entry FIFO to store correctable error events and addresses for each IP interface.
- For uncorrectable faults of non-CPU access, capture the address and raise a signal to the ESM group 2.
- The CAM is implemented in memory mapped registers. The CPU can read and write to any entry for diagnostic test as if a real CAM memory macro.

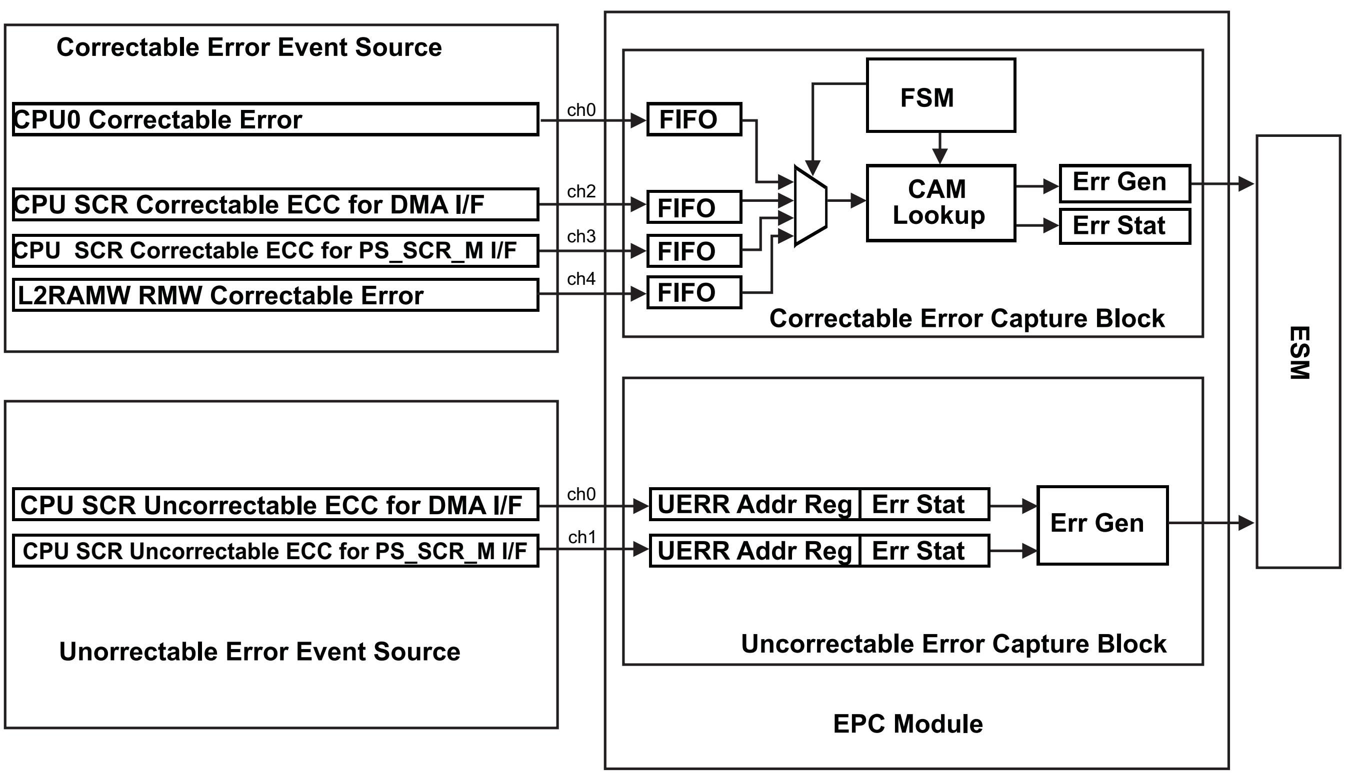


Figure 6-18. EPC Block Diagram

6.16.1 EPC Module Operation

6.16.1.1 Correctable Error Handling

When a correctable error is detected in the system by an IP, it sends the error signal along with the error address to EPC module. The EPC module will scan this error address in the 16-entry CAM. If there is a match then the address is discard and no error is generated to ESM by the ECP. It takes one cycle to scan one address at a time through the CAM. The idea is to allow the system to tolerate a correctable error occurring on the same address because this error has been handled before by the CPU. This error scenario is particularly frequent when the software is in a for loop fetching the same address. Because there are multiple IPs which can simultaneously detect correctable errors in the system, the EPC employs a 4-entry FIFO per IP interface so that error addresses are not lost.

If an address is not matched in the CAM then it depends if there is empty entry in the CAM. If there is an empty entry then the new address is stored into the empty entry. For each entry there is a 4-bit valid key. When a new address is stored the 4-bit key is updated with "1010". It is programmable to generate a correctable error to the ESM if the address is not matched and there is an empty CAM entry. Once CPU is interrupted, it can choose to evaluate the error address and handle it accordingly. The software can also invalidate the entry by writing "0101".

If an dress is not matched and there is no empty entry in the CAM then the correctable error is immediately sent to the ESM. The new error address is lost if there is no empty entry left in the CAM.

6.16.1.2 Uncorrectable Error Handling

Uncorrectable errors reported by the IP (non-CPU access) are immediately captured for their error addresses and update to the uncorrectable error status register. For more information see the device specific technical reference guide SPNU563.

6.17 DMA Controller

The DMA controller is used to transfer data between two locations in the memory map in the background of CPU operations. Typically, the DMA is used to:

- Transfer blocks of data between external and internal data memories
- Restructure portions of internal data memory
- Continually service a peripheral

6.17.1 DMA Features

- 64-bit OCP protocol to perform bus master accesses
- INCR-4 64-bit burst accesses
- Multithreading architecture allowing data of two different channel transfers to be interleaved during nonburst accesses
- 2-port configuration for parallel bus master
- Channels can be assigned to either high-priority queue or low-priority queue. Within each queue, fixed or round-robin priorities can be serviced
- Built-in ECC generation and evaluation logic for internal RAM storing channel transfer information
- Supports multiple interrupt outputs for mapping to multiple interrupt controllers in multicore systems
- 48 requests can be mapped to any 32 channels
- Supports LE endianess
- External ECC Gen/Eval block of DMA support ECC generation for data transactions, and parity for address, and control signals (following Cortex-R5F standard)
- 8 MPU regions
- Channel chaining capability
- Hardware and software DMA requests
- 8-, 16-, 32-, or 64-bit transactions supported
- Multiple addressing modes for source/destination (fixed, increment, offset)
- Auto-initiation

6.17.2 DMA Transfer Port Assignment

There are two ports, port A and port B attached to the DMA controller. When configuring a DMA channel for a transfer, the application must also specify the port associated with the transfer source and destination. Table 6-40 lists the mapping between each port and the resources. For example, if a transfer is to be made from the the flash to the SRAM, the application will need configure the desired DMA channel in the PARx register to select port A as the target for both the source and destination. If a transfer is to be made from the SRAM to a peripheral or a peripheral memory, the application will need to configure the desired DMA channel in the PARx register to select port A for read and port B for write. Likewise, if a transfer is from a peripheral to the SRAM then the PARx will be configured to select port B for read and port A for write.

Table 6-40. DMA Port Assignment

TARGET NAME	ACCESS PORT OF DMA
Flash	Port A
SRAM	Port A
EMIF	Port A
Flash OTP/ECC/EEPROM	Port A
All other targets (peripherals, peripheral memories)	Port B

6.17.3 Default DMA Request Map

The DMA module on this microcontroller has 32 channels and up to 48 hardware DMA requests. The module contains DREQASIx registers which are used to map the DMA requests to the DMA channels. By default, channel 0 is mapped to request 0, channel 1 to request 1, and so on.

Some DMA requests have multiple sources, see Table 6-41. The application must ensure that only one of these DMA request sources is enabled at any time.

Table 6-41. DMA Request Line Connection

MODULES	DMA REQUEST SOURCES	DMA REQUEST
MIBSPI1	MIBSPI1[1](1)	DMAREQ[0]
MIBSPI1	MIBSPI1[0](2)	DMAREQ[1]
MIBSPI2	MIBSPI2[1](1)	DMAREQ[2]
MIBSPI2	MIBSPI2[0](2)	DMAREQ[3]
MIBSPI1 / MIBSPI3 / DCAN2	MIBSPI1[2] / MIBSPI3[2] / DCAN2 IF3	DMAREQ[4]
MIBSPI1 / MIBSPI3 / DCAN2	MIBSPI1[3] / MIBSPI3[3] / DCAN2 IF2	DMAREQ[5]
DCAN1 / MIBSPI5	DCAN1 IF2 / MIBSPI5[2]	DMAREQ[6]
MIBADC1 / MIBSPI5	MIBADC1 event / MIBSPI5[3]	DMAREQ[7]
MIBSPI1 / MIBSPI3 / DCAN1	MIBSPI1[4] / MIBSPI3[4] / DCAN1 IF1	DMAREQ[8]
MIBSPI1 / MIBSPI3 / DCAN2	MIBSPI1[5] / MIBSPI3[5] / DCAN2 IF1	DMAREQ[9]
MIBADC1 / I2C / MIBSPI5	MIBADC1 G1 / I2C receive / MIBSPI5[4]	DMAREQ[10]
MIBADC1 / I2C / MIBSPI5	MIBADC1 G2 / I2C transmit / MIBSPI5[5]	DMAREQ[11]
RTI1 / MIBSPI1 / MIBSPI3	RTI1 DMAREQ0 / MIBSPI1[6] / MIBSPI3[6]	DMAREQ[12]
RTI1 / MIBSPI1 / MIBSPI3	RTI1 DMAREQ1 / MIBSPI1[7] / MIBSPI3[7]	DMAREQ[13]
MIBSPI3 / MibADC2 / MIBSPI5	MIBSPI3[1](1) / MibADC2 event / MIBSPI5[6]	DMAREQ[14]
MIBSPI3 / MIBSPI5	MIBSPI3[0](2) / MIBSPI5[7]	DMAREQ[15]
MIBSPI1 / MIBSPI3 / DCAN1 / MibADC2	MIBSPI1[8] / MIBSPI3[8] / DCAN1 IF3 / MibADC2 G1	DMAREQ[16]
MIBSPI1 / MIBSPI3 / DCAN3 / MibADC2	MIBSPI1[9] / MIBSPI3[9] / DCAN3 IF1 / MibADC2 G2	DMAREQ[17]
RTI1 / MIBSPI5	RTI1 DMAREQ2 / MIBSPI5[8]	DMAREQ[18]
RTI1 / MIBSPI5	RTI1 DMAREQ3 / MIBSPI5[9]	DMAREQ[19]
NHET1 / NHET2 / DCAN3	NHET1 DMAREQ[4] / NHET2 DMAREQ[4] / DCAN3 IF2	DMAREQ[20]
NHET1 / NHET2 / DCAN3	NHET1 DMAREQ[5] / NHET2 DMAREQ[5] / DCAN3 IF3	DMAREQ[21]
MIBSPI1 / MIBSPI3 / MIBSPI5	MIBSPI1[10] / MIBSPI3[10] / MIBSPI5[10]	DMAREQ[22]
MIBSPI1 / MIBSPI3 / MIBSPI5	MIBSPI1[11] / MIBSPI3[11] / MIBSPI5[11]	DMAREQ[23]
NHET1 / NHET2 / MIBSPI4 / MIBSPI5	NHET1 DMAREQ[6] / NHET2 DMAREQ[6] / MIBSPI4[1] (1)/ MIBSPI5[12]	DMAREQ[24]
NHET1 / NHET2 / MIBSPI4 / MIBSPI5	NHET1 DMAREQ[7] / NHET2 DMAREQ[7] / MIBSPI4[0](2) / MIBSPI5[13]	DMAREQ[25]
CRC1 / MIBSPI1 / MIBSPI3	CRC1 DMAREQ[0] / MIBSPI1[12] / MIBSPI3[12]	DMAREQ[26]
CRC1 / MIBSPI1 / MIBSPI3	CRC1 DMAREQ[1] / MIBSPI1[13] / MIBSPI3[13]	DMAREQ[27]
LIN1 / MIBSPI5	LIN1 receive / MIBSPI5[14]	DMAREQ[28]
LIN1 / MIBSPI5	LIN1 transmit / MIBSPI5[15]	DMAREQ[29]
MIBSPI1 / MIBSPI3 / SCI3 / MIBSPI5	MIBSPI1[14] / MIBSPI3[14] / SCI3 receive / MIBSPI5[1](1)	DMAREQ[30]
MIBSPI1 / MIBSPI3 / SCI3 / MIBSPI5	MIBSPI1[15] / MIBSPI3[15] / SCI3 transmit / MIBSPI5[0](2)	DMAREQ[31]
I2C2 / ePWM1 / MIBSPI2 / MIBSPI4 / GIOA	I2C2 receive / ePWM1_SOCA / MIBSPI2[2] / MIBSPI4[2] / GIOA[0]	DMAREQ[32]
I2C2 / ePWM 1 / MIBSPI2 / MIBSPI4 / GIOA	I2C2 transmit / ePWM1_SOCB / MIBSPI2[3] / MIBSPI4[3] / GIOA[1]	DMAREQ[33]
ePWM2 / MIBSPI2 / MIBSPI4 / GIOA	ePWM2_SOCA / MIBSPI2[4] / MIBSPI4[4] / GIOA[2]	DMAREQ[34]
ePWM2 / MIBSPI2 / MIBSPI4 / GIOA	ePWM2_SOCB / MIBSPI2[5] / MIBSPI4[5] / GIOA[3]	DMAREQ[35]
ePWM3 / MIBSPI2 / MIBSPI4 / GIOA	ePWM3_SOCA / MIBSPI2[6] / MIBSPI4[6] / GIOA[4]	DMAREQ[36]
ePWM3 / MIBSPI2 / MIBSPI4 / GIOA	ePWM3_SOCB / MIBSPI2[7] / MIBSPI4[7] / GIOA[5]	DMAREQ[37]
CRC2 / ePWM4 / MIBSPI2 / MIBSPI4 / GIOA	CRC2 DMAREQ[0] / ePWM4_SOCA / MIBSPI2[8] / MIBSPI4[8] / GIOA[6]	DMAREQ[38]
CRC2 / ePWM4 / MIBSPI2 / MIBSPI4 / GIOA	CRC2 DMAREQ[1] / ePWM4_SOCB / MIBSPI2[9] / MIBSPI4[9] / GIOA[7]	DMAREQ[39]
LIN2 / ePWM5 / MIBSPI2 / MIBSPI4 / GIOB	LIN2 receive / ePWM5_SOCA / MIBSPI2[10] / MIBSPI4[10] / GIOB[0]	DMAREQ[40]
LIN2 / ePWM5 / MIBSPI2 / MIBSPI4 / GIOB	LIN2 transmit / ePWM5_SOCB / MIBSPI2[11] / MIBSPI4[11] / GIOB[1]	DMAREQ[41]

(1) SPI1, SPI2, SPI3, SPI4, SPI5 receive in compatibility mode
(2) SPI1, SPI2, SPI3, SPI4, SPI5 transmit in compatibility mode

Table 6-41. DMA Request Line Connection (continued)

MODULES	DMA REQUEST SOURCES	DMA REQUEST
SCI4 / ePWM6 / MIBSPI2 / MIBSPI4 / GIOB	SCI4 receive / ePWM6_SOCA / MIBSPI2[12] / MIBSPI4[12] / GIOB[2]	DMAREQ[42]
SCI4 / ePWM6 / MIBSPI2 / MIBSPI4 / GIOB	SCI4 transmit / ePWM6_SOCB / MIBSPI2[13] / MIBSPI4[13] / GIOB[3]	DMAREQ[43]
ePWM7 / MIBSPI2 / MIBSPI4 / GIOB	ePWM7_SOCA / MIBSPI2[14] / MIBSPI4[14] / GIOB[4]	DMAREQ[44]
ePWM7 / MIBSPI2 / MIBSPI4 / GIOB / DCAN4	ePWM7_SOCB / MIBSPI2[15] / MIBSPI4[15] / GIOB[5] / DCAN4 IF1	DMAREQ[45]
GIOB / DCAN4	GIOB[6] / DCAN4_IF2	DMAREQ[46]
GIOB / DCAN4	GIOB[7] / DCAN4_IF3	DMAREQ[47]

6.17.4 Using a GIO terminal as a DMA Request Input

Each GIO terminal can also directly be used as DMA request input as listed in Table 6-41. The polarity of the GIO terminal to trigger a DMA request can be selected inside the DMA module. To use the GIO terminal as a DMA request input, the corresponding select bit must be set to low. See Figure 6-19 for an illustration. For more information see the technical reference guide SPNU563.

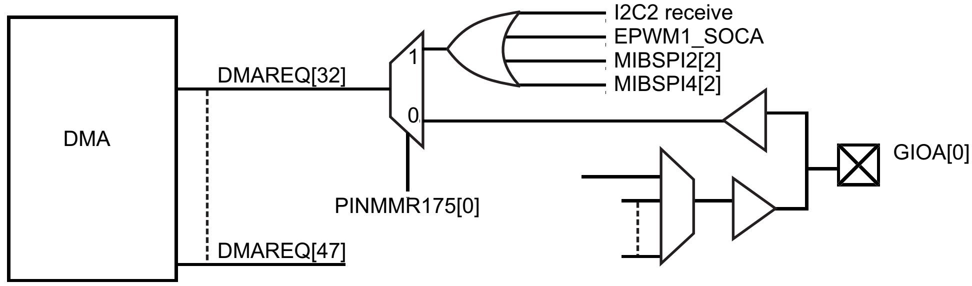


Figure 6-19. Using a GIO terminal as a DMA Request Input

Table 6-42. GIO DMA Request Disable Mapping

GIO TERMINAL	GIO DMA REQUEST SELECT BIT
GIOA[0]	PINMMR175[0]
GIOA[1]	PINMMR175[8]
GIOA[2]	PINMMR175[16]
GIOA[3]	PINMMR175[24]
GIOA[4]	PINMMR176[0]
GIOA[5]	PINMMR176[8]
GIOA[6]	PINMMR176[16]
GIOA[7]	PINMMR176[24]
GIOB[0]	PINMMR177[0]
GIOB[1]	PINMMR177[8]
GIOB[2]	PINMMR177[16]
GIOB[3]	PINMMR177[24]
GIOB[4]	PINMMR178[0]
GIOB[5]	PINMMR178[8]
GIOB[6]	PINMMR178[16]
GIOB[7]	PINMMR178[24]

6.18 Real-Time Interrupt Module

The real-time interrupt (RTI) module provides timer functionality for operating systems and for benchmarking code. The RTI module can incorporate several counters that define the time bases needed for scheduling an operating system.

The timers also let you benchmark certain areas of code by reading the values of the counters at the beginning and the end of the desired code range and calculating the difference between the values.

6.18.1 Features

The RTI module has the following features:

- Two independent 64-bit counter blocks
- Four configurable compares for generating operating system ticks or DMA requests. Each event can be driven by either counter block 0 or counter block 1.
- Fast enabling/disabling of events
- Two timestamp (capture) functions for system or peripheral interrupts, one for each counter block

6.18.2 Block Diagrams

Figure 6-20 shows a high-level block diagram for one of the two 64-bit counter blocks inside the RTI module. Both the counter blocks are identical except the Network Time Unit (NTUx) inputs are only available as time-base inputs for the counter block 0. Figure 6-21 shows the compare unit block diagram of the RTI module.

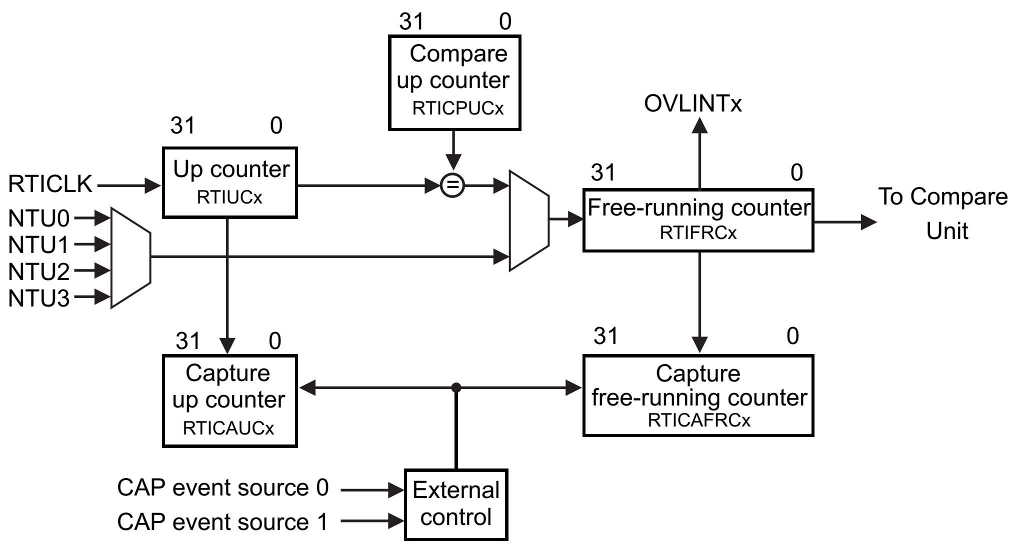


Figure 6-20. Counter Block Diagram

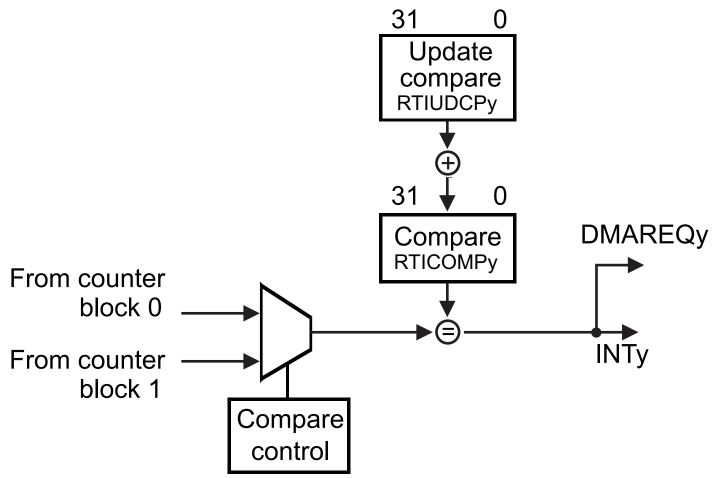


Figure 6-21. Compare Block Diagram

6.18.3 Clock Source Options

The RTI module uses the RTI1CLK clock domain for generating the RTI time bases.

The application can select the clock source for the RTI1CLK by configuring the RCLKSRC register in the system module at address 0xFFFFFF50. The default source for RTI1CLK is VCLK.

For more information on clock sources, see Table 6-11 and Table 6-16.

6.18.4 Network Time Synchronization Inputs

The RTI module supports four Network Time Unit (NTU) inputs that signal internal system events, and which can be used to synchronize the time base used by the RTI module. On this device, these NTU inputs are connected as shown in Table 6-43.

Table 6-43. Network Time Synchronization Inputs

NTU INPUT	SOURCE
0	FlexRay Macrotick
1	FlexRay Start of Cycle
2	PLL2 Clock output
3	EXTCLKIN1 clock input

6.19 Error Signaling Module

The Error Signaling Module (ESM) manages the various error conditions on the TMS570LCx microcontroller. The error condition is handled based on a fixed severity level assigned to it. Any severe error condition can be configured to drive a low level on a dedicated device terminal called nERROR. The nERROR can be used as an indicator to an external monitor circuit to put the system into a safe state.

6.19.1 ESM Features

The features of the ESM are:

- 160 interrupt/error channels are supported, divided into three groups
  - 96 channels with maskable interrupt and configurable error terminal behavior
  - 32 error channels with nonmaskable interrupt and predefined error terminal behavior
  - 32 channels with predefined error terminal behavior only
- Error terminal to signal severe device failure
- Configurable time base for error signal
- Error forcing capability

6.19.2 ESM Channel Assignments

The ESM integrates all the device error conditions and groups them in the order of severity. Group1 is used for errors of the lowest severity while Group3 is used for errors of the highest severity. The device response to each error is determined by the severity group to which the error is connected. Table 6-45 lists the channel assignment for each group.

Table 6-44. ESM Groups

ERROR GROUP	INTERRUPT CHARACTERISTICS	INFLUENCE ON ERROR TERMINAL
Group1	Maskable, low or high priority	Configurable
Group2	Nonmaskable, high priority	Fixed
Group3	No interrupt generated	Fixed

Table 6-45. ESM Channel Assignments

ESM ERROR SOURCES	GROUP	CHANNELS
Group1		
Reserved	Group1	0
MibADC2 - parity	Group1	1
DMA - MPU error for CPU (DMAOCP_MPVINT(0))	Group1	2
DMA - ECC uncorrectable error	Group1	3
EPC - Correctable Error	Group1	4
Reserved	Group1	5
L2FMC - correctable error (implicit OTP read).	Group1	6
NHET1 - parity	Group1	7
HET TU1/HET TU2 - parity	Group1	8
HET TU1/HET TU2 - MPU	Group1	9
PLL1 - slip	Group1	10
LPO Clock Monitor - interrupt	Group1	11
FlexRay RAM - ECC uncorrectable error	Group1	12
Reserved	Group1	13
FlexRay TU RAM - ECC uncorrectable error (TU_UCT_err)	Group1	14

Table 6-45. ESM Channel Assignments (continued)

ESM ERROR SOURCES	GROUP	CHANNELS
VIM RAM - ECC uncorrectable error	Group1	15
FlexRay TU - MPU violation (TU_MPV_err)	Group1	16
MibSPI1 - ECC uncorrectable error	Group1	17
MibSPI3 - ECC uncorrectable error	Group1	18
MibADC1 - parity	Group1	19
DMA - Bus Error	Group1	20
DCAN1 - ECC uncorrectable error	Group1	21
DCAN3 - ECC uncorrectable error	Group1	22
DCAN2 - ECC uncorrectable error	Group1	23
MibSPI5 - ECC uncorrectable error	Group1	24
Reserved	Group1	25
L2RAMW - correctable error	Group1	26
Cortex-R5F CPU - self-test	Group1	27
Reserved	Group1	28
Reserved	Group1	29
DCC1 - error	Group1	30
CCM-R5F - self-test	Group1	31
Reserved	Group1	32
Reserved	Group1	33
NHET2 - parity	Group1	34
Reserved	Group1	35
Reserved	Group1	36
IOMM - Mux configuration error	Group1	37
Power domain compare error	Group1	38
Power domain self-test error	Group1	39
eFuse farm – EFC error	Group1	40
eFuse farm - self-test error	Group1	41
PLL2 - slip	Group1	42
Ethernet Controller master interface	Group1	43
Reserved	Group1	44
Reserved	Group1	45
Cortex-R5F Core - cache correctable error event	Group1	46
ACP d-cache invalidate	Group1	47
Reserved	Group1	48
MibSPI2 - ECC uncorrectable error	Group1	49
MibSPI4 - ECC uncorrectable error	Group1	50
DCAN4 - ECC uncorrectable error	Group1	51
CPU Interconnect Subsystem - Global error	Group1	52
CPU Interconnect Subsystem - Global Parity Error	Group1	53
NHET1/2 - self-test error	Group1	54
NMPU - EMAC MPU Error	Group1	55
Reserved	Group1	56
Reserved	Group1	57
Reserved	Group1	58
Reserved	Group1	59
Reserved	Group1	60
NMPU - PS_SCR_S MPU Error	Group1	61

Table 6-45. ESM Channel Assignments (continued)

<table>
<tr><th colspan="3">ESM ERROR SOURCES</th><th>GROUP</th><th>CHANNELS</th></tr>
<tr><td colspan="3">DCC2 - error</td><td>Group1</td><td>62</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>63</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>64</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>65</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>66</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>67</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>68</td></tr>
<tr><td colspan="3">NMPU - DMA Port A MPU Error</td><td>Group1</td><td>69</td></tr>
<tr><td colspan="3">DMA - Transaction Bus Parity Error</td><td>Group1</td><td>70</td></tr>
<tr><td colspan="3">FlexRay TU RAM- ECC single bit error (TU_SBE_err)</td><td>Group1</td><td>71</td></tr>
<tr><td colspan="3">FlexRay - ECC single bit error</td><td>Group1</td><td>72</td></tr>
<tr><td colspan="3">DCAN1 - ECC single bit error</td><td>Group1</td><td>73</td></tr>
<tr><td colspan="3">DCAN2 - ECC single bit error</td><td>Group1</td><td>74</td></tr>
<tr><td colspan="3">DCAN3 - ECC single bit error</td><td>Group1</td><td>75</td></tr>
<tr><td colspan="3">DCAN4 - ECC single bit error</td><td>Group1</td><td>76</td></tr>
<tr><td colspan="3">MIBSPI1 - ECC single bit error</td><td>Group1</td><td>77</td></tr>
<tr><td colspan="3">MIBSPI2 - ECC single bit error</td><td>Group1</td><td>78</td></tr>
<tr><td colspan="3">MIBSPI3 - ECC single bit error</td><td>Group1</td><td>79</td></tr>
<tr><td colspan="3">MIBSPI4 - ECC single bit error</td><td>Group1</td><td>80</td></tr>
<tr><td colspan="3">MIBSPI5 - ECC single bit error</td><td>Group1</td><td>81</td></tr>
<tr><td colspan="3">DMA - ECC single bit error</td><td>Group1</td><td>82</td></tr>
<tr><td colspan="3">VIM - ECC single bit error</td><td>Group1</td><td>83</td></tr>
<tr><td colspan="3">EMIF 64-bit Bridge I/F ECC uncorrectable error</td><td>Group1</td><td>84</td></tr>
<tr><td colspan="3">EMIF 64-bit Bridge I/F ECC single bit error</td><td>Group1</td><td>85</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>86</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>87</td></tr>
<tr><td colspan="3">DMA - Register Soft Error</td><td>Group1</td><td>88</td></tr>
<tr><td colspan="3">L2FMC - Register Soft Error</td><td>Group1</td><td>89</td></tr>
<tr><td colspan="3">SYS - Register Soft Error</td><td>Group1</td><td>90</td></tr>
<tr><td colspan="3">SCM - Time-out Error</td><td>Group1</td><td>91</td></tr>
<tr><td colspan="3">CCM-R5F - Operating status</td><td>Group1</td><td>92</td></tr>
<tr><td colspan="3">Reserved</td><td>Group1</td><td>93-95</td></tr>
<tr><th colspan="5">Group2</th></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>0</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>1</td></tr>
<tr><td colspan="3">CCM-R5F - CPU compare error</td><td>Group2</td><td>2</td></tr>
<tr><td colspan="3">Cortex-R5F Core - All fatal bus error events. [Commonly caused by improper or incomplete ECC values in Flash.]</td><td rowspan="3">Group2</td><td rowspan="3">3</td></tr>
<tr><td>Event Reference</td><td>Event Description</td><td>EVNTBUSm bit</td></tr>
<tr><td>0x71</td><td>Bus ECC</td><td>48</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>4</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>5</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>6</td></tr>
<tr><td colspan="3">L2RAMW - Uncorrectable error type B</td><td>Group2</td><td>7</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>8</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>9</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>10</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>11</td></tr>
</table>

Table 6-45. ESM Channel Assignments (continued)

<table>
<tr><th colspan="3">ESM ERROR SOURCES</th><th>GROUP</th><th>CHANNELS</th></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>12</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>13</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>14</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>15</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>16</td></tr>
<tr><td colspan="3">L2FMC - parity error
• Mcmd parity error on Idle command
• POM idle state parity error
• Port A/B Idle state parity error</td><td>Group2</td><td>17</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>18</td></tr>
<tr><td colspan="3">L2FMC - double bit ECC error-error due to implicit OTP reads</td><td>Group2</td><td>19</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>20</td></tr>
<tr><td colspan="3">EPC - Uncorrectable Error</td><td>Group2</td><td>21</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>22</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>23</td></tr>
<tr><td colspan="3">RTI_WWD_NMI</td><td>Group2</td><td>24</td></tr>
<tr><td colspan="3">CCM-R5F VIM compare error</td><td>Group2</td><td>25</td></tr>
<tr><td colspan="3">CPU1 AXIM Bus Monitor failure</td><td>Group2</td><td>26</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>27</td></tr>
<tr><td colspan="3">CCM-R5F - Power Domain monitor error</td><td>Group2</td><td>28</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>29</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>30</td></tr>
<tr><td colspan="3">Reserved</td><td>Group2</td><td>31</td></tr>
<tr><td colspan="5">Group3</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>0</td></tr>
<tr><td colspan="3">eFuse Farm - autoload error</td><td>Group3</td><td>1</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>2</td></tr>
<tr><td colspan="3">L2RAMW - double bit ECC uncorrectable error</td><td>Group3</td><td>3</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>4</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>5</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>6</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>7</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>8</td></tr>
<tr><td colspan="3">Cortex-R5F Core - All fatal events (OR of:</td><td rowspan="4">Group3</td><td rowspan="4">9</td></tr>
<tr><td>Event Reference Value</td><td>Event Description</td><td>EVNTBUSm Bit</td></tr>
<tr><td>0x60</td><td>Data Cache</td><td>33</td></tr>
<tr><td>0x61</td><td>Data Cache tag/dirty</td><td>34</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>10</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>11</td></tr>
<tr><td colspan="3">CPU Interconnect Subsystem - Diagnostic Error</td><td>Group3</td><td>12</td></tr>
<tr><td colspan="3">L2FMC - uncorrectable error due to:
• address parity/internal parity error
• address tag
• internal switch time-out</td><td>Group3</td><td>13</td></tr>
<tr><td colspan="3">L2RAMW - Uncorrectable error Type A</td><td>Group3</td><td>14</td></tr>
<tr><td colspan="3">L2RAMW - Address/Control parity error</td><td>Group3</td><td>15</td></tr>
<tr><td colspan="3">Reserved</td><td>Group3</td><td>16</td></tr>
</table>

Table 6-45. ESM Channel Assignments (continued)

ESM ERROR SOURCES	GROUP	CHANNELS
Reserved	Group3	17
Reserved	Group3	18
Reserved	Group3	19
Reserved	Group3	20
Reserved	Group3	21
Reserved	Group3	22
Reserved	Group3	23
Reserved	Group3	24

6.20 Reset / Abort / Error Sources

Table 6-46. Reset/Abort/Error Sources

ERROR SOURCE	SYSTEM MODE	ERROR RESPONSE	ESM HOOKUP GROUP.CHANNEL
CPU TRANSACTIONS			
Precise write error (NCNB/Strongly Ordered)	User/Privilege	Precise Abort (CPU)	N/A
Precise read error (NCB/Device or Normal)	User/Privilege	Precise Abort (CPU)	N/A
Imprecise write error (NCB/Device or Normal)	User/Privilege	Imprecise Abort (CPU)	N/A
Illegal instruction	User/Privilege	Undefined Instruction Trap (CPU)(1)	N/A
MPU access violation	User/Privilege	Abort (CPU)	N/A
Correctable error	User/Privilege	ESM	1.4
Uncorrectable error	User/Privilege	ESM => NMI	2.21
LEVEL 2 SRAM			
CPU Write ECC single error (correctable)	User/Privilege	ESM	1.26
ECC double bit error: Read-Modify-Write (RMW) ECC double error CPU Write ECC double error	User/Privilege	Bus Error, ESM => nERROR	3.3
Uncorrectable error Type A: Write SECDED malfunction error Redundant address decode error Read SECDED malfunction error	User/Privilege	Bus Error, ESM => nERROR	3.14
Uncorrectable error type B: Memory scrubbing SECDED malfunction error Memory scrubbing Redundant address decode error Memory scrubbing address/control parity error Write data merged mux diagnostic error Write SECDED malfunction diagnostic error Read SECDED malfunction diagnostic error Write ECC correctable and uncorrectable diagnostic error Read ECC correctable and uncorrectable diagnostic error Write data merged mux error Redundant address decode diagnostic error Command parity error on idle	User/Privilege	ESM => NMI	2.7
Address/Control parity error	User/Privilege	Bus Error, ESM => nERROR	3.15
Level 2 RAM illegal address error Memory initialization error	User/Privilege	Bus Error	N/A
FLASH			
L2FMC correctable error - single bit ECC error for implicit OTP read	User/Privilege	ESM	1.6
L2FMC uncorrectable error - double bit ECC error for implicit OTP read	User/Privilege	ESM => NMI	2.19
L2FMC fatal uncorrectable error: address parity error/internal parity error address tag error Internal switch time-out	User/Privilege	Bus Error, ESM => nERROR	3.13
L2FMC parity error: Mcmd parity error on Idle command POM idle state parity error Port A/B Idle state parity error	User/Privilege	ESM => NMI	2.17
L2FMC nonfatal uncorrectable error: Response error on POM Response parity error on POM Bank accesses during special operation (program/erase) by the FSM Bank/Pump in sleep Unimplemented special/unavailable space	User/Privilege	Bus Error	N/A

(1) The Undefined Instruction TRAP is not detectable outside the CPU. The trap is taken only if the instruction reaches the execute stage of the CPU.

Table 6-46. Reset/Abort/Error Sources (continued)

ERROR SOURCE	SYSTEM MODE	ERROR RESPONSE	ESM HOOKUP GROUP.CHANNEL
L2FMC register soft error.	User/Privilege	ESM	1.89
DMA TRANSACTIONS			
Memory access permission violation	User/Privilege	ESM	1.2
Memory ECC uncorrectable error	User/Privilege	ESM	1.3
Transaction Error: that is, Bus Parity Error	User/Privilege	ESM	1.70
Memory ECC single bit error	User/Privilege	ESM	1.82
DMA register soft error	User/Privilege	ESM	1.88
DMA bus error	User/Privilege	ESM	1.20
EMIF_ECC			
64-bit Bridge I/F ECC uncorrectable error	User/Privilege	ESM	1.84
64-bit Bridge I/F ECC single error	User/Privilege	ESM	1.85
HET TU1 (HTU1)			
NCNB (Strongly Ordered) transaction with slave error response	User/Privilege	Interrupt => VIM	N/A
External imprecise error (Illegal transaction with ok response)	User/Privilege	Interrupt => VIM	N/A
Memory access permission violation	User/Privilege	ESM	1.9
Memory parity error	User/Privilege	ESM	1.8
HET TU2 (HTU2)			
NCNB (Strongly Ordered) transaction with slave error response	User/Privilege	Interrupt => VIM	N/A
External imprecise error (Illegal transaction with ok response)	User/Privilege	Interrupt => VIM	N/A
Memory access permission violation	User/Privilege	ESM	1.9
Memory parity error	User/Privilege	ESM	1.8
N2HET1			
Memory parity error	User/Privilege	ESM	1.7
N2HET2			
Memory parity error	User/Privilege	ESM	1.34
MibSPI			
MibSPI1 memory ECC uncorrectable error	User/Privilege	ESM	1.17
MibSPI2 memory ECC uncorrectable error	User/Privilege	ESM	1.49
MibSPI3 memory ECC uncorrectable error	User/Privilege	ESM	1.18
MibSPI4 memory ECC uncorrectable error	User/Privilege	ESM	1.50
MibSPI5 memory ECC uncorrectable error	User/Privilege	ESM	1.24
MibSPI1 memory ECC single error	User/Privilege	ESM	1.77
MibSPI2 memory ECC single error	User/Privilege	ESM	1.78
MibSPI3 memory ECC single error	User/Privilege	ESM	1.79
MibSPI4 memory ECC single error	User/Privilege	ESM	1.80
MibSPI5 memory ECC single error	User/Privilege	ESM	1.81
MibADC			
MibADC1 Memory parity error	User/Privilege	ESM	1.19
MibADC2 Memory parity error	User/Privilege	ESM	1.1
DCAN			
DCAN1 memory ECC uncorrectable error	User/Privilege	ESM	1.21
DCAN2 memory ECC uncorrectable error	User/Privilege	ESM	1.23
DCAN3 memory ECC uncorrectable error	User/Privilege	ESM	1.22
DCAN4 memory ECC uncorrectable error	User/Privilege	ESM	1.51
DCAN1 memory ECC single error	User/Privilege	ESM	1.73

Table 6-46. Reset/Abort/Error Sources (continued)

ERROR SOURCE	SYSTEM MODE	ERROR RESPONSE	ESM HOOKUP GROUP.CHANNEL
DCAN2 memory ECC single error	User/Privilege	ESM	1.74
DCAN3 memory ECC single error	User/Privilege	ESM	1.75
DCAN4 memory ECC single error	User/Privilege	ESM	1.76
PLL			
PLL1 slip error	User/Privilege	ESM	1.10
PLL2 slip error	User/Privilege	ESM	1.42
Clock Monitor			
Clock monitor interrupt	User/Privilege	ESM	1.11
DCC			
DCC1 error	User/Privilege	ESM	1.30
DCC2 error	User/Privilege	ESM	1.62
CCM-R5F			
Self-test failure	User/Privilege	ESM	1.31
CPU Bus Compare failure	User/Privilege	ESM => NMI	2.2
VIM Bus Compare failure	User/Privilege	ESM => NMI	2.25
Power Domain Monitor failure	User/Privilege	ESM => NMI	2.28
CCM-R5F operating status (asserted when not in lockstep or CCM-R5F is in self-test mode)	User/Privilege	ESM	1.92
EPC (Error Profiling Controller)			
Correctable Error	User/Privilege	ESM	1.4
Uncorrectable Error	User/Privilege	ESM => NMI	2.21
SCM (SCR Control module)			
Time-out Error	User/Privilege	ESM	1.91
FlexRay			
Memory ECC uncorrectable error	User/Privilege	ESM	1.12
Memory ECC single error	User/Privilege	ESM	1.72
FlexRay TU			
NCNB (Strongly Ordered) transaction with slave error response	User/Privilege	Interrupt => VIM	N/A
External imprecise error (Illegal transaction with ok response)	User/Privilege	Interrupt => VIM	N/A
Memory access permission violation	User/Privilege	ESM	1.16
Memory ECC uncorrectable error	User/Privilege	ESM	1.14
Memory ECC single bit error	User/Privilege	ESM	1.71
Ethernet master interface			
Any error reported by slave being accessed	User/Privilege	ESM	1.43
VIM			
Memory ECC uncorrectable error	User/Privilege	ESM	1.15
Memory ECC single bit error	User/Privilege	ESM	1.83
Voltage Monitor			
VMON out of voltage range	N/A	Reset	N/A
Self-Test (LBIST)			
Cortex-R5F CPU self-test (LBIST) error	User/Privilege	ESM	1.27
NHET Self-test (LBIST) error	User/Privilege	ESM	1.54
IOMM (terminal multiplexing control)			
Mux configuration error	User/Privilege	ESM	1.37
Power Domain Control			
Power Domain control access privilege error	User	Imprecise Abort (CPU)	N/A

Table 6-46. Reset/Abort/Error Sources (continued)

ERROR SOURCE	SYSTEM MODE	ERROR RESPONSE	ESM HOOKUP GROUP.CHANNEL
PSCON compare error	User/Privilege	ESM	1.38
PSCON self-test error	User/Privilege	ESM	1.39
Efuse farm			
eFuse farm autoload error	User/Privilege	ESM	3.1
eFuse farm error	User/Privilege	ESM	1.40
eFuse farm self-test error	User/Privilege	ESM	1.41
WIndowed Watchdog			
WWD Nonmaskable Interrupt Exception	N/A	ESM	2.24
Errors Reflected in the SYSESR Register			
Power-Up Reset	N/A	Reset	N/A
Oscillator fail / PLL slip(2)	N/A	Reset	N/A
Watchdog exception	N/A	Reset	N/A
CPUx Reset	N/A	Reset	N/A
Software Reset	N/A	Reset	N/A
External Reset	N/A	Reset	N/A
Register Soft Error	User/Privilege	ESM	1.90
CPU Interconnect Subsystem			
Diagnostic error	User/Privilege	ESM => Error terminal	3.12
Global error	User/Privilege	ESM	1.52
Global Parity error	User/Privilege	ESM	1.53
NMPU for EMAC			
MPU Access violation error	User/Privilege	ESM	1.55
NMPU for PS_SCR_S			
MPU Access violation error	User/Privilege	ESM	1.61
NMPU for DMA Port A			
MPU Access violation error	User/Privilege	ESM	1.69
PCR1			
MasterID filtering MPU Access violation error	User/Privilege	Bus Error	N/A
PCR2			
MasterID filtering MPU Access violation error	User/Privilege	Bus Error	N/A
PCR3			
MasterID filtering MPU Access violation error	User/Privilege	Bus Error	N/A

(2) Oscillator fail/PLL slip can be configured in the system register (SYS.PLLCTL1) to generate a reset.

6.21 Digital Windowed Watchdog

This device includes a Digital Windowed Watchdog (DWWD) module that protects against runaway code execution (see Figure 6-22).

The DWWD module allows the application to configure the time window within which the DWWD module expects the application to service the watchdog. A watchdog violation occurs if the application services the watchdog outside of this window, or fails to service the watchdog at all. The application can choose to generate a system reset or a nonmaskable interrupt to the CPU in case of a watchdog violation.

The watchdog is disabled by default and must be enabled by the application. Once enabled, the watchdog can only be disabled upon a system reset.

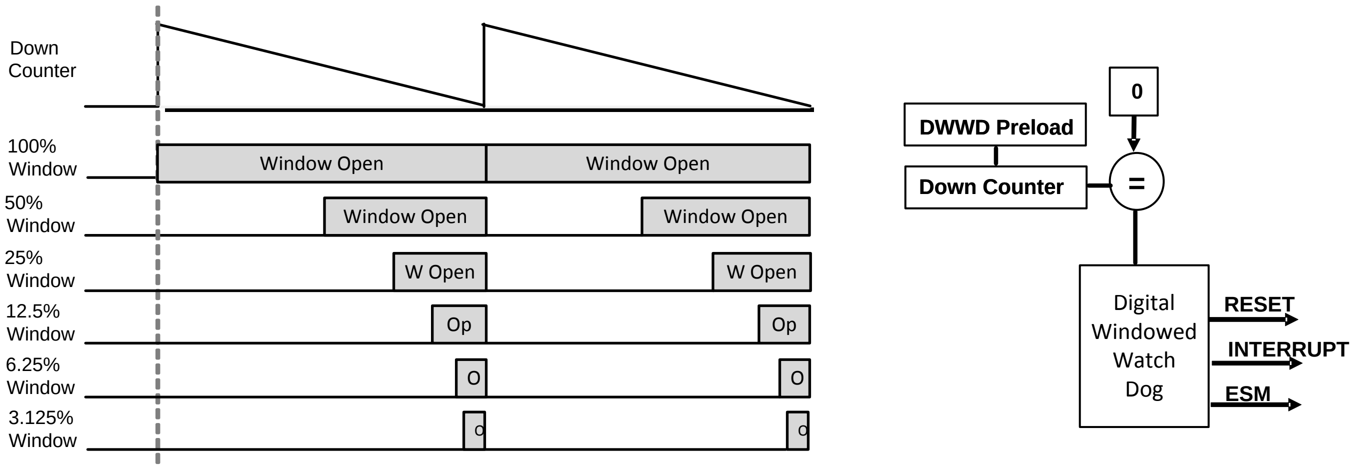


Figure 6-22. Digital Windowed Watchdog Example

6.22 Debug Subsystem

6.22.1 Block Diagram

The device contains an ICEPICK module (version C) to allow JTAG access to the scan chains (see Figure 6-23).

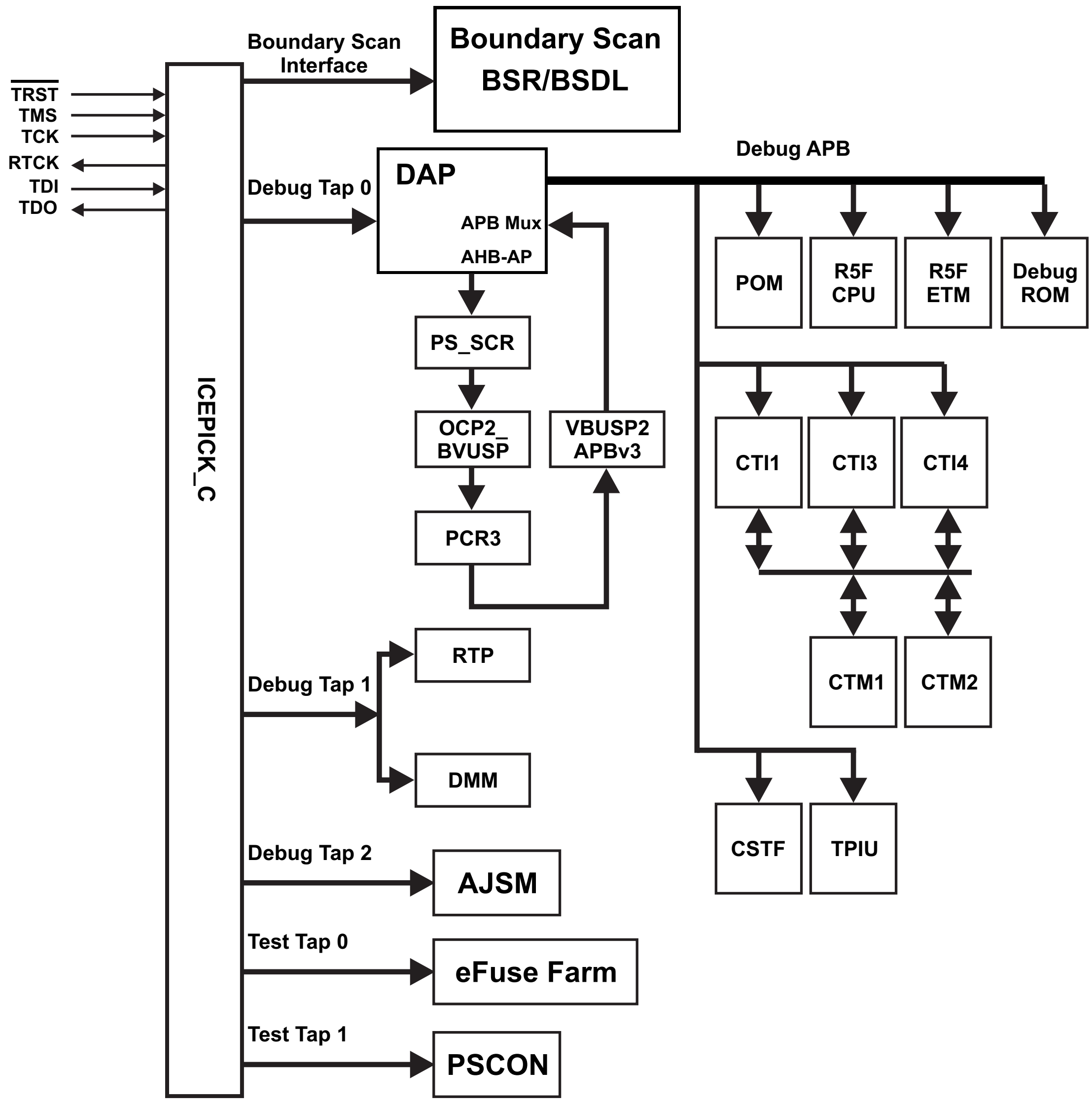


Figure 6-23. Debug Subsystem Block Diagram

6.22.2 Debug Components Memory Map

Table 6-47. Debug Components Memory Map

MODULE NAME	FRAME CHIP SELECT	FRAME ADDRESS RANGE		FRAME SIZE	ACTUAL SIZE	RESPONSE FOR ACCESS TO UNIMPLEMENTED LOCATIONS IN FRAME
		START	END			
CoreSight Debug ROM	CSCS0	0xFFA0_0000	0xFFA0_0FFF	4KB	4KB	Reads return zeros, writes have no effect
Cortex-R5F Debug	CSCS1	0xFFA0_1000	0xFFA0_1FFF	4KB	4KB	Reads return zeros, writes have no effect
ETM-R5	CSCS2	0xFFA0_2000	0xFFA0_2FFF	4KB	4KB	Reads return zeros, writes have no effect
CoreSight TPIU	CSCS3	0xFFA0_3000	0xFFA0_3FFF	4KB	4KB	Reads return zeros, writes have no effect
POM	CSCS4	0xFFA0_4000	0xFFA0_4FFF	4KB	4KB	Reads return zeros, writes have no effect
CTI1	CSCS7	0xFFA0_7000	0xFFA0_7FFF	4KB	4KB	Reads return zeros, writes have no effect
CTI3	CSCS9	0xFFA0_9000	0xFFA0_9FFF	4KB	4KB	Reads return zeros, writes have no effect
CTI4	CSCS10	0xFFA0_A000	0xFFA0_AFFF	4KB	4KB	Reads return zeros, writes have no effect
CSTF	CSCS11	0xFFA0_B000	0xFFA0_BFFF	4KB	4KB	Reads return zeros, writes have no effect

6.22.3 Embedded Cross Trigger

The Embedded Cross Trigger (ECT) is a modular component that supports the interaction and synchronization of multiple triggering events within a SoC.

The ECT consists of two modules:

- A (Cross Trigger Interface) CTI. The CTI provides the interface between a component or subsystem and the Cross Trigger Matrix (CTM).
- A CTM. The CTM combines the trigger requests generated from CTIs and broadcasts them to all CTIs as channel triggers. This enables subsystems to interact, cross trigger, with one another.

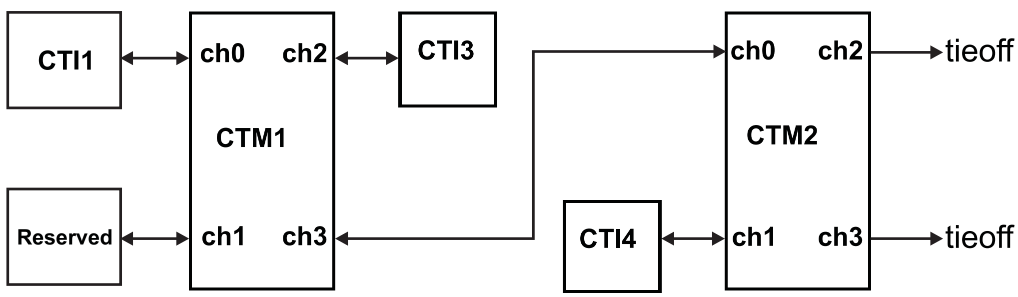


Figure 6-24. CTI/CTM Integration

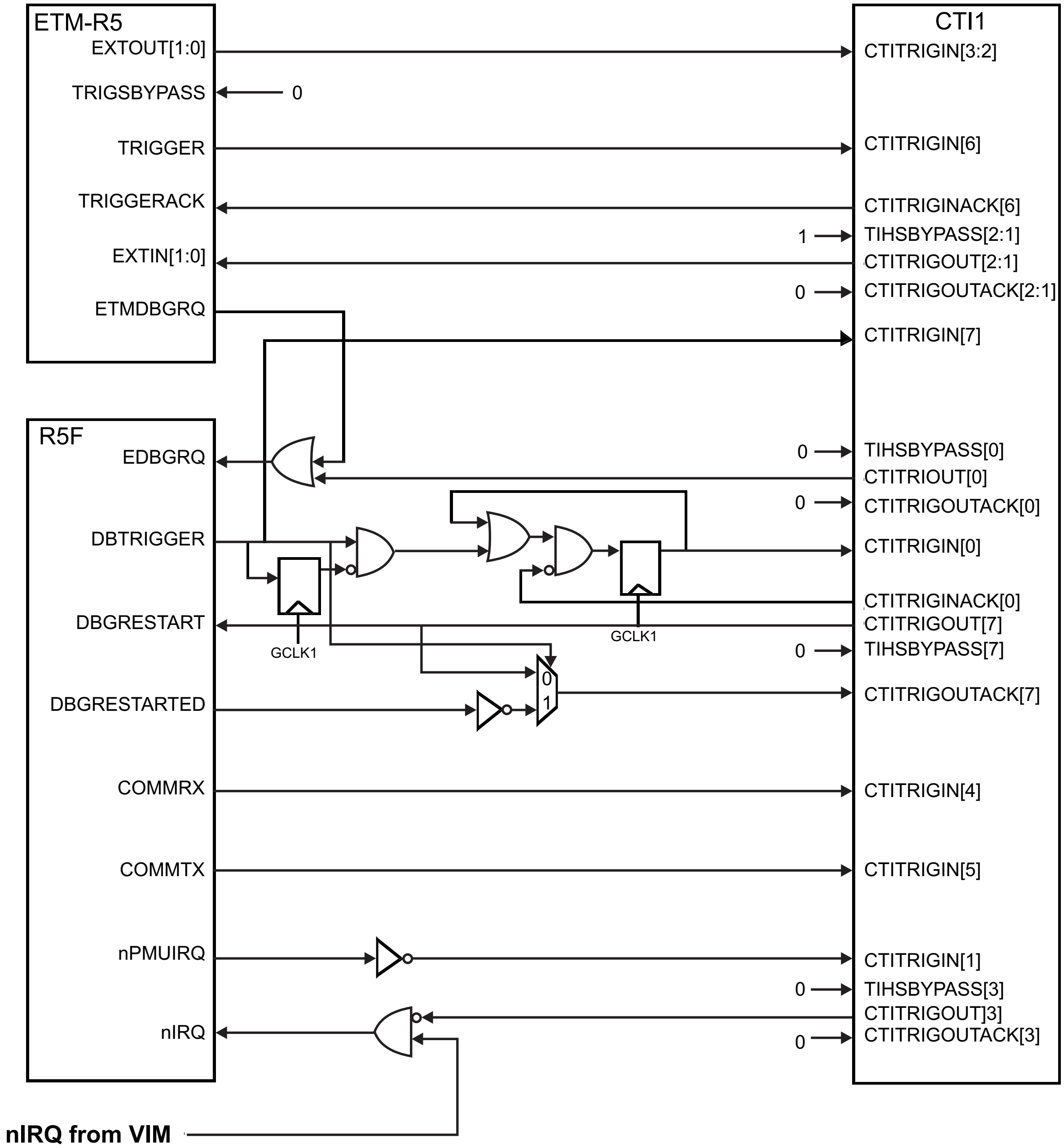


Figure 6-25. CTI1 Mapping

NOTE

ETM-R5, Cortex-R5F and CTI1 run at same frequency.

Table 6-48. CTI1 Mapping

CTI TRIGGER	Module Signal
Trigger Input 0	From Cortex-R5F DBTRIGGER
Trigger Input 1	From Cortex-R5F nPMUIRQ
Trigger Input 2	From ETM-R5 EXTOUT[0]
Trigger Input 3	From ETM-R5 EXTOUT[1]
Trigger Input 4	From Cortex-R5F COMMRX
Trigger Input 5	From Cortex-R5F COMMTX
Trigger Input 6	From ETM-R5 TRIGGER
Trigger Input 7	From Cortex-R5F DBTRIGGER
Trigger Output 0	To Cortex-R5F EDBGRQ
Trigger Output 1	To ETM-R5 EXTIN[0]
Trigger Output 2	To ETM-R5 EXTIN[1]
Trigger Output 3	To Cortex-R5F nIRQ
Trigger Output 4	Reserved
Trigger Output 5	Reserved
Trigger Output 6	Reserved
Trigger Output 7	To Cortex-R5F DBGRESTARTED

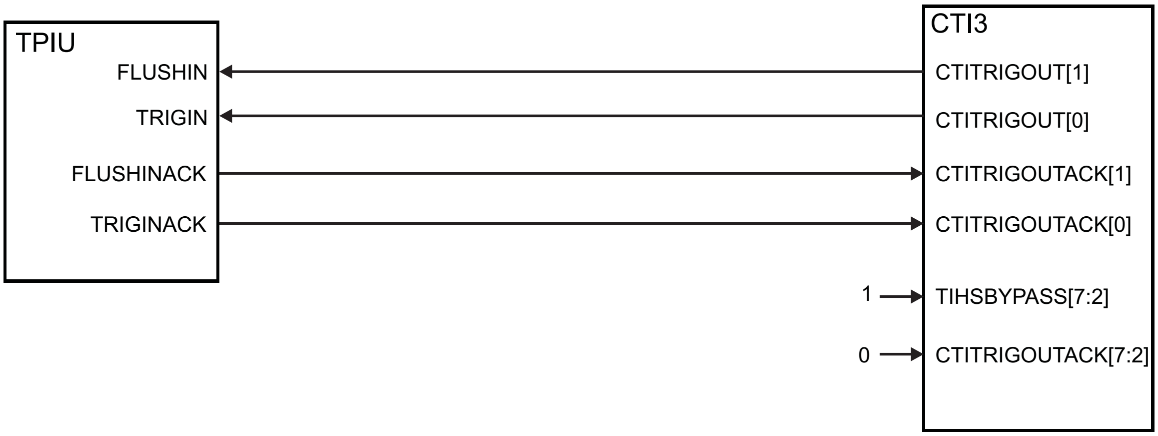


Figure 6-26. CTI3 Mapping

NOTE

TPIU and CTI3 run at different frequencies.

Table 6-49. CTI3 Mapping

CTI TRIGGER	Module Signal
Trigger Input 0	Reserved
Trigger Input 1	Reserved
Trigger Input 2	Reserved
Trigger Input 3	Reserved
Trigger Input 4	Reserved
Trigger Input 5	Reserved
Trigger Input 6	Reserved
Trigger Input 7	Reserved
Trigger Output 0	To TPIU TRIGIN
Trigger Output 1	To TPIU FLUSHIN
Trigger Output 2	Reserved
Trigger Output 3	Reserved
Trigger Output 4	Reserved
Trigger Output 5	Reserved
Trigger Output 6	Reserved
Trigger Output 7	Reserved

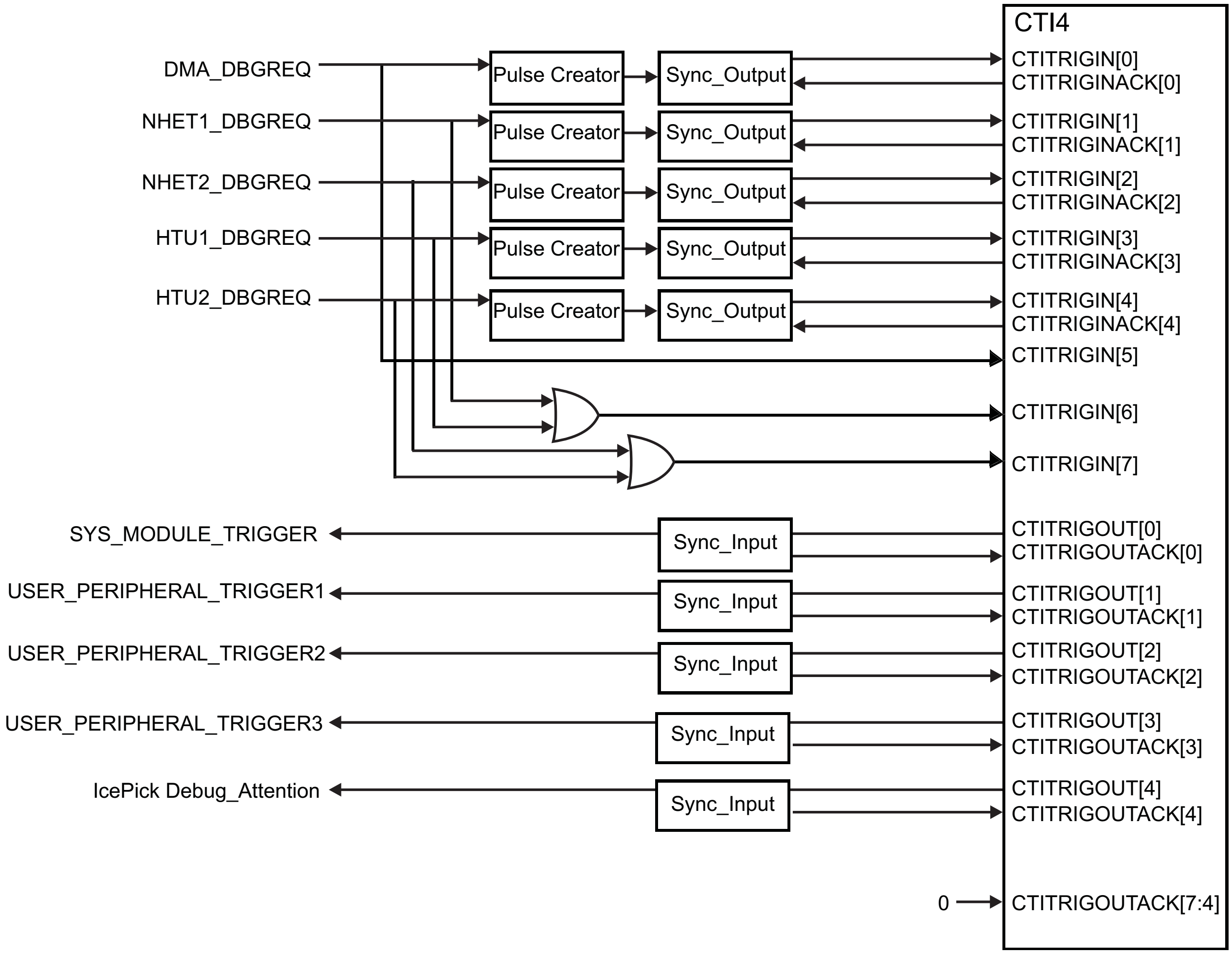


Figure 6-27. CTI4 Mapping

Table 6-50. CTI4 Mapping

CTI TRIGGER	Module Signal
Trigger Input 0	From DMA_DBGREQ
Trigger Input 1	From N2HET1_DBGREQ
Trigger Input 2	From N2HET2_DBGREQ
Trigger Input 3	From HTU1_DBGREQ
Trigger Input 4	From HTU2_DBGREQ
Trigger Input 5	From DMA_DBGREQ
Trigger Input 6	From N2HET1_DBGREQ or HTU1_DBGREQ
Trigger Input 7	From N2HET2_DBGREQ or HTU2_DBGREQ
Trigger Output 0	To SYS_MODULE_TRIGGER
Trigger Output 1	To USER_PERIPHERAL_TRIGGER1
Trigger Output 2	To USER_PERIPHERAL_TRIGGER2
Trigger Output 3	To USER_PERIPHERAL_TRIGGER3
Trigger Output 4	To IcePick Debug_Attention
Trigger Output 5	Reserved
Trigger Output 6	Reserved
Trigger Output 7	Reserved

Table 6-51. Peripheral Suspend Generation

TRIGGER OUTPUT	MODULE SIGNAL CONNECTED	DESCRIPTION
SYS_MODULE_TRIGGER	L2FMC_CPU_EMUSUSP	L2FMC Wrapper Suspend
	CCM_R5_CPU_EMUSUSP	CCM_R5 module suspend
	CRC_CPU_EMUSUSP	CRC1 / CRC2 module suspend
	SYS_CPU_EMUSUSP	SYS module Suspend
USER_PERIPHERAL_TRIGGER1	DMA_SUSPEND	DMA Suspend
	RTI_CPU_SUSPEND	RTI1 / RTI2 Suspend
	AWM_CPU_SUSPEND	AWM1 / AWM2 Suspend
	HTU_CPU_EMUSUSP	HTU1 / HTU2 Suspend
	SCI_CPU_EMUSUSP	SCI3 / SCI4 Suspend
	LIN_CPU_EMUSUSP	LIN1 / LIN2 Suspend
	I2C_CPU_EMUSUSP	I2C1 / I2C2 Suspend
	EMAC_CPU_EMUSUSP	EMAC Suspend
	EQEP_CPU_EMUSUSP	EQEP Suspend
	ECAP_CPU_EMUSUSP	ECAP Suspend
	DMM_CPU_EMUSUSP	DMM Suspend
	DCC_CPU_EMUSUSP	DCC1 / DCC2 Suspend
USER_PERIPHERAL_TRIGGER2	DCAN_CPU_EMUSUSP	DCAN1 / DCAN2 / DCAN3 / DCAN4 Suspend
USER_PERIPHERAL_TRIGGER3	ePWM_CPU_EMUSUSP	ePWM1..7 Trip Zone TZ6n and ePWM1..7 Suspend

6.22.4 JTAG Identification Code

The JTAG ID code for this device is the same as the device ICEPick Identification Code. For the JTAG ID Code per silicon revision, see Table 6-52.

Table 6-52. JTAG ID Code

SILICON REVISION	ID
Rev A	0x0B95A02F
Rev B	0x1B95A02F

6.22.5 Debug ROM

The Debug ROM stores the location of the components on the Debug APB bus (see Table 6-53).

Table 6-53. Debug ROM Table

ADDRESS	DESCRIPTION	VALUE
0x000	Cortex-R5F	0x00001003
0x004	ETM-R5	0x00002003
0x008	TPIU	0x00003003
0x00C	POM	0x00004003
0x018	CTI1	0x00007003
0x020	CTI3	0x00009003
0x024	CTI4	0x0000A003
0x028	CSTF	0x0000B003
0x02C	end of table	0x00000000

6.22.6 JTAG Scan Interface Timings

Table 6-54. JTAG Scan Interface Timing(1)

NO.	PARAMETER		MIN	MAX	UNIT
	fTCK	TCK frequency (at HCLKmax)		12	MHz
	fRTCK	RTCK frequency (at TCKmax and HCLKmax)	10		MHz
1	td(TCK -RTCK)	Delay time, TCK to RTCK		24	ns
2	tsu(TDI/TMS - RTCKr)	Setup time, TDI, TMS before RTCK rise (RTCKr)	26		ns
3	th(RTCKr -TDI/TMS)	Hold time, TDI, TMS after RTCKr	0		ns
4	th(RTCKr -TDO)	Hold time, TDO after RTCKf	0		ns
5	td(TCKf -TDO)	Delay time, TDO valid after RTCK fall (RTCKf)		12	ns

(1) Timings for TDO are specified for a maximum of 50-pF load on TDO.

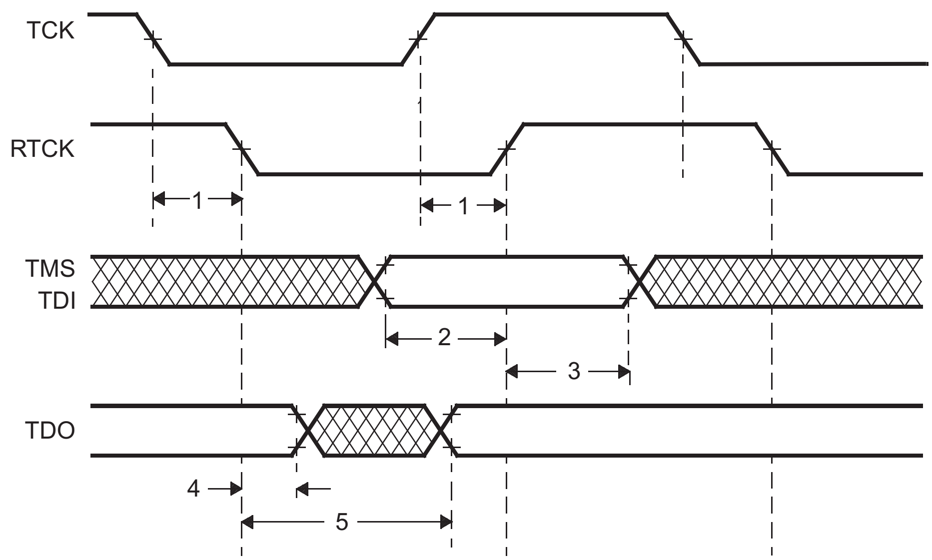


Figure 6-28. JTAG Timing

6.22.7 Advanced JTAG Security Module

This device includes a an Advanced JTAG Security Module (AJSM) module. The AJSM provides maximum security to the memory content of the device by letting users secure the device after programming.

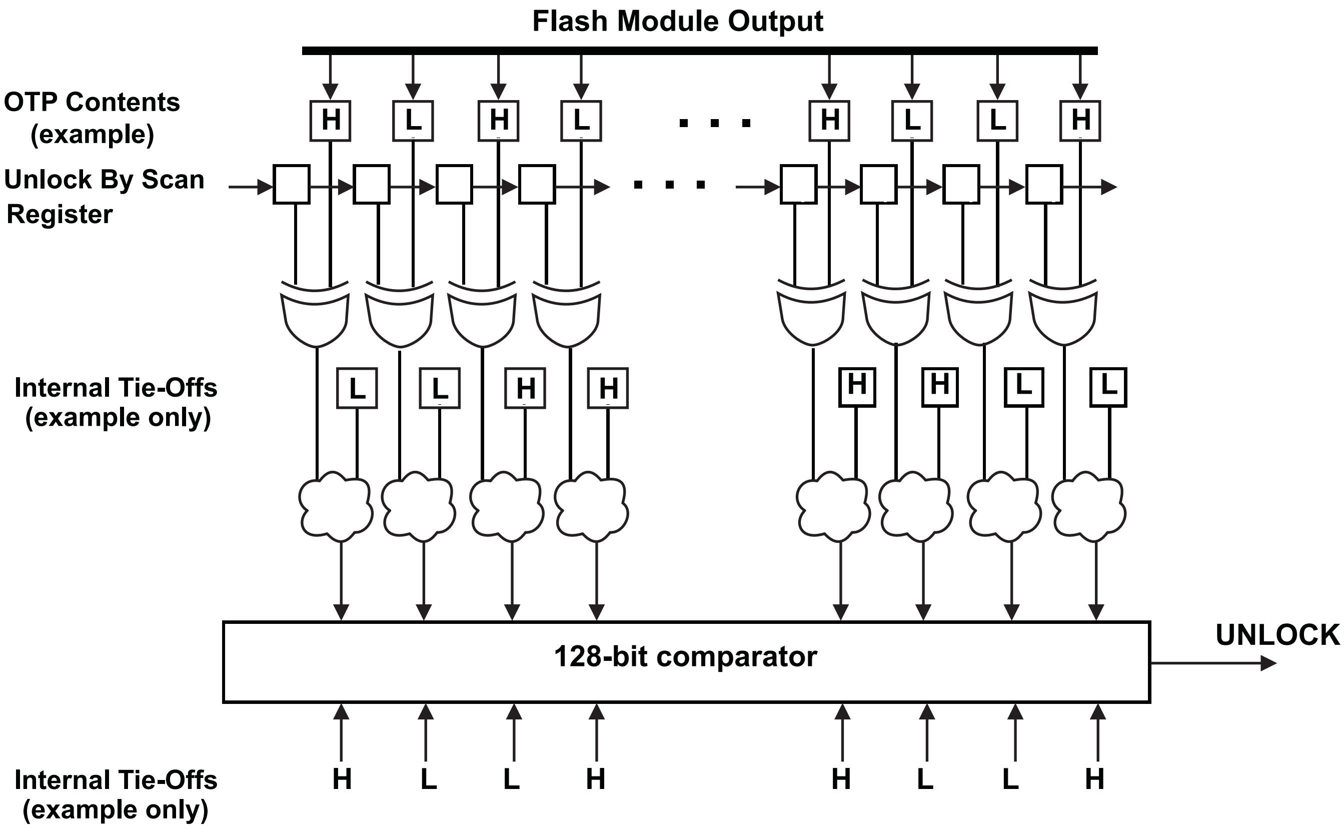


Figure 6-29. AJSM Unlock

The device is unsecure by default by virtue of a 128-bit visible unlock code programmed in the OTP address 0xF0000000. The OTP contents are XOR-ed with the contents of the "Unlock By Scan" register. The outputs of these XOR gates are again combined with a set of secret internal tie-offs. The output of this combinational logic is compared against a secret hard-wired 128-bit value. A match results in the UNLOCK signal being asserted, so that the device is now unsecure.

A user can secure the device by changing at least 1 bit in the visible unlock code from 1 to 0. Changing a 0 to 1 is not possible because the visible unlock code is stored in the One Time Programmable (OTP) flash region. Also, changing all 128 bits to zeros is not a valid condition and will permanently secure the device.

Once secured, a user can unsecure the device by scanning an appropriate value into the "Unlock By Scan" register of the AJSM. The value to be scanned is such that the XOR of the OTP contents and the Unlock-By-Scan register contents results in the original visible unlock code.

The Unlock-By-Scan register is reset only upon asserting power-on reset (nPORRST).

A secure device only permits JTAG accesses to the AJSM scan chain through the Secondary Tap 2 of the ICEPick module. All other secondary taps, test taps, and the boundary scan interface are not accessible in this state.

6.22.8 Embedded Trace Macrocell (ETM-R5)

The device contains a ETM-R5 module with a 32-bit internal data port. The ETM-R5 module is connected to a Trace Port Interface Unit (TPIU) with a 32-bit data bus. The TPIU provides a 35-bit (32-bit data, 3-bit control) external interface for trace. The ETM-R5 is CoreSight compliant and follows the ETM v3 specification. For more details, see the ARM CoreSight ETM-R5 TRM specification.

6.22.8.1 ETM TRACECLKIN Selection

The ETM clock source can be selected as either VCLK or the external ETMTRACECLKIN terminal. The selection is chosen by the EXTCTLOUT[1:0] control bits of the TPIU (default is '00'). The address of this register is the TPIU base address + 0x404.

Before the user begins accessing TPIU registers, the TPIU should be unlocked through the CoreSight key and 1 or 2 written to this register.

Table 6-55. TPIU / TRACECLKIN Selection

EXTCTLOUT[1:0]	TPIU/TRACECLKIN
00	Tied-zero
01	VCLK
10	ETMTRACECLKIN
11	Tied-zero

6.22.8.2 Timing Specifications

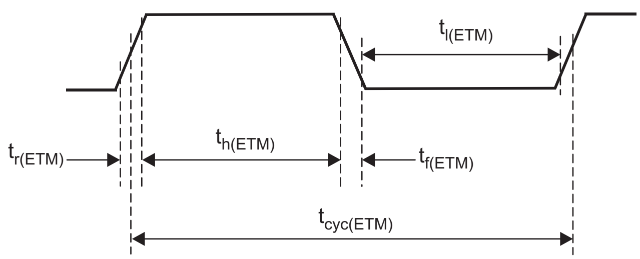


Figure 6-30. ETMTRACECLKOUT Timing

Table 6-56. ETMTRACECLK Timing

PARAMETER		MIN	MAX	UNIT
$t_{cyc(ETM)}$	Clock period	18.18		ns
$t_{l(ETM)}$	Low pulse width	6		ns
$t_{h(ETM)}$	High pulse width	6		ns
$t_{r(ETM)}$	Clock and data rise time		3	ns
$t_{f(ETM)}$	Clock and data fall time		3	ns

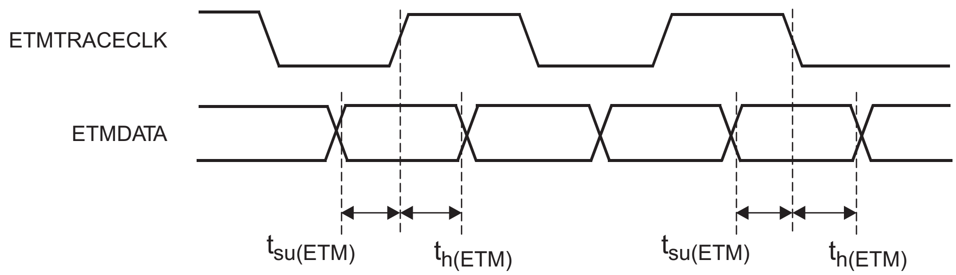


Figure 6-31. ETMDATA Timing

Table 6-57. ETMDATA Timing

PARAMETER		MIN	MAX	UNIT
$t_{su(ETM)}$	Data setup time	2.5		ns
$t_{h(ETM)}$	Data hold time	1.5		ns

NOTE

The ETMTRACECLK and ETMDATA timing is based on a 15-pF load and for ambient temperatures lower than 85°C.

6.22.9 RAM Trace Port (RTP)

The RTP provides the ability to datalog the RAM contents of the TMS570 devices or accesses to peripherals without program intrusion. It can trace all data write or read accesses to internal RAM. In addition, it provides the capability to directly transfer data to a FIFO to support a CPU-controlled transmission of the data. The trace data is transmitted over a dedicated external interface.

6.22.9.1 RTP Features

The RTP offers the following features:

- Two modes of operation - Trace Mode and Direct Data Mode
  - Trace Mode
    - Nonintrusive data trace on write or read operation
    - Visibility of RAM content at any time on external capture hardware
    - Trace of peripheral accesses
    - 2 configurable trace regions for each RAM module to limit amount of data to be traced
    - FIFO to store data and address of data of multiple read/write operations
    - Trace of CPU and/or DMA accesses with indication of the master in the transmitted data packet
  - Direct Data Mode
    - Directly write data with the CPU or trace read operations to a FIFO, without transmitting header and address information
- Dedicated synchronous interface to transmit data to external devices
- Free-running clock generation or clock stop mode between transmissions
- Up to 100 Mbps terminal transfer rate for transmitting data
- Pins not used in functional mode can be used as GIOs

6.22.9.2 Timing Specifications

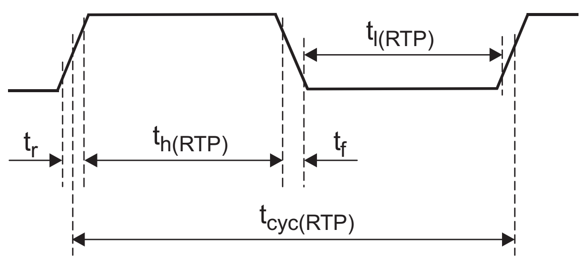


Figure 6-32. RTPCLK Timing

Table 6-58. RTPCLK Timing

PARAMETER		MIN	MIN	UNIT
$t_{cyc(RTP)}$	Clock period	9.09 (= 110 MHz)		ns
$t_{h(RTP)}$	High pulse width	$((t_{cyc(RTP)})/2) - ((t_r+t_f)/2)$		ns
$t_{l(RTP)}$	Low pulse width	$((t_{cyc(RTP)})/2) - ((t_r+t_f)/2)$		ns

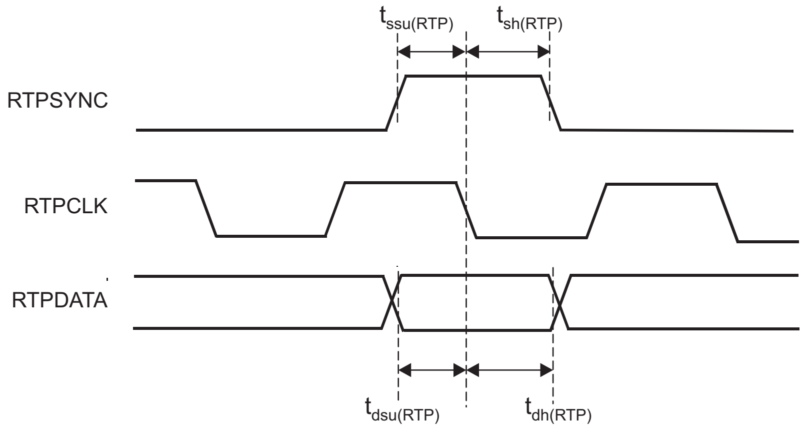


Figure 6-33. RTPDATA Timing

Table 6-59. RTPDATA Timing

PARAMETER		MIN	MAX	UNIT
$t_{dsu(RTP)}$	Data setup time	3		ns
$t_{dh(RTP)}$	Data hold time	1		ns
$t_{ssu(RTP)}$	SYNC setup time	3		ns
$t_{sh(RTP)}$	SYNC hold time	1		ns

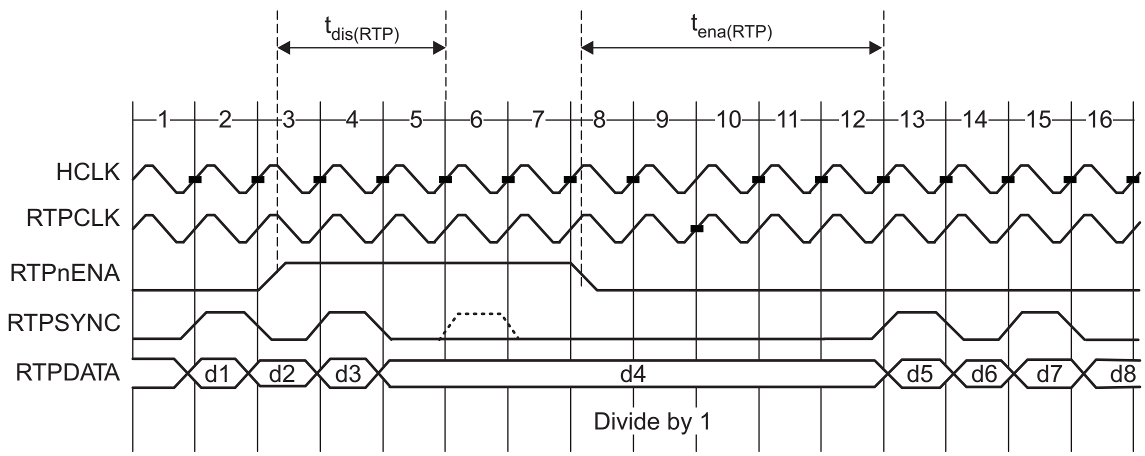


Figure 6-34. RTPnENA timing

Table 6-60. RTPnENA timing

PARAMETER		MIN	MAX	UNIT
$t_{dis(RTP)}$	Disable time, time RTPnENA must go high before what would be the next RTPSYNC, to ensure delaying the next packet	$3t_{c(HCLK)} + t_{r(RTPSYNC)} + 12$		ns
$t_{ena(RTP)}$	Enable time, time after RTPnENA goes low before a packet that has been halted, resumes	$4t_{c(HCLK)} + t_{r(RTPSYNC)}$	$5t_{c(HCLK)} + t_{r(RTPSYNC)} + 12$	ns

6.22.10 Data Modification Module (DMM)

The Data Modification Module (DMM) provides the capability to modify data in the entire 4GB address space of the TMS570devices from an external peripheral, with minimal interruption of the application.

6.22.10.1 DMM Features

The DMM module has the following features:

- Acts as a bus master, enabling direct writes to the 4GB address space without CPU intervention
- Writes to memory locations specified in the received packet (leverages packets defined by trace mode of the RAM Trace Port (RTP) module
- Writes received data to consecutive addresses, which are specified by the DMM module (leverages packets defined by direct data mode of the RTP module)
- Configurable port width (1-, 2-, 4-, 8-, 16-pins)
- Up to 100 Mbps terminal data rate
- Unused pins configurable as GIO pins

6.22.10.2 Timing Specifications

Table 6-61. DMMCLK Timing (see Figure 6-35)

PARAMETER		MIN	MAX	UNIT
$t_{cyc(DMM)}$	Cycle time, DMMCLK clock period	9.09		ns
$t_{h(DMM)}$	High-pulse width	$((t_{cyc(DMM)})/2) - ((t_r+t_f)/2)$		ns
$t_{l(DMM)}$	Low-pulse width	$((t_{cyc(DMM)})/2) - ((t_r+t_f)/2)$		ns

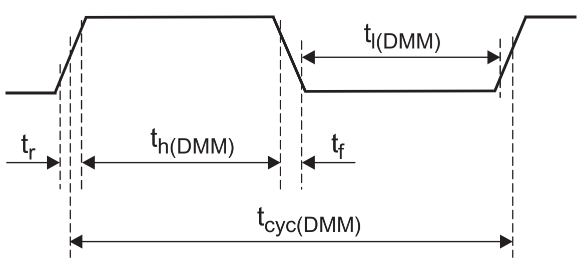


Figure 6-35. DMMCLK Timing

Table 6-62. DMMDATA Timing (see Figure 6-36)

PARAMETER		MIN	MAX	UNIT
$t_{ssu(DMM)}$	Setup time, SYNC active before clk falling edge	2		ns
$t_{sh(DMM)}$	Hold time, clk falling edge after SYNC deactive	3		ns
$t_{dsu(DMM)}$	Setup time, DATA before clk falling edge	2		ns
$t_{dh(DMM)}$	Hold time, clk falling edge after DATA hold time	3		ns

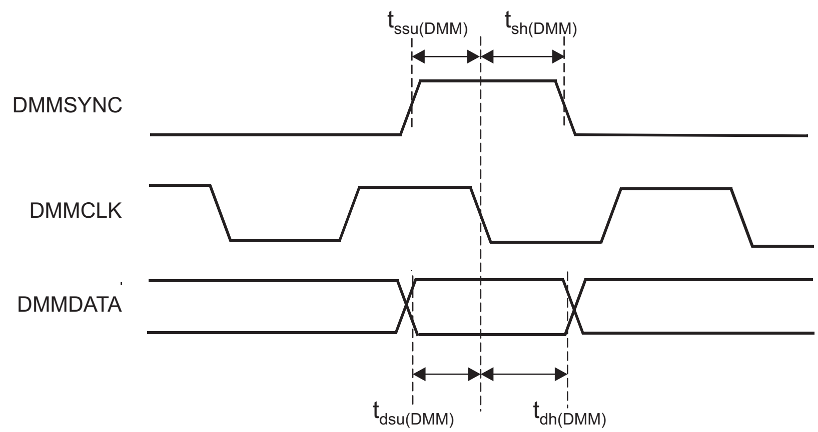


Figure 6-36. DMMDATA Timing

Figure 6-37 shows a case with 1 DMM packet per 2 DMMCLK cycles (Mode = Direct Data Mode, data width = 8, portwidth = 4) where none of the packets received by the DMM are sent out, leading to filling up of the internal buffers. The DMMnENA signal is shown asserted, after the first two packets have been received and synchronized to the HCLK domain. Here, the DMM has the capacity to accept packets D4x, D5x, D6x, D7x. Packet D8 would result in an overflow. Once DMMnENA is asserted, the DMM expects to stop receiving packets after 4 HCLK cycles; once DMMnENA is deasserted, the DMM can handle packets immediately (after 0 HCLK cycles).

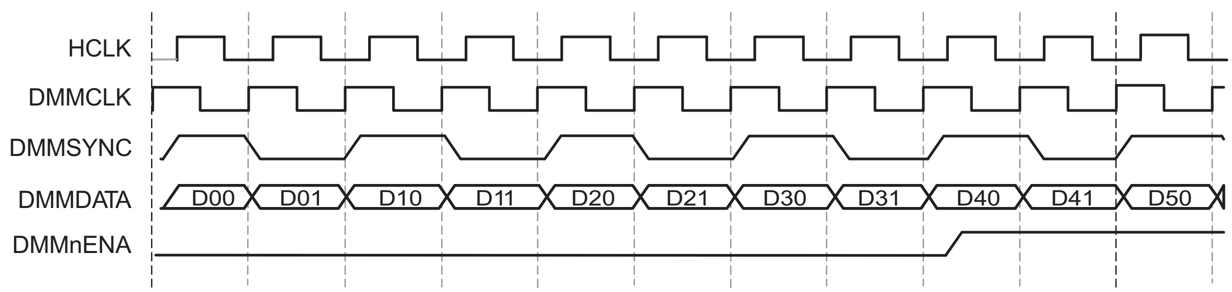


Figure 6-37. DMMnENA Timing

6.22.11 Boundary Scan Chain

The device supports BSDL-compliant boundary scan for testing pin-to-pin compatibility. The boundary scan chain is connected to the Boundary Scan Interface of the ICEPICK module (see Figure 6-38).

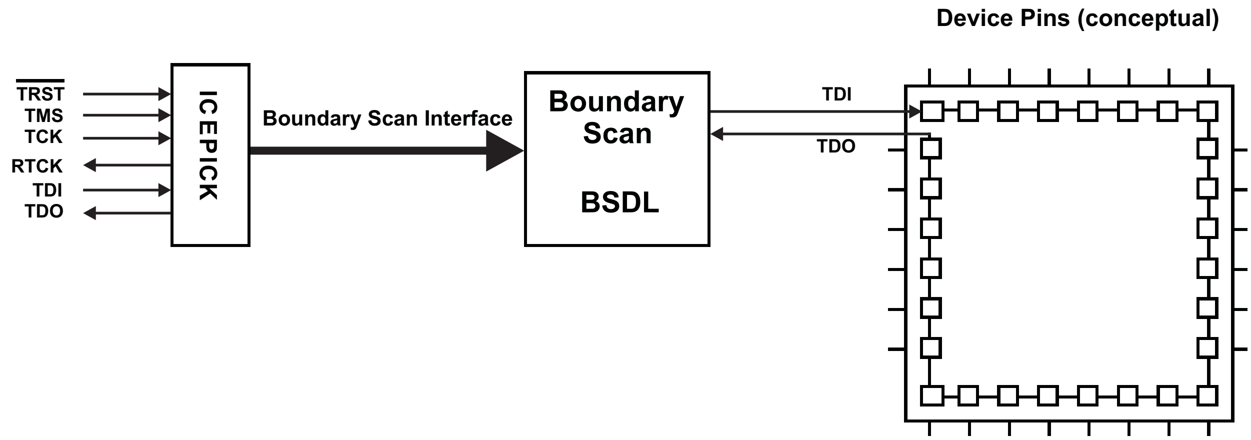


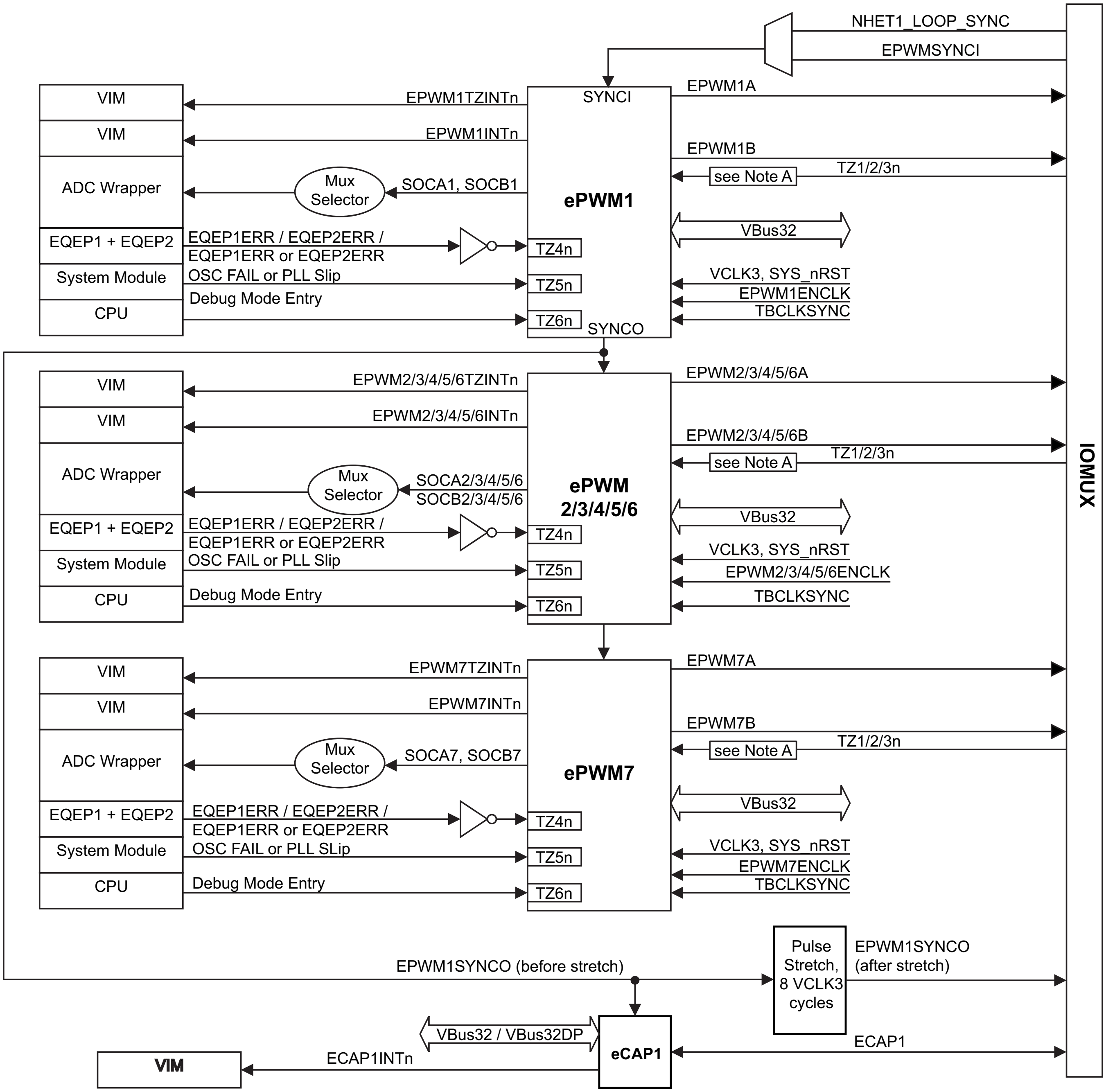
Figure 6-38. Boundary Scan Implementation (Conceptual Diagram)

Data is serially shifted into all boundary-scan buffers through TDI, and out through TDO.

7 Peripheral Information and Electrical Specifications

7.1 Enhanced Translator PWM Modules (ePWM)

Figure 7-1 shows the connections between the seven ePWM modules (ePWM1–ePWM7) on the device.



A. For more detail on the ePWMx input synchronization selection, see Figure 7-2.

Figure 7-1. ePWMx Module Interconnections

Figure 7-2 shows the detailed input synchronization selection (asynchronous, double-synchronous, or double synchronous + filter width) for ePWMx.

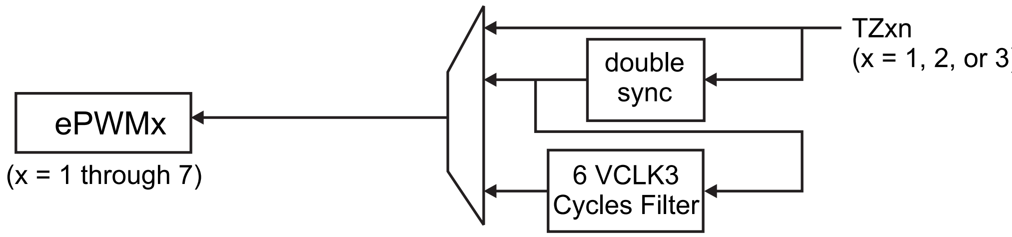


Figure 7-2. ePWMx Input Synchronization Selection Detail

7.1.1 ePWM Clocking and Reset

Each ePWM module has a clock enable (ePWMxENCLK) which is controlled by its respective Peripheral Power Down bit in the PSPWRDWNCLRx register of the PCR2 module. To properly reset the peripherals, the peripherals must be released from reset by setting the PENA bit of the CLKCNTL register in the system module. In additional, the peripherals must be released from their power down state by clearing their respective bit in the PSPWRDWNCLRx register. By default after reset, the peripherals are in powerdown state.

Table 7-1. ePWMx Clock Enable Control

ePWM MODULE INSTANCE	CONTROL REGISTER TO ENABLE CLOCK	DEFAULT VALUE
ePWM1	PSPWRDWNCLR3[16]	1
ePWM2	PSPWRDWNCLR3[17]	1
ePWM3	PSPWRDWNCLR3[18]	1
ePWM4	PSPWRDWNCLR3[19]	1
ePWM5	PSPWRDWNCLR3[12]	1
ePWM6	PSPWRDWNCLR3[13]	1
ePWM7	PSPWRDWNCLR3[14]	1

7.1.2 Synchronization of ePWMx Time-Base Counters

A time-base synchronization scheme connects all of the ePWM modules on a device. Each ePWM module has a synchronization input (EPWMxSYNCI) and a synchronization output (EPWMxSYNCO). The input synchronization for the first instance (ePWM1) comes from an external pin. Figure 7-1 shows the synchronization connections for all the ePWMx modules. Each ePWM module can be configured to use or ignore the synchronization input. For more information, see the ePWM module chapter of the device-specific TRM.

7.1.3 Synchronizing all ePWM Modules to the N2HET1 Module Time Base

The connection between the NHET1_LOOP_SYNC and the SYNCI input of ePWM1 module is implemented as shown in Figure 7-3.

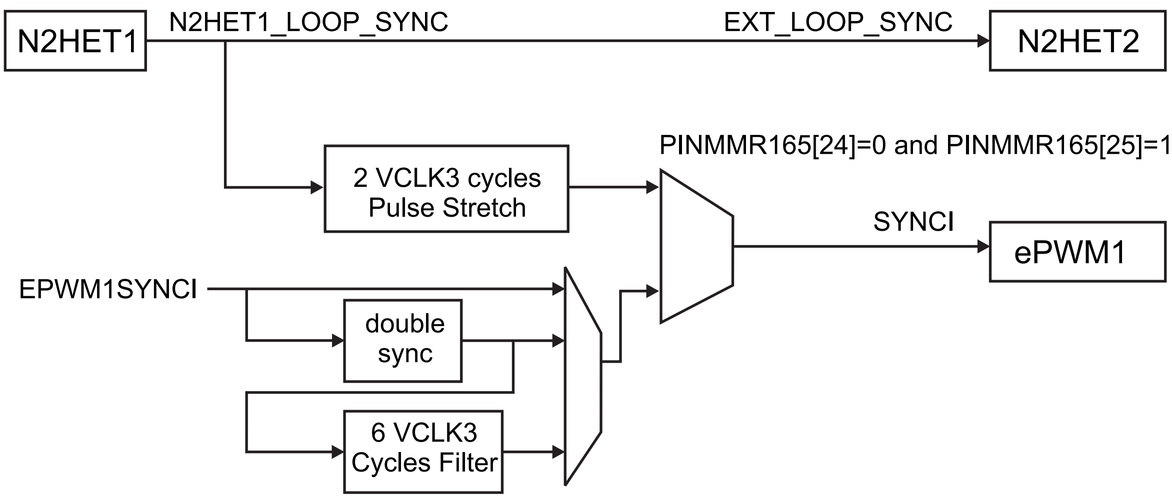


Figure 7-3. Synchronizing Time Bases Between N2HET1, N2HET2 and ePWMx Modules

7.1.4 Phase-Locking the Time-Base Clocks of Multiple ePWM Modules

The TBCLKSYNC bit can be used to globally synchronize the time-base clocks of all enabled ePWM modules on a device. This bit is implemented as PINMMR166[1] register bit 1.

When TBCLKSYNC = 0, the time-base clock of all ePWM modules is stopped. This is the default condition.

When TBCLKSYNC = 1, all ePWM time-base clocks are started with the rising edge of TBCLK aligned.

For perfectly synchronized TBCLKs, the prescaler bits in the TBCTL register of each ePWM module must be set identically. The proper procedure for enabling the ePWM clocks is as follows:

- Each ePWM is individually associated with a power down bit in the PSPWRDWNCLRx register of the PCR2 module. Enable the individual ePWM module clocks (if disable) using the control registers in the PCR2.
- Configure TBCLKSYNC = 0. This will stop the time-base clock within any enabled ePWM module.
- Configure the prescaler values and desired ePWM modes.
- Configure TBCLKSYNC = 1.

7.1.5 ePWM Synchronization with External Devices

The output sync from the ePWM1 module is also exported to the I/O Mux such that multiple devices can be synchronized together. The signal pulse must be stretched by 8 VCLK3 cycles before being exported on the IO Mux pin as the ePWMSYNCO signal.

7.1.6 ePWM Trip Zones

The ePWMx modules have 6 trip zone inputs each. These are active-low signals. The application can control the ePWMx module response to each of the trip zone input separately. The timing requirements from the assertion of the trip zone inputs to the actual response are specified in the electrical and timing section of this document.

7.1.6.1 Trip Zones TZ1n, TZ2n, TZ3n

These 3 trip zone inputs are driven by external circuits and are connected to device-level inputs. These signals are either connected asynchronously to the ePWMx trip zone inputs, or double-synchronized with VCLK3, or double-synchronized and then filtered with a 6-cycle VCLK3-based counter before connecting to the ePWMx (see Figure 7-2). By default, the trip zone inputs are asynchronously connected to the ePWMx modules.

Table 7-2. Connection to ePWMx Modules for Device-Level Trip Zone Inputs

TRIP ZONE INPUT	CONTROL FOR ASYNCHRONOUS CONNECTION TO ePWMx	CONTROL FOR DOUBLE-SYNCHRONIZED CONNECTION TO ePWMx	CONTROL FOR DOUBLE-SYNCHRONIZED AND FILTERED CONNECTION TO ePWMx (1)
TZ1n	PINMMR172[18:16] = 001	PINMMR172[18:16] = 010	PINMMR172[18:16] = 100
TZ2n	PINMMR172[26:24] = 001	PINMMR172[26:24] = 010	PINMMR172[26:24] = 100
TZ3n	PINMMR173[2:0] = 001	PINMMR173[2:0] = 010	PINMMR173[2:0] = 100

(1) The filter width is 6 VCLK3 cycles.

7.1.6.2 Trip Zone TZ4n

This trip zone input is dedicated to eQEPx error indications. There are 2 eQEP modules on this device. Each eQEP module indicates a phase error by driving its EQEPxERR output high. The following control registers allow the application to configure the trip zone input (TZ4n) to each ePWMx module based on the requirements of the applicationapplication's requirements.

Table 7-3. TZ4n Connections for ePWMx Modules

ePWMx	CONTROL FOR TZ4n = NOT(EQEP1ERR OR EQEP2ERR)	CONTROL FOR TZ4n = NOT(EQEP1ERR)	CONTROL FOR TZ4n = NOT(EQEP2ERR)
ePWM1	PINMMR167[2:0] = 001	PINMMR167[2:0] = 010	PINMMR167[2:0] = 100
ePWM2	PINMMR167[10:8] = 001	PINMMR167[10:8] = 010	PINMMR167[10:8] = 100
ePWM3	PINMMR167[18:16] = 001	PINMMR167[18:16] = 010	PINMMR167[18:16] = 100
ePWM4	PINMMR167[26:24] = 001	PINMMR167[26:24] = 010	PINMMR167[26:24] = 100
ePWM5	PINMMR168[2:0] = 001	PINMMR168[2:0] = 010	PINMMR168[2:0] = 100
ePWM6	PINMMR168[10:8] = 001	PINMMR168[10:8] = 010	PINMMR168[10:8] = 100
ePWM7	PINMMR168[18:16] = 001	PINMMR168[18:16] = 010	PINMMR168[18:16] = 100

NOTE

The EQEPxERR signal is an active high signal coming out of EQEPx module. As listed in Table 7-3, the selected combination of the EQEPxERR signals must be inverted before connecting to the TZ4n input of the ePWMx modules.

7.1.6.3 Trip Zone TZ5n

This trip zone input is dedicated to a clock failure on the device. That is, this trip zone input is asserted whenever an oscillator failure or a PLL slip is detected on the device. The applciation can use this trip zone input for each ePWMx module to prevent the external system from going out of control when the device clocks are not within expected range (system running at limp clock).

The oscillator failure and PLL slip signals used for this trip zone input are taken from the status flags in the system module. These level signals are set until cleared by the application.

7.1.6.4 Trip Zone TZ6n

This trip zone input to the ePWMx modules is dedicated to a debug mode entry of the CPU. If enabled, the user can force the PWM outputs to a known state when the emulator stops the CPU. This prevents the external system from going out of control when the CPU is stopped.

NOTE

There is a signal called DBGACK that the CPU drives when it enters debug mode. This signal must be inverted and used as the Debug Mode Entry signal for the trip zone input.

7.1.7 Triggering of ADC Start of Conversion Using ePWMx SOCA and SOCB Outputs

A special scheme is implemented to select the actual signal used for triggering the start of conversion on the two ADCs on this device. This scheme is defined in Section 7.4.2.3.

7.1.8 Enhanced Translator-Pulse Width Modulator (ePWMx) Electrical Data/Timing

Table 7-4. ePWMx Timing Requirements

		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(SYNCIN)}$	Synchronization input pulse width	Asynchronous	2 $t_{c(VCLK3)}$		cycles
		Synchronous	2 $t_{c(VCLK3)}$		cycles
		Synchronous with input filter	2 $t_{c(VCLK3)}$ + filter width(1)		cycles

(1) The filter width is 6 VCLK3 cycles.

Table 7-5. ePWMx Switching Characteristics

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(PWM)}$	Pulse duration, ePWMx output high or low		33.33		ns
$t_{w(SYNCOUT)}$	Synchronization Output Pulse Width		8 $t_{c(VCLK3)}$		cycles
$t_{d(PWM)tza}$	Delay time, trip input active to PWM forced high, OR Delay time, trip input active to PWM forced low	No pin load		25	ns
$t_{d(TZ-PWM)HZ}$	Delay time, trip input active to PWM Hi-Z			20	ns

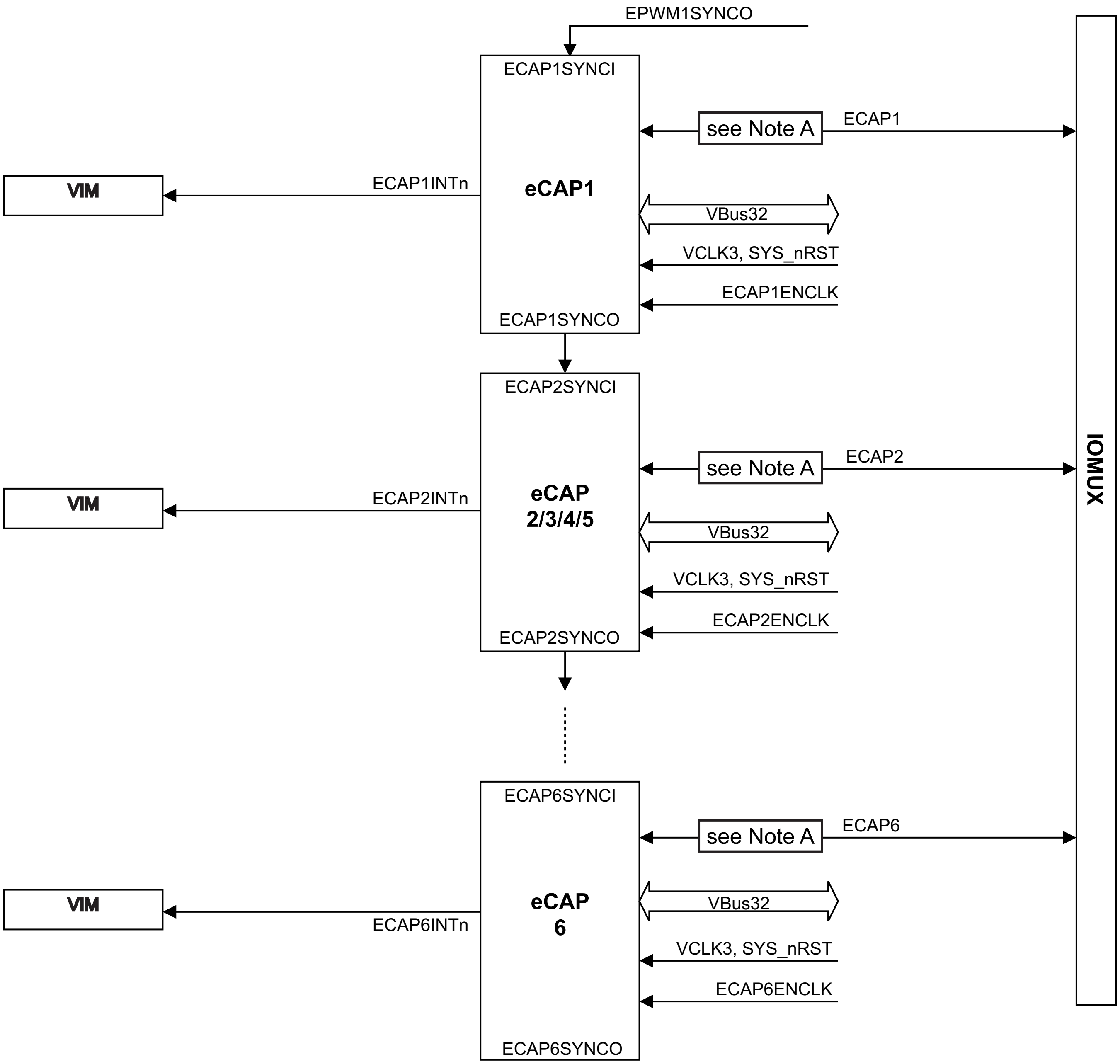
Table 7-6. ePWMx Trip-Zone Timing Requirements

		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(TZ)}$	Pulse duration, TZn input low	Asynchronous	2 * TBePWMx		cycles
		Synchronous	2 $t_{c(VCLK3)}$		
		Synchronous with input filter	2 $t_{c(VCLK3)}$ + filter width(1)		

(1) The filter width is 6 VCLK3 cycles.

7.2 Enhanced Capture Modules (eCAP)

Figure 7-4 shows how the eCAP modules are interconnected on this microcontroller.



A. For more detail on the eCAPx input synchronization selection, see Figure 7-5.

Figure 7-4. eCAP Module Connections

Figure 7-5 shows the detailed input synchronization selection (asynchronous, double-synchronous, or double synchronous + filter width) for eCAPx.

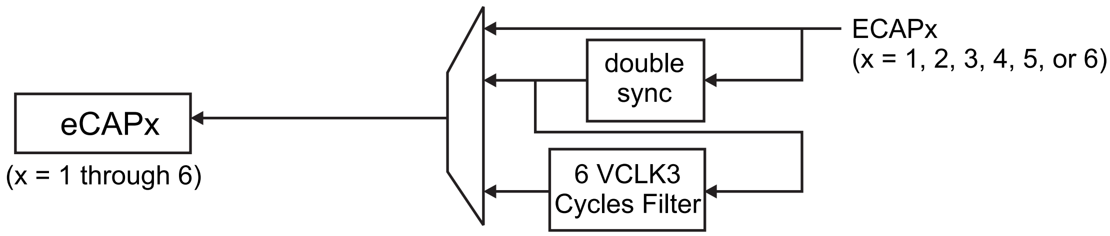


Figure 7-5. eCAPx Input Synchronization Selection Detail

7.2.1 Clock Enable Control for eCAPx Modules

Each of the eCAPx modules has a clock enable (ECAPxENCLK) which is controlled by its respective Peripheral Power Down bit in the PSPWRDWNCLRx register of the PCR2 module. To properly reset the peripherals, the peripherals must be released from reset by setting the PENA bit of the CLKCNTL register in the system module. In addition, the peripherals must be released from their power down state by clearing the respective bit in the PSPWRDWNCLRx register. By default, after reset, the peripherals are in the power down state.

Table 7-7. eCAPx Clock Enable Control

eCAP MODULE INSTANCE	CONTROL REGISTER TO ENABLE CLOCK	DEFAULT VALUE
eCAP1	PSPWRDWNCLR3[15]	1
eCAP2	PSPWRDWNCLR3[8]	1
eCAP3	PSPWRDWNCLR3[9]	1
eCAP4	PSPWRDWNCLR3[10]	1
eCAP5	PSPWRDWNCLR3[11]	1
eCAP6	PSPWRDWNCLR3[4]	1

7.2.2 PWM Output Capability of eCAPx

When not used in capture mode, each of the eCAPx modules can be used as a single-channel PWM output. This is called the Auxiliary PWM (APWM) mode of operation of the eCAPx modules. For more information, see the eCAP module chapter of the device-specific TRM.

7.2.3 Input Connection to eCAPx Modules

The input connection to each of the eCAPx modules can be selected between a double-VCLK3-synchronized input or a double-VCLK3-synchronized and filtered input, as listed in Table 7-8.

Table 7-8. Device-Level Input Connection to eCAPx Modules

INPUT SIGNAL	CONTROL FOR DOUBLE-SYNCHRONIZED CONNECTION TO eCAPx	CONTROL FOR DOUBLE-SYNCHRONIZED AND FILTERED CONNECTION TO eCAPx (1)
eCAP1	PINMMR169[2:0] = 001	PINMMR169[2:0] = 010
eCAP2	PINMMR169[10:8] = 001	PINMMR169[10:8] = 010
eCAP3	PINMMR169[18:16] = 001	PINMMR169[18:16] = 010
eCAP4	PINMMR169[26:24] = 001	PINMMR169[26:24] = 010
eCAP5	PINMMR170[2:0] = 001	PINMMR170[2:0] = 010
eCAP6	PINMMR170[10:8] = 001	PINMMR170[10:8] = 010

(1) The filter width is 6 VCLK3 cycles.

7.2.4 Enhanced Capture Module (eCAP) Electrical Data/Timing

Table 7-9. eCAPx Timing Requirements

		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(CAP)}$	Pulse width, capture input	Synchronous	2 $t_{c(VCLK3)}$		cycles
		Synchronous with input filter	2 $t_{c(VCLK3)}$ + filter width (1)		cycles

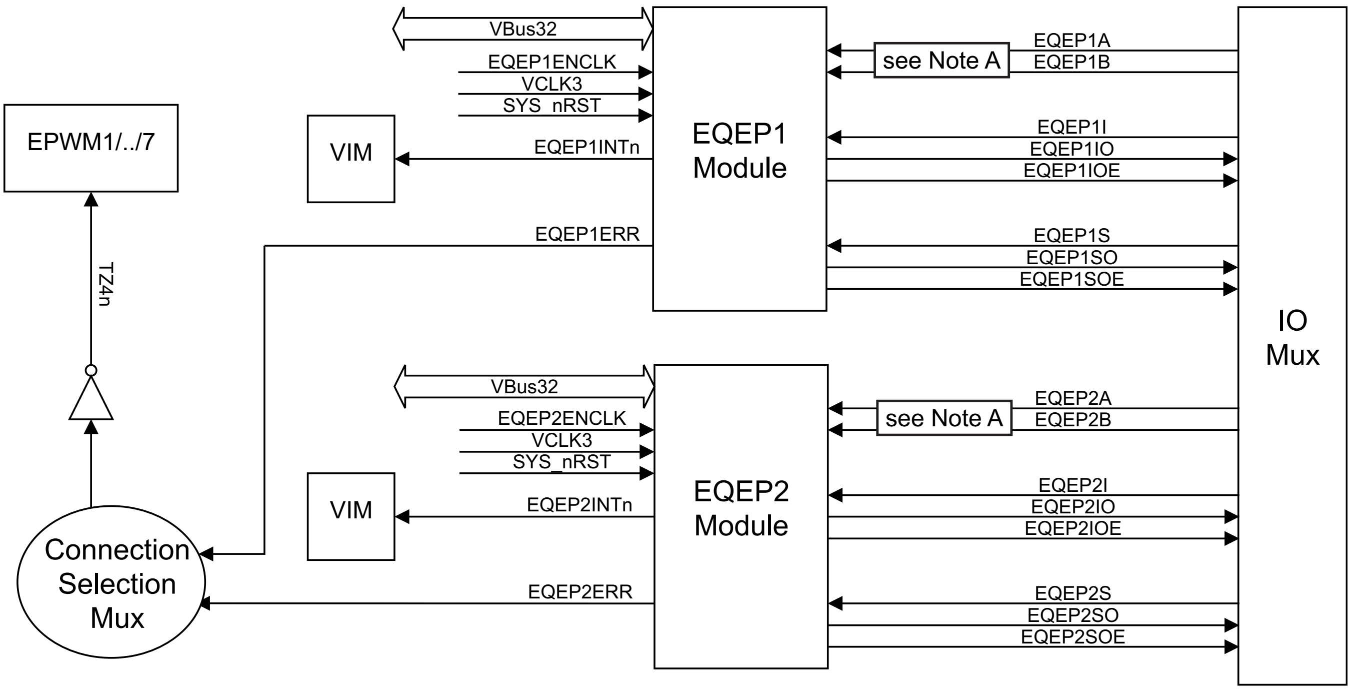
(1) The filter width is 6 VCLK3 cycles.

Table 7-10. eCAPx Switching Characteristics

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(APWM)}$	Pulse duration, APWMx output high or low		20		ns

7.3 Enhanced Quadrature Encoder (eQEP)

Figure 7-6 shows the eQEP module interconnections on the device.



A. For more detail on the eQEPx input synchronization selection, see Figure 7-7.

Figure 7-6. eQEP Module Interconnections

Figure 7-7 shows the detailed input synchronization selection (asynchronous, double-synchronous, or double synchronous + filter width) for eQEPx.

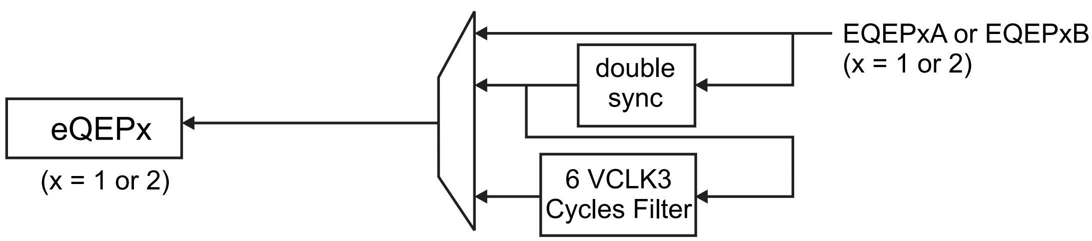


Figure 7-7. eQEPx Input Synchronization Selection Detail

7.3.1 Clock Enable Control for eQEPx Modules

Each of the EQEPx modules has a clock enable (EQEPxENCLK) which is controlled by its respective Peripheral Power Down bit in the PSPWRDWNCLRx register of the PCR2 module. To properly reset the peripherals, the peripherals must be released from reset by setting the PENA bit of the CLKCNTL register in the system module. In addition, the peripherals must be released from their power down state by clearing the respective bit in the PSPWRDWNCLRx register. By default after reset, the peripherals are in power down state.

Table 7-11. eQEPx Clock Enable Control

eQEP MODULE INSTANCE	CONTROL REGISTER TO ENABLE CLOCK	DEFAULT VALUE
eQEP1	PSPWRDWNCLR3[5]	1
eQEP2	PSPWRDWNCLR3[6]	1

7.3.2 Using eQEPx Phase Error to Trip ePWMx Outputs

The eQEP module sets the EQEPERR signal output whenever a phase error is detected in its inputs EQEPxA and EQEPxB. This error signal from both the eQEP modules is input to the connection selection multiplexer. This multiplexer is defined in Table 7-3. As shown in Figure 7-6, the output of this selection multiplexer is inverted and connected to the TZ4n trip-zone input of all ePWMx modules. This connection allows the application to define the response of each ePWMx module on a phase error indicated by the eQEP modules.

7.3.3 Input Connection to eQEPx Modules

The input connection to each of the eQEP modules can be selected between a double-VCLK3-synchronized input or a double-VCLK3-synchronized and filtered input, as listed in Table 7-12.

Table 7-12. Device-Level Input Connection to eQEPx Modules

INPUT SIGNAL	CONTROL FOR DOUBLE-SYNCHRONIZED CONNECTION TO eQEPx	CONTROL FOR DOUBLE-SYNCHRONIZED AND FILTERED CONNECTION(1) TO eQEPx
eQEP1A	PINMMR170[18:16] = 001	PINMMR170[18:16] = 010
eQEP1B	PINMMR170[26:24] = 001	PINMMR170[26:24] = 010
eQEP1I	PINMMR171[2:0] = 001	PINMMR171[2:0] = 010
eQEP1S	PINMMR171[10:8] = 001	PINMMR171[10:8] = 010
eQEP2A	PINMMR171[18:16] = 001	PINMMR171[18:16] = 010
eQEP2B	PINMMR171[26:24] = 001	PINMMR171[26:24] = 010
eQEP2I	PINMMR172[2:0] = 001	PINMMR172[2:0] = 010
eQEP2S	PINMMR172[10:8] = 001	PINMMR172[10:8] = 010

(1) The filter width is 6 VCLK3 cycles.

7.3.4 Enhanced Quadrature Encoder Pulse (eQEPx) Timing

Table 7-13. eQEPx Timing Requirements(1)

		TEST CONDITIONS	MIN	MAX	UNIT
$t_{w(QEPP)}$	QEP input period	Synchronous	2 $t_{c(VCLK3)}$		cycles
		Synchronous with input filter	2 $t_{c(VCLK3)}$ + filter width		
$t_{w(INDEXH)}$	QEP Index Input High Time	Synchronous	2 $t_{c(VCLK3)}$		cycles
		Synchronous with input filter	2 $t_{c(VCLK3)}$ + filter width		
$t_{w(INDEXL)}$	QEP Index Input Low Time	Synchronous	2 $t_{c(VCLK3)}$		cycles
		Synchronous with input filter	2 $t_{c(VCLK3)}$ + filter width		
$t_{w(STROBH)}$	QEP Strobe Input High Time	Synchronous	2 $t_{c(VCLK3)}$		cycles
		Synchronous with input filter	2 $t_{c(VCLK3)}$ + filter width		
$t_{w(STROBL)}$	QEP Strobe Input Low Time	Synchronous	2 $t_{c(VCLK3)}$		cycles
		Synchronous with input filter	2 $t_{c(VCLK3)}$ + filter width		

(1) The filter width is 6 VCLK3 cycles.

Table 7-14. eQEPx Switching Characteristics

PARAMETER		MIN	MAX	UNIT
$t_{d(CNTR)xin}$	Delay time, external clock to counter increment		4 $t_{c(VCLK3)}$	cycles
$t_{d(PCS-OUT)QEP}$	Delay time, QEP input edge to position compare sync output		6 $t_{c(VCLK3)}$	cycles

7.4 12-bit Multibuffered Analog-to-Digital Converter (MibADC)

The MibADC has a separate power bus for its analog circuitry that enhances the Analog-to-Digital (A-to-D) performance by preventing digital switching noise on the logic circuitry which could be present on V_{SS} and V_{CC} from coupling into the A-to-D analog stage. All A-to-D specifications are given with respect to AD_{REFLO}, unless otherwise noted.

Table 7-15. MibADC Overview

DESCRIPTION	VALUE
Resolution	12 bits
Monotonic	Assured
Output conversion code	00h to FFFh [00 for $V_{AI} \leq AD_{REFLO}$; FFF for $V_{AI} \geq AD_{REFHI}$]

7.4.1 *MibADC Features*

- 10-/12-bit resolution
- AD_{REFHI} and AD_{REFLO} pins (high and low reference voltages)
- Total Sample/Hold/Convert time: 600 ns Typical Minimum at 30 MHz ADCLK
- One memory region per conversion group is available (Event Group, Group 1, and Group 2)
- Allocation of channels to conversion groups is completely programmable
- Memory regions are serviced either by interrupt or by DMA
- Programmable interrupt threshold counter is available for each group
- Programmable magnitude threshold interrupt for each group for any one channel
- Option to read either 8-, 10-, or 12-bit values from memory regions
- Single or continuous conversion modes
- Embedded self-test
- Embedded calibration logic
- Enhanced power-down mode
  – Optional feature to automatically power down ADC core when no conversion is in progress
- External event pin (ADEVT) programmable as general-purpose I/O

7.4.2 *Event Trigger Options*

The ADC module supports three conversion groups: Event Group, Group1, and Group2. Each of these three groups can be configured to be triggered by a hardware event. In that case, the application can select from among eight event sources to be the trigger for a group's conversions.

7.4.2.1 MibADC1 Event Trigger Hookup

Table 7-16 lists the event sources that can trigger the conversions for the MibADC1 groups.

Table 7-16. MibADC1 Event Trigger Selection

GROUP SOURCE SELECT BITS (G1SRC, G2SRC OR EVSRC)	EVENT NO.	PINMMR161[0]	PINMMR161[1]	CONTROL FOR OPTION A	CONTROL FOR OPTION B	TRIGGER SOURCE
000	1	x	x	—	—	AD1EVT
001	2	1	0	PINMMR161[8] = x	PINMMR161[9] = x	N2HET1[8]
		0	1	PINMMR161[8] = 1	PINMMR161[9] = 0	N2HET2[5]
		0	1	PINMMR161[8] = 0	PINMMR161[9] = 1	e_TPWM_B
010	3	1	0	—	—	N2HET1[10]
		0	1	—	—	N2HET1[27]
011	4	1	0	PINMMR161[16] = x	PINMMR161[17] = x	RTI1 Comp0
		0	1	PINMMR161[16] = 1	PINMMR161[17] = 0	RTI1 Comp0
		0	1	PINMMR161[16] = 0	PINMMR161[17] = 1	e_TPWM_A1
100	5	1	0	—	—	N2HET1[12]
		0	1	—	—	N2HET1[17]
101	6	1	0	PINMMR161[24] = x	PINMMR161[25] = x	N2HET1[14]
		0	1	PINMMR161[24] = 1	PINMMR161[25] = 0	N2HET1[19]
		0	1	PINMMR161[24] = 0	PINMMR161[25] = 1	N2HET2[1]
110	7	1	0	PINMMR162[0] = x	PINMMR162[1] = x	GIOB[0]
		0	1	PINMMR162[0] = 1	PINMMR162[1] = 0	N2HET1[11]
		0	1	PINMMR162[0] = 0	PINMMR162[1] = 1	ePWM_A2
111	8	1	0	PINMMR162[8] = x	PINMMR162[9] = x	GIOB[1]
		0	1	PINMMR162[8] = 1	PINMMR162[9] = 0	N2HET2[13]
		0	1	PINMMR162[8] = 0	PINMMR162[9] = 1	ePWM_AB

NOTE

For ADEVT trigger source, the connection to the MibADC1 module trigger input is made from the output side of the input buffer. This way, a trigger condition can be generated either by configuring ADEVT as an output function on to the pad (through the mux control), or by driving the ADEVT signal from an external trigger source as input. If the mux control module is used to select different functionality instead of the ADEVT signal, then care must be taken to disable ADEVT from triggering conversions; there is no multiplexing on the input connection.

If ePWM_B, ePWM_A2, ePWM_AB, N2HET2[1], N2HET2[5], N2HET2[13], N2HET1[11], N2HET1[17], or N2HET1[19] is used to trigger the ADC, the connection to the ADC is made directly from the N2HET or ePWM module outputs. As a result, the ADC can be triggered without having to enable the signal from being output on a device terminal.

NOTE

For N2HETx trigger sources, the connection to the MibADC1 module trigger input is made from the input side of the output buffer (at the N2HETx module boundary). This way, a trigger condition can be generated even if the N2HETx signal is not selected to be output on the pad.

NOTE

For the RTI compare 0 interrupt source, the connection is made directly from the output of the RTI module. That is, the interrupt condition can be used as a trigger source even if the actual interrupt is not signaled to the CPU.

7.4.2.2 MibADC2 Event Trigger Hookup

Table 7-17 lists the event sources that can trigger the conversions for the MibADC2 groups.

Table 7-17. MibADC2 Event Trigger Selection

GROUP SOURCE SELECT BITS (G1SRC, G2SRC, or EVSRC)	EVENT NO.	PINMMR161[0]	PINMMR161[1]	CONTROL FOR OPTION A	CONTROL FOR OPTION B	TRIGGER SOURCE
000	1	x	x	NA	NA	AD2EVT
001	2	1	0	PINMMR162[16] = x	PINMMR162[17] = x	N2HET1[8]
		0	1	PINMMR162[16] = 1	PINMMR162[17] = 0	N2HET2[5]
		0	1	PINMMR162[16] = 0	PINMMR162[17] = 1	e_TPWM_B
010	3	1	0	NA	NA	N2HET1[10]
		0	1	NA	NA	N2HET1[27]
011	4	1	0	PINMMR162[24] = x	PINMMR162[25] = x	RTI1 Comp0
		0	1	PINMMR162[24] = 1	PINMMR162[25] = 0	RTI1 Comp0
		0	1	PINMMR162[24] = 0	PINMMR162[25] = 1	e_TPWM_A1
100	5	1	0	NA	NA	N2HET1[12]
		0	1	NA	NA	N2HET1[17]
101	6	1	0	PINMMR163[0] = x	PINMMR163[0] = x	N2HET1[14]
		0	1	PINMMR163[0] = 1	PINMMR163[0] = 0	N2HET1[19]
		0	1	PINMMR163[0] = 0	PINMMR163[0] = 1	N2HET2[1]
110	7	1	0	PINMMR163[8] = x	PINMMR163[8] = x	GIOB[0]
		0	1	PINMMR163[8] = 1	PINMMR163[8] = 0	N2HET1[11]
		0	1	PINMMR163[8] = 0	PINMMR163[8] = 1	ePWM_A2
111	8	1	0	PINMMR163[16] = x	PINMMR163[16] = x	GIOB[1]
		0	1	PINMMR163[16] = 1	PINMMR163[16] = 0	N2HET2[13]
		0	1	PINMMR163[16] = 0	PINMMR163[16] = 1	ePWM_AB

NOTE

For AD2EVT trigger source, the connection to the MibADC2 module trigger input is made from the output side of the input buffer. This way, a trigger condition can be generated either by configuring AD2EVT as an output function on to the pad (through the mux control), or by driving the AD2EVT signal from an external trigger source as input. If the mux control module is used to select different functionality instead of the AD2EVT signal, then care must be taken to disable AD2EVT from triggering conversions; there is no multiplexing on the input connections.

If ePWM_B, ePWM_A2, ePWM_AB, N2HET2[1], N2HET2[5], N2HET2[13], N2HET1[11], N2HET1[17], or N2HET1[19] is used to trigger the ADC, the connection to the ADC is made directly from the N2HET or ePWM module outputs. As a result, the ADC can be triggered without having to enable the signal from being output on a device terminal.

NOTE

For N2HETx trigger sources, the connection to the MibADC2 module trigger input is made from the input side of the output buffer (at the N2HETx module boundary). This way, a trigger condition can be generated even if the N2HETx signal is not selected to be output on the pad.

NOTE

For the RTI compare 0 interrupt source, the connection is made directly from the output of the RTI module. That is, the interrupt condition can be used as a trigger source even if the actual interrupt is not signaled to the CPU.

7.4.2.3 Controlling ADC1 and ADC2 Event Trigger Options Using SOC Output from ePWM Modules

As shown in Figure 7-8, the ePWMxSOCA and ePWMxSOCB outputs from each ePWM module are used to generate four signals – ePWM_B, ePWM_A1, ePWM_A2, and ePWM_AB, that are available to trigger the ADC based on the application requirement.

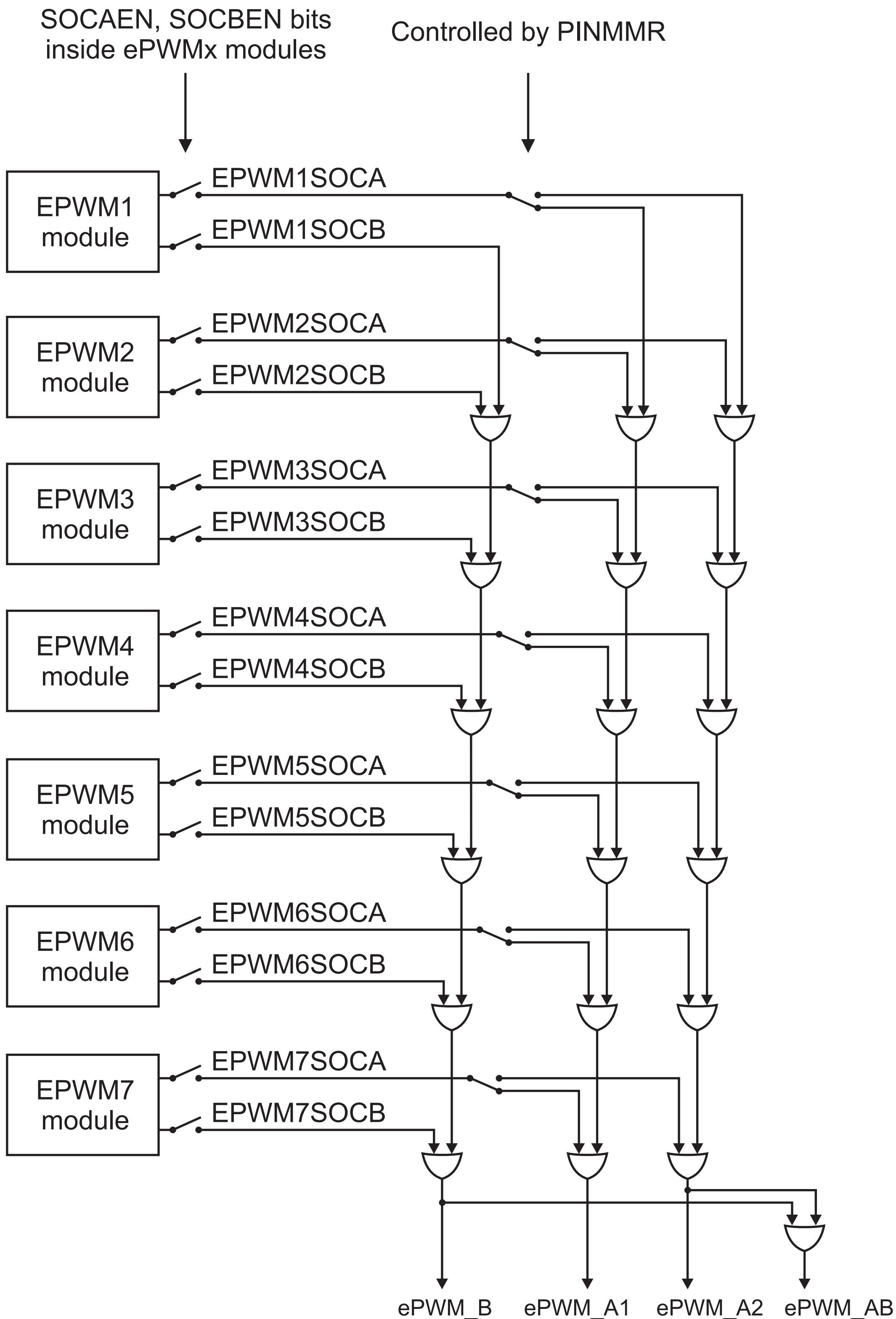


Figure 7-8. ADC Trigger Source Generation from ePWMx

Table 7-18. Control Bit to SOC Output

CONTROL BIT	SOC OUTPUT
PINMMR164[0]	SOC1A_SEL
PINMMR164[8]	SOC2A_SEL
PINMMR164[16]	SOC3A_SEL
PINMMR164[24]	SOC4A_SEL
PINMMR165[0]	SOC5A_SEL
PINMMR165[8]	SOC6A_SEL
PINMMR165[16]	SOC7A_SEL

The SOCA output from each ePWM module is connected to a "switch" shown in Figure 7-8. This switch is implemented by using the control registers in the PINMMR module. Figure 7-9 is an example of the implementation is shown for the switch on SOC1A. The switches on the other SOCA signals are implemented in the same way.

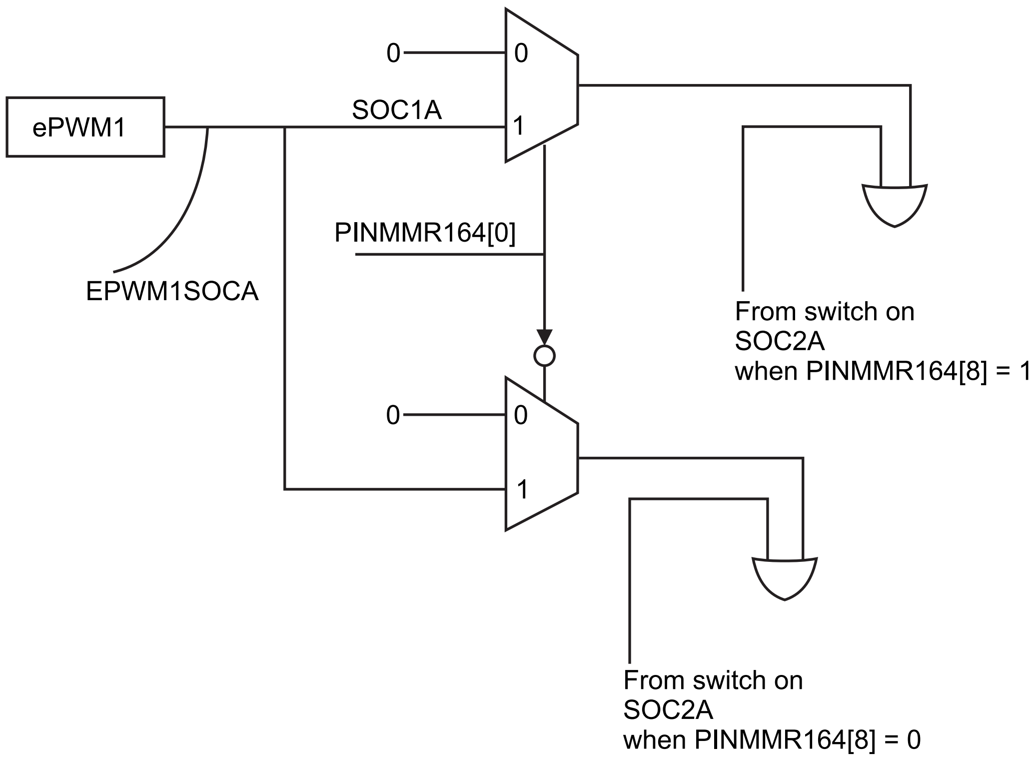


Figure 7-9. ePWM1SOC1A Switch Implementation

The logic equations for the four outputs from the combinational logic shown in Figure 7-8 are:

ePWM_B = SOC1B or SOC2B or SOC3B or SOC4B or SOC5B or SOC6B or SOC7B (1)

ePWM_A1 = [SOC1A and not(SOC1A_SEL)] or [SOC2A and not(SOC2A_SEL)] or [SOC3A and not(SOC3A_SEL)] or [SOC4A and not(SOC4A_SEL)] or [SOC5A and not(SOC5A_SEL)] or [SOC6A and not(SOC6A_SEL)] or [SOC7A and not(SOC7A_SEL)] (2)

ePWM_A2 = [SOC1A and SOC1A_SEL] or [SOC2A and SOC2A_SEL] or [SOC3A and SOC3A_SEL] or [SOC4A and SOC4A_SEL] or [SOC5A and SOC5A_SEL] or [SOC6A and SOC6A_SEL] or [SOC7A and SOC7A_SEL] (3)

ePWM_AB = ePWM_B or ePWM_A2 (4)

7.4.3 ADC Electrical and Timing Specifications

Table 7-19. MibADC Recommended Operating Conditions

PARAMETER		MIN	MAX	UNIT
AD_{REFHI}	A-to-D high-voltage reference source	AD_{REFLO}	V_{CCAD} (1)	V
AD_{REFLO}	A-to-D low-voltage reference source	V_{SSAD} (1)	AD_{REFHI}	V
V_{AI}	Analog input voltage	AD_{REFLO}	AD_{REFHI}	V
I_{AIC}	Analog input clamp current(2) (VAI < VSSAD – 0.3 or VAI > VCCAD + 0.3)	–2	2	mA

(1) For V_{CCAD} and V_{SSAD} recommended operating conditions, see Section 5.4.
(2) Input currents into any ADC input channel outside the specified limits could affect conversion results of other channels.

Table 7-20. MibADC Electrical Characteristics Over Full Ranges of Recommended Operating Conditions(1)

PARAMETER		DESCRIPTION/CONDITIONS		MIN	MAX	UNIT
R_{mux}	Analog input mux on-resistance	See Figure 7-10			250	Ω
R_{samp}	ADC sample switch on-resistance	See Figure 7-10			250	Ω
C_{mux}	Input mux capacitance	See Figure 7-10			16	pF
C_{samp}	ADC sample capacitance	See Figure 7-10			13	pF
I_{AIL}	Analog off-state input leakage current	V_{CCAD} = 3.6 V	$V_{SSAD} \le V_{IN} < V_{SSAD}$ + 100 mV	–300	200	nA
			V_{SSAD} + 100 mV $\le V_{IN} \le V_{CCAD}$ - 200 mV	–200	200	
			V_{CCAD} - 200 mV $< V_{IN} \le V_{CCAD}$	–200	500	
I_{AIL}	Analog off-state input leakage current	V_{CCAD} = 5.25 V	$V_{SSAD} \le V_{IN} < V_{SSAD}$ + 300 mV	–1000	250	nA
			V_{SSAD} + 300 mV $\le V_{IN} \le V_{CCAD}$ - 300 mV	–250	250	
			V_{CCAD} - 300 mV $< V_{IN} \le V_{CCAD}$	–250	1000	
I_{AOSB} (2)	Analog on-state input bias current	V_{CCAD} = 3.6 V	$V_{SSAD} \le V_{IN} < V_{SSAD}$ + 100 mV	–10	2	µA
			V_{SSAD} + 100 mV $< V_{IN} < V_{CCAD}$ - 200 mV	–4	2	
			V_{CCAD} - 200 mV $< V_{IN} < V_{CCAD}$	–4	16	
I_{AOSB} (2)	Analog on-state input bias current	V_{CCAD} = 5.25 V	$V_{SSAD} \le V_{IN} < V_{SSAD}$ + 300 mV	–12	3	µA
			V_{SSAD} + 300 mV $\le V_{IN} \le V_{CCAD}$ - 300 mV	–5	3	
			V_{CCAD} - 300 mV $< V_{IN} \le V_{CCAD}$	–5	18	

(1) For I_{CCAD} and $I_{CCREFHI}$ see Section 5.7.
(2) If a shared channel is being converted by both ADC converters at the same time, the on-state leakage is doubled.

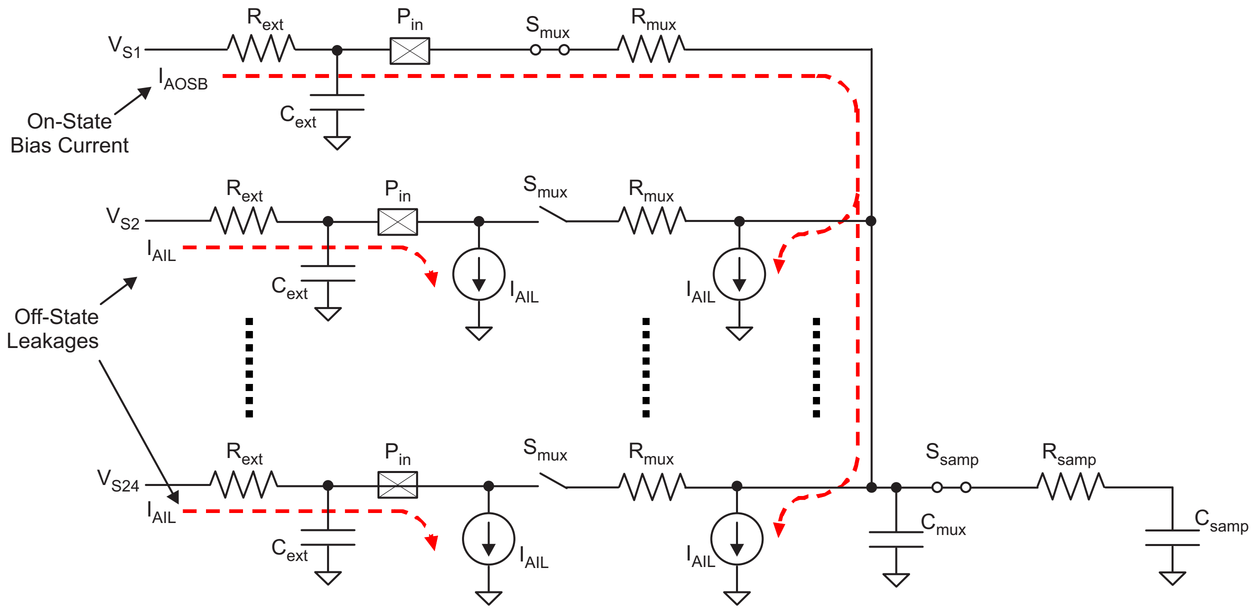


Figure 7-10. MibADC Input Equivalent Circuit

Table 7-21. MibADC Timing Specifications

PARAMETER		MIN	NOM	MAX	UNIT
$t_{c(ADCLK)}$ (1)	Cycle time, MibADC clock	0.033			µs
$t_{d(SH)}$ (2)	Delay time, sample and hold time	0.2			µs
12-BIT MODE					
$t_{d(C)}$	Delay time, conversion time	0.4			µs
$t_{d(SHC)}$ (3)	Delay time, total sample/hold and conversion time	0.6			µs
10-BIT MODE					
$t_{d(C)}$	Delay time, conversion time	0.33			µs
$t_{d(SHC)}$ (3)	Delay time, total sample/hold and conversion time	0.53			µs

(1) The MibADC clock is the ADCLK, generated by dividing down the VCLK1 by a prescale factor defined by the ADCLOCKCR register bits 4:0.
(2) The sample and hold time for the ADC conversions is defined by the ADCLK frequency and the AD<GP>SAMP register for each conversion group. The sample time must be determined by accounting for the external impedance connected to the input channel as well as the internal impedance of the ADC.
(3) This is the minimum sample/hold and conversion time that can be achieved. These parameters are dependent on many factors (for example, the prescale settings).

Table 7-22. MibADC Operating Characteristics Over 3.0 V to 3.6 V Operating Conditions(1) (2)

PARAMETER		DESCRIPTION/CONDITIONS		MIN	MAX	UNIT
CR	Conversion range over which specified accuracy is maintained	AD_{REFHI} - AD_{REFLO}		3	3.6	V
Z_{SET}	Zero Scale Offset	Difference between the first ideal transition (from code 000h to 001h) and the actual transition	10-bit mode		1	LSB
			12-bit mode		2	LSB
F_{SET}	Full Scale Offset	Difference between the range of the measured code transitions (from first to last) and the range of the ideal code transitions	10-bit mode		2	LSB
			12-bit mode		3	LSB
E_{DNL}	Differential nonlinearity error	Difference between the actual step width and the ideal value. (See Figure 7-11)	10-bit mode	–1	1.5	LSB
			12-bit mode	–1	2	LSB
E_{INL}	Integral nonlinearity error	Maximum deviation from the best straight line through the MibADC. MibADC transfer characteristics, excluding the quantization error.	10-bit mode	–2	2	LSB
			12-bit mode	–2	2	LSB
E_{TOT}	Total unadjusted error (after calibration)	Maximum value of the difference between an analog value and the ideal midstep value.	10-bit mode	–2	2	LSB
			12-bit mode	–4	4	LSB

(1) 1 LSB = $(AD_{REFHI} – AD_{REFLO})/ 2^{12}$ for 12-bit mode
(2) 1 LSB = $(AD_{REFHI} – AD_{REFLO})/ 2^{10}$ for 10-bit mode

Table 7-23. MibADC Operating Characteristics Over 3.6 V to 5.25 V Operating Conditions(1) (2)

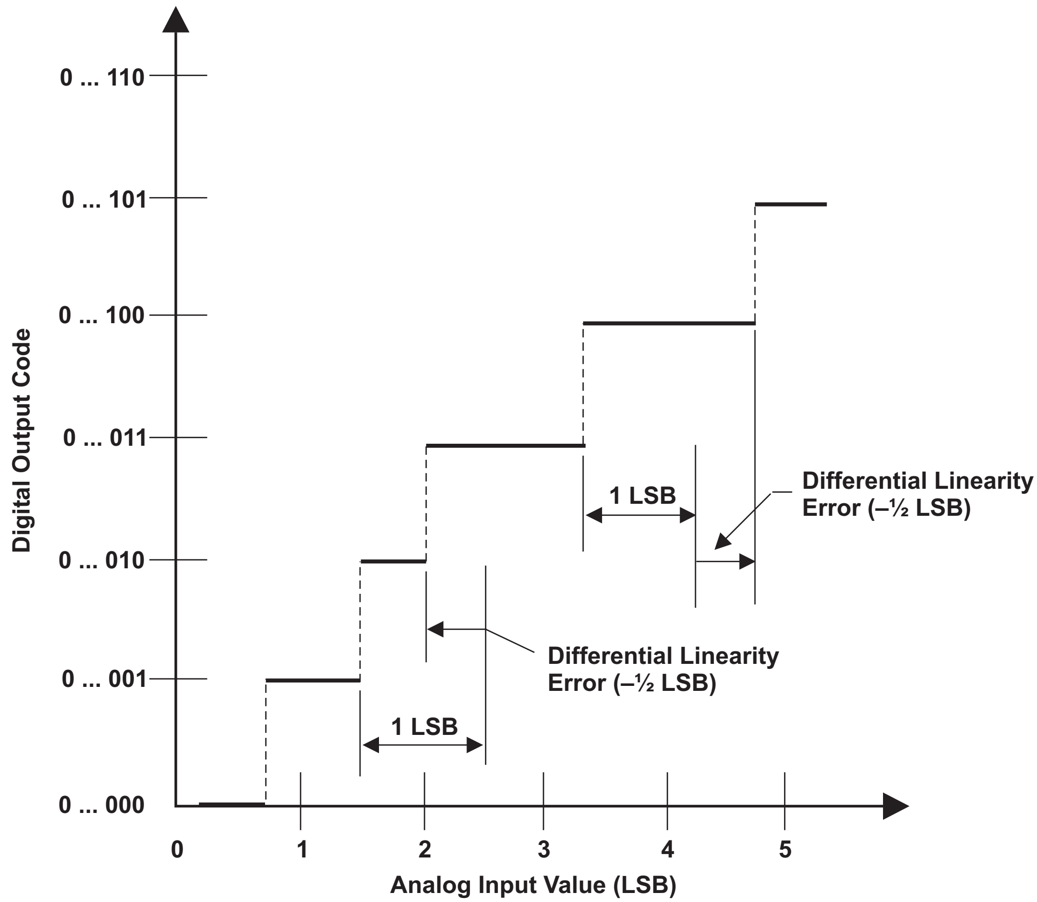
PARAMETER		DESCRIPTION/CONDITIONS		MIN	MAX	UNIT
CR	Conversion range over which specified accuracy is maintained	AD_{REFHI} - AD_{REFLO}		3.6	5.25	V
Z_{SET}	Zero Scale Offset	Difference between the first ideal transition (from code 000h to 001h) and the actual transition	10-bit mode		1	LSB
			12-bit mode		2	LSB
F_{SET}	Full Scale Offset	Difference between the range of the measured code transitions (from first to last) and the range of the ideal code transitions	10-bit mode		2	LSB
			12-bit mode		3	LSB
E_{DNL}	Differential nonlinearity error	Difference between the actual step width and the ideal value. (See Figure 7-11)	10-bit mode	–1	1.5	LSB
			12-bit mode	–1	3	LSB
E_{INL}	Integral nonlinearity error	Maximum deviation from the best straight line through the MibADC. MibADC transfer characteristics, excluding the quantization error.	10-bit mode	–2	2	LSB
			12-bit mode	–4.5	2	LSB
E_{TOT}	Total unadjusted error (after calibration)	Maximum value of the difference between an analog value and the ideal midstep value.	10-bit mode	–2	2	LSB
			12-bit mode	–6	5	LSB

(1) 1 LSB = $(AD_{REFHI} – AD_{REFLO})/ 2^{12}$ for 12-bit mode
(2) 1 LSB = $(AD_{REFHI} – AD_{REFLO})/ 2^{10}$ for 10-bit mode

7.4.4 *Performance (Accuracy) Specifications*

7.4.4.1 MibADC Nonlinearity Errors

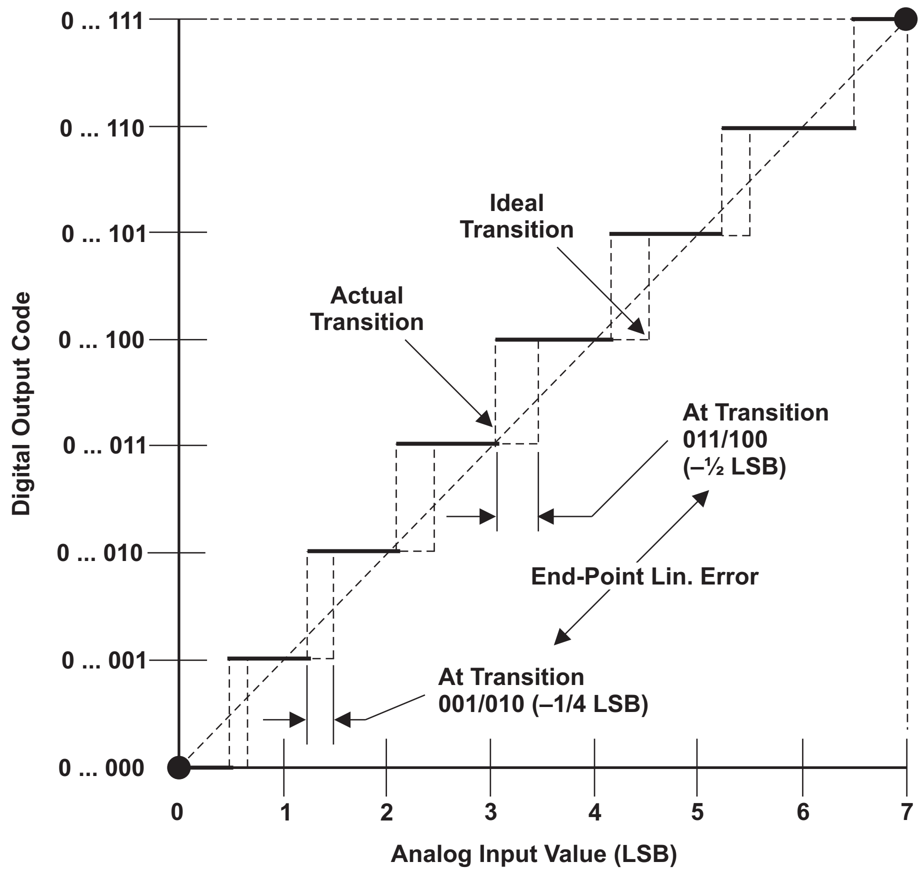
The differential nonlinearity error shown in Figure 7-11 (sometimes referred to as differential linearity) is the difference between an actual step width and the ideal value of 1 LSB.



A. 1 LSB = $(AD_{REFHI} – AD_{REFLO})/2^{12}$

Figure 7-11. Differential Nonlinearity (DNL) Error

The integral nonlinearity error shown in Figure 7-12 (sometimes referred to as linearity error) is the deviation of the values on the actual transfer function from a straight line.

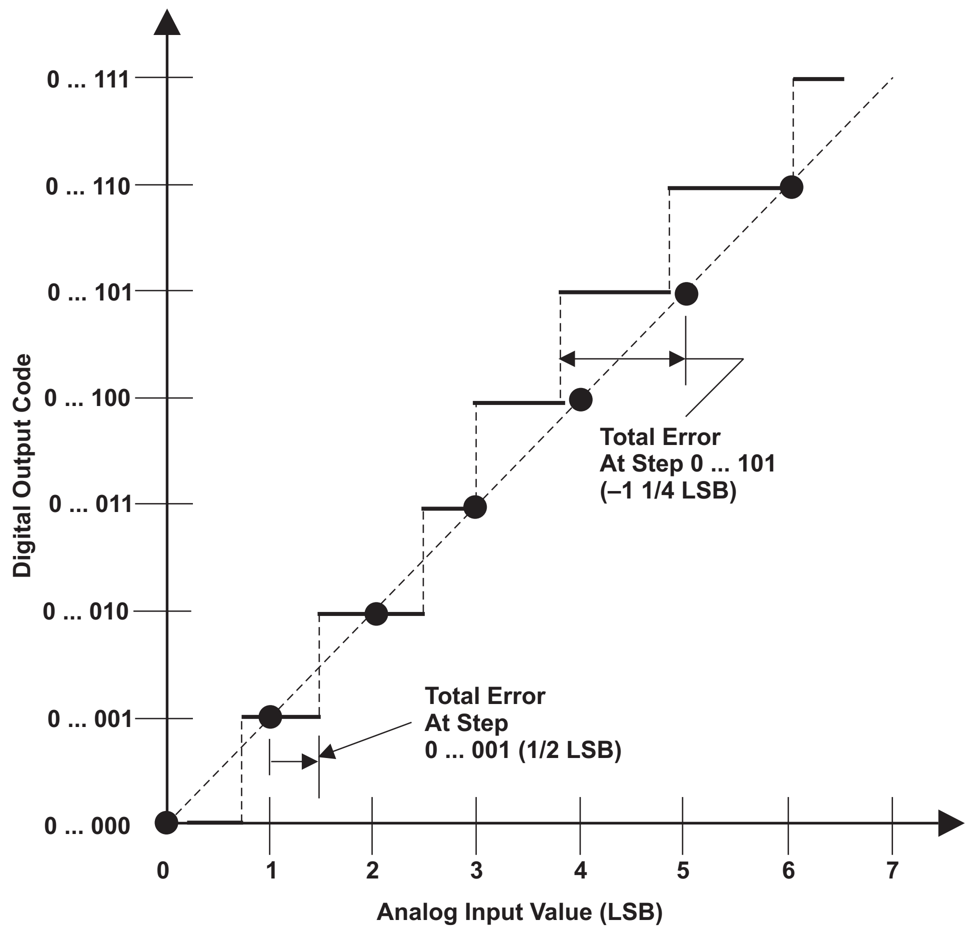


A. 1 LSB = $(AD_{REFHI} - AD_{REFLO})/2^{12}$

Figure 7-12. Integral Nonlinearity (INL) Error(A)

7.4.4.2 MibADC Total Error

The absolute accuracy or total error of an MibADC as shown in Figure 7-13 is the maximum value of the difference between an analog value and the ideal midstep value.



A. 1 LSB = $(AD_{REFHI} - AD_{REFLO})/2^{12}$

Figure 7-13. Absolute Accuracy (Total) Error(A)

7.5 General-Purpose Input/Output

The GPIO module on this device supports two ports, GIOA and GIOB. The I/O pins are bidirectional and bit-programmable. Both GIOA and GIOB support external interrupt capability.

7.5.1 Features

The GPIO module has the following features:

- Each I/O pin can be configured as:
  - Input
  - Output
  - Open Drain
- The interrupts have the following characteristics:
  - Programmable interrupt detection either on both edges or on a single edge (set in GIOINTDET)
  - Programmable edge-detection polarity, either rising or falling edge (set in GIOPOL register)
  - Individual interrupt flags (set in GIOFLG register)
  - Individual interrupt enables, set and cleared through GIOENASET and GIOENACLR registers respectively
  - Programmable interrupt priority, set through GIOLVLSET and GIOLVLCLR registers
- Internal pullup/pulldown allows unused I/O pins to be left unconnected

For information on input and output timings see Section 5.10.1 and Section 5.10.2.

7.6 Enhanced High-End Timer (N2HET)

The N2HET is an advanced intelligent timer that provides sophisticated timing functions for real-time applications. The timer is software-controlled, using a reduced instruction set, with a specialized timer micromachine and an attached I/O port. The N2HET can be used for pulse width modulated outputs, capture or compare inputs, or general-purpose I/O.. It is especially well suited for applications requiring multiple sensor information and drive actuators with complex and accurate time pulses.

7.6.1 Features

The N2HET module has the following features:

- Programmable timer for input and output timing functions
- Reduced instruction set (30 instructions) for dedicated time and angle functions
- 256 words of instruction RAM protected by parity
- User defined number of 25-bit virtual counters for timer, event counters and angle counters
- 7-bit hardware counters for each pin allow up to 32-bit resolution in conjunction with the 25-bit virtual counters
- Up to 32 pins usable for input signal measurements or output signal generation
- Programmable suppression filter for each input pin with adjustable limiting frequency
- Low CPU overhead and interrupt load
- Efficient data transfer to or from the CPU memory with dedicated High-End-Timer Transfer Unit (HTU) or DMA
- Diagnostic capabilities with different loopback mechanisms and pin status readback functionality

7.6.2 N2HET RAM Organization

The timer RAM uses 4 RAM banks, where each bank has two port access capability. This means that one RAM address may be written while another address is read. The RAM words are 96-bits wide, which are split into three 32-bit fields (program, control, and data).

7.6.3 Input Timing Specifications

The N2HET instructions PCNT and WCAP impose some timing constraints on the input signals.

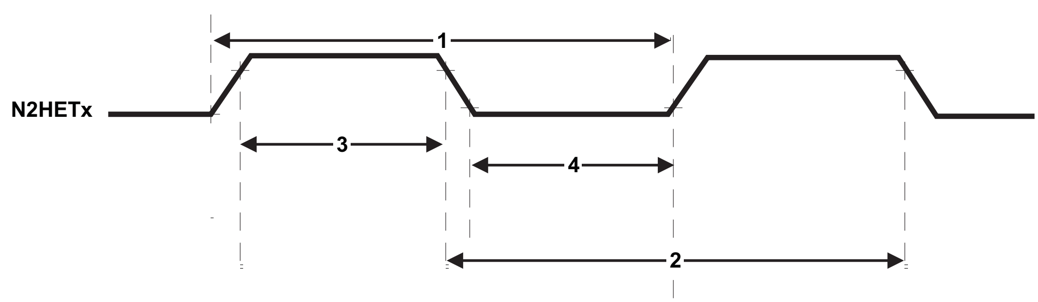


Figure 7-14. N2HET Input Capture Timings

Table 7-24. Dynamic Characteristics for the N2HET Input Capture Functionality

PARAMETER		MIN	MAX	UNIT
1	Input signal period, PCNT or WCAP for rising edge to rising edge	(HRP) (LRP) $tc_{(VCLK2)}$ + 2	2^{25} (HRP) (LRP) $tc_{(VCLK2)}$ - 2	ns
2	Input signal period, PCNT or WCAP for falling edge to falling edge	(HRP) (LRP) $tc_{(VCLK2)}$ + 2	2^{25} (HRP) (LRP) $tc_{(VCLK2)}$ - 2	ns
3	Input signal high phase, PCNT or WCAP for rising edge to falling edge	2 (HRP) $tc_{(VCLK2)}$ + 2	2^{25} (HRP) (LRP) $tc_{(VCLK2)}$ - 2	ns
4	Input signal low phase, PCNT or WCAP for falling edge to rising edge	2 (HRP) $tc_{(VCLK2)}$ + 2	2^{25} (HRP) (LRP) $tc_{(VCLK2)}$ - 2	ns

7.6.4 N2HET1-N2HET2 Interconnections

In some applications the N2HET resolutions must be synchronized. Some other applications require a single time base to be used for all PWM outputs and input timing captures.

The N2HET provides such a synchronization mechanism. The Clk_master/slave (HETGCR.16) configures the N2HET in master or slave mode (default is slave mode). A N2HET in master mode provides a signal to synchronize the prescalers of the slave N2HET. The slave N2HET synchronizes its loop resolution to the loop resolution signal sent by the master. The slave does not require this signal after it receives the first synchronization signal. However, anytime the slave receives the resynchronization signal from the master, the slave must synchronize itself again..

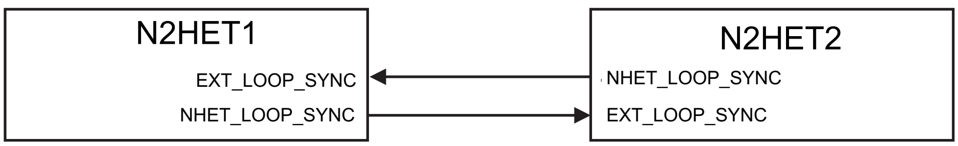


Figure 7-15. N2HET1 – N2HET2 Synchronization Hookup

7.6.5 N2HET Checking

7.6.5.1 Internal Monitoring

To assure correctness of the high-end timer operation and output signals, the two N2HET modules can be used to monitor each other's signals as shown in Figure 7-16. The direction of the monitoring is controlled by the I/O multiplexing control module.

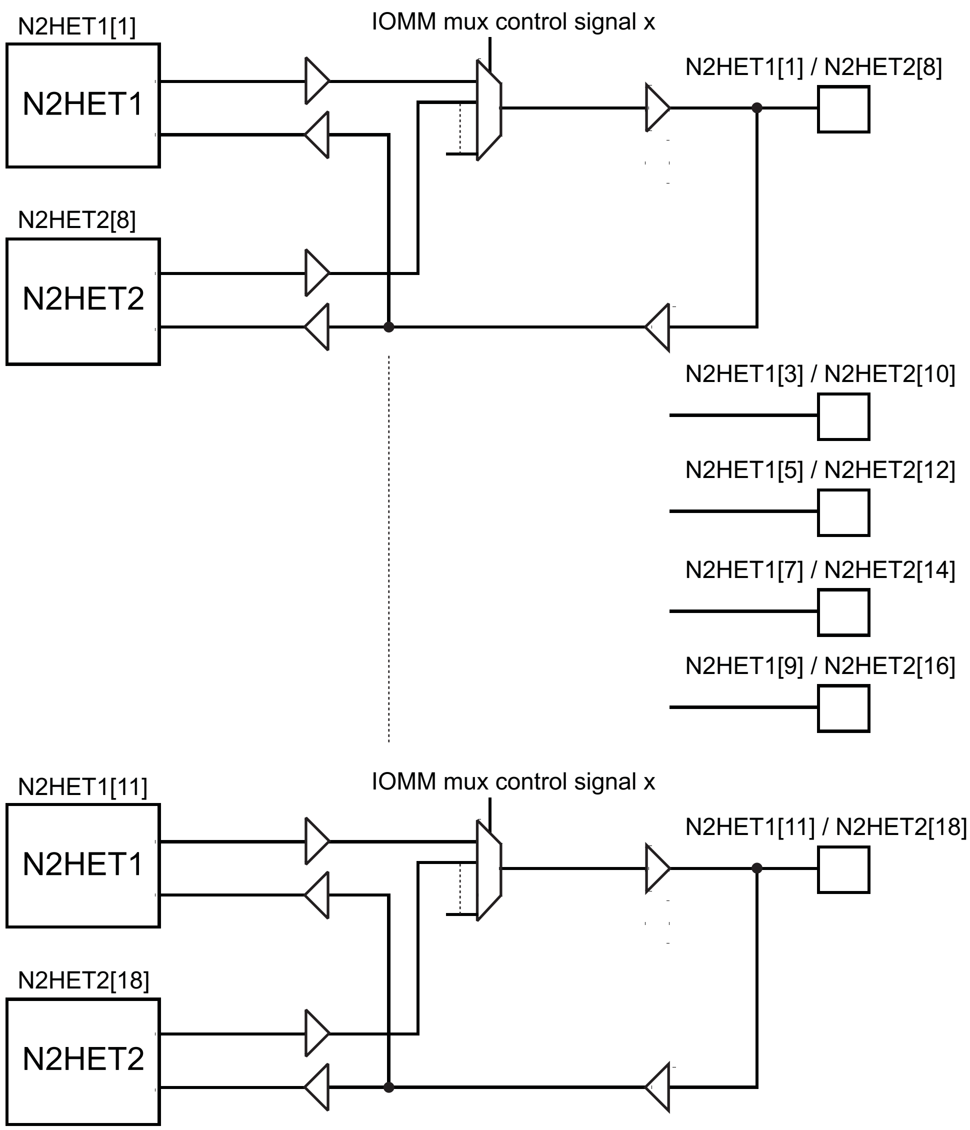


Figure 7-16. N2HET Monitoring

7.6.5.2 Output Monitoring using Dual Clock Comparator (DCC)

N2HET1[31] is connected as a clock source for counter 1 in DCC1. This allows the application to measure the frequency of the pulse-width modulated (PWM) signal on N2HET1[31].

Similarly, N2HET2[0] is connected as a clock source for counter 1 in DCC2. This allows the application to measure the frequency of the pulse-width modulated (PWM) signal on N2HET2[0].

Both N2HET1[31] and N2HET2[0] can be configured to be internal-only channels. That is, the connection to the DCC module is made directly from the output of the N2HETx module (from the input of the output buffer).

For more information on DCC see Section 6.7.3.

7.6.6 Disabling N2HET Outputs

Some applications require the N2HET outputs to be disabled under some fault condition. The N2HET module provides this capability through the "Pin Disable" input signal. This signal, when driven low, causes the N2HET outputs identified by a programmable register (HETPINDIS) to be tri-stated. Refer to the IOMM chapter in the device specific technical reference manual for more details on the "N2HET Pin Disable" feature.

GIOA[5] is connected to the "Pin Disable" input for N2HET1, and GIOB[2] is connected to the "Pin Disable" input for N2HET2.

7.6.7 High-End Timer Transfer Unit (HET-TU)

A High End Timer Transfer Unit (HET-TU) can perform DMA type transactions to transfer N2HET data to or from main memory. A Memory Protection Unit (MPU) is built into the HET-TU.

7.6.7.1 Features

- CPU and DMA independent
- Master Port to access system memory
- 8 control packets supporting dual buffer configuration
- Control packet information is stored in RAM protected by parity
- Event synchronization (HET transfer requests)
- Supports 32 or 64 bit transactions
- Addressing modes for HET address (8 byte or 16 byte) and system memory address (fixed, 32 bit or 64bit)
- One shot, circular and auto switch buffer transfer modes
- Request lost detection

7.6.7.2 Trigger Connections

Table 7-25. HET TU1 Request Line Connection

Modules	Request Source	HET TU1 Request
N2HET1	HTUREQ[0]	HET TU1 DCP[0]
N2HET1	HTUREQ[1]	HET TU1 DCP[1]
N2HET1	HTUREQ[2]	HET TU1 DCP[2]
N2HET1	HTUREQ[3]	HET TU1 DCP[3]
N2HET1	HTUREQ[4]	HET TU1 DCP[4]
N2HET1	HTUREQ[5]	HET TU1 DCP[5]
N2HET1	HTUREQ[6]	HET TU1 DCP[6]
N2HET1	HTUREQ[7]	HET TU1 DCP[7]

Table 7-26. HET TU2 Request Line Connection

Modules	Request Source	HET TU2 Request
N2HET2	HTUREQ[0]	HET TU2 DCP[0]
N2HET2	HTUREQ[1]	HET TU2 DCP[1]
N2HET2	HTUREQ[2]	HET TU2 DCP[2]
N2HET2	HTUREQ[3]	HET TU2 DCP[3]
N2HET2	HTUREQ[4]	HET TU2 DCP[4]
N2HET2	HTUREQ[5]	HET TU2 DCP[5]
N2HET2	HTUREQ[6]	HET TU2 DCP[6]
N2HET2	HTUREQ[7]	HET TU2 DCP[7]

7.7 FlexRay Interface

The FlexRay module performs communication according to the FlexRay protocol specification v2.1. The sample clock bitrate can be programmed to values up to 10 MBit per second. Additional bus driver (BD) hardware is required for connection to the physical layer.

For communication on a FlexRay network, individual message buffers with up to 254 data bytes are configurable. The message storage consists of a single-ported message RAM that holds up to 128 message buffers. All functions concerning the handling of messages are implemented in the message handler. Those functions are the acceptance filtering, the transfer of messages between the two FlexRay Channel Protocol Controllers and the message RAM, maintaining the transmission schedule as well as providing message status information.

The register set of the FlexRay module can be accessed directly by the CPU through the VBUS interface. These registers are used to control, configure and monitor the FlexRay channel protocol controllers, message handler, global time unit, system universal control, frame/symbol processing, network management, interrupt control, and to access the message RAM through the I/O buffer.

7.7.1 Features

The FlexRay module has the following features:

- Conformance with FlexRay protocol specification v2.1
- Data rates of up to 10 Mbps on each channel
- Up to 128 message buffers
- 8KB of message RAM for storage of for example, 128 message buffers with max. 48 byte data section or up to 30 message buffers with 254 byte data section
- Configuration of message buffers with different payload lengths
- One configurable receive FIFO
- Each message buffer can be configured as receive buffer, as transmit buffer or as part of the receive FIFO
- CPU access to message buffers through input and output buffer
- FlexRay Transfer Unit (FTU) for automatic data transfer between data memory and message buffers without CPU interaction
- Filtering for slot counter, cycle counter, and channel ID
- Maskable module interrupts
- Supports Network Management
- ECC protection on the message RAM

7.7.2 Electrical and Timing Specifications

Table 7-27. Timing Requirements for FlexRay Inputs(1)

Parameter		MIN	MAX	UNIT
t_{pw}	Input minimum pulse width to meet the FlexRay sampling requirement	$t_{c(VCLKA2)}$ + 2.5(2)		ns

(1) $t_{c(VCLKA2)}$ = sample clock cycle time for FlexRay = 1 / $f_{(VCLKA2)}$
(2) $t_{RxAsymDelay}$ parameter

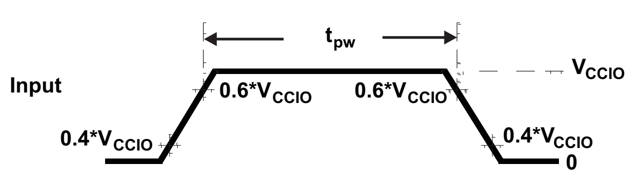


Figure 7-17. FlexRay Inputs

Table 7-28. FlexRay Jitter Timing(1)

PARAMETER		MIN	MAX	UNIT
t_{Tx1bit}	Clock jitter and signal symmetry	98	102	ns
$t_{Tx10bit}$	FlexRay BSS (byte start sequence) to BSS	999	1001	ns
$t_{Tx10bitAvg}$	Average over 10000 samples	999.5	1000.5	ns
$t_{RxAsymDelay}$ (2)	Delay difference between rise and fall from Rx pin to sample point in FlexRay core	–	2.5	ns
$t_{jit(SCLK)}$	Jitter for the 80-MHz Sample Clock generated by the PLL	–	0.5	ns

(1) This parameter will be characterized, but not production-tested.
(2) This value is based on design simulation.

7.7.3 FlexRay Transfer Unit

The FlexRay Transfer Unit is able to transfer data between the input buffer (IBF) and output buffer (OBF) of the communication controller and the system memory without CPU interaction.

Because the FlexRay module is accessed through the FTU, the FTU must be powered up by the setting bit 23 in the Peripheral Power Down Registers of the System Module before accessing any FlexRay module register.

For more information on the FTU see the device specific technical reference manual.

7.8 Controller Area Network (DCAN)

The DCAN supports the CAN 2.0B protocol standard and uses a serial, multimaster communication protocol that efficiently supports distributed real-time control with robust communication rates of up to 1 megabit per second (Mbps). The DCAN is ideal for applications operating in noisy and harsh environments (e.g., automotive and industrial fields) that require reliable serial communication or multiplexed wiring.

7.8.1 Features

Features of the DCAN module include:

- Supports CAN protocol version 2.0 part A, B
- Bit rates up to 1 MBit/s
- The CAN kernel can be clocked by the oscillator for baud-rate generation.
- 64 mailboxes on each DCAN
- Individual identifier mask for each message object
- Programmable FIFO mode for message objects
- Programmable loop-back modes for self-test operation
- Automatic bus on after Bus-Off state by a programmable 32-bit timer
- Message RAM protected by ECC
- Direct access to Message RAM during test mode
- CAN Rx / Tx pins configurable as general purpose IO pins
- Message RAM Auto Initialization
- DMA support

For more information on the DCAN see the device specific technical reference manual.

7.8.2 Electrical and Timing Specifications

Table 7-29. Dynamic Characteristics for the DCANx TX and RX pins

Parameter		MIN	MAX	Unit
$t_{d(CANnTX)}$	Delay time, transmit shift register to CANnTX pin (1)		15	ns
$t_{d(CANnRX)}$	Delay time, CANnRX pin to receive shift register		5	ns

(1) These values do not include rise/fall times of the output buffer.

7.9 Local Interconnect Network Interface (LIN)

The SCI/LIN module can be programmed to work either as an SCI or as a LIN. The core of the module is an SCI. The SCI's hardware features are augmented to achieve LIN compatibility.

The SCI module is a universal asynchronous receiver-transmitter that implements the standard nonreturn to zero format. The SCI can be used to communicate, for example, through an RS-232 port or over a K-line.

The LIN standard is based on the SCI (UART) serial data link format. The communication concept is single-master/multiple-slave with a message identification for multicast transmission between any network nodes.

7.9.1 LIN Features

The following are features of the LIN module:

- Compatible to LIN 1.3, 2.0 and 2.1 protocols
- Multibuffered receive and transmit units DMA capability for minimal CPU intervention
- Identification masks for message filtering
- Automatic Master Header Generation
  - Programmable Synch Break Field
  - Synch Field
  - Identifier Field
- Slave Automatic Synchronization
  - Synch break detection
  - Optional baudrate update
  - Synchronization Validation
- 2^{31} programmable transmission rates with 7 fractional bits
- Error detection
- 2 Interrupt lines with priority encoding

7.10 Serial Communication Interface (SCI)

7.10.1 Features

- Standard universal asynchronous receiver-transmitter (UART) communication
- Supports full- or half-duplex operation
- Standard nonreturn to zero (NRZ) format
- Double-buffered receive and transmit functions
- Configurable frame format of 3 to 13 bits per character based on the following:
  - Data word length programmable from one to eight bits
  - Additional address bit in address-bit mode
  - Parity programmable for zero or one parity bit, odd or even parity
  - Stop programmable for one or two stop bits
- Asynchronous or isosynchronous communication modes
- Two multiprocessor communication formats allow communication between more than two devices.
- Sleep mode is available to free CPU resources during multiprocessor communication.
- The 24-bit programmable baud rate supports 2^{24} different baud rates provide high accuracy baud rate selection.
- Four error flags and Five status flags provide detailed information regarding SCI events.
- Capability to use DMA for transmit and receive data.

7.11 Inter-Integrated Circuit (I2C)

The inter-integrated circuit (I2C) module is a multimaster communication module providing an interface between the TMS570 microcontroller and devices compliant with Philips Semiconductor I2C-bus specification version 2.1 and connected by an I2C-bus. This module will support any slave or master I2C compatible device.

7.11.1 Features

The I2C has the following features:

- Compliance to the Philips I2C bus specification, v2.1 (The I2C Specification, Philips document number 9398 393 40011)
  - Bit/Byte format transfer
  - 7-bit and 10-bit device addressing modes
  - General call
  - START byte
  - Multimaster transmitter/ slave receiver mode
  - Multimaster receiver/ slave transmitter mode
  - Combined master transmit/receive and receive/transmit mode
  - Transfer rates of 10 kbps up to 400 kbps (Phillips fast-mode rate)
- Free data format
- Two DMA events (transmit and receive)
- DMA event enable/disable capability
- Seven interrupts that can be used by the CPU
- Module enable/disable capability
- The SDA and SCL are optionally configurable as general purpose I/O
- Slew rate control of the outputs
- Open drain control of the outputs
- Programmable pullup/pulldown capability on the inputs
- Supports Ignore NACK mode

NOTE

This I2C module does not support:

- High-speed (HS) mode
- C-bus compatibility mode
- The combined format in 10-bit address mode (the I2C sends the slave address second byte every time it sends the slave address first byte)

7.11.2 I2C I/O Timing Specifications

Table 7-30. I2C Signals (SDA and SCL) Switching Characteristics[(1)]

PARAMETER		STANDARD MODE		FAST MODE		UNIT
		MIN	MAX	MIN	MAX	
$t_{c(I2CCLK)}$	Cycle time, Internal Module clock for I2C, prescaled from VCLK	75.2	149	75.2	149	ns
$f_{(SCL)}$	SCL Clock frequency	0	100	0	400	kHz
$t_{c(SCL)}$	Cycle time, SCL	10		2.5		µs
$t_{su(SCLH-SDAL)}$	Setup time, SCL high before SDA low (for a repeated START condition)	4.7		0.6		µs
$t_{h(SCLL-SDAL)}$	Hold time, SCL low after SDA low (for a repeated START condition)	4		0.6		µs
$t_{w(SCLL)}$	Pulse duration, SCL low	4.7		1.3		µs
$t_{w(SCLH)}$	Pulse duration, SCL high	4		0.6		µs
$t_{su(SDA-SCLH)}$	Setup time, SDA valid before SCL high	250		100		ns
$t_{h(SDA-SCLL)}$	Hold time, SDA valid after SCL low (for I2C bus devices)	0	3.45[(2)]	0	0.9	µs
$t_{w(SDAH)}$	Pulse duration, SDA high between STOP and START conditions	4.7		1.3		µs
$t_{su(SCLH-SDAH)}$	Setup time, SCL high before SDA high (for STOP condition)	4.0		0.6		µs
$t_{w(SP)}$	Pulse duration, spike (must be suppressed)			0	50	ns
C_b[(3)]	Capacitive load for each bus line		400		400	pF

(1) The I2C pins SDA and SCL do not feature fail-safe I/O buffers. These pins could potentially draw current when the device is powered down.
(2) The maximum $t_{h(SDA-SCLL)}$ for I2C bus devices has only to be met if the device does not stretch the low period ($t_{w(SCLL)}$) of the SCL signal.
(3) C_b = The total capacitance of one bus line in pF.

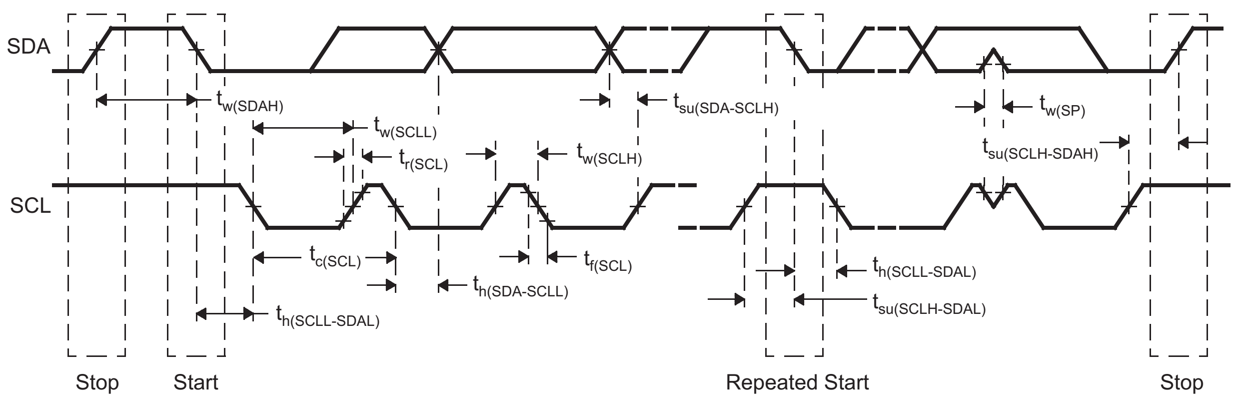


Figure 7-18. I2C Timings

NOTE

- A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the VIHmin of the SCL signal) to bridge the undefined region of the falling edge of SCL.
- The maximum $t_{h(SDA\text{-}SCLL)}$ has only to be met if the device does not stretch the LOW period ($t_{w(SCLL)}$) of the SCL signal.
- A Fast-mode I2C-bus device can be used in a Standard-mode I2C-bus system, but the requirement $t_{su(SDA\text{-}SCLH)} \geq 250$ ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line tr max + $t_{su(SDA\text{-}SCLH)}$.
- C_b = total capacitance of one bus line in pF. If mixed with fast-mode devices, faster fall-times are allowed.

7.12 Multibuffered / Standard Serial Peripheral Interface

The MibSPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (2 to 16 bits) to be shifted in and out of the device at a programmed bit-transfer rate. Typical applications for the SPI include interfacing to external peripherals, such as I/Os, memories, display drivers, and analog-to-digital converters.

7.12.1 Features

Both Standard and MibSPI modules have the following features:

- 16-bit shift register
- Receive buffer register
- 11-bit baud clock generator
- SPICLK can be internally-generated (master mode) or received from an external clock source (slave mode)
- Each word transferred can have a unique format
- SPI I/Os not used in the communication can be used as digital input/output signals

Table 7-31. MibSPI Configurations

MibSPIx/SPIx	I/Os
MibSPI1	MIBSPI1SIMO[1:0], MIBSPI1SOMI[1:0], MIBSPI1CLK, MIBSPI1nCS[5:0], MIBSPI1nENA
MibSPI3	MIBSPI3SIMO, MIBSPI3SOMI, MIBSPI3CLK, MIBSPI3nCS[5:0], MIBSPI3nENA
MibSPI5	MIBSPI5SIMO[3:0], MIBSPI5SOMI[3:0], MIBSPI5CLK, MIBSPI5nCS[5:0], MIBSPI5nENA
MibSPI2	MIBSPI2SIMO,MIBSPI2SOMI,MIBSPI2CLK,MIBSPI2nCS[1:0],MIBSPI2nENA
MibSPI4	MIBSPI4SIMO,MIBSPI4SOMI,MIBSPI4CLK,MIBSPI4nCS[5:0],MIBSPI4nENA

7.12.2 MibSPI Transmit and Receive RAM Organization

The Multibuffer RAM is comprised of 256 buffers for MibSPI1 and 128 buffers for all other MibSPI. Each entry in the Multibuffer RAM consists of 4 parts: a 16-bit transmit field, a 16-bit receive field, a 16-bit control field and a 16-bit status field. The Multibuffer RAM can be partitioned into multiple transfer groups with a variable number of buffers each.

Multibuffered RAM Transfer Groups

MibSPIx/SPIx MODULES	NO OF CHIP SELECTS	MIBSPIxnCS[x]	NO. OF RAM BUFFERS	NO. OF TRANSFER GROUPS
MibSPI1	6	MIBSPI1nCS[5:0]	256	8
MibSPI2	2	MIBSPI2nCS[1:0]	128	8
MibSPI3	6	MIBSPI3nCS[5:0]	128	8
MibSPI4	6	MIBSPI4nCS[5:0]	128	8
MibSPI5	6	MIBSPI5nCS[5:0]	128	8

7.12.3 MibSPI Transmit Trigger Events

Each of the transfer groups can be configured individually. For each of the transfer groups a trigger event and a trigger source can be chosen. A trigger event can be for example a rising edge or a permanent low level at a selectable trigger source. For example, up to 15 trigger sources are available which can be used by each transfer group.

7.12.3.1 MIBSPI1 Event Trigger Hookup

Table 7-32. MIBSPI1 Event Trigger Hookup

Event #	TGxCTRL TRIGSRC[3:0]	Trigger
Disabled	0000	No trigger source
EVENT0	0001	GIOA[0]
EVENT1	0010	GIOA[1]
EVENT2	0011	GIOA[2]
EVENT3	0100	GIOA[3]
EVENT4	0101	GIOA[4]
EVENT5	0110	GIOA[5]
EVENT6	0111	GIOA[6]
EVENT7	1000	GIOA[7]
EVENT8	1001	N2HET1[8]
EVENT9	1010	N2HET1[10]
EVENT10	1011	N2HET1[12]
EVENT11	1100	N2HET1[14]
EVENT12	1101	N2HET1[16]
EVENT13	1110	N2HET1[18]
EVENT14	1111	Intern Tick counter

NOTE

For N2HET1 trigger sources, the connection to the MibSPI1 module trigger input is made from the input side of the output buffer (at the N2HET1 module boundary). This way, a trigger condition can be generated even if the N2HET1 signal is not selected to be output on the pad.

NOTE

For GIOx trigger sources, the connection to the MibSPI1 module trigger input is made from the output side of the input buffer. This way, a trigger condition can be generated either by selecting the GIOx pin as an output pin plus selecting the pin to be a GIOx pin, or by driving the GIOx pin from an external trigger source. If the mux control module is used to select different functionality instead of the GIOx signal, then care must be taken to disable GIOx from triggering MibSPI1 transfers; there is no multiplexing on the input connections.

7.12.3.2 MIBSPI2 Event Trigger Hookup

Table 7-33. MIBSPI2 Event Trigger Hookup

Event #	TGxCTRL TRIGSRC[3:0]	Trigger
Disabled	0000	No trigger source
EVENT0	0001	GIOA[0]
EVENT1	0010	GIOA[1]
EVENT2	0011	GIOA[2]
EVENT3	0100	GIOA[3]
EVENT4	0101	GIOA[4]
EVENT5	0110	GIOA[5]
EVENT6	0111	GIOA[6]
EVENT7	1000	GIOA[7]
EVENT8	1001	N2HET1[8]
EVENT9	1010	N2HET1[10]
EVENT10	1011	N2HET1[12]
EVENT11	1100	N2HET1[14]
EVENT12	1101	N2HET1[16]
EVENT13	1110	N2HET1[18]
EVENT14	1111	Intern Tick counter

NOTE

For N2HET1 trigger sources, the connection to the MibSPI1 module trigger input is made from the input side of the output buffer (at the N2HET1 module boundary). This way, a trigger condition can be generated even if the N2HET1 signal is not selected to be output on the pad.

NOTE

For GIOx trigger sources, the connection to the MibSPI1 module trigger input is made from the output side of the input buffer. This way, a trigger condition can be generated either by selecting the GIOx pin as an output pin plus selecting the pin to be a GIOx pin, or by driving the GIOx pin from an external trigger source. If the mux control module is used to select different functionality instead of the GIOx signal, then care must be taken to disable GIOx from triggering MibSPI1 transfers; there is no multiplexing on the input connections.

7.12.3.3 MIBSPI3 Event Trigger Hookup

Table 7-34. MIBSPI3 Event Trigger Hookup

Event #	TGxCTRL TRIGSRC[3:0]	Trigger
Disabled	0000	No trigger source
EVENT0	0001	GIOA[0]
EVENT1	0010	GIOA[1]
EVENT2	0011	GIOA[2]
EVENT3	0100	GIOA[3]
EVENT4	0101	GIOA[4]
EVENT5	0110	GIOA[5]
EVENT6	0111	GIOA[6]
EVENT7	1000	GIOA[7]
EVENT8	1001	H2ET1[8]

Table 7-34. MIBSPI3 Event Trigger Hookup (continued)

Event #	TGxCTRL TRIGSRC[3:0]	Trigger
EVENT9	1010	N2HET1[10]
EVENT10	1011	N2HET1[12]
EVENT11	1100	N2HET1[14]
EVENT12	1101	N2HET1[16]
EVENT13	1110	N2HET1[18]
EVENT14	1111	Intern Tick counter

NOTE

For N2HET1 trigger sources, the connection to the MibSPI3 module trigger input is made from the input side of the output buffer (at the N2HET1 module boundary). This way, a trigger condition can be generated even if the N2HET1 signal is not selected to be output on the pad.

NOTE

For GIOx trigger sources, the connection to the MibSPI3 module trigger input is made from the output side of the input buffer. This way, a trigger condition can be generated either by selecting the GIOx pin as an output pin plus selecting the pin to be a GIOx pin, or by driving the GIOx pin from an external trigger source. If the mux control module is used to select different functionality instead of the GIOx signal, then care must be taken to disable GIOx from triggering MibSPI3 transfers; there is no multiplexing on the input connections.

7.12.3.4 MIBSPI4 Event Trigger Hookup

Table 7-35. MIBSPI4 Event Trigger Hookup

Event #	TGxCTRL TRIGSRC[3:0]	Trigger
Disabled	0000	No trigger source
EVENT0	0001	GIOA[0]
EVENT1	0010	GIOA[1]
EVENT2	0011	GIOA[2]
EVENT3	0100	GIOA[3]
EVENT4	0101	GIOA[4]
EVENT5	0110	GIOA[5]
EVENT6	0111	GIOA[6]
EVENT7	1000	GIOA[7]
EVENT8	1001	N2HET1[8]
EVENT9	1010	N2HET1[10]
EVENT10	1011	N2HET1[12]
EVENT11	1100	N2HET1[14]
EVENT12	1101	N2HET1[16]
EVENT13	1110	N2HET1[18]
EVENT14	1111	Intern Tick counter

NOTE

For N2HET1 trigger sources, the connection to the MibSPI1 module trigger input is made from the input side of the output buffer (at the N2HET1 module boundary). This way, a trigger condition can be generated even if the N2HET1 signal is not selected to be output on the pad.

NOTE

For GIOx trigger sources, the connection to the MibSPI1 module trigger input is made from the output side of the input buffer. This way, a trigger condition can be generated either by selecting the GIOx pin as an output pin plus selecting the pin to be a GIOx pin, or by driving the GIOx pin from an external trigger source. If the mux control module is used to select different functionality instead of the GIOx signal, then care must be taken to disable GIOx from triggering MibSPI1 transfers; there is no multiplexing on the input connections.

7.12.3.5 MIBSPI5 Event Trigger Hookup

Table 7-36. MIBSPI5 Event Trigger Hookup

Event #	TGxCTRL TRIGSRC[3:0]	Trigger
Disabled	0000	No trigger source
EVENT0	0001	GIOA[0]
EVENT1	0010	GIOA[1]
EVENT2	0011	GIOA[2]
EVENT3	0100	GIOA[3]
EVENT4	0101	GIOA[4]
EVENT5	0110	GIOA[5]
EVENT6	0111	GIOA[6]
EVENT7	1000	GIOA[7]
EVENT8	1001	N2HET1[8]

Table 7-36. MIBSPI5 Event Trigger Hookup (continued)

EVENT9	1010	N2HET1[10]
EVENT10	1011	N2HET1[12]
EVENT11	1100	N2HET1[14]
EVENT12	1101	N2HET1[16]
EVENT13	1110	N2HET1[18]
EVENT14	1111	Intern Tick counter

NOTE

For N2HET1 trigger sources, the connection to the MibSPI5 module trigger input is made from the input side of the output buffer (at the N2HET1 module boundary). This way, a trigger condition can be generated even if the N2HET1 signal is not selected to be output on the pad.

NOTE

For GIOx trigger sources, the connection to the MibSPI5 module trigger input is made from the output side of the input buffer. This way, a trigger condition can be generated either by selecting the GIOx pin as an output pin + selecting the pin to be a GIOx pin, or by driving the GIOx pin from an external trigger source. If the mux control module is used to select different functionality instead of the GIOx signal, then care must be taken to disable GIOx from triggering MibSPI5 transfers; there is no multiplexing on the input connections.

7.12.4 MibSPI/SPI Master Mode I/O Timing Specifications

Table 7-37. SPI Master Mode External Timing Parameters (CLOCK PHASE = 0, SPICLK = output, SPISIMO = output, and SPISOMI = input)[(1)(2)(3)]

NO.	Parameter			MIN	MAX	Unit
1	$t_{c(SPC)M}$	Cycle time, SPICLK[(4)]		40	$256t_{c(VCLK)}$	ns
2[(5)]	$t_{w(SPCH)M}$	Pulse duration, SPICLK high (clock polarity = 0)		$0.5t_{c(SPC)M} - t_{r(SPC)M} - 3$	$0.5t_{c(SPC)M} + 3$	ns
	$t_{w(SPCL)M}$	Pulse duration, SPICLK low (clock polarity = 1)		$0.5t_{c(SPC)M} - t_{f(SPC)M} - 3$	$0.5t_{c(SPC)M} + 3$	
3[(5)]	$t_{w(SPCL)M}$	Pulse duration, SPICLK low (clock polarity = 0)		$0.5t_{c(SPC)M} - t_{f(SPC)M} - 3$	$0.5t_{c(SPC)M} + 3$	ns
	$t_{w(SPCH)M}$	Pulse duration, SPICLK high (clock polarity = 1)		$0.5t_{c(SPC)M} - t_{r(SPC)M} - 3$	$0.5t_{c(SPC)M} + 3$	
4[(5)]	$t_{d(SPCH-SIMO)M}$	Delay time, SPISIMO valid before SPICLK low (clock polarity = 0)		$0.5t_{c(SPC)M} - 6$		ns
	$t_{d(SPCL-SIMO)M}$	Delay time, SPISIMO valid before SPICLK high (clock polarity = 1)		$0.5t_{c(SPC)M} - 6$		
5[(5)]	$t_{v(SPCL-SIMO)M}$	Valid time, SPISIMO data valid after SPICLK low (clock polarity = 0)		$0.5t_{c(SPC)M} - t_{f(SPC)} - 4$		ns
	$t_{v(SPCH-SIMO)M}$	Valid time, SPISIMO data valid after SPICLK high (clock polarity = 1)		$0.5t_{c(SPC)M} - t_{r(SPC)} - 4$		
6[(5)]	$t_{su(SOMI-SPCL)M}$	Setup time, SPISOMI before SPICLK low (clock polarity = 0)		$t_{f(SPC)} + 2.2$		ns
	$t_{su(SOMI-SPCH)M}$	Setup time, SPISOMI before SPICLK high (clock polarity = 1)		$t_{r(SPC)} + 2.2$		
7[(5)]	$t_{h(SPCL-SOMI)M}$	Hold time, SPISOMI data valid after SPICLK low (clock polarity = 0)		10		ns
	$t_{h(SPCH-SOMI)M}$	Hold time, SPISOMI data valid after SPICLK high (clock polarity = 1)		10		
8[(6)]	$t_{C2TDELAY}$	Setup time CS active until SPICLK high (clock polarity = 0)	CSHOLD = 0	$C2TDELAY*t_{c(VCLK)} + 2*t_{c(VCLK)} - t_{f(SPICS)} + t_{r(SPC)} - 7$	$(C2TDELAY+2) * t_{c(VCLK)} - t_{f(SPICS)} + t_{r(SPC)} + 5.5$	ns
			CSHOLD = 1	$C2TDELAY*t_{c(VCLK)} + 3*t_{c(VCLK)} - t_{f(SPICS)} + t_{r(SPC)} - 7$	$(C2TDELAY+3) * t_{c(VCLK)} - t_{f(SPICS)} + t_{r(SPC)} + 5.5$	
		Setup time CS active until SPICLK low (clock polarity = 1)	CSHOLD = 0	$C2TDELAY*t_{c(VCLK)} + 2*t_{c(VCLK)} - t_{f(SPICS)} + t_{f(SPC)} - 7$	$(C2TDELAY+2) * t_{c(VCLK)} - t_{f(SPICS)} + t_{f(SPC)} + 5.5$	ns
			CSHOLD = 1	$C2TDELAY*t_{c(VCLK)} + 3*t_{c(VCLK)} - t_{f(SPICS)} + t_{f(SPC)} - 7$	$(C2TDELAY+3) * t_{c(VCLK)} - t_{f(SPICS)} + t_{f(SPC)} + 5.5$	
9[(6)]	$t_{T2CDELAY}$	Hold time SPICLK low until CS inactive (clock polarity = 0)		$0.5*t_{c(SPC)M} + T2CDELAY*t_{c(VCLK)} + t_{c(VCLK)} - t_{f(SPC)} + t_{r(SPICS)} - 7$	$0.5*t_{c(SPC)M} + T2CDELAY*t_{c(VCLK)} + t_{c(VCLK)} - t_{f(SPC)} + t_{r(SPICS)} + 11$	ns
		Hold time SPICLK high until CS inactive (clock polarity = 1)		$0.5*t_{c(SPC)M} + T2CDELAY*t_{c(VCLK)} + t_{c(VCLK)} - t_{r(SPC)} + tr(SPICS) - 7$	$0.5*t_{c(SPC)M} + T2CDELAY*t_{c(VCLK)} + t_{c(VCLK)} - t_{r(SPC)} + t_{r(SPICS)} + 11$	ns
10	t_{SPIENA}	SPIENAn Sample point		$(C2TDELAY+1) * t_{c(VCLK)} - t_{f(SPICS)} - 29$	$(C2TDELAY+1)*t_{c(VCLK)}$	ns
11	$t_{SPIENAW}$	SPIENAn Sample point from write to buffer			$(C2TDELAY+2)*t_{c(VCLK)}$	ns

(1) The MASTER bit (SPIGCR1.0) is set and the CLOCK PHASE bit (SPIFMTx.16) is cleared.
(2) $t_{c(VCLK)}$ = interface clock cycle time = 1 / $f_{(VCLK)}$
(3) For rise and fall timings, see Table 5-5.
(4) When the SPI is in Master mode, the following must be true:
For PS values from 1 to 255: $t_{c(SPC)M} \geq (PS +1)t_{c(VCLK)} \geq 40ns$, where PS is the prescale value set in the SPIFMTx.[15:8] register bits.
For PS values of 0: $t_{c(SPC)M} = 2t_{c(VCLK)} \geq 40ns$.
The external load on the SPICLK pin must be less than 60pF.
(5) The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPIFMTx.17).
(6) C2TDELAY and T2CDELAY is programmed in the SPIDELAY register

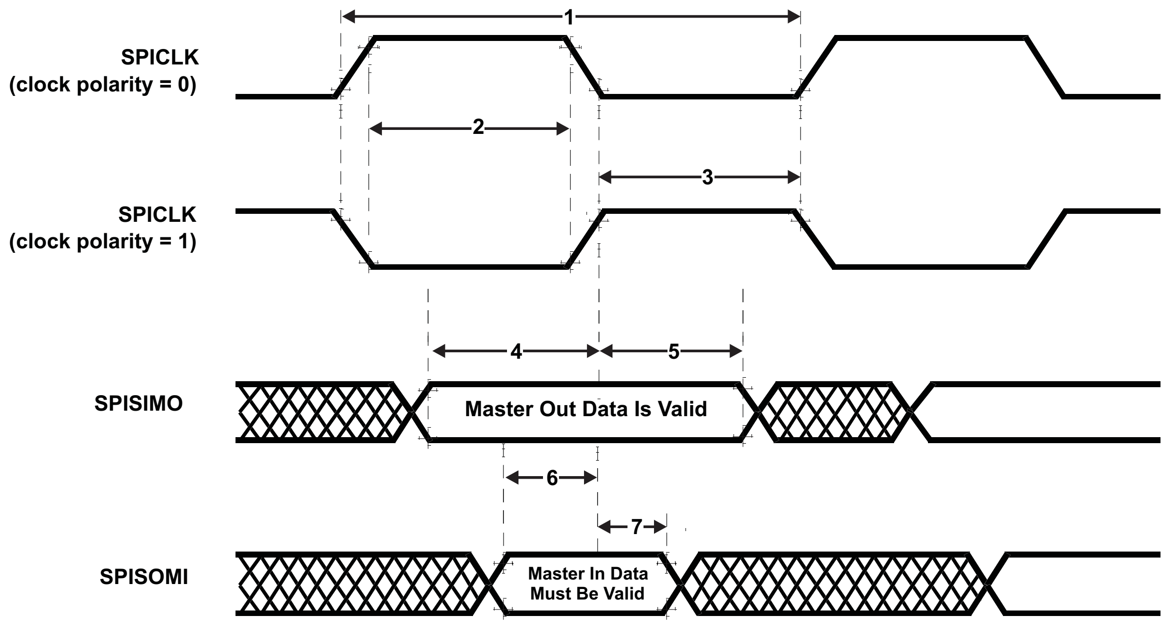


Figure 7-19. SPI Master Mode External Timing (CLOCK PHASE = 0)

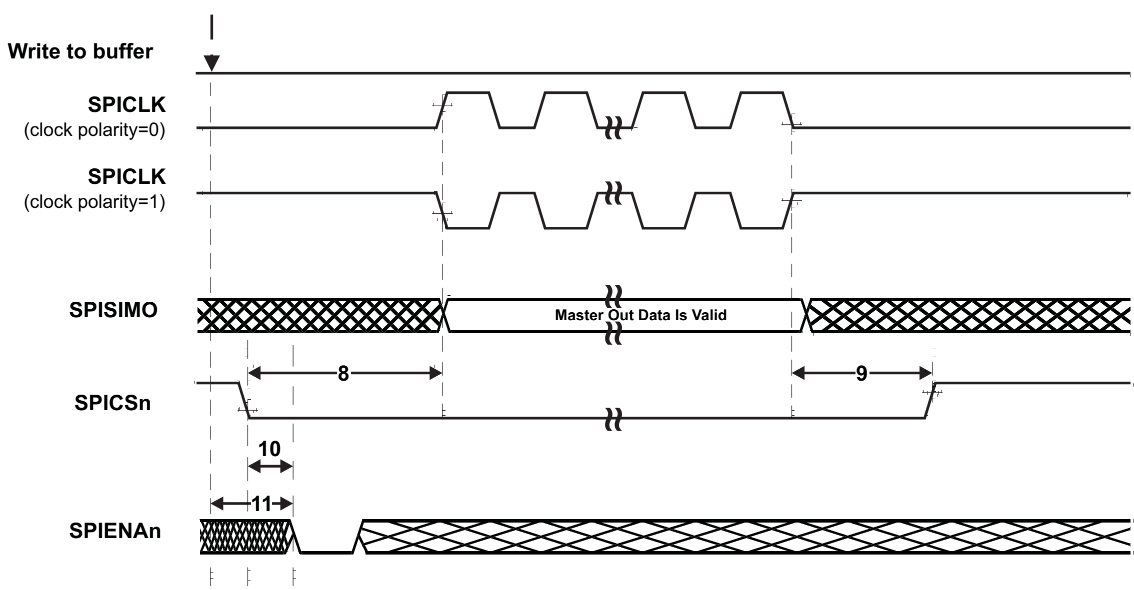


Figure 7-20. SPI Master Mode Chip Select Timing (CLOCK PHASE = 0)

Table 7-38. SPI Master Mode External Timing Parameters (CLOCK PHASE = 1, SPICLK = output, SPISIMO = output, and SPISOMI = input)(1)(2)(3)

NO.	Parameter			MIN	MAX	Unit
1	$t_{c(SPC)M}$	Cycle time, SPICLK (4)		40	$256t_{c(VCLK)}$	ns
2(5)	$t_{w(SPCH)M}$	Pulse duration, SPICLK high (clock polarity = 0)		$0.5t_{c(SPC)M} - t_{r(SPC)M} - 3$	$0.5t_{c(SPC)M} + 3$	ns
	$t_{w(SPCL)M}$	Pulse duration, SPICLK low (clock polarity = 1)		$0.5t_{c(SPC)M} - t_{f(SPC)M} - 3$	$0.5t_{c(SPC)M} + 3$	
3(5)	$t_{w(SPCL)M}$	Pulse duration, SPICLK low (clock polarity = 0)		$0.5t_{c(SPC)M} - t_{f(SPC)M} - 3$	$0.5t_{c(SPC)M} + 3$	ns
	$t_{w(SPCH)M}$	Pulse duration, SPICLK high (clock polarity = 1)		$0.5t_{c(SPC)M} - t_{r(SPC)M} - 3$	$0.5t_{c(SPC)M} + 3$	
4(5)	$t_{v(SIMO-SPCH)M}$	Valid time, SPICLK high after SPISIMO data valid (clock polarity = 0)		$0.5t_{c(SPC)M} - 6$		ns
	$t_{v(SIMO-SPCL)M}$	Valid time, SPICLK low after SPISIMO data valid (clock polarity = 1)		$0.5t_{c(SPC)M} - 6$		
5(5)	$t_{v(SPCH-SIMO)M}$	Valid time, SPISIMO data valid after SPICLK high (clock polarity = 0)		$0.5t_{c(SPC)M} - t_{r(SPC)} - 4$		ns
	$t_{v(SPCL-SIMO)M}$	Valid time, SPISIMO data valid after SPICLK low (clock polarity = 1)		$0.5t_{c(SPC)M} - t_{f(SPC)} - 4$		
6(5)	$t_{su(SOMI-SPCH)M}$	Setup time, SPISOMI before SPICLK high (clock polarity = 0)		$t_{r(SPC)} + 2.2$		ns
	$t_{su(SOMI-SPCL)M}$	Setup time, SPISOMI before SPICLK low (clock polarity = 1)		$t_{f(SPC)} + 2.2$		
7(5)	$t_{v(SPCH-SOMI)M}$	Valid time, SPISOMI data valid after SPICLK high (clock polarity = 0)		10		ns
	$t_{v(SPCL-SOMI)M}$	Valid time, SPISOMI data valid after SPICLK low (clock polarity = 1)		10		
8(6)	$t_{C2TDELAY}$	Setup time CS active until SPICLK high (clock polarity = 0)	CSHOLD = 0	$0.5*t_{c(SPC)M} + (C2TDELAY+2) * t_{c(VCLK)} - t_{f(SPICS)} + t_{r(SPC)} - 7$	$0.5*t_{c(SPC)M} + (C2TDELAY+2) * t_{c(VCLK)} - t_{f(SPICS)} + t_{r(SPC)} + 5.5$	ns
			CSHOLD = 1	$0.5*t_{c(SPC)M} + (C2TDELAY+3) * t_{c(VCLK)} - t_{f(SPICS)} + t_{r(SPC)} - 7$	$0.5*t_{c(SPC)M} + (C2TDELAY+3) * t_{c(VCLK)} - t_{f(SPICS)} + t_{r(SPC)} + 5.5$	
		Setup time CS active until SPICLK low (clock polarity = 1)	CSHOLD = 0	$0.5*t_{c(SPC)M} + (C2TDELAY+2) * t_{c(VCLK)} - t_{f(SPICS)} + t_{f(SPC)} - 7$	$0.5*t_{c(SPC)M} + (C2TDELAY+2) * t_{c(VCLK)} - t_{f(SPICS)} + t_{f(SPC)} + 5.5$	ns
			CSHOLD = 1	$0.5*t_{c(SPC)M} + (C2TDELAY+3) * t_{c(VCLK)} - t_{f(SPICS)} + t_{f(SPC)} - 7$	$0.5*t_{c(SPC)M} + (C2TDELAY+3) * t_{c(VCLK)} - t_{f(SPICS)} + t_{f(SPC)} + 5.5$	
9(6)	$t_{T2CDELAY}$	Hold time SPICLK low until CS inactive (clock polarity = 0)		$T2CDELAY*t_{c(VCLK)} + t_{c(VCLK)} - t_{f(SPC)} + t_{r(SPICS)} - 7$	$T2CDELAY*t_{c(VCLK)} + t_{c(VCLK)} - t_{f(SPC)} + t_{r(SPICS)} + 11$	ns
		Hold time SPICLK high until CS inactive (clock polarity = 1)		$T2CDELAY*t_{c(VCLK)} + t_{c(VCLK)} - t_{r(SPC)} + t_{r(SPICS)} - 7$	$T2CDELAY*t_{c(VCLK)} + t_{c(VCLK)} - t_{r(SPC)} + t_{r(SPICS)} + 11$	ns
10	t_{SPIENA}	SPIENAn Sample Point		$(C2TDELAY+1)* t_{c(VCLK)} - t_{f(SPICS)} - 29$	$(C2TDELAY+1)*t_{c(VCLK)}$	ns
11	$t_{SPIENAW}$	SPIENAn Sample point from write to buffer			$(C2TDELAY+2)*t_{c(VCLK)}$	ns

(1) The MASTER bit (SPIGCR1.0) is set and the CLOCK PHASE bit (SPIFMTx.16) is set.
(2) $t_{c(VCLK)}$ = interface clock cycle time = 1 / $f_{(VCLK)}$
(3) For rise and fall timings, see the Table 5-5.
(4) When the SPI is in Master mode, the following must be true:
For PS values from 1 to 255: $t_{c(SPC)M} \geq (PS + 1)t_{c(VCLK)} \geq 40ns$, where PS is the prescale value set in the SPIFMTx.[15:8] register bits.
For PS values of 0: $t_{c(SPC)M} = 2t_{c(VCLK)} \geq 40ns$.
The external load on the SPICLK pin must be less than 60pF.
(5) The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPIFMTx.17).
(6) C2TDELAY and T2CDELAY is programmed in the SPIDELAY register

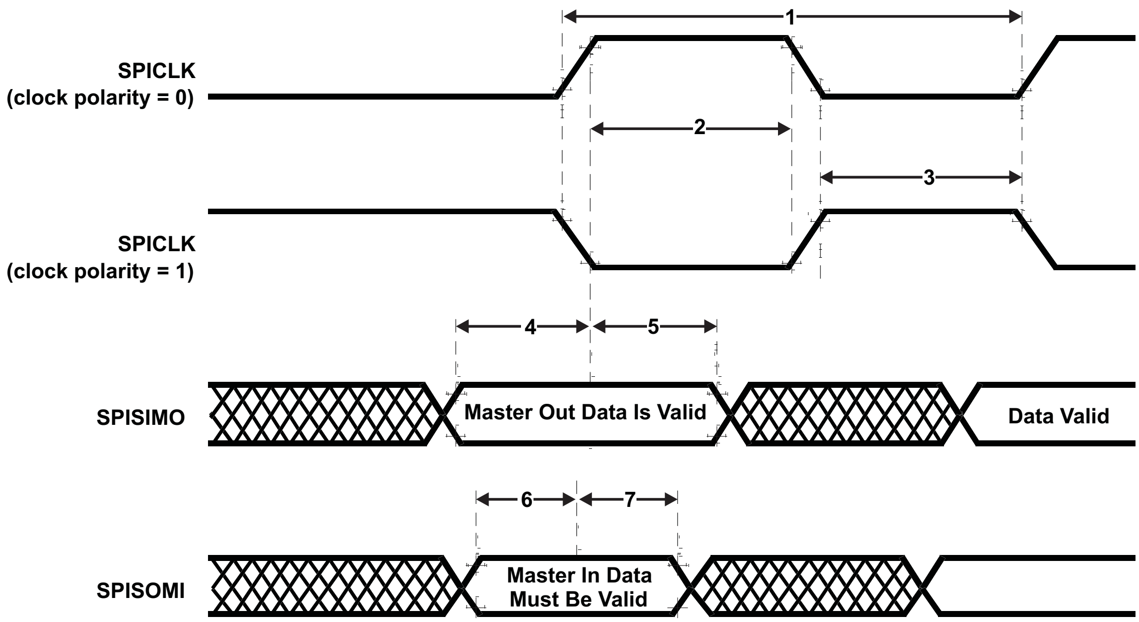


Figure 7-21. SPI Master Mode External Timing (CLOCK PHASE = 1)

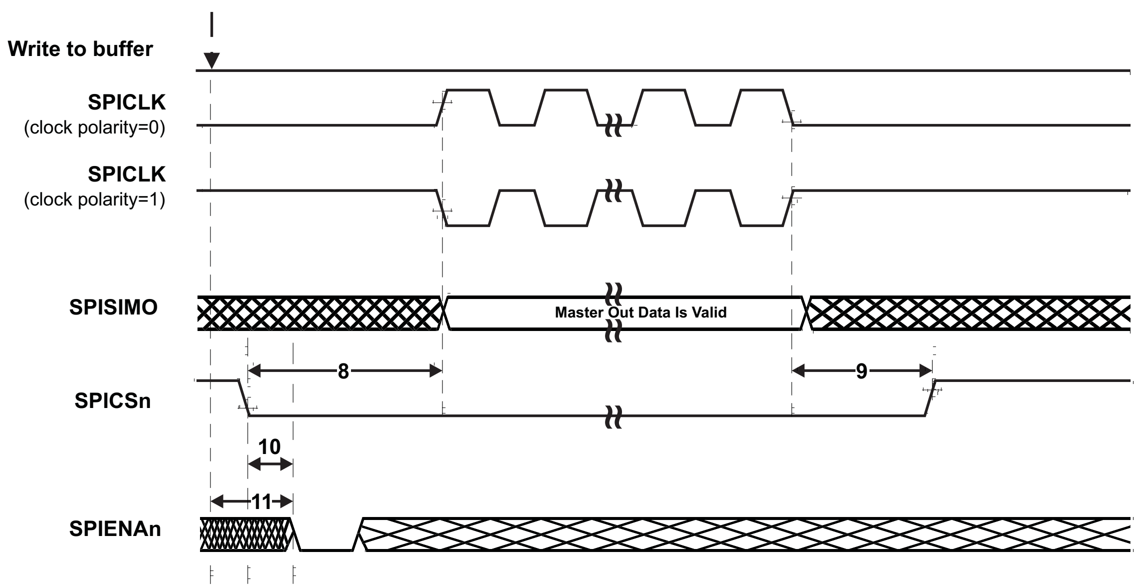


Figure 7-22. SPI Master Mode Chip Select Timing (CLOCK PHASE = 1)

7.12.5 SPI Slave Mode I/O Timings

Table 7-39. SPI Slave Mode External Timing Parameters (CLOCK PHASE = 0, SPICLK = input, SPISIMO = input, and SPISOMI = output)(1)(2)(3)(4)

NO.	Parameter		MIN	MAX	Unit
1	$t_{c(SPC)S}$	Cycle time, SPICLK(5)	40		ns
2(6)	$t_{w(SPCH)S}$	Pulse duration, SPICLK high (clock polarity = 0)	14		ns
	$t_{w(SPCL)S}$	Pulse duration, SPICLK low (clock polarity = 1)	14		
3(6)	$t_{w(SPCL)S}$	Pulse duration, SPICLK low (clock polarity = 0)	14		ns
	$t_{w(SPCH)S}$	Pulse duration, SPICLK high (clock polarity = 1)	14		
4(6)	$t_{d(SPCH-SOMI)S}$	Delay time, SPISOMI valid after SPICLK high (clock polarity = 0)		$t_{rf(SOMI)}$ + 20	ns
	$t_{d(SPCL-SOMI)S}$	Delay time, SPISOMI valid after SPICLK low (clock polarity = 1)		$t_{rf(SOMI)}$ + 20	
5(6)	$t_{h(SPCH-SOMI)S}$	Hold time, SPISOMI data valid after SPICLK high (clock polarity =0)	2		ns
	$t_{h(SPCL-SOMI)S}$	Hold time, SPISOMI data valid after SPICLK low (clock polarity =1)	2		
6(6)	$t_{su(SIMO-SPCL)S}$	Setup time, SPISIMO before SPICLK low (clock polarity = 0)	4		ns
	$t_{su(SIMO-SPCH)S}$	Setup time, SPISIMO before SPICLK high (clock polarity = 1)	4		
7(6)	$t_{h(SPCL-SIMO)S}$	Hold time, SPISIMO data valid after SPICLK low (clock polarity = 0)	2		ns
	$t_{h(SPCH-SIMO)S}$	Hold time, SPISIMO data valid after S PICLK high (clock polarity = 1)	2		
8	$t_{d(SPCL-SENAH)S}$	Delay time, SPIENAn high after last SPICLK low (clock polarity = 0)	$1.5t_{c(VCLK)}$	$2.5t_{c(VCLK)}+t_{r(ENAn)}+22$	ns
	$t_{d(SPCH-SENAH)S}$	Delay time, SPIENAn high after last SPICLK high (clock polarity = 1)	$1.5t_{c(VCLK)}$	$2.5t_{c(VCLK)}+t_{r(ENAn)}+22$	
9	$t_{d(SCSL-SENAL)S}$	Delay time, SPIENAn low after SPICSn low (if new data has been written to the SPI buffer)	$t_{f(ENAn)}$	$t_{c(VCLK)}+t_{f(ENAn)}+27$	ns

(1) The MASTER bit (SPIGCR1.0) is cleared and the CLOCK PHASE bit (SPIFMTx.16) is cleared.
(2) If the SPI is in slave mode, the following must be true: $t_{c(SPC)S} \geq (PS + 1)\, t_{c(VCLK)}$, where PS = prescale value set in SPIFMTx.[15:8].
(3) For rise and fall timings, see Table 5-5.
(4) $t_{c(VCLK)}$ = interface clock cycle time = 1 /$f_{(VCLK)}$
(5) When the SPI is in Slave mode, the following must be true:
For PS values from 1 to 255: $t_{c(SPC)S} \geq (PS +1)t_{c(VCLK)} \geq 40ns$, where PS is the prescale value set in the SPIFMTx.[15:8] register bits.
For PS values of 0: $t_{c(SPC)S} = 2t_{c(VCLK)} \geq 40ns$.
(6) The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPIFMTx.17).

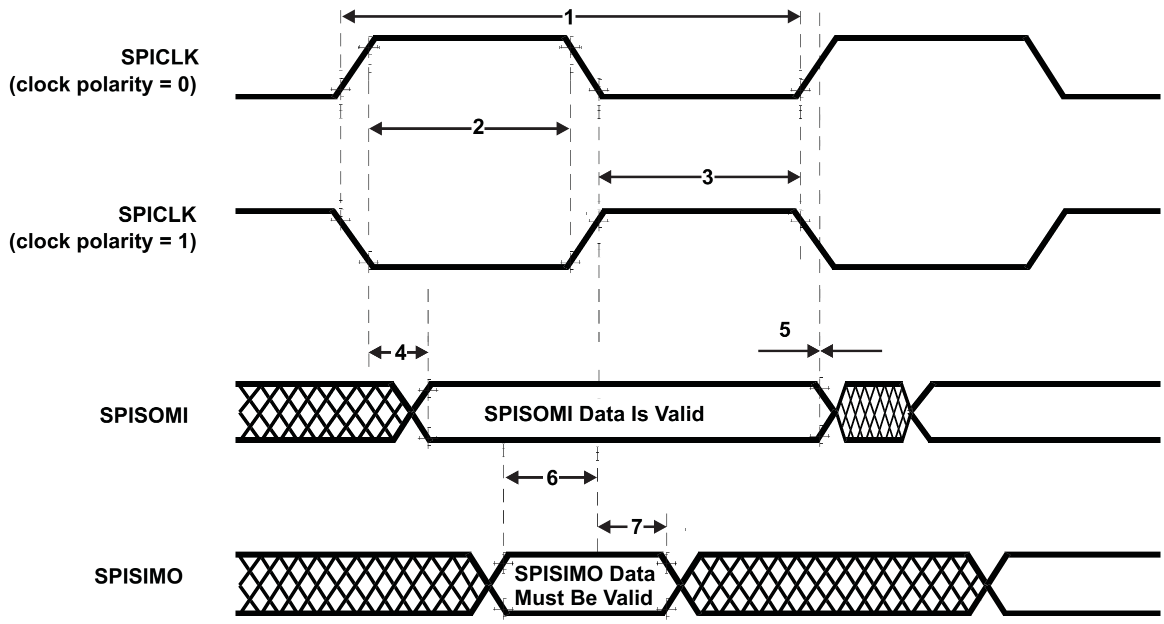


Figure 7-23. SPI Slave Mode External Timing (CLOCK PHASE = 0)

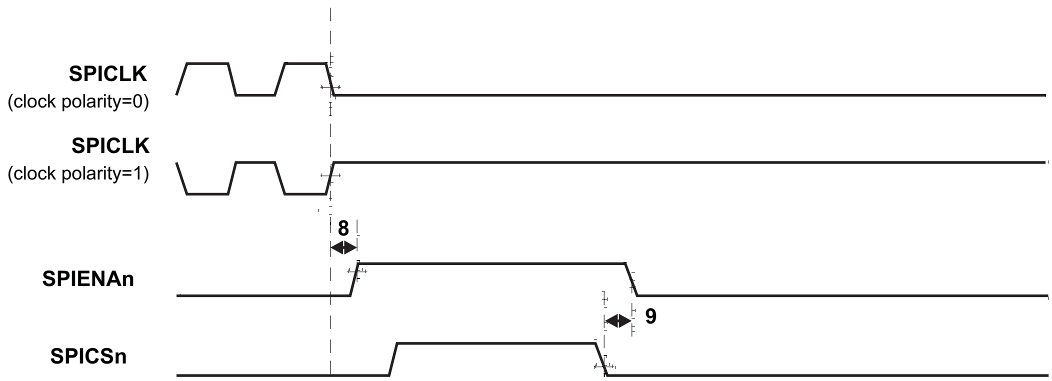


Figure 7-24. SPI Slave Mode Enable Timing (CLOCK PHASE = 0)

Table 7-40. SPI Slave Mode External Timing Parameters (CLOCK PHASE = 1, SPICLK = input, SPISIMO = input, and SPISOMI = output)(1)(2)(3)(4)

NO.	Parameter		MIN	MAX	Unit
1	$t_{c(SPC)S}$	Cycle time, SPICLK(5)	40		ns
2(6)	$t_{w(SPCH)S}$	Pulse duration, SPICLK high (clock polarity = 0)	14		ns
	$t_{w(SPCL)S}$	Pulse duration, SPICLK low (clock polarity = 1)	14		
3(6)	$t_{w(SPCL)S}$	Pulse duration, SPICLK low (clock polarity = 0)	14		ns
	$t_{w(SPCH)S}$	Pulse duration, SPICLK high (clock polarity = 1)	14		
4(6)	$t_{d(SOMI-SPCL)S}$	Dealy time, SPISOMI data valid after SPICLK low (clock polarity = 0)		$t_{rf(SOMI)}$ + 20	ns
	$t_{d(SOMI-SPCH)S}$	Delay time, SPISOMI data valid after SPICLK high (clock polarity = 1)		$t_{rf(SOMI)}$ + 20	
5(6)	$t_{h(SPCL-SOMI)S}$	Hold time, SPISOMI data valid after SPICLK high (clock polarity =0)	2		ns
	$t_{h(SPCH-SOMI)S}$	Hold time, SPISOMI data valid after SPICLK low (clock polarity =1)	2		
6(6)	$t_{su(SIMO-SPCH)S}$	Setup time, SPISIMO before SPICLK high (clock polarity = 0)	4		ns
	$t_{su(SIMO-SPCL)S}$	Setup time, SPISIMO before SPICLK low (clock polarity = 1)	4		
7(6)	$t_{v(SPCH-SIMO)S}$	High time, SPISIMO data valid after SPICLK high (clock polarity = 0)	2		ns
	$t_{v(SPCL-SIMO)S}$	High time, SPISIMO data valid after SPICLK low (clock polarity = 1)	2		
8	$t_{d(SPCH-SENAH)S}$	Delay time, SPIENAn high after last SPICLK high (clock polarity = 0)	$1.5t_{c(VCLK)}$	$2.5t_{c(VCLK)}+t_{r(ENAn)}$ + 22	ns
	$t_{d(SPCL-SENAH)S}$	Delay time, SPIENAn high after last SPICLK low (clock polarity = 1)	$1.5t_{c(VCLK)}$	$2.5t_{c(VCLK)}+t_{r(ENAn)}$ + 22	
9	$t_{d(SCSL-SENAL)S}$	Delay time, SPIENAn low after SPICSn low (if new data has been written to the SPI buffer)	$t_{f(ENAn)}$	$t_{c(VCLK)}+t_{f(ENAn)}$+ 27	ns
10	$t_{d(SCSL-SOMI)S}$	Delay time, SOMI valid after SPICSn low (if new data has been written to the SPI buffer)	$t_{c(VCLK)}$	$2t_{c(VCLK)}+t_{rf(SOMI)}$+ 28	ns

(1) The MASTER bit (SPIGCR1.0) is cleared and the CLOCK PHASE bit (SPIFMTx.16) is set.
(2) If the SPI is in slave mode, the following must be true: tc(SPC)S ≤ (PS + 1) tc(VCLK), where PS = prescale value set in SPIFMTx.[15:8].
(3) For rise and fall timings, see Table 5-5.
(4) $t_{c(VCLK)}$ = interface clock cycle time = 1 /$f_{(VCLK)}$
(5) When the SPI is in Slave mode, the following must be true:
For PS values from 1 to 255: $t_{c(SPC)S} \geq (PS + 1)t_{c(VCLK)} \geq$ 40ns, where PS is the prescale value set in the SPIFMTx.[15:8] register bits.
For PS values of 0: $t_{c(SPC)S} = 2t_{c(VCLK)} \geq$ 40ns.
(6) The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPIFMTx.17).

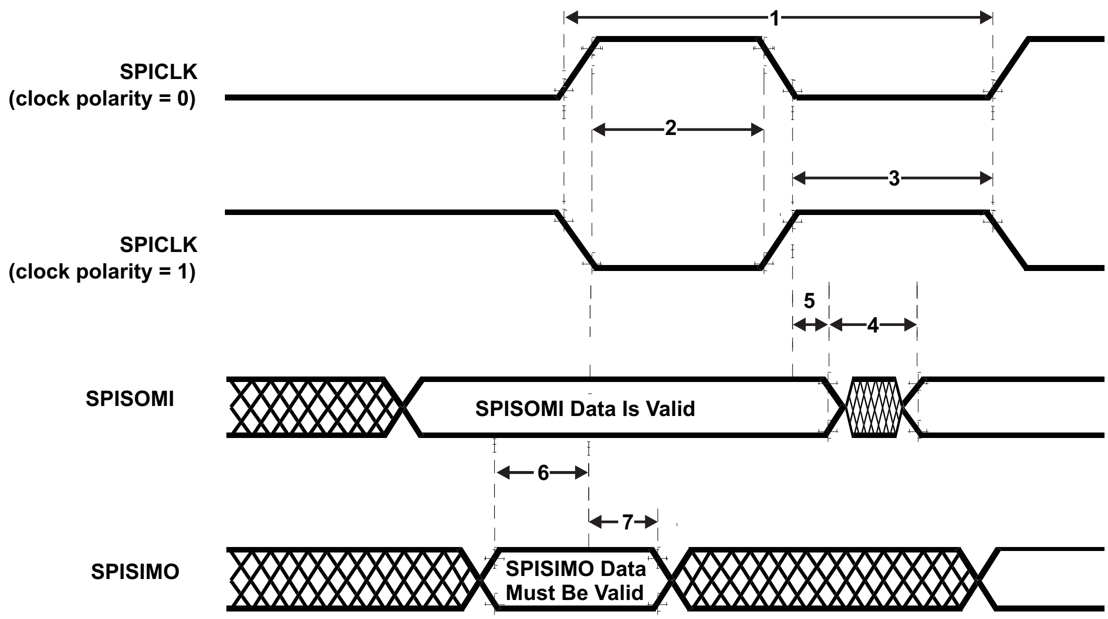


Figure 7-25. SPI Slave Mode External Timing (CLOCK PHASE = 1)

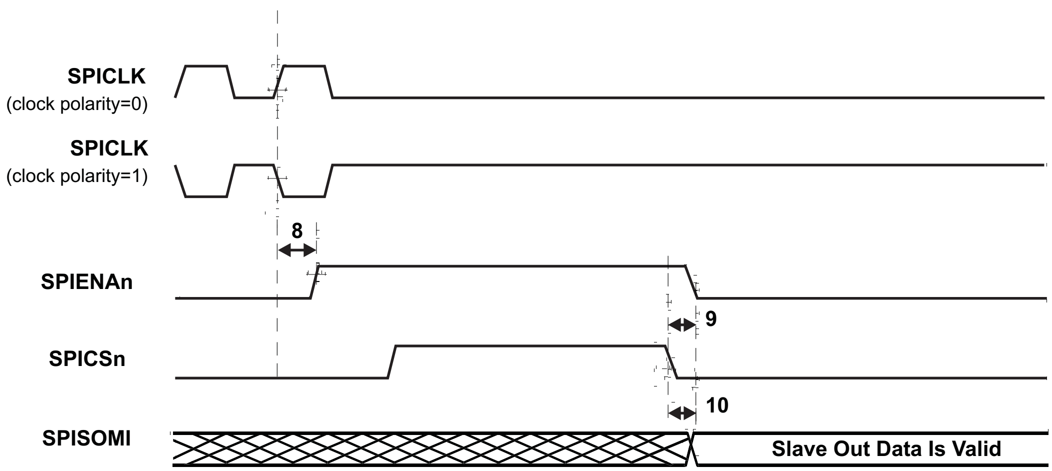


Figure 7-26. SPI Slave Mode Enable Timing (CLOCK PHASE = 1)

7.13 Ethernet Media Access Controller

The Ethernet Media Access Controller (EMAC) provides an efficient interface between the device and the network. The EMAC supports both 10Base-T and 100Base-TX, or 10 Mbits/second (Mbps) and 100 Mbps in either half- or full-duplex mode, with hardware flow control and quality of service (QoS) support.

The EMAC controls the flow of packet data from the device to the PHY. The MDIO module controls PHY configuration and status monitoring.

Both the EMAC and the MDIO modules interface to the device through a custom interface that allows efficient data transmission and reception. This custom interface is referred to as the EMAC control module, and is considered integral to the EMAC/MDIO peripheral. The control module is also used to multiplex and control interrupts.

7.13.1 Ethernet MII Electrical and Timing Specifications

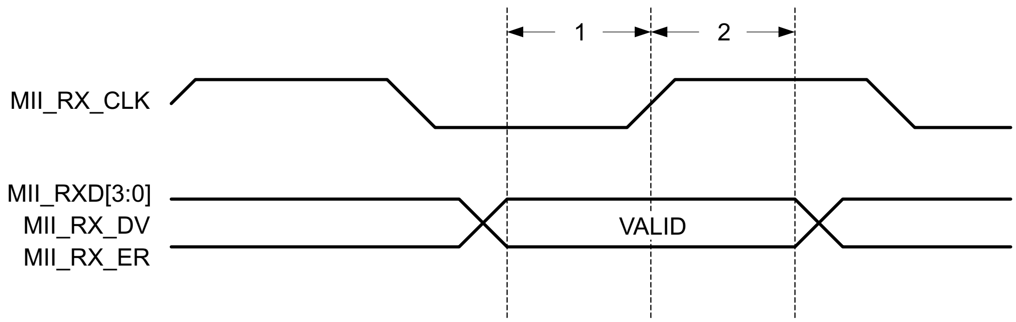


Figure 7-27. MII Receive Timing

Table 7-41. MII Receive Timing

Parameter	Description	MIN	MAX
$t_{su(GMIIMRXD)}$	Setup time, GMIIMRXD to GMIIMRCLK rising edge	8ns	
$t_{su(GMIIMRXDV)}$	Setup time, GMIIMRXDV to GMIIMRCLK rising edge	8ns	
$t_{su(GMIIMRXER)}$	Setup time, GMIIMRXER to GMIIMRCLK rising edge	8ns	
$t_{h(GMIIMRXD)}$	Hold time, GMIIMRXD valid after GMIIRCLK rising edge	8ns	
$t_{h(GMIIMRXDV)}$	Hold time, GMIIMRXDV valid after GMIIRCLK rising edge	8ns	
$t_{h(GMIIMRXER)}$	Hold time, GMIIMRXDV valid after GMIIRCLK rising edge	8ns	

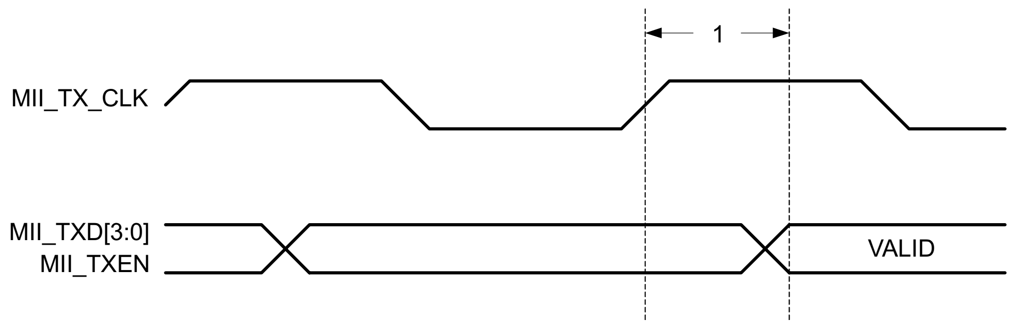


Figure 7-28. MII Transmit Timing

Table 7-42. MII Transmit Timing

Parameter	Description	MIN	MAX
$t_{d(GMIIMTXD)}$	Delay time, GMIIMTCLK rising edge to GMIIMTXD	5ns	25ns
$t_{d(GMIIMTXEN)}$	Delay time, GMIIMTCLK rising edge to GMIIMTXEN	5ns	25ns

7.13.2 Ethernet RMII Timing

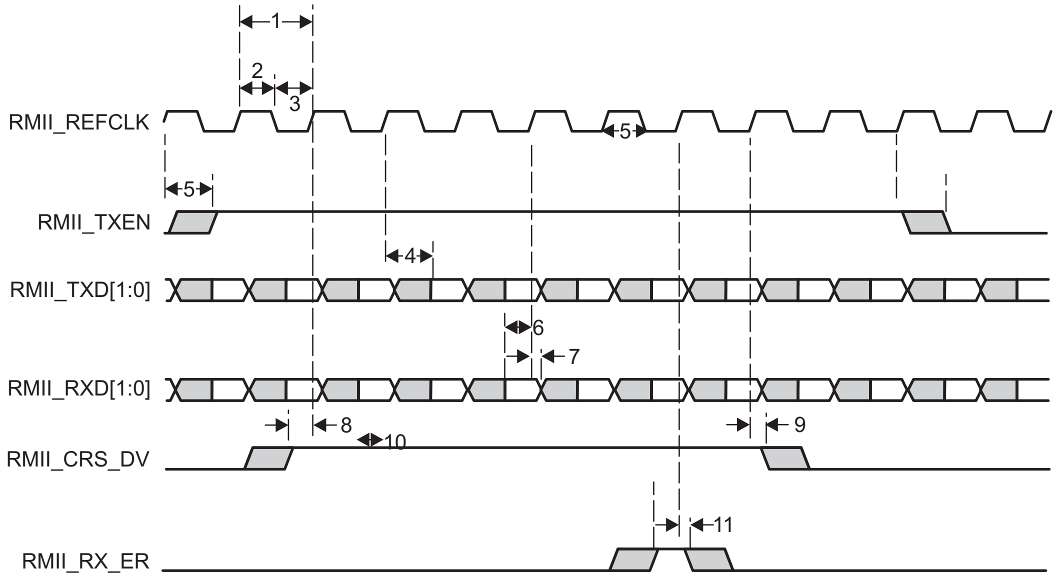


Figure 7-29. RMII Timing Diagram

Table 7-43. RMII Timing Requirements

NO.	Parameter		Value			Unit
			MIN	NOM	MAX	
1	tc(REFCLK)	Cycle time, RMII_REF_CLK	-	20	-	ns
2	tw(REFCLKH)	Pulse width, RMII_REF_CLK High	7	-	13	ns
3	tw(REFCLKL)	Pulse width, RMII_REF_CLK Low	7	-	13	ns
6	tsu(RXD-REFCLK)	Input setup time, RMII_RXD valid before RMII_REF_CLK High	4	-	-	ns
7	th(REFCLK-RXD)	Input hold time, RMII_RXD valid after RMII_REF_CLK High	2	-	-	ns
8	tsu(CRSDV-REFCLK)	Input setup time, RMII_CRSDV valid before RMII_REF_CLK High	4	-	-	ns
9	th(REFCLK-CRSDV)	Input hold time, RMII_CRSDV valid after RMII_REF_CLK High	2	-	-	ns
10	tsu(RXER-REFCLK)	Input setup time, RMII_RXER valid before RMII_REF_CLK High	4	-	-	ns
11	th(REFCLK-RXER)	Input hold time, RMII_RXER valid after RMII_REF_CLK High	2	-	-	ns
4	td(REFCLK-TXD)	Output delay time, RMII_REF_CLK High to RMII_TXD valid	2	-	16	ns
5	td(REFCLK-TXEN)	Output delay time, RMII_REF_CLK High to RMII_TX_EN valid	2	-	16	ns

7.13.3 Management Data Input/Output (MDIO)

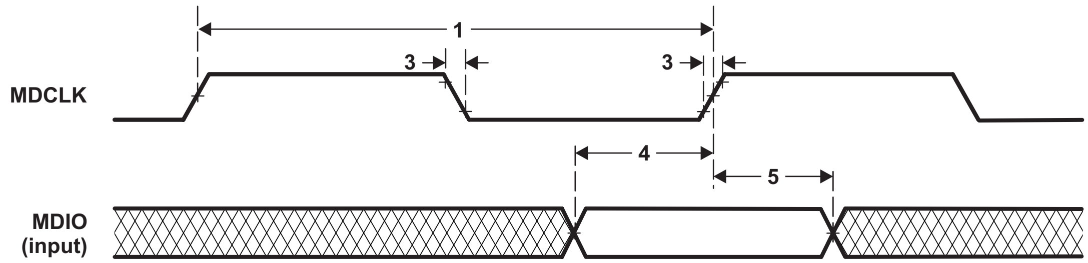


Figure 7-30. MDIO Input Timing

Table 7-44. MDIO Input Timing Requirements

NO.	Parameter		Value		Unit
			MIN	MAX	
1	tc(MDCLK)	Cycle time, MDCLK	400	-	ns
2	tw(MDCLK)	Pulse duration, MDCLK high/low	180	-	ns
3	tt(MDCLK)	Transition time, MDCLK	-	5	ns
4	tsu(MDIO-MDCLKH)	Setup time, MDIO data input valid before MDCLK High	12 (1)	-	ns
5	th(MDCLKH-MDIO)	Hold time, MDIO data input valid after MDCLK High	1	-	ns

(1) This is a discrepancy to IEEE 802.3, but is compatible with many PHY devices.

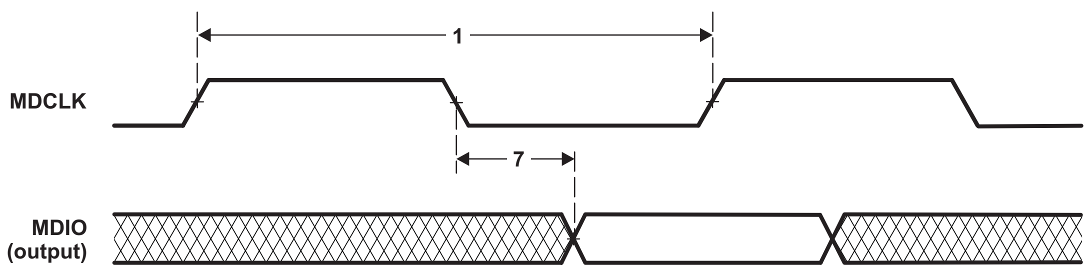


Figure 7-31. MDIO Output Timing

Table 7-45. MDIO Output Timing Requirements

NO.	Parameter		Value		Unit
			MIN	MAX	
1	tc(MDCLK)	Cycle time, MDCLK	400	-	ns
7	td(MDCLKL-MDIO)	Delay time, MDCLK low to MDIO data output valid	0	100	ns

8 Applications, Implementation, and Layout

NOTE

Information in the following sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

8.1 TI Design or Reference Design

TI Designs Reference Design Library is a robust reference design library spanning analog, embedded processor, and connectivity. Created by TI experts to help you jump start your system design, all TI Designs include schematic or block diagrams, BOMs, and design files to speed your time to market. Search and download designs at **TIDesigns**.

9 Device and Documentation Support

9.1 Device Support

9.1.1 Development Support

Texas Instruments (TI) offers an extensive line of development tools for the Hercules™ Safety generation of MCUs, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules.

The following products support development of Hercules™-based applications:

Software Development Tools

- Code Composer Studio™ Integrated Development Environment (IDE)
  - C/C++ Compiler
  - Code generation tools
  - Assembler/Linker
  - Cycle Accurate Simulator
- Application algorithms
- Sample applications code

Hardware Development Tools

- Development and evaluation boards
- JTAG-based emulators - XDS100 v2, XDS200, XDS560™ v2 emulator
- Flash programming tools
- Power supply
- Documentation and cables

9.1.2 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all devices. Each commercial family member has one of three prefixes: TMX, TMP, or TMS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX) through fully qualified production devices (TMS).

Device development evolutionary flow:

TMX Experimental device that is not necessarily representative of the final device's electrical specifications.

TMP Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification.

TMS Fully-qualified production device.

TMX and TMP devices are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

TMS devices 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 (TMX or TMP) 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.

Figure 9-1 shows the numbering and symbol nomenclature for the TMS570LC4357.

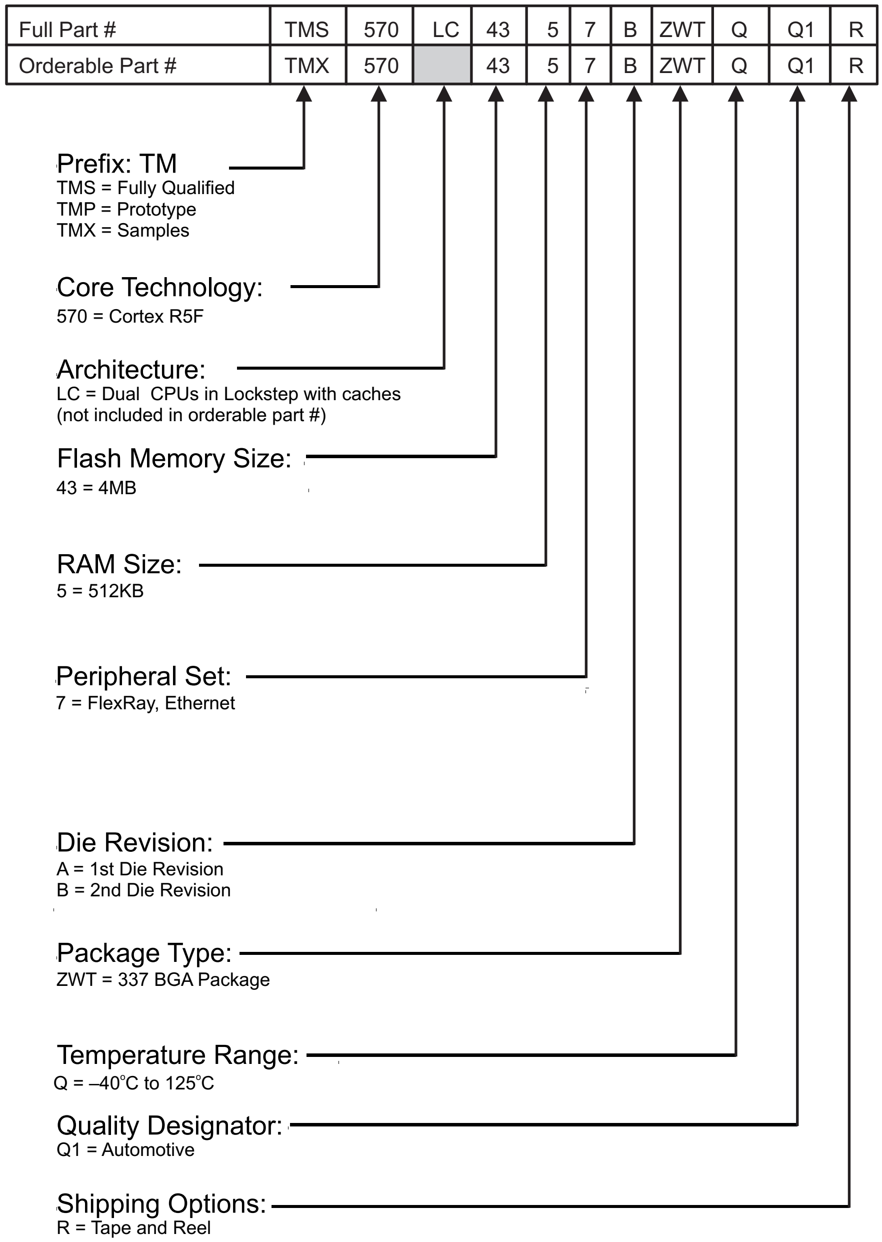


Figure 9-1. TMS570LC4357 Device Numbering Conventions

9.2 Documentation Support

9.2.1 Related Documentation from Texas Instruments

The following documents describe the *TMS570LC4357* microcontroller..

SPNU563 ***TMS570LC43x 16/32-Bit RISC Flash Microcontroller Technical Reference Manual*** details the integration, the environment, the functional description, and the programming models for each peripheral and subsystem in the device.

SPNZ180 ***TMS570LC4357 Microcontroller, Silicon Revision A, Silicon Errata*** describes the usage notes and known exceptions to the functional specifications for the device silicon revision(s).

SPNZ232 ***TMS570LC4x Microcontroller, Silicon Revision B, Silicon Errata*** describes the usage notes and known exceptions to the functional specifications for the device silicon revision(s).

9.2.2 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. In the upper right corner, click on Alert me to register and receive a weekly digest of any product information that has changed. For change details, review the revision history included in any revised document.

9.2.3 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™ Online 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.

9.3 Trademarks

Hercules, Code Composer Studio, XDS560, E2E are trademarks of Texas Instruments.
ETM is a trademark of ARM Limited.
ARM, Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
CoreSight is a trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved..
All other trademarks are the property of their respective owners.

9.4 Electrostatic Discharge Caution

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

9.5 Glossary

TI Glossary This glossary lists and explains terms, acronyms, and definitions.

9.6 Device Identification

9.6.1 Device Identification Code Register

The device identification code register is memory mapped to address FFFF FFF0h and identifies several aspects of the device including the silicon version. The details of the device identification code register are provided in Table 9-1. The device identification code register value for this device is:

- Rev A = 0x8044AD05
- Rev B = 0x8044AD0D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP-15	UNIQUE ID														TECH
R-1	R-00000000100010														R-0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TECH			I/O VOLTAGE	PERIPH PARITY	FLASH ECC		RAM ECC	VERSION					1	0	1
R-101			R-0	R-1	R-10		R-1	R-00000					R-1	R-0	R-1

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Figure 9-2. Device ID Bit Allocation Register

Table 9-1. Device ID Bit Allocation Register Field Descriptions

Bit	Field	Value	Description
31	CP15		Indicates the presence of coprocessor 15
		1	CP15 present
30-17	UNIQUE ID	100011	Silicon version (revision) bits. This bitfield holds a unique number for a dedicated device configuration (die).
16-13	TECH		Process technology on which the device is manufactured.
		0101	F021
12	I/O VOLTAGE		I/O voltage of the device.
		0	I/O are 3.3v
11	PERIPHERAL PARITY		Peripheral Parity
		1	Parity on peripheral memories
10-9	FLASH ECC		Flash ECC
		10	Program memory with ECC
8	RAM ECC		Indicates if RAM ECC is present.
		1	ECC implemented
7-3	REVISION		Revision of the Device.
2-0	101		The platform family ID is always 0b101

9.6.2 Die Identification Registers

The two die ID registers at addresses 0xFFFFFF7C and 0xFFFFFF80 form a 64-bit die id with the information as listed in Table 9-2.

Table 9-2. Die-ID Registers

Item	# of Bits	Bit Location
X Coord. on Wafer	12	0xFFFFFF7C[11:0]
Y Coord. on Wafer	12	0xFFFFFF7C[23:12]
Wafer #	8	0xFFFFFF7C[31:24]
Lot #	24	0xFFFFFF80[23:0]
Reserved	8	0xFFFFFF80[31:24]

9.7 Module Certifications

The following communications modules have received certification of adherence to a standard.

9.7.1 FlexRay Certifications

FlexRay™ Protocol Conformance Certificate

Device (IUT):

Name:	TMS570LC4357 Rev B
Package:	ZWT (S-PBGA-N337) Plastic Ball Grid Array
Version:	Core Release Register: 0x10390206 (CREL[31:0]) Device Identification Code: 0x8044AD0D (DEVID[31:0])
Vendor:	Texas Instruments Incorporated 12500 TI Boulevard Dallas, Texas 75243 USA

Test basis:
FlexRay™ protocol version: 2.1 / 2.1RevA
Test specification version: 2.1.2

Test execution:
Date: 24.09.2015
Hour of completion: 14:01

Test results:
Test cases executed: 275
Test cases passed: 275
Test cases failed: 0

Test report:
Execution ID:
TMS570LS4355BZWT201509241401

Essen, 19.10.2015

Michael Pluta

TÜV NORD Mobilität GmbH & Co.KG
Institute for Vehicle Technology and Mobility

IUT-Details – According to the vendor's data sheet, the IUT has the following peculiarities and optional features:

Peculiarity	Value
MTS transmission activation adjustment time-string	0:x:x:0:0:0
MTS transmission deactivation adjustment time-string	0:x:x:0:0:0
MTS transmission deactivation required	False
cIntDecoderDelay [ST]	8
cColdstartCollisionAbortDelay [µT]	10
Message ID filtering impl. Via valid message indicator	False

Optional feature	Supported/Unsupported
Message ID filtering	Unsupported
Relative timer	Supported
Network Management Vector	Supported
(Re)setting of the 'transmit buffer valid flag'	Supported

This certificate is valid for the hardware and software configuration documented in the test report.

TÜV NORD Mobilität

Figure 9-3. FlexRay Certification for ZWT Package

9.7.2 DCAN Certification

Testhouse
C&S group GmbH
Am Exer 19b
D-38302 Wolfenbuettel
Phone: +49 5331/90 555-0
Fax: +49 5331/90 555-110

C&S

CAN

Authentication **on CAN Conformance**

Texas Instruments
P10_0294_021_CAN_DL_Test_Authentication_r01.doc
Date of Approval: 2011-Feb-08

C&S is worldwide recognized as a neutral expert in testing of communication systems such as CAN Transceiver, CAN, CAN Software Drivers, (CAN) Network Management, FlexRay and LIN.

Herewith C&S group is proud to confirm that the followings tests on the subsequently specified device implementations have been performed by C&S resulting in the findings given below:

C&S Conformance Test Results

Manufacturer	Texas Instruments
Component/Part Number	TMSx70 x021 Microcontroller Family, DCAN Core Release 0xA3170504, 980 A2C0007940000 X470MUF C63C1 P80576 24 YFB-08A9X6W
Date of Tests	February 2011
Version of Test Specification	CAN Conformance Test 1 ISO CAN Conformance Tests according to "ISO 16845:2004 Road vehicles - Controller area network (CAN) - Conformance test plan" and C&S enhancement/ corrections according to "CAN CONFORMANCE TESTING Test Specification C&S Version 2.0 RC" 2 C&S Register Functionality Tests according to "C&S Register Functionality Test Specification V2.0" 3 C&S Robustness Tests according to "C&S Robustness Test Specification V1.4"
Corresponding Test Report	P10_0294_020_CAN_DL_Test_report_r01
1 ISO CAN conformance tests	**Pass**
2 C&S Register Functionality tests	**Pass**
3 C&S Robustness tests	**Pass**
• Further Observations	**None**

Frank Fischer, CTO

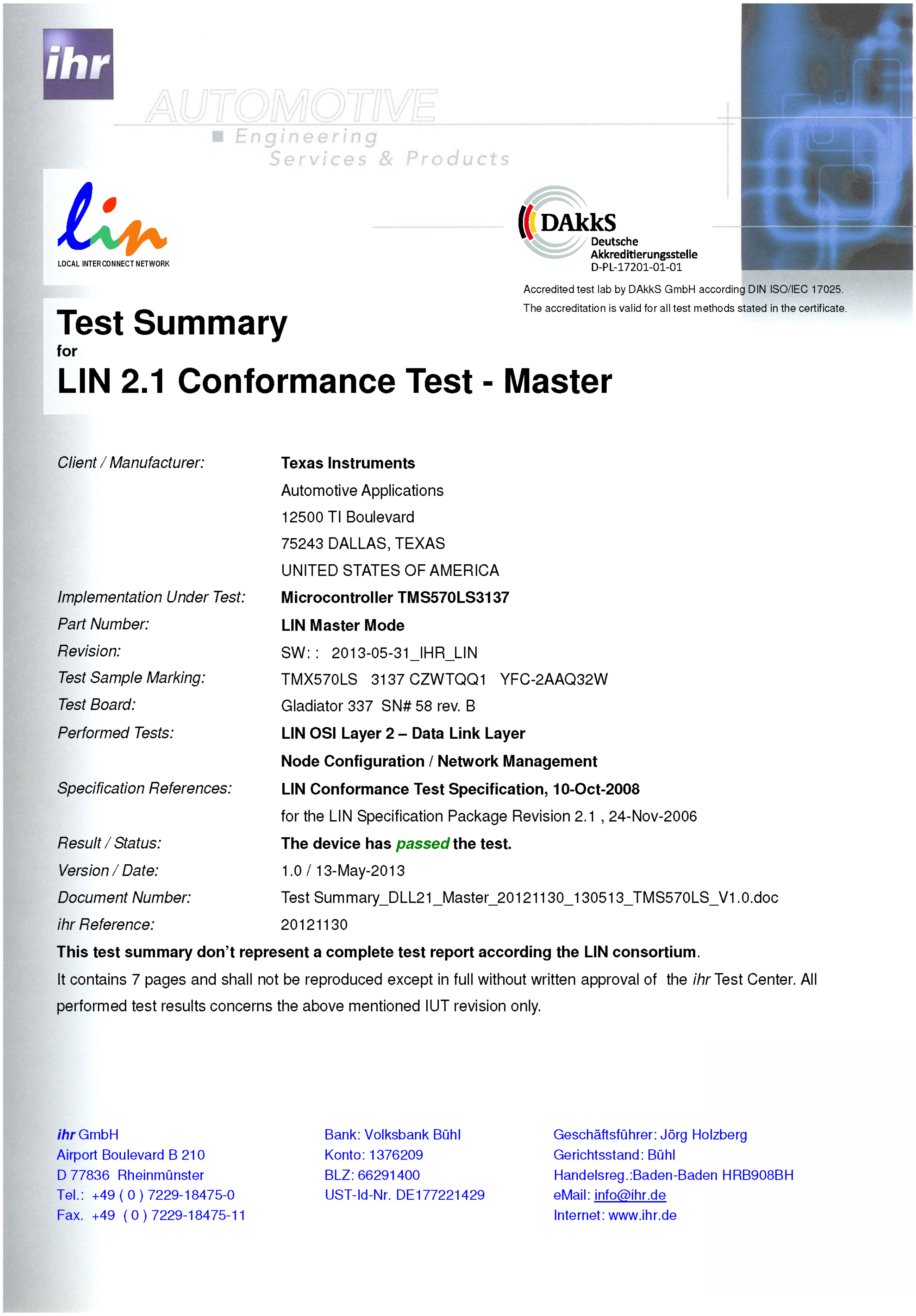
Lothar Kukla, Project Manager

Quote No. P10_0294 R01

Figure 9-4. DCAN Certification

9.7.3 LIN Certification

9.7.3.1 LIN Master Mode



AUTOMOTIVE
Engineering
Services & Products

LOCAL INTERCONNECT NETWORK

DAkkS
Deutsche
Akkreditierungsstelle
D-PL-17201-01-01

Accredited test lab by DAkkS GmbH according DIN ISO/IEC 17025.
The accreditation is valid for all test methods stated in the certificate.

Test Summary
for
LIN 2.1 Conformance Test - Master

Client / Manufacturer:	**Texas Instruments** Automotive Applications 12500 TI Boulevard 75243 DALLAS, TEXAS UNITED STATES OF AMERICA
Implementation Under Test:	**Microcontroller TMS570LS3137**
Part Number:	**LIN Master Mode**
Revision:	SW: : 2013-05-31_IHR_LIN
Test Sample Marking:	TMX570LS 3137 CZWTQQ1 YFC-2AAQ32W
Test Board:	Gladiator 337 SN# 58 rev. B
Performed Tests:	**LIN OSI Layer 2 – Data Link Layer** **Node Configuration / Network Management**
Specification References:	**LIN Conformance Test Specification, 10-Oct-2008** for the LIN Specification Package Revision 2.1 , 24-Nov-2006
Result / Status:	**The device has *passed* the test.**
Version / Date:	1.0 / 13-May-2013
Document Number:	Test Summary_DLL21_Master_20121130_130513_TMS570LS_V1.0.doc
ihr Reference:	20121130

This test summary don't represent a complete test report according the LIN consortium.

It contains 7 pages and shall not be reproduced except in full without written approval of the *ihr* Test Center. All performed test results concerns the above mentioned IUT revision only.

ihr GmbH
Airport Boulevard B 210
D 77836 Rheinmünster
Tel.: +49 (0) 7229-18475-0
Fax. +49 (0) 7229-18475-11

Bank: Volksbank Bühl
Konto: 1376209
BLZ: 66291400
UST-Id-Nr. DE177221429

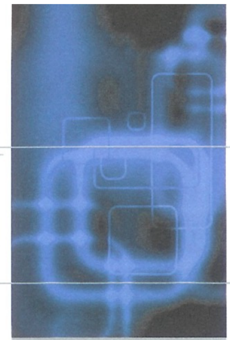
Geschäftsführer: Jörg Holzberg
Gerichtsstand: Bühl
Handelsreg.:Baden-Baden HRB908BH
eMail: info@ihr.de
Internet: www.ihr.de

Figure 9-5. LIN Certification - Master Mode

9.7.3.2 LIN Slave Mode - Fixed Baud Rate

ihr

AUTOMOTIVE
Engineering
Services & Products



LOCAL INTERCONNECT NETWORK

DAkkS Deutsche Akkreditierungsstelle D-PL-17201-01-01

Accredited test lab by DAkkS GmbH according DIN ISO/IEC 17025.
The accreditation is valid for all test methods stated in the certificate.

Test Summary
for
LIN 2.1 Conformance Test - Slave

Client / Manufacturer:	**Texas Instruments** Automotive Applications 12500 TI Boulevard 75243 DALLAS, TEXAS UNITED STATES OF AMERICA
Implementation Under Test:	**Microcontroller TMS570LS3137** **LIN Slave Mode - Fixed Baud Rate Mode**
Revision:	SW: 2013-05-31_IHR_LIN
Test Sample Marking:	TMX570LS 3137 CZWTQQ1 YFC-2AAQ32W
Test Board:	Gladiator 337 SN# 58 rev. B
Performed Tests:	**LIN OSI Layer 2 – Data Link Layer** **Node Configuration / Network Management**
Specification References:	**LIN Conformance Test Specification, 10-Oct-2008** for the LIN Specification Package Revision 2.1 , 24-Nov-2006
Result / Status:	**The device has *passed* the test.**
Version / Date:	1.0 / 13-May-2013
Document Number:	Test Summary_DLL21_Slave_Fixed_20121130_130513_TMS570LS_V1.0.doc
ihr *Reference:*	20121130

This test summary don't represent a complete test report according the LIN consortium.

It contains 16 pages and shall not be reproduced except in full without written approval of the ***ihr*** Test Center.

All performed test results concerns the above mentioned IUT revision only.

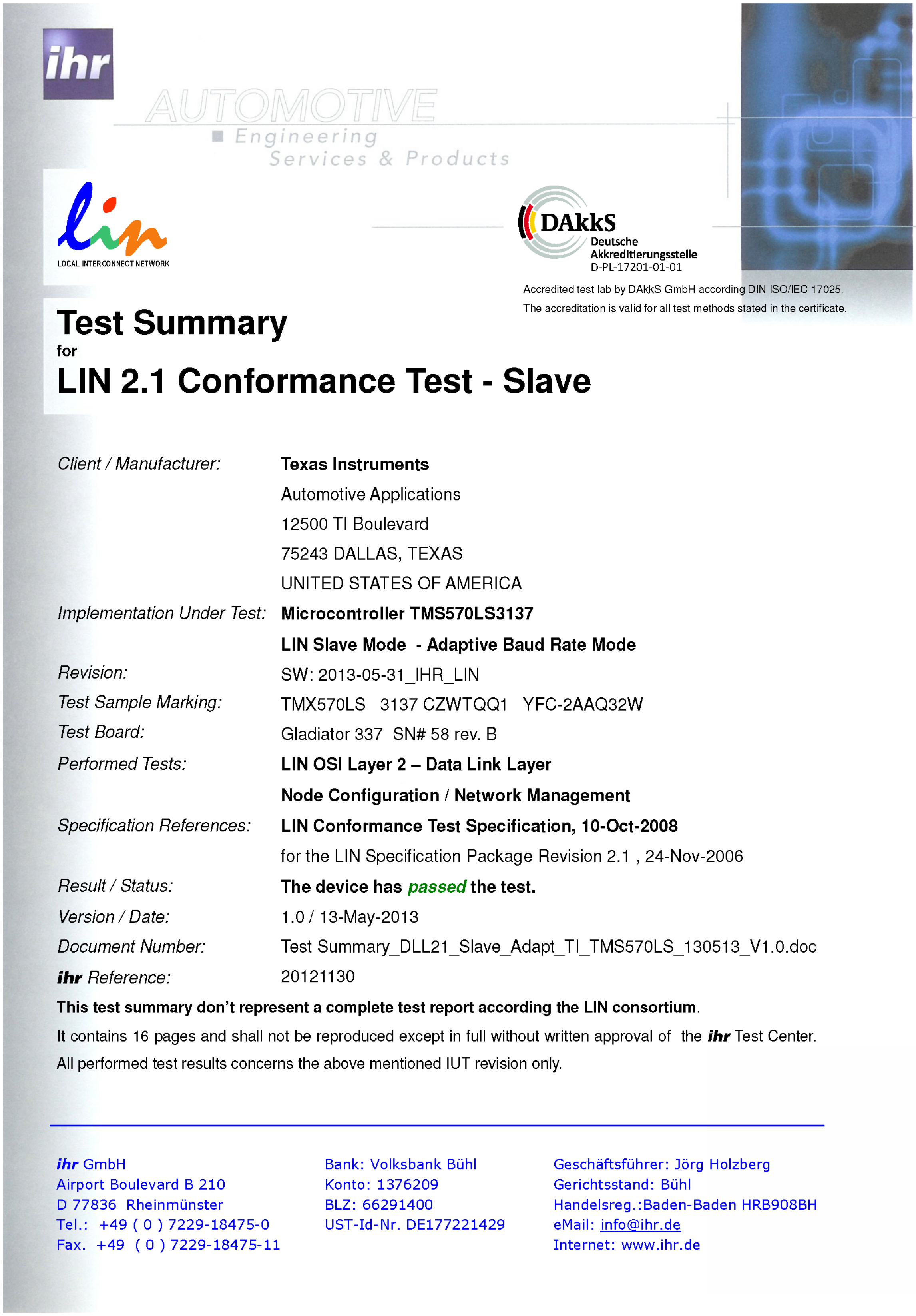
ihr GmbH
Airport Boulevard B 210
D 77836 Rheinmünster
Tel.: +49 (0) 7229-18475-0
Fax. +49 (0) 7229-18475-11

Bank: Volksbank Bühl
Konto: 1376209
BLZ: 66291400
UST-Id-Nr. DE177221429

Geschäftsführer: Jörg Holzberg
Gerichtsstand: Bühl
Handelsreg.:Baden-Baden HRB908BH
eMail: info@ihr.de
Internet: www.ihr.de

Figure 9-6. LIN Certification - Slave Mode - Fixed Baud Rate

9.7.3.3 LIN Slave Mode - Adaptive Baud Rate



ihr

AUTOMOTIVE
Engineering
Services & Products

LOCAL INTERCONNECT NETWORK

DAkkS Deutsche Akkreditierungsstelle D-PL-17201-01-01

Accredited test lab by DAkkS GmbH according DIN ISO/IEC 17025.
The accreditation is valid for all test methods stated in the certificate.

Test Summary
for
LIN 2.1 Conformance Test - Slave

Client / Manufacturer:	**Texas Instruments** Automotive Applications 12500 TI Boulevard 75243 DALLAS, TEXAS UNITED STATES OF AMERICA
Implementation Under Test:	**Microcontroller TMS570LS3137** **LIN Slave Mode - Adaptive Baud Rate Mode**
Revision:	SW: 2013-05-31_IHR_LIN
Test Sample Marking:	TMX570LS 3137 CZWTQQ1 YFC-2AAQ32W
Test Board:	Gladiator 337 SN# 58 rev. B
Performed Tests:	**LIN OSI Layer 2 – Data Link Layer** **Node Configuration / Network Management**
Specification References:	**LIN Conformance Test Specification, 10-Oct-2008** for the LIN Specification Package Revision 2.1 , 24-Nov-2006
Result / Status:	**The device has *passed* the test.**
Version / Date:	1.0 / 13-May-2013
Document Number:	Test Summary_DLL21_Slave_Adapt_TI_TMS570LS_130513_V1.0.doc
***ihr** Reference:*	20121130

This test summary don't represent a complete test report according the LIN consortium.

It contains 16 pages and shall not be reproduced except in full without written approval of the ***ihr*** Test Center.

All performed test results concerns the above mentioned IUT revision only.

ihr GmbH
Airport Boulevard B 210
D 77836 Rheinmünster
Tel.: +49 (0) 7229-18475-0
Fax. +49 (0) 7229-18475-11

Bank: Volksbank Bühl
Konto: 1376209
BLZ: 66291400
UST-Id-Nr. DE177221429

Geschäftsführer: Jörg Holzberg
Gerichtsstand: Bühl
Handelsreg.:Baden-Baden HRB908BH
eMail: info@ihr.de
Internet: www.ihr.de

Figure 9-7. LIN Certification - Slave Mode - Adaptive Baud Rate

10 Mechanical Data

10.1 Packaging Information

The following pages include mechanical packaging and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
TMS5704357BZWTQQ1	ACTIVE	NFBGA	ZWT	337	450	Green (RoHS & no Sb/Br)	SNAGCU	Level-3-260C-168 HR	-40 to 125	TMS570 4357BZWTQQ1	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) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

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

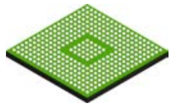
(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

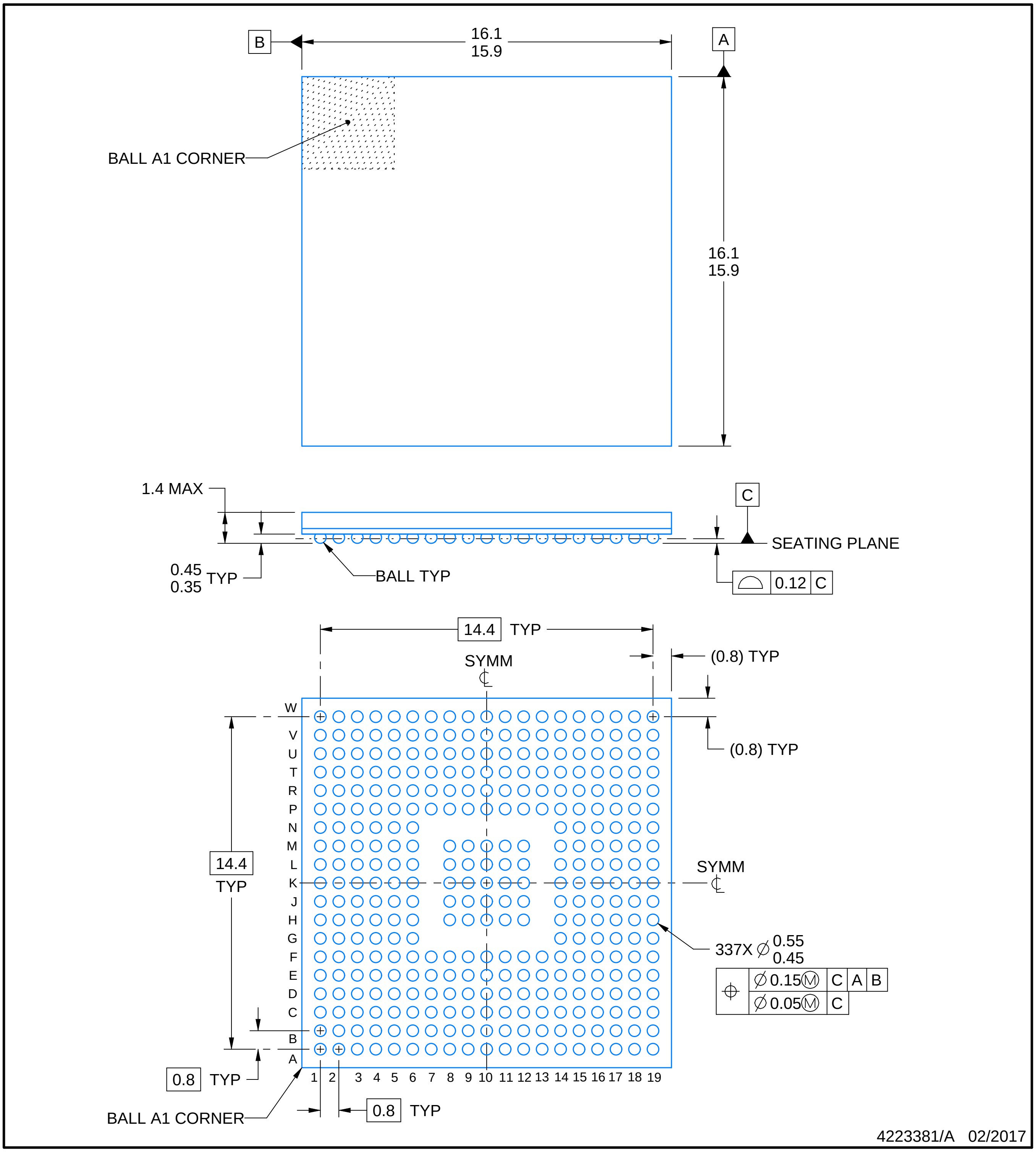
(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

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.

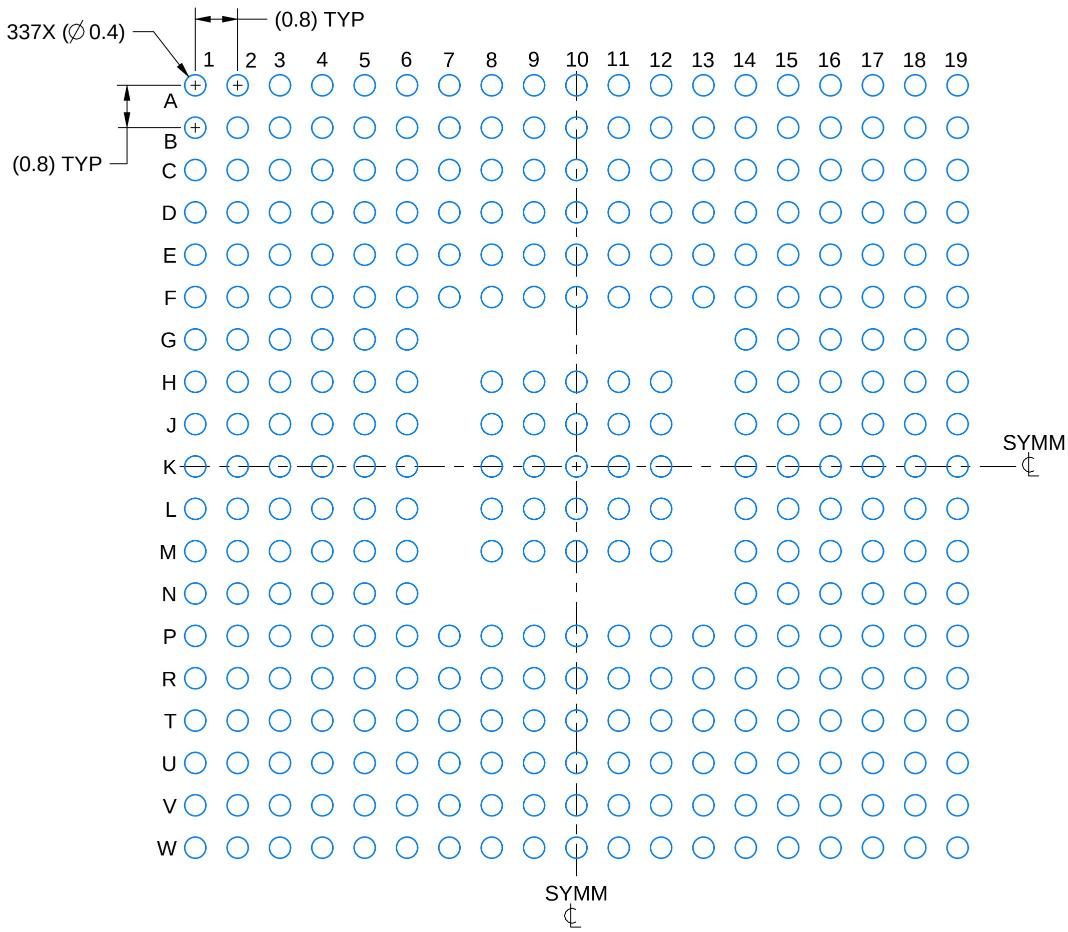
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.



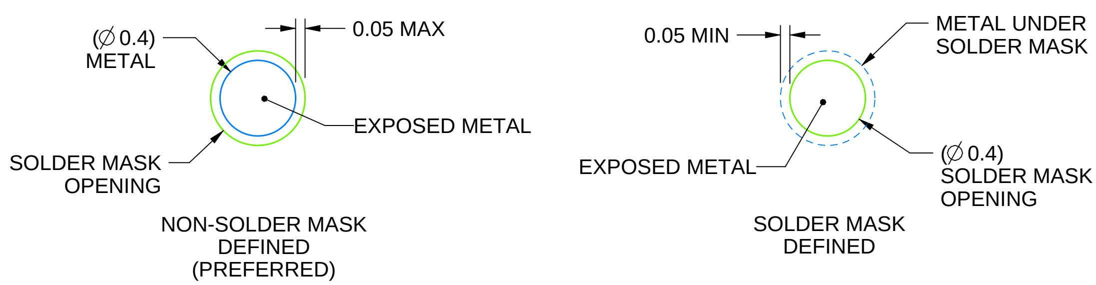


NOTES:

1. All linear dimensions are in millimeters. Any dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.



LAND PATTERN EXAMPLE
EXPOSED METAL SHOWN
SCALE:7X

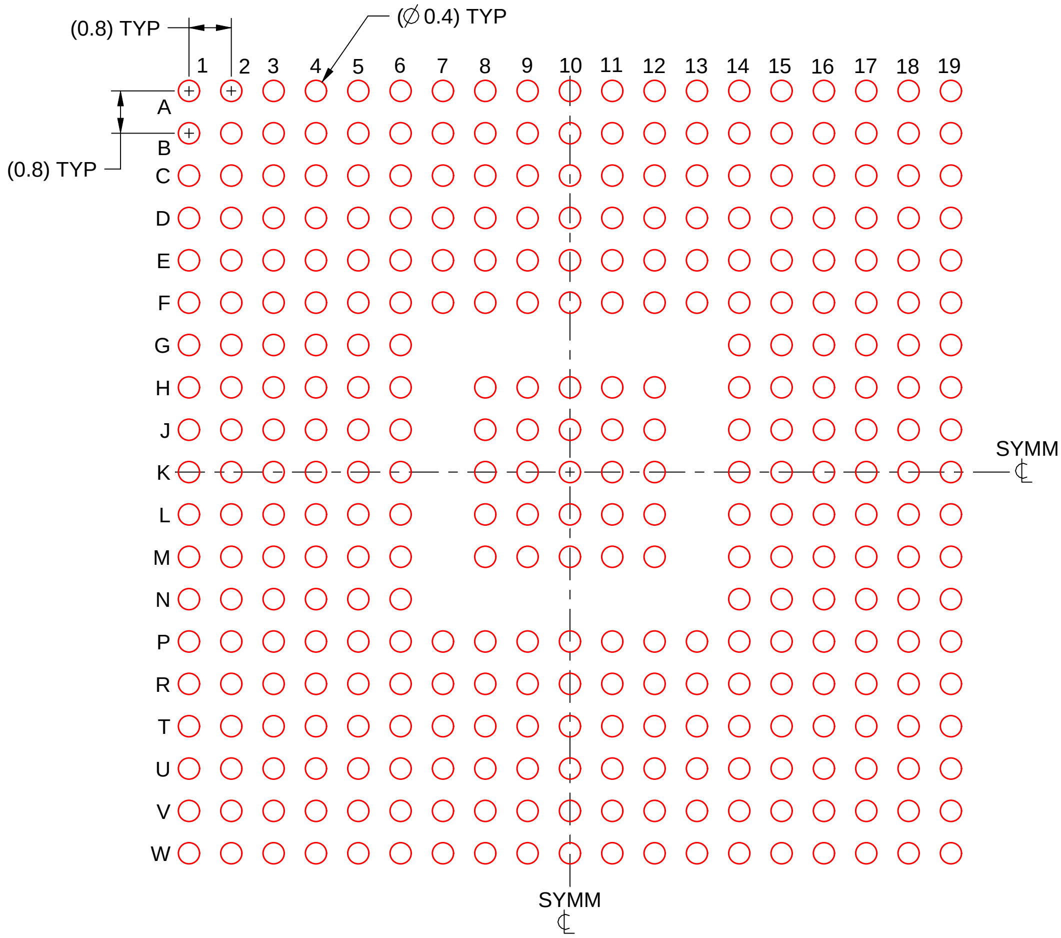


SOLDER MASK DETAILS
NOT TO SCALE

4223381/A 02/2017

NOTES: (continued)

3. Final dimensions may vary due to manufacturing tolerance considerations and also routing constraints. For information, see Texas Instruments literature number SPRAA99 (www.ti.com/lit/spraa99).



SOLDER PASTE EXAMPLE
BASED ON 0.15 mm THICK STENCIL
SCALE:7X

4223381/A 02/2017

NOTES: (continued)

4. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release.