



**THE DATASHEET OF  
TMS320C6743DPTP3**



# TMS320C6743 Fixed- and Floating-Point Digital Signal Processor

## 1 TMS320C6743 Fixed- and Floating-Point Digital Signal Processor

### 1.1 Features

- Applications
  - Networking
  - High-Speed Encoding
  - Professional Audio™
- Software Support
  - TI DSP/BIOS™
  - Chip Support Library and DSP Library
- 375-MHz TMS320C674x Fixed- and Floating-Point VLIW DSP Core
  - Load-Store Architecture with Nonaligned Support
  - 64 General-Purpose Registers (32-Bit)
  - Six ALU (32- and 40-Bit) Functional Units
    - Supports 32-Bit Integer, SP (IEEE Single Precision/32-Bit) and DP (IEEE Double Precision/64-Bit) Floating Point
    - Supports up to Four SP Additions Per Clock, Four DP Additions Every 2 Clocks
    - Supports up to Two Floating Point (SP or DP) Reciprocal Approximation (RCPxP) and Square-Root Reciprocal Approximation (RSQRxP) Operations Per Cycle
  - Two Multiply Functional Units
    - Mixed-Precision IEEE Floating Point Multiply Supported up to:
      - 2 SP x SP -> SP Per Clock
      - 2 SP x SP -> DP Every Two Clocks
      - 2 SP x DP -> DP Every Three Clocks
      - 2 DP x DP -> DP Every Four Clocks
    - Fixed-Point Multiply Supports Two 32 x 32-Bit Multiplies, Four 16 x 16-Bit Multiplies, or Eight 8 x 8-Bit Multiplies per Clock Cycle, and Complex Multiplies
  - Instruction Packing Reduces Code Size
  - All Instructions Conditional
  - Hardware Support for Modulo Loop Operation
  - Protected Mode Operation
  - Exceptions Support for Error Detection and Program Redirection
- C674x Instruction Set Features
  - Superset of the C67x+ and C64x+ ISAs
  - 3000 MIPS and 2250 MFLOPS C674x
  - Byte-Addressable (8-, 16-, 32-, and 64-Bit Data)
  - 8-Bit Overflow Protection
  - Bit-Field Extract, Set, Clear
    - Normalization, Saturation, Bit-Counting
    - Compact 16-Bit Instructions
- C674x Two-Level Cache Memory Architecture
  - 32KB of L1P Program RAM/Cache
  - 32KB of L1D Data RAM/Cache
  - 128KB of L2 Unified Mapped RAM/Cache
  - Flexible RAM/Cache Partition (L1 and L2)
- Enhanced Direct Memory Access Controller 3 (EDMA3):
  - 2 Transfer Controllers
  - 32 Independent DMA Channels
  - 8 Quick DMA Channels
  - Programmable Transfer Burst Size
- 3.3-V LVCMOS I/Os
- Two External Memory Interfaces:
  - EMIFA
    - NOR (8-Bit-Wide Data)
    - NAND (8-Bit-Wide Data)
  - EMIFB
    - 16-bit SDRAM, up to 128MB
- Two Configurable 16550-Type UART Modules:
  - UART0 with Modem Control Signals
  - 16-Byte FIFO
  - 16x or 13x Oversampling Option
- One Serial Peripheral Interface (SPI) with One Chip Select
- Multimedia Card (MMC)/Secure Digital (SD)
- Two Master and Slave Inter-Integrated Circuit (I<sup>2</sup>C Bus™)
- Programmable Real-Time Unit Subsystem (PRUSS)
  - Two Independent Programmable Real-Time Unit (PRU) Cores
    - 32-Bit Load-Store RISC Architecture
    - 4KB of Instruction RAM per Core
    - 512 Bytes of Data RAM per Core
    - PRUSS can be Disabled Through Software to Save Power
    - Register 30 of each PRU is Exported from the Subsystem in Addition to the Normal R31 Output of the PRU Cores
  - Standard Power-Management Mechanism
    - Clock Gating
    - Entire Subsystem Under a Single PSC Clock Gating Domain
  - Dedicated Interrupt Controller



- Dedicated Switched Central Resource
- Two Multichannel Audio Serial Ports (McASPs):
  - Supports TDM, I2S, and Similar Formats
  - FIFO Buffers for Transmit and Receive
- 10/100 Mbps RMI Ethernet Media Access Controller (EMAC):
  - IEEE 802.3 Compliant (3.3-V I/O Only)
  - RMI Media-Independent Interface
  - Management Data I/O (MDIO) Module
- One 64-Bit General-Purpose Timer (Configurable as Two 32-Bit Timers)
- One 64-Bit General-Purpose Watchdog Timer (Configurable as Two 32-Bit Timers)
- Three Enhanced Pulse Width Modulators (eHRPWMs):
  - Dedicated 16-Bit Time-Base Counter with Period and Frequency Control
- 6 Single Edge, 6 Dual Edge Symmetric, or 3 Dual Edge Asymmetric Outputs
- Dead-Band Generation
- PWM Chopping by High-Frequency Carrier
- Trip Zone Input
- Three 32-Bit Event Capture (eCAP) Modules:
  - Configurable as 3 Capture Inputs or 3 Auxiliary Pulse Width Modulator (APWM) Outputs
  - Single-Shot Capture of up to Four Event Time-Stamps
- Two 32-Bit Enhanced Quadrature Encoder Pulse (eQEP) Modules
- 256-Ball Pb-Free Plastic Ball Grid Array (PBGA) [ZKB Suffix], 1.0-mm Ball Pitch
- 176-Pin Thin Quad Flat Pack (TQFP) [PTP Suffix], 0.5-mm Pin Pitch
- Commercial or Automotive Temperature

## 1.2 Applications

- A/V Receivers
- Automotive Amplifiers
- Soundbars
- Home Theatre Systems
- Professional Audio
- Network Streaming Audio

### 1.3 Description

The C6743 device is a low-power digital signal processor based on C674x DSP core. The device consumes significantly lower power than other members of the TMS320C6000™ platform of DSPs.

The C6743 device enables original-equipment manufacturers (OEMs) and original-design manufacturers (ODMs) to quickly bring to market devices featuring high processing performance.

The C6743 DSP core uses a two-level cache-based architecture. The Level 1 program cache (L1P) is a 32-KB direct mapped cache and the Level 1 data cache (L1D) is a 32-KB 2-way set-associative cache. The Level 2 program cache (L2P) consists of a 128-KB of memory space that is shared between program and data space. L2 memory can be configured as mapped memory, cache, or combinations of the two.

The peripheral set includes: a 10/100 Mbps Ethernet MAC (EMAC) with a management data input/output (MDIO) module; two I<sup>2</sup>C Bus interfaces; two multichannel audio serial ports (McASPs) with 14/9 serializers and FIFO buffers; two 64-bit general-purpose timers each configurable (one configurable as watchdog); up to 8 banks of 16 pins of general-purpose input/output (GPIO) with programmable interrupt/event generation modes, multiplexed with other peripherals; two UART interfaces (one with both  $\overline{RTS}$  and  $\overline{CTS}$ ); three enhanced high-resolution pulse width modulator (eHRPWM) peripherals; three 32-bit enhanced capture (eCAP) module peripherals which can be configured as 3 capture inputs or 3 auxiliary pulse width modulator (APWM) outputs; two 32-bit enhanced quadrature encoded pulse (eQEP) peripherals; and 2 external memory interfaces (EMIFs): an asynchronous external memory interface (EMIFA) for slower memories or peripherals, and a higher speed memory interface (EMIFB) for SDRAM.

The Ethernet Media Access Controller (EMAC) provides an efficient interface between the C6743 and the network. The EMAC supports both 10Base-T and 100Base-TX, or 10 Mbps and 100 Mbps in either half- or full-duplex mode. Additionally, an MDIO interface is available for PHY configuration.

The rich peripheral set provides the ability to control external peripheral devices and communicate with external processors. For details on each of the peripherals, see the related sections later in this document and the associated peripheral reference guides.

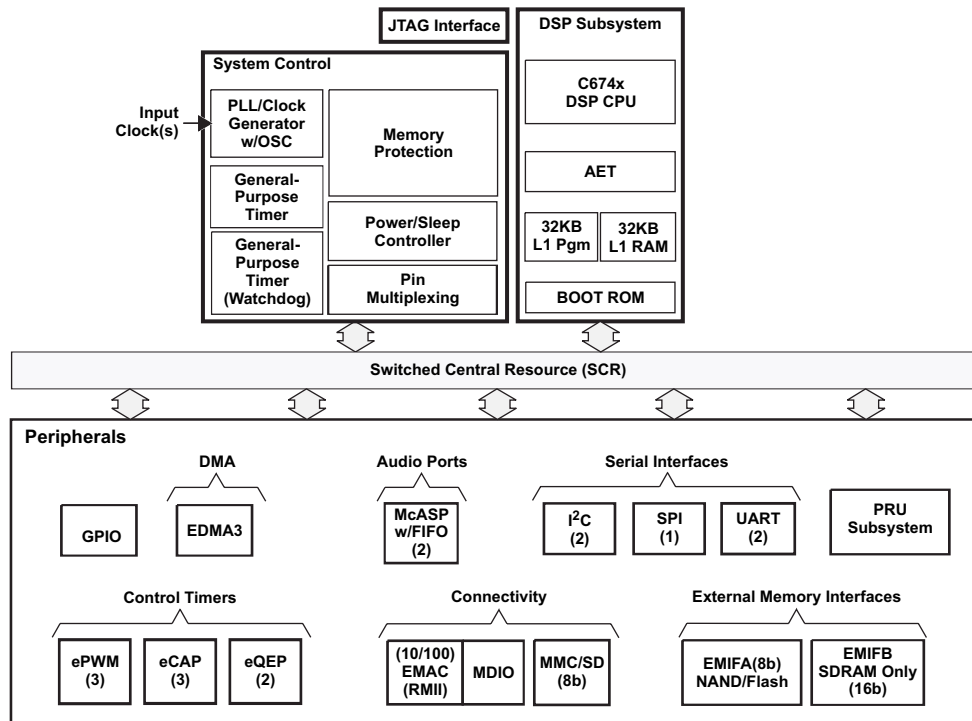
**Device Information<sup>(1)</sup>**

| PART NUMBER | PACKAGE     | BODY SIZE           |
|-------------|-------------|---------------------|
| TMS320C6743 | BGA (256)   | 17.00 mm x 17.00 mm |
| TMS320C6743 | HLQFP (176) | 24.00 mm x 24.00 mm |

(1) For more information on these devices, see [Section 8](#), Mechanical Packaging and Orderable Information.

### 1.4 Functional Block Diagram

Figure 1-1 shows the functional block diagram of the TMS320C6743 device.



(1) Not all peripherals are available at the same time due to multiplexing.

**Figure 1-1. TMS320C6743 Functional Block Diagram**

## Table of Contents

|   |   |
|---|---|
| <b>1 TMS320C6743 Fixed- and Floating-Point Digital Signal Processor</b> ..... <a href="#">1</a>   |   |
| 1.1 Features ..... <a href="#">1</a>  | 6.8 General-Purpose Input/Output (GPIO) ..... <a href="#">56</a>                                |
| 1.2 Applications ..... <a href="#">2</a>  | 6.9 EDMA ..... <a href="#">59</a>   |
| 1.3 Description ..... <a href="#">3</a>   | 6.10 External Memory Interface A (EMIFA) ..... <a href="#">64</a>                               |
| 1.4 Functional Block Diagram ..... <a href="#">4</a>  | 6.11 External Memory Interface B (EMIFB) ..... <a href="#">70</a>                               |
| <b>2 Revision History</b> ..... <a href="#">6</a>   | 6.12 Memory Protection Units ..... <a href="#">76</a>   |
| <b>3 Device Overview</b> ..... <a href="#">7</a>  | 6.13 MMC / SD / SDIO (MMCSD) ..... <a href="#">79</a>   |
| 3.1 Device Characteristics ..... <a href="#">7</a>  | 6.14 Ethernet Media Access Controller (EMAC) ..... <a href="#">82</a>                           |
| 3.2 Device Compatibility ..... <a href="#">8</a>  | 6.15 Management Data Input/Output (MDIO) ..... <a href="#">87</a>                               |
| 3.3 DSP Subsystem ..... <a href="#">8</a>   | 6.16 Multichannel Audio Serial Ports (McASP0, McASP1) ..... <a href="#">89</a>                  |
| 3.4 Memory Map Summary ..... <a href="#">18</a>   | 6.17 Serial Peripheral Interface Ports (SPI0) ..... <a href="#">99</a>                          |
| 3.5 Pin Assignments ..... <a href="#">21</a>  | 6.18 Enhanced Capture (eCAP) Peripheral ..... <a href="#">112</a>                               |
| 3.6 Terminal Functions ..... <a href="#">23</a>   | 6.19 Enhanced Quadrature Encoder (eQEP) Peripheral ..... <a href="#">115</a>                    |
| <b>4 Device Configuration</b> ..... <a href="#">36</a>  | 6.20 Enhanced Pulse Width Modulator (eHRPWM) Modules ..... <a href="#">117</a>                  |
| 4.1 Boot Modes ..... <a href="#">36</a>   | 6.21 Timers ..... <a href="#">121</a>   |
| 4.2 SYSCFG Module ..... <a href="#">36</a>  | 6.22 Inter-Integrated Circuit Serial Ports (I2C0, I2C1) .. <a href="#">123</a>                  |
| 4.3 Pullup/Pulldown Resistors ..... <a href="#">38</a>  | 6.23 Universal Asynchronous Receiver/Transmitter (UART) ..... <a href="#">127</a>               |
| <b>5 Device Operating Conditions</b> ..... <a href="#">39</a>   | 6.24 Power and Sleep Controller (PSC) ..... <a href="#">129</a>                                 |
| 5.1 Absolute Maximum Ratings Over Operating Junction Temperature Range (Unless Otherwise Noted) ..... <a href="#">39</a>                                      | 6.25 Programmable Real-Time Unit Subsystem (PRUSS) ..... <a href="#">132</a>                    |
| 5.2 Handling Ratings ..... <a href="#">39</a>   | 6.26 Emulation Logic ..... <a href="#">135</a>  |
| 5.3 Recommended Operating Conditions ..... <a href="#">40</a>   | 6.27 IEEE 1149.1 JTAG ..... <a href="#">137</a>   |
| 5.4 Notes on Recommended Power-On Hours (POH) . <a href="#">41</a>  | <b>7 Device and Documentation Support</b> ..... <a href="#">139</a>                             |
| 5.5 Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Junction Temperature (Unless Otherwise Noted) ..... <a href="#">42</a> | 7.1 Device Support ..... <a href="#">139</a>  |
| <b>6 Peripheral Information and Electrical Specifications</b> ..... <a href="#">43</a>  | 7.2 Documentation Support ..... <a href="#">140</a>   |
| 6.1 Parameter Information ..... <a href="#">43</a>  | 7.3 Support Resources ..... <a href="#">141</a>   |
| 6.2 Recommended Clock and Control Signal Transition Behavior ..... <a href="#">44</a>   | 7.4 Trademarks ..... <a href="#">141</a>  |
| 6.3 Power Supplies ..... <a href="#">44</a>   | 7.5 Electrostatic Discharge Caution ..... <a href="#">141</a>                                   |
| 6.4 Reset ..... <a href="#">45</a>  | 7.6 Glossary ..... <a href="#">141</a>  |
| 6.5 Crystal Oscillator or External Clock Input ..... <a href="#">47</a>   | <b>8 Mechanical Packaging and Orderable Information</b> ..... <a href="#">142</a>               |
| 6.6 Clock PLLs ..... <a href="#">48</a>   | 8.1 Thermal Data for ZKB ..... <a href="#">142</a>  |
| 6.7 DSP Interrupts ..... <a href="#">52</a>   | 8.2 Thermal Data for PTP ..... <a href="#">143</a>  |
|   | 8.3 Supplementary Information About the 176-pin PTP PowerPAD™ Package ..... <a href="#">143</a> |
|   | 8.4 Mechanical Drawings ..... <a href="#">144</a>   |

## 2 Revision History

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

This data manual revision history highlights the changes made to the SPRS565C device-specific data manual to make it an SPRS565D revision.

### Revision History

| SEE  | ADDITIONS/MODIFICATIONS/DELETIONS   |
|--|---|
| Global   | <ul style="list-style-type: none"> <li>Updated Features, Applications, and Description for consistency and translation.</li> <li>Moved Trademarks information from first page to within <a href="#">Section 7</a>, Device and Documentation Support.</li> <li>Moved ESDS Warning to within <a href="#">Section 7</a>, Device and Documentation Support.</li> </ul>  |
| <a href="#">Section 1.1</a><br>Features  | Deleted Highlights section. Information was duplicated elsewhere in Features.   |
| <a href="#">Section 1.2</a><br>Applications  | Added <b>NEW</b> section.   |
| <a href="#">Section 1.3</a><br>Description   | Added <b>NEW</b> Device Information Table.  |
| <a href="#">Section 1.4</a><br>Functional Block<br>Diagram                             | Updated/Changed text before FBD from "...block diagram of TMS320C6747..." to "...block diagram of TMS320C6743..."<br><a href="#">Figure 1-1</a> , TMS320C6743 Functional Block Diagram: <ul style="list-style-type: none"> <li>Added PRU Subsystem block</li> </ul>   |
| <a href="#">Section 3.3.2.3</a><br>C674x CPU   | <a href="#">Table 3-2</a> , C674x Cache Registers: <ul style="list-style-type: none"> <li>Updated/Changed REGISTER DESCRIPTION for BYTE ADDRESSES 0000, 0020, and 0040 with "(See the Technical Reference Manual..."</li> </ul>   |
| <a href="#">Section 3.6.17</a><br>Reserved and No<br>Connect Terminal<br>Functions     | <a href="#">Table 3-21</a> , Reserved and No Connect Terminal Functions: <ul style="list-style-type: none"> <li>Updated/Changed RSV4 DESCRIPTION from "...This pin may be tied high or low." to "...For proper device operation, this pin must be tied low or to CVDD."</li> </ul>  |
| <a href="#">Section 4</a><br>Device Configuration                                      | <a href="#">Section 4.2</a> , SYSCFG Module: <ul style="list-style-type: none"> <li>Deleted bullet "A special key sequence must be written to KICK0, KICK1..."</li> </ul>   |
| <a href="#">Section 5</a><br>Device Operating<br>Conditions                            | <a href="#">Section 5.2</a> , Handling Ratings: <ul style="list-style-type: none"> <li>Split handling, ratings, and certifications from the Abs Max table and placed in <b>NEW</b> Handling Ratings table.</li> </ul>   |
| <a href="#">Section 5.4</a><br>Notes on<br>Recommended Power-<br>On Hours (POH)        | <a href="#">Table 5-1</a> , Recommended Power-On Hours: <ul style="list-style-type: none"> <li>Added Silicon Revision column.</li> </ul>  |
| <a href="#">Section 6.10.4</a><br>EMIFA Electrical<br>Data/Timing                      | <a href="#">Table 6-20</a> , EMIFA Asynchronous Memory Switching Characteristics; <ul style="list-style-type: none"> <li>Updated/Changed the MIN, NOM, and MAX equations for NO. 3, 10, 15, and 24 from "...(<math>EWC \cdot 16</math>)..." to "...<b>EWC</b>..."</li> </ul>  |
| <a href="#">Section 6.11.3</a><br>EMIFB Electrical<br>Data/Timing                      | <a href="#">Table 6-25</a> , EMIFB SDRAM Interface Switching Characteristics: <ul style="list-style-type: none"> <li>Added new column "<b>Commercial (Default) Temperature Range</b>". Existing column became "<b>Automotive Temperature Range</b>"</li> <li>Updated/Changed PARAMETER No. 1 MIN for Automotive from "6.579" to "<b>7.5</b>"</li> <li>Updated/Changed PARAMETER No. 2 MIN for Automotive from "2.63" to "<b>3</b>"</li> </ul> |
| <a href="#">Section 6.16</a><br>Multichannel Audio<br>Serial Ports (McASP0,<br>McASP1) | <a href="#">Table 6-42</a> , McASP Registers Accessed Through DMA Port: <ul style="list-style-type: none"> <li>Updated/Changed Read Accesses Register Description from "XBUSEL = 0 in XFMT" to "RBUSEL = 0 in RFMT"</li> <li>Updated/Changed Write Accesses Register Description from "RBUSEL = 0 in RFMT" to "XBUSEL = 0 in XFMT"</li> </ul>   |
| <a href="#">Section 7.2</a><br>Documentation<br>Support                                | Updated/Changed "SPRUGJ0 TMS320C6743 DSP System Reference Guide" to "SPRUH90 TMS320C6743 DSP Technical Reference Manual"  |
| <a href="#">Section 7.3</a><br>Community Resources                                     | Added <b>NEW</b> section.   |
| <a href="#">Section 7.6</a><br>Glossary  | Added <b>NEW</b> section.   |

### 3 Device Overview

#### 3.1 Device Characteristics

Table 3-1 provides an overview of the C6743 Low power digital signal processor. The table shows significant features of the device, including the capacity of on-chip RAM, peripherals, and the package type with pin count.

**Table 3-1. Characteristics of the C6743 Processor**

| HARDWARE FEATURES  |   | C6743  |  |
|--|---|--|--|
| Peripherals<br>Not all peripherals pins are available at the same time (for more detail, see the Device Configurations section). | EMIFB   | SDRAM only, 16-bit bus width, up to 128 MB (PTP)   | SDRAM only, 16-bit bus width, up to 128 MB (ZKB) |
|  | EMIFA   | Asynchronous (8-bit bus width) RAM, Flash, NOR, NAND   |  |
|  | Flash Card Interface  | MMC and SD cards supported.  |  |
|  | EDMA3   | 32 independent channels, 8 QDMA channels, 2 Transfer controllers   |  |
|  | Timers  | 2 64-Bit General Purpose (configurable as 2 separate 32-bit timers, 1 configurable as Watch Dog)   |  |
|  | UART  | 2 (One with RTS and CTS flow control)  |  |
|  | SPI   | One with one hardware chip select  |  |
|  | I <sup>2</sup> C  | 2 (both Master/Slave)  |  |
|  | Multichannel Audio Serial Port [McASP]                                  | 2 (each with transmit/receive, FIFO buffer, 14/9 serializers)  |  |
|  | 10/100 Ethernet MAC with Management Data I/O                            | 1 (RMII Interface)   |  |
|  | eHRPWM  | 6 Single Edge, 6 Dual Edge Symmetric, or 3 Dual Edge Asymmetric Outputs  |  |
|  | eCAP  | 3 32-bit capture inputs or 3 32-bit auxiliary PWM outputs  |  |
|  | eQEP  | 2 32-bit QEP channels with 4 inputs/channel  |  |
|  | General-Purpose Input/Output Port                                       | 8 banks of 16-bit  |  |
| PRU Subsystem (PRUSS)  | 2 Programmable PRU Cores  |  |  |
| On-Chip Memory   | Size (Bytes)  | 320KB RAM  |  |
|  | Organization  | <b>DSP</b><br>32KB L1 Program (L1P)/Cache (up to 32KB)<br>32KB L1 Data (L1D)/Cache (up to 32KB)<br>128KB Unified Mapped RAM/Cache (L2)<br>DSP Memories can be made accessible to EDMA3, and other peripherals. |  |
| C674x CPU ID + CPU Rev ID  | Control Status Register (CSR.[31:16])                                   | 0x1400   |  |
| C674x Megamodule Revision  | Revision ID Register (MM_REVID[15:0])                                   | 0x0000   |  |
| JTAG BSDL_ID   | DEVIDR0 register  | 0x8B7DF02F (Silicon Revision 1.0)<br>0x8B7DF02F (Silicon Revision 1.1)<br>0x9B7DF02F (Silicon Revision 3.0, 2.1, and 2.0)  |  |
| CPU Frequency  | MHz   | C674x DSP 375/(200) MHz  |  |
| Voltage  | Core (V)  | 1.2 V  |  |
|  | I/O (V)   | 3.3 V  |  |
| Package  |   | 24 mm x 24 mm, 176-Pin, 0.5 mm pitch, TQFP (PTP)   | 17 mm x 17 mm, 256-Ball 1 mm pitch, PBGA (ZKB)   |
| Product Status <sup>(1)</sup>  | Product Preview (PP), Advance Information (AI), or Production Data (PD) | PD   |  |

(1) PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

## 3.2 Device Compatibility

The C674x DSP core is code-compatible with the C6000™ DSP platform and supports features of both the C64x+ and C67x+ DSP families.

## 3.3 DSP Subsystem

The DSP Subsystem includes the following features:

- C674x DSP CPU
- 32KB L1 Program (L1P)/Cache (up to 32KB)
- 32KB L1 Data (L1D)/Cache (up to 32KB)
- 128KB Unified Mapped RAM/Cache (L2)
- Boot ROM (cannot be used for application code)
- Little endian

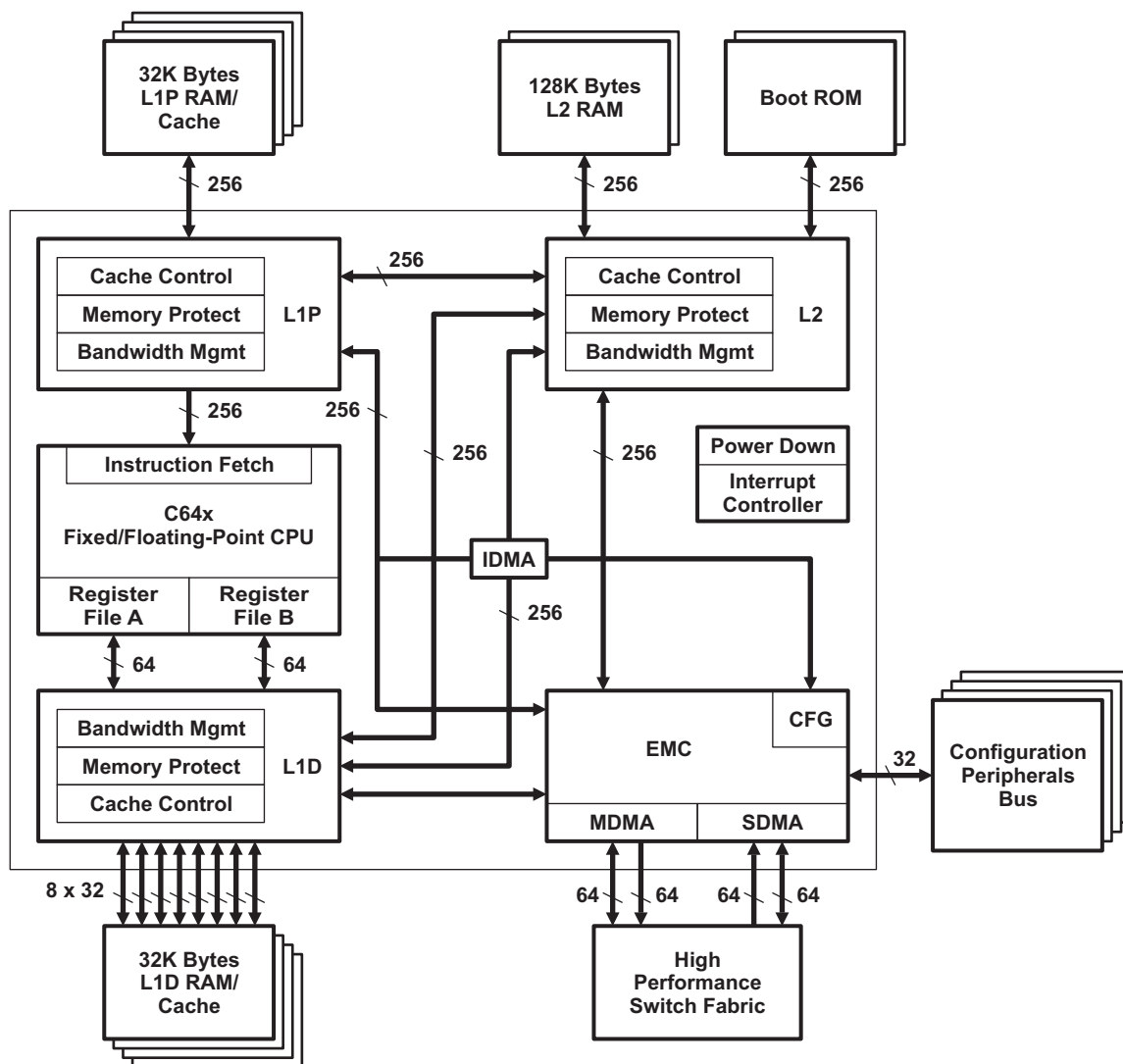


Figure 3-1. C674x Megamodule Block Diagram

### 3.3.1 C674x DSP CPU Description

The C674x Central Processing Unit (CPU) consists of eight functional units, two register files, and two data paths as shown in [Figure 3-2](#). The two general-purpose register files (A and B) each contain 32 32-bit registers for a total of 64 registers. The general-purpose registers can be used for data or can be data address pointers. The data types supported include packed 8-bit data, packed 16-bit data, 32-bit data, 40-bit data, and 64-bit data. Values larger than 32 bits, such as 40-bit-long or 64-bit-long values are stored in register pairs, with the 32 LSBs of data placed in an even register and the remaining 8 or 32 MSBs in the next upper register (which is always an odd-numbered register).

The eight functional units (.M1, .L1, .D1, .S1, .M2, .L2, .D2, and .S2) are each capable of executing one instruction every clock cycle. The .M functional units perform all multiply operations. The .S and .L units perform a general set of arithmetic, logical, and branch functions. The .D units primarily load data from memory to the register file and store results from the register file into memory.

Each C674x .M unit can perform one of the following each clock cycle: one 32 x 32 bit multiply, one 16 x 32 bit multiply, two 16 x 16 bit multiplies, two 16 x 32 bit multiplies, two 16 x 16 bit multiplies with add/subtract capabilities, four 8 x 8 bit multiplies, four 8 x 8 bit multiplies with add operations, and four 16 x 16 multiplies with add/subtract capabilities (including a complex multiply). There is also support for Galois field multiplication for 8-bit and 32-bit data. Many communications algorithms such as FFTs and modems require complex multiplication. The complex multiply (CMPY) instruction takes four 16-bit inputs and produces a 32-bit real and a 32-bit imaginary output. There are also complex multiplies with rounding capability that produces one 32-bit packed output that contain 16-bit real and 16-bit imaginary values. The 32 x 32 bit multiply instructions provide the extended precision necessary for high-precision algorithms on a variety of signed and unsigned 32-bit data types.

The .L or (Arithmetic Logic Unit) now incorporates the ability to do parallel add/subtract operations on a pair of common inputs. Versions of this instruction exist to work on 32-bit data or on pairs of 16-bit data performing dual 16-bit add and subtracts in parallel. There are also saturated forms of these instructions.

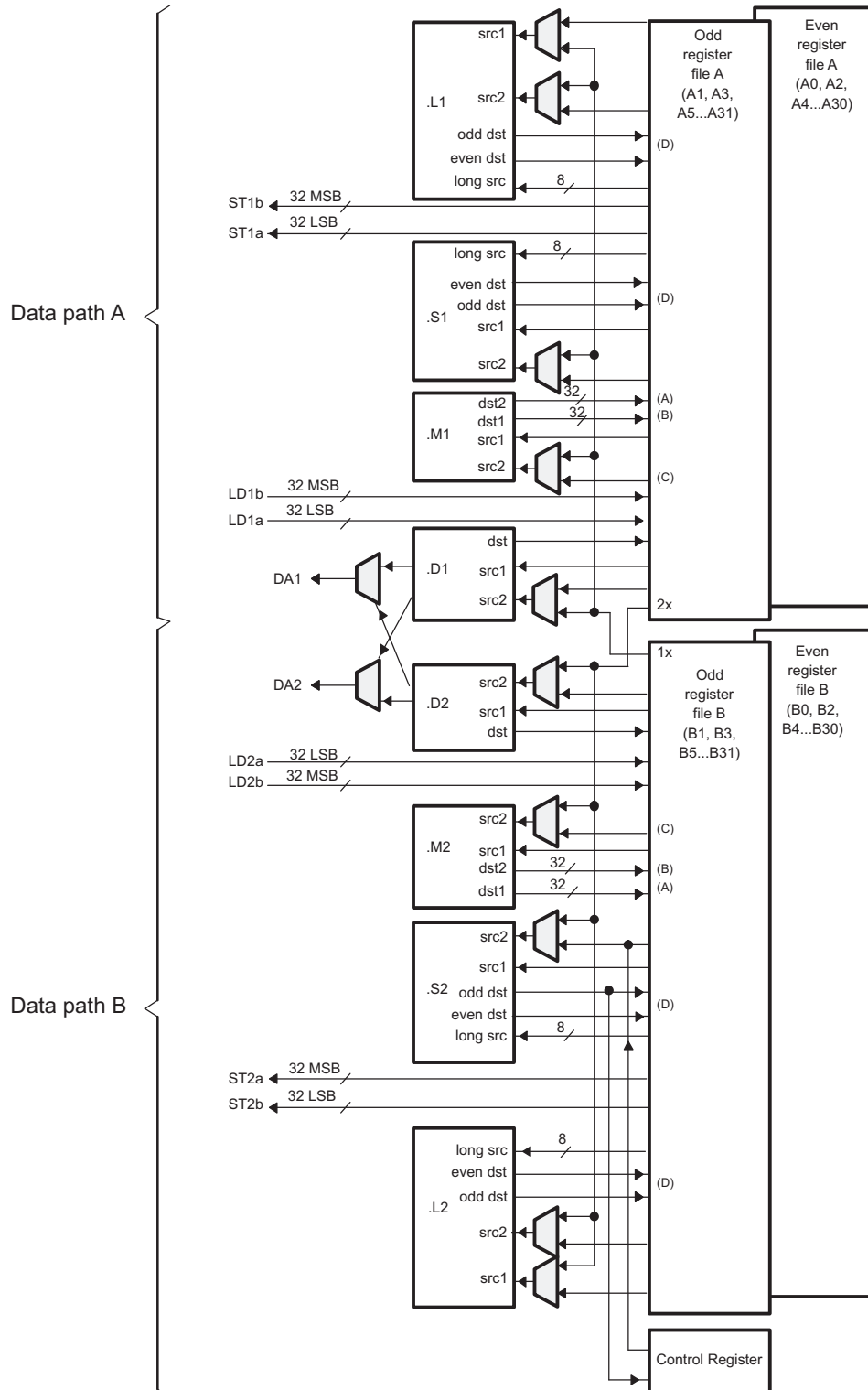
The C674x core enhances the .S unit in several ways. On the previous cores, dual 16-bit MIN2 and MAX2 comparisons were only available on the .L units. On the C674x core they are also available on the .S unit which increases the performance of algorithms that do searching and sorting. Finally, to increase data packing and unpacking throughput, the .S unit allows sustained high performance for the quad 8-bit/16-bit and dual 16-bit instructions. Unpack instructions prepare 8-bit data for parallel 16-bit operations. Pack instructions return parallel results to output precision including saturation support.

Other new features include:

- **SPLOOP** - A small instruction buffer in the CPU that aids in creation of software pipelining loops where multiple iterations of a loop are executed in parallel. The SPLOOP buffer reduces the code size associated with software pipelining. Furthermore, loops in the SPLOOP buffer are fully interruptible.
- **Compact Instructions** - The native instruction size for the C6000 devices is 32 bits. Many common instructions such as MPY, AND, OR, ADD, and SUB can be expressed as 16 bits if the C674x compiler can restrict the code to use certain registers in the register file. This compression is performed by the code generation tools.
- **Instruction Set Enhancement** - As noted above, there are new instructions such as 32-bit multiplications, complex multiplications, packing, sorting, bit manipulation, and 32-bit Galois field multiplication.
- **Exceptions Handling** - Intended to aid the programmer in isolating bugs. The C674x CPU is able to detect and respond to exceptions, both from internally detected sources (such as illegal op-codes) and from system events (such as a watchdog time expiration).
- **Privilege** - Defines user and supervisor modes of operation, allowing the operating system to give a basic level of protection to sensitive resources. Local memory is divided into multiple pages, each with read, write, and execute permissions.
- **Time-Stamp Counter** - Primarily targeted for Real-Time Operating System (RTOS) robustness, a free-running time-stamp counter is implemented in the CPU which is **not** sensitive to system stalls.

For more details on the C674x CPU and its enhancements over the C64x architecture, see the following documents:

- *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* ([SPRU732](#))
- *TMS320C64x Technical Overview* ([SPRU395](#))



- A. On .M unit, dst2 is 32 MSB.
- B. On .M unit, dst1 is 32 LSB.
- C. On C64x CPU .M unit, src2 is 32 bits; on C64x+ CPU .M unit, src2 is 64 bits.
- D. On .L and .S units, odd dst connects to odd register files and even dst connects to even register files.

**Figure 3-2. TMS320C674x CPU (DSP Core) Data Paths**

### 3.3.2 DSP Memory Mapping

The DSP memory map is shown in [Section 3.4](#).

#### 3.3.2.1 External Memories

The DSP has access to the following External memories:

- Asynchronous EMIF / NAND / NOR Flash (EMIFA)
- SDRAM (EMIFB)

#### 3.3.2.2 DSP Internal Memories

The DSP has access to the following DSP memories:

- L2 RAM
- L1P RAM
- L1D RAM

#### 3.3.2.3 C674x CPU

The C674x core uses a two-level cache-based architecture. The Level 1 Program cache (L1P) is 32 KB direct mapped cache and the Level 1 Data cache (L1D) is 32 KB 2-way set associated cache. The Level 2 memory/cache (L2) consists of a 128 KB memory space that is shared between program and data space. L2 memory can be configured as mapped memory, cache, or a combination of both.

[Table 3-2](#) shows a memory map of the C674x CPU cache registers for the device.

**Table 3-2. C674x Cache Registers**

| HEX ADDRESS RANGE         | REGISTER NAME | DESCRIPTION  |
|---------------------------|---------------|--|
| 0x0184 0000               | L2CFG         | L2 Cache configuration register (See the Technical Reference Manual <a href="#">SPRUH90</a> for reset configuration)       |
| 0x0184 0020               | L1PCFG        | L1P Size Cache configuration register (See the Technical Reference Manual <a href="#">SPRUH90</a> for reset configuration) |
| 0x0184 0024               | L1PCC         | L1P Freeze Mode Cache configuration register   |
| 0x0184 0040               | L1DCFG        | L1D Size Cache configuration register (See the Technical Reference Manual <a href="#">SPRUH90</a> for reset configuration) |
| 0x0184 0044               | L1DCC         | L1D Freeze Mode Cache configuration register   |
| 0x0184 0048 - 0x0184 0FFC | -             | Reserved   |
| 0x0184 1000               | EDMAWEIGHT    | L2 EDMA access control register  |
| 0x0184 1004 - 0x0184 1FFC | -             | Reserved   |
| 0x0184 2000               | L2ALLOC0      | L2 allocation register 0   |
| 0x0184 2004               | L2ALLOC1      | L2 allocation register 1   |
| 0x0184 2008               | L2ALLOC2      | L2 allocation register 2   |
| 0x0184 200C               | L2ALLOC3      | L2 allocation register 3   |
| 0x0184 2010 - 0x0184 3FFF | -             | Reserved   |
| 0x0184 4000               | L2WBAR        | L2 writeback base address register   |
| 0x0184 4004               | L2WWC         | L2 writeback word count register   |
| 0x0184 4010               | L2WIBAR       | L2 writeback invalidate base address register  |
| 0x0184 4014               | L2WIWC        | L2 writeback invalidate word count register  |
| 0x0184 4018               | L2IBAR        | L2 invalidate base address register  |
| 0x0184 401C               | L2IWC         | L2 invalidate word count register  |
| 0x0184 4020               | L1PIBAR       | L1P invalidate base address register   |
| 0x0184 4024               | L1PIWC        | L1P invalidate word count register   |
| 0x0184 4030               | L1DWIBAR      | L1D writeback invalidate base address register   |
| 0x0184 4034               | L1DWIWC       | L1D writeback invalidate word count register   |

**Table 3-2. C674x Cache Registers (continued)**

| HEX ADDRESS RANGE         | REGISTER NAME   | DESCRIPTION   |
|---------------------------|-----------------|---|
| 0x0184 4038               | -               | Reserved  |
| 0x0184 4040               | L1DWBAR         | L1D Block Writeback   |
| 0x0184 4044               | L1DWWC          | L1D Block Writeback   |
| 0x0184 4048               | L1DIBAR         | L1D invalidate base address register  |
| 0x0184 404C               | L1DIWC          | L1D invalidate word count register  |
| 0x0184 4050 - 0x0184 4FFF | -               | Reserved  |
| 0x0184 5000               | L2WB            | L2 writeback all register   |
| 0x0184 5004               | L2WBINV         | L2 writeback invalidate all register  |
| 0x0184 5008               | L2INV           | L2 Global Invalidate without writeback  |
| 0x0184 500C - 0x0184 5027 | -               | Reserved  |
| 0x0184 5028               | L1PINV          | L1P Global Invalidate   |
| 0x0184 502C - 0x0184 5039 | -               | Reserved  |
| 0x0184 5040               | L1DWB           | L1D Global Writeback  |
| 0x0184 5044               | L1DWBINV        | L1D Global Writeback with Invalidate  |
| 0x0184 5048               | L1DINV          | L1D Global Invalidate without writeback   |
| 0x0184 8000 – 0x0184 80FF | MAR0 - MAR63    | Reserved 0x0000 0000 – 0x3FFF FFFF  |
| 0x0184 8100 – 0x0184 817F | MAR64 – MAR95   | Reserved 0x4000 0000 – 0x5FFF FFFF  |
| 0x0184 8180 – 0x0184 8187 | MAR96 - MAR97   | Memory Attribute Registers for EMIFA Async Data (CS2) 0x6000 0000 – 0x61FF FFFF |
| 0x0184 8188 – 0x0184 818F | MAR98 – MAR99   | Memory Attribute Registers for EMIFA Async Data (CS3) 0x6200 0000 – 0x63FF FFFF |
| 0x0184 8190 – 0x0184 8197 | MAR100 – MAR101 | Reserved 0x6400 0000 – 0x65FF FFFF  |
| 0x0184 8198 – 0x0184 819F | MAR102 – MAR103 | Reserved 0x6600 0000 – 0x67FF FFFF  |
| 0x0184 81A0 – 0x0184 81FF | MAR104 – MAR127 | Reserved 0x6800 0000 – 0x7FFF FFFF  |
| 0x0184 8200               | MAR128          | Reserved 0x8000 0000 – 0x81FF FFFF  |
| 0x0184 8204 – 0x0184 82FF | MAR129 – MAR191 | Reserved 0x8200 0000 – 0xBFFF FFFF  |
| 0x0184 8300 – 0x0184 837F | MAR192 – MAR223 | Memory Attribute Registers for EMIFB SDRAM Data (CS0) 0xC000 0000 – 0xDFFF FFFF |
| 0x0184 8380 – 0x0184 83FF | MAR224 – MAR255 | Reserved 0xE000 0000 – 0xFFFF FFFF  |

**Table 3-3. C674x L1/L2 Memory Protection Registers**

| HEX ADDRESS RANGE         | REGISTER NAME | DESCRIPTION   |
|---------------------------|---------------|---|
| 0x0184 A000               | L2MPFAR       | L2 memory protection fault address register   |
| 0x0184 A004               | L2MPFSR       | L2 memory protection fault status register  |
| 0x0184 A008               | L2MPFCR       | L2 memory protection fault command register   |
| 0x0184 A00C - 0x0184 A0FF | -             | Reserved  |
| 0x0184 A100               | L2MPLK0       | L2 memory protection lock key bits [31:0]   |
| 0x0184 A104               | L2MPLK1       | L2 memory protection lock key bits [63:32]  |
| 0x0184 A108               | L2MPLK2       | L2 memory protection lock key bits [95:64]  |
| 0x0184 A10C               | L2MPLK3       | L2 memory protection lock key bits [127:96]   |
| 0x0184 A110               | L2MPLKCMD     | L2 memory protection lock key command register  |
| 0x0184 A114               | L2MPLKSTAT    | L2 memory protection lock key status register   |
| 0x0184 A118 - 0x0184 A1FF | -             | Reserved  |
| 0x0184 A200               | L2MPPA0       | L2 memory protection page attribute register 0<br>(controls memory address 0x0080 0000 - 0x0080 1FFF) |
| 0x0184 A204               | L2MPPA1       | L2 memory protection page attribute register 1<br>(controls memory address 0x0080 2000 - 0x0080 3FFF) |
| 0x0184 A208               | L2MPPA2       | L2 memory protection page attribute register 2<br>(controls memory address 0x0080 4000 - 0x0080 5FFF) |

**Table 3-3. C674x L1/L2 Memory Protection Registers (continued)**

| HEX ADDRESS RANGE | REGISTER NAME | DESCRIPTION  |
|-------------------|---------------|--|
| 0x0184 A20C       | L2MPPA3       | L2 memory protection page attribute register 3<br>(controls memory address 0x0080 6000 - 0x0080 7FFF)  |
| 0x0184 A210       | L2MPPA4       | L2 memory protection page attribute register 4<br>(controls memory address 0x0080 8000 - 0x0080 9FFF)  |
| 0x0184 A214       | L2MPPA5       | L2 memory protection page attribute register 5<br>(controls memory address 0x0080 A000 - 0x0080 BFFF)  |
| 0x0184 A218       | L2MPPA6       | L2 memory protection page attribute register 6<br>(controls memory address 0x0080 C000 - 0x0080 DFFF)  |
| 0x0184 A21C       | L2MPPA7       | L2 memory protection page attribute register 7<br>(controls memory address 0x0080 E000 - 0x0080 FFFF)  |
| 0x0184 A220       | L2MPPA8       | L2 memory protection page attribute register 8<br>(controls memory address 0x0081 0000 - 0x0081 1FFF)  |
| 0x0184 A224       | L2MPPA9       | L2 memory protection page attribute register 9<br>(controls memory address 0x0081 2000 - 0x0081 3FFF)  |
| 0x0184 A228       | L2MPPA10      | L2 memory protection page attribute register 10<br>(controls memory address 0x0081 4000 - 0x0081 5FFF) |
| 0x0184 A22C       | L2MPPA11      | L2 memory protection page attribute register 11<br>(controls memory address 0x0081 6000 - 0x0081 7FFF) |
| 0x0184 A230       | L2MPPA12      | L2 memory protection page attribute register 12<br>(controls memory address 0x0081 8000 - 0x0081 9FFF) |
| 0x0184 A234       | L2MPPA13      | L2 memory protection page attribute register 13<br>(controls memory address 0x0081 A000 - 0x0081 BFFF) |
| 0x0184 A238       | L2MPPA14      | L2 memory protection page attribute register 14<br>(controls memory address 0x0081 C000 - 0x0081 DFFF) |
| 0x0184 A23C       | L2MPPA15      | L2 memory protection page attribute register 15<br>(controls memory address 0x0081 E000 - 0x0081 FFFF) |
| 0x0184 A240       | L2MPPA16      | L2 memory protection page attribute register 16<br>(controls memory address 0x0082 0000 - 0x0082 1FFF) |
| 0x0184 A244       | L2MPPA17      | L2 memory protection page attribute register 17<br>(controls memory address 0x0082 2000 - 0x0082 3FFF) |
| 0x0184 A248       | L2MPPA18      | L2 memory protection page attribute register 18<br>(controls memory address 0x0082 4000 - 0x0082 5FFF) |
| 0x0184 A24C       | L2MPPA19      | L2 memory protection page attribute register 19<br>(controls memory address 0x0082 6000 - 0x0082 7FFF) |
| 0x0184 A250       | L2MPPA20      | L2 memory protection page attribute register 20<br>(controls memory address 0x0082 8000 - 0x0082 9FFF) |
| 0x0184 A254       | L2MPPA21      | L2 memory protection page attribute register 21<br>(controls memory address 0x0082 A000 - 0x0082 BFFF) |
| 0x0184 A258       | L2MPPA22      | L2 memory protection page attribute register 22<br>(controls memory address 0x0082 C000 - 0x0082 DFFF) |
| 0x0184 A25C       | L2MPPA23      | L2 memory protection page attribute register 23<br>(controls memory address 0x0082 E000 - 0x0082 FFFF) |
| 0x0184 A260       | L2MPPA24      | L2 memory protection page attribute register 24<br>(controls memory address 0x0083 0000 - 0x0083 1FFF) |
| 0x0184 A264       | L2MPPA25      | L2 memory protection page attribute register 25<br>(controls memory address 0x0083 2000 - 0x0083 3FFF) |
| 0x0184 A268       | L2MPPA26      | L2 memory protection page attribute register 26<br>(controls memory address 0x0083 4000 - 0x0083 5FFF) |
| 0x0184 A26C       | L2MPPA27      | L2 memory protection page attribute register 27<br>(controls memory address 0x0083 6000 - 0x0083 7FFF) |
| 0x0184 A270       | L2MPPA28      | L2 memory protection page attribute register 28<br>(controls memory address 0x0083 8000 - 0x0083 9FFF) |
| 0x0184 A274       | L2MPPA29      | L2 memory protection page attribute register 29<br>(controls memory address 0x0083 A000 - 0x0083 BFFF) |
| 0x0184 A278       | L2MPPA30      | L2 memory protection page attribute register 30<br>(controls memory address 0x0083 C000 - 0x0083 DFFF) |

**Table 3-3. C674x L1/L2 Memory Protection Registers (continued)**

| HEX ADDRESS RANGE | REGISTER NAME | DESCRIPTION  |
|-------------------|---------------|--|
| 0x0184 A27C       | L2MPPA31      | L2 memory protection page attribute register 31<br>(controls memory address 0x0083 E000 - 0x0083 FFFF) |
| 0x0184 A280       | L2MPPA32      | L2 memory protection page attribute register 32<br>(controls memory address 0x0070 0000 - 0x0070 7FFF) |
| 0x0184 A284       | L2MPPA33      | L2 memory protection page attribute register 33<br>(controls memory address 0x0070 8000 - 0x0070 FFFF) |
| 0x0184 A288       | L2MPPA34      | L2 memory protection page attribute register 34<br>(controls memory address 0x0071 0000 - 0x0071 7FFF) |
| 0x0184 A28C       | L2MPPA35      | L2 memory protection page attribute register 35<br>(controls memory address 0x0071 8000 - 0x0071 FFFF) |
| 0x0184 A290       | L2MPPA36      | L2 memory protection page attribute register 36<br>(controls memory address 0x0072 0000 - 0x0072 7FFF) |
| 0x0184 A294       | L2MPPA37      | L2 memory protection page attribute register 37<br>(controls memory address 0x0072 8000 - 0x0072 FFFF) |
| 0x0184 A298       | L2MPPA38      | L2 memory protection page attribute register 38<br>(controls memory address 0x0073 0000 - 0x0073 7FFF) |
| 0x0184 A29C       | L2MPPA39      | L2 memory protection page attribute register 39<br>(controls memory address 0x0073 8000 - 0x0073 FFFF) |
| 0x0184 A2A0       | L2MPPA40      | L2 memory protection page attribute register 40<br>(controls memory address 0x0074 0000 - 0x0074 7FFF) |
| 0x0184 A2A4       | L2MPPA41      | L2 memory protection page attribute register 41<br>(controls memory address 0x0074 8000 - 0x0074 FFFF) |
| 0x0184 A2A8       | L2MPPA42      | L2 memory protection page attribute register 42<br>(controls memory address 0x0075 0000 - 0x0075 7FFF) |
| 0x0184 A2AC       | L2MPPA43      | L2 memory protection page attribute register 43<br>(controls memory address 0x0075 8000 - 0x0075 FFFF) |
| 0x0184 A2B0       | L2MPPA44      | L2 memory protection page attribute register 44<br>(controls memory address 0x0076 0000 - 0x0076 7FFF) |
| 0x0184 A2B4       | L2MPPA45      | L2 memory protection page attribute register 45<br>(controls memory address 0x0076 8000 - 0x0076 FFFF) |
| 0x0184 A2B8       | L2MPPA46      | L2 memory protection page attribute register 46<br>(controls memory address 0x0077 0000 - 0x0077 7FFF) |
| 0x0184 A2BC       | L2MPPA47      | L2 memory protection page attribute register 47<br>(controls memory address 0x0077 8000 - 0x0077 FFFF) |
| 0x0184 A2C0       | L2MPPA48      | L2 memory protection page attribute register 48<br>(controls memory address 0x0078 0000 - 0x0078 7FFF) |
| 0x0184 A2C4       | L2MPPA49      | L2 memory protection page attribute register 49<br>(controls memory address 0x0078 8000 - 0x0078 FFFF) |
| 0x0184 A2C8       | L2MPPA50      | L2 memory protection page attribute register 50<br>(controls memory address 0x0079 0000 - 0x0079 7FFF) |
| 0x0184 A2CC       | L2MPPA51      | L2 memory protection page attribute register 51<br>(controls memory address 0x0079 8000 - 0x0079 FFFF) |
| 0x0184 A2D0       | L2MPPA52      | L2 memory protection page attribute register 52<br>(controls memory address 0x007A 0000 - 0x007A 7FFF) |
| 0x0184 A2D4       | L2MPPA53      | L2 memory protection page attribute register 53<br>(controls memory address 0x007A 8000 - 0x007A FFFF) |
| 0x0184 A2D8       | L2MPPA54      | L2 memory protection page attribute register 54<br>(controls memory address 0x007B 0000 - 0x007B 7FFF) |
| 0x0184 A2DC       | L2MPPA55      | L2 memory protection page attribute register 55<br>(controls memory address 0x007B 8000 - 0x007B FFFF) |
| 0x0184 A2E0       | L2MPPA56      | L2 memory protection page attribute register 56<br>(controls memory address 0x007C 0000 - 0x007C 7FFF) |
| 0x0184 A2E4       | L2MPPA57      | L2 memory protection page attribute register 57<br>(controls memory address 0x007C 8000 - 0x007C FFFF) |
| 0x0184 A2E8       | L2MPPA58      | L2 memory protection page attribute register 58<br>(controls memory address 0x007D 0000 - 0x007D 7FFF) |

**Table 3-3. C674x L1/L2 Memory Protection Registers (continued)**

| HEX ADDRESS RANGE         | REGISTER NAME | DESCRIPTION   |
|---------------------------|---------------|---|
| 0x0184 A2EC               | L2MPPA59      | L2 memory protection page attribute register 59<br>(controls memory address 0x007D 8000 - 0x007D FFFF)  |
| 0x0184 A2F0               | L2MPPA60      | L2 memory protection page attribute register 60<br>(controls memory address 0x007E 0000 - 0x007E 7FFF)  |
| 0x0184 A2F4               | L2MPPA61      | L2 memory protection page attribute register 61<br>(controls memory address 0x007E 8000 - 0x007E FFFF)  |
| 0x0184 A2F8               | L2MPPA62      | L2 memory protection page attribute register 62<br>(controls memory address 0x007F 0000 - 0x007F 7FFF)  |
| 0x0184 A2FC               | L2MPPA63      | L2 memory protection page attribute register 63<br>(controls memory address 0x007F 8000 - 0x007F FFFF)  |
| 0x0184 A300 - 0x0184 A3FF | -             | Reserved  |
| 0x0184 A400               | L1PMPFAR      | L1P memory protection fault address register  |
| 0x0184 A404               | L1PMPFSR      | L1P memory protection fault status register   |
| 0x0184 A408               | L1PMPFCR      | L1P memory protection fault command register  |
| 0x0184 A40C - 0x0184 A4FF | -             | Reserved  |
| 0x0184 A500               | L1PMPLK0      | L1P memory protection lock key bits [31:0]  |
| 0x0184 A504               | L1PMPLK1      | L1P memory protection lock key bits [63:32]   |
| 0x0184 A508               | L1PMPLK2      | L1P memory protection lock key bits [95:64]   |
| 0x0184 A50C               | L1PMPLK3      | L1P memory protection lock key bits [127:96]  |
| 0x0184 A510               | L1PMPLKCMD    | L1P memory protection lock key command register   |
| 0x0184 A514               | L1PMPLKSTAT   | L1P memory protection lock key status register  |
| 0x0184 A518 - 0x0184 A5FF | -             | Reserved  |
| 0x0184 A600 - 0x0184 A63F | -             | Reserved. <sup>(1)</sup>  |
| 0x0184 A640               | L1PMPPA16     | L1P memory protection page attribute register 16<br>(controls memory address 0x00E0 0000 - 0x00E0 07FF) |
| 0x0184 A644               | L1PMPPA17     | L1P memory protection page attribute register 17<br>(controls memory address 0x00E0 0800 - 0x00E0 0FFF) |
| 0x0184 A648               | L1PMPPA18     | L1P memory protection page attribute register 18<br>(controls memory address 0x00E0 1000 - 0x00E0 17FF) |
| 0x0184 A64C               | L1PMPPA19     | L1P memory protection page attribute register 19<br>(controls memory address 0x00E0 1800 - 0x00E0 1FFF) |
| 0x0184 A650               | L1PMPPA20     | L1P memory protection page attribute register 20<br>(controls memory address 0x00E0 2000 - 0x00E0 27FF) |
| 0x0184 A654               | L1PMPPA21     | L1P memory protection page attribute register 21<br>(controls memory address 0x00E0 2800 - 0x00E0 2FFF) |
| 0x0184 A658               | L1PMPPA22     | L1P memory protection page attribute register 22<br>(controls memory address 0x00E0 3000 - 0x00E0 37FF) |
| 0x0184 A65C               | L1PMPPA23     | L1P memory protection page attribute register 23<br>(controls memory address 0x00E0 3800 - 0x00E0 3FFF) |
| 0x0184 A660               | L1PMPPA24     | L1P memory protection page attribute register 24<br>(controls memory address 0x00E0 4000 - 0x00E0 47FF) |
| 0x0184 A664               | L1PMPPA25     | L1P memory protection page attribute register 25<br>(controls memory address 0x00E0 4800 - 0x00E0 4FFF) |
| 0x0184 A668               | L1PMPPA26     | L1P memory protection page attribute register 26<br>(controls memory address 0x00E0 5000 - 0x00E0 57FF) |
| 0x0184 A66C               | L1PMPPA27     | L1P memory protection page attribute register 27<br>(controls memory address 0x00E0 5800 - 0x00E0 5FFF) |
| 0x0184 A670               | L1PMPPA28     | L1P memory protection page attribute register 28<br>(controls memory address 0x00E0 6000 - 0x00E0 67FF) |
| 0x0184 A674               | L1PMPPA29     | L1P memory protection page attribute register 29<br>(controls memory address 0x00E0 6800 - 0x00E0 6FFF) |

(1) These addresses correspond to the L1P memory protection page attribute registers 0-15 (L1PMPPA0-L1PMPPA15) of the C674x megamodule. These registers are not supported for this device.

**Table 3-3. C674x L1/L2 Memory Protection Registers (continued)**

| HEX ADDRESS RANGE         | REGISTER NAME | DESCRIPTION   |
|---------------------------|---------------|---|
| 0x0184 A678               | L1PMPPA30     | L1P memory protection page attribute register 30<br>(controls memory address 0x00E0 7000 - 0x00E0 77FF) |
| 0x0184 A67C               | L1PMPPA31     | L1P memory protection page attribute register 31<br>(controls memory address 0x00E0 7800 - 0x00E0 7FFF) |
| 0x0184 A67F – 0x0184 ABFF | -             | Reserved  |
| 0x0184 AC00               | L1DMPFAR      | L1D memory protection fault address register  |
| 0x0184 AC04               | L1DMPFSR      | L1D memory protection fault status register   |
| 0x0184 AC08               | L1DMPFCR      | L1D memory protection fault command register  |
| 0x0184 AC0C - 0x0184 ACFF | -             | Reserved  |
| 0x0184 AD00               | L1DMPLK0      | L1D memory protection lock key bits [31:0]  |
| 0x0184 AD04               | L1DMPLK1      | L1D memory protection lock key bits [63:32]   |
| 0x0184 AD08               | L1DMPLK2      | L1D memory protection lock key bits [95:64]   |
| 0x0184 AD0C               | L1DMPLK3      | L1D memory protection lock key bits [127:96]  |
| 0x0184 AD10               | L1DMPLKCMD    | L1D memory protection lock key command register   |
| 0x0184 AD14               | L1DMPLKSTAT   | L1D memory protection lock key status register  |
| 0x0184 AD18 - 0x0184 ADFF | -             | Reserved  |
| 0x0184 AE00 - 0x0184 AE3F | -             | Reserved. <sup>(2)</sup>  |
| 0x0184 AE40               | L1DMPPA16     | L1D memory protection page attribute register 16<br>(controls memory address 0x00F0 0000 - 0x00F0 07FF) |
| 0x0184 AE44               | L1DMPPA17     | L1D memory protection page attribute register 17<br>(controls memory address 0x00F0 0800 - 0x00F0 0FFF) |
| 0x0184 AE48               | L1DMPPA18     | L1D memory protection page attribute register 18<br>(controls memory address 0x00F0 1000 - 0x00F0 17FF) |
| 0x0184 AE4C               | L1DMPPA19     | L1D memory protection page attribute register 19<br>(controls memory address 0x00F0 1800 - 0x00F0 1FFF) |
| 0x0184 AE50               | L1DMPPA20     | L1D memory protection page attribute register 20<br>(controls memory address 0x00F0 2000 - 0x00F0 27FF) |
| 0x0184 AE54               | L1DMPPA21     | L1D memory protection page attribute register 21<br>(controls memory address 0x00F0 2800 - 0x00F0 2FFF) |
| 0x0184 AE58               | L1DMPPA22     | L1D memory protection page attribute register 22<br>(controls memory address 0x00F0 3000 - 0x00F0 37FF) |
| 0x0184 AE5C               | L1DMPPA23     | L1D memory protection page attribute register 23<br>(controls memory address 0x00F0 3800 - 0x00F0 3FFF) |
| 0x0184 AE60               | L1DMPPA24     | L1D memory protection page attribute register 24<br>(controls memory address 0x00F0 4000 - 0x00F0 47FF) |
| 0x0184 AE64               | L1DMPPA25     | L1D memory protection page attribute register 25<br>(controls memory address 0x00F0 4800 - 0x00F0 4FFF) |
| 0x0184 AE68               | L1DMPPA26     | L1D memory protection page attribute register 26<br>(controls memory address 0x00F0 5000 - 0x00F0 57FF) |
| 0x0184 AE6C               | L1DMPPA27     | L1D memory protection page attribute register 27<br>(controls memory address 0x00F0 5800 - 0x00F0 5FFF) |
| 0x0184 AE70               | L1DMPPA28     | L1D memory protection page attribute register 28<br>(controls memory address 0x00F0 6000 - 0x00F0 67FF) |
| 0x0184 AE74               | L1DMPPA29     | L1D memory protection page attribute register 29<br>(controls memory address 0x00F0 6800 - 0x00F0 6FFF) |
| 0x0184 AE78               | L1DMPPA30     | L1D memory protection page attribute register 30<br>(controls memory address 0x00F0 7000 - 0x00F0 77FF) |
| 0x0184 AE7C               | L1DMPPA31     | L1D memory protection page attribute register 31<br>(controls memory address 0x00F0 7800 - 0x00F0 7FFF) |
| 0x0184 AE80 – 0x0185 FFFF | -             | Reserved  |

(2) These addresses correspond to the L1D memory protection page attribute registers 0-15 (L1DMPPA0-L1DMPPA15) of the C674x megamodule. These registers are not supported for this device.

### 3.4 Memory Map Summary

**Note:** Read/Write accesses to illegal or reserved addresses in the memory map may cause undefined behavior.

**Table 3-4. C6743 Top Level Memory Map**

| START ADDRESS | END ADDRESS | SIZE  | DSP MEM MAP               | EDMA MEM MAP | PRUSS MEM MAP             | MASTER PERIPHERAL MEM MAP |
|---------------|-------------|-------|---------------------------|--------------|---------------------------|---------------------------|
| 0x0000 0000   | 0x0000 0FFF | 4K    |                           |              | PRUSS Local Address Space |                           |
| 0x0000 1000   | 0x006F FFFF |       |                           |              |                           |                           |
| 0x0070 0000   | 0x007F FFFF | 1024K | DSP L2 ROM <sup>(1)</sup> |              |                           |                           |
| 0x0080 0000   | 0x0081 FFFF |       |                           |              |                           |                           |
| 0x0082 0000   | 0x0083 FFFF | 128K  | DSP L2 RAM                |              |                           |                           |
| 0x0084 0000   | 0x00DF FFFF |       |                           |              |                           |                           |
| 0x00E0 0000   | 0x00E0 7FFF | 32K   | DSP L1P RAM               |              |                           |                           |
| 0x00E0 8000   | 0x00EF FFFF |       |                           |              |                           |                           |
| 0x00F0 0000   | 0x00F0 7FFF | 32K   | DSP L1D RAM               |              |                           |                           |
| 0x00F0 8000   | 0x017F FFFF |       |                           |              |                           |                           |
| 0x0180 0000   | 0x0180 FFFF | 64K   | DSP Interrupt Controller  |              |                           |                           |
| 0x0181 0000   | 0x0181 0FFF | 4K    | DSP Powerdown Controller  |              |                           |                           |
| 0x0181 1000   | 0x0181 1FFF | 4K    | DSP Security ID           |              |                           |                           |
| 0x0181 2000   | 0x0181 2FFF | 4K    | DSP Revision ID           |              |                           |                           |
| 0x0181 3000   | 0x0181 FFFF |       |                           |              |                           |                           |
| 0x0182 0000   | 0x0182 FFFF | 64K   | DSP EMC                   |              |                           |                           |
| 0x0183 0000   | 0x0183 FFFF | 64K   | DSP Internal Reserved     |              |                           |                           |
| 0x0184 0000   | 0x0184 FFFF | 64K   | DSP Memory System         |              |                           |                           |
| 0x0185 0000   | 0x01BF FFFF |       |                           |              |                           |                           |
| 0x01C0 0000   | 0x01C0 7FFF | 32K   |                           |              | EDMA3 CC                  |                           |
| 0x01C0 8000   | 0x01C0 83FF | 1024  |                           |              | EDMA3 TC0                 |                           |
| 0x01C0 8400   | 0x01C0 87FF | 1024  |                           |              | EDMA3 TC1                 |                           |
| 0x01C0 8800   | 0x01C0 FFFF |       |                           |              |                           |                           |
| 0x01C1 0000   | 0x01C1 0FFF | 4K    |                           |              | PSC 0                     |                           |
| 0x01C1 1000   | 0x01C1 1FFF | 4K    |                           |              | PLL Controller            |                           |
| 0x01C1 2000   | 0x01C1 3FFF |       |                           |              |                           |                           |
| 0x01C1 4000   | 0x01C1 4FFF | 4K    |                           |              | BootConfig                |                           |
| 0x01C1 5000   | 0x01C1 FFFF |       |                           |              |                           |                           |
| 0x01C2 0000   | 0x01C2 0FFF | 4K    |                           |              | Timer64P 0                |                           |
| 0x01C2 1000   | 0x01C2 1FFF | 4K    |                           |              | Timer64P 1                |                           |
| 0x01C2 2000   | 0x01C2 2FFF | 4K    |                           |              | I2C 0                     |                           |
| 0x01C2 3000   | 0x01C2 FFFF |       |                           |              |                           |                           |
| 0x01C4 0000   | 0x01C4 0FFF | 4K    |                           |              | MMC/SD 0                  |                           |
| 0x01C4 1000   | 0x01C4 1FFF | 4K    |                           |              | SPI 0                     |                           |
| 0x01C4 2000   | 0x01C4 2FFF | 4K    |                           |              | UART 0                    |                           |
| 0x01C4 3000   | 0x01CF FFFF |       |                           |              |                           |                           |
| 0x01D0 0000   | 0x01D0 0FFF | 4K    |                           |              | McASP 0 Control           |                           |
| 0x01D0 1000   | 0x01D0 1FFF | 4K    |                           |              | McASP 0 AFIFO Ctrl        |                           |

(1) The DSP L2 ROM is used for boot purposes and cannot be programmed with application code.

**Table 3-4. C6743 Top Level Memory Map (continued)**

| START ADDRESS | END ADDRESS | SIZE  | DSP MEM MAP | EDMA MEM MAP | PRUSS MEM MAP                 | MASTER PERIPHERAL MEM MAP |
|---------------|-------------|-------|-------------|--------------|-------------------------------|---------------------------|
| 0x01D0 2000   | 0x01D0 2FFF | 4K    |             |              | McASP 0 Data                  |                           |
| 0x01D0 3000   | 0x01D0 3FFF |       |             |              |                               |                           |
| 0x01D0 4000   | 0x01D0 4FFF | 4K    |             |              | McASP 1 Control               |                           |
| 0x01D0 5000   | 0x01D0 5FFF | 4K    |             |              | McASP 1 AFIFO Ctrl            |                           |
| 0x01D0 6000   | 0x01D0 6FFF | 4K    |             |              | McASP 1 Data                  |                           |
| 0x01D0 7000   | 0x01D0 CFFF |       |             |              |                               |                           |
| 0x01D0 D000   | 0x01D0 DFFF | 4K    |             |              | UART2                         |                           |
| 0x01D0 E000   | 0x01E1 3FFF |       |             |              |                               |                           |
| 0x01E1 4000   | 0x01E1 4FFF | 4K    |             |              | Memory Protection Unit 1      |                           |
| 0x01E1 5000   | 0x01E1 5FFF | 4K    |             |              | Memory Protection Unit 2      |                           |
| 0x01E1 6000   | 0x01E1 FFFF |       |             |              |                               |                           |
| 0x01E2 0000   | 0x01E2 1FFF | 8K    |             |              | EMAC Control Module RAM       |                           |
| 0x01E2 2000   | 0x01E2 2FFF | 4K    |             |              | EMAC Control Module Registers |                           |
| 0x01E2 3000   | 0x01E2 3FFF | 4K    |             |              | EMAC Control Registers        |                           |
| 0x01E2 4000   | 0x01E2 4FFF | 4K    |             |              | EMAC MDIO port                |                           |
| 0x01E2 5000   | 0x01E2 5FFF |       |             |              |                               |                           |
| 0x01E2 6000   | 0x01E2 6FFF | 4K    |             |              | GPIO                          |                           |
| 0x01E2 7000   | 0x01E2 7FFF | 4K    |             |              | PSC 1                         |                           |
| 0x01E2 8000   | 0x01E2 8FFF | 4K    |             |              | 12C 1                         |                           |
| 0x01E2 9000   | 0x01EF FFFF |       |             |              |                               |                           |
| 0x01F0 0000   | 0x01F0 0FFF | 4K    |             |              | eHRPWM 0                      |                           |
| 0x01F0 1000   | 0x01F0 1FFF | 4K    |             |              | HRPWM 0                       |                           |
| 0x01F0 2000   | 0x01F0 2FFF | 4K    |             |              | eHRPWM 1                      |                           |
| 0x01F0 3000   | 0x01F0 3FFF | 4K    |             |              | HRPWM 1                       |                           |
| 0x01F0 4000   | 0x01F0 4FFF | 4K    |             |              | eHRPWM 2                      |                           |
| 0x01F0 5000   | 0x01F0 5FFF | 4K    |             |              | HRPWM 2                       |                           |
| 0x01F0 6000   | 0x01F0 6FFF | 4K    |             |              | ECAP 0                        |                           |
| 0x01F0 7000   | 0x01F0 7FFF | 4K    |             |              | ECAP 1                        |                           |
| 0x01F0 8000   | 0x01F0 8FFF | 4K    |             |              | ECAP 2                        |                           |
| 0x01F0 9000   | 0x01F0 9FFF | 4K    |             |              | EQEP 0                        |                           |
| 0x01F0 A000   | 0x01F0 AFFF | 4K    |             |              | EQEP 1                        |                           |
| 0x01F0 B000   | 0x116F FFFF |       |             |              |                               |                           |
| 0x1170 0000   | 0x117F FFFF | 1024K |             |              | DSP L2 ROM <sup>(1)</sup>     |                           |
| 0x1180 0000   | 0x1181 FFFF |       |             |              |                               |                           |
| 0x1182 0000   | 0x1183 FFFF | 128K  |             |              | DSP L2 RAM                    |                           |
| 0x1184 0000   | 0x11DF FFFF |       |             |              |                               |                           |
| 0x11E0 0000   | 0x11E0 7FFF | 32K   |             |              | DSP L1P RAM                   |                           |
| 0x11E0 8000   | 0x11EF FFFF |       |             |              |                               |                           |
| 0x11F0 0000   | 0x11F0 7FFF | 32K   |             |              | DSP L1D RAM                   |                           |
| 0x11F0 8000   | 0x5FFF FFFF |       |             |              |                               |                           |
| 0x6000 0000   | 0x61FF FFFF | 32M   |             |              | EMIFA async data (CS2)        |                           |
| 0x6200 0000   | 0x63FF FFFF | 32M   |             |              | EMIFA async data (CS3)        |                           |
| 0x6400 0000   | 0x67FF FFFF |       |             |              |                               |                           |
| 0x6800 0000   | 0x6800 7FFF | 32K   |             |              | EMIFA Control Regs            |                           |
| 0x6800 8000   | 0xAFFF FFFF |       |             |              |                               |                           |
| 0xB000 0000   | 0xB000 7FFF | 32K   |             |              | EMIFB Control Regs            |                           |
| 0xB000 8000   | 0xBFFF FFFF |       |             |              |                               |                           |

Table 3-4. C6743 Top Level Memory Map (continued)

| START ADDRESS | END ADDRESS | SIZE | DSP MEM MAP      | EDMA MEM MAP | PRUSS MEM MAP | MASTER PERIPHERAL MEM MAP |
|---------------|-------------|------|------------------|--------------|---------------|---------------------------|
| 0xC000 0000   | 0xC7FF FFFF | 128M | EMIFB SDRAM Data |              |               |                           |
| 0xC800 0000   | 0xFFFF FFFF |      |                  |              |               |                           |

### 3.5 Pin Assignments

Extensive use of pin multiplexing is used to accommodate the largest number of peripheral functions in the smallest possible package. Pin multiplexing is controlled using a combination of hardware configuration at device reset and software programmable register settings.

#### 3.5.1 Pin Map (Bottom View)

Figure 3-3 and Figure 3-4 show the pin assignments for ZKB package and PTP packages, respectively.

|   | 1                             | 2                                      | 3  | 4   | 5  | 6  | 7                                  | 8                                  | 9                                  | 10                  | 11                    | 12                    | 13   | 14                            | 15   | 16                    |   |
|---|-------------------------------|--|--|---|--|--|------------------------------------|------------------------------------|------------------------------------|---------------------|-----------------------|-----------------------|--|-------------------------------|--|-----------------------|---|
| T | Vss                           | Vss                                    | AXR1[0]/<br>GP4[0]   | NC  | SPI0_CLK/<br>EQEP1/<br>GP5[2]/<br>BOOT[2]                | EQEP1S/<br>GP5[7]/<br>BOOT[7]                  | EMA_CS[3]/<br>GP2[6]               | NC                                 | EMA_A[0]/<br>GP1[0]                | EMA_A[4]/<br>GP1[4] | EMA_A[8]/<br>GP1[8]   | NC                    | EMA_D[0]/<br>MMCS_DAT[0]/<br>GP0[0]/<br>BOOT[12] | GP0[9]                        | Vss  | Vss                   | T |
| R | DVdd                          | AXR1[1]/<br>GP4[1]                     | UART0_RXD/<br>I2C0_SDA<br>TM64P0_IN[2]/<br>GP5[9]/BOOT[8]  | UART2_RXD/<br>GP5[12]                                       | SPI0_ENA/<br>UART0_CTS/<br>EQEP0A/<br>GP5[3]/<br>BOOT[3] | SPI0_SOM[0]/<br>EQEP0/<br>GP5[0]/<br>BOOT[0]   | EMA_OE/<br>AXR0[13]/<br>GP2[7]     | EMA_BA[0]/<br>GP1[14]              | EMA_A[1]/<br>MMCS_D_CLK/<br>GP1[1] | EMA_A[5]/<br>GP1[5] | EMA_A[9]/<br>GP1[9]   | NC                    | EMA_D[2]/<br>MMCS_DAT[2]/<br>GP0[2]              | GP0[10]                       | EMA_D[1]/<br>MMCS_DAT[1]/<br>GP0[1]              | DVdd                  | R |
| P | AXR1[3]/<br>EQEP1A/<br>GP4[3] | AXR1[2]/<br>GP4[2]                     | UART0_TXD/<br>I2C0_SCL<br>TM64P0_OUT[2]/<br>GP5[9]/BOOT[8] | UART2_TXD/<br>GP5[13]                                       | I2C1_SCL/<br>GP5[5]/<br>BOOT[5]                          | SPI0_SIMO[0]/<br>EQEP0S/<br>GP5[1]/<br>BOOT[1] | EMA_CS[2]/<br>GP2[5]/<br>BOOT[15]  | EMA_BA[1]/<br>GP1[13]              | EMA_A[2]/<br>MMCS_CMD/<br>GP1[2]   | EMA_A[6]/<br>GP1[6] | EMA_A[11]/<br>GP1[11] | NC                    | EMA_D[4]/<br>MMCS_DAT[4]/<br>GP0[4]              | GP0[12]                       | EMA_D[3]/<br>MMCS_DAT[3]/<br>GP0[3]              | GP0[11]               | P |
| N | AXR1[5]/<br>EPWM2B/<br>GP4[5] | AXR1[4]/<br>EQEP1B/<br>GP4[4]          | NC   | SPI0_SCS[0]/<br>UART0_RTS/<br>EQEP0B/<br>GP5[4]/<br>BOOT[4] | I2C1_SDA/<br>GP5[6]/<br>BOOT[6]                          | EMA_WAIT[0]/<br>GP2[10]                        | NC                                 | EMA_A[10]/<br>GP1[10]              | EMA_A[3]/<br>GP1[3]                | EMA_A[7]/<br>GP1[7] | EMA_A[12]/<br>GP1[12] | GP0[8]                | EMA_D[6]/<br>MMCS_DAT[6]/<br>GP0[6]              | GP0[14]                       | EMA_D[5]/<br>MMCS_DAT[5]/<br>GP0[5]              | GP0[13]               | N |
| M | NC                            | AXR1[8]/<br>EPWM1A/<br>GP4[8]          | AXR1[7]/<br>EPWM1B/<br>GP4[7]                              | AXR1[6]/<br>EPWM2A/<br>GP4[6]                               | DVdd   | Vss  | Vss                                | DVdd                               | DVdd                               | Vss                 | Vss                   | DVdd                  | EMA_WE/<br>AXR0[12]/<br>GP2[3]/<br>BOOT[14]      | NC                            | EMA_D[7]/<br>MMCS_DAT[7]/<br>GP0[7]/<br>BOOT[13] | GP0[15]               | M |
| L | AHCLKR1/<br>GP4[11]           | ACLKR1/<br>ECAP2/<br>APWM2/<br>GP4[12] | AFSR1/<br>GP4[13]  | NC  | DVdd   | CVdd   | Vss                                | Vss                                | Vss                                | Vss                 | DVdd                  | DVdd                  | EMB_CAS  | NC                            | NC   | NC                    | L |
| K | GP7[14]                       | AHCLKX1/<br>EPWM0B/<br>GP3[14]         | ACLKX1/<br>EPWM0A/<br>GP3[15]                              | AFSX1/<br>EPWMSYNC1/<br>EPWMSYNC0/<br>GP4[10]               | DVdd   | CVdd   | CVdd                               | Vss                                | Vss                                | CVdd                | CVdd                  | DVdd                  | NC   | EMB_WE_<br>DQM[0]/<br>GP5[15] | EMB_WE   | NC                    | K |
| J | TMS                           | TDI                                    | TDO  | TRST  | EMU0/<br>GP7[15]   | CVdd   | CVdd                               | Vss                                | Vss                                | CVdd                | CVdd                  | CVdd                  | EMB_D[5]/<br>GP6[5]                              | NC                            | EMB_D[6]/<br>GP6[6]                              | EMB_D[7]/<br>GP6[7]   | J |
| H | RSV4                          | NC                                     | TCK  | NC  | NC   | RVdd   | CVdd                               | Vss                                | Vss                                | CVdd                | CVdd                  | RVdd                  | EMB_D[3]/<br>GP6[3]                              | NC                            | NC   | EMB_D[4]/<br>GP6[4]   | H |
| G | CVdd                          | Vss                                    | RESET  | Vss   | DVdd   | CVdd   | CVdd                               | Vss                                | Vss                                | CVdd                | CVdd                  | DVdd                  | EMB_D[1]/<br>GP6[1]                              | NC                            | NC   | EMB_D[2]/<br>GP6[2]   | G |
| F | OSCOU                         | OSCIN                                  | NC   | Vss   | DVdd   | CVdd   | RSV1                               | Vss                                | Vss                                | Vss                 | DVdd                  | DVdd                  | EMB_D[15]/<br>GP6[15]                            | NC                            | NC   | EMB_D[0]/<br>GP6[0]   | F |
| E | PLL0_VSSA                     | OSCVSS                                 | NC   | NC  | DVdd   | Vss  | Vss                                | DVdd                               | DVdd                               | Vss                 | Vss                   | DVdd                  | EMB_D[13]/<br>GP6[13]                            | NC                            | NC   | EMB_D[14]/<br>GP6[14] | E |
| D | PLL0_VDDA                     | NC                                     | NC   | AMUTE1/<br>EPWMTZ/<br>GP4[14]                               | AFSX0/<br>GP2[13]/<br>BOOT[10]                           | AXR0[10]/<br>GP3[10]                           | AXR0[6]/<br>RMIL_RXER/<br>GP3[6]   | AXR0[2]/<br>RMIL_TXEN/<br>GP3[2]   | EMB_CS[0]                          | EMB_A[0]/<br>GP7[2] | EMB_A[4]/<br>GP7[6]   | EMB_A[8]/<br>GP7[10]  | EMB_D[9]/<br>GP6[9]                              | EMB_D[10]/<br>GP6[10]         | EMB_D[11]/<br>GP6[11]                            | EMB_D[12]/<br>GP6[12] | D |
| C | NC                            | NC                                     | NC   | AFSR0/<br>GP3[12]   | ACLKX0/<br>ECAP0/<br>APWM0/<br>GP2[12]                   | AXR0[9]/<br>GP3[9]                             | AXR0[5]/<br>RMIL_RXD[1]/<br>GP3[5] | AXR0[1]/<br>RMIL_TXD[0]/<br>GP3[1] | EMB_BA[0]/<br>GP7[1]               | EMB_A[1]/<br>GP7[3] | EMB_A[5]/<br>GP7[7]   | EMB_A[9]/<br>GP7[11]  | EMB_SDCKE  | EMB_CLK                       | EMB_WE_<br>DQM[1]/<br>GP5[14]                    | EMB_D[8]/<br>GP6[8]   | C |
| B | RSV2                          | Vss                                    | Vss  | ACLKR0/<br>ECAP1/<br>APWM1/<br>GP2[15]                      | AHCLKX0/<br>AHCLKX2/<br>GP2[11]                          | AXR0[8]/<br>MDIO_D/<br>GP3[8]                  | AXR0[4]/<br>RMIL_RXD[0]/<br>GP3[4] | AXR0[0]/<br>RMIL_TXD[0]/<br>GP3[0] | EMB_BA[1]/<br>GP7[0]               | EMB_A[2]/<br>GP7[4] | EMB_A[6]/<br>GP7[8]   | EMB_A[11]/<br>GP7[13] | NC   | NC                            | EMB_A[12]/<br>GP3[13]                            | DVdd                  | B |
| A | Vss                           | Vss                                    | Vss  | AHCLKR0/<br>RMIL_MHZ_<br>50_CLK/<br>GP2[14]/<br>BOOT[11]    | AXR0[11]/<br>GP3[11]                                     | AXR0[7]/<br>MDIO_CLK/<br>GP3[7]                | AXR0[3]/<br>RMIL_CRD_DV/<br>GP3[3] | EMB_RAS                            | EMB_A[10]/<br>GP7[12]              | EMB_A[3]/<br>GP7[5] | EMB_A[7]/<br>GP7[9]   | NC                    | NC   | NC                            | Vss  | Vss                   | A |

Note: NC = No Connect

Figure 3-3. Pin Map (ZKB)

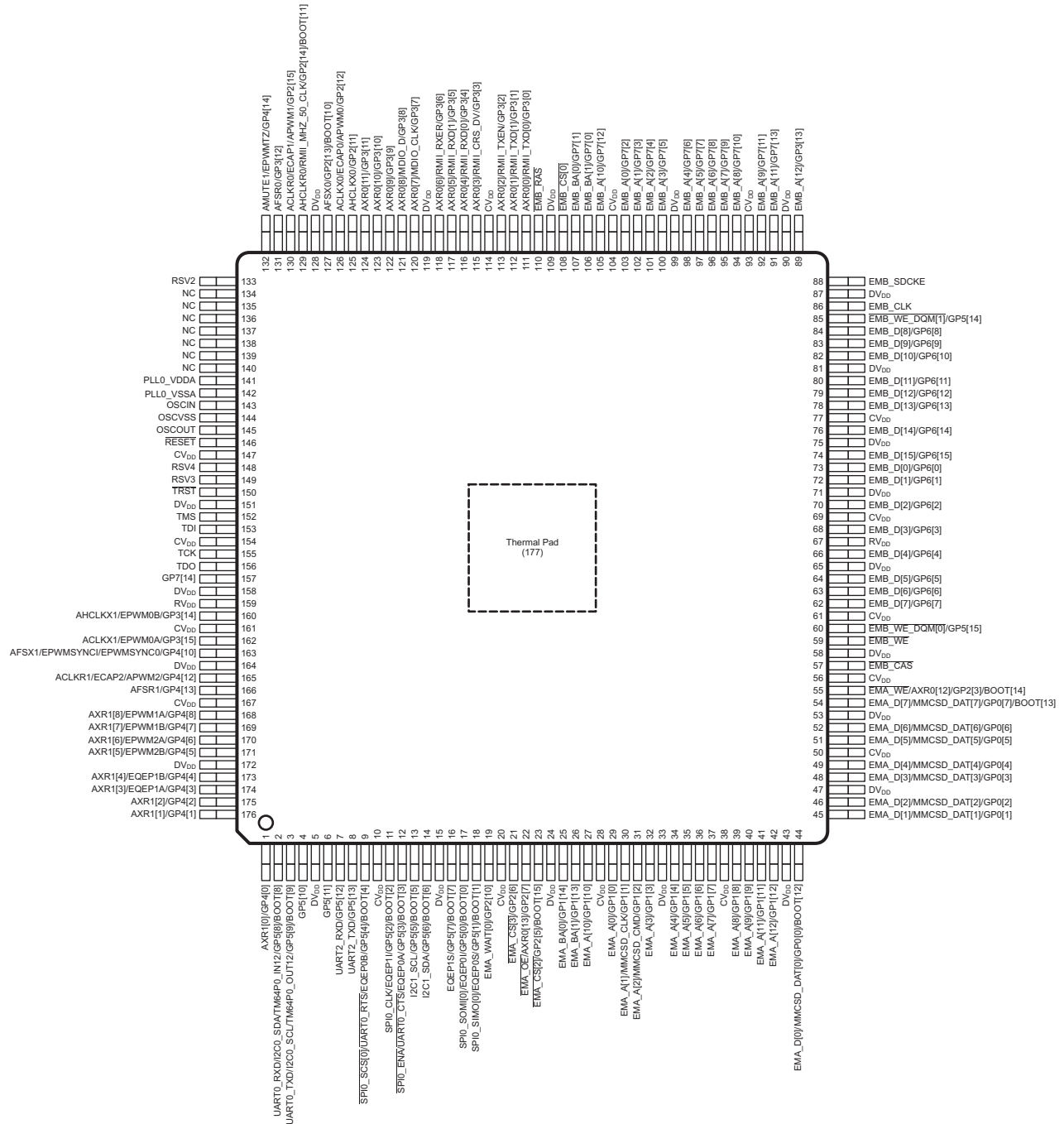


Figure 3-4. Pin Map (PTP)

### 3.6 Terminal Functions

Table 3-5 to Section 3.6.18 identify the external signal names, the associated pin/ball numbers along with the mechanical package designator, the pin type (I, O, IO, OZ, or PWR), whether the pin/ball has any internal pullup/pulldown resistors, whether the pin/ball is configurable as an IO in GPIO mode, and a functional pin description.

#### 3.6.1 Device Reset and JTAG

**Table 3-5. Reset and JTAG Terminal Functions**

| SIGNAL NAME               | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED | DESCRIPTION           |
|---------------------------|--------|-----|---------------------|---------------------|-------|-----------------------|
|                           | PTP    | ZKB |                     |                     |       |                       |
| <b>RESET</b>              |        |     |                     |                     |       |                       |
| $\overline{\text{RESET}}$ | 146    | G3  | I                   |                     |       | Device reset input    |
| <b>JTAG</b>               |        |     |                     |                     |       |                       |
| TMS                       | 152    | J1  | I                   | IPU                 |       | JTAG test mode select |
| TDI                       | 153    | J2  | I                   | IPU                 |       | JTAG test data input  |
| TDO                       | 156    | J3  | O                   | IPU                 |       | JTAG test data output |
| TCK                       | 155    | H3  | I                   | IPU                 |       | JTAG test clock       |
| $\overline{\text{TRST}}$  | 150    | J4  | I                   | IPD                 |       | JTAG test reset       |
| EMU0/GP7[15]              | -      | J5  | I/O                 | IPD                 | GPIO  | Emulation Signal      |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

#### 3.6.2 High-Frequency Oscillator and PLL

**Table 3-6. High-Frequency Oscillator and PLL Terminal Functions**

| SIGNAL NAME             | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | DESCRIPTION                                 |
|-------------------------|--------|-----|---------------------|---------------------|---|
|                         | PTP    | ZKB |                     |                     |   |
| <b>1.2-V OSCILLATOR</b> |        |     |                     |                     |   |
| OSCIN                   | 143    | F2  | I                   |                     | Oscillator input                            |
| OSCOU                   | 145    | F1  | O                   |                     | Oscillator output                           |
| OSCVSS                  | 144    | E2  | GND                 |                     | Oscillator ground (for filter only)         |
| <b>1.2-V PLL</b>        |        |     |                     |                     |   |
| PLL0_VDDA               | 141    | D1  | PWR                 |                     | PLL analog $V_{DD}$ (1.2-V filtered supply) |
| PLL0_VSSA               | 142    | E1  | GND                 |                     | PLL analog $V_{SS}$ (for filter)            |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.3 External Memory Interface A (ASYNC)

Table 3-7. External Memory Interface A (EMIFA) Terminal Functions

| SIGNAL NAME                          | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED              | DESCRIPTION                |              |
|--------------------------------------|--------|-----|---------------------|---------------------|--------------------|----------------------------|--------------|
|                                      | PTP    | ZKB |                     |                     |                    |                            |              |
| EMA_D[7]/MMCS_DAT[7]/GP0[7]/BOOT[13] | 54     | M15 | I/O                 | IPU                 | MMC/SD, GPIO, BOOT | EMIFA data bus             |              |
| EMA_D[6]/MMCS_DAT[6]/GP0[6]          | 52     | N13 | I/O                 | IPU                 | MMC/SD, GPIO       |                            |              |
| EMA_D[5]/MMCS_DAT[5]/GP0[5]          | 51     | N15 | I/O                 | IPU                 |                    |                            |              |
| EMA_D[4]/MMCS_DAT[4]/GP0[4]          | 49     | P13 | I/O                 | IPU                 |                    |                            |              |
| EMA_D[3]/MMCS_DAT[3]/GP0[3]          | 48     | P15 | I/O                 | IPU                 |                    |                            |              |
| EMA_D[2]/MMCS_DAT[2]/GP0[2]          | 46     | R13 | I/O                 | IPU                 |                    |                            |              |
| EMA_D[1]/MMCS_DAT[1]/GP0[1]          | 45     | R15 | I/O                 | IPU                 |                    |                            |              |
| EMA_D[0]/MMCS_DAT[0]/GP0[0]/BOOT[12] | 44     | T13 | I/O                 | IPU                 | MMC/SD, GPIO, BOOT | EMIFA address bus          |              |
| EMA_A[12]/GP1[12]                    | 42     | N11 | O                   | IPU                 | GPIO               |                            |              |
| EMA_A[11]/GP1[11]                    | 41     | P11 | O                   | IPU                 |                    |                            |              |
| EMA_A[10]/GP1[10]                    | 27     | N8  | O                   | IPU                 |                    |                            |              |
| EMA_A[9]/GP1[9]                      | 40     | R11 | O                   | IPU                 |                    |                            |              |
| EMA_A[8]/GP1[8]                      | 39     | T11 | O                   | IPU                 |                    |                            |              |
| EMA_A[7]/GP1[7]                      | 37     | N10 | O                   | IPD                 |                    |                            |              |
| EMA_A[6]/GP1[6]                      | 36     | P10 | O                   | IPD                 |                    |                            |              |
| EMA_A[5]/GP1[5]                      | 35     | R10 | O                   | IPD                 |                    |                            |              |
| EMA_A[4]/GP1[4]                      | 34     | T10 | O                   | IPD                 |                    |                            |              |
| EMA_A[3]/GP1[3]                      | 32     | N9  | O                   | IPD                 |                    |                            |              |
| EMA_A[2]/MMCS_D_CMD/GP1[2]           | 31     | P9  | O                   | IPU                 |                    |                            | MMCS_D, GPIO |
| EMA_A[1]/MMCS_D_CLK/GP1[1]           | 30     | R9  | O                   | IPU                 |                    |                            |              |
| EMA_A[0]/GP1[0]                      | 29     | T9  | O                   | IPD                 |                    |                            | GPIO         |
| EMA_BA[1]/GP1[13]                    | 26     | P8  | O                   | IPU                 |                    |                            |              |
| EMA_BA[0]/GP1[14]                    | 25     | R8  | O                   | IPU                 |                    |                            |              |
| EMA_CS[3]/GP2[6]                     | 21     | T7  | O                   | IPU                 | GPIO               | EMIFA Async Chip Select    |              |
| EMA_CS[2]/GP2[5]/BOOT[15]            | 23     | P7  | O                   | IPU                 | GPIO, BOOT         |                            |              |
| EMA_WE /AXR0[12]/GP2[3]/BOOT[14]     | 55     | M13 | O                   | IPU                 | MCASP0, GPIO, BOOT | EMIFA write enable         |              |
| EMA_OE /AXR0[13]/GP2[7]              | 22     | R7  | O                   | IPU                 | McASP0, GPIO       | EMIFA output enable        |              |
| EMA_WAIT[0]/GP2[10]                  | 19     | N6  | I                   | IPU                 | GPIO               | EMIFA wait input/interrupt |              |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.4 External Memory Interface B (SDRAM only)

**Table 3-8. External Memory Interface B (EMIFB) Terminal Functions**

| SIGNAL NAME       | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED | DESCRIPTION                            |
|-------------------|--------|-----|---------------------|---------------------|-------|--|
|                   | PTP    | ZKB |                     |                     |       |  |
| EMB_D[15]/GP6[15] | 74     | F13 | I/O                 | IPD                 | GPIO  | EMIFB SDRAM data bus                   |
| EMB_D[14]/GP6[14] | 76     | E16 | I/O                 | IPD                 |       |  |
| EMB_D[13]/GP6[13] | 78     | E13 | I/O                 | IPD                 |       |  |
| EMB_D[12]/GP6[12] | 79     | D16 | I/O                 | IPD                 |       |  |
| EMB_D[11]/GP6[11] | 80     | D15 | I/O                 | IPD                 |       |  |
| EMB_D[10]/GP6[10] | 82     | D14 | I/O                 | IPD                 |       |  |
| EMB_D[9]/GP6[9]   | 83     | D13 | I/O                 | IPD                 |       |  |
| EMB_D[8]/GP6[8]   | 84     | C16 | I/O                 | IPD                 |       |  |
| EMB_D[7]/GP6[7]   | 62     | J16 | I/O                 | IPD                 |       |  |
| EMB_D[6]/GP6[6]   | 63     | J15 | I/O                 | IPD                 |       |  |
| EMB_D[5]/GP6[5]   | 64     | J13 | I/O                 | IPD                 |       |  |
| EMB_D[4]/GP6[4]   | 66     | H16 | I/O                 | IPD                 |       |  |
| EMB_D[3]/GP6[3]   | 68     | H13 | I/O                 | IPD                 |       |  |
| EMB_D[2]/GP6[2]   | 70     | G16 | I/O                 | IPD                 |       |  |
| EMB_D[1]/GP6[1]   | 72     | G13 | I/O                 | IPD                 |       |  |
| EMB_D[0]/GP6[0]   | 73     | F16 | I/O                 | IPD                 |       |  |
| EMB_A[12]/GP3[13] | 89     | B15 | O                   | IPD                 | GPIO  | EMIFB SDRAM row/column address bus     |
| EMB_A[11]/GP7[13] | 91     | B12 | O                   | IPD                 |       |  |
| EMB_A[10]/GP7[12] | 105    | A9  | O                   | IPD                 |       |  |
| EMB_A[9]/GP7[11]  | 92     | C12 | O                   | IPD                 |       |  |
| EMB_A[8]/GP7[10]  | 94     | D12 | O                   | IPD                 |       |  |
| EMB_A[7]/GP7[9]   | 95     | A11 | O                   | IPD                 |       |  |
| EMB_A[6]/GP7[8]   | 96     | B11 | O                   | IPD                 |       |  |
| EMB_A[5]/GP7[7]   | 97     | C11 | O                   | IPD                 |       |  |
| EMB_A[4]/GP7[6]   | 98     | D11 | O                   | IPD                 | GPIO  | EMIFB SDRAM row/column address         |
| EMB_A[3]/GP7[5]   | 100    | A10 | O                   | IPD                 |       |  |
| EMB_A[2]/GP7[4]   | 101    | B10 | O                   | IPD                 |       |  |
| EMB_A[1]/GP7[3]   | 102    | C10 | O                   | IPD                 |       |  |
| EMB_A[0]/GP7[2]   | 103    | D10 | O                   | IPD                 |       |  |
| EMB_BA[1]/GP7[0]  | 106    | B9  | O                   | IPU                 | GPIO  | EMIFB SDRAM bank address               |
| EMB_BA[0]/GP7[1]  | 107    | C9  | O                   | IPU                 |       |  |
| EMB_CLK           | 86     | C14 | O                   | IPU                 | GPIO  | EMIF SDRAM clock                       |
| EMB_SDCKE         | 88     | C13 | O                   | IPU                 |       | EMIFB SDRAM clock enable               |
| EMB_WE            | 59     | K15 | I/O                 | IPU                 |       | EMIFB write enable                     |
| EMB_RAS           | 110    | A8  | O                   | IPU                 |       | EMIFB SDRAM row address strobe         |
| EMB_CAS           | 57     | L13 | O                   | IPU                 |       | EMIFB column address strobe            |
| EMB_CS[0]         | 108    | D9  | O                   | IPU                 |       | EMIFB SDRAM chip select 0              |
| EMB_WE_DQM[1]     | 85     | C15 | O                   | IPU                 | GPIO  | EMIFB write enable/data mask for EMB_D |
| EMB_WE_DQM[0]     | 60     | K14 | O                   | IPU                 |       |  |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.5 Serial Peripheral Interface Modules (SPI0)

**Table 3-9. Serial Peripheral Interface (SPI) Terminal Functions**

| SIGNAL NAME  | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED                     | DESCRIPTION                   |
|--|--------|-----|---------------------|---------------------|---------------------------|-------------------------------|
|  | PTP    | ZKB |                     |                     |                           |                               |
| <b>SPI0</b>  |        |     |                     |                     |                           |                               |
| <b>SPI0_SCS[0]</b> / $\overline{\text{UART0\_RTS}}$ /EQEP0B/GP5[4]/BOOT[4] | 9      | N4  | I/O                 | IPU                 | UART0, EQEP0B, GPIO, BOOT | SPI0 chip select              |
| <b>SPI0_ENA</b> / $\overline{\text{UART0\_CTS}}$ /EQEP0A/GP5[3]/BOOT[3]    | 12     | R5  | I/O                 | IPU                 | UART0, EQEP0A, GPIO, BOOT | SPI0 enable                   |
| <b>SPI0_CLK</b> /EQEP1I/GP5[2]/BOOT[2]                                     | 11     | T5  | I/O                 | IPD                 | eQEP1, GPIO, BOOT         | SPI0 clock                    |
| <b>SPI0_SIMO[0]</b> /EQEP0S/GP5[1]/BOOT[1]                                 | 18     | P6  | I/O                 | IPD                 | eQEP0, GPIO, BOOT         | SPI0 data slave-in-master-out |
| <b>SPI0_SOMI[0]</b> /EQEP0I/GP5[0]/BOOT[0]                                 | 17     | R6  | I/O                 | IPD                 | eQEP0, GPIO, BOOT         | SPI0 data slave-out-master-in |

(1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.

**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.

(2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.6 Enhanced Capture/Auxiliary PWM Modules (eCAP0, eCAP1, eCAP2)

The eCAP Module pins function as either input captures or auxiliary PWM 32-bit outputs, depending upon how the eCAP module is programmed.

**Table 3-10. Enhanced Capture Module (eCAP) Terminal Functions**

| SIGNAL NAME                         | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED        | DESCRIPTION  |
|-------------------------------------|--------|-----|---------------------|---------------------|--------------|--|
|                                     | PTP    | ZKB |                     |                     |              |  |
| <b>eCAP0</b>                        |        |     |                     |                     |              |  |
| ACLKX0/ <b>ECAP0/APWM0</b> /GP2[12] | 126    | C5  | I/O                 | IPD                 | McASP0, GPIO | enhanced capture 0 input or auxiliary PWM 0 output |
| <b>eCAP1</b>                        |        |     |                     |                     |              |  |
| ACLKR0/ <b>ECAP1/APWM1</b> /GP2[15] | 130    | B4  | I/O                 | IPD                 | McASP0, GPIO | enhanced capture 1 input or auxiliary PWM 1 output |
| <b>eCAP2</b>                        |        |     |                     |                     |              |  |
| ACLKR1/ <b>ECAP2/APWM2</b> /GP4[12] | 165    | L2  | I/O                 | IPD                 | McASP1, GPIO | enhanced capture 2 input or auxiliary PWM 2 output |

(1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.

**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.

(2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.7 Enhanced Pulse Width Modulators (eHRPWM0, eHRPWM1, eHRPWM2)

**Table 3-11. Enhanced Pulse Width Modulator (eHRPWM) Terminal Functions**

| SIGNAL NAME                       | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED                          | DESCRIPTION   |
|-----------------------------------|--------|-----|---------------------|---------------------|--------------------------------|---|
|                                   | PTP    | ZKB |                     |                     |                                |   |
| <b>eHRPWM0</b>                    |        |     |                     |                     |                                |   |
| ACLKX1/EPWM0A/GP3[15]             | 162    | K3  | I/O                 | IPD                 | McASP1, GPIO                   | eHRPWM0 A output (with high-resolution)                     |
| AHCLKX1/EPWM0B/GP3[14]            | 160    | K2  | I/O                 | IPD                 |                                | eHRPWM0 B output  |
| AMUTE1/EPWMTZ/GP4[14]             | 132    | D4  | I/O                 | IPD                 | McASP1, eHRPWM1, GPIO, eHRPWM2 | eHRPWM0 trip zone input                                     |
| AFSX1/EPWMSYNCI/EPWMSYNCO/GP4[10] | 163    | K4  | I/O                 | IPD                 | McASP1, eHRPWM0, GPIO          | Sync input to eHRPWM0 module or sync output to external PWM |
| <b>eHRPWM1</b>                    |        |     |                     |                     |                                |   |
| AXR1[8]/EPWM1A/GP4[8]             | 168    | M2  | I/O                 | IPD                 | McASP1, GPIO                   | eHRPWM1 A (with high-resolution)                            |
| AXR1[7]/EPWM1B/GP4[7]             | 169    | M3  | I/O                 | IPD                 |                                | eHRPWM1 B output  |
| AMUTE1/EPWMTZ/GP4[14]             | 132    | D4  | I/O                 | IPD                 | McASP1, eHRPWM0, GPIO, eHRPWM2 | eHRPWM1 trip zone input                                     |
| <b>eHRPWM2</b>                    |        |     |                     |                     |                                |   |
| AXR1[6]/EPWM2A/GP4[6]             | 170    | M4  | I/O                 | IPD                 | McASP1, GPIO                   | eHRPWM2 A (with high-resolution)                            |
| AXR1[5]/EPWM2B/GP4[5]             | 171    | N1  | I/O                 | IPD                 |                                | eHRPWM2 B output  |
| AMUTE1/EPWMTZ/GP4[14]             | 132    | D4  | I/O                 | IPD                 | McASP1, eHRPWM0, GPIO, eHRPWM2 | eHRPWM2 trip zone input                                     |

(1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.

**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.

(2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.8 Enhanced Quadrature Encoder Pulse Module (eQEP)

**Table 3-12. Enhanced Quadrature Encoder Pulse Module (eQEP) Terminal Functions**

| SIGNAL NAME                                 | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED                      | DESCRIPTION             |
|---|--------|-----|---------------------|---------------------|----------------------------|-------------------------|
|   | PTP    | ZKB |                     |                     |                            |                         |
| <b>eQEP0</b>                                |        |     |                     |                     |                            |                         |
| SPI0_ENA/UART0_CTS/EQEP0A/GP5[3]/BOOT[3]    | 12     | R5  | I                   | IPU                 | SPI0, UART0,<br>GPIO, BOOT | eQEP0A quadrature input |
| SPI0_SCS[0]/UART0_RTS/EQEP0B/GP5[4]/BOOT[4] | 9      | N4  | I                   | IPU                 |                            | eQEP0B quadrature input |
| SPI0_SOMI[0]/EQEP0I/GP5[0]/BOOT[0]          | 17     | R6  | I                   | IPD                 | SPI0, GPIO, BOOT           | eQEP0 index             |
| SPI0_SIMO[0]/EQEP0S/GP5[1]/BOOT[1]          | 18     | P6  | I                   | IPD                 |                            | eQEP0 strobe            |
| <b>eQEP1</b>                                |        |     |                     |                     |                            |                         |
| AXR1[3]/EQEP1A/GP4[3]                       | 174    | P1  | I                   | IPD                 | McASP1, GPIO               | eQEP1A quadrature input |
| AXR1[4]/EQEP1B/GP4[4]                       | 173    | N2  | I                   | IPD                 |                            | eQEP1B quadrature input |
| SPI0_CLK/EQEP1I/GP5[2]/BOOT[2]              | 11     | T5  | I                   | IPD                 | SPI0, GPIO, BOOT           | eQEP1 index             |
| EQEP1S/GP5[7]/BOOT[7]                       | 16     | T6  | I                   | IPD                 | GPIO, BOOT                 | eQEP1 strobe            |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.9 Boot

**Table 3-13. Boot Terminal Functions<sup>(1)</sup>**

| SIGNAL NAME  | PIN NO |     | TYPE <sup>(2)</sup> | PULL <sup>(3)</sup> | MUXED                     | DESCRIPTION |
|--|--------|-----|---------------------|---------------------|---------------------------|-------------|
|  | PTP    | ZKB |                     |                     |                           |             |
| $\overline{\text{EMA\_CS}}[2]/\text{GP2}[5]/\text{BOOT}[15]$   | 23     | P7  | I                   | IPU                 | EMIFA, GPIO               | BOOT[15]    |
| $\overline{\text{EMA\_WE}}/\text{AXR0}[12]/\text{GP2}[3]/\text{BOOT}[14]$                                | 55     | M13 | I                   | IPU                 | EMIFA, McASP0, GPIO       | BOOT[14]    |
| $\text{EMA\_D}[7]/\text{MMCS\_D}[7]/\text{GP0}[7]/\text{BOOT}[13]$                                       | 54     | M15 | I                   | IPU                 | EMIFA, MMC/SD, GPIO       | BOOT[13]    |
| $\text{EMA\_D}[0]/\text{MMCS\_D}[0]/\text{GP0}[0]/\text{BOOT}[12]$                                       | 44     | T13 | I                   | IPU                 |                           | BOOT[12]    |
| $\text{AHCLKR0}/\text{RMII\_MHZ\_50\_CLK}/\text{GP2}[14]/\text{BOOT}[11]$                                | 129    | A4  | I                   | IPD                 | McASP0, EMAC, GPIO        | BOOT[11]    |
| $\text{AFSX0}/\text{GP2}[13]/\text{BOOT}[10]$  | 127    | D5  | I                   | IPD                 | McASP0, GPIO              | BOOT[10]    |
| $\text{UART0\_TXD}/\text{I2C0\_SCL}/\text{TM64P0\_OUT12}/\text{GP5}[9]/\text{BOOT}[9]$                   | 3      | P3  | I                   | IPU                 | UART0, I2C0, Timer0, GPIO | BOOT[9]     |
| $\text{UART0\_RXD}/\text{I2C0\_SDA}/\text{TM64P0\_IN12}/\text{GP5}[8]/\text{BOOT}[8]$                    | 2      | R3  | I                   | IPU                 | UART0, I2C0, Timer0, GPIO | BOOT[8]     |
| $\text{EQEP1S}/\text{GP5}[7]/\text{BOOT}[7]$   | 16     | T6  | I                   | IPD                 | eQEP1, GPIO               | BOOT[7]     |
| $\text{I2C1\_SDA}/\text{GP5}[6]/\text{BOOT}[6]$  | 14     | N5  | I                   | IPU                 | I2C1, GPIO                | BOOT[6]     |
| $\text{I2C1\_SCL}/\text{GP5}[5]/\text{BOOT}[5]$  | 13     | P5  | I                   | IPU                 |                           | BOOT[5]     |
| $\overline{\text{SPI0\_SCS}}[0]/\overline{\text{UART0\_RTS}}/\text{EQEP0B}/\text{GP5}[4]/\text{BOOT}[4]$ | 9      | N4  | I                   | IPU                 | SPI0, UART0, eQEP0, GPIO  | BOOT[4]     |
| $\overline{\text{SPI0\_ENA}}/\text{UART0\_CTS}/\text{EQEP0A}/\text{GP5}[3]/\text{BOOT}[3]$               | 12     | R5  | I                   | IPU                 | SPI0, UART0, eQEP0, GPIO  | BOOT[3]     |
| $\text{SPI0\_CLK}/\text{EQEP1I}/\text{GP5}[2]/\text{BOOT}[2]$  | 11     | T5  | I                   | IPD                 | SPI0, eQEP1, GPIO         | BOOT[2]     |
| $\text{SPI0\_SIMO}[0]/\text{EQEP0S}/\text{GP5}[1]/\text{BOOT}[1]$  | 18     | P6  | I                   | IPD                 | SPI0, eQEP0, GPIO         | BOOT[1]     |
| $\text{SPI0\_SOMI}[0]/\text{EQEP0I}/\text{GP5}[0]/\text{BOOT}[0]$  | 17     | R6  | I                   | IPD                 |                           | BOOT[0]     |

(1) Boot decoding will be defined in the ROM datasheet.

(2) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.

**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.

(3) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.10 Universal Asynchronous Receiver/Transmitters (UART0, UART2)

**Table 3-14. Universal Asynchronous Receiver/Transmitter (UART) Terminal Functions**

| SIGNAL NAME   | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED                     | DESCRIPTION                |
|---|--------|-----|---------------------|---------------------|---------------------------|----------------------------|
|   | PTP    | ZKB |                     |                     |                           |                            |
| <b>UART0</b>  |        |     |                     |                     |                           |                            |
| UART0_RXD/I2C0_SDA/TM64P0_IN12/GP5[8]/BOOT[8]   | 2      | R3  | I                   | IPU                 | I2C0, BOOT, Timer0, GPIO, | UART0 receive data         |
| UART0_TXD/I2C0_SCL/TM64P0_OUT12/GP5[9]/BOOT[9]  | 3      | P3  | O                   | IPU                 | I2C0, Timer0, GPIO, BOOT  | UART0 transmit data        |
| $\overline{\text{SPIO\_SCS}}[0]/ \overline{\text{UART0\_RTS}} / \text{EQEP0B}/\text{GP5}[4]/\text{BOOT}[4]$ | 9      | N4  | O                   | IPU                 | SPI0, eQEP0, GPIO, BOOT   | UART0 ready-to-send output |
| $\overline{\text{SPIO\_ENA}}/ \overline{\text{UART0\_CTS}} / \text{EQEP0A}/\text{GP5}[3]/\text{BOOT}[3]$    | 12     | R5  | I                   | IPU                 |                           | UART0 clear-to-send input  |
| <b>UART2</b>  |        |     |                     |                     |                           |                            |
| UART2_RXD/GP5[12]   | 7      | R4  | I                   | IPU                 | GPIO                      | UART2 receive data         |
| UART2_TXD/GP5[13]   | 8      | P4  | O                   | IPU                 |                           | UART2 transmit data        |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.11 Inter-Integrated Circuit Modules (I2C0, I2C1)

**Table 3-15. Inter-Integrated Circuit (I2C) Terminal Functions**

| SIGNAL NAME                                    | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED                     | DESCRIPTION       |
|--|--------|-----|---------------------|---------------------|---------------------------|-------------------|
|  | PTP    | ZKB |                     |                     |                           |                   |
| <b>I2C0</b>                                    |        |     |                     |                     |                           |                   |
| UART0_RXD/I2C0_SDA/TM64P0_IN12/GP5[8]/BOOT[8]  | 2      | R3  | I/O                 | IPU                 | UART0, Timer0, GPIO, BOOT | I2C0 serial data  |
| UART0_TXD/I2C0_SCL/TM64P0_OUT12/GP5[9]/BOOT[9] | 3      | P3  | I/O                 | IPU                 | UART0, Timer0, GPIO, BOOT | I2C0 serial clock |
| <b>I2C1</b>                                    |        |     |                     |                     |                           |                   |
| I2C1_SDA/GP5[6]/BOOT[6]                        | 14     | N5  | I/O                 | IPU                 | GPIO, BOOT                | I2C1 serial data  |
| I2C1_SCL/GP5[5]/BOOT[5]                        | 13     | P5  | I/O                 | IPU                 |                           | I2C1 serial clock |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.12 Timers

**Table 3-16. Timers Terminal Functions**

| SIGNAL NAME   | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED                      | DESCRIPTION         |
|---|--------|-----|---------------------|---------------------|----------------------------|---------------------|
|   | PTP    | ZKB |                     |                     |                            |                     |
| <b>TIMER0</b>   |        |     |                     |                     |                            |                     |
| UART0_RXD/I2C0_SDA/TM64P0_IN12/GP5[8]/BOOT[8]                                     | 2      | R3  | I                   | IPU                 | UART0, I2C0,<br>GPIO, BOOT | Timer0 lower input  |
| UART0_TXD/I2C0_SCL/TM64P0_OUT12/GP5[9]/BOOT[9]                                    | 3      | P3  | O                   | IPU                 |                            | Timer0 lower output |
| <b>TIMER1 (Watchdog)</b>  |        |     |                     |                     |                            |                     |
| No external pins. The Timer1 peripheral pins are not pinned out as external pins. |        |     |                     |                     |                            |                     |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.13 Multichannel Audio Serial Ports (McASP0, McASP1)

**Table 3-17. Multichannel Audio Serial Ports (McASPs) Terminal Functions**

| SIGNAL NAME                              | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED             | DESCRIPTION                  |
|--|--------|-----|---------------------|---------------------|-------------------|------------------------------|
|  | PTP    | ZKB |                     |                     |                   |                              |
| <b>McASP0</b>                            |        |     |                     |                     |                   |                              |
| EMA_OE/AXR0[13]/GP2[7]                   | 22     | R7  | I/O                 | IPU                 | EMIFA, GPIO       | McASP0 serial data           |
| EMA_WE/AXR0[12]/GP2[3]/BOOT[14]          | 55     | M13 | I/O                 | IPU                 | EMIFA, GPIO, BOOT |                              |
| AXR0[11]/GP3[11]                         | 124    | A5  | I/O                 | IPD                 | GPIO              |                              |
| AXR0[10]/GP3[10]                         | 123    | D6  | I/O                 | IPD                 | GPIO              |                              |
| AXR0[9]/GP3[9]                           | 122    | C6  | I/O                 | IPD                 | GPIO              |                              |
| AXR0[8]/MDIO_D/GP3[8]                    | 121    | B6  | I/O                 | IPU                 | MDIO, GPIO        |                              |
| AXR0[7]/MDIO_CLK/GP3[7]                  | 120    | A6  | I/O                 | IPD                 |                   |                              |
| AXR0[6]/RMII_RXER/GP3[6]                 | 118    | D7  | I/O                 | IPD                 | EMAC, GPIO        |                              |
| AXR0[5]/RMII_RXD[1]/GP3[5]               | 117    | C7  | I/O                 | IPD                 |                   |                              |
| AXR0[4]/RMII_RXD[0]/GP3[4]               | 116    | B7  | I/O                 | IPD                 |                   |                              |
| AXR0[3]/RMII_CRS_DV/GP3[3]               | 115    | A7  | I/O                 | IPD                 |                   |                              |
| AXR0[2]/RMII_TXEN/GP3[2]                 | 113    | D8  | I/O                 | IPD                 |                   |                              |
| AXR0[1]/RMII_TXD[1]/GP3[1]               | 112    | C8  | I/O                 | IPD                 |                   |                              |
| AXR0[0]/RMII_TXD[0]/GP3[0]               | 111    | B8  | I/O                 | IPD                 |                   |                              |
| AHCLKX0/GP2[11]                          | 125    | B5  | I/O                 | IPD                 | GPIO              | McASP0 transmit master clock |
| ACLKX0/ECAP0/APWM0/GP2[12]               | 126    | C5  | I/O                 | IPD                 | eCAP0, GPIO       | McASP0 transmit bit clock    |
| AFSX0/GP2[13]/BOOT[10]                   | 127    | D5  | I/O                 | IPD                 | GPIO, BOOT        | McASP0 transmit frame sync   |
| AHCLKR0/RMII_MHZ_50_CLK/GP2[14]/BOOT[11] | 129    | A4  | I/O                 | IPD                 | EMAC, GPIO, BOOT  | McASP0 receive master clock  |
| ACLKR0/ECAP1/APWM1/GP2[15]               | 130    | B4  | I/O                 | IPD                 | eCAP1, GPIO       | McASP0 receive bit clock     |
| AFSR0/GP3[12]                            | 131    | C4  | I/O                 | IPD                 | GPIO              | McASP0 receive frame sync    |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

**Table 3-17. Multichannel Audio Serial Ports (McASPs) Terminal Functions (continued)**

| SIGNAL NAME                       | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED                           | DESCRIPTION                  |
|-----------------------------------|--------|-----|---------------------|---------------------|---------------------------------|------------------------------|
|                                   | PTP    | ZKB |                     |                     |                                 |                              |
| <b>McASP1</b>                     |        |     |                     |                     |                                 |                              |
| AXR1[8]/EPWM1A/GP4[8]             | 168    | M2  | I/O                 | IPD                 | eHRPWM1 A, GPIO                 | McASP1 serial data           |
| AXR1[7]/EPWM1B/GP4[7]             | 169    | M3  | I/O                 | IPD                 | eHRPWM1 B, GPIO                 |                              |
| AXR1[6]/EPWM2A/GP4[6]             | 170    | M4  | I/O                 | IPD                 | eHRPWM2 A, GPIO                 |                              |
| AXR1[5]/EPWM2B/GP4[5]             | 171    | N1  | I/O                 | IPD                 | eHRPWM2 B, GPIO                 |                              |
| AXR1[4]/EQEP1B/GP4[4]             | 173    | N2  | I/O                 | IPD                 | eQEP, GPIO                      |                              |
| AXR1[3]/EQEP1A/GP4[3]             | 174    | P1  | I/O                 | IPD                 |                                 |                              |
| AXR1[2]/GP4[2]                    | 175    | P2  | I/O                 | IPD                 | GPIO                            |                              |
| AXR1[1]/GP4[1]                    | 176    | R2  | I/O                 | IPD                 |                                 |                              |
| AXR1[0]/GP4[0]                    | 1      | T3  | I/O                 | IPD                 |                                 |                              |
| AHCLKX1/EPWM0B/GP3[14]            | 160    | K2  | I/O                 | IPD                 | eHRPWM0, GPIO                   | McASP1 transmit master clock |
| ACLKX1/EPWM0A/GP3[15]             | 162    | K3  | I/O                 | IPD                 | eHRPWM0, GPIO                   | McASP1 transmit bit clock    |
| AFSX1/EPWMSYNCO/EPWMSYNCO/GP4[10] | 163    | K4  | I/O                 | IPD                 | eHRPWM0, GPIO                   | McASP1 transmit frame sync   |
| AHCLKR1/GP4[11]                   | -      | L1  | I/O                 | IPD                 | GPIO                            | McASP1 receive master clock  |
| ACLKR1/ECAP2/APWM2/GP4[12]        | 165    | L2  | I/O                 | IPD                 | eCAP2, GPIO                     | McASP1 receive bit clock     |
| AFSR1/GP4[13]                     | 166    | L3  | I/O                 | IPD                 | GPIO                            | McASP1 receive frame sync    |
| AMUTE1/EPWMTZ/GP4[14]             | 132    | D4  | O                   | IPD                 | eHRPWM0, eHRPWM1, GPIO, eHRPWM2 | McASP1 mute output           |

### 3.6.14 Ethernet Media Access Controller (EMAC)

**Table 3-18. Ethernet Media Access Controller (EMAC) Terminal Functions**

| SIGNAL NAME                              | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED              | DESCRIPTION                        |
|--|--------|-----|---------------------|---------------------|--------------------|------------------------------------|
|  | PTP    | ZKB |                     |                     |                    |                                    |
| <b>RMII</b>                              |        |     |                     |                     |                    |                                    |
| AHCLKR0/RMII_MHZ_50_CLK/GP2[14]/BOOT[11] | 129    | A4  | I/O                 | IPD                 | McASP0, GPIO, BOOT | EMAC 50-MHz clock input or output  |
| AXR0[6]/RMII_RXER/GP3[6]                 | 118    | D7  | I                   | IPD                 | McASP0, GPIO       | EMAC RMII receiver error           |
| AXR0[5]/RMII_RXD[1]/GP3[5]               | 117    | C7  | I                   | IPD                 |                    | EMAC RMII receive data             |
| AXR0[4]/RMII_RXD[0]/GP3[4]               | 116    | B7  | I                   | IPD                 |                    | EMAC RMII carrier sense data valid |
| AXR0[3]/RMII_CRS_DV/GP3[3]               | 115    | A7  | I                   | IPD                 |                    | EMAC RMII transmit enable          |
| AXR0[2]/RMII_TXEN/GP3[2]                 | 113    | D8  | O                   | IPD                 |                    | EMAC RMII transmit data            |
| AXR0[1]/RMII_TXD[1]/GP3[1]               | 112    | C8  | O                   | IPD                 |                    |                                    |
| AXR0[0]/RMII_TXD[0]/GP3[0]               | 111    | B8  | O                   | IPD                 |                    |                                    |
| <b>MDIO</b>                              |        |     |                     |                     |                    |                                    |
| AXR0[8]/MDIO_D/GP3[8]                    | 121    | B6  | I/O                 | IPU                 | McASP0, GPIO       | MDIO serial data                   |
| AXR0[7]/MDIO_CLK/GP3[7]                  | 120    | A6  | O                   | IPD                 |                    | MDIO data clock                    |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.15 Multimedia Card/Secure Digital (MMC/SD)

**Table 3-19. Multimedia Card/Secure Digital (MMC/SD) Terminal Functions**

| SIGNAL NAME                            | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED             | DESCRIPTION |
|--|--------|-----|---------------------|---------------------|-------------------|-------------|
|  | PTP    | ZKB |                     |                     |                   |             |
| EMA_A[1]/MMCS_D_CLK/GP1[1]             | 30     | R9  | O                   | IPU                 | EMIFA, GPIO       | MMCS_D_CLK  |
| EMA_A[2]/MMCS_D_CMD/GP1[2]             | 31     | P9  | I/O                 | IPU                 |                   | MMCS_D_CMD  |
| EMA_D[7]/MMCS_D_DAT[7]/GP0[7]/BOOT[13] | 54     | M15 | I/O                 | IPU                 | EMIFA, GPIO, BOOT | MMC/SD data |
| EMA_D[6]/MMCS_D_DAT[6]/GP0[6]          | 52     | N13 | I/O                 | IPU                 | EMIFA, GPIO       |             |
| EMA_D[5]/MMCS_D_DAT[5]/GP0[5]          | 51     | N15 | I/O                 | IPU                 |                   |             |
| EMA_D[4]/MMCS_D_DAT[4]/GP0[4]          | 49     | P13 | I/O                 | IPU                 |                   |             |
| EMA_D[3]/MMCS_D_DAT[3]/GP0[3]          | 48     | P15 | I/O                 | IPU                 |                   |             |
| EMA_D[2]/MMCS_D_DAT[2]/GP0[2]          | 46     | R13 | I/O                 | IPU                 |                   |             |
| EMA_D[1]/MMCS_D_DAT[1]/GP0[1]          | 45     | R15 | I/O                 | IPU                 |                   |             |
| EMA_D[0]/MMCS_D_DAT[0]/GP0[0]/BOOT[12] | 44     | T13 | I/O                 | IPU                 | EMIFA, GPIO, BOOT |             |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor

### 3.6.16 General-Purpose IO Only Terminal Functions

Table 3-20. General-Purpose IO Only Terminal Functions

| SIGNAL NAME            | PIN NO |     | TYPE <sup>(1)</sup> | PULL <sup>(2)</sup> | MUXED     | DESCRIPTION        |
|------------------------|--------|-----|---------------------|---------------------|-----------|--------------------|
|                        | PTP    | ZKB |                     |                     |           |                    |
| GP0[15]                | -      | M16 | I/O/Z               | IPD                 | None      | General-Purpose IO |
| GP0[14]                | -      | N14 | I/O/Z               | IPD                 |           |                    |
| GP0[13]                | -      | N16 | I/O/Z               | IPD                 |           |                    |
| GP0[12]                | -      | P14 | I/O/Z               | IPD                 |           |                    |
| GP0[11]                | -      | P16 | I/O/Z               | IPD                 |           |                    |
| GP0[10]                | -      | R14 | I/O/Z               | IPD                 |           |                    |
| GP0[9]                 | -      | T14 | I/O/Z               | IPD                 |           |                    |
| GP0[8]                 | -      | N12 | I/O/Z               | IPD                 |           |                    |
| GP5[11]                | 6      | -   | I/O/Z               | IPD                 |           |                    |
| GP5[10]                | 4      | -   | I/O/Z               | IPD                 |           |                    |
| GP7[14] <sup>(3)</sup> | 157    | K1  | I/O                 | IPD                 |           |                    |
| EMU0/GP7[15]           | -      | J5  | I/O                 | IPD                 | Emulation |                    |

- (1) I = Input, O = Output, I/O = Bidirectional, Z = High impedance, PWR = Supply voltage, GND = Ground, A = Analog signal.  
**Note:** For multiplexed pins where functions have different types (i.e., input versus output), the table reflects the pin function direction for that particular peripheral.
- (2) IPD = Internal Pulldown resistor, IPU = Internal Pullup resistor
- (3) GP7[14] is initially configured as a reserved function after reset and will not be in a predictable state. This signal will be stable only after the GPIO configuration for this pin has been completed. Users should carefully consider the system implications of this pin being in an unknown state after reset.

### 3.6.17 Reserved and No Connect Terminal Functions

Table 3-21. Reserved and No Connect Terminal Functions

| SIGNAL NAME | PIN NO                        |  | TYPE <sup>(1)</sup> | DESCRIPTION   |
|-------------|-------------------------------|--|---------------------|---|
|             | PTP                           | ZKB  |                     |   |
| RSV1        | -                             | F7   | -                   | Reserved. (Leave unconnected, <b>do not</b> connect to power or ground.)  |
| RSV2        | 133                           | B1   | PWR                 | Reserved. For proper device operation, this pin <b>must</b> be tied either directly to CV <sub>DD</sub> or left unconnected (do not connect to ground). |
| RSV3        | 149                           | -  | PWR                 | Reserved. For proper device operation, this pin <b>must</b> be tied directly to CV <sub>DD</sub> .  |
| RSV4        | 148                           | H1   | I                   | Reserved. For proper device operation, this pin must be tied low or to CV <sub>DD</sub> .   |
| NC          | 134, 135,<br>137, 138,<br>140 | A12, A13, A14, B13,<br>B14, C1, C2, C3, D2,<br>D3, E3, E4, E14, E15,<br>F14, F15, G14, G15, H2,<br>H5, H14, H15, J14, K13,<br>K16, L4, L14, L15, L16,<br>M1, M14, N3, N7, P12,<br>R12, T4, T8, T12 | -                   | No Connect (leave unconnected)  |
|             | 136, 139                      | F3, H4,  | -                   | This pin may be left unconnected or connected to V <sub>SS</sub> .  |

- (1) PWR = Supply voltage.

### 3.6.18 Supply and Ground Terminal Functions

**Table 3-22. Supply and Ground Terminal Functions**

| SIGNAL NAME                       | PIN NO  |   | TYPE <sup>(1)</sup> | DESCRIPTION                           |
|-----------------------------------|---|---|---------------------|---------------------------------------|
|                                   | PTP   | ZKB   |                     |                                       |
| <b>CVDD</b> (Core supply)         | 10, 20, 28, 38, 50, 56,<br>61, 69, 77, 93, 104,<br>114, 147, 154, 161, 167                                | F6, G1, G6, G7, G10,<br>G11, H7, H10, H11, J6,<br>J7, J10, J11, J12, K6,<br>K7, K10, K11, L6  | PWR                 | 1.2-V core supply voltage pins        |
| <b>RVDD</b> (Internal RAM supply) | 67, 159   | H6, H12   | PWR                 | 1.2V internal ram supply voltage pins |
| <b>DVDD</b> (I/O supply)          | 5, 15, 24, 33, 43, 47,<br>53, 58, 65, 71, 75, 81,<br>87, 90, 99, 109, 119,<br>128, 151, 158, 164,<br>172, | B16, E5, E8, E9, E12,<br>F5, F11, F12, G5, G12,<br>K5, K12, L5, L11, L12,<br>M5, M8, M9, M12, R1,<br>R16  | PWR                 | 3.3-V I/O supply voltage pins.        |
| <b>V<sub>SS</sub></b> (Ground)    | 177   | A1, A2, A3, A15, A16,<br>B2, B3, E6, E7, E10,<br>E11, F4, F8, F9, F10,<br>G2, G4, G8, G9, H8,<br>H9, J8, J9, K8, K9, L7,<br>L8, L9, L10, M6, M7,<br>M10, M11, T1, T2, T15,<br>T16 | GND                 | Ground pins.                          |

(1) PWR = Supply voltage, GND - Ground.

## 4 Device Configuration

### 4.1 Boot Modes

This device supports a variety of boot modes through an internal ROM bootloader. This device does not support dedicated hardware boot modes; therefore, all boot modes utilize the internal ROM. The input states of the BOOT pins are sampled and latched into the BOOTCFG register, which is part of the system configuration (SYSCFG) module, when device reset is deasserted. Boot mode selection is determined by the values of the BOOT pins.

See Using the D800K001 Bootloader Application Report ([SPRAB04](#)) for more details on the ROM Boot Loader.

The following boot modes are supported:

- NAND Flash boot
  - 8-bit NAND
- NOR Flash boot
  - NOR Direct boot (8-bit)
  - NOR Legacy boot (8-bit)
  - NOR AIS boot (8-bit)
- I2C0 / I2C1 Boot
  - EEPROM (Master Mode)
  - External Host (Slave Mode)
- SPI0 Boot
  - Serial Flash (Master Mode)
  - SERIAL EEPROM (Master Mode)
  - External Host (Slave Mode)
- UART0 / UART2 Boot
  - External Host

### 4.2 SYSCFG Module

The following system level features of the chip are controlled by the SYSCFG peripheral:

- Readable Device, Die, and Chip Revision ID
- Control of Pin Multiplexing
- Priority of bus accesses different bus masters in the system
- Capture at power on reset the chip BOOT[15:0] pin values and make them available to software
- Special case settings for peripherals:
  - Locking of PLL controller settings
  - Default burst sizes for EDMA3 TC0 and TC1
  - Selection of the source for the eCAP module input capture (including on chip sources)
  - Clock source selection for EMIFA and EMIFB
- Selects the source of emulation suspend signal (from DSP) of peripherals supporting this function.

Since the SYSCFG peripheral controls global operation of the device, its registers are protected against erroneous accesses by several mechanisms:

- Additionally, many registers are accessible only by a host (DSP) when it is operating in its privileged mode. (ex. from the kernel, but not from user space code).

**Table 4-1. System Configuration (SYSCFG) Module Register Access**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION             | ACCESS |
|--------------|---------------|----------------------------------|--------|
| 0x01C1 4000  | REVID         | Revision Identification Register | —      |

**Table 4-1. System Configuration (SYSCFG) Module Register Access (continued)**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                     | ACCESS          |
|--------------|---------------|--|-----------------|
| 0x01C1 4008  | DIEIDR0       | Device Identification Register 0         | —               |
| 0x01C1 400C  | DIEIDR1       | Device Identification Register 1         | —               |
| 0x01C1 4010  | DIEIDR2       | Device Identification Register 2         | —               |
| 0x01C1 4014  | DIEIDR3       | Device Identification Register 3         | —               |
| 0x01C1 4018  | DEVIDR0       | JTAG Identification Register             | —               |
| 0x01C1 4020  | BOOTCFG       | Boot Configuration Register              | Privileged mode |
| 0x01C1 4024  | CHIPREVID     | Silicon Revision Identification Register | Privileged mode |
| 0x01C1 4038  | KICK0R        | Kick 0 Register                          | Privileged mode |
| 0x01C1 403C  | KICK1R        | Kick 1 Register                          | Privileged mode |
| 0x01C1 4040  | HOST0CFG      | Host 0 Configuration Register            | —               |
| 0x01C1 4044  | HOST1CFG      | Host 1 Configuration Register            | —               |
| 0x01C1 40E0  | IRAWSTAT      | Interrupt Raw Status/Set Register        | Privileged mode |
| 0x01C1 40E4  | IENSTAT       | Interrupt Enable Status/Clear Register   | Privileged mode |
| 0x01C1 40E8  | IENSET        | Interrupt Enable Register                | Privileged mode |
| 0x01C1 40EC  | IENCLR        | Interrupt Enable Clear Register          | Privileged mode |
| 0x01C1 40F0  | EOI           | End of Interrupt Register                | Privileged mode |
| 0x01C1 40F4  | FLTADDRR      | Fault Address Register                   | Privileged mode |
| 0x01C1 40F8  | FLTSTAT       | Fault Status Register                    | —               |
| 0x01C1 4110  | MSTPRI0       | Master Priority 0 Register               | Privileged mode |
| 0x01C1 4114  | MSTPRI1       | Master Priority 1 Register               | Privileged mode |
| 0x01C1 4118  | MSTPRI2       | Master Priority 2 Register               | Privileged mode |
| 0x01C1 4120  | PINMUX0       | Pin Multiplexing Control 0 Register      | Privileged mode |
| 0x01C1 4124  | PINMUX1       | Pin Multiplexing Control 1 Register      | Privileged mode |
| 0x01C1 4128  | PINMUX2       | Pin Multiplexing Control 2 Register      | Privileged mode |
| 0x01C1 412C  | PINMUX3       | Pin Multiplexing Control 3 Register      | Privileged mode |
| 0x01C1 4130  | PINMUX4       | Pin Multiplexing Control 4 Register      | Privileged mode |
| 0x01C1 4134  | PINMUX5       | Pin Multiplexing Control 5 Register      | Privileged mode |
| 0x01C1 4138  | PINMUX6       | Pin Multiplexing Control 6 Register      | Privileged mode |
| 0x01C1 413C  | PINMUX7       | Pin Multiplexing Control 7 Register      | Privileged mode |
| 0x01C1 4140  | PINMUX8       | Pin Multiplexing Control 8 Register      | Privileged mode |
| 0x01C1 4144  | PINMUX9       | Pin Multiplexing Control 9 Register      | Privileged mode |
| 0x01C1 4148  | PINMUX10      | Pin Multiplexing Control 10 Register     | Privileged mode |
| 0x01C1 414C  | PINMUX11      | Pin Multiplexing Control 11 Register     | Privileged mode |
| 0x01C1 4150  | PINMUX12      | Pin Multiplexing Control 12 Register     | Privileged mode |
| 0x01C1 4154  | PINMUX13      | Pin Multiplexing Control 13 Register     | Privileged mode |
| 0x01C1 4158  | PINMUX14      | Pin Multiplexing Control 14 Register     | Privileged mode |
| 0x01C1 415C  | PINMUX15      | Pin Multiplexing Control 15 Register     | Privileged mode |
| 0x01C1 4160  | PINMUX16      | Pin Multiplexing Control 16 Register     | Privileged mode |
| 0x01C1 4164  | PINMUX17      | Pin Multiplexing Control 17 Register     | Privileged mode |
| 0x01C1 4168  | PINMUX18      | Pin Multiplexing Control 18 Register     | Privileged mode |
| 0x01C1 416C  | PINMUX19      | Pin Multiplexing Control 19 Register     | Privileged mode |
| 0x01C1 4170  | SUSPSRC       | Suspend Source Register                  | Privileged mode |
| 0x01C1 4174  | CHIPSIG       | Chip Signal Register                     | —               |
| 0x01C1 4178  | CHIPSIG_CLR   | Chip Signal Clear Register               | —               |
| 0x01C1 417C  | CFGCHIP0      | Chip Configuration 0 Register            | Privileged mode |
| 0x01C1 4180  | CFGCHIP1      | Chip Configuration 1 Register            | Privileged mode |
| 0x01C1 4184  | CFGCHIP2      | Chip Configuration 2 Register            | Privileged mode |

**Table 4-1. System Configuration (SYSCFG) Module Register Access (continued)**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION          | ACCESS          |
|--------------|---------------|-------------------------------|-----------------|
| 0x01C1 4188  | CFGCHIP3      | Chip Configuration 3 Register | Privileged mode |
| 0x01C1 418C  | CFGCHIP4      | Chip Configuration 4 Register | Privileged mode |

### 4.3 Pullup/Pulldown Resistors

Proper board design should ensure that input pins to the device always be at a valid logic level and not floating. This may be achieved via pullup/pulldown resistors. The device features internal pullup (IPU) and internal pulldown (IPD) resistors on most pins to eliminate the need, unless otherwise noted, for external pullup/pulldown resistors.

An external pullup/pulldown resistor needs to be used in the following situations:

- **Boot and Configuration Pins:** If the pin is both routed out and 3-stated (not driven), an external pullup/pulldown resistor is strongly recommended, even if the IPU/IPD matches the desired value/state.
- **Other Input Pins:** If the IPU/IPD does not match the desired value/state, use an external pullup/pulldown resistor to pull the signal to the opposite rail.

For the boot and configuration pins, if they are both routed out and 3-stated (not driven), it is strongly recommended that an external pullup/pulldown resistor be implemented. Although, internal pullup/pulldown resistors exist on these pins and they may match the desired configuration value, providing external connectivity can help ensure that valid logic levels are latched on these device boot and configuration pins. In addition, applying external pullup/pulldown resistors on the boot and configuration pins adds convenience to the user in debugging and flexibility in switching operating modes.

Tips for choosing an external pullup/pulldown resistor:

- Consider the total amount of current that may pass through the pullup or pulldown resistor. Make sure to include the leakage currents of all the devices connected to the net, as well as any internal pullup or pulldown resistors.
- Decide a target value for the net. For a pulldown resistor, this should be below the lowest  $V_{IL}$  level of all inputs connected to the net. For a pullup resistor, this should be above the highest  $V_{IH}$  level of all inputs on the net. A reasonable choice would be to target the  $V_{OL}$  or  $V_{OH}$  levels for the logic family of the limiting device; which, by definition, have margin to the  $V_{IL}$  and  $V_{IH}$  levels.
- Select a pullup/pulldown resistor with the largest possible value; but, which can still ensure that the net will reach the target pulled value when maximum current from all devices on the net is flowing through the resistor. The current to be considered includes leakage current plus, any other internal and external pullup/pulldown resistors on the net.
- For bidirectional nets, there is an additional consideration which sets a lower limit on the resistance value of the external resistor. Verify that the resistance is small enough that the weakest output buffer can drive the net to the opposite logic level (including margin).
- Remember to include tolerances when selecting the resistor value.
- For pullup resistors, also remember to include tolerances on the IO supply rail.
- For most systems, a 1-k $\Omega$  resistor can be used to oppose the IPU/IPD while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.
- For most systems, a 20-k $\Omega$  resistor can be used to compliment the IPU/IPD on the boot and configuration pins while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.
- For more detailed information on input current ( $I_I$ ), and the low-/high-level input voltages ( $V_{IL}$  and  $V_{IH}$ ) for the device, see [Section 5.3](#), Recommended Operating Conditions.
- For the internal pullup/pulldown resistors for all device pins, see the peripheral/system-specific terminal functions table.

## 5 Device Operating Conditions

### 5.1 Absolute Maximum Ratings Over Operating Junction Temperature Range (Unless Otherwise Noted) <sup>(1)</sup>

|  |  |   |
|--|--|---|
| Supply voltage ranges                                    | Core<br>(CVDD, RVDD, PLL0_VDDA) <sup>(2)</sup>   | -0.5 V to 1.4 V                                   |
|  | I/O, 3.3V<br>(DVDD) <sup>(2)</sup>   | -0.5 V to 3.8V                                    |
| Input voltage ranges                                     | V <sub>I</sub> I/O, 1.2V<br>(OSCIN)  | -0.3 V to CVDD + 0.35V                            |
|  | V <sub>I</sub> I/O, 3.3V<br>(Steady State)   | -0.3V to DVDD + 0.3V                              |
|  | V <sub>I</sub> I/O, 3.3V<br>(Transient Overshoot/Undershoot)   | 20% of DVDD for up to<br>20% of the signal period |
| Output voltage ranges                                    | V <sub>O</sub> I/O, 3.3V<br>(Steady State)   | -0.5 V to DVDD + 0.3V                             |
|  | V <sub>O</sub> I/O, 3.3V<br>(Transient Overshoot/Undershoot)   | 20% of DVDD for up to<br>20% of the signal period |
| Clamp Current  | Input or Output Voltages 0.3V above or below their respective power rails. Limit clamp current that flows through the I/O's internal diode protection cells. | ±20mA   |
| Operating Junction Temperature ranges,<br>T <sub>J</sub> | (default)  | 0°C to 90°C                                       |
|  | (T version)  | -40°C to 125°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) All voltage values are with respect to V<sub>SS</sub>, PLL0\_VSSA, OSCVSS

### 5.2 Handling Ratings

|  |   |            | UNIT |
|--|---|------------|------|
| Storage temperature range, T <sub>stg</sub>            | (default)                                 | -55 to 150 | °C   |
| ESD Stress Voltage,<br>V <sub>ESD</sub> <sup>(1)</sup> | Human Body Model (HBM) <sup>(2)</sup>     | >2000      | V    |
|  | Charged Device Model (CDM) <sup>(3)</sup> | >500       | V    |

- (1) Electrostatic discharge (ESD) to measure device sensitivity/immunity to damage caused by electrostatic discharges into the device.
- (2) Level listed above is the passing level per ANSI/ESDA/JEDEC JS-001-2010. JEDEC document JEP155 states that 500V HBM allows safe manufacturing with a standard ESD control process, and manufacturing with less than 500V HBM is possible if necessary precautions are taken. Pins listed as 1000V may actually have higher performance.
- (3) Level listed above is the passing level per EIA-JEDEC JESD22-C101E. JEDEC document JEP157 states that 250V CDM allows safe manufacturing with a standard ESD control process. Pins listed as 250V may actually have higher performance.

### 5.3 Recommended Operating Conditions

|                        |   | MIN                   | NOM | MAX                        | UNIT |
|------------------------|---|-----------------------|-----|----------------------------|------|
| CVDD                   | Supply voltage, Core (CVDD, PLL0_VDDA)  | 1.14                  | 1.2 | 1.32                       | V    |
| RVDD                   | Supply Voltage, Internal RAM  | 1.14                  | 1.2 | 1.32                       | V    |
| DVDD                   | Supply voltage, I/O, 3.3V (DVDD)  | 3.0                   | 3.3 | 3.45                       | V    |
| V <sub>SS</sub>        | Supply ground (V <sub>SS</sub> , PLL0_VSSA, OSCVSS <sup>(1)</sup> )                               | 0                     | 0   | 0                          | V    |
| V <sub>IH</sub>        | High-level input voltage, I/O, 3.3V   | 2                     |     |                            | V    |
|                        | High-level input voltage, OSCIN   | 0.7*CVDD              |     |                            | V    |
| V <sub>IL</sub>        | Low-level input voltage, I/O, 3.3V  |                       |     | 0.8                        | V    |
|                        | Low-level input voltage, OSCIN  |                       |     | 0.3*CVDD                   | V    |
| V <sub>HYS</sub>       | Input Hysteresis  |                       | 160 |                            | mV   |
| t <sub>t</sub>         | Transition time, 10%-90%, All Inputs (unless otherwise specified in the electrical data sections) |                       |     | 0.25P or 10 <sup>(2)</sup> | ns   |
| T <sub>J</sub>         | Operating junction temperature range  | Default               |     | 90                         | °C   |
|                        |   | Automotive (T suffix) | -40 | 125                        | °C   |
| F <sub>SYSCLK1,6</sub> | DSP Operating Frequency (SYSCLK1,6)   | Default               |     | 375 or 200                 | MHz  |
|                        |   | Automotive (T suffix) | 0   | 375 or 200                 | MHz  |

- (1) When an external crystal is used oscillator (OSC\_VSS) ground must be kept separate from other grounds and connected directly to the crystal load capacitor ground. These pins are shorted to V<sub>SS</sub> on the device itself and should not be connected to V<sub>SS</sub> on the circuit board. If a crystal is not used and the clock input is driven directly, then the oscillator V<sub>SS</sub> may be connected to board ground.
- (2) Whichever is smaller. P = the period of the applied signal. Maintaining transition times as fast as possible is recommended to improve noise immunity on input signals.

## 5.4 Notes on Recommended Power-On Hours (POH)

The information in the section below is provided solely for your convenience and does not extend or modify the warranty provided under TI's standard terms and conditions for TI semiconductor products.

To avoid significant degradation, the device power-on hours (POH) must be limited to the following:

**Table 5-1. Recommended Power-On Hours**

| SILICON REVISION | SPEED GRADE | OPERATING JUNCTION TEMPERATURE (Tj) | NOMINAL CVDD VOLTAGE (V) | POWER-ON HOURS [POH] (HOURS) |
|------------------|-------------|-------------------------------------|--------------------------|------------------------------|
| D                | 300 MHZ     | 0 to 90 °C                          | 1.2V                     | 100,000                      |
| D                | 200 MHZ     | 0 to 90 °C                          | 1.2V                     | 100,000                      |
| D                | 200 MHZ     | -40 to 125 °C                       | 1.2V                     | 20,000                       |
| D                | 375 MHz     | 0 to 90 °C                          | 1.2V                     | 100,000                      |
| D                | 375 MHZ     | -40 to 125 °C                       | 1.2V                     | 20,000                       |

**Note:** Logic functions and parameter values are not assured out of the range specified in the recommended operating conditions.

**The above notations cannot be deemed a warranty or deemed to extend or modify the warranty under TI's standard terms and conditions for TI semiconductor products.**

## 5.5 Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Junction Temperature (Unless Otherwise Noted)

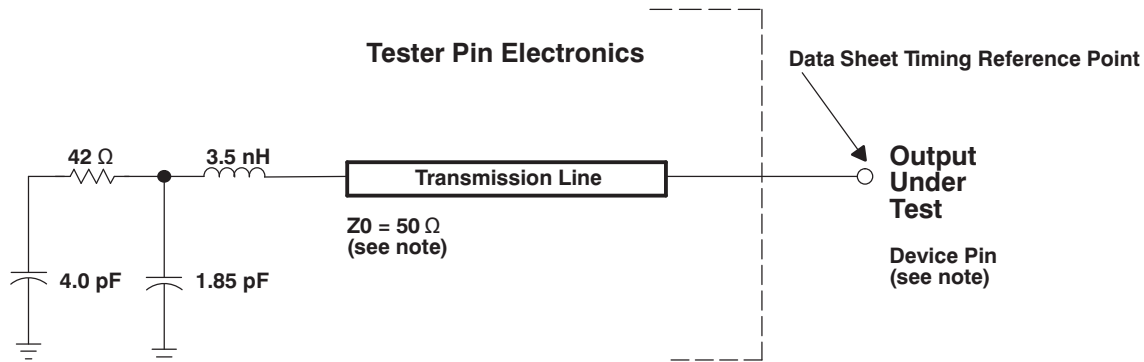
| PARAMETER                      |                                      | TEST CONDITIONS  | MIN  | TYP | MAX  | UNIT |
|--------------------------------|--------------------------------------|--|------|-----|------|------|
| V <sub>OH</sub>                | High-level output voltage (3.3V I/O) | DVDD= 3.15V, I <sub>OH</sub> = 4 mA  | 2.4  |     |      | V    |
|                                |                                      | DVDD= 3.15V, I <sub>OH</sub> = 100 μA  | 2.95 |     |      | V    |
| V <sub>OL</sub>                | Low-level output voltage (3.3V I/O)  | DVDD= 3.15V, I <sub>OL</sub> = -4mA  |      |     | 0.4  | V    |
|                                |                                      | DVDD= 3.15V, I <sub>OL</sub> = -100 μA   |      |     | 0.2  | V    |
| I <sub>I</sub> <sup>(1)</sup>  | Input current                        | V <sub>I</sub> = V <sub>SS</sub> to DVDD without opposing internal resistor                      |      |     | ±35  | μA   |
|                                |                                      | V <sub>I</sub> = V <sub>SS</sub> to DVDD with opposing internal pullup resistor <sup>(2)</sup>   | -30  |     | -200 | μA   |
|                                |                                      | V <sub>I</sub> = V <sub>SS</sub> to DVDD with opposing internal pulldown resistor <sup>(2)</sup> | 50   |     | 300  | μA   |
| I <sub>OH</sub>                | High-level output current            | All peripherals  |      |     | -4   | mA   |
| I <sub>OL</sub>                | Low-level output current             | All peripherals  |      |     | 4    | mA   |
| I <sub>OZ</sub> <sup>(3)</sup> | I/O Off-state output current         | VO = VDD or V <sub>SS</sub> ; Internal pull disabled   |      |     | ±35  | μA   |
| C <sub>I</sub>                 | Input capacitance                    | LVC MOS signals  |      |     | 3    | pF   |
|                                |                                      | OSCIN  |      |     | 2    | pF   |
| C <sub>O</sub>                 | Output capacitance                   | LVC MOS signals  |      |     | 3    | pF   |

- (1) I<sub>I</sub> applies to input-only pins and bi-directional pins. For input-only pins, I<sub>I</sub> indicates the input leakage current. For bi-directional pins, I<sub>I</sub> indicates the input leakage current and off-state (Hi-Z) output leakage current.
- (2) Applies only to pins with an internal pullup (IPU) or pulldown (IPD) resistor.
- (3) I<sub>OZ</sub> applies to output-only pins, indicating off-state (Hi-Z) output leakage current.

## 6 Peripheral Information and Electrical Specifications

### 6.1 Parameter Information

#### 6.1.1 Parameter Information Device-Specific Information



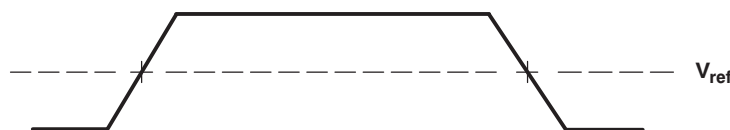
- A. The data sheet provides timing at the device pin. For output timing analysis, the tester pin electronics and its transmission line effects must be taken into account. A transmission line with a delay of 2 ns or longer can be used to produce the desired transmission line effect. The transmission line is intended as a load only. It is not necessary to add or subtract the transmission line delay (2 ns or longer) from the data sheet timings. Input requirements in this data sheet are tested with an input slew rate of < 4 Volts per nanosecond (4 V/ns) at the device pin and the input signals are driven between 0V and the appropriate IO supply rail for the signal.

**Figure 6-1. Test Load Circuit for AC Timing Measurements**

The load capacitance value stated is only for characterization and measurement of AC timing signals. This load capacitance value does not indicate the maximum load the device is capable of driving.

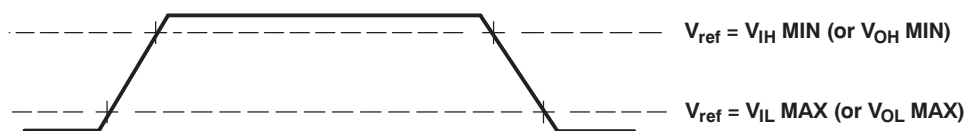
#### 6.1.1.1 Signal Transition Levels

All input and output timing parameters are referenced to  $V_{ref}$  for both "0" and "1" logic levels. For 3.3 V I/O,  $V_{ref} = 1.65$  V. For 1.2 V I/O,  $V_{ref} = 0.6$  V.



**Figure 6-2. Input and Output Voltage Reference Levels for AC Timing Measurements**

All rise and fall transition timing parameters are referenced to  $V_{IL\ MAX}$  and  $V_{IH\ MIN}$  for input clocks,  $V_{OL\ MAX}$  and  $V_{OH\ MIN}$  for output clocks.



**Figure 6-3. Rise and Fall Transition Time Voltage Reference Levels**

## 6.2 Recommended Clock and Control Signal Transition Behavior

All clocks and control signals **must** transition between  $V_{IH}$  and  $V_{IL}$  (or between  $V_{IL}$  and  $V_{IH}$ ) in a monotonic manner.

## 6.3 Power Supplies

### 6.3.1 Power-On Sequence

The device should be powered-on in the following order:

1. Logic supplies:
  - a. CVDD core logic supply
  - b. Other static logic supplies (RVDD, PLL0\_VDDA).

Groups 1a) and 1b) may be powered up together; or 1a) first, followed by 1b).

2. All digital IO supplies (DVDD).

There is no specific required voltage ramp rate for any of the supplies.

**Note:** Future devices may support higher performance at a higher core logic voltage (CVDD). If future migration is desired, the current design should provide separate supplies for 1a) and 1b). If not, then 1a) and 1b) may be provided by a single supply.

RESET must be maintained active until all power supplies have reached their nominal values.

### 6.3.2 Power-Off Sequence

The power supplies can be powered-off in any order as long as the 3.3V supplies do not remain powered with the other supplies unpowered.

## 6.4 Reset

### 6.4.1 Power-On Reset (POR)

A power-on reset (POR) is required to place the device in a known good state after power-up. Power-On Reset is initiated by bringing  $\overline{\text{RESET}}$  and  $\overline{\text{TRST}}$  low at the same time. POR sets all of the device internal logic to its default state. All pins are tri-stated with the exception of GP7[14]. During reset, GP7[14] is configured as a reserved function, and its behavior is not guaranteed; the user should be aware that this pin will drive a level, and in fact may toggle, during reset.

While both  $\overline{\text{TRST}}$  and  $\overline{\text{RESET}}$  need to be asserted upon power up, only  $\overline{\text{RESET}}$  needs to be released for the device to boot properly.  $\overline{\text{TRST}}$  may be asserted indefinitely for normal operation, keeping the JTAG port interface and device's emulation logic in the reset state.

$\overline{\text{TRST}}$  only needs to be released when it is necessary to use a JTAG controller to debug the device or exercise the device's boundary scan functionality. Note:  $\overline{\text{TRST}}$  is synchronous and must be clocked by TCK; otherwise, the boundary scan logic may not respond as expected after  $\overline{\text{TRST}}$  is asserted.

$\overline{\text{RESET}}$  must be released only in order for boundary-scan JTAG to read the variant field of IDCODE correctly. Other boundary-scan instructions work correctly independent of current state of  $\overline{\text{RESET}}$ . For maximum reliability, the device includes an internal pulldown on the  $\overline{\text{TRST}}$  pin to ensure that  $\overline{\text{TRST}}$  will always be asserted upon power up and the device's internal emulation logic will always be properly initialized.

JTAG controllers from Texas Instruments actively drive  $\overline{\text{TRST}}$  high. However, some third-party JTAG controllers may not drive  $\overline{\text{TRST}}$  high but expect the use of a pullup resistor on  $\overline{\text{TRST}}$ . When using this type of JTAG controller, assert  $\overline{\text{TRST}}$  to initialize the device after powerup and externally drive  $\overline{\text{TRST}}$  high before attempting any emulation or boundary scan operations.

A summary of the effects of Power-On Reset is given below:

- All internal logic (including emulation logic and the PLL logic) is reset to its default state
- Internal memory is not maintained through a POR
- All device pins go to a high-impedance state

#### CAUTION

**A watchdog reset triggers a POR.**

### 6.4.2 Warm Reset

A warm reset provides a limited reset to the device. Warm Reset is initiated by bringing only  $\overline{\text{RESET}}$  low ( $\overline{\text{TRST}}$  is maintained high through a warm reset). Warm reset sets certain portions of the device to their default state while leaving others unaltered. All pins are tri-stated with the exception of GP7[14]. During reset, GP7[14] is configured as a reserved function, and its behavior is not deterministic; the user should be aware that this pin will drive a level, and in fact may toggle, during reset.

During emulation, the emulator will maintain  $\overline{\text{TRST}}$  high so only warm reset (not POR) is available during emulation debug and development.

A summary of the effects of Warm Reset is given below:

- All internal logic (except for the emulation logic and the PLL logic) is reset to its default state
- Internal memory is maintained through a warm reset
- All device pins go to a high-impedance state

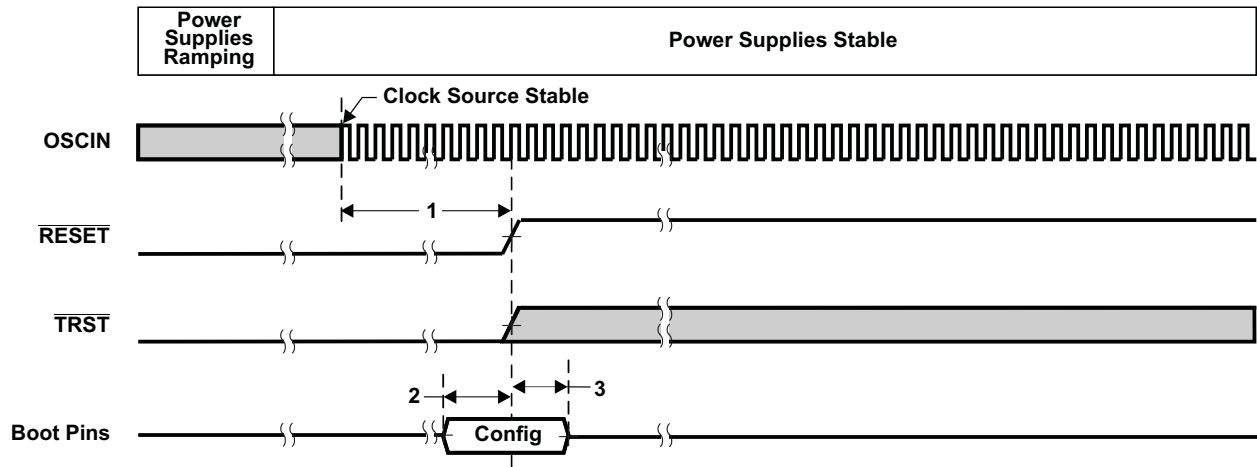
### 6.4.3 Reset Electrical Data Timings

Table 6-1 assumes testing over the recommended operating conditions.

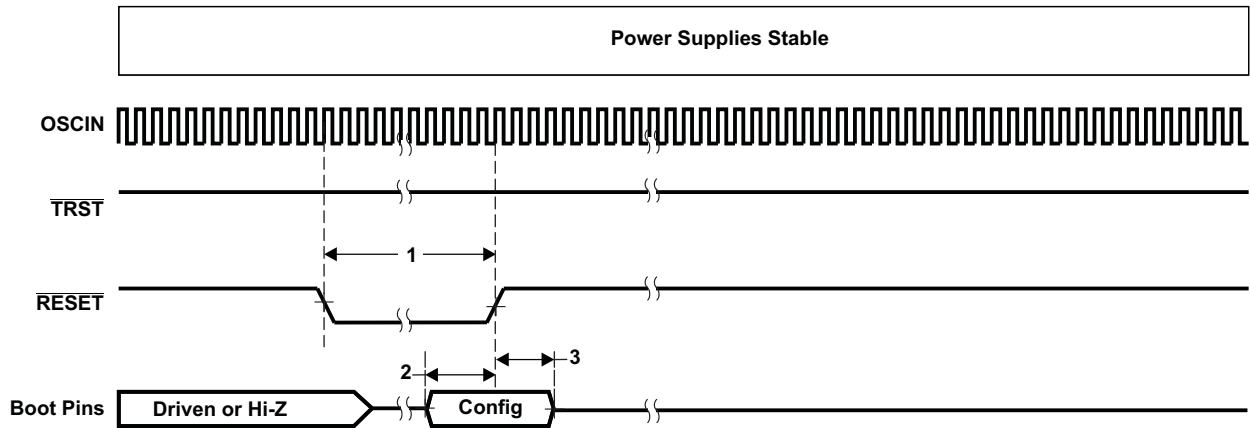
**Table 6-1. Reset Timing Requirements<sup>(1)</sup>**

| NO. |                    |  | MIN | MAX | UNIT |
|-----|--------------------|--|-----|-----|------|
| 1   | $t_{w(RSTL)}$      | Pulse width, $\overline{RESET}/\overline{TRST}$ low                        | 100 |     | ns   |
| 2   | $t_{su(BPV-RSTH)}$ | Setup time, boot pins valid before $\overline{RESET}/\overline{TRST}$ high | 20  |     | ns   |
| 3   | $t_{h(RSTH-BPV)}$  | Hold time, boot pins valid after $\overline{RESET}/\overline{TRST}$ high   | 20  |     | ns   |

(1) For power-on reset (POR), the reset timings in this table refer to  $\overline{RESET}$  and  $\overline{TRST}$  together. For warm reset, the reset timings in this table refer to  $\overline{RESET}$  only ( $\overline{TRST}$  is held high).



**Figure 6-4. Power-On Reset (RESET and TRST active) Timing**



**Figure 6-5. Warm Reset (RESET active, TRST high) Timing**

## 6.5 Crystal Oscillator or External Clock Input

The device includes two choices to provide an external clock input, which is fed to the on-chip PLL to generate high-frequency system clocks. These options are illustrated in Figure 6-6 and Figure 6-7. For input clock frequencies between 12 and 20 MHz, a crystal with 80 ohm max ESR is recommended. For input clock frequencies between 20 and 30 MHz, a crystal with 60 ohm max ESR is recommended. Typical load capacitance values are 10-20 pF, where the load capacitance is the series combination of C1 and C2.

The CLKMODE bit in the PLLCTL register must be 0 to use the on-chip oscillator. If CLKMODE is set to 1, the internal oscillator is disabled.

- Figure 6-6 illustrates the option that uses on-chip 1.2V oscillator with external crystal circuit.
- Figure 6-7 illustrates the option that uses an external 1.2V clock input.

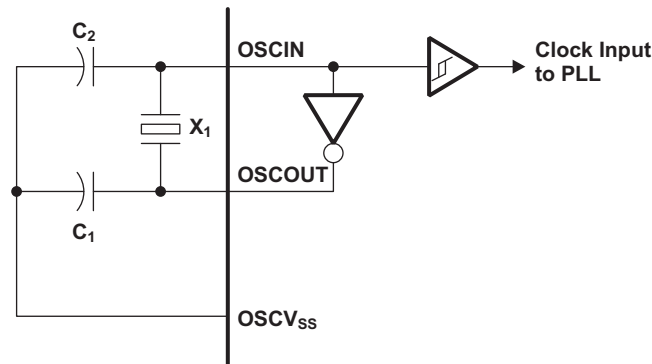


Figure 6-6. On-Chip 1.2V Oscillator

Table 6-2. Oscillator Timing Requirements

| NO               | PARAMETER                                 | MIN | MAX | UNIT |
|------------------|---|-----|-----|------|
| f <sub>osc</sub> | Oscillator frequency range (OSCIN/OSCOUT) | 12  | 30  | MHz  |

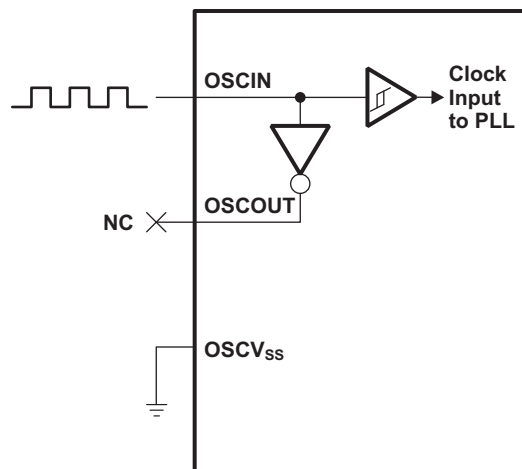


Figure 6-7. External 1.2V Clock Source

**Table 6-3. OSCIN Timing Requirements for Externally Driven Clock**

| NO                            |  | MIN                              | MAX                   | UNIT |
|-------------------------------|--|----------------------------------|-----------------------|------|
| $f_{\text{OSCIN}}$            | OSCIN frequency range (OSCIN)              | 12                               | 50                    | MHz  |
| $t_{\text{c}}(\text{OSCIN})$  | Cycle time, external clock driven on OSCIN | 20                               |                       | ns   |
| $t_{\text{w}}(\text{OSCINH})$ | Pulse width high, external clock on OSCIN  | $0.4 t_{\text{c}}(\text{OSCIN})$ |                       | ns   |
| $t_{\text{w}}(\text{OSCINL})$ | Pulse width low, external clock on OSCIN   | $0.4 t_{\text{c}}(\text{OSCIN})$ |                       | ns   |
| $t_{\text{t}}(\text{OSCIN})$  | Transition time, OSCIN                     |                                  | $0.25P$ or $10^{(1)}$ | ns   |
| $t_{\text{j}}(\text{OSCIN})$  | Period jitter, OSCIN                       |                                  | $0.02P$               | ns   |

(1) Whichever is smaller. P = the period of the applied signal. Maintaining transition times as fast as possible is recommended to improve noise immunity on input signals.

## 6.6 Clock PLLs

The device has one PLL controller that provides clock to different parts of the system. PLL0 provides clocks (though various dividers) to most of the components of the device.

The PLL controller provides the following:

- Glitch-Free Transitions (on changing clock settings)
- Domain Clocks Alignment
- Clock Gating
- PLL power down

The various clock outputs given by the controller are as follows:

- Domain Clocks: SYSCLK [1:n]
- Auxiliary Clock from reference clock source: AUXCLK

Various dividers that can be used are as follows:

- Post-PLL Divider: POSTDIV
- SYSCLK Divider: D1,  $\frac{1}{4}$ , Dn

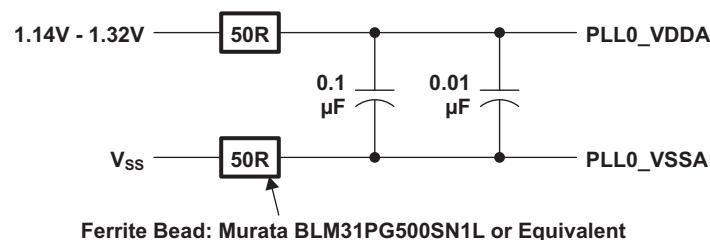
Various other controls supported are as follows:

- PLL Multiplier Control: PLLM
- Software programmable PLL Bypass: PLEN

### 6.6.1 PLL Device-Specific Information

The device DSP generates the high-frequency internal clocks it requires through an on-chip PLL.

The PLL requires some external filtering components to reduce power supply noise as shown in [Figure 6-8](#).



**Figure 6-8. PLL External Filtering Components**

The input to the PLL is either from the on-chip oscillator or from an external clock on the OSCIN pin. The PLL outputs seven clocks that have programmable divider options. [Figure 6-9](#) illustrates the PLL Topology.

The PLL is disabled by default after a device reset. It must be configured by software according to the allowable operating conditions listed in Table 6-4 before enabling the DSP to run from the PLL by setting PLEN = 1.

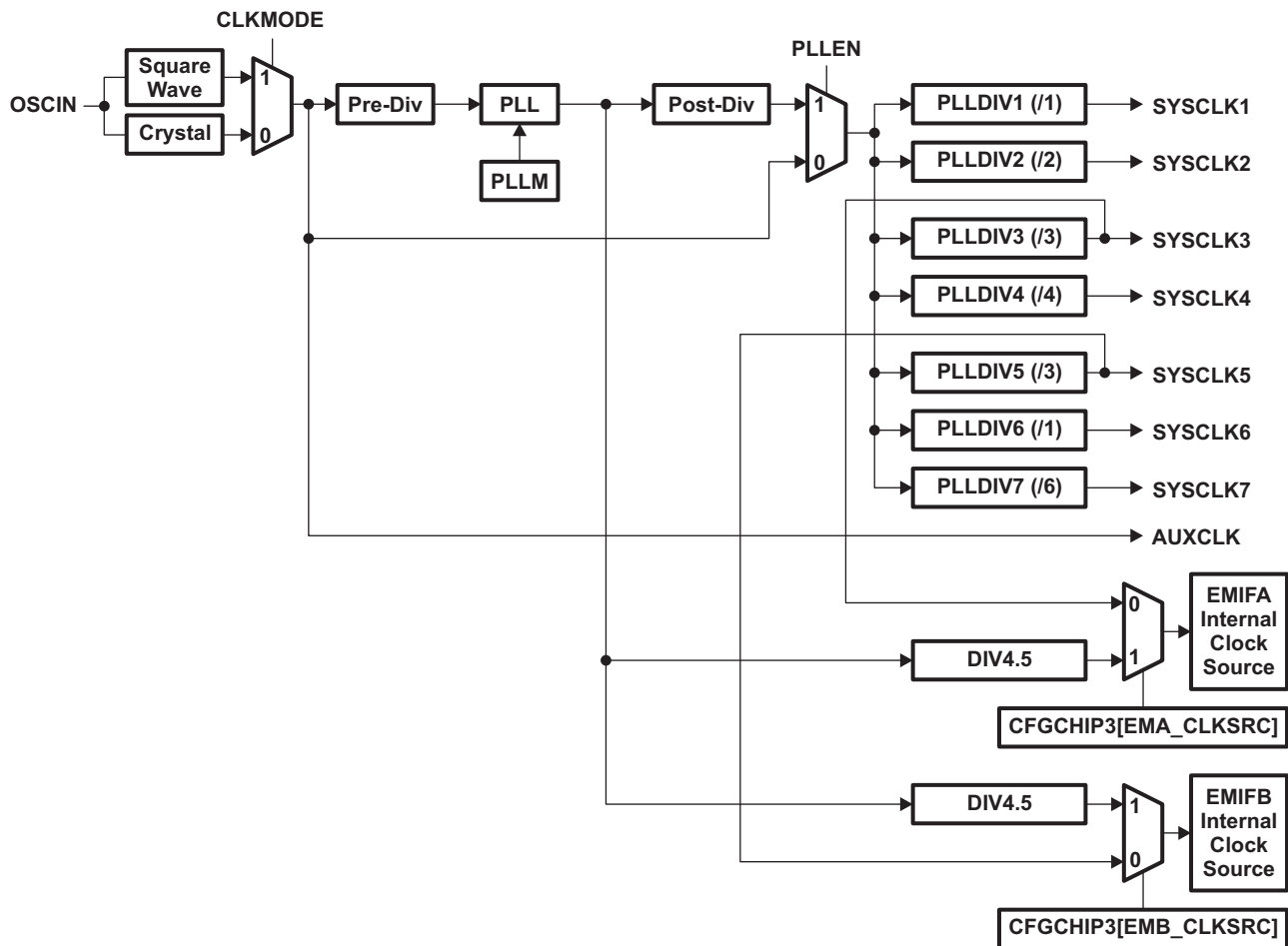


Figure 6-9. PLL Topology

**Table 6-4. Allowed PLL Operating Conditions**

| NO | PARAMETER  | Default Value | MIN  | MAX  | UNIT         |
|----|--|---------------|------|--|--------------|
| 1  | PLLRST: Assertion time during initialization   | N/A           | 1000 | N/A  | ns           |
| 2  | Lock time: The time that the application has to wait for the PLL to acquire locks before setting PLEN, after changing PREDIV, PLLM, or OSCIN | N/A           | N/A  | $\text{Max PLL Lock Time} = \frac{2000 N}{\sqrt{m}}$ where N = Pre-Divider Ratio<br>M = PLL Multiplier | OSCIN cycles |
| 3  | PREDIV   | /1            | /1   | /32  |              |
| 4  | PLL input frequency ( PLLREF)  |               | 12   | 30 (if internal oscillator is used)<br>50 (if external clock source is used)                           | MHz          |
| 5  | PLL multiplier values ( PLLM) <sup>(1)</sup>   | x20           | x4   | x32  |              |
| 6  | PLL output frequency. ( PLOUT )  | N/A           | 300  | 600  | MHz          |
| 7  | POSTDIV  | /1            | /1   | /32  |              |

(1) The multiplier values must be chosen such that the PLL output frequency (at PLOUT) is between 300 and 600 MHz, but the frequency going into the SYSCLK dividers (after the post divider) cannot exceed the maximum clock frequency defined for the device at a given voltage operating point.

### 6.6.2 Device Clock Generation

PLL0 is controlled by PLL Controller 0. The PLLC0 manages the clock ratios, alignment, and gating for the system clocks to the chip. The PLLC is responsible for controlling all modes of the PLL through software, in terms of pre-division of the clock inputs, multiply factor within the PLL, and post-division for each of the chip-level clocks from the PLL output. The PLLC also controls reset propagation through the chip, clock alignment, and test points.

PLLC0 generates several clocks from the PLL0 output clock for use by the various modules. These are summarized in [Table 6-5](#). The clock ratios between SYSCLK1, SYSCLK2, SYSCLK4 and SYSCLK6 must always be maintained as shown in the table.

**Table 6-5. System PLLC0 Output Clocks**

| OUTPUT CLOCK | USED BY  | DEFAULT RATIO (RELATIVE TO SYSCLK1) | NOTES  |
|--------------|--|-------------------------------------|--|
| SYSCLK1      | DSP  | /1                                  | No Required Ratio                              |
| SYSCLK2      | EDMA, DSP ports, EMIFB (ports to switch fabric), ECAP 0/1/2, EPWM 0/1/2, EQEP 0/1, McASP/FIFO 0/1, UART 2, HRPWM 0/1/2, SPI0 | /2                                  | SYSCLK1 / 2                                    |
| SYSCLK3      | EMIFA  | /3                                  | No Required Ratio                              |
| SYSCLK4      | SYSCFG, Interrupt Controller, PLLC0, PSC 0, EMAC/MDIO, GPIO, I2C 1, PSC 1  | /4                                  | SYSCLK1 / 4                                    |
| SYSCLK5      | EMIFB  | /3                                  | No Required Ratio                              |
| SYSCLK7      | RMII clock to EMAC   | /6                                  | No Required Ratio ;<br>Should be set to 50 MHz |
| AUXCLK       | McASP AuxClk, Timer64P0, Timer64P1, I2C0,  | N/A                                 | No Required Ratio                              |
| DIV4p5       | 133MHz clock source for EMIFB  | PLL output/4.5                      | No Required Ratio                              |

- The divide values in the PLL Controller 0 for SYSCLK1/SYSCLK6, SYSCLK2 and SYSCLK4 are not fixed so that user can change the divide values for power saving reasons. But users are responsible to guarantee that the divide ratios between these clock domains must be fixed to 1:2:4.
- Although the PLL is capable of running at 600 MHz, the SYSCLK dividers in the PLLC0 are not (maximum 400 MHz). For this reason, the post-divider in the PLLC0 should be configured for /2 to provide 300 MHz to each of the SYSCLK dividers.
- The DIV4p5 (/4.5) hardware clock divider is provided to generate 133 MHz from the 600 MHz PLL clock for use as clocks to the EMIFs.

### 6.6.3 PLL Controller 0 Registers

**Table 6-6. PLL Controller 0 Registers**

| ADDRESS     | REGISTER NAME | REGISTER DESCRIPTION                        |
|-------------|---------------|---|
| 0x01C1 1000 | REVID         | Revision Identification Register            |
| 0x01C1 10E4 | RSTYPE        | Reset Type Status Register                  |
| 0x01C1 1100 | PLLCTL        | PLL Control Register                        |
| 0x01C1 1104 | -             | Reserved                                    |
| 0x01C1 1110 | PLLM          | PLL Multiplier Control Register             |
| 0x01C1 1114 | PREDIV        | PLL Pre-Divider Control Register            |
| 0x01C1 1118 | PLLDIV1       | PLL Controller Divider 1 Register           |
| 0x01C1 111C | PLLDIV2       | PLL Controller Divider 2 Register           |
| 0x01C1 1120 | PLLDIV3       | PLL Controller Divider 3 Register           |
| 0x01C1 1124 | -             | Reserved                                    |
| 0x01C1 1128 | POSTDIV       | PLL Post-Divider Control Register           |
| 0x01C1 1138 | PLLCMD        | PLL Controller Command Register             |
| 0x01C1 113C | PLLSTAT       | PLL Controller Status Register              |
| 0x01C1 1140 | ALNCTL        | PLL Controller Clock Align Control Register |
| 0x01C1 1144 | DCHANGE       | PLLDIV Ratio Change Status Register         |
| 0x01C1 1148 | CKEN          | Clock Enable Control Register               |
| 0x01C1 114C | CKSTAT        | Clock Status Register                       |
| 0x01C1 1150 | SYSTAT        | SYSCLK Status Register                      |
| 0x01C1 1160 | PLLDIV4       | PLL Controller Divider 4 Register           |
| 0x01C1 1164 | PLLDIV5       | PLL Controller Divider 5 Register           |
| 0x01C1 1168 | PLLDIV6       | PLL Controller Divider 6 Register           |
| 0x01C1 116C | PLLDIV7       | PLL Controller Divider 7 Register           |

## 6.7 DSP Interrupts

The C674x DSP interrupt controller combines device events into 12 prioritized interrupts. The source for each of the 12 CPU interrupts is user programmable and is listed in [Table 6-7](#). Also, the interrupt controller controls the generation of the CPU exception, NMI, and emulation interrupts. [Table 6-8](#) summarizes the C674x interrupt controller registers and memory locations.

**Table 6-7. C6743 DSP Interrupts**

| EVT#  | INTERRUPT NAME  | SOURCE                                    |
|-------|-----------------|---|
| 0     | EVT0            | C674x Int Ctl 0                           |
| 1     | EVT1            | C674x Int Ctl 1                           |
| 2     | EVT2            | C674x Int Ctl 2                           |
| 3     | EVT3            | C674x Int Ctl 3                           |
| 4     | T64P0_TINT12    | Timer64P0 - TINT12                        |
| 5     | SYSCFG_CHIPINT2 | SYSCFG_CHIPSIG Register                   |
| 6     | PRU_EVTOUT0     | PRU Interrupt                             |
| 7     | EHRPWM0         | HiResTimer/PWM0 Interrupt                 |
| 8     | EDMA3_CC0_INT1  | EDMA3 CC0 Region 1 interrupt              |
| 9     | EMU-DTDMA       | C674x-ECM                                 |
| 10    | EHRPWM0TZ       | HiResTimer/PWM0 Trip Zone Interrupt       |
| 11    | EMU-RTDXRX      | C674x-RTDX                                |
| 12    | EMU-RTDXTX      | C674x-RTDX                                |
| 13    | IDMAINT0        | C674x-EMC                                 |
| 14    | IDMAINT1        | C674x-EMC                                 |
| 15    | MMCSDB_INT0     | MMCSDB MMC/SD Interrupt                   |
| 16    | MMCSDB_INT1     | MMCSDB SDIO Interrupt                     |
| 17    | PRU_EVTOUT1     | PRU Interrupt                             |
| 18    | EHRPWM1         | HiResTimer/PWM1 Interrupt                 |
| 19-21 | -               | Reserved                                  |
| 22    | PRU_EVTOUT2     | PRU Interrupt                             |
| 23    | EHRPWM1TZ       | HiResTimer/PWM1 Trip Zone Interrupt       |
| 24    | EHRPWM2         | HiResTimer/PWM2 Interrupt                 |
| 25    | EHRPWM2TZ       | HiResTimer/PWM2 Trip Zone Interrupt       |
| 26    | EMAC_C0RXTHRESH | EMAC - Core 0 Receive Threshold Interrupt |
| 27    | EMAC_C0RX       | EMAC - Core 0 Receive Interrupt           |
| 28    | EMAC_C0TX       | EMAC - Core 0 Transmit Interrupt          |
| 29    | EMAC_C0MISC     | EMAC - Core 0 Miscellaneous Interrupt     |
| 30    | EMAC_C1RXTHRESH | EMAC - Core 1 Receive Threshold Interrupt |
| 31    | EMAC_C1RX       | EMAC - Core 1 Receive Interrupt           |
| 32    | EMAC_C1TX       | EMAC - Core 1 Transmit Interrupt          |
| 33    | EMAC_C1MISC     | EMAC - Core 1 Miscellaneous Interrupt     |
| 34    | -               | Reserved                                  |
| 35    | PRU_EVTOUT3     | PRU Interrupt                             |
| 36    | IIC0_INT        | I2C0                                      |
| 37    | SP0_INT         | SPI0                                      |
| 38    | UART0_INT       | UART0                                     |
| 39    | PRU_EVTOUT5     | PRU Interrupt                             |
| 40    | T64P1_TINT12    | Timer64P1 Interrupt 12                    |
| 41    | GPIO_B1INT      | GPIO Bank 1 Interrupt                     |
| 42    | IIC1_INT        | I2C1                                      |

**Table 6-7. C6743 DSP Interrupts (continued)**

| EVT#  | INTERRUPT NAME   | SOURCE   |
|-------|------------------|--|
| 43    | -                | Reserved   |
| 44    | PRU_EVTOUT6      | PRU Interrupt  |
| 45    | ECAP0            | ECAP0  |
| 46    | -                | Reserved   |
| 47    | ECAP1            | ECAP1  |
| 48    | T64P1_TINT34     | Timer64P1 Interrupt 34                                   |
| 49    | GPIO_B2INT       | GPIO Bank 2 Interrupt                                    |
| 50    | PRU_EVTOUT7      | PRU Interrupt  |
| 51    | ECAP2            | ECAP2  |
| 52    | GPIO_B3INT       | GPIO Bank 3 Interrupt                                    |
| 53    | EQEP1            | EQEP1  |
| 54    | GPIO_B4INT       | GPIO Bank 4 Interrupt                                    |
| 55    | EMIFA_INT        | EMIFA  |
| 56    | EDMA3_CC0_ERRINT | EDMA3 Channel Controller 0                               |
| 57    | EDMA3_TC0_ERRINT | EDMA3 Transfer Controller 0                              |
| 58    | EDMA3_TC1_ERRINT | EDMA3 Transfer Controller 1                              |
| 59    | GPIO_B5INT       | GPIO Bank 5 Interrupt                                    |
| 60    | EMIFB_INT        | EMIFB Memory Error Interrupt                             |
| 61    | MCASP_INT        | McASP0,1 Combined RX/TX Interrupts                       |
| 62    | GPIO_B6INT       | GPIO Bank 6 Interrupt                                    |
| 63    | -                | Reserved   |
| 64    | T64P0_TINT34     | Timer64P0 Interrupt 34                                   |
| 65    | GPIO_B0INT       | GPIO Bank 0 Interrupt                                    |
| 66    | PRU_EVTOUT4      | PRU Interrupt  |
| 67    | SYSCFG_CHIPINT3  | SYSCFG_CHIPSIG Register                                  |
| 68    | EQEP0            | EQEP0  |
| 69    | UART2_INT        | UART2  |
| 70    | PSC0_ALLINT      | PSC0   |
| 71    | PSC1_ALLINT      | PSC1   |
| 72    | GPIO_B7INT       | GPIO Bank 7 Interrupt                                    |
| 73    | -                | Reserved   |
| 74    | MPU_BOOTCFG_ERR  | Shared MPU and SYSCFG Address/Protection Error Interrupt |
| 75-77 | -                | Reserved   |
| 78    | T64P0_CMPINT0    | Timer64P0 - Compare 0                                    |
| 79    | T64P0_CMPINT1    | Timer64P0 - Compare 1                                    |
| 80    | T64P0_CMPINT2    | Timer64P0 - Compare 2                                    |
| 81    | T64P0_CMPINT3    | Timer64P0 - Compare 3                                    |
| 82    | T64P0_CMPINT4    | Timer64P0 - Compare 4                                    |
| 83    | T64P0_CMPINT5    | Timer64P0 - Compare 5                                    |
| 84    | T64P0_CMPINT6    | Timer64P0 - Compare 6                                    |
| 85    | T64P0_CMPINT7    | Timer64P0 - Compare 7                                    |
| 86    | T64P1_CMPINT0    | Timer64P1 - Compare 0                                    |
| 87    | T64P1_CMPINT1    | Timer64P1 - Compare 1                                    |
| 88    | T64P1_CMPINT2    | Timer64P1 - Compare 2                                    |
| 89    | T64P1_CMPINT3    | Timer64P1 - Compare 3                                    |
| 90    | T64P1_CMPINT4    | Timer64P1 - Compare 4                                    |
| 91    | T64P1_CMPINT5    | Timer64P1 - Compare 5                                    |

**Table 6-7. C6743 DSP Interrupts (continued)**

| <b>EVT#</b> | <b>INTERRUPT NAME</b> | <b>SOURCE</b>         |
|-------------|-----------------------|-----------------------|
| 92          | T64P1_CMPINT6         | Timer64P1 - Compare 6 |
| 93          | T64P1_CMPINT7         | Timer64P1 - Compare 7 |
| 94-95       | -                     | Reserved              |
| 96          | INTERR                | C674x-Int Ctl         |
| 97          | EMC_IDMAERR           | C674x-EMC             |
| 98-112      | -                     | Reserved              |
| 113         | PMC_ED                | C674x-PMC             |
| 114-115     | -                     | Reserved              |
| 116         | UMC_ED1               | C674x-UMC             |
| 117         | UMC_ED2               | C674x-UMC             |
| 118         | PDC_INT               | C674x-PDC             |
| 119         | SYS_CMPA              | C674x-SYS             |
| 120         | PMC_CMPA              | C674x-PMC             |
| 121         | PMC_CMPA              | C674x-PMC             |
| 122         | DMC_CMPA              | C674x-DMC             |
| 123         | DMC_CMPA              | C674x-DMC             |
| 124         | UMC_CMPA              | C674x-UMC             |
| 125         | UMC_CMPA              | C674x-UMC             |
| 126         | EMC_CMPA              | C674x-EMC             |
| 127         | EMC_BUSERR            | C674x-EMC             |

**Table 6-8. C674x DSP Interrupt Controller Registers**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION             |
|--------------|---------------|----------------------------------|
| 0x0180 0000  | EVTFLAG0      | Event flag register 0            |
| 0x0180 0004  | EVTFLAG1      | Event flag register 1            |
| 0x0180 0008  | EVTFLAG2      | Event flag register 2            |
| 0x0180 000C  | EVTFLAG3      | Event flag register 3            |
| 0x0180 0020  | EVTSET0       | Event set register 0             |
| 0x0180 0024  | EVTSET1       | Event set register 1             |
| 0x0180 0028  | EVTSET2       | Event set register 2             |
| 0x0180 002C  | EVTSET3       | Event set register 3             |
| 0x0180 0040  | EVTCLR0       | Event clear register 0           |
| 0x0180 0044  | EVTCLR1       | Event clear register 1           |
| 0x0180 0048  | EVTCLR2       | Event clear register 2           |
| 0x0180 004C  | EVTCLR3       | Event clear register 3           |
| 0x0180 0080  | EVTMASK0      | Event mask register 0            |
| 0x0180 0084  | EVTMASK1      | Event mask register 1            |
| 0x0180 0088  | EVTMASK2      | Event mask register 2            |
| 0x0180 008C  | EVTMASK3      | Event mask register 3            |
| 0x0180 00A0  | MEVTFLAG0     | Masked event flag register 0     |
| 0x0180 00A4  | MEVTFLAG1     | Masked event flag register 1     |
| 0x0180 00A8  | MEVTFLAG2     | Masked event flag register 2     |
| 0x0180 00AC  | MEVTFLAG3     | Masked event flag register 3     |
| 0x0180 00C0  | EXPMASK0      | Exception mask register 0        |
| 0x0180 00C4  | EXPMASK1      | Exception mask register 1        |
| 0x0180 00C8  | EXPMASK2      | Exception mask register 2        |
| 0x0180 00CC  | EXPMASK3      | Exception mask register 3        |
| 0x0180 00E0  | MEXPFLAG0     | Masked exception flag register 0 |
| 0x0180 00E4  | MEXPFLAG1     | Masked exception flag register 1 |
| 0x0180 00E8  | MEXPFLAG2     | Masked exception flag register 2 |
| 0x0180 00EC  | MEXPFLAG3     | Masked exception flag register 3 |

## 6.8 General-Purpose Input/Output (GPIO)

The GPIO peripheral provides general-purpose pins that can be configured as either inputs or outputs. When configured as an output, a write to an internal register can control the state driven on the output pin. When configured as an input, the state of the input is detectable by reading the state of an internal register. In addition, the GPIO peripheral can produce CPU interrupts and EDMA events in different interrupt/event generation modes. The GPIO peripheral provides generic connections to external devices. The GPIO pins are grouped into banks of 16 pins per bank (i.e., bank 0 consists of GPIO [0:15]). See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

The GPIO peripheral supports the following:

- Up to 128 Pins on ZKB and up to 109 Pins on PTP package configurable as GPIO
- External Interrupt and DMA request Capability
  - Every GPIO pin may be configured to generate an interrupt request on detection of rising and/or falling edges on the pin.
  - The interrupt requests within each bank are combined (logical or) to create eight unique bank level interrupt requests.
  - The bank level interrupt service routine may poll the INTSTATx register for its bank to determine which pin(s) have triggered the interrupt.
  - GPIO Banks 0, 1, 2, 3, 4, 5, 6, and 7 Interrupts assigned to DSP Events 65, 41, 49, 52, 54, 59, 62 and 72 respectively
  - Additionally, GPIO Banks 0, 1, 2, 3, 4, and 5 Interrupts assigned to EDMA events 6, 7, 22, 23, 28, and 29 respectively.
- Set/clear functionality: Firmware writes 1 to corresponding bit position(s) to set or to clear GPIO signal(s). This allows multiple firmware processes to toggle GPIO output signals without critical section protection (disable interrupts, program GPIO, re-enable interrupts, to prevent context switching to another process during GPIO programming).
- Separate Input/Output registers
- Output register in addition to set/clear so that, if preferred by firmware, some GPIO output signals can be toggled by direct write to the output register(s).
- Output register, when read, reflects output drive status. This, in addition to the input register reflecting pin status and open-drain I/O cell, allows wired logic be implemented.

The memory map for the GPIO registers is shown in [Table 6-9](#).

### 6.8.1 GPIO Register Description(s)

**Table 6-9. GPIO Registers**

| GPIO BYTE ADDRESS         | REGISTER NAME  | REGISTER DESCRIPTION                                    |
|---------------------------|----------------|---|
| 0x01E2 6000               | REV            | Peripheral Revision Register                            |
| 0x01E2 6004               | -              | Reserved  |
| 0x01E2 6008               | BINTEN         | GPIO Interrupt Per-Bank Enable Register                 |
| <b>GPIO Banks 0 and 1</b> |                |   |
| 0x01E2 6010               | DIR01          | GPIO Banks 0 and 1 Direction Register                   |
| 0x01E2 6014               | OUT_DATA01     | GPIO Banks 0 and 1 Output Data Register                 |
| 0x01E2 6018               | SET_DATA01     | GPIO Banks 0 and 1 Set Data Register                    |
| 0x01E2 601C               | CLR_DATA01     | GPIO Banks 0 and 1 Clear Data Register                  |
| 0x01E2 6020               | IN_DATA01      | GPIO Banks 0 and 1 Input Data Register                  |
| 0x01E2 6024               | SET_RIS_TRIG01 | GPIO Banks 0 and 1 Set Rising Edge Interrupt Register   |
| 0x01E2 6028               | CLR_RIS_TRIG01 | GPIO Banks 0 and 1 Clear Rising Edge Interrupt Register |
| 0x01E2 602C               | SET_FAL_TRIG01 | GPIO Banks 0 and 1 Set Falling Edge Interrupt Register  |

**Table 6-9. GPIO Registers (continued)**

| GPIO BYTE ADDRESS         | REGISTER NAME  | REGISTER DESCRIPTION                                     |
|---------------------------|----------------|--|
| 0x01E2 6030               | CLR_FAL_TRIG01 | GPIO Banks 0 and 1 Clear Falling Edge Interrupt Register |
| 0x01E2 6034               | INTSTAT01      | GPIO Banks 0 and 1 Interrupt Status Register             |
| <b>GPIO Banks 2 and 3</b> |                |  |
| 0x01E2 6038               | DIR23          | GPIO Banks 2 and 3 Direction Register                    |
| 0x01E2 603C               | OUT_DATA23     | GPIO Banks 2 and 3 Output Data Register                  |
| 0x01E2 6040               | SET_DATA23     | GPIO Banks 2 and 3 Set Data Register                     |
| 0x01E2 6044               | CLR_DATA23     | GPIO Banks 2 and 3 Clear Data Register                   |
| 0x01E2 6048               | IN_DATA23      | GPIO Banks 2 and 3 Input Data Register                   |
| 0x01E2 604C               | SET_RIS_TRIG23 | GPIO Banks 2 and 3 Set Rising Edge Interrupt Register    |
| 0x01E2 6050               | CLR_RIS_TRIG23 | GPIO Banks 2 and 3 Clear Rising Edge Interrupt Register  |
| 0x01E2 6054               | SET_FAL_TRIG23 | GPIO Banks 2 and 3 Set Falling Edge Interrupt Register   |
| 0x01E2 6058               | CLR_FAL_TRIG23 | GPIO Banks 2 and 3 Clear Falling Edge Interrupt Register |
| 0x01E2 605C               | INTSTAT23      | GPIO Banks 2 and 3 Interrupt Status Register             |
| <b>GPIO Banks 4 and 5</b> |                |  |
| 0x01E2 6060               | DIR45          | GPIO Banks 4 and 5 Direction Register                    |
| 0x01E2 6064               | OUT_DATA45     | GPIO Banks 4 and 5 Output Data Register                  |
| 0x01E2 6068               | SET_DATA45     | GPIO Banks 4 and 5 Set Data Register                     |
| 0x01E2 606C               | CLR_DATA45     | GPIO Banks 4 and 5 Clear Data Register                   |
| 0x01E2 6070               | IN_DATA45      | GPIO Banks 4 and 5 Input Data Register                   |
| 0x01E2 6074               | SET_RIS_TRIG45 | GPIO Banks 4 and 5 Set Rising Edge Interrupt Register    |
| 0x01E2 6078               | CLR_RIS_TRIG45 | GPIO Banks 4 and 5 Clear Rising Edge Interrupt Register  |
| 0x01E2 607C               | SET_FAL_TRIG45 | GPIO Banks 4 and 5 Set Falling Edge Interrupt Register   |
| 0x01E2 6080               | CLR_FAL_TRIG45 | GPIO Banks 4 and 5 Clear Falling Edge Interrupt Register |
| 0x01E2 6084               | INTSTAT45      | GPIO Banks 4 and 5 Interrupt Status Register             |
| <b>GPIO Banks 6 and 7</b> |                |  |
| 0x01E2 6088               | DIR67          | GPIO Banks 6 and 7 Direction Register                    |
| 0x01E2 608C               | OUT_DATA67     | GPIO Banks 6 and 7 Output Data Register                  |
| 0x01E2 6090               | SET_DATA67     | GPIO Banks 6 and 7 Set Data Register                     |
| 0x01E2 6094               | CLR_DATA67     | GPIO Banks 6 and 7 Clear Data Register                   |
| 0x01E2 6098               | IN_DATA67      | GPIO Banks 6 and 7 Input Data Register                   |
| 0x01E2 609C               | SET_RIS_TRIG67 | GPIO Banks 6 and 7 Set Rising Edge Interrupt Register    |
| 0x01E2 60A0               | CLR_RIS_TRIG67 | GPIO Banks 6 and 7 Clear Rising Edge Interrupt Register  |
| 0x01E2 60A4               | SET_FAL_TRIG67 | GPIO Banks 6 and 7 Set Falling Edge Interrupt Register   |
| 0x01E2 60A8               | CLR_FAL_TRIG67 | GPIO Banks 6 and 7 Clear Falling Edge Interrupt Register |
| 0x01E2 60AC               | INTSTAT67      | GPIO Banks 6 and 7 Interrupt Status Register             |

## 6.8.2 GPIO Peripheral Input/Output Electrical Data/Timing

**Table 6-10. Timing Requirements for GPIO Inputs<sup>(1)</sup> (see Figure 6-10)**

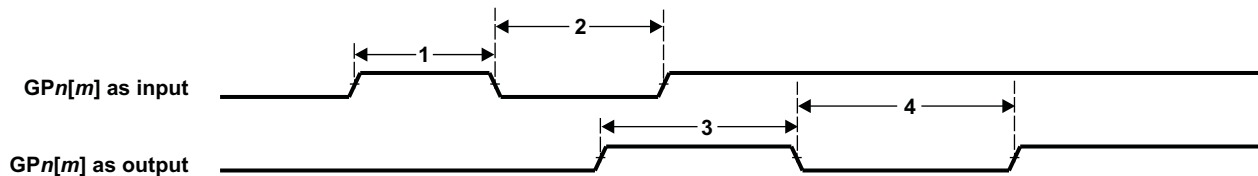
| NO. |               |                           | MIN            | MAX | UNIT |
|-----|---------------|---------------------------|----------------|-----|------|
| 1   | $t_{w(GPIH)}$ | Pulse duration, GPIx high | $2C^{(1) (2)}$ |     | ns   |
| 2   | $t_{w(GPIL)}$ | Pulse duration, GPIx low  | $2C^{(1) (2)}$ |     |      |

- (1) The pulse width given is sufficient to generate a CPU interrupt or an EDMA event. However, if a user wants to have the device recognize the GPIx changes through software polling of the GPIO register, the GPIx duration must be extended to allow the device enough time to access the GPIO register through the internal bus.
- (2)  $C = \text{SYSCLK4}$  period in ns.

**Table 6-11. Switching Characteristics Over Recommended Operating Conditions for GPIO Outputs (see Figure 6-10)**

| NO. |               | PARAMETER                 | MIN            | MAX | UNIT |
|-----|---------------|---------------------------|----------------|-----|------|
| 3   | $t_{w(GPOH)}$ | Pulse duration, GPOx high | $2C^{(1) (2)}$ |     | ns   |
| 4   | $t_{w(GPOL)}$ | Pulse duration, GPOx low  | $2C^{(1) (2)}$ |     | ns   |

- (1) This parameter value should not be used as a maximum performance specification. Actual performance of back-to-back accesses of the GPIO is dependent upon internal bus activity.
- (2)  $C = \text{SYSCLK4}$  period in ns.



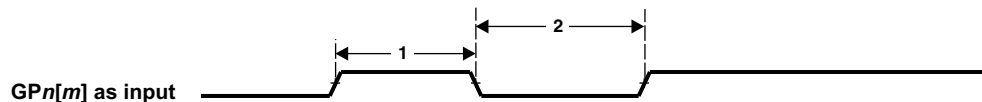
**Figure 6-10. GPIO Port Timing**

## 6.8.3 GPIO Peripheral External Interrupts Electrical Data/Timing

**Table 6-12. Timing Requirements for External Interrupts<sup>(1)</sup> (see Figure 6-11)**

| NO. |                |  | MIN            | MAX | UNIT |
|-----|----------------|--|----------------|-----|------|
| 1   | $t_{w(ILOW)}$  | Width of the external interrupt pulse low  | $2C^{(1) (2)}$ |     | ns   |
| 2   | $t_{w(IHIGH)}$ | Width of the external interrupt pulse high | $2C^{(1) (2)}$ |     |      |

- (1) The pulse width given is sufficient to generate an interrupt or an EDMA event. However, if a user wants to have the device recognize the GPIO changes through software polling of the GPIO register, the GPIO duration must be extended to allow the device enough time to access the GPIO register through the internal bus.
- (2)  $C = \text{SYSCLK4}$  period in ns.



**Figure 6-11. GPIO External Interrupt Timing**

## 6.9 EDMA

Table 6-13 is the list of EDMA3 Channel Controller Registers and Table 6-14 is the list of EDMA3 Transfer Controller registers. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

**Table 6-13. EDMA3 Channel Controller (EDMA3CC) Registers**

| BYTE ADDRESS              | REGISTER NAME | REGISTER DESCRIPTION                            |
|---------------------------|---------------|---|
| 0x01C0 0000               | PID           | Peripheral Identification Register              |
| 0x01C0 0004               | CCCFG         | EDMA3CC Configuration Register                  |
| <b>Global Registers</b>   |               |   |
| 0x01C0 0200               | QCHMAP0       | QDMA Channel 0 Mapping Register                 |
| 0x01C0 0204               | QCHMAP1       | QDMA Channel 1 Mapping Register                 |
| 0x01C0 0208               | QCHMAP2       | QDMA Channel 2 Mapping Register                 |
| 0x01C0 020C               | QCHMAP3       | QDMA Channel 3 Mapping Register                 |
| 0x01C0 0210               | QCHMAP4       | QDMA Channel 4 Mapping Register                 |
| 0x01C0 0214               | QCHMAP5       | QDMA Channel 5 Mapping Register                 |
| 0x01C0 0218               | QCHMAP6       | QDMA Channel 6 Mapping Register                 |
| 0x01C0 021C               | QCHMAP7       | QDMA Channel 7 Mapping Register                 |
| 0x01C0 0240               | DMAQNUM0      | DMA Channel Queue Number Register 0             |
| 0x01C0 0244               | DMAQNUM1      | DMA Channel Queue Number Register 1             |
| 0x01C0 0248               | DMAQNUM2      | DMA Channel Queue Number Register 2             |
| 0x01C0 024C               | DMAQNUM3      | DMA Channel Queue Number Register 3             |
| 0x01C0 0260               | QDMAQNUM      | QDMA Channel Queue Number Register              |
| 0x01C0 0284               | QUEPRI        | Queue Priority Register <sup>(1)</sup>          |
| 0x01C0 0300               | EMR           | Event Missed Register                           |
| 0x01C0 0308               | EMCR          | Event Missed Clear Register                     |
| 0x01C0 0310               | QEMR          | QDMA Event Missed Register                      |
| 0x01C0 0314               | QEMCR         | QDMA Event Missed Clear Register                |
| 0x01C0 0318               | CCERR         | EDMA3CC Error Register                          |
| 0x01C0 031C               | CCERRCLR      | EDMA3CC Error Clear Register                    |
| 0x01C0 0320               | EEVAL         | Error Evaluate Register                         |
| 0x01C0 0340               | DRAE0         | DMA Region Access Enable Register for Region 0  |
| 0x01C0 0348               | DRAE1         | DMA Region Access Enable Register for Region 1  |
| 0x01C0 0350               | DRAE2         | DMA Region Access Enable Register for Region 2  |
| 0x01C0 0358               | DRAE3         | DMA Region Access Enable Register for Region 3  |
| 0x01C0 0380               | QRAE0         | QDMA Region Access Enable Register for Region 0 |
| 0x01C0 0384               | QRAE1         | QDMA Region Access Enable Register for Region 1 |
| 0x01C0 0388               | QRAE2         | QDMA Region Access Enable Register for Region 2 |
| 0x01C0 038C               | QRAE3         | QDMA Region Access Enable Register for Region 3 |
| 0x01C0 0400 - 0x01C0 043C | Q0E0-Q0E15    | Event Queue Entry Registers Q0E0-Q0E15          |
| 0x01C0 0440 - 0x01C0 047C | Q1E0-Q1E15    | Event Queue Entry Registers Q1E0-Q1E15          |
| 0x01C0 0600               | QSTAT0        | Queue 0 Status Register                         |
| 0x01C0 0604               | QSTAT1        | Queue 1 Status Register                         |
| 0x01C0 0620               | QWMTHRA       | Queue Watermark Threshold A Register            |
| 0x01C0 0640               | CCSTAT        | EDMA3CC Status Register                         |

(1) On previous architectures, the EDMA3TC priority was controlled by the queue priority register (QUEPRI) in the EDMA3CC memory-map. However for this device, the priority control for the transfer controllers is controlled by the chip-level registers in the System Configuration Module. You should use the chip-level registers and not QUEPRI to configure the TC priority.

**Table 6-13. EDMA3 Channel Controller (EDMA3CC) Registers (continued)**

| BYTE ADDRESS                             | REGISTER NAME | REGISTER DESCRIPTION                |
|--|---------------|-------------------------------------|
| <b>Global Channel Registers</b>          |               |                                     |
| 0x01C0 1000                              | ER            | Event Register                      |
| 0x01C0 1008                              | ECR           | Event Clear Register                |
| 0x01C0 1010                              | ESR           | Event Set Register                  |
| 0x01C0 1018                              | CER           | Chained Event Register              |
| 0x01C0 1020                              | EER           | Event Enable Register               |
| 0x01C0 1028                              | EECR          | Event Enable Clear Register         |
| 0x01C0 1030                              | EESR          | Event Enable Set Register           |
| 0x01C0 1038                              | SER           | Secondary Event Register            |
| 0x01C0 1040                              | SECR          | Secondary Event Clear Register      |
| 0x01C0 1050                              | IER           | Interrupt Enable Register           |
| 0x01C0 1058                              | IECR          | Interrupt Enable Clear Register     |
| 0x01C0 1060                              | IESR          | Interrupt Enable Set Register       |
| 0x01C0 1068                              | IPR           | Interrupt Pending Register          |
| 0x01C0 1070                              | ICR           | Interrupt Clear Register            |
| 0x01C0 1078                              | IEVAL         | Interrupt Evaluate Register         |
| 0x01C0 1080                              | QER           | QDMA Event Register                 |
| 0x01C0 1084                              | QEER          | QDMA Event Enable Register          |
| 0x01C0 1088                              | QEECR         | QDMA Event Enable Clear Register    |
| 0x01C0 108C                              | QEESR         | QDMA Event Enable Set Register      |
| 0x01C0 1090                              | QSER          | QDMA Secondary Event Register       |
| 0x01C0 1094                              | QSECR         | QDMA Secondary Event Clear Register |
| <b>Shadow Region 0 Channel Registers</b> |               |                                     |
| 0x01C0 2000                              | ER            | Event Register                      |
| 0x01C0 2008                              | ECR           | Event Clear Register                |
| 0x01C0 2010                              | ESR           | Event Set Register                  |
| 0x01C0 2018                              | CER           | Chained Event Register              |
| 0x01C0 2020                              | EER           | Event Enable Register               |
| 0x01C0 2028                              | EECR          | Event Enable Clear Register         |
| 0x01C0 2030                              | EESR          | Event Enable Set Register           |
| 0x01C0 2038                              | SER           | Secondary Event Register            |
| 0x01C0 2040                              | SECR          | Secondary Event Clear Register      |
| 0x01C0 2050                              | IER           | Interrupt Enable Register           |
| 0x01C0 2058                              | IECR          | Interrupt Enable Clear Register     |
| 0x01C0 2060                              | IESR          | Interrupt Enable Set Register       |
| 0x01C0 2068                              | IPR           | Interrupt Pending Register          |
| 0x01C0 2070                              | ICR           | Interrupt Clear Register            |
| 0x01C0 2078                              | IEVAL         | Interrupt Evaluate Register         |
| 0x01C0 2080                              | QER           | QDMA Event Register                 |
| 0x01C0 2084                              | QEER          | QDMA Event Enable Register          |
| 0x01C0 2088                              | QEECR         | QDMA Event Enable Clear Register    |
| 0x01C0 208C                              | QEESR         | QDMA Event Enable Set Register      |
| 0x01C0 2090                              | QSER          | QDMA Secondary Event Register       |
| 0x01C0 2094                              | QSECR         | QDMA Secondary Event Clear Register |

**Table 6-13. EDMA3 Channel Controller (EDMA3CC) Registers (continued)**

| BYTE ADDRESS                             | REGISTER NAME | REGISTER DESCRIPTION                |
|--|---------------|-------------------------------------|
| <b>Shadow Region 1 Channel Registers</b> |               |                                     |
| 0x01C0 2200                              | ER            | Event Register                      |
| 0x01C0 2208                              | ECR           | Event Clear Register                |
| 0x01C0 2210                              | ESR           | Event Set Register                  |
| 0x01C0 2218                              | CER           | Chained Event Register              |
| 0x01C0 2220                              | EER           | Event Enable Register               |
| 0x01C0 2228                              | EECR          | Event Enable Clear Register         |
| 0x01C0 2230                              | EESR          | Event Enable Set Register           |
| 0x01C0 2238                              | SER           | Secondary Event Register            |
| 0x01C0 2240                              | SECR          | Secondary Event Clear Register      |
| 0x01C0 2250                              | IER           | Interrupt Enable Register           |
| 0x01C0 2258                              | IECR          | Interrupt Enable Clear Register     |
| 0x01C0 2260                              | IESR          | Interrupt Enable Set Register       |
| 0x01C0 2268                              | IPR           | Interrupt Pending Register          |
| 0x01C0 2270                              | ICR           | Interrupt Clear Register            |
| 0x01C0 2278                              | IEVAL         | Interrupt Evaluate Register         |
| 0x01C0 2280                              | QER           | QDMA Event Register                 |
| 0x01C0 2284                              | QEER          | QDMA Event Enable Register          |
| 0x01C0 2288                              | QEECR         | QDMA Event Enable Clear Register    |
| 0x01C0 228C                              | QEESR         | QDMA Event Enable Set Register      |
| 0x01C0 2290                              | QSER          | QDMA Secondary Event Register       |
| 0x01C0 2294                              | QSECR         | QDMA Secondary Event Clear Register |
| 0x01C0 4000 - 0x01C0 4FFF                | —             | Parameter RAM (PaRAM)               |

**Table 6-14. EDMA3 Transfer Controller (EDMA3TC) Registers**

| TRANSFER CONTROLLER 0 BYTE ADDRESS | TRANSFER CONTROLLER 1 BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                                     |
|------------------------------------|------------------------------------|---------------|--|
| 0x01C0 8000                        | 0x01C0 8400                        | PID           | Peripheral Identification Register                       |
| 0x01C0 8004                        | 0x01C0 8404                        | TCCFG         | EDMA3TC Configuration Register                           |
| 0x01C0 8100                        | 0x01C0 8500                        | TCSTAT        | EDMA3TC Channel Status Register                          |
| 0x01C0 8120                        | 0x01C0 8520                        | ERRSTAT       | Error Status Register                                    |
| 0x01C0 8124                        | 0x01C0 8524                        | ERREN         | Error Enable Register                                    |
| 0x01C0 8128                        | 0x01C0 8528                        | ERRCLR        | Error Clear Register                                     |
| 0x01C0 812C                        | 0x01C0 852C                        | ERRDET        | Error Details Register                                   |
| 0x01C0 8130                        | 0x01C0 8530                        | ERRCMD        | Error Interrupt Command Register                         |
| 0x01C0 8140                        | 0x01C0 8540                        | RDRATE        | Read Command Rate Register                               |
| 0x01C0 8240                        | 0x01C0 8640                        | SAOPT         | Source Active Options Register                           |
| 0x01C0 8244                        | 0x01C0 8644                        | SASRC         | Source Active Source Address Register                    |
| 0x01C0 8248                        | 0x01C0 8648                        | SACNT         | Source Active Count Register                             |
| 0x01C0 824C                        | 0x01C0 864C                        | SADST         | Source Active Destination Address Register               |
| 0x01C0 8250                        | 0x01C0 8650                        | SABIDX        | Source Active B-Index Register                           |
| 0x01C0 8254                        | 0x01C0 8654                        | SAMPPRXY      | Source Active Memory Protection Proxy Register           |
| 0x01C0 8258                        | 0x01C0 8658                        | SACNTRLD      | Source Active Count Reload Register                      |
| 0x01C0 825C                        | 0x01C0 865C                        | SASRCBREF     | Source Active Source Address B-Reference Register        |
| 0x01C0 8260                        | 0x01C0 8660                        | SADSTBREF     | Source Active Destination Address B-Reference Register   |
| 0x01C0 8280                        | 0x01C0 8680                        | DFCNTRLD      | Destination FIFO Set Count Reload Register               |
| 0x01C0 8284                        | 0x01C0 8684                        | DFSRCBREF     | Destination FIFO Set Source Address B-Reference Register |

**Table 6-14. EDMA3 Transfer Controller (EDMA3TC) Registers (continued)**

| TRANSFER CONTROLLER 0 BYTE ADDRESS | TRANSFER CONTROLLER 1 BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION  |
|------------------------------------|------------------------------------|---------------|---|
| 0x01C0 8288                        | 0x01C0 8688                        | DFDSTBREF     | Destination FIFO Set Destination Address B-Reference Register |
| 0x01C0 8300                        | 0x01C0 8700                        | DFOPT0        | Destination FIFO Options Register 0                           |
| 0x01C0 8304                        | 0x01C0 8704                        | DFSRC0        | Destination FIFO Source Address Register 0                    |
| 0x01C0 8308                        | 0x01C0 8708                        | DFCNT0        | Destination FIFO Count Register 0                             |
| 0x01C0 830C                        | 0x01C0 870C                        | DFDST0        | Destination FIFO Destination Address Register 0               |
| 0x01C0 8310                        | 0x01C0 8710                        | DFBIDX0       | Destination FIFO B-Index Register 0                           |
| 0x01C0 8314                        | 0x01C0 8714                        | DFMPPRXY0     | Destination FIFO Memory Protection Proxy Register 0           |
| 0x01C0 8340                        | 0x01C0 8740                        | DFOPT1        | Destination FIFO Options Register 1                           |
| 0x01C0 8344                        | 0x01C0 8744                        | DFSRC1        | Destination FIFO Source Address Register 1                    |
| 0x01C0 8348                        | 0x01C0 8748                        | DFCNT1        | Destination FIFO Count Register 1                             |
| 0x01C0 834C                        | 0x01C0 874C                        | DFDST1        | Destination FIFO Destination Address Register 1               |
| 0x01C0 8350                        | 0x01C0 8750                        | DFBIDX1       | Destination FIFO B-Index Register 1                           |
| 0x01C0 8354                        | 0x01C0 8754                        | DFMPPRXY1     | Destination FIFO Memory Protection Proxy Register 1           |
| 0x01C0 8380                        | 0x01C0 8780                        | DFOPT2        | Destination FIFO Options Register 2                           |
| 0x01C0 8384                        | 0x01C0 8784                        | DFSRC2        | Destination FIFO Source Address Register 2                    |
| 0x01C0 8388                        | 0x01C0 8788                        | DFCNT2        | Destination FIFO Count Register 2                             |
| 0x01C0 838C                        | 0x01C0 878C                        | DFDST2        | Destination FIFO Destination Address Register 2               |
| 0x01C0 8390                        | 0x01C0 8790                        | DFBIDX2       | Destination FIFO B-Index Register 2                           |
| 0x01C0 8394                        | 0x01C0 8794                        | DFMPPRXY2     | Destination FIFO Memory Protection Proxy Register 2           |
| 0x01C0 83C0                        | 0x01C0 87C0                        | DFOPT3        | Destination FIFO Options Register 3                           |
| 0x01C0 83C4                        | 0x01C0 87C4                        | DFSRC3        | Destination FIFO Source Address Register 3                    |
| 0x01C0 83C8                        | 0x01C0 87C8                        | DFCNT3        | Destination FIFO Count Register 3                             |
| 0x01C0 83CC                        | 0x01C0 87CC                        | DFDST3        | Destination FIFO Destination Address Register 3               |
| 0x01C0 83D0                        | 0x01C0 87D0                        | DFBIDX3       | Destination FIFO B-Index Register 3                           |
| 0x01C0 83D4                        | 0x01C0 87D4                        | DFMPPRXY3     | Destination FIFO Memory Protection Proxy Register 3           |

Table 6-15 shows an abbreviation of the set of registers which make up the parameter set for each of 128 EDMA events. Each of the parameter register sets consist of 8 32-bit word entries. Table 6-16 shows the parameter set entry registers with relative memory address locations within each of the parameter sets.

**Table 6-15. EDMA Parameter Set RAM**

| HEX ADDRESS RANGE         | DESCRIPTION                         |
|---------------------------|-------------------------------------|
| 0x01C0 4000 - 0x01C0 401F | Parameters Set 0 (8 32-bit words)   |
| 0x01C0 4020 - 0x01C0 403F | Parameters Set 1 (8 32-bit words)   |
| 0x01C0 4040 - 0x01C0 405F | Parameters Set 2 (8 32-bit words)   |
| 0x01C0 4060 - 0x01C0 407F | Parameters Set 3 (8 32-bit words)   |
| 0x01C0 4080 - 0x01C0 409F | Parameters Set 4 (8 32-bit words)   |
| 0x01C0 40A0 - 0x01C0 40BF | Parameters Set 5 (8 32-bit words)   |
| ...                       | ...                                 |
| 0x01C0 4FC0 - 0x01C0 4FDF | Parameters Set 126 (8 32-bit words) |
| 0x01C0 4FE0 - 0x01C0 4FFF | Parameters Set 127 (8 32-bit words) |

**Table 6-16. Parameter Set Entries**

| HEX OFFSET ADDRESS<br>WITHIN THE PARAMETER SET | ACRONYM      | PARAMETER ENTRY                     |
|--|--------------|-------------------------------------|
| 0x0000   | OPT          | Option                              |
| 0x0004   | SRC          | Source Address                      |
| 0x0008   | A_B_CNT      | A Count, B Count                    |
| 0x000C   | DST          | Destination Address                 |
| 0x0010   | SRC_DST_BIDX | Source B Index, Destination B Index |
| 0x0014   | LINK_BCNTRLD | Link Address, B Count Reload        |
| 0x0018   | SRC_DST_CIDX | Source C Index, Destination C Index |
| 0x001C   | CCNT         | C Count                             |

**Table 6-17. EDMA Events**

| EVENT | EVENT NAME / SOURCE    | EVENT | EVENT NAME / SOURCE   |
|-------|------------------------|-------|-----------------------|
| 0     | McASP0 Receive         | 16    | MMCS0 Receive         |
| 1     | McASP0 Transmit        | 17    | MMCS0 Transmit        |
| 2     | McASP1 Receive         | 18    | -                     |
| 3     | McASP1 Transmit        | 19    | -                     |
| 4     | -                      | 20    | PRU_EVTOUT6           |
| 5     | -                      | 21    | PRU_EVTOUT7           |
| 6     | GPIO Bank 0 Interrupt  | 22    | GPIO Bank 2 Interrupt |
| 7     | GPIO Bank 1 Interrupt  | 23    | GPIO Bank 3 Interrupt |
| 8     | UART0 Receive          | 24    | I2C0 Receive          |
| 9     | UART0 Transmit         | 25    | I2C0 Transmit         |
| 10    | Timer64P0 Event Out 12 | 26    | I2C1 Receive          |
| 11    | Timer64P0 Event Out 34 | 27    | I2C1 Transmit         |
| 12    | -                      | 28    | GPIO Bank 4 Interrupt |
| 13    | -                      | 29    | GPIO Bank 5 Interrupt |
| 14    | SPI0 Receive           | 30    | UART2 Receive         |
| 15    | SPI0 Transmit          | 31    | UART2 Transmit        |

## 6.10 External Memory Interface A (EMIFA)

EMIFA is one of two external memory interfaces supported on the device. It is primarily intended to support asynchronous memory types, such as NAND and NOR flash and Asynchronous SRAM. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

### 6.10.1 EMIFA Asynchronous Memory Support

EMIFA supports asynchronous:

- SRAM memories
- NAND Flash memories
- NOR Flash memories

The EMIFA data bus width is up to 16-bits on the ZKB package and 8 bits on the PTP package. Both devices support up to fifteen address lines and an external wait/interrupt input. Up to four asynchronous chip selects are supported by EMIFA ( $\overline{\text{EMA\_CS}}[5:2]$ ). All four chip selects are available on the ZKB package. Two of the four are available on the PTP package ( $\overline{\text{EMA\_CS}}[3:2]$ ).

Each chip select has the following individually programmable attributes:

- Data Bus Width
- Read cycle timings: setup, hold, strobe
- Write cycle timings: setup, hold, strobe
- Bus turn around time
- Extended Wait Option With Programmable Timeout
- Select Strobe Option
- NAND flash controller supports 1-bit and 4-bit ECC calculation on blocks of 512 bytes.

### 6.10.2 EMIFA Connection Examples

A likely use case with more than one EMIFA chip select used for NAND flash is illustrated in [Figure 6-12](#). This figure shows how two multiplane NAND flash devices with two chip selects each would connect to the EMIFA. In this case if NAND is the boot memory, then the boot image needs to be stored in the NAND area selected by  $\overline{\text{EMA\_CS}}[3]$ . Part of the application image could spill over into the NAND regions selected by other EMIFA chip selects; but would rely on the code stored in the  $\overline{\text{EMA\_CS}}[3]$  area to bootload it.

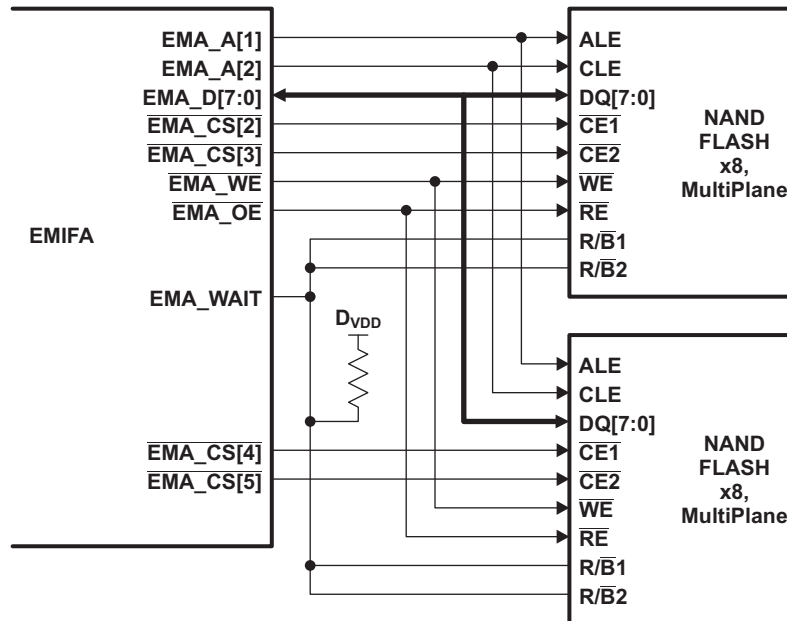


Figure 6-12. C6743 EMIFA Connection Diagram: Multiple NAND Flash Planes

### 6.10.3 External Memory Interface (EMIF) Registers

Table 6-18 is a list of the EMIF registers.

Table 6-18. External Memory Interface (EMIFA) Registers

| BYTE ADDRESS | REGISTER NAME   | REGISTER DESCRIPTION                                    |
|--------------|-----------------|---|
| 0x6800 0000  | MIDR            | Module ID Register                                      |
| 0x6800 0004  | AWCC            | Asynchronous Wait Cycle Configuration Register          |
| 0x6800 0008  | SDCR            | SDRAM Configuration Register (Not supported)            |
| 0x6800 000C  | SDRCR           | SDRAM Refresh Control Register (Not supported)          |
| 0x6800 0010  | CE2CFG          | Asynchronous 1 Configuration Register                   |
| 0x6800 0014  | CE3CFG          | Asynchronous 2 Configuration Register                   |
| 0x6800 0018  | CE4CFG          | Asynchronous 3 Configuration Register                   |
| 0x6800 001C  | CE5CFG          | Asynchronous 4 Configuration Register                   |
| 0x6800 0020  | SDTIMR          | SDRAM Timing Register (Not supported)                   |
| 0x6800 003C  | SDSRETR         | SDRAM Self Refresh Exit Timing Register (Not supported) |
| 0x6800 0040  | INTRAW          | EMIFA Interrupt Raw Register                            |
| 0x6800 0044  | INTMSK          | EMIFA Interrupt Mask Register                           |
| 0x6800 0048  | INTMSKSET       | EMIFA Interrupt Mask Set Register                       |
| 0x6800 004C  | INTMSKCLR       | EMIFA Interrupt Mask Clear Register                     |
| 0x6800 0060  | NANDFCR         | NAND Flash Control Register                             |
| 0x6800 0064  | NANDFSR         | NAND Flash Status Register                              |
| 0x6800 0070  | NANDF1ECC       | NAND Flash 1 ECC Register (CS2 Space)                   |
| 0x6800 0074  | NANDF2ECC       | NAND Flash 2 ECC Register (CS3 Space)                   |
| 0x6800 0078  | NANDF3ECC       | NAND Flash 3 ECC Register (CS4 Space)                   |
| 0x6800 007C  | NANDF4ECC       | NAND Flash 4 ECC Register (CS5 Space)                   |
| 0x6800 00BC  | NAND4BITECCLOAD | NAND Flash 4-Bit ECC Load Register                      |
| 0x6800 00C0  | NAND4BITECC1    | NAND Flash 4-Bit ECC Register 1                         |
| 0x6800 00C4  | NAND4BITECC2    | NAND Flash 4-Bit ECC Register 2                         |
| 0x6800 00C8  | NAND4BITECC3    | NAND Flash 4-Bit ECC Register 3                         |

**Table 6-18. External Memory Interface (EMIFA) Registers (continued)**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                          |
|--------------|---------------|---|
| 0x6800 00CC  | NAND4BITECC4  | NAND Flash 4-Bit ECC Register 4               |
| 0x6800 00D0  | NANDERRADD1   | NAND Flash 4-Bit ECC Error Address Register 1 |
| 0x6800 00D4  | NANDERRADD2   | NAND Flash 4-Bit ECC Error Address Register 2 |
| 0x6800 00D8  | NANDERRVAL1   | NAND Flash 4-Bit ECC Error Value Register 1   |
| 0x6800 00DC  | NANDERRVAL2   | NAND Flash 4-Bit ECC Error Value Register 2   |

### 6.10.4 EMIFA Electrical Data/Timing

Table 6-19 and Table 6-20 assume testing over recommended operating conditions.

**Table 6-19. EMIFA Asynchronous Memory Timing Requirements<sup>(1)</sup>**

| NO.                     |                        |  | MIN  | NOM | MAX | UNIT |
|-------------------------|------------------------|--|------|-----|-----|------|
| <b>READS and WRITES</b> |                        |  |      |     |     |      |
| E                       | $t_{c(CLK)}$           | Cycle time, EMIFA module clock   | 10   |     |     | ns   |
| 2                       | $t_{w(EM\_WAIT)}$      | Pulse duration, EM_WAIT assertion and deassertion                      | 2E   |     |     | ns   |
| <b>READS</b>            |                        |  |      |     |     |      |
| 12                      | $t_{su(EMDV-EMOEH)}$   | Setup time, EMA_D[15:0] valid before $\overline{EM\_OE}$ high          | 3    |     |     | ns   |
| 13                      | $t_{h(EMOEH-EMDIV)}$   | Hold time, EMA_D[15:0] valid after $\overline{EM\_OE}$ high            | 0    |     |     | ns   |
| 14                      | $t_{su(EMOEL-EMWAIT)}$ | Setup time, EM_WAIT asserted before end of Strobe Phase <sup>(2)</sup> | 4E+3 |     |     | ns   |
| <b>WRITES</b>           |                        |  |      |     |     |      |
| 28                      | $t_{su(EMWEL-EMWAIT)}$ | Setup time, EM_WAIT asserted before end of Strobe Phase <sup>(2)</sup> | 4E+3 |     |     | ns   |

- (1) E = EMA\_CLK period or in ns. EMA\_CLK is selected either as SYSCLK3 or the PLL output clock divided by 4.5. As an example, when SYSCLK3 is selected and set to 100MHz, E=10ns.
- (2) Setup before end of STROBE phase (if no extended wait states are inserted) by which EM\_WAIT must be asserted to add extended wait states. Figure 6-15 and Figure 6-16 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 6-20. EMIFA Asynchronous Memory Switching Characteristics<sup>(1) (2) (3)</sup>**

| NO.                     | PARAMETER             |   | MIN                 | NOM               | MAX                 | UNIT |
|-------------------------|-----------------------|---|---------------------|-------------------|---------------------|------|
| <b>READS and WRITES</b> |                       |   |                     |                   |                     |      |
| 1                       | $t_d(TURNAROUND)$     | Turn around time  | (TA)*E-3            | (TA)*E            | (TA)*E+3            | ns   |
| <b>READS</b>            |                       |   |                     |                   |                     |      |
| 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, $\overline{EMA\_CE}$ [5:2] low to $\overline{EMA\_OE}$ low (SS = 0)  | (RS)*E-3            | (RS)*E            | (RS)*E+3            | ns   |
|                         |                       | Output setup time, $\overline{EMA\_CE}$ [5:2] low to $\overline{EMA\_OE}$ low (SS = 1)  | -3                  | 0                 | +3                  | ns   |
| 5                       | $t_{h(EMOEH-EMCEH)}$  | Output hold time, $\overline{EMA\_OE}$ high to $\overline{EMA\_CE}$ [5:2] high (SS = 0) | (RH)*E-3            | (RH)*E            | (RH)*E+3            | ns   |
|                         |                       | Output hold time, $\overline{EMA\_OE}$ high to $\overline{EMA\_CE}$ [5:2] high (SS = 1) | -3                  | 0                 | +3                  | ns   |
| 6                       | $t_{su(EMBAV-EMOEL)}$ | Output setup time, $\overline{EMA\_BA}$ [1:0] valid to $\overline{EMA\_OE}$ low         | (RS)*E-3            | (RS)*E            | (RS)*E+3            | ns   |
| 7                       | $t_{h(EMOEH-EMBAIV)}$ | Output hold time, $\overline{EMA\_OE}$ high to $\overline{EMA\_BA}$ [1:0] invalid       | (RH)*E-3            | (RH)*E            | (RH)*E+3            | ns   |
| 8                       | $t_{su(EMBAV-EMOEL)}$ | Output setup time, $\overline{EMA\_A}$ [13:0] valid to $\overline{EMA\_OE}$ low         | (RS)*E-3            | (RS)*E            | (RS)*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 via the Asynchronous Bank and Asynchronous Wait Cycle Configuration Registers. These support the following range of values: TA[4-1], RS[16-1], RST[64-1], RH[8-1], WS[16-1], WST[64-1], WH[8-1], and MEW[1-256].
- (2) E = EMA\_CLK period or in ns. EMA\_CLK is selected either as SYSCLK3 or the PLL output clock divided by 4.5. As an example, when SYSCLK3 is selected and set to 100MHz, E=10ns.
- (3) EWC = external wait cycles determined by EMA\_WAIT input signal. EWC supports the following range of values EWC[256-1]. Note that the maximum wait time before timeout is specified by bit field MEWC in the Asynchronous Wait Cycle Configuration Register.

**Table 6-20. EMIFA Asynchronous Memory Switching Characteristics<sup>(1) (2) (3)</sup> (continued)**

| NO.           | PARAMETER   | MIN                 | NOM               | MAX                 | UNIT |
|---------------|---|---------------------|-------------------|---------------------|------|
| 9             | $t_{H(EMOEH-EMAIV)}$ Output hold time, $\overline{EMA\_OE}$ high to $\overline{EMA\_A[13:0]}$ invalid       | (RH)*E-3            | (RH)*E            | (RH)*E+3            | ns   |
| 10            | $t_{W(EMOEL)}$ $\overline{EMA\_OE}$ active low width (EW = 0)   | (RST)*E-3           | (RST)*E           | (RST)*E+3           | ns   |
|               | $\overline{EMA\_OE}$ 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 $\overline{EMA\_WAIT}$ deasserted to $\overline{EMA\_OE}$ high       | 3E-3                | 4E                | 4E+3                | ns   |
| <b>WRITES</b> |   |                     |                   |                     |      |
| 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   |
| 16            | $t_{su(EMCEL-EMWEL)}$ Output setup time, $\overline{EMA\_CE[5:2]}$ low to $\overline{EMA\_WE}$ low (SS = 0) | (WS)*E-3            | (WS)*E            | (WS)*E+3            | ns   |
|               | Output setup time, $\overline{EMA\_CE[5:2]}$ low to $\overline{EMA\_WE}$ low (SS = 1)                       | -3                  | 0                 | +3                  | ns   |
| 17            | $t_{H(EMWEH-EMCEH)}$ Output hold time, $\overline{EMA\_WE}$ high to $\overline{EMA\_CE[5:2]}$ high (SS = 0) | (WH)*E-3            | (WH)*E            | (WH)*E+3            | ns   |
|               | Output hold time, $\overline{EMA\_WE}$ high to $\overline{EMA\_CE[5:2]}$ high (SS = 1)                      | -3                  | 0                 | +3                  | ns   |
| 18            | $t_{su(EMDQMV-EMWEL)}$ Output setup time, $\overline{EMA\_BA[1:0]}$ valid to $\overline{EMA\_WE}$ low       | (WS)*E-3            | (WS)*E            | (WS)*E+3            | ns   |
| 19            | $t_{H(EMWEH-EMDQMV)}$ Output hold time, $\overline{EMA\_WE}$ high to $\overline{EMA\_BA[1:0]}$ invalid      | (WH)*E-3            | (WH)*E            | (WH)*E+3            | ns   |
| 20            | $t_{su(EMBAV-EMWEL)}$ Output setup time, $\overline{EMA\_BA[1:0]}$ valid to $\overline{EMA\_WE}$ low        | (WS)*E-3            | (WS)*E            | (WS)*E+3            | ns   |
| 21            | $t_{H(EMWEH-EMBAV)}$ Output hold time, $\overline{EMA\_WE}$ high to $\overline{EMA\_BA[1:0]}$ invalid       | (WH)*E-3            | (WH)*E            | (WH)*E+3            | ns   |
| 22            | $t_{su(EMAV-EMWEL)}$ Output setup time, $\overline{EMA\_A[13:0]}$ valid to $\overline{EMA\_WE}$ low         | (WS)*E-3            | (WS)*E            | (WS)*E+3            | ns   |
| 23            | $t_{H(EMWEH-EMAV)}$ Output hold time, $\overline{EMA\_WE}$ high to $\overline{EMA\_A[13:0]}$ invalid        | (WH)*E-3            | (WH)*E            | (WH)*E+3            | ns   |
| 24            | $t_{W(EMWEL)}$ $\overline{EMA\_WE}$ active low width (EW = 0)   | (WST)*E-3           | (WST)*E           | (WST)*E+3           | ns   |
|               | $\overline{EMA\_WE}$ 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 $\overline{EMA\_WAIT}$ deasserted to $\overline{EMA\_WE}$ high       | 3E-3                | 4E                | 4E+3                | ns   |
| 26            | $t_{su(EMDV-EMWEL)}$ Output setup time, $\overline{EMA\_D[15:0]}$ valid to $\overline{EMA\_WE}$ low         | (WS)*E-3            | (WS)*E            | (WS)*E+3            | ns   |
| 27            | $t_{H(EMWEH-EMDIV)}$ Output hold time, $\overline{EMA\_WE}$ high to $\overline{EMA\_D[15:0]}$ invalid       | (WH)*E-3            | (WH)*E            | (WH)*E+3            | ns   |

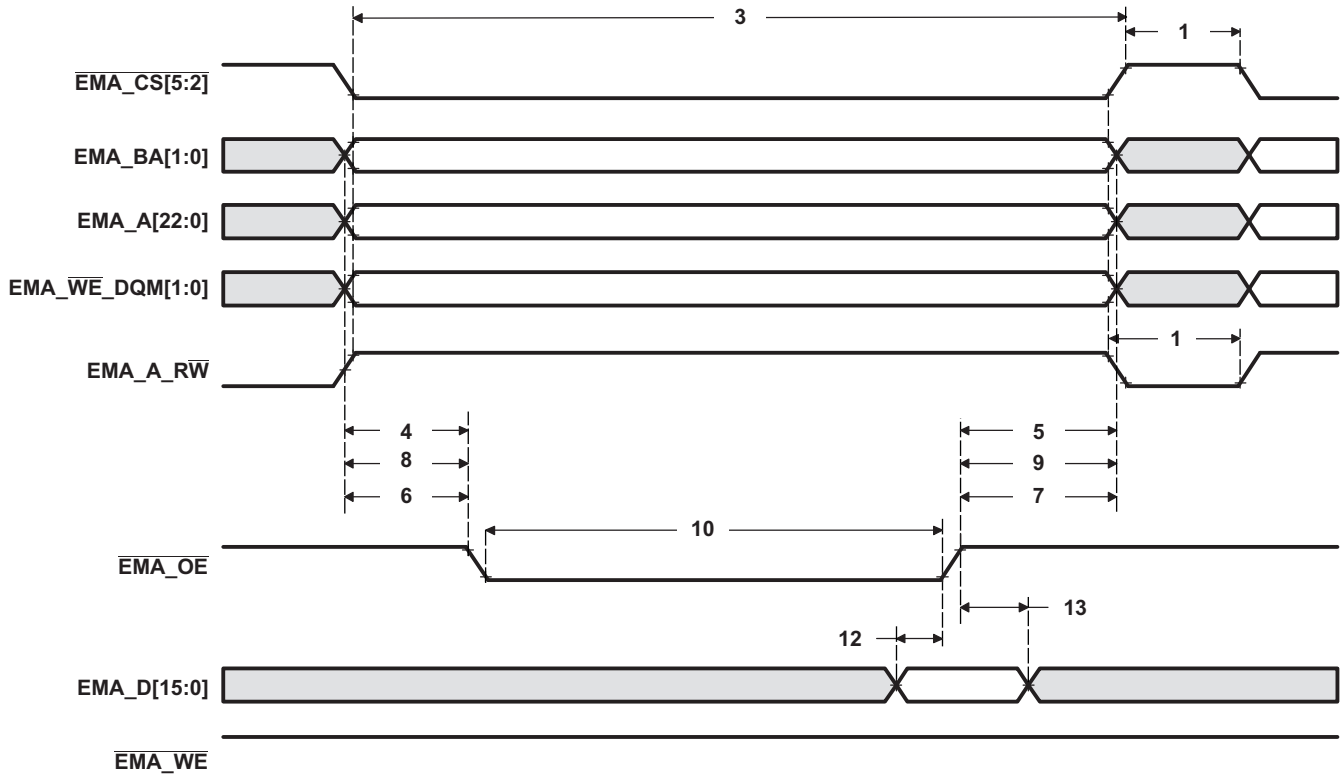


Figure 6-13. Asynchronous Memory Read Timing for EMIFA

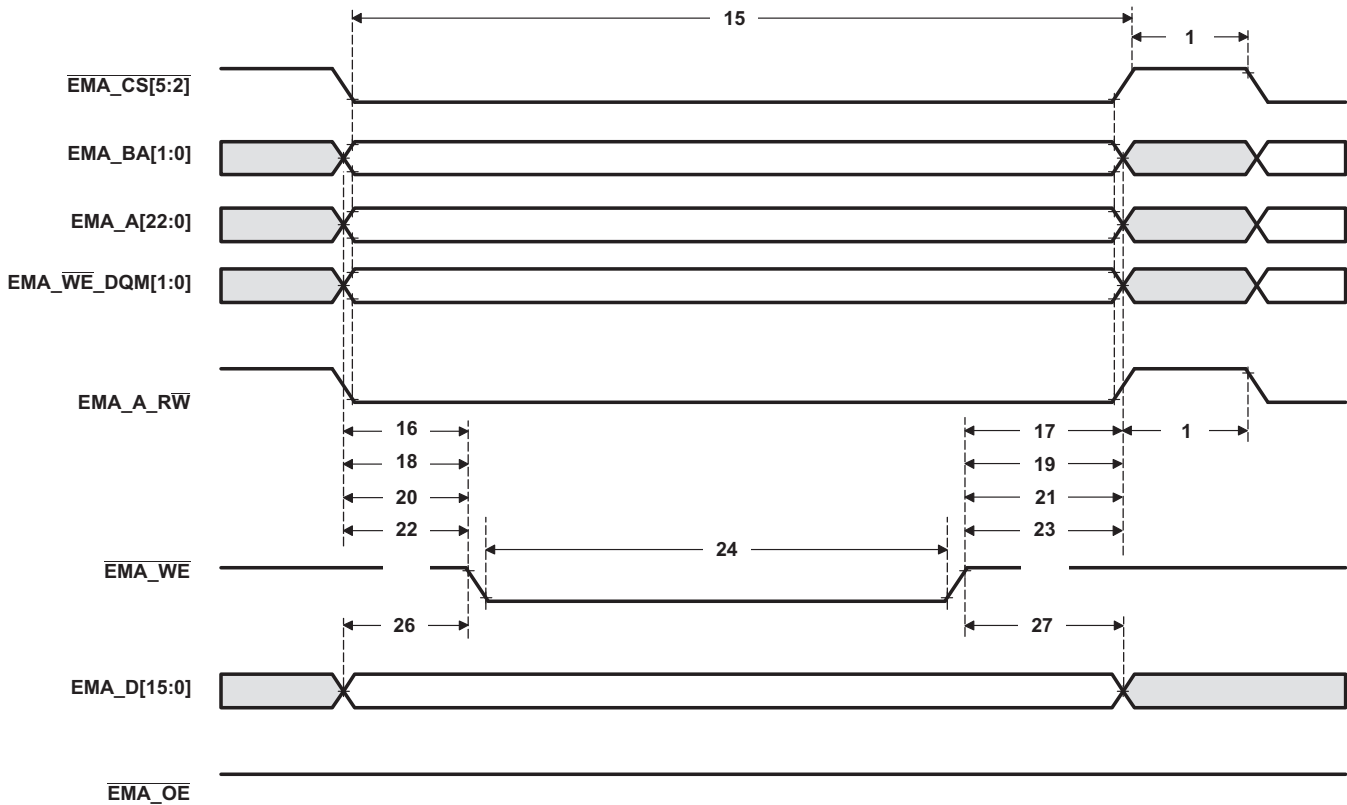


Figure 6-14. Asynchronous Memory Write Timing for EMIFA

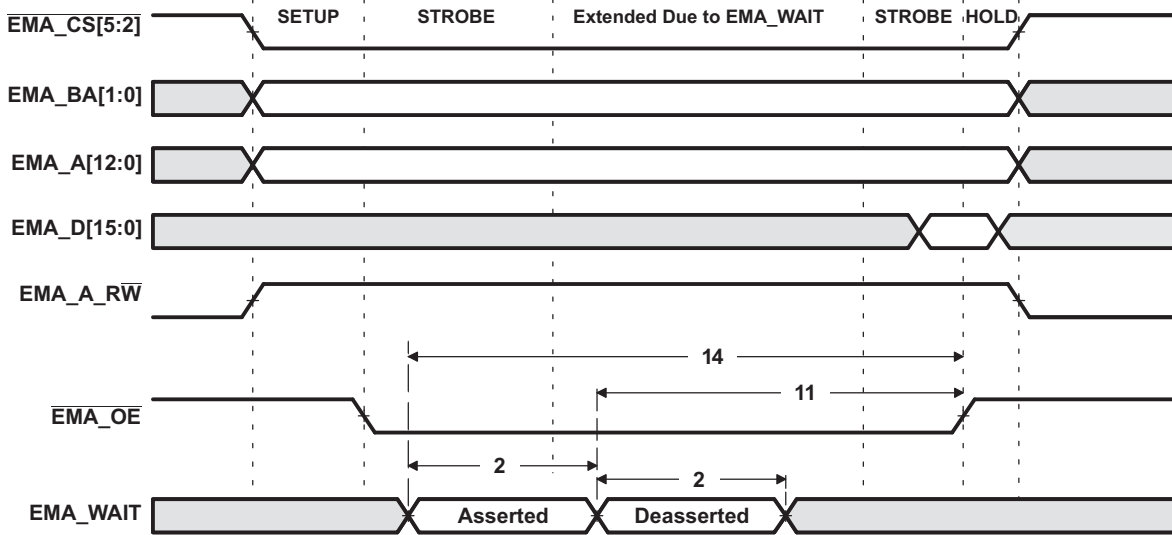


Figure 6-15. EMA\_WAIT Read Timing Requirements

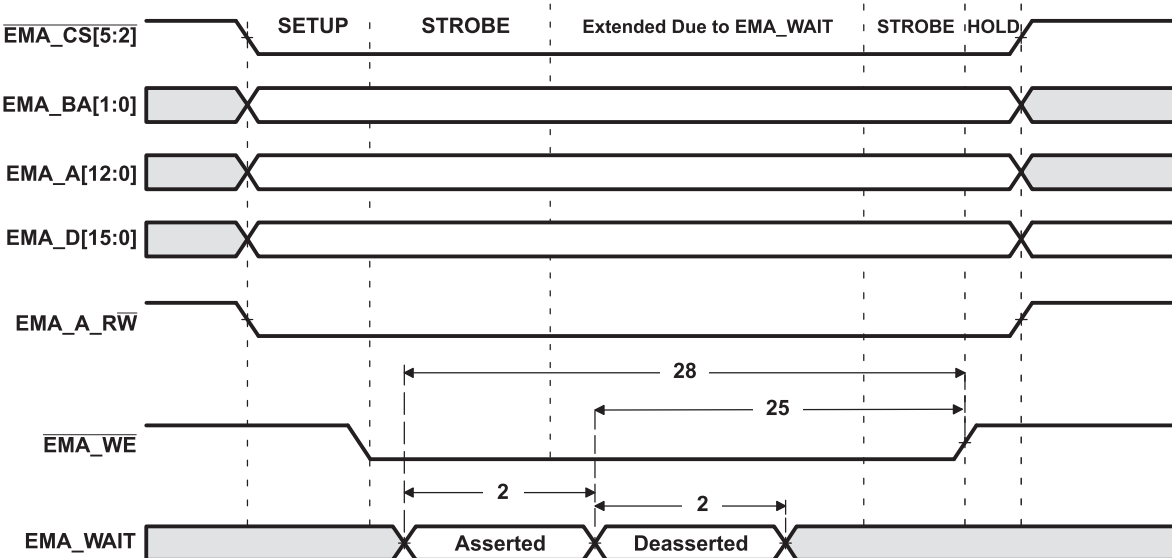


Figure 6-16. EMA\_WAIT Write Timing Requirements

## 6.11 External Memory Interface B (EMIFB)

Figure 6-17, EMIFB Functional Block Diagram illustrates a high-level view of the EMIFB and its connections within the device. Multiple requesters have access to EMIFB through a switched central resource (indicated as crossbar in the figure). The EMIFB implements a split transaction internal bus, allowing concurrence between reads and writes from the various requesters. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

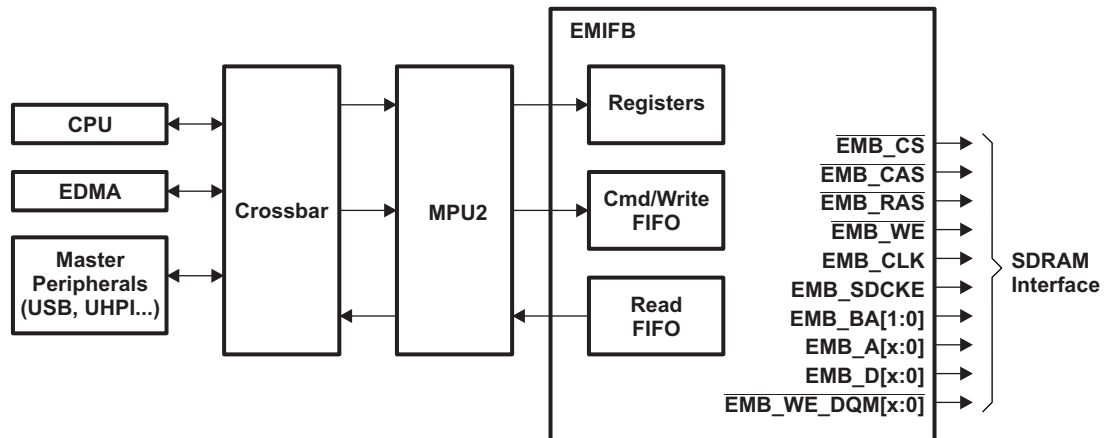


Figure 6-17. EMIFB Functional Block Diagram

EMIFB supports a 3.3V LVCMOS Interface.

### 6.11.1 EMIFB SDRAM Loading Limitations

EMIFB supports SDRAM up to 152MHz with up to two SDRAM or asynchronous memory loads. Additional loads will limit the SDRAM operation to lower speeds and the maximum speed should be confirmed by board simulation using IBIS models.

### 6.11.2 Interfacing to SDRAM

The EMIFB supports a glueless interface to SDRAM devices with the following characteristics:

- Pre-charge bit is A[10]
- Supports 8, 9, 10 or 11 column address bits.
- Supports up to 13 row address bits is 13.
- Supports 1, 2 or 4 internal banks.

Table 6-21 shows the supported SDRAM configurations for EMIFB.

**Table 6-21. EMIFB Supported SDRAM Configurations<sup>(1)</sup>**

| SDRAM Memory Data Bus Width (bits) | Number of Memories | EMIFB Data Bus Size | Rows | Columns | Banks | Total Memory (Mbits) | Total Memory (Mbytes) | Memory Density (Mbits) |
|------------------------------------|--------------------|---------------------|------|---------|-------|----------------------|-----------------------|------------------------|
| 16                                 | 1                  | 16                  | 13   | 8       | 1     | 32                   | 4                     | 32                     |
|                                    | 1                  | 16                  | 13   | 8       | 2     | 64                   | 8                     | 64                     |
|                                    | 1                  | 16                  | 13   | 8       | 4     | 128                  | 16                    | 128                    |
|                                    | 1                  | 16                  | 13   | 9       | 1     | 64                   | 8                     | 64                     |
|                                    | 1                  | 16                  | 13   | 9       | 2     | 128                  | 16                    | 128                    |
|                                    | 1                  | 16                  | 13   | 9       | 4     | 256                  | 32                    | 256                    |
|                                    | 1                  | 16                  | 13   | 10      | 1     | 128                  | 16                    | 128                    |
|                                    | 1                  | 16                  | 13   | 10      | 2     | 256                  | 32                    | 256                    |
|                                    | 1                  | 16                  | 13   | 10      | 4     | 512                  | 64                    | 512                    |
|                                    | 1                  | 16                  | 13   | 11      | 1     | 256                  | 32                    | 256                    |
|                                    | 1                  | 16                  | 13   | 11      | 2     | 512                  | 64                    | 512                    |
|                                    | 1                  | 16                  | 13   | 11      | 4     | 1024                 | 128                   | 1024                   |
| 8                                  | 2                  | 16                  | 13   | 8       | 1     | 32                   | 4                     | 16                     |
|                                    | 2                  | 16                  | 13   | 8       | 2     | 64                   | 8                     | 32                     |
|                                    | 2                  | 16                  | 13   | 8       | 4     | 128                  | 16                    | 64                     |
|                                    | 2                  | 16                  | 13   | 9       | 1     | 64                   | 8                     | 32                     |
|                                    | 2                  | 16                  | 13   | 9       | 2     | 128                  | 16                    | 64                     |
|                                    | 2                  | 16                  | 13   | 9       | 4     | 256                  | 32                    | 128                    |
|                                    | 2                  | 16                  | 13   | 10      | 1     | 128                  | 16                    | 64                     |
|                                    | 2                  | 16                  | 13   | 10      | 2     | 256                  | 32                    | 128                    |
|                                    | 2                  | 16                  | 13   | 10      | 4     | 512                  | 64                    | 256                    |
|                                    | 2                  | 16                  | 13   | 11      | 1     | 256                  | 32                    | 128                    |
|                                    | 2                  | 16                  | 13   | 11      | 2     | 512                  | 64                    | 256                    |
|                                    | 2                  | 16                  | 13   | 11      | 4     | 1024                 | 128                   | 512                    |

(1) The shaded cells indicate configurations that are possible on the EMIFB interface but as of this writing SDRAM memories capable of supporting these densities are not available in the market.

Figure 6-18 shows an interface between the EMIFB and a 2M × 16 × 4 bank SDRAM device. In addition, and shows an interface between the EMIFB and two 4M × 16 × 4 bank SDRAM devices. Refer to Table 6-22, as an example that shows additional list of commonly-supported SDRAM devices and the required connections for the address pins. Note that in Table 6-22, page size/column size (not indicated in the table) is varied to get the required addressability range.

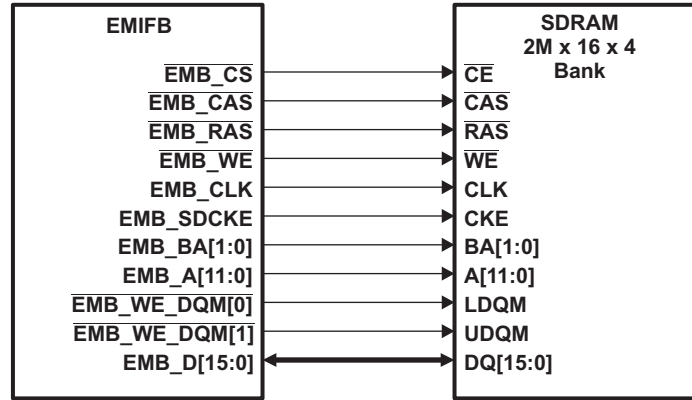


Figure 6-18. EMIFB to 2M x 16 x 4 Bank SDRAM Interface

**Table 6-22. Example of 16/32-Bit EMIFB Address Pin Connections**

| SDRAM SIZE | WIDTH | BANKS | MEMORY         | ADDRESS PINS           |
|------------|-------|-------|----------------|------------------------|
| 64M bits   | x16   | 4     | SDRAM<br>EMIFB | A[11:0]<br>EMB_A[11:0] |
|            | x32   | 4     | SDRAM<br>EMIFB | A[10:0]<br>EMB_A[10:0] |
| 128M bits  | x16   | 4     | SDRAM<br>EMIFB | A[11:0]<br>EMB_A[11:0] |
|            | x32   | 4     | SDRAM<br>EMIFB | A[11:0]<br>EMB_A[11:0] |
| 256M bits  | x16   | 4     | SDRAM<br>EMIFB | A[12:0]<br>EMB_A[12:0] |
|            | x32   | 4     | SDRAM<br>EMIFB | A[11:0]<br>EMB_A[11:0] |
| 512M bits  | x16   | 4     | SDRAM<br>EMIFB | A[12:0]<br>EMB_A[12:0] |
|            | x32   | 4     | SDRAM<br>EMIFB | A[12:0]<br>EMB_A[12:0] |

Table 6-23 is a list of the EMIFB registers.

**Table 6-23. EMIFB Base Controller Registers**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                              |
|--------------|---------------|---|
| 0xB000 0000  | MIDR          | Module ID Register                                |
| 0xB000 0008  | SDCFG         | SDRAM Configuration Register                      |
| 0xB000 000C  | SDRFC         | SDRAM Refresh Control Register                    |
| 0xB000 0010  | SDTIM1        | SDRAM Timing Register 1                           |
| 0xB000 0014  | SDTIM2        | SDRAM Timing Register 2                           |
| 0xB000 001C  | SDCFG2        | SDRAM Configuration 2 Register                    |
| 0xB000 0020  | BPRIO         | Peripheral Bus Burst Priority Register            |
| 0xB000 0040  | PC1           | Performance Counter 1 Register                    |
| 0xB000 0044  | PC2           | Performance Counter 2 Register                    |
| 0xB000 0048  | PCC           | Performance Counter Configuration Register        |
| 0xB000 004C  | PCMRS         | Performance Counter Master Region Select Register |
| 0xB000 0050  | PCT           | Performance Counter Time Register                 |
| 0xB000 00C0  | IRR           | Interrupt Raw Register                            |
| 0xB000 00C4  | IMR           | Interrupt Mask Register                           |
| 0xB000 00C8  | IMSR          | Interrupt Mask Set Register                       |
| 0xB000 00CC  | IMCR          | Interrupt Mask Clear Register                     |

### 6.11.3 EMIFB Electrical Data/Timing

**Table 6-24. EMIFB SDRAM Interface Timing Requirements**

| NO. |                            |  | MIN | MAX | UNIT |
|-----|----------------------------|--|-----|-----|------|
| 19  | $t_{su}(EMA\_DV-EM\_CLKH)$ | Input setup time, read data valid on EMB_D[31:0] before EMB_CLK rising | 0.8 |     | ns   |
| 20  | $t_h(CLKH-DIV)$            | Input hold time, read data valid on EMB_D[31:0] after EMB_CLK rising   | 1.5 |     | ns   |

**Table 6-25. EMIFB SDRAM Interface Switching Characteristics**

| NO. | PARAMETER            | Commercial (Default) Temperature Range                                     |     | Automotive Temperature Range |     | UNIT |
|-----|----------------------|--|-----|------------------------------|-----|------|
|     |                      | MIN  | MAX | MIN                          | MAX |      |
| 1   | $t_c(CLK)$           | Cycle time, EMIF clock EMB_CLK   |     | 7.5                          | 7.5 | ns   |
| 2   | $t_w(CLK)$           | Pulse width, EMIF clock EMB_CLK high or low                                |     | 3                            | 3   | ns   |
| 3   | $t_d(CLKH-CSV)$      | Delay time, EMB_CLK rising to $\overline{EMB\_CS}[0]$ valid                |     | 5.1                          | 5.1 | ns   |
| 4   | $t_{oh}(CLKH-CSIV)$  | Output hold time, EMB_CLK rising to $\overline{EMB\_CS}[0]$ invalid        |     | 1.1                          | 0.9 | ns   |
| 5   | $t_d(CLKH-DQMV)$     | Delay time, EMB_CLK rising to EMB_ $\overline{WE}$ _DQM[3:0] valid         |     | 5.1                          | 5.1 | ns   |
| 6   | $t_{oh}(CLKH-DQMIV)$ | Output hold time, EMB_CLK rising to EMB_ $\overline{WE}$ _DQM[3:0] invalid |     | 1.1                          | 0.9 | ns   |
| 7   | $t_d(CLKH-AV)$       | Delay time, EMB_CLK rising to EMB_A[12:0] and EMB_BA[1:0] valid            |     | 5.1                          | 5.1 | ns   |
| 8   | $t_{oh}(CLKH-AIV)$   | Output hold time, EMB_CLK rising to EMB_A[12:0] and EMB_BA[1:0] invalid    |     | 1.1                          | 0.9 | ns   |
| 9   | $t_d(CLKH-DV)$       | Delay time, EMB_CLK rising to EMB_D[31:0] valid                            |     | 5.1                          | 5.1 | ns   |
| 10  | $t_{oh}(CLKH-DIV)$   | Output hold time, EMB_CLK rising to EMB_D[31:0] invalid                    |     | 1.1                          | 0.9 | ns   |
| 11  | $t_d(CLKH-RASV)$     | Delay time, EMB_CLK rising to $\overline{EMB\_RAS}$ valid                  |     | 5.1                          | 5.1 | ns   |
| 12  | $t_{oh}(CLKH-RASIV)$ | Output hold time, EMB_CLK rising to $\overline{EMB\_RAS}$ invalid          |     | 1.1                          | 0.9 | ns   |
| 13  | $t_d(CLKH-CASV)$     | Delay time, EMB_CLK rising to $\overline{EMB\_CAS}$ valid                  |     | 5.1                          | 5.1 | ns   |
| 14  | $t_{oh}(CLKH-CASIV)$ | Output hold time, EMB_CLK rising to $\overline{EMB\_CAS}$ invalid          |     | 1.1                          | 0.9 | ns   |
| 15  | $t_d(CLKH-WEV)$      | Delay time, EMB_CLK rising to $\overline{EMB\_WE}$ valid                   |     | 5.1                          | 5.1 | ns   |
| 16  | $t_{oh}(CLKH-WEIV)$  | Output hold time, EMB_CLK rising to $\overline{EMB\_WE}$ invalid           |     | 1.1                          | 0.9 | ns   |
| 17  | $t_{dis}(CLKH-DHZ)$  | Delay time, EMB_CLK rising to EMB_D[31:0] tri-stated                       |     | 5.1                          | 5.1 | ns   |
| 18  | $t_{ena}(CLKH-DLZ)$  | Output hold time, EMB_CLK rising to EMB_D[31:0] driving                    |     | 1.1                          | 0.9 | ns   |

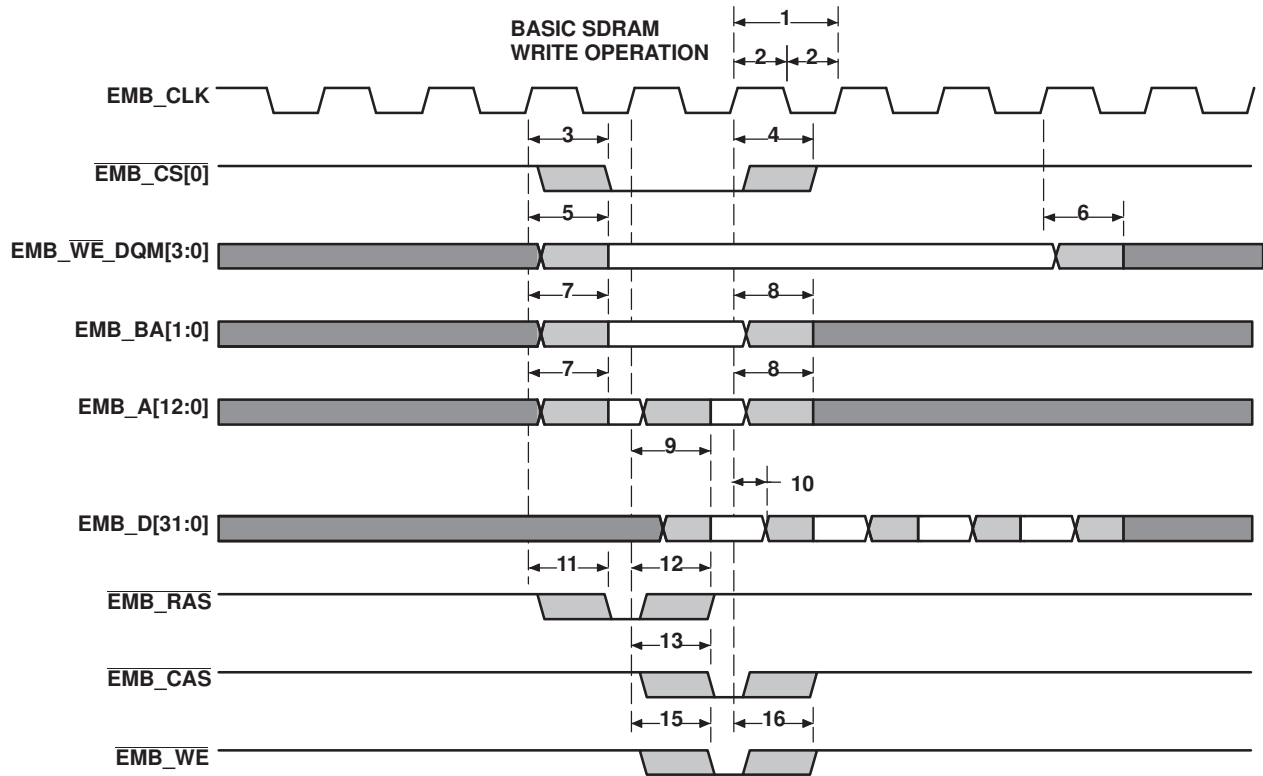


Figure 6-19. EMIFB Basic SDRAM Write Operation

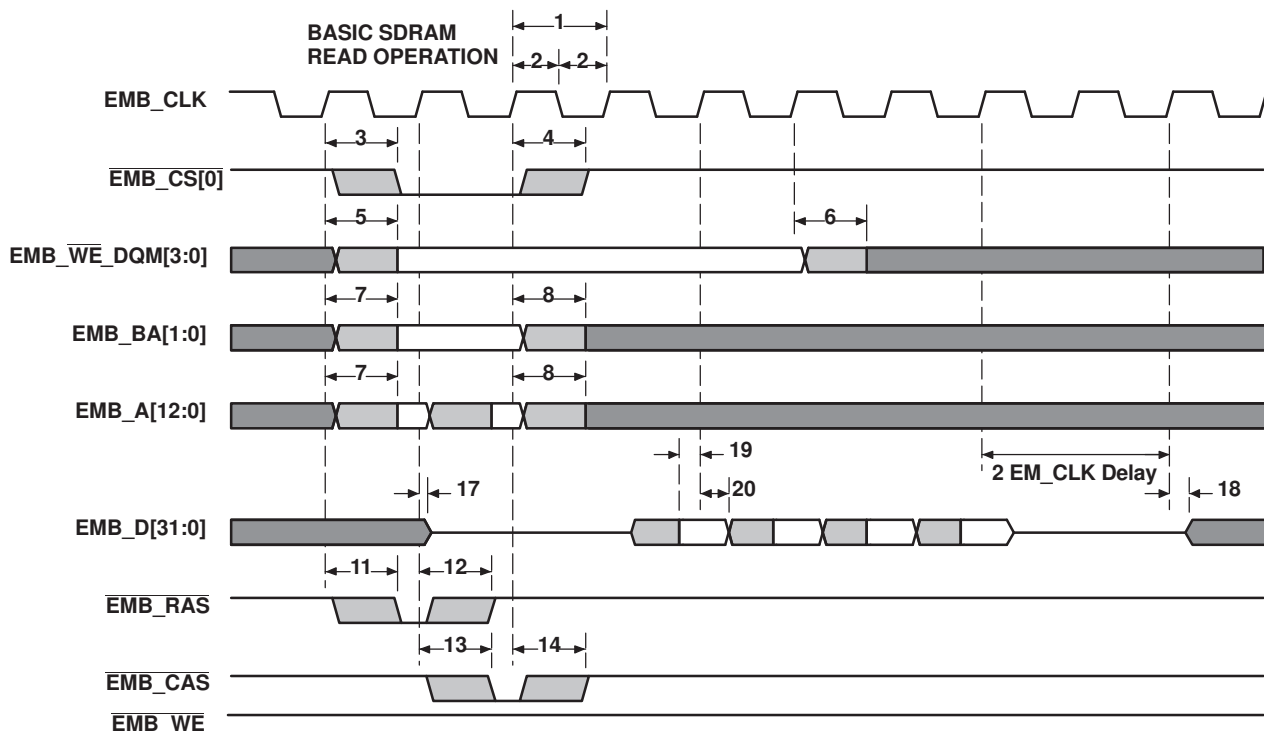


Figure 6-20. EMIFB Basic SDRAM Read Operation

## 6.12 Memory Protection Units

The MPU performs memory protection checking. It receives requests from a bus master in the system and checks the address against the fixed and programmable regions to see if the access is allowed. If allowed, the transfer is passed unmodified to its output bus (to the targeted address). If the transfer is illegal (fails the protection check) then the MPU does not pass the transfer to the output bus but rather services the transfer internally back to the input bus (to prevent a hang) returning the fault status to the requestor as well as generating an interrupt about the fault. The following features are supported by the MPU:

- Provides memory protection for fixed and programmable address ranges.
- Supports multiple programmable address region.
- Supports secure and debug access privileges.
- Supports read, write, and execute access privileges.
- Supports privid(8) associations with ranges.
- Generates an interrupt when there is a protection violation, and saves violating transfer parameters.
- MMR access is also protected.

**Table 6-26. MPU1 Registers**

| BYTE ADDRESS              | REGISTER NAME | REGISTER DESCRIPTION                                    |
|---------------------------|---------------|---|
| 0x01E1 4000               | REVID         | Revision ID   |
| 0x01E1 4004               | CONFIG        | Configuration   |
| 0x01E1 4010               | IRAWSTAT      | Interrupt raw status/set                                |
| 0x01E1 4014               | IENSTAT       | Interrupt enable status/clear                           |
| 0x01E1 4018               | IENSET        | Interrupt enable  |
| 0x01E1 401C               | IENCLR        | Interrupt enable clear                                  |
| 0x01E1 4020 - 0x01E1 41FF | -             | Reserved  |
| 0x01E1 4200               | PROG1_MPSAR   | Programmable range 1, start address                     |
| 0x01E1 4204               | PROG1_MPEAR   | Programmable range 1, end address                       |
| 0x01E1 4208               | PROG1_MPPA    | Programmable range 1, memory page protection attributes |
| 0x01E1 420C - 0x01E1 420F | -             | Reserved  |
| 0x01E1 4210               | PROG2_MPSAR   | Programmable range 2, start address                     |
| 0x01E1 4214               | PROG2_MPEAR   | Programmable range 2, end address                       |
| 0x01E1 4218               | PROG2_MPPA    | Programmable range 2, memory page protection attributes |
| 0x01E1 421C - 0x01E1 421F | -             | Reserved  |
| 0x01E1 4220               | PROG3_MPSAR   | Programmable range 3, start address                     |
| 0x01E1 4224               | PROG3_MPEAR   | Programmable range 3, end address                       |
| 0x01E1 4228               | PROG3_MPPA    | Programmable range 3, memory page protection attributes |
| 0x01E1 422C - 0x01E1 422F | -             | Reserved  |
| 0x01E1 4230               | PROG4_MPSAR   | Programmable range 4, start address                     |
| 0x01E1 4234               | PROG4_MPEA    | Programmable range 4, end address                       |
| 0x01E1 4238               | PROG4_MPPA    | Programmable range 4, memory page protection attributes |
| 0x01E1 423C - 0x01E1 423F | -             | Reserved  |
| 0x01E1 4240               | PROG5_MPSAR   | Programmable range 5, start address                     |
| 0x01E1 4244               | PROG5_MPEAR   | Programmable range 5, end address                       |
| 0x01E1 4248               | PROG5_MPPA    | Programmable range 5, memory page protection attributes |
| 0x01E1 424C - 0x01E1 424F | -             | Reserved  |
| 0x01E1 4250               | PROG6_MPSAR   | Programmable range 6, start address                     |
| 0x01E1 4254               | PROG6_MPEAR   | Programmable range 6, end address                       |
| 0x01E1 4258               | PROG6_MPPA    | Programmable range 6, memory page protection attributes |
| 0x01E1 425C - 0x01E1 42FF | -             | Reserved  |
| 0x01E1 4300               | FLTADDRR      | Fault address   |
| 0x01E1 4304               | FLTSTAT       | Fault status  |

**Table 6-26. MPU1 Registers (continued)**

| BYTE ADDRESS              | REGISTER NAME | REGISTER DESCRIPTION |
|---------------------------|---------------|----------------------|
| 0x01E1 4308               | FLTCLR        | Fault clear          |
| 0x01E1 430C - 0x01E1 4FFF | -             | Reserved             |

**Table 6-27. MPU2 Registers**

| BYTE ADDRESS              | REGISTER NAME | REGISTER DESCRIPTION                                    |
|---------------------------|---------------|---|
| 0x01E1 5000               | REVID         | Revision ID   |
| 0x01E1 5004               | CONFIG        | Configuration   |
| 0x01E1 5010               | IRAWSTAT      | Interrupt raw status/set                                |
| 0x01E1 5014               | IENSTAT       | Interrupt enable status/clear                           |
| 0x01E1 5018               | IENSET        | Interrupt enable  |
| 0x01E1 501C               | IENCLR        | Interrupt enable clear                                  |
| 0x01E1 5020 - 0x01E1 51FF | -             | Reserved  |
| 0x01E1 5200               | PROG1_MPSAR   | Programmable range 1, start address                     |
| 0x01E1 5204               | PROG1_MPEAR   | Programmable range 1, end address                       |
| 0x01E1 5208               | PROG1_MPPA    | Programmable range 1, memory page protection attributes |
| 0x01E1 520C - 0x01E1 520F | -             | Reserved  |
| 0x01E1 5210               | PROG2_MPSAR   | Programmable range 2, start address                     |
| 0x01E1 5214               | PROG2_MPEAR   | Programmable range 2, end address                       |
| 0x01E1 5218               | PROG2_MPPA    | Programmable range 2, memory page protection attributes |
| 0x01E1 521C - 0x01E1 521F | -             | Reserved  |
| 0x01E1 5220               | PROG3_MPSAR   | Programmable range 3, start address                     |
| 0x01E1 5224               | PROG3_MPEAR   | Programmable range 3, end address                       |
| 0x01E1 5228               | PROG3_MPPA    | Programmable range 3, memory page protection attributes |
| 0x01E1 522C - 0x01E1 522F | -             | Reserved  |
| 0x01E1 5230               | PROG4_MPSAR   | Programmable range 4, start address                     |
| 0x01E1 5234               | PROG4_MPEA    | Programmable range 4, end address                       |
| 0x01E1 5238               | PROG4_MPPA    | Programmable range 4, memory page protection attributes |
| 0x01E1 523C - 0x01E1 523F | -             | Reserved  |
| 0x01E1 5240               | PROG5_MPSAR   | Programmable range 5, start address                     |
| 0x01E1 5244               | PROG5_MPEAR   | Programmable range 5, end address                       |
| 0x01E1 5248               | PROG5_MPPA    | Programmable range 5, memory page protection attributes |
| 0x01E1 524C - 0x01E1 524F | -             | Reserved  |
| 0x01E1 5250               | PROG6_MPSAR   | Programmable range 6, start address                     |
| 0x01E1 5254               | PROG6_MPEAR   | Programmable range 6, end address                       |
| 0x01E1 5258               | PROG6_MPPA    | Programmable range 6, memory page protection attributes |
| 0x01E1 525C - 0x01E1 525F | -             | Reserved  |
| 0x01E1 5260               | PROG7_MPSAR   | Programmable range 7, start address                     |
| 0x01E1 5264               | PROG7_MPEAR   | Programmable range 7, end address                       |
| 0x01E1 5268               | PROG7_MPPA    | Programmable range 7, memory page protection attributes |
| 0x01E1 526C - 0x01E1 526F | -             | Reserved  |
| 0x01E1 5270               | PROG8_MPSAR   | Programmable range 8, start address                     |
| 0x01E1 5274               | PROG8_MPEAR   | Programmable range 8, end address                       |
| 0x01E1 5278               | PROG8_MPPA    | Programmable range 8, memory page protection attributes |
| 0x01E1 527C - 0x01E1 527F | -             | Reserved  |
| 0x01E1 5280               | PROG9_MPSAR   | Programmable range 9, start address                     |
| 0x01E1 5284               | PROG9_MPEAR   | Programmable range 9, end address                       |
| 0x01E1 5288               | PROG9_MPPA    | Programmable range 9, memory page protection attributes |

**Table 6-27. MPU2 Registers (continued)**

| BYTE ADDRESS              | REGISTER NAME | REGISTER DESCRIPTION                                     |
|---------------------------|---------------|--|
| 0x01E1 528C - 0x01E1 528F | -             | Reserved   |
| 0x01E1 5290               | PROG10_MPSAR  | Programmable range 10, start address                     |
| 0x01E1 5294               | PROG10_MPEAR  | Programmable range 10, end address                       |
| 0x01E1 5298               | PROG10_MPPA   | Programmable range 10, memory page protection attributes |
| 0x01E1 529C - 0x01E1 529F | -             | Reserved   |
| 0x01E1 52A0               | PROG11_MPSAR  | Programmable range 11, start address                     |
| 0x01E1 52A4               | PROG11_MPEAR  | Programmable range 11, end address                       |
| 0x01E1 52A8               | PROG11_MPPA   | Programmable range 11, memory page protection attributes |
| 0x01E1 52AC - 0x01E1 52AF | -             | Reserved   |
| 0x01E1 52B0               | PROG12_MPSAR  | Programmable range 12, start address                     |
| 0x01E1 52B4               | PROG12_MPEAR  | Programmable range 12, end address                       |
| 0x01E1 52B8               | PROG12_MPPA   | Programmable range 12, memory page protection attributes |
| 0x01E1 52BC - 0x01E1 52FF | -             | Reserved   |
| 0x01E1 5300               | FLTADDRR      | Fault address  |
| 0x01E1 5304               | FLTSTAT       | Fault status   |
| 0x01E1 5308               | FLTCLR        | Fault clear  |
| 0x01E1 530C - 0x01E1 5FFF | -             | Reserved   |

## 6.13 MMC / SD / SDIO (MMCSD)

The C6743 includes an MMCSD controller which is compliant with MMC V3.31, Secure Digital Part 1 Physical Layer Specification V1.1 and Secure Digital Input Output (SDIO) V2.0 specifications. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

The MMC/SD Controller has following features:

- MultiMediaCard (MMC)
- Secure Digital (SD) Memory Card
- MMC/SD protocol support
- SD high capacity support
- SDIO protocol support
- Programmable clock frequency
- 512 bit Read/Write FIFO to lower system overhead
- Slave EDMA transfer capability

The C6743 MMC/SD Controller does not support SPI mode.

### 6.13.1 MMCSD Peripheral Register Description(s)

**Table 6-28. Multimedia Card/Secure Digital (MMC/SD) Card Controller Registers**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                  |
|--------------|---------------|---------------------------------------|
| 0x01C4 0000  | MMCCCTL       | MMC Control Register                  |
| 0x01C4 0004  | MMCCCLK       | MMC Memory Clock Control Register     |
| 0x01C4 0008  | MMCCST0       | MMC Status Register 0                 |
| 0x01C4 000C  | MMCCST1       | MMC Status Register 1                 |
| 0x01C4 0010  | MMCCIM        | MMC Interrupt Mask Register           |
| 0x01C4 0014  | MMCCTOR       | MMC Response Time-Out Register        |
| 0x01C4 0018  | MMCC TOD      | MMC Data Read Time-Out Register       |
| 0x01C4 001C  | MMCCBLEN      | MMC Block Length Register             |
| 0x01C4 0020  | MMCCNBLK      | MMC Number of Blocks Register         |
| 0x01C4 0024  | MMCCNBLC      | MMC Number of Blocks Counter Register |
| 0x01C4 0028  | MMCCDRR       | MMC Data Receive Register             |
| 0x01C4 002C  | MMCCDXR       | MMC Data Transmit Register            |
| 0x01C4 0030  | MMCCCMD       | MMC Command Register                  |
| 0x01C4 0034  | MMCCARGHL     | MMC Argument Register                 |
| 0x01C4 0038  | MMCCRSP01     | MMC Response Register 0 and 1         |
| 0x01C4 003C  | MMCCRSP23     | MMC Response Register 2 and 3         |
| 0x01C4 0040  | MMCCRSP45     | MMC Response Register 4 and 5         |
| 0x01C4 0044  | MMCCRSP67     | MMC Response Register 6 and 7         |
| 0x01C4 0048  | MMCCDRSP      | MMC Data Response Register            |
| 0x01C4 0050  | MMCCIDX       | MMC Command Index Register            |
| 0x01C4 0064  | SDIOCTL       | SDIO Control Register                 |
| 0x01C4 0068  | SDIOST0       | SDIO Status Register 0                |
| 0x01C4 006C  | SDIOIEN       | SDIO Interrupt Enable Register        |
| 0x01C4 0070  | SDIOIST       | SDIO Interrupt Status Register        |
| 0x01C4 0074  | MMCCFIFOCTL   | MMC FIFO Control Register             |

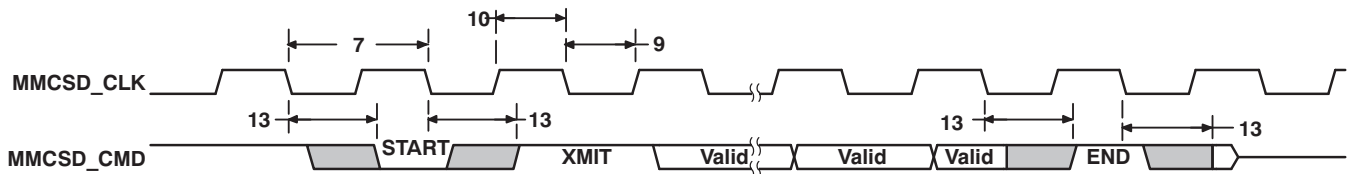
### 6.13.2 MMC/SD Electrical Data/Timing

**Table 6-29. Timing Requirements for MMC/SD Module**  
(see [Figure 6-22](#) and [Figure 6-24](#))

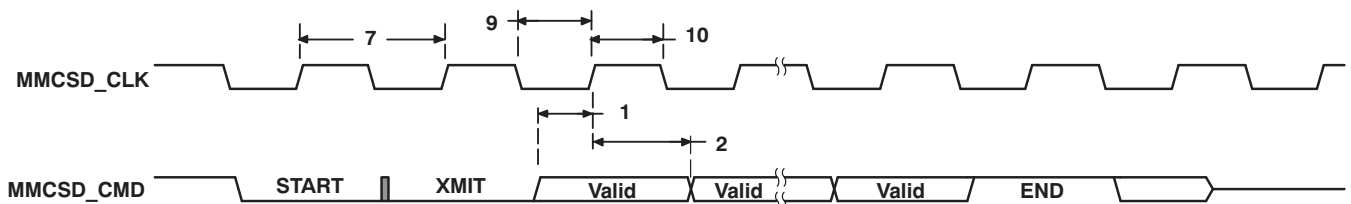
| NO. |                     |  | MIN | MAX | UNIT |
|-----|---------------------|--|-----|-----|------|
| 1   | $t_{su}(CMDV-CLKH)$ | Setup time, MMCSD_CMD valid before MMCSD_CLK high  | 3.2 |     | ns   |
| 2   | $t_h(CLKH-CMDV)$    | Hold time, MMCSD_CMD valid after MMCSD_CLK high    | 1.5 |     | ns   |
| 3   | $t_{su}(DATV-CLKH)$ | Setup time, MMCSD_DATx valid before MMCSD_CLK high | 3.2 |     | ns   |
| 4   | $t_h(CLKH-DATV)$    | Hold time, MMCSD_DATx valid after MMCSD_CLK high   | 1.5 |     | ns   |

**Table 6-30. Switching Characteristics Over Recommended Operating Conditions for MMC/SD Module**  
(see [Figure 6-21](#) through [Figure 6-24](#))

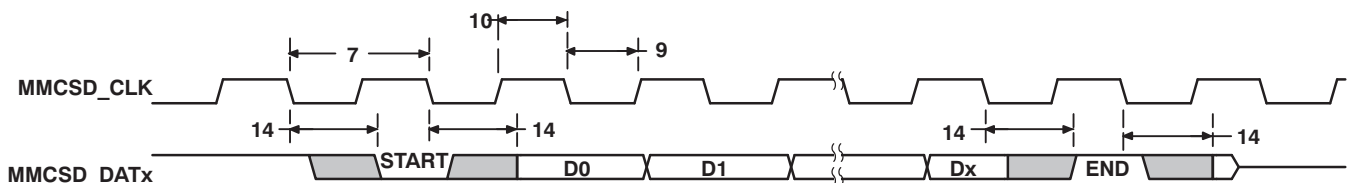
| NO. | PARAMETER       | MIN  | MAX | UNIT |
|-----|-----------------|------|-----|------|
| 7   | $f_{(CLK)}$     | 0    | 52  | ns   |
| 8   | $f_{(CLK\_ID)}$ | 0    | 400 | ns   |
| 9   | $t_{W(CLKL)}$   | 6.5  |     | ns   |
| 10  | $t_{W(CLKH)}$   | 6.5  |     | ns   |
| 11  | $t_r(CLK)$      |      | 3   | ns   |
| 12  | $t_f(CLK)$      |      | 3   | ns   |
| 13  | $t_d(CLKL-CMD)$ | -4.5 | 2.5 | ns   |
| 14  | $t_d(CLKL-DAT)$ | -4.5 | 2.5 | ns   |



**Figure 6-21. MMC/SD Host Command Timing**



**Figure 6-22. MMC/SD Card Response Timing**



**Figure 6-23. MMC/SD Host Write Timing**

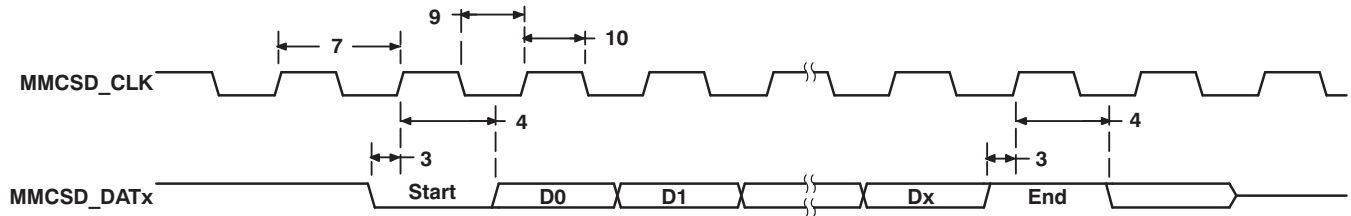


Figure 6-24. MMC/SD Host Read and Card CRC Status Timing

## 6.14 Ethernet Media Access Controller (EMAC)

The Ethernet Media Access Controller (EMAC) provides an efficient interface between C6743 and the network. The EMAC supports both 10Base-T and 100Base-TX, or 10 Mbps/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 C6743 device to the PHY. The MDIO module controls PHY configuration and status monitoring.

Both the EMAC and the MDIO modules interface to the C6743 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. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide* (SPRUFK9) for more details.

### 6.14.1 EMAC Peripheral Register Description(s)

**Table 6-31. Ethernet Media Access Controller (EMAC) Registers**

| BYTE ADDRESS | REGISTER NAME     | REGISTER DESCRIPTION  |
|--------------|-------------------|---|
| 0x01E2 3000  | TXREV             | Transmit Revision Register                                      |
| 0x01E2 3004  | TXCONTROL         | Transmit Control Register                                       |
| 0x01E2 3008  | TXTEARDOWN        | Transmit Teardown Register                                      |
| 0x01E2 3010  | RXREV             | Receive Revision Register                                       |
| 0x01E2 3014  | RXCONTROL         | Receive Control Register  |
| 0x01E2 3018  | RXTEARDOWN        | Receive Teardown Register                                       |
| 0x01E2 3080  | TXINTSTATRAW      | Transmit Interrupt Status (Unmasked) Register                   |
| 0x01E2 3084  | TXINTSTATMASKED   | Transmit Interrupt Status (Masked) Register                     |
| 0x01E2 3088  | TXINTMASKSET      | Transmit Interrupt Mask Set Register                            |
| 0x01E2 308C  | TXINTMASKCLEAR    | Transmit Interrupt Clear Register                               |
| 0x01E2 3090  | MACINVECTOR       | MAC Input Vector Register                                       |
| 0x01E2 3094  | MACEOIVECTOR      | MAC End Of Interrupt Vector Register                            |
| 0x01E2 30A0  | RXINTSTATRAW      | Receive Interrupt Status (Unmasked) Register                    |
| 0x01E2 30A4  | RXINTSTATMASKED   | Receive Interrupt Status (Masked) Register                      |
| 0x01E2 30A8  | RXINTMASKSET      | Receive Interrupt Mask Set Register                             |
| 0x01E2 30AC  | RXINTMASKCLEAR    | Receive Interrupt Mask Clear Register                           |
| 0x01E2 30B0  | MACINTSTATRAW     | MAC Interrupt Status (Unmasked) Register                        |
| 0x01E2 30B4  | MACINTSTATMASKED  | MAC Interrupt Status (Masked) Register                          |
| 0x01E2 30B8  | MACINTMASKSET     | MAC Interrupt Mask Set Register                                 |
| 0x01E2 30BC  | MACINTMASKCLEAR   | MAC Interrupt Mask Clear Register                               |
| 0x01E2 3100  | RXMBPENABLE       | Receive Multicast/Broadcast/Promiscuous Channel Enable Register |
| 0x01E2 3104  | RXUNICASTSET      | Receive Unicast Enable Set Register                             |
| 0x01E2 3108  | RXUNICASTCLEAR    | Receive Unicast Clear Register                                  |
| 0x01E2 310C  | RXMAXLEN          | Receive Maximum Length Register                                 |
| 0x01E2 3110  | RXBUFFEROFFSET    | Receive Buffer Offset Register                                  |
| 0x01E2 3114  | RXFILTERLOWTHRESH | Receive Filter Low Priority Frame Threshold Register            |
| 0x01E2 3120  | RX0FLOWTHRESH     | Receive Channel 0 Flow Control Threshold Register               |
| 0x01E2 3124  | RX1FLOWTHRESH     | Receive Channel 1 Flow Control Threshold Register               |
| 0x01E2 3128  | RX2FLOWTHRESH     | Receive Channel 2 Flow Control Threshold Register               |
| 0x01E2 312C  | RX3FLOWTHRESH     | Receive Channel 3 Flow Control Threshold Register               |
| 0x01E2 3130  | RX4FLOWTHRESH     | Receive Channel 4 Flow Control Threshold Register               |
| 0x01E2 3134  | RX5FLOWTHRESH     | Receive Channel 5 Flow Control Threshold Register               |
| 0x01E2 3138  | RX6FLOWTHRESH     | Receive Channel 6 Flow Control Threshold Register               |

**Table 6-31. Ethernet Media Access Controller (EMAC) Registers (continued)**

| BYTE ADDRESS              | REGISTER NAME                     | REGISTER DESCRIPTION  |
|---------------------------|-----------------------------------|---|
| 0x01E2 313C               | RX7FLOWTHRESH                     | Receive Channel 7 Flow Control Threshold Register                 |
| 0x01E2 3140               | RX0FREEBUFFER                     | Receive Channel 0 Free Buffer Count Register                      |
| 0x01E2 3144               | RX1FREEBUFFER                     | Receive Channel 1 Free Buffer Count Register                      |
| 0x01E2 3148               | RX2FREEBUFFER                     | Receive Channel 2 Free Buffer Count Register                      |
| 0x01E2 314C               | RX3FREEBUFFER                     | Receive Channel 3 Free Buffer Count Register                      |
| 0x01E2 3150               | RX4FREEBUFFER                     | Receive Channel 4 Free Buffer Count Register                      |
| 0x01E2 3154               | RX5FREEBUFFER                     | Receive Channel 5 Free Buffer Count Register                      |
| 0x01E2 3158               | RX6FREEBUFFER                     | Receive Channel 6 Free Buffer Count Register                      |
| 0x01E2 315C               | RX7FREEBUFFER                     | Receive Channel 7 Free Buffer Count Register                      |
| 0x01E2 3160               | MACCONTROL                        | MAC Control Register  |
| 0x01E2 3164               | MACSTATUS                         | MAC Status Register   |
| 0x01E2 3168               | EMCONTROL                         | Emulation Control Register  |
| 0x01E2 316C               | FIFOCONTROL                       | FIFO Control Register   |
| 0x01E2 3170               | MACCONFIG                         | MAC Configuration Register  |
| 0x01E2 3174               | SOFTRESET                         | Soft Reset Register   |
| 0x01E2 31D0               | MACSRCADDRLO                      | MAC Source Address Low Bytes Register                             |
| 0x01E2 31D4               | MACSRCADDRHI                      | MAC Source Address High Bytes Register                            |
| 0x01E2 31D8               | MACHASH1                          | MAC Hash Address Register 1                                       |
| 0x01E2 31DC               | MACHASH2                          | MAC Hash Address Register 2                                       |
| 0x01E2 31E0               | BOFFTEST                          | Back Off Test Register  |
| 0x01E2 31E4               | TPACETEST                         | Transmit Pacing Algorithm Test Register                           |
| 0x01E2 31E8               | RXPAUSE                           | Receive Pause Timer Register                                      |
| 0x01E2 31EC               | TXPAUSE                           | Transmit Pause Timer Register                                     |
| 0x01E2 3200 - 0x01E2 32FC | (see <a href="#">Table 6-32</a> ) | EMAC Statistics Registers   |
| 0x01E2 3500               | MACADDRLO                         | MAC Address Low Bytes Register, Used in Receive Address Matching  |
| 0x01E2 3504               | MACADDRHI                         | MAC Address High Bytes Register, Used in Receive Address Matching |
| 0x01E2 3508               | MACINDEX                          | MAC Index Register  |
| 0x01E2 3600               | TX0HDP                            | Transmit Channel 0 DMA Head Descriptor Pointer Register           |
| 0x01E2 3604               | TX1HDP                            | Transmit Channel 1 DMA Head Descriptor Pointer Register           |
| 0x01E2 3608               | TX2HDP                            | Transmit Channel 2 DMA Head Descriptor Pointer Register           |
| 0x01E2 360C               | TX3HDP                            | Transmit Channel 3 DMA Head Descriptor Pointer Register           |
| 0x01E2 3610               | TX4HDP                            | Transmit Channel 4 DMA Head Descriptor Pointer Register           |
| 0x01E2 3614               | TX5HDP                            | Transmit Channel 5 DMA Head Descriptor Pointer Register           |
| 0x01E2 3618               | TX6HDP                            | Transmit Channel 6 DMA Head Descriptor Pointer Register           |
| 0x01E2 361C               | TX7HDP                            | Transmit Channel 7 DMA Head Descriptor Pointer Register           |
| 0x01E2 3620               | RX0HDP                            | Receive Channel 0 DMA Head Descriptor Pointer Register            |
| 0x01E2 3624               | RX1HDP                            | Receive Channel 1 DMA Head Descriptor Pointer Register            |
| 0x01E2 3628               | RX2HDP                            | Receive Channel 2 DMA Head Descriptor Pointer Register            |
| 0x01E2 362C               | RX3HDP                            | Receive Channel 3 DMA Head Descriptor Pointer Register            |
| 0x01E2 3630               | RX4HDP                            | Receive Channel 4 DMA Head Descriptor Pointer Register            |
| 0x01E2 3634               | RX5HDP                            | Receive Channel 5 DMA Head Descriptor Pointer Register            |
| 0x01E2 3638               | RX6HDP                            | Receive Channel 6 DMA Head Descriptor Pointer Register            |
| 0x01E2 363C               | RX7HDP                            | Receive Channel 7 DMA Head Descriptor Pointer Register            |
| 0x01E2 3640               | TX0CP                             | Transmit Channel 0 Completion Pointer Register                    |
| 0x01E2 3644               | TX1CP                             | Transmit Channel 1 Completion Pointer Register                    |
| 0x01E2 3648               | TX2CP                             | Transmit Channel 2 Completion Pointer Register                    |
| 0x01E2 364C               | TX3CP                             | Transmit Channel 3 Completion Pointer Register                    |

**Table 6-31. Ethernet Media Access Controller (EMAC) Registers (continued)**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                           |
|--------------|---------------|--|
| 0x01E2 3650  | TX4CP         | Transmit Channel 4 Completion Pointer Register |
| 0x01E2 3654  | TX5CP         | Transmit Channel 5 Completion Pointer Register |
| 0x01E2 3658  | TX6CP         | Transmit Channel 6 Completion Pointer Register |
| 0x01E2 365C  | TX7CP         | Transmit Channel 7 Completion Pointer Register |
| 0x01E2 3660  | RX0CP         | Receive Channel 0 Completion Pointer Register  |
| 0x01E2 3664  | RX1CP         | Receive Channel 1 Completion Pointer Register  |
| 0x01E2 3668  | RX2CP         | Receive Channel 2 Completion Pointer Register  |
| 0x01E2 366C  | RX3CP         | Receive Channel 3 Completion Pointer Register  |
| 0x01E2 3670  | RX4CP         | Receive Channel 4 Completion Pointer Register  |
| 0x01E2 3674  | RX5CP         | Receive Channel 5 Completion Pointer Register  |
| 0x01E2 3678  | RX6CP         | Receive Channel 6 Completion Pointer Register  |
| 0x01E2 367C  | RX7CP         | Receive Channel 7 Completion Pointer Register  |

**Table 6-32. EMAC Statistics Registers**

| BYTE ADDRESS | REGISTER NAME     | REGISTER DESCRIPTION   |
|--------------|-------------------|--|
| 0x01E2 3200  | RXGOODFRAMES      | Good Receive Frames Register   |
| 0x01E2 3204  | RXBCASTFRAMES     | Broadcast Receive Frames Register<br>(Total number of good broadcast frames received)                  |
| 0x01E2 3208  | RXMCASTFRAMES     | Multicast Receive Frames Register<br>(Total number of good multicast frames received)                  |
| 0x01E2 320C  | RXPAUSEFRAMES     | Pause Receive Frames Register  |
| 0x01E2 3210  | RXCRCERRORS       | Receive CRC Errors Register<br>(Total number of frames received with CRC errors)                       |
| 0x01E2 3214  | RXALIGNCODEERRORS | Receive Alignment/Code Errors Register<br>(Total number of frames received with alignment/code errors) |
| 0x01E2 3218  | RXOVERSIZED       | Receive Oversized Frames Register<br>(Total number of oversized frames received)                       |
| 0x01E2 321C  | RXJABBER          | Receive Jabber Frames Register<br>(Total number of jabber frames received)                             |
| 0x01E2 3220  | RXUNDERSIZED      | Receive Undersized Frames Register<br>(Total number of undersized frames received)                     |
| 0x01E2 3224  | RXFRAGMENTS       | Receive Frame Fragments Register   |
| 0x01E2 3228  | RXFILTERED        | Filtered Receive Frames Register   |
| 0x01E2 322C  | RXQOSFILTERED     | Received QOS Filtered Frames Register  |
| 0x01E2 3230  | RXOCTETS          | Receive Octet Frames Register<br>(Total number of received bytes in good frames)                       |
| 0x01E2 3234  | TXGOODFRAMES      | Good Transmit Frames Register<br>(Total number of good frames transmitted)                             |
| 0x01E2 3238  | TXBCASTFRAMES     | Broadcast Transmit Frames Register   |
| 0x01E2 323C  | TXMCASTFRAMES     | Multicast Transmit Frames Register   |
| 0x01E2 3240  | TXPAUSEFRAMES     | Pause Transmit Frames Register   |
| 0x01E2 3244  | TXDEFERRED        | Deferred Transmit Frames Register  |
| 0x01E2 3248  | TXCOLLISION       | Transmit Collision Frames Register   |
| 0x01E2 324C  | TXSINGLECOLL      | Transmit Single Collision Frames Register  |
| 0x01E2 3250  | TXMULTICOLL       | Transmit Multiple Collision Frames Register  |
| 0x01E2 3254  | TXEXCESSIVECOLL   | Transmit Excessive Collision Frames Register   |
| 0x01E2 3258  | TXLATECOLL        | Transmit Late Collision Frames Register  |
| 0x01E2 325C  | TXUNDERRUN        | Transmit Underrun Error Register   |
| 0x01E2 3260  | TXCARRIERSENSE    | Transmit Carrier Sense Errors Register   |
| 0x01E2 3264  | TXOCTETS          | Transmit Octet Frames Register   |

**Table 6-32. EMAC Statistics Registers (continued)**

| BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION   |
|--------------|---------------|--|
| 0x01E2 3268  | FRAME64       | Transmit and Receive 64 Octet Frames Register                    |
| 0x01E2 326C  | FRAME65T127   | Transmit and Receive 65 to 127 Octet Frames Register             |
| 0x01E2 3270  | FRAME128T255  | Transmit and Receive 128 to 255 Octet Frames Register            |
| 0x01E2 3274  | FRAME256T511  | Transmit and Receive 256 to 511 Octet Frames Register            |
| 0x01E2 3278  | FRAME512T1023 | Transmit and Receive 512 to 1023 Octet Frames Register           |
| 0x01E2 327C  | FRAME1024TUP  | Transmit and Receive 1024 to 1518 Octet Frames Register          |
| 0x01E2 3280  | NETOCTETS     | Network Octet Frames Register                                    |
| 0x01E2 3284  | RXSOFOVERRUNS | Receive FIFO or DMA Start of Frame Overruns Register             |
| 0x01E2 3288  | RXMOFOVERRUNS | Receive FIFO or DMA Middle of Frame Overruns Register            |
| 0x01E2 328C  | RXDMAOVERRUNS | Receive DMA Start of Frame and Middle of Frame Overruns Register |

**Table 6-33. EMAC Control Module Registers**

| BYTE ADDRESS | REGISTER NAME  | REGISTER DESCRIPTION  |
|--------------|----------------|---|
| 0x01E2 2000  | REV            | EMAC Control Module Revision Register   |
| 0x01E2 2004  | SOFTRESET      | EMAC Control Module Software Reset Register                                       |
| 0x01E2 200C  | INTCONTROL     | EMAC Control Module Interrupt Control Register                                    |
| 0x01E2 2010  | C0RXTHRESHEN   | EMAC Control Module Interrupt Core 0 Receive Threshold Interrupt Enable Register  |
| 0x01E2 2014  | C0RXEN         | EMAC Control Module Interrupt Core 0 Receive Interrupt Enable Register            |
| 0x01E2 2018  | C0TXEN         | EMAC Control Module Interrupt Core 0 Transmit Interrupt Enable Register           |
| 0x01E2 201C  | C0MISCEN       | EMAC Control Module Interrupt Core 0 Miscellaneous Interrupt Enable Register      |
| 0x01E2 2020  | C1RXTHRESHEN   | EMAC Control Module Interrupt Core 1 Receive Threshold Interrupt Enable Register  |
| 0x01E2 2024  | C1RXEN         | EMAC Control Module Interrupt Core 1 Receive Interrupt Enable Register            |
| 0x01E2 2028  | C1TXEN         | EMAC Control Module Interrupt Core 1 Transmit Interrupt Enable Register           |
| 0x01E2 202C  | C1MISCEN       | EMAC Control Module Interrupt Core 1 Miscellaneous Interrupt Enable Register      |
| 0x01E2 2030  | C2RXTHRESHEN   | EMAC Control Module Interrupt Core 2 Receive Threshold Interrupt Enable Register  |
| 0x01E2 2034  | C2RXEN         | EMAC Control Module Interrupt Core 2 Receive Interrupt Enable Register            |
| 0x01E2 2038  | C2TXEN         | EMAC Control Module Interrupt Core 2 Transmit Interrupt Enable Register           |
| 0x01E2 203C  | C2MISCEN       | EMAC Control Module Interrupt Core 2 Miscellaneous Interrupt Enable Register      |
| 0x01E2 2040  | C0RXTHRESHSTAT | EMAC Control Module Interrupt Core 0 Receive Threshold Interrupt Status Register  |
| 0x01E2 2044  | C0RXSTAT       | EMAC Control Module Interrupt Core 0 Receive Interrupt Status Register            |
| 0x01E2 2048  | C0TXSTAT       | EMAC Control Module Interrupt Core 0 Transmit Interrupt Status Register           |
| 0x01E2 204C  | C0MISCSTAT     | EMAC Control Module Interrupt Core 0 Miscellaneous Interrupt Status Register      |
| 0x01E2 2050  | C1RXTHRESHSTAT | EMAC Control Module Interrupt Core 1 Receive Threshold Interrupt Status Register  |
| 0x01E2 2054  | C1RXSTAT       | EMAC Control Module Interrupt Core 1 Receive Interrupt Status Register            |
| 0x01E2 2058  | C1TXSTAT       | EMAC Control Module Interrupt Core 1 Transmit Interrupt Status Register           |
| 0x01E2 205C  | C1MISCSTAT     | EMAC Control Module Interrupt Core 1 Miscellaneous Interrupt Status Register      |
| 0x01E2 2060  | C2RXTHRESHSTAT | EMAC Control Module Interrupt Core 2 Receive Threshold Interrupt Status Register  |
| 0x01E2 2064  | C2RXSTAT       | EMAC Control Module Interrupt Core 2 Receive Interrupt Status Register            |
| 0x01E2 2068  | C2TXSTAT       | EMAC Control Module Interrupt Core 2 Transmit Interrupt Status Register           |
| 0x01E2 206C  | C2MISCSTAT     | EMAC Control Module Interrupt Core 2 Miscellaneous Interrupt Status Register      |
| 0x01E2 2070  | C0RXIMAX       | EMAC Control Module Interrupt Core 0 Receive Interrupts Per Millisecond Register  |
| 0x01E2 2074  | C0TXIMAX       | EMAC Control Module Interrupt Core 0 Transmit Interrupts Per Millisecond Register |
| 0x01E2 2078  | C1RXIMAX       | EMAC Control Module Interrupt Core 1 Receive Interrupts Per Millisecond Register  |
| 0x01E2 207C  | C1TXIMAX       | EMAC Control Module Interrupt Core 1 Transmit Interrupts Per Millisecond Register |
| 0x01E2 2080  | C2RXIMAX       | EMAC Control Module Interrupt Core 2 Receive Interrupts Per Millisecond Register  |
| 0x01E2 2084  | C2TXIMAX       | EMAC Control Module Interrupt Core 2 Transmit Interrupts Per Millisecond Register |

**Table 6-34. EMAC Control Module RAM**

| BYTE ADDRESS              | DESCRIPTION                         |
|---------------------------|-------------------------------------|
| 0x01E2 0000 - 0x01E2 1FFF | EMAC Local Buffer Descriptor Memory |

**6.14.2 EMAC Electrical Data/Timing**

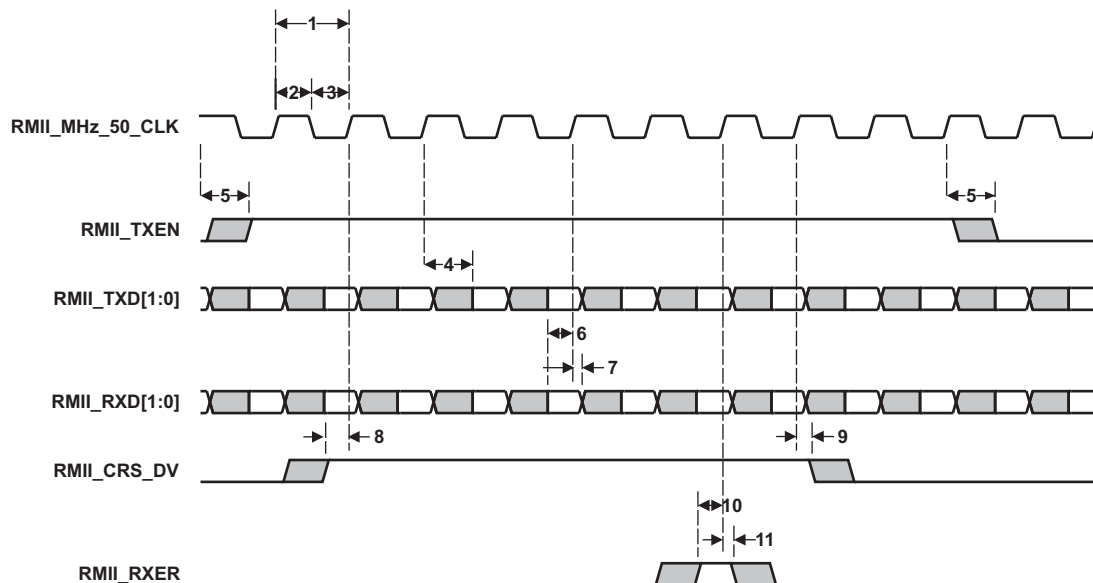
**Table 6-35. RMII Timing Requirements<sup>(1)</sup>**

| NO. |                   |   | MIN | TYP | MAX | UNIT |
|-----|-------------------|---|-----|-----|-----|------|
| 1   | tc(REFCLK)        | Cycle Time, RMII_MHZ_50_CLK                               |     | 20  |     | ns   |
| 2   | tw(REFCLKH)       | Pulse Width, RMII_MHZ_50_CLK High                         | 7   |     | 13  | ns   |
| 3   | tw(REFCLKL)       | Pulse Width, RMII_MHZ_50_CLK Low                          | 7   |     | 13  | ns   |
| 6   | tsu(RXD-REFCLK)   | Input Setup Time, RXD Valid before RMII_MHZ_50_CLK High   | 4   |     |     | ns   |
| 7   | th(REFCLK-RXD)    | Input Hold Time, RXD Valid after RMII_MHZ_50_CLK High     | 2   |     |     | ns   |
| 8   | tsu(CRSDV-REFCLK) | Input Setup Time, CRSDV Valid before RMII_MHZ_50_CLK High | 4   |     |     | ns   |
| 9   | th(REFCLK-CRSDV)  | Input Hold Time, CRSDV Valid after RMII_MHZ_50_CLK High   | 2   |     |     | ns   |
| 10  | tsu(RXER-REFCLK)  | Input Setup Time, RXER Valid before RMII_MHZ_50_CLK High  | 4   |     |     | ns   |
| 11  | th(REFCLK-RXER)   | Input Hold Time, RXER Valid after RMII_MHZ_50_CLK High    | 2   |     |     | ns   |

(1) Per the RMII industry specification, the RMII reference clock (RMII\_MHZ\_50\_CLK) must have jitter tolerance of 50 ppm or less.

**Table 6-36. RMII Switching Characteristics**

| NO. |                 |   | MIN | TYP | MAX | UNIT |
|-----|-----------------|---|-----|-----|-----|------|
| 4   | td(REFCLK-TXD)  | Output Delay Time, RMII_MHZ_50_CLK High to TXD Valid  | 2.5 |     | 13  | ns   |
| 5   | td(REFCLK-TXEN) | Output Delay Time, RMII_MHZ_50_CLK High to TXEN Valid | 2.5 |     | 13  | ns   |



**Figure 6-25. RMII Timing Diagram**

## 6.15 Management Data Input/Output (MDIO)

The Management Data Input/Output (MDIO) module continuously polls all 32 MDIO addresses in order to enumerate all PHY devices in the system.

The Management Data Input/Output (MDIO) module implements the 802.3 serial management interface to interrogate and control Ethernet PHY(s) using a shared two-wire bus. Host software uses the MDIO module to configure the auto-negotiation parameters of each PHY attached to the EMAC, retrieve the negotiation results, and configure required parameters in the EMAC module for correct operation. The module is designed to allow almost transparent operation of the MDIO interface, with very little maintenance from the core processor. Only one PHY may be connected at any given time. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

### 6.15.1 MDIO Peripheral Register Description(s)

For a list of supported MDIO registers see [Table 6-37](#) [MDIO Registers].

**Table 6-37. MDIO Register Memory Map**

| BYTE ADDRESS              | REGISTER NAME    | REGISTER DESCRIPTION                                     |
|---------------------------|------------------|--|
| 0x01E2 4000               | REV              | Revision Identification Register                         |
| 0x01E2 4004               | CONTROL          | MDIO Control Register                                    |
| 0x01E2 4008               | ALIVE            | MDIO PHY Alive Status Register                           |
| 0x01E2 400C               | LINK             | MDIO PHY Link Status Register                            |
| 0x01E2 4010               | LINKINTRAW       | MDIO Link Status Change Interrupt (Unmasked) Register    |
| 0x01E2 4014               | LINKINTMASKED    | MDIO Link Status Change Interrupt (Masked) Register      |
| 0x01E2 4018               | –                | Reserved   |
| 0x01E2 4020               | USERINTRAW       | MDIO User Command Complete Interrupt (Unmasked) Register |
| 0x01E2 4024               | USERINTMASKED    | MDIO User Command Complete Interrupt (Masked) Register   |
| 0x01E2 4028               | USERINTMASKSET   | MDIO User Command Complete Interrupt Mask Set Register   |
| 0x01E2 402C               | USERINTMASKCLEAR | MDIO User Command Complete Interrupt Mask Clear Register |
| 0x01E2 4030 - 0x01E2 407C | –                | Reserved   |
| 0x01E2 4080               | USERACCESS0      | MDIO User Access Register 0                              |
| 0x01E2 4084               | USERPHYSEL0      | MDIO User PHY Select Register 0                          |
| 0x01E2 4088               | USERACCESS1      | MDIO User Access Register 1                              |
| 0x01E2 408C               | USERPHYSEL1      | MDIO User PHY Select Register 1                          |
| 0x01E2 4090 - 0x01E2 47FF | –                | Reserved   |

### 6.15.2 Management Data Input/Output (MDIO) Electrical Data/Timing

**Table 6-38. Timing Requirements for MDIO Input (see [Figure 6-26](#) and [Figure 6-27](#))**

| NO. |                                  |  | MIN | MAX | UNIT |
|-----|----------------------------------|--|-----|-----|------|
| 1   | $t_c(\text{MDIO\_CLK})$          | Cycle time, MDIO_CLK                                     | 400 |     | ns   |
| 2   | $t_w(\text{MDIO\_CLK})$          | Pulse duration, MDIO_CLK high/low                        | 180 |     | ns   |
| 3   | $t_t(\text{MDIO\_CLK})$          | Transition time, MDIO_CLK                                |     | 5   | ns   |
| 4   | $t_{su}(\text{MDIO-MDIO\_CLKH})$ | Setup time, MDIO_D data input valid before MDIO_CLK high | 10  |     | ns   |
| 5   | $t_h(\text{MDIO\_CLKH-MDIO})$    | Hold time, MDIO_D data input valid after MDIO_CLK high   | 0   |     | ns   |

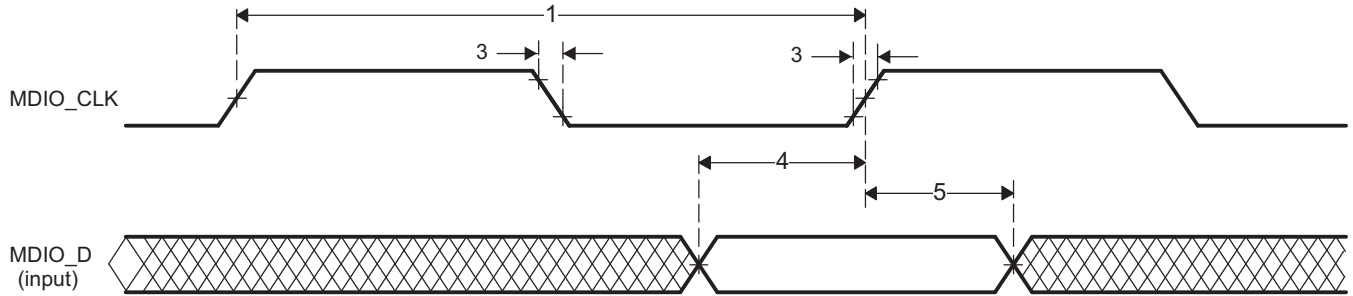


Figure 6-26. MDIO Input Timing

Table 6-39. Switching Characteristics Over Recommended Operating Conditions for MDIO Output (see Figure 6-27)

| NO. | PARAMETER   | MIN | MAX | UNIT |
|-----|---|-----|-----|------|
| 7   | $t_{d(MDIO\_CLKL-MDIO)}$ Delay time, MDIO_CLK low to MDIO_D data output valid | 0   | 100 | ns   |

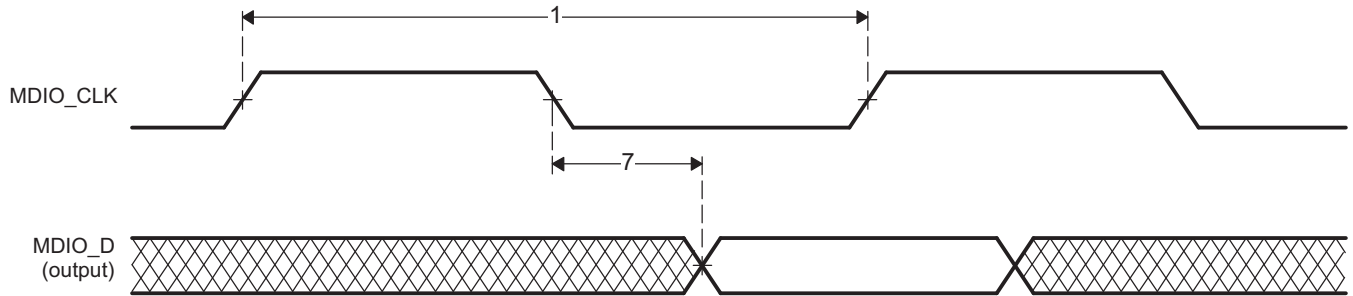


Figure 6-27. MDIO Output Timing

### 6.16 Multichannel Audio Serial Ports (McASP0, McASP1)

The McASP serial port is specifically designed for multichannel audio applications. Its key features are:

- Flexible clock and frame sync generation logic and on-chip dividers
- Up to fourteen transmit or receive data pins and serializers
- Large number of serial data format options, including:
  - TDM Frames with 2 to 32 time slots per frame (periodic) or 1 slot per frame (burst)
  - Time slots of 8, 12, 16, 20, 24, 28, and 32 bits
  - First bit delay 0, 1, or 2 clocks
  - MSB or LSB first bit order
  - Left- or right-aligned data words within time slots
- Extensive error checking and mute generation logic
- All unused pins GPIO-capable

Additionally, while the McASP modules are backward compatible with the McASP on previous devices; the McASP also includes the following new features:

- Transmit & Receive FIFO Buffers for each McASP. Allows the McASP to operate at a higher sample rate by making it more tolerant to DMA latency.
- Dynamic Adjustment of Clock Dividers
  - Clock Divider Value may be changed without resetting the McASP

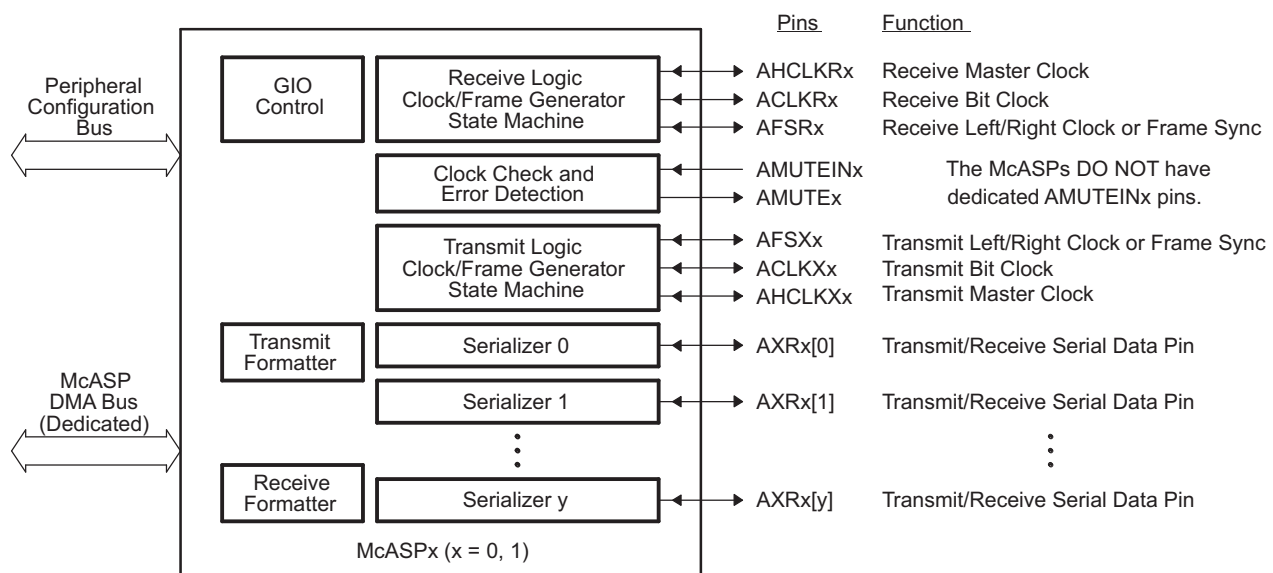
See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

The three McASPs on the C6743 are configured with the following options:

**Table 6-40. C6743 McASP Configurations<sup>(1)</sup>**

| MODULE | SERIALIZERS | AFIFO                    | DIT | C6743 PINS   |
|--------|-------------|--------------------------|-----|--|
| McASP0 | 16          | 64 Word RX<br>64 Word TX | N   | AXR0[15:0], AHCLKR0, ACLKR0, AFSR0, AHCLKX0, ACLKX0, AFSX0                     |
| McASP1 | 12          | 64 Word RX<br>64 Word TX | N   | AXR1[11:10], AXR1[8:0], AHCLKR1, ACLKR1, AFSR1, AHCLKX1, ACLKX1, AFSX1, AMUTE1 |

(1) Pins available are the maximum number of pins that may be configured for a particular McASP; not including pin multiplexing.



**Figure 6-28. McASP Block Diagram**

### 6.16.1 McASP Peripheral Registers Description(s)

Registers for the McASP are summarized in [Table 6-41](#). The registers are accessed through the peripheral configuration port. The receive buffer registers (RBUF) and transmit buffer registers (XBUF) can also be accessed through the DMA port, as listed in [Table 6-42](#)

Registers for the McASP Audio FIFO (AFIFO) are summarized in [Table 6-43](#). Note that the AFIFO Write FIFO (WFIFO) and Read FIFO (RFIFO) have independent control and status registers. The AFIFO control registers are accessed through the peripheral configuration port.

**Table 6-41. McASP Registers Accessed Through Peripheral Configuration Port**

| McASP0<br>BYTE<br>ADDRESS | McASP1<br>BYTE<br>ADDRESS | REGISTER NAME | REGISTER DESCRIPTION   |
|---------------------------|---------------------------|---------------|--|
| 0x01D0 0000               | 0x01D0 4000               | REV           | Revision identification register   |
| 0x01D0 0010               | 0x01D0 4010               | PFUNC         | Pin function register  |
| 0x01D0 0014               | 0x01D0 4014               | PDIR          | Pin direction register   |
| 0x01D0 0018               | 0x01D0 4018               | PDOUT         | Pin data output register   |
| 0x01D0 001C               | 0x01D0 401C               | PDIN          | Read returns: Pin data input register  |
| 0x01D0 001C               | 0x01D0 401C               | PDSET         | Writes affect: Pin data set register (alternate write address: PDOUT)  |
| 0x01D0 0020               | 0x01D0 4020               | PDCLR         | Pin data clear register (alternate write address: PDOUT)   |
| 0x01D0 0044               | 0x01D0 4044               | GBLCTL        | Global control register  |
| 0x01D0 0048               | 0x01D0 4048               | AMUTE         | Audio mute control register  |
| 0x01D0 004C               | 0x01D0 404C               | DLBCTL        | Digital loopback control register  |
| 0x01D0 0050               | 0x01D0 4050               | DITCTL        | DIT mode control register  |
| 0x01D0 0060               | 0x01D0 4060               | RGBLCTL       | Receiver global control register: Alias of GBLCTL, only receive bits are affected - allows receiver to be reset independently from transmitter     |
| 0x01D0 0064               | 0x01D0 4064               | RMASK         | Receive format unit bit mask register  |
| 0x01D0 0068               | 0x01D0 4068               | RFMT          | Receive bit stream format register   |
| 0x01D0 006C               | 0x01D0 406C               | AFSRECTL      | Receive frame sync control register  |
| 0x01D0 0070               | 0x01D0 4070               | ACLKRCTL      | Receive clock control register   |
| 0x01D0 0074               | 0x01D0 4074               | AHCLKRCTL     | Receive high-frequency clock control register  |
| 0x01D0 0078               | 0x01D0 4078               | RTDM          | Receive TDM time slot 0-31 register  |
| 0x01D0 007C               | 0x01D0 407C               | RINTCTL       | Receiver interrupt control register  |
| 0x01D0 0080               | 0x01D0 4080               | RSTAT         | Receiver status register   |
| 0x01D0 0084               | 0x01D0 4084               | RSLOT         | Current receive TDM time slot register   |
| 0x01D0 0088               | 0x01D0 4088               | RCLKCHK       | Receive clock check control register   |
| 0x01D0 008C               | 0x01D0 408C               | REVTCTL       | Receiver DMA event control register  |
| 0x01D0 00A0               | 0x01D0 40A0               | XGBLCTL       | Transmitter global control register. Alias of GBLCTL, only transmit bits are affected - allows transmitter to be reset independently from receiver |
| 0x01D0 00A4               | 0x01D0 40A4               | XMASK         | Transmit format unit bit mask register   |
| 0x01D0 00A8               | 0x01D0 40A8               | XFMT          | Transmit bit stream format register  |
| 0x01D0 00AC               | 0x01D0 40AC               | AFSXCTL       | Transmit frame sync control register   |
| 0x01D0 00B0               | 0x01D0 40B0               | ACLKXCTL      | Transmit clock control register  |
| 0x01D0 00B4               | 0x01D0 40B4               | AHCLKXCTL     | Transmit high-frequency clock control register   |
| 0x01D0 00B8               | 0x01D0 40B8               | XTDM          | Transmit TDM time slot 0-31 register   |
| 0x01D0 00BC               | 0x01D0 40BC               | XINTCTL       | Transmitter interrupt control register   |
| 0x01D0 00C0               | 0x01D0 40C0               | XSTAT         | Transmitter status register  |
| 0x01D0 00C4               | 0x01D0 40C4               | XSLOT         | Current transmit TDM time slot register  |
| 0x01D0 00C8               | 0x01D0 40C8               | XCLKCHK       | Transmit clock check control register  |
| 0x01D0 00CC               | 0x01D0 40CC               | XEVTCTL       | Transmitter DMA event control register   |

**Table 6-41. McASP Registers Accessed Through Peripheral Configuration Port (continued)**

| McASP0<br>BYTE<br>ADDRESS | McASP1<br>BYTE<br>ADDRESS | REGISTER NAME | REGISTER DESCRIPTION  |
|---------------------------|---------------------------|---------------|---|
| 0x01D0 0100               | 0x01D0 4100               | DITCSRA0      | Left (even TDM time slot) channel status register (DIT mode) 0    |
| 0x01D0 0104               | 0x01D0 4104               | DITCSRA1      | Left (even TDM time slot) channel status register (DIT mode) 1    |
| 0x01D0 0108               | 0x01D0 4108               | DITCSRA2      | Left (even TDM time slot) channel status register (DIT mode) 2    |
| 0x01D0 010C               | 0x01D0 410C               | DITCSRA3      | Left (even TDM time slot) channel status register (DIT mode) 3    |
| 0x01D0 0110               | 0x01D0 4110               | DITCSRA4      | Left (even TDM time slot) channel status register (DIT mode) 4    |
| 0x01D0 0114               | 0x01D0 4114               | DITCSRA5      | Left (even TDM time slot) channel status register (DIT mode) 5    |
| 0x01D0 0118               | 0x01D0 4118               | DITCSRB0      | Right (odd TDM time slot) channel status register (DIT mode) 0    |
| 0x01D0 011C               | 0x01D0 411C               | DITCSRB1      | Right (odd TDM time slot) channel status register (DIT mode) 1    |
| 0x01D0 0120               | 0x01D0 4120               | DITCSRB2      | Right (odd TDM time slot) channel status register (DIT mode) 2    |
| 0x01D0 0124               | 0x01D0 4124               | DITCSRB3      | Right (odd TDM time slot) channel status register (DIT mode) 3    |
| 0x01D0 0128               | 0x01D0 4128               | DITCSRB4      | Right (odd TDM time slot) channel status register (DIT mode) 4    |
| 0x01D0 012C               | 0x01D0 412C               | DITCSRB5      | Right (odd TDM time slot) channel status register (DIT mode) 5    |
| 0x01D0 0130               | 0x01D0 4130               | DITUDRA0      | Left (even TDM time slot) channel user data register (DIT mode) 0 |
| 0x01D0 0134               | 0x01D0 4134               | DITUDRA1      | Left (even TDM time slot) channel user data register (DIT mode) 1 |
| 0x01D0 0138               | 0x01D0 4138               | DITUDRA2      | Left (even TDM time slot) channel user data register (DIT mode) 2 |
| 0x01D0 013C               | 0x01D0 413C               | DITUDRA3      | Left (even TDM time slot) channel user data register (DIT mode) 3 |
| 0x01D0 0140               | 0x01D0 4140               | DITUDRA4      | Left (even TDM time slot) channel user data register (DIT mode) 4 |
| 0x01D0 0144               | 0x01D0 4144               | DITUDRA5      | Left (even TDM time slot) channel user data register (DIT mode) 5 |
| 0x01D0 0148               | 0x01D0 4148               | DITUDRB0      | Right (odd TDM time slot) channel user data register (DIT mode) 0 |
| 0x01D0 014C               | 0x01D0 414C               | DITUDRB1      | Right (odd TDM time slot) channel user data register (DIT mode) 1 |
| 0x01D0 0150               | 0x01D0 4150               | DITUDRB2      | Right (odd TDM time slot) channel user data register (DIT mode) 2 |
| 0x01D0 0154               | 0x01D0 4154               | DITUDRB3      | Right (odd TDM time slot) channel user data register (DIT mode) 3 |
| 0x01D0 0158               | 0x01D0 4158               | DITUDRB4      | Right (odd TDM time slot) channel user data register (DIT mode) 4 |
| 0x01D0 015C               | 0x01D0 415C               | DITUDRB5      | Right (odd TDM time slot) channel user data register (DIT mode) 5 |
| 0x01D0 0180               | 0x01D0 4180               | SRCTL0        | Serializer control register 0                                     |
| 0x01D0 0184               | 0x01D0 4184               | SRCTL1        | Serializer control register 1                                     |
| 0x01D0 0188               | 0x01D0 4188               | SRCTL2        | Serializer control register 2                                     |
| 0x01D0 018C               | 0x01D0 418C               | SRCTL3        | Serializer control register 3                                     |
| 0x01D0 0190               | 0x01D0 4190               | SRCTL4        | Serializer control register 4                                     |
| 0x01D0 0194               | 0x01D0 4194               | SRCTL5        | Serializer control register 5                                     |
| 0x01D0 0198               | 0x01D0 4198               | SRCTL6        | Serializer control register 6                                     |
| 0x01D0 019C               | 0x01D0 419C               | SRCTL7        | Serializer control register 7                                     |
| 0x01D0 01A0               | 0x01D0 41A0               | SRCTL8        | Serializer control register 8                                     |
| 0x01D0 01A4               | 0x01D0 41A4               | SRCTL9        | Serializer control register 9                                     |
| 0x01D0 01A8               | 0x01D0 41A8               | SRCTL10       | Serializer control register 10                                    |
| 0x01D0 01AC               | 0x01D0 41AC               | SRCTL11       | Serializer control register 11                                    |
| 0x01D0 01B0               | 0x01D0 41B0               | SRCTL12       | Serializer control register 12                                    |
| 0x01D0 01B4               | 0x01D0 41B4               | SRCTL13       | Serializer control register 13                                    |
| 0x01D0 01B8               | 0x01D0 41B8               | SRCTL14       | Serializer control register 14                                    |
| 0x01D0 01BC               | 0x01D0 41BC               | SRCTL15       | Serializer control register 15                                    |

**Table 6-41. McASP Registers Accessed Through Peripheral Configuration Port (continued)**

| McASP0<br>BYTE<br>ADDRESS | McASP1<br>BYTE<br>ADDRESS | REGISTER NAME         | REGISTER DESCRIPTION                       |
|---------------------------|---------------------------|-----------------------|--|
| 0x01D0 0200               | 0x01D0 4200               | XBUF0 <sup>(1)</sup>  | Transmit buffer register for serializer 0  |
| 0x01D0 0204               | 0x01D0 4204               | XBUF1 <sup>(1)</sup>  | Transmit buffer register for serializer 1  |
| 0x01D0 0208               | 0x01D0 4208               | XBUF2 <sup>(1)</sup>  | Transmit buffer register for serializer 2  |
| 0x01D0 020C               | 0x01D0 420C               | XBUF3 <sup>(1)</sup>  | Transmit buffer register for serializer 3  |
| 0x01D0 0210               | 0x01D0 4210               | XBUF4 <sup>(1)</sup>  | Transmit buffer register for serializer 4  |
| 0x01D0 0214               | 0x01D0 4214               | XBUF5 <sup>(1)</sup>  | Transmit buffer register for serializer 5  |
| 0x01D0 0218               | 0x01D0 4218               | XBUF6 <sup>(1)</sup>  | Transmit buffer register for serializer 6  |
| 0x01D0 021C               | 0x01D0 421C               | XBUF7 <sup>(1)</sup>  | Transmit buffer register for serializer 7  |
| 0x01D0 0220               | 0x01D0 4220               | XBUF8 <sup>(1)</sup>  | Transmit buffer register for serializer 8  |
| 0x01D0 0224               | 0x01D0 4224               | XBUF9 <sup>(1)</sup>  | Transmit buffer register for serializer 9  |
| 0x01D0 0228               | 0x01D0 4228               | XBUF10 <sup>(1)</sup> | Transmit buffer register for serializer 10 |
| 0x01D0 022C               | 0x01D0 422C               | XBUF11 <sup>(1)</sup> | Transmit buffer register for serializer 11 |
| 0x01D0 0230               | 0x01D0 4230               | XBUF12 <sup>(1)</sup> | Transmit buffer register for serializer 12 |
| 0x01D0 0234               | 0x01D0 4234               | XBUF13 <sup>(1)</sup> | Transmit buffer register for serializer 13 |
| 0x01D0 0238               | 0x01D0 4238               | XBUF14 <sup>(1)</sup> | Transmit buffer register for serializer 14 |
| 0x01D0 023C               | 0x01D0 423C               | XBUF15 <sup>(1)</sup> | Transmit buffer register for serializer 15 |
| 0x01D0 0280               | 0x01D0 4280               | RBUF0 <sup>(2)</sup>  | Receive buffer register for serializer 0   |
| 0x01D0 0284               | 0x01D0 4284               | RBUF1 <sup>(2)</sup>  | Receive buffer register for serializer 1   |
| 0x01D0 0288               | 0x01D0 4288               | RBUF2 <sup>(2)</sup>  | Receive buffer register for serializer 2   |
| 0x01D0 028C               | 0x01D0 428C               | RBUF3 <sup>(2)</sup>  | Receive buffer register for serializer 3   |
| 0x01D0 0290               | 0x01D0 4290               | RBUF4 <sup>(2)</sup>  | Receive buffer register for serializer 4   |
| 0x01D0 0294               | 0x01D0 4294               | RBUF5 <sup>(2)</sup>  | Receive buffer register for serializer 5   |
| 0x01D0 0298               | 0x01D0 4298               | RBUF6 <sup>(2)</sup>  | Receive buffer register for serializer 6   |
| 0x01D0 029C               | 0x01D0 429C               | RBUF7 <sup>(2)</sup>  | Receive buffer register for serializer 7   |
| 0x01D0 02A0               | 0x01D0 42A0               | RBUF8 <sup>(2)</sup>  | Receive buffer register for serializer 8   |
| 0x01D0 02A4               | 0x01D0 42A4               | RBUF9 <sup>(2)</sup>  | Receive buffer register for serializer 9   |
| 0x01D0 02A8               | 0x01D0 42A8               | RBUF10 <sup>(2)</sup> | Receive buffer register for serializer 10  |
| 0x01D0 02AC               | 0x01D0 42AC               | RBUF11 <sup>(2)</sup> | Receive buffer register for serializer 11  |
| 0x01D0 02B0               | 0x01D0 42B0               | RBUF12 <sup>(2)</sup> | Receive buffer register for serializer 12  |
| 0x01D0 02B4               | 0x01D0 42B4               | RBUF13 <sup>(2)</sup> | Receive buffer register for serializer 13  |
| 0x01D0 02B8               | 0x01D0 42B8               | RBUF14 <sup>(2)</sup> | Receive buffer register for serializer 14  |
| 0x01D0 02BC               | 0x01D0 42BC               | RBUF15 <sup>(2)</sup> | Receive buffer register for serializer 15  |

(1) Writes to XBUF originate from peripheral configuration port only when XBUSEL = 1 in XFMT.

(2) Reads from RBUF originate on peripheral configuration port only when RBUSEL = 1 in RFMT.

**Table 6-42. McASP Registers Accessed Through DMA Port**

| HEX<br>ADDRESS    | McASP0<br>BYTE ADDRESS | McASP1<br>BYTE ADDRESS | REGISTER<br>NAME | REGISTER DESCRIPTION  |
|-------------------|------------------------|------------------------|------------------|---|
| Read<br>Accesses  | 01D0 2000              | 01D0 6000              | RBUF             | Receive buffer DMA port address. Cycles through receive serializers, skipping over transmit serializers and inactive serializers. Starts at the lowest serializer at the beginning of each time slot. Reads from DMA port only if RBUSEL = 0 in RFMT. |
| Write<br>Accesses | 01D0 2000              | 01D0 6000              | XBUF             | Transmit buffer DMA port address. Cycles through transmit serializers, skipping over receive and inactive serializers. Starts at the lowest serializer at the beginning of each time slot. Writes to DMA port only if XBUSEL = 0 in XFMT.             |

**Table 6-43. McASP AFIFO Registers Accessed Through Peripheral Configuration Port**

| McASP0<br>BYTE ADDRESS | McASP1<br>BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                   |
|------------------------|------------------------|---------------|--|
| 0x01D0 1000            | 0x01D0 5000            | AFIFOREV      | AFIFO revision identification register |
| 0x01D0 1010            | 0x01D0 5010            | WFIFOCTL      | Write FIFO control register            |
| 0x01D0 1014            | 0x01D0 5014            | WFIFOSTS      | Write FIFO status register             |
| 0x01D0 1018            | 0x01D0 5018            | RFIFOCTL      | Read FIFO control register             |
| 0x01D0 101C            | 0x01D0 501C            | RFIFOSTS      | Read FIFO status register              |

## 6.16.2 McASP Electrical Data/Timing

### 6.16.2.1 Multichannel Audio Serial Port 0 (McASP0) Timing

Table 6-44 and Table 6-45 assume testing over recommended operating conditions (see Figure 6-29 and Figure 6-30).

**Table 6-44. McASP0 Timing Requirements<sup>(1)(2)</sup>**

| NO. |                        |  | MIN                 | MAX | UNIT |
|-----|------------------------|--|---------------------|-----|------|
| 1   | $t_{c(AHCLKRX)}$       | Cycle time, AHCLKR0 external, AHCLKR0 input                        | 25                  |     | ns   |
|     |                        | Cycle time, AHCLKX0 external, AHCLKX0 input                        | 25                  |     |      |
| 2   | $t_{w(AHCLKRX)}$       | Pulse duration, AHCLKR0 external, AHCLKR0 input                    | 12.5                |     | ns   |
|     |                        | Pulse duration, AHCLKX0 external, AHCLKX0 input                    | 12.5                |     |      |
| 3   | $t_{c(ACLKRX)}$        | Cycle time, ACLKR0 external, ACLKR0 input                          | greater of 2P or 25 |     | ns   |
|     |                        | Cycle time, ACLKX0 external, ACLKX0 input                          | greater of 2P or 25 |     |      |
| 4   | $t_{w(ACLKRX)}$        | Pulse duration, ACLKR0 external, ACLKR0 input                      | 12.5                |     | ns   |
|     |                        | Pulse duration, ACLKX0 external, ACLKX0 input                      | 12.5                |     |      |
| 5   | $t_{su(AFSRX-ACLKRX)}$ | Setup time, AFSR0 input to ACLKR0 internal <sup>(3)</sup>          | 9.4                 |     | ns   |
|     |                        | Setup time, AFSX0 input to ACLKX0 internal                         | 9.4                 |     |      |
|     |                        | Setup time, AFSR0 input to ACLKR0 external input <sup>(3)</sup>    | 2.9                 |     |      |
|     |                        | Setup time, AFSX0 input to ACLKX0 external input                   | 2.9                 |     |      |
|     |                        | Setup time, AFSR0 input to ACLKR0 external output <sup>(3)</sup>   | 2.9                 |     |      |
|     |                        | Setup time, AFSX0 input to ACLKX0 external output                  | 2.9                 |     |      |
| 6   | $t_{h(ACLKRX-AFSRX)}$  | Hold time, AFSR0 input after ACLKR0 internal <sup>(3)</sup>        | -1.2                |     | ns   |
|     |                        | Hold time, AFSX0 input after ACLKX0 internal                       | -1.2                |     |      |
|     |                        | Hold time, AFSR0 input after ACLKR0 external input <sup>(3)</sup>  | 0.9                 |     |      |
|     |                        | Hold time, AFSX0 input after ACLKX0 external input                 | 0.9                 |     |      |
|     |                        | Hold time, AFSR0 input after ACLKR0 external output <sup>(3)</sup> | 0.9                 |     |      |
|     |                        | Hold time, AFSX0 input after ACLKX0 external output                | 0.9                 |     |      |
| 7   | $t_{su(AXR-ACLKRX)}$   | Setup time, AXR0[n] input to ACLKR0 internal <sup>(3)</sup>        | 9.4                 |     | ns   |
|     |                        | Setup time, AXR0[n] input to ACLKX0 internal <sup>(4)</sup>        | 9.4                 |     |      |
|     |                        | Setup time, AXR0[n] input to ACLKR0 external input <sup>(3)</sup>  | 2.9                 |     |      |
|     |                        | Setup time, AXR0[n] input to ACLKX0 external input <sup>(4)</sup>  | 2.9                 |     |      |
|     |                        | Setup time, AXR0[n] input to ACLKR0 external output <sup>(3)</sup> | 2.9                 |     |      |
|     |                        | Setup time, AXR0[n] input to ACLKX0 external output <sup>(4)</sup> | 2.9                 |     |      |

- (1) ACLKX0 internal – McASP0 ACLKXCTL.CLKXM = 1, PDIR.ACLKX = 1  
ACLKX0 external input – McASP0 ACLKXCTL.CLKXM = 0, PDIR.ACLKX = 0  
ACLKX0 external output – McASP0 ACLKXCTL.CLKXM = 0, PDIR.ACLKX = 1  
ACLKR0 internal – McASP0 ACLKRCTL.CLKRM = 1, PDIR.ACLKR = 1  
ACLKR0 external input – McASP0 ACLKRCTL.CLKRM = 0, PDIR.ACLKR = 0  
ACLKR0 external output – McASP0 ACLKRCTL.CLKRM = 0, PDIR.ACLKR = 1

(2) P = SYSCLK2 period

(3) McASP0 ACLKXCTL.ASYNC=1: Receiver is clocked by its own ACLKR0

(4) McASP0 ACLKXCTL.ASYNC=0: Receiver is clocked by transmitter's ACLKX0

**Table 6-44. McASP0 Timing Requirements<sup>(1)(2)</sup> (continued)**

| NO. |                     |  | MIN  | MAX | UNIT |
|-----|---------------------|--|------|-----|------|
| 8   | $t_{h(ACLKRX-AXR)}$ | Hold time, AXR0[n] input after ACLKR0 internal <sup>(3)</sup>        | -1.3 |     | ns   |
|     |                     | Hold time, AXR0[n] input after ACLKX0 internal <sup>(4)</sup>        | -1.3 |     |      |
|     |                     | Hold time, AXR0[n] input after ACLKR0 external input <sup>(3)</sup>  | 0.5  |     |      |
|     |                     | Hold time, AXR0[n] input after ACLKX0 external input <sup>(4)</sup>  | 0.5  |     |      |
|     |                     | Hold time, AXR0[n] input after ACLKR0 external output <sup>(3)</sup> | 0.5  |     |      |
|     |                     | Hold time, AXR0[n] input after ACLKX0 external output <sup>(4)</sup> | 0.5  |     |      |

**Table 6-45. McASP0 Switching Characteristics<sup>(1)</sup>**

| NO. | PARAMETER              | MIN  | MAX                                | UNIT |    |
|-----|------------------------|--|------------------------------------|------|----|
| 9   | $t_{c(AHCLKRX)}$       | Cycle time, AHCLKR0 internal, AHCLKR0 output                   | 25                                 |      | ns |
|     |                        | Cycle time, AHCLKR0 external, AHCLKR0 output                   | 25                                 |      |    |
|     |                        | Cycle time, AHCLKX0 internal, AHCLKX0 output                   | 25                                 |      |    |
|     |                        | Cycle time, AHCLKX0 external, AHCLKX0 output                   | 25                                 |      |    |
| 10  | $t_{w(AHCLKRX)}$       | Pulse duration, AHCLKR0 internal, AHCLKR0 output               | $(AHR/2) - 2.5^{(2)}$              |      | ns |
|     |                        | Pulse duration, AHCLKR0 external, AHCLKR0 output               | $(AHR/2) - 2.5^{(2)}$              |      |    |
|     |                        | Pulse duration, AHCLKX0 internal, AHCLKX0 output               | $(AHX/2) - 2.5^{(3)}$              |      |    |
|     |                        | Pulse duration, AHCLKX0 external, AHCLKX0 output               | $(AHX/2) - 2.5^{(3)}$              |      |    |
| 11  | $t_{c(ACLKRX)}$        | Cycle time, ACLKR0 internal, ACLKR0 output                     | greater of 2P or 25 <sup>(4)</sup> |      | ns |
|     |                        | Cycle time, ACLKR0 external, ACLKR0 output                     | greater of 2P or 25 <sup>(4)</sup> |      |    |
|     |                        | Cycle time, ACLKX0 internal, ACLKX0 output                     | greater of 2P or 25 <sup>(4)</sup> |      |    |
|     |                        | Cycle time, ACLKX0 external, ACLKX0 output                     | greater of 2P or 25 <sup>(4)</sup> |      |    |
| 12  | $t_{w(ACLKRX)}$        | Pulse duration, ACLKR0 internal, ACLKR0 output                 | $(AR/2) - 2.5^{(5)}$               |      | ns |
|     |                        | Pulse duration, ACLKR0 external, ACLKR0 output                 | $(AR/2) - 2.5^{(5)}$               |      |    |
|     |                        | Pulse duration, ACLKX0 internal, ACLKX0 output                 | $(AX/2) - 2.5^{(6)}$               |      |    |
|     |                        | Pulse duration, ACLKX0 external, ACLKX0 output                 | $(AX/2) - 2.5^{(6)}$               |      |    |
| 13  | $t_{d(ACLKRX-AFSRX)}$  | Delay time, ACLKR0 internal, AFSR output <sup>(7)</sup>        | 0                                  | 5.8  | ns |
|     |                        | Delay time, ACLKX0 internal, AFSX output                       | 0                                  | 5.8  |    |
|     |                        | Delay time, ACLKR0 external input, AFSR output <sup>(7)</sup>  | 2.5                                | 11.6 |    |
|     |                        | Delay time, ACLKX0 external input, AFSX output                 | 2.5                                | 11.6 |    |
|     |                        | Delay time, ACLKR0 external output, AFSR output <sup>(7)</sup> | 2.5                                | 11.6 |    |
|     |                        | Delay time, ACLKX0 external output, AFSX output                | 2.5                                | 11.6 |    |
| 14  | $t_{d(ACLKX-AXRV)}$    | Delay time, ACLKX0 internal, AXR0[n] output                    | 0                                  | 5.8  | ns |
|     |                        | Delay time, ACLKX0 external input, AXR0[n] output              | 2.5                                | 11.6 |    |
|     |                        | Delay time, ACLKX0 external output, AXR0[n] output             | 2.5                                | 11.6 |    |
| 15  | $t_{dis(ACLKX-AXRHZ)}$ | Disable time, ACLKX0 internal, AXR0[n] output                  | 0                                  | 5.8  | ns |
|     |                        | Disable time, ACLKX0 external input, AXR0[n] output            | 3                                  | 11.6 |    |
|     |                        | Disable time, ACLKX0 external output, AXR0[n] output           | 3                                  | 11.6 |    |

- (1) McASP0 ACLKX0 internal – ACLKXCTL.CLKXM = 1, PDIR.ACLKX = 1  
 ACLKX0 external input – McASP0 ACLKXCTL.CLKXM = 0, PDIR.ACLKX = 0  
 ACLKX0 external output – McASP0ACLKXCTL.CLKXM = 0, PDIR.ACLKX = 1  
 ACLKR0 internal – McASP0 ACLKR0CTL.CLKRM = 1, PDIR.ACLKR = 1  
 ACLKR0 external input – McASP0 ACLKR0CTL.CLKRM = 0, PDIR.ACLKR = 0  
 ACLKR0 external output – McASP0 ACLKR0CTL.CLKRM = 0, PDIR.ACLKR = 1
- (2) AHR - Cycle time, AHCLKR0.  
 (3) AHX - Cycle time, AHCLKX0.  
 (4) P = SYSCLK2 period  
 (5) AR - ACLKR0 period.  
 (6) AX - ACLKX0 period.  
 (7) McASP0 ACLKXCTL.ASYNC=1: Receiver is clocked by its own ACLKR0

### 6.16.2.2 Multichannel Audio Serial Port 1 (McASP1) Timing

Table 6-46 and Table 6-47 assume testing over recommended operating conditions (see Figure 6-29 and Figure 6-30).

**Table 6-46. McASP1 Timing Requirements<sup>(1)(2)</sup>**

| NO. |                        |  | MIN                 | MAX | UNIT |
|-----|------------------------|--|---------------------|-----|------|
| 1   | $t_{c(AHCLKRX)}$       | Cycle time, AHCLKR1 external, AHCLKR1 input                          | 25                  |     | ns   |
|     |                        | Cycle time, AHCLKX1 external, AHCLKX1 input                          | 25                  |     |      |
| 2   | $t_{w(AHCLKRX)}$       | Pulse duration, AHCLKR1 external, AHCLKR1 input                      | 12.5                |     | ns   |
|     |                        | Pulse duration, AHCLKX1 external, AHCLKX1 input                      | 12.5                |     |      |
| 3   | $t_{c(ACLKRX)}$        | Cycle time, ACLKR1 external, ACLKR1 input                            | greater of 2P or 25 |     | ns   |
|     |                        | Cycle time, ACLKX1 external, ACLKX1 input                            | greater of 2P or 25 |     |      |
| 4   | $t_{w(ACLKRX)}$        | Pulse duration, ACLKR1 external, ACLKR1 input                        | 12.5                |     | ns   |
|     |                        | Pulse duration, ACLKX1 external, ACLKX1 input                        | 12.5                |     |      |
| 5   | $t_{su(AFSRX-ACLKRX)}$ | Setup time, AFSR1 input to ACLKR1 internal <sup>(3)</sup>            | 10.4                |     | ns   |
|     |                        | Setup time, AFSX1 input to ACLKX1 internal                           | 10.4                |     |      |
|     |                        | Setup time, AFSR1 input to ACLKR1 external input <sup>(3)</sup>      | 2.6                 |     |      |
|     |                        | Setup time, AFSX1 input to ACLKX1 external input                     | 2.6                 |     |      |
|     |                        | Setup time, AFSR1 input to ACLKR1 external output <sup>(3)</sup>     | 2.6                 |     |      |
|     |                        | Setup time, AFSX1 input to ACLKX1 external output                    | 2.6                 |     |      |
| 6   | $t_{h(ACLKRX-AFSRX)}$  | Hold time, AFSR1 input after ACLKR1 internal <sup>(3)</sup>          | -1.9                |     | ns   |
|     |                        | Hold time, AFSX1 input after ACLKX1 internal                         | -1.9                |     |      |
|     |                        | Hold time, AFSR1 input after ACLKR1 external input <sup>(3)</sup>    | 0.7                 |     |      |
|     |                        | Hold time, AFSX1 input after ACLKX1 external input                   | 0.7                 |     |      |
|     |                        | Hold time, AFSR1 input after ACLKR1 external output <sup>(3)</sup>   | 0.7                 |     |      |
|     |                        | Hold time, AFSX1 input after ACLKX1 external output                  | 0.7                 |     |      |
| 7   | $t_{su(AXR-ACLKRX)}$   | Setup time, AXR1[n] input to ACLKR1 internal <sup>(3)</sup>          | 10.4                |     | ns   |
|     |                        | Setup time, AXR1[n] input to ACLKX1 internal <sup>(4)</sup>          | 10.4                |     |      |
|     |                        | Setup time, AXR1[n] input to ACLKR1 external input <sup>(3)</sup>    | 2.6                 |     |      |
|     |                        | Setup time, AXR1[n] input to ACLKX1 external input <sup>(4)</sup>    | 2.6                 |     |      |
|     |                        | Setup time, AXR1[n] input to ACLKR1 external output <sup>(3)</sup>   | 2.6                 |     |      |
|     |                        | Setup time, AXR1[n] input to ACLKX1 external output <sup>(4)</sup>   | 2.6                 |     |      |
| 8   | $t_{h(ACLKRX-AXR)}$    | Hold time, AXR1[n] input after ACLKR1 internal <sup>(3)</sup>        | -1.8                |     | ns   |
|     |                        | Hold time, AXR1[n] input after ACLKX1 internal <sup>(4)</sup>        | -1.8                |     |      |
|     |                        | Hold time, AXR1[n] input after ACLKR1 external input <sup>(3)</sup>  | 0.5                 |     |      |
|     |                        | Hold time, AXR1[n] input after ACLKX1 external input <sup>(4)</sup>  | 0.5                 |     |      |
|     |                        | Hold time, AXR1[n] input after ACLKR1 external output <sup>(3)</sup> | 0.5                 |     |      |
|     |                        | Hold time, AXR1[n] input after ACLKX1 external output <sup>(4)</sup> | 0.5                 |     |      |

- (1) ACLKX1 internal – McASP1 ACLKXCTL.CLKXM = 1, PDIR.ACLKX = 1  
 ACLKX1 external input – McASP1 ACLKXCTL.CLKXM = 0, PDIR.ACLKX = 0  
 ACLKX1 external output – McASP1 ACLKXCTL.CLKXM = 0, PDIR.ACLKX = 1  
 ACLKR1 internal – McASP1 ACLKRCTL.CLKRM = 1, PDIR.ACLKR = 1  
 ACLKR1 external input – McASP1 ACLKRCTL.CLKRM = 0, PDIR.ACLKR = 0  
 ACLKR1 external output – McASP1 ACLKRCTL.CLKRM = 0, PDIR.ACLKR = 1
- (2) P = SYSCLK2 period
- (3) McASP1 ACLKXCTL.ASYNC=1: Receiver is clocked by its own ACLKR1
- (4) McASP1 ACLKXCTL.ASYNC=0: Receiver is clocked by transmitter's ACLKX1

**Table 6-47. McASP1 Switching Characteristics<sup>(1)</sup>**

| NO. | PARAMETER              |  | MIN                                | MAX  | UNIT |
|-----|------------------------|--|------------------------------------|------|------|
| 9   | $t_{c(AHCLKRX)}$       | Cycle time, AHCLKR1 internal, AHCLKR1 output                   | 25                                 |      | ns   |
|     |                        | Cycle time, AHCLKR1 external, AHCLKR1 output                   | 25                                 |      |      |
|     |                        | Cycle time, AHCLKX1 internal, AHCLKX1 output                   | 25                                 |      |      |
|     |                        | Cycle time, AHCLKX1 external, AHCLKX1 output                   | 25                                 |      |      |
| 10  | $t_{w(AHCLKRX)}$       | Pulse duration, AHCLKR1 internal, AHCLKR1 output               | $(AHR/2) - 2.5^{(2)}$              |      | ns   |
|     |                        | Pulse duration, AHCLKR1 external, AHCLKR1 output               | $(AHR/2) - 2.5^{(2)}$              |      |      |
|     |                        | Pulse duration, AHCLKX1 internal, AHCLKX1 output               | $(AHX/2) - 2.5^{(3)}$              |      |      |
|     |                        | Pulse duration, AHCLKX1 external, AHCLKX1 output               | $(AHX/2) - 2.5^{(3)}$              |      |      |
| 11  | $t_{c(ACLKRX)}$        | Cycle time, ACLKR1 internal, ACLKR1 output                     | greater of 2P or 25 <sup>(4)</sup> |      | ns   |
|     |                        | Cycle time, ACLKR1 external, ACLKR1 output                     | greater of 2P or 25 <sup>(4)</sup> |      |      |
|     |                        | Cycle time, ACLKX1 internal, ACLKX1 output                     | greater of 2P or 25 <sup>(4)</sup> |      |      |
|     |                        | Cycle time, ACLKX1 external, ACLKX1 output                     | greater of 2P or 25 <sup>(4)</sup> |      |      |
| 12  | $t_{w(ACLKRX)}$        | Pulse duration, ACLKR1 internal, ACLKR1 output                 | $(AR/2) - 2.5^{(5)}$               |      | ns   |
|     |                        | Pulse duration, ACLKR1 external, ACLKR1 output                 | $(AR/2) - 2.5^{(5)}$               |      |      |
|     |                        | Pulse duration, ACLKX1 internal, ACLKX1 output                 | $(AX/2) - 2.5^{(6)}$               |      |      |
|     |                        | Pulse duration, ACLKX1 external, ACLKX1 output                 | $(AX/2) - 2.5^{(6)}$               |      |      |
| 13  | $t_{d(ACLKRX-AFSRX)}$  | Delay time, ACLKR1 internal, AFSR output <sup>(7)</sup>        | 0.5                                | 6.7  | ns   |
|     |                        | Delay time, ACLKX1 internal, AFSX output                       | 0.5                                | 6.7  |      |
|     |                        | Delay time, ACLKR1 external input, AFSR output <sup>(7)</sup>  | 3.4                                | 13.8 |      |
|     |                        | Delay time, ACLKX1 external input, AFSX output                 | 3.4                                | 13.8 |      |
|     |                        | Delay time, ACLKR1 external output, AFSR output <sup>(7)</sup> | 3.4                                | 13.8 |      |
|     |                        | Delay time, ACLKX1 external output, AFSX output                | 3.4                                | 13.8 |      |
| 14  | $t_{d(ACLKX-AXRV)}$    | Delay time, ACLKX1 internal, AXR1[n] output                    | 0.5                                | 6.7  | ns   |
|     |                        | Delay time, ACLKX1 external input, AXR1[n] output              | 3.4                                | 13.8 |      |
|     |                        | Delay time, ACLKX1 external output, AXR1[n] output             | 3.4                                | 13.8 |      |
| 15  | $t_{dis(ACLKX-AXRHZ)}$ | Disable time, ACLKX1 internal, AXR1[n] output                  | 0.5                                | 6.7  | ns   |
|     |                        | Disable time, ACLKX1 external input, AXR1[n] output            | 3.9                                | 13.8 |      |
|     |                        | Disable time, ACLKX1 external output, AXR1[n] output           | 3.9                                | 13.8 |      |

- (1) McASP1 ACLKX1 internal – ACLKXCTL.CLKXM = 1, PDIR.ACLKX = 1  
 McASP1 ACLKX1 external input – ACLKXCTL.CLKXM = 0, PDIR.ACLKX = 0  
 McASP1 ACLKX1 external output – ACLKXCTL.CLKXM = 0, PDIR.ACLKX = 1  
 McASP1 ACLKR1 internal – ACLKR1CTL.CLKRM = 1, PDIR.ACLKR = 1  
 McASP1 ACLKR1 external input – ACLKRCTL.CLKRM = 0, PDIR.ACLKR = 0  
 McASP1 ACLKR1 external output – ACLKRCTL.CLKRM = 0, PDIR.ACLKR = 1

(2) AHR - Cycle time, AHCLKR1.

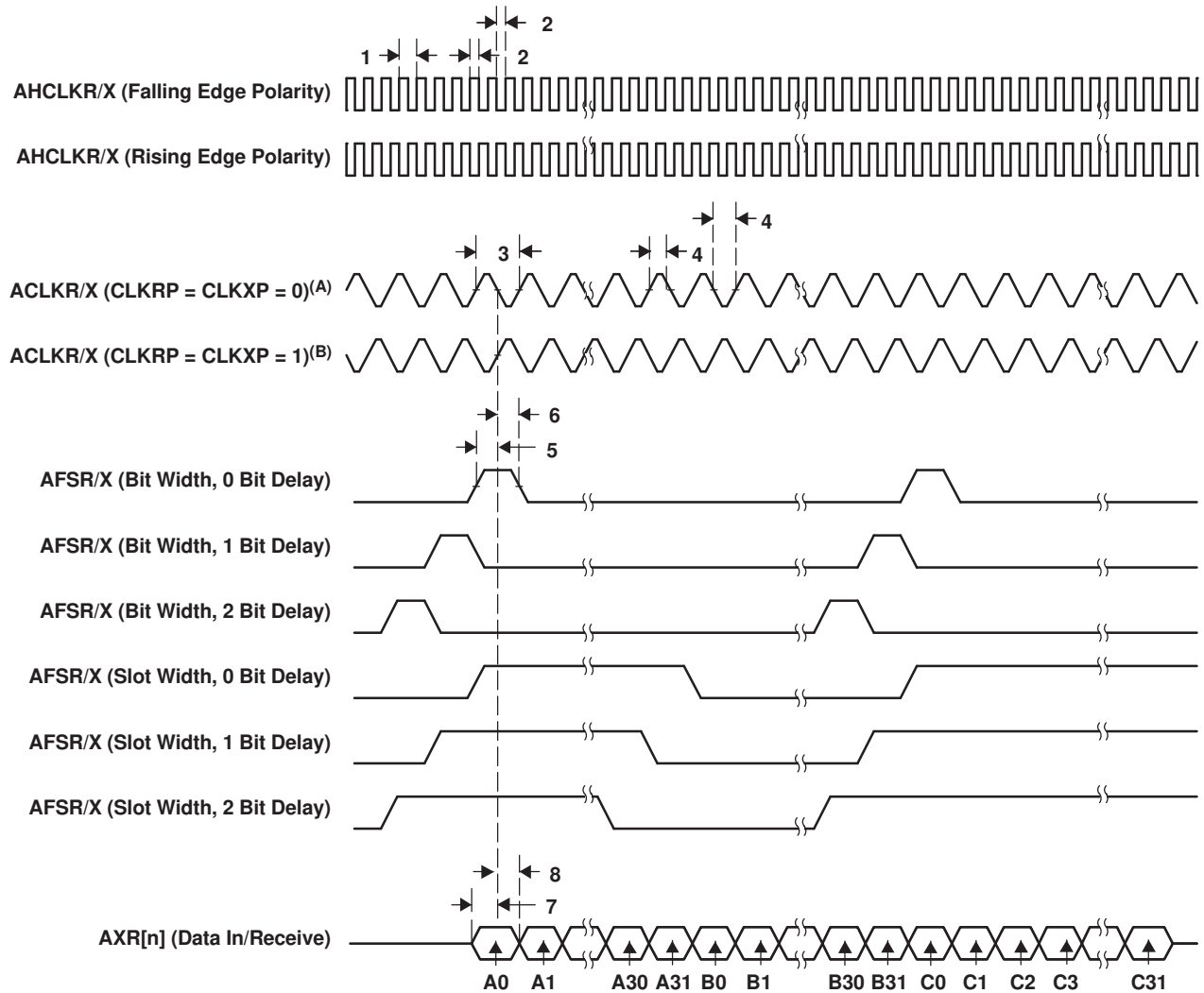
(3) AHX - Cycle time, AHCLKX1.

(4) P = SYSCLK2 period

(5) AR - ACLKR1 period.

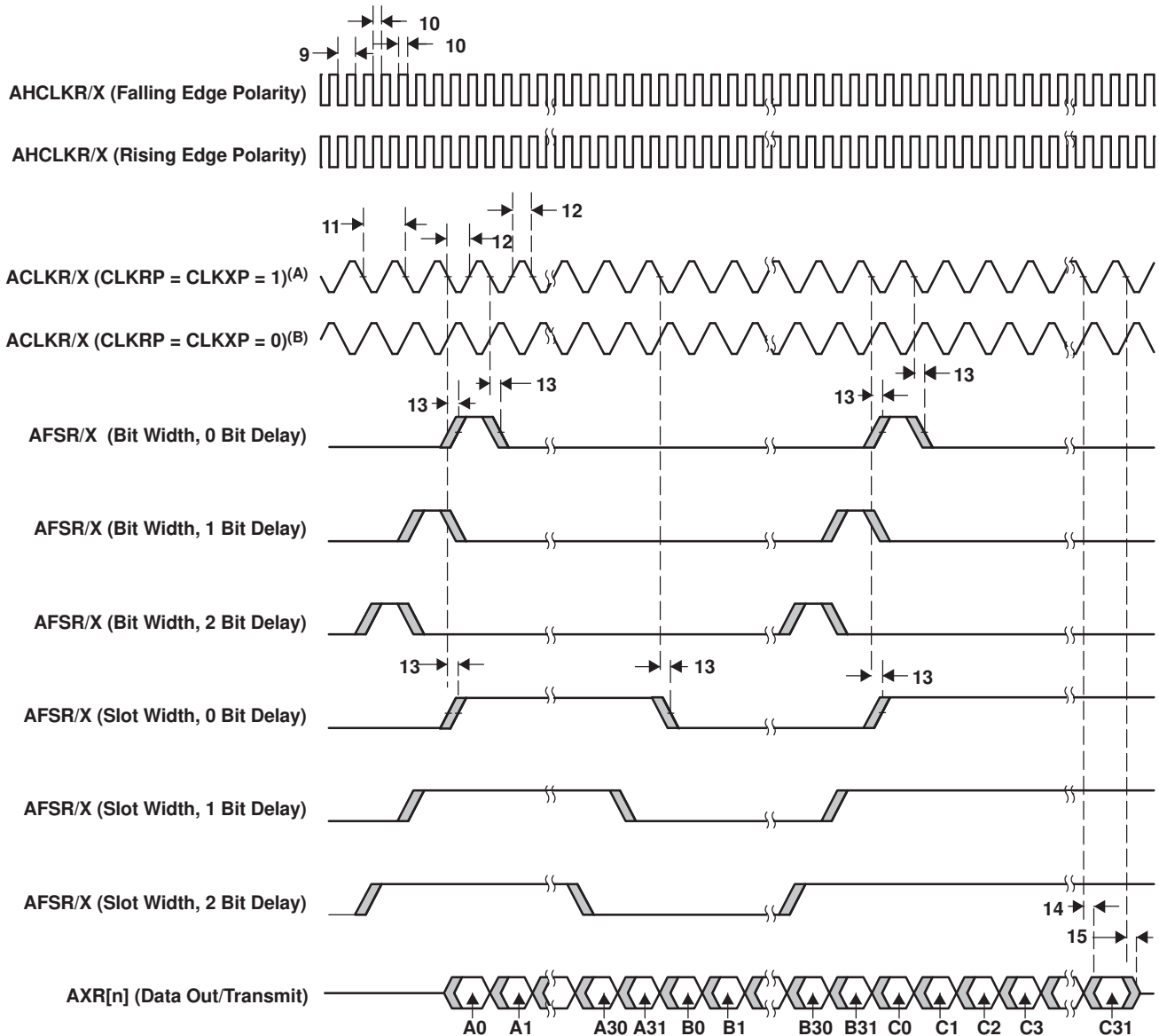
(6) AX - ACLKX1 period.

(7) McASP1 ACLKXCTL.ASYNC=1: Receiver is clocked by its own ACLKR1



- A. For CLKRP = CLKXP = 0, the McASP transmitter is configured for rising edge (to shift data out) and the McASP receiver is configured for falling edge (to shift data in).
- B. For CLKRP = CLKXP = 1, the McASP transmitter is configured for falling edge (to shift data out) and the McASP receiver is configured for rising edge (to shift data in).

**Figure 6-29. McASP Input Timings**



- A. For CLKRP = CLKXP = 1, the McASP transmitter is configured for falling edge (to shift data out) and the McASP receiver is configured for rising edge (to shift data in).
- B. For CLKRP = CLKXP = 0, the McASP transmitter is configured for rising edge (to shift data out) and the McASP receiver is configured for falling edge (to shift data in).

**Figure 6-30. McASP Output Timings**

## 6.17 Serial Peripheral Interface Ports (SPI0)

Figure 6-31 is a block diagram of the SPI module, which is a simple shift register and buffer plus control logic. Data is written to the shift register before transmission occurs and is read from the buffer at the end of transmission. The SPI can operate either as a master, in which case, it initiates a transfer and drives the SPIx\_CLK pin, or as a slave. Four clock phase and polarity options are supported as well as many data formatting options.

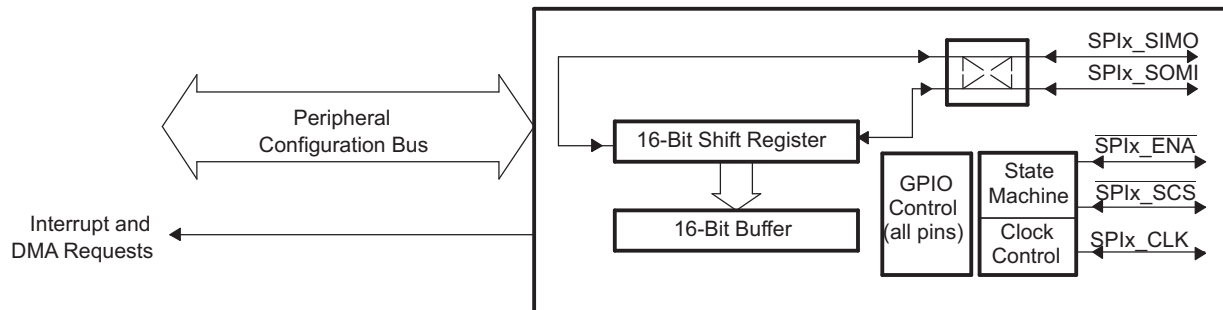


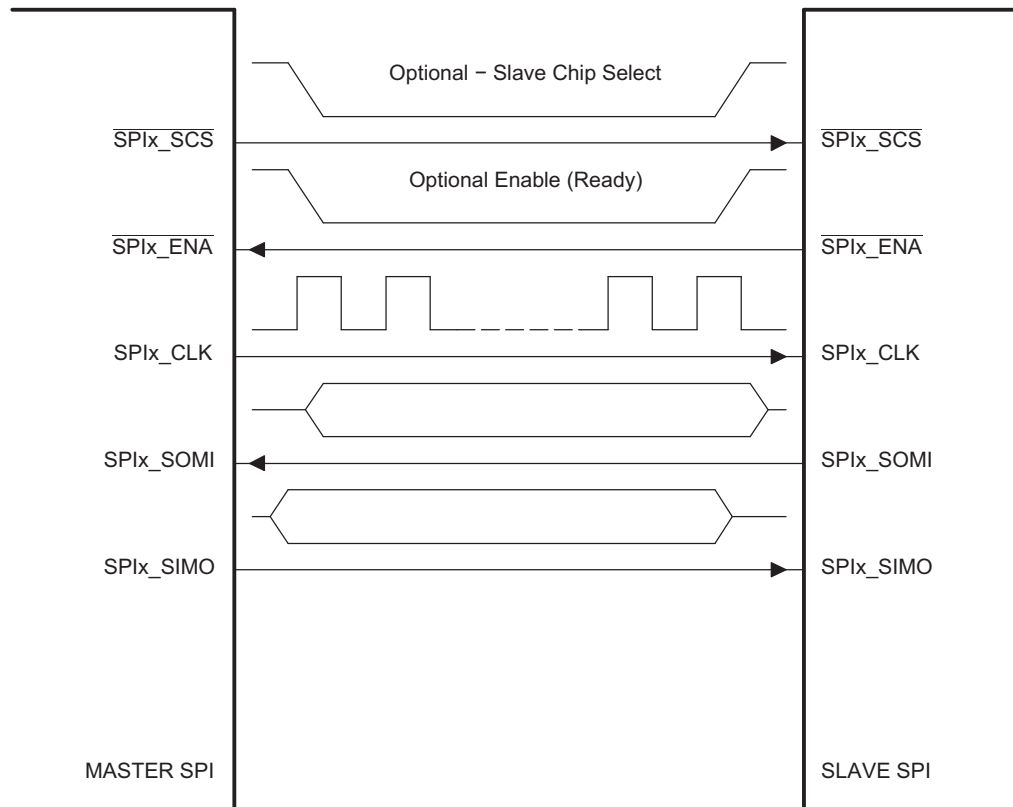
Figure 6-31. Block Diagram of SPI Module

The SPI supports 3-, 4-, and 5-pin operation with three basic pins (SPIx\_CLK, SPIx\_SIMO, and SPIx\_SOMI) and two optional pins (SPIx\_SCS, SPIx\_ENA).

The optional  $\overline{\text{SPIx\_SCS}}$  (Slave Chip Select) pin is most useful to enable in slave mode when there are other slave devices on the same SPI port. The C6743 will only shift data and drive the SPIx\_SOMI pin when  $\overline{\text{SPIx\_SCS}}$  is held low.

In slave mode,  $\overline{\text{SPIx\_ENA}}$  is an optional output and can be driven in either a push-pull or open-drain manner. The  $\overline{\text{SPIx\_ENA}}$  output provides the status of the internal transmit buffer (SPIDAT0/1 registers). In four-pin mode with the enable option,  $\overline{\text{SPIx\_ENA}}$  is asserted only when the transmit buffer is full, indicating that the slave is ready to begin another transfer. In five-pin mode, the  $\overline{\text{SPIx\_ENA}}$  is additionally qualified by  $\overline{\text{SPIx\_SCS}}$  being asserted. This allows a single handshake line to be shared by multiple slaves on the same SPI bus.

In master mode, the  $\overline{\text{SPIx\_ENA}}$  pin is an optional input and the master can be configured to delay the start of the next transfer until the slave asserts  $\overline{\text{SPIx\_ENA}}$ . The addition of this handshake signal simplifies SPI communications and, on average, increases SPI bus throughput since the master does not need to delay each transfer long enough to allow for the worst-case latency of the slave device. Instead, each transfer can begin as soon as both the master and slave have actually serviced the previous SPI transfer. Although the SPI module supports two interrupt outputs, SPIx\_INT1 is the only interrupt connected on this device. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide* (SPRUFK9) for more details.



**Figure 6-32. Illustration of SPI Master-to-SPI Slave Connection**

### 6.17.1 SPI Peripheral Registers Description(s)

Table 6-48 is a list of the SPI registers.

**Table 6-48. SPIx Configuration Registers**

| SPI0<br>BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                     |
|----------------------|---------------|--|
| 0x01C4 1000          | SPIGCR0       | Global Control Register 0                |
| 0x01C4 1004          | SPIGCR1       | Global Control Register 1                |
| 0x01C4 1008          | SPIINT0       | Interrupt Register                       |
| 0x01C4 100C          | SPIVLV        | Interrupt Level Register                 |
| 0x01C4 1010          | SPIFLG        | Flag Register                            |
| 0x01C4 1014          | SPIPC0        | Pin Control Register 0 (Pin Function)    |
| 0x01C4 1018          | SPIPC1        | Pin Control Register 1 (Pin Direction)   |
| 0x01C4 101C          | SPIPC2        | Pin Control Register 2 (Pin Data In)     |
| 0x01C4 1020          | SPIPC3        | Pin Control Register 3 (Pin Data Out)    |
| 0x01C4 1024          | SPIPC4        | Pin Control Register 4 (Pin Data Set)    |
| 0x01C4 1028          | SPIPC5        | Pin Control Register 5 (Pin Data Clear)  |
| 0x01C4 102C          | Reserved      | Reserved - Do not write to this register |
| 0x01C4 1030          | Reserved      | Reserved - Do not write to this register |
| 0x01C4 1034          | Reserved      | Reserved - Do not write to this register |
| 0x01C4 1038          | SPIDAT0       | Shift Register 0 (without format select) |
| 0x01C4 103C          | SPIDAT1       | Shift Register 1 (with format select)    |
| 0x01C4 1040          | SPIBUF        | Buffer Register                          |
| 0x01C4 1044          | SPIEMU        | Emulation Register                       |

**Table 6-48. SPIx Configuration Registers (continued)**

| SPI0<br>BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                     |
|----------------------|---------------|--|
| 0x01C4 1048          | SPIDELAY      | Delay Register                           |
| 0x01C4 104C          | SPIDEF        | Default Chip Select Register             |
| 0x01C4 1050          | SPIFMT0       | Format Register 0                        |
| 0x01C4 1054          | SPIFMT1       | Format Register 1                        |
| 0x01C4 1058          | SPIFMT2       | Format Register 2                        |
| 0x01C4 105C          | SPIFMT3       | Format Register 3                        |
| 0x01C4 1060          | Reserved      | Reserved - Do not write to this register |
| 0x01C4 1064          | INTVEC1       | Interrupt Vector for SPI INT1            |

## 6.17.2 SPI Electrical Data/Timing

### 6.17.2.1 Serial Peripheral Interface (SPI) Timing

Table 6-49 assumes testing over recommended operating conditions (see Figure 6-33 through Figure 6-36).

**Table 6-49. General Timing Requirements for SPI0 Master Modes<sup>(1)</sup>**

| NO. |                      |  | MIN  | MAX                   | UNIT |
|-----|----------------------|--|--|-----------------------|------|
| 1   | $t_{c(SPC)M}$        | Cycle Time, SPI0_CLK, All Master Modes   | greater of 3P or 20                            | 256P                  | ns   |
| 2   | $t_{w(SPCH)M}$       | Pulse Width High, SPI0_CLK, All Master Modes   | $0.5t_{c(SPC)M} - 1$                           |                       | ns   |
| 3   | $t_{w(SPCL)M}$       | Pulse Width Low, SPI0_CLK, All Master Modes  | $0.5t_{c(SPC)M} - 1$                           |                       | ns   |
| 4   | $t_{d(SIMO\_SPC)M}$  | Delay, initial data bit valid on SPI0_SIMO after initial edge on SPI0_CLK <sup>(2)</sup> | Polarity = 0, Phase = 0, to SPI0_CLK rising    | 5                     | ns   |
|     |                      |  | Polarity = 0, Phase = 1, to SPI0_CLK rising    | $-0.5t_{c(SPC)M} + 5$ |      |
|     |                      |  | Polarity = 1, Phase = 0, to SPI0_CLK falling   | 5                     |      |
|     |                      |  | Polarity = 1, Phase = 1, to SPI0_CLK falling   | $-0.5t_{c(SPC)M} + 5$ |      |
| 5   | $t_{d(SPC\_SIMO)M}$  | Delay, subsequent bits valid on SPI0_SIMO after transmit edge of SPI0_CLK                | Polarity = 0, Phase = 0, from SPI0_CLK rising  | 5                     | ns   |
|     |                      |  | Polarity = 0, Phase = 1, from SPI0_CLK falling | 5                     |      |
|     |                      |  | Polarity = 1, Phase = 0, from SPI0_CLK falling | 5                     |      |
|     |                      |  | Polarity = 1, Phase = 1, from SPI0_CLK rising  | 5                     |      |
| 6   | $t_{oh(SPC\_SIMO)M}$ | Output hold time, SPI0_SIMO valid after receive edge of SPI0_CLK                         | Polarity = 0, Phase = 0, from SPI0_CLK falling | $0.5t_{c(SPC)M} - 3$  | ns   |
|     |                      |  | Polarity = 0, Phase = 1, from SPI0_CLK rising  | $0.5t_{c(SPC)M} - 3$  |      |
|     |                      |  | Polarity = 1, Phase = 0, from SPI0_CLK rising  | $0.5t_{c(SPC)M} - 3$  |      |
|     |                      |  | Polarity = 1, Phase = 1, from SPI0_CLK falling | $0.5t_{c(SPC)M} - 3$  |      |
| 7   | $t_{su(SOMI\_SPC)M}$ | Input Setup Time, SPI0_SOMI valid before receive edge of SPI0_CLK                        | Polarity = 0, Phase = 0, to SPI0_CLK falling   | 0                     | ns   |
|     |                      |  | Polarity = 0, Phase = 1, to SPI0_CLK rising    | 0                     |      |
|     |                      |  | Polarity = 1, Phase = 0, to SPI0_CLK rising    | 0                     |      |
|     |                      |  | Polarity = 1, Phase = 1, to SPI0_CLK falling   | 0                     |      |
| 8   | $t_{ih(SPC\_SOMI)M}$ | Input Hold Time, SPI0_SOMI valid after receive edge of SPI0_CLK                          | Polarity = 0, Phase = 0, from SPI0_CLK falling | 5                     | ns   |
|     |                      |  | Polarity = 0, Phase = 1, from SPI0_CLK rising  | 5                     |      |
|     |                      |  | Polarity = 1, Phase = 0, from SPI0_CLK rising  | 5                     |      |
|     |                      |  | Polarity = 1, Phase = 1, from SPI0_CLK falling | 5                     |      |

(1) P = SYSCLK2 period

(2) First bit may be MSB or LSB depending upon SPI configuration. MO(0) refers to first bit and MO(n) refers to last bit output on SPI0\_SIMO. MI(0) refers to the first bit input and MI(n) refers to the last bit input on SPI0\_SOMI.

**Table 6-50. General Timing Requirements for SPI0 Slave Modes<sup>(1)</sup>**

| NO. |                      |  | MIN  | MAX                  | UNIT |
|-----|----------------------|--|--|----------------------|------|
| 9   | $t_{c(SPC)}S$        | Cycle Time, SPI0_CLK, All Slave Modes  | greater of 3P or 40                            |                      | ns   |
| 10  | $t_{w(SPCH)}S$       | Pulse Width High, SPI0_CLK, All Slave Modes  | 18   |                      | ns   |
| 11  | $t_{w(SPCL)}S$       | Pulse Width Low, SPI0_CLK, All Slave Modes   | 18   |                      | ns   |
| 12  | $t_{su(SOMI\_SPC)}S$ | Setup time, transmit data written to SPI before initial clock edge from master. <sup>(2) (3)</sup> | Polarity = 0, Phase = 0, to SPI0_CLK rising    | 2P                   | ns   |
|     |                      |  | Polarity = 0, Phase = 1, to SPI0_CLK rising    | 2P                   |      |
|     |                      |  | Polarity = 1, Phase = 0, to SPI0_CLK falling   | 2P                   |      |
|     |                      |  | Polarity = 1, Phase = 1, to SPI0_CLK falling   | 2P                   |      |
| 13  | $t_{d(SPC\_SOMI)}S$  | Delay, subsequent bits valid on SPI0_SOMI after transmit edge of SPI0_CLK                          | Polarity = 0, Phase = 0, from SPI0_CLK rising  | 18.5                 | ns   |
|     |                      |  | Polarity = 0, Phase = 1, from SPI0_CLK falling | 18.5                 |      |
|     |                      |  | Polarity = 1, Phase = 0, from SPI0_CLK falling | 18.5                 |      |
|     |                      |  | Polarity = 1, Phase = 1, from SPI0_CLK rising  | 18.5                 |      |
| 14  | $t_{oh(SPC\_SOMI)}S$ | Output hold time, SPI0_SOMI valid after receive edge of SPI0_CLK                                   | Polarity = 0, Phase = 0, from SPI0_CLK falling | $0.5t_{c(SPC)}S - 3$ | ns   |
|     |                      |  | Polarity = 0, Phase = 1, from SPI0_CLK rising  | $0.5t_{c(SPC)}S - 3$ |      |
|     |                      |  | Polarity = 1, Phase = 0, from SPI0_CLK rising  | $0.5t_{c(SPC)}S - 3$ |      |
|     |                      |  | Polarity = 1, Phase = 1, from SPI0_CLK falling | $0.5t_{c(SPC)}S - 3$ |      |
| 15  | $t_{su(SIMO\_SPC)}S$ | Input Setup Time, SPI0_SIMO valid before receive edge of SPI0_CLK                                  | Polarity = 0, Phase = 0, to SPI0_CLK falling   | 0                    | ns   |
|     |                      |  | Polarity = 0, Phase = 1, to SPI0_CLK rising    | 0                    |      |
|     |                      |  | Polarity = 1, Phase = 0, to SPI0_CLK rising    | 0                    |      |
|     |                      |  | Polarity = 1, Phase = 1, to SPI0_CLK falling   | 0                    |      |
| 16  | $t_{ih(SPC\_SIMO)}S$ | Input Hold Time, SPI0_SIMO valid after receive edge of SPI0_CLK                                    | Polarity = 0, Phase = 0, from SPI0_CLK falling | 5                    | ns   |
|     |                      |  | Polarity = 0, Phase = 1, from SPI0_CLK rising  | 5                    |      |
|     |                      |  | Polarity = 1, Phase = 0, from SPI0_CLK rising  | 5                    |      |
|     |                      |  | Polarity = 1, Phase = 1, from SPI0_CLK falling | 5                    |      |

(1) P = SYSCLK2 period

(2) First bit may be MSB or LSB depending upon SPI configuration. SO(0) refers to first bit and SO(n) refers to last bit output on SPI0\_SOMI. SI(0) refers to the first bit input and SI(n) refers to the last bit input on SPI0\_SIMO.

(3) Measured from the termination of the write of new data to the SPI module. In analyzing throughput requirements, additional internal bus cycles must be accounted for to allow data to be written to the SPI module by the DSP CPU.

**Table 6-51. Additional<sup>(1)</sup> SPI0 Master Timings, 4-Pin Enable Option<sup>(2)</sup> <sup>(3)</sup>**

| NO. | PARAMETER   | MIN   | MAX                         | UNIT |
|-----|---|---|-----------------------------|------|
| 17  | $t_{d(ENA\_SPC)M}$<br>Delay from slave assertion of $\overline{SPI0\_ENA}$ active to first $SPI0\_CLK$ from master. <sup>(4)</sup>  | Polarity = 0, Phase = 0, to $SPI0\_CLK$ rising    | $3P + 3.6$                  | ns   |
|     |   | Polarity = 0, Phase = 1, to $SPI0\_CLK$ rising    | $0.5t_{c(SPC)M} + 3P + 3.6$ |      |
|     |   | Polarity = 1, Phase = 0, to $SPI0\_CLK$ falling   | $3P + 3.6$                  |      |
|     |   | Polarity = 1, Phase = 1, to $SPI0\_CLK$ falling   | $0.5t_{c(SPC)M} + 3P + 3.6$ |      |
| 18  | $t_{d(SPC\_ENA)M}$<br>Max delay for slave to deassert $\overline{SPI0\_ENA}$ after final $SPI0\_CLK$ edge to ensure master does not begin the next transfer. <sup>(5)</sup> | Polarity = 0, Phase = 0, from $SPI0\_CLK$ falling | $0.5t_{c(SPC)M} + P + 5$    | ns   |
|     |   | Polarity = 0, Phase = 1, from $SPI0\_CLK$ falling | $P + 5$                     |      |
|     |   | Polarity = 1, Phase = 0, from $SPI0\_CLK$ rising  | $0.5t_{c(SPC)M} + P + 5$    |      |
|     |   | Polarity = 1, Phase = 1, from $SPI0\_CLK$ rising  | $P + 5$                     |      |

(1) These parameters are in addition to the general timings for SPI master modes (Table 6-49).

(2) P = SYSCLK2 period

(3) Figure shows only Polarity = 0, Phase = 0 as an example. Table gives parameters for all four master clocking modes.

(4) In the case where the master SPI is ready with new data before  $\overline{SPI0\_ENA}$  assertion.

(5) In the case where the master SPI is ready with new data before  $\overline{SPI0\_EN A}$  deassertion.

**Table 6-52. Additional<sup>(1)</sup> SPI0 Master Timings, 4-Pin Chip Select Option<sup>(2)</sup> <sup>(3)</sup>**

| NO. | PARAMETER  | MIN   | MAX                       | UNIT |
|-----|--|---|---------------------------|------|
| 19  | $t_{d(SCS\_SPC)M}$<br>Delay from $\overline{SPI0\_SCS}$ active to first $SPI0\_CLK$ <sup>(4)</sup> <sup>(5)</sup>                  | Polarity = 0, Phase = 0, to $SPI0\_CLK$ rising    | $2P - 5$                  | ns   |
|     |  | Polarity = 0, Phase = 1, to $SPI0\_CLK$ rising    | $0.5t_{c(SPC)M} + 2P - 5$ |      |
|     |  | Polarity = 1, Phase = 0, to $SPI0\_CLK$ falling   | $2P - 5$                  |      |
|     |  | Polarity = 1, Phase = 1, to $SPI0\_CLK$ falling   | $0.5t_{c(SPC)M} + 2P - 5$ |      |
| 20  | $t_{d(SPC\_SCS)M}$<br>Delay from final $SPI0\_CLK$ edge to master deasserting $\overline{SPI0\_SCS}$ <sup>(6)</sup> <sup>(7)</sup> | Polarity = 0, Phase = 0, from $SPI0\_CLK$ falling | $0.5t_{c(SPC)M} + P - 3$  | ns   |
|     |  | Polarity = 0, Phase = 1, from $SPI0\_CLK$ falling | $P - 3$                   |      |
|     |  | Polarity = 1, Phase = 0, from $SPI0\_CLK$ rising  | $0.5t_{c(SPC)M} + P - 3$  |      |
|     |  | Polarity = 1, Phase = 1, from $SPI0\_CLK$ rising  | $P - 3$                   |      |

(1) These parameters are in addition to the general timings for SPI master modes (Table 6-49).

(2) P = SYSCLK2 period

(3) Figure shows only Polarity = 0, Phase = 0 as an example. Table gives parameters for all four master clocking modes.

(4) In the case where the master SPI is ready with new data before  $\overline{SPI0\_SCS}$  assertion.

(5) This delay can be increased under software control by the register bit field SPIDELAY.C2TDELAY[4:0].

(6) Except for modes when SPIDAT1.CSHOLD is enabled and there is additional data to transmit. In this case,  $\overline{SPI0\_SCS}$  will remain asserted.

(7) This delay can be increased under software control by the register bit field SPIDELAY.T2CDELAY[4:0].

**Table 6-53. Additional<sup>(1)</sup> SPI0 Master Timings, 5-Pin Option<sup>(2) (3)</sup>**

| NO. |                      |   | MIN   | MAX                         | UNIT |
|-----|----------------------|---|---|-----------------------------|------|
| 18  | $t_{d(SPC\_ENA)M}$   | Max delay for slave to deassert $\overline{SPI0\_ENA}$ after final $SPI0\_CLK$ edge to ensure master does not begin the next transfer. <sup>(4)</sup>           | Polarity = 0, Phase = 0, from $SPI0\_CLK$ falling | $0.5t_{c(SPC)M} + P + 5$    | ns   |
|     |                      |   | Polarity = 0, Phase = 1, from $SPI0\_CLK$ falling | $P + 5$                     |      |
|     |                      |   | Polarity = 1, Phase = 0, from $SPI0\_CLK$ rising  | $0.5t_{c(SPC)M} + P + 5$    |      |
|     |                      |   | Polarity = 1, Phase = 1, from $SPI0\_CLK$ rising  | $P + 5$                     |      |
| 20  | $t_{d(SPC\_SCS)M}$   | Delay from final $SPI0\_CLK$ edge to master deasserting $\overline{SPI0\_SCS}$ <sup>(5) (6)</sup>   | Polarity = 0, Phase = 0, from $SPI0\_CLK$ falling | $0.5t_{c(SPC)M} + P - 3$    | ns   |
|     |                      |   | Polarity = 0, Phase = 1, from $SPI0\_CLK$ falling | $P - 3$                     |      |
|     |                      |   | Polarity = 1, Phase = 0, from $SPI0\_CLK$ rising  | $0.5t_{c(SPC)M} + P - 3$    |      |
|     |                      |   | Polarity = 1, Phase = 1, from $SPI0\_CLK$ rising  | $P - 3$                     |      |
| 21  | $t_{d(SCSL\_ENAL)M}$ | Max delay for slave SPI to drive $\overline{SPI0\_ENA}$ valid after master asserts $\overline{SPI0\_SCS}$ to delay the master from beginning the next transfer, |   | $C2TDELAY + P$              | ns   |
| 22  | $t_{d(SCS\_SPC)M}$   | Delay from $\overline{SPI0\_SCS}$ active to first $SPI0\_CLK$ <sup>(7) (8) (9)</sup>  | Polarity = 0, Phase = 0, to $SPI0\_CLK$ rising    | $2P - 5$                    | ns   |
|     |                      |   | Polarity = 0, Phase = 1, to $SPI0\_CLK$ rising    | $0.5t_{c(SPC)M} + 2P - 5$   |      |
|     |                      |   | Polarity = 1, Phase = 0, to $SPI0\_CLK$ falling   | $2P - 5$                    |      |
|     |                      |   | Polarity = 1, Phase = 1, to $SPI0\_CLK$ falling   | $0.5t_{c(SPC)M} + 2P - 5$   |      |
| 23  | $t_{d(ENA\_SPC)M}$   | Delay from assertion of $\overline{SPI0\_ENA}$ low to first $SPI0\_CLK$ edge. <sup>(10)</sup>   | Polarity = 0, Phase = 0, to $SPI0\_CLK$ rising    | $3P + 3.6$                  | ns   |
|     |                      |   | Polarity = 0, Phase = 1, to $SPI0\_CLK$ rising    | $0.5t_{c(SPC)M} + 3P + 3.6$ |      |
|     |                      |   | Polarity = 1, Phase = 0, to $SPI0\_CLK$ falling   | $3P + 3.6$                  |      |
|     |                      |   | Polarity = 1, Phase = 1, to $SPI0\_CLK$ falling   | $0.5t_{c(SPC)M} + 3P + 3.6$ |      |

(1) These parameters are in addition to the general timings for SPI master modes (Table 6-50).

(2) P = SYSCLK2 period

(3) Figure shows only Polarity = 0, Phase = 0 as an example. Table gives parameters for all four master clocking modes.

(4) In the case where the master SPI is ready with new data before  $\overline{SPI0\_ENA}$  deassertion.

(5) Except for modes when  $SPIDAT1.CSHOLD$  is enabled and there is additional data to transmit. In this case,  $\overline{SPI0\_SCS}$  will remain asserted.

(6) This delay can be increased under software control by the register bit field  $SPIDELAY.T2CDELAY[4:0]$ .

(7) If  $\overline{SPI0\_ENA}$  is asserted immediately such that the transmission is not delayed by  $\overline{SPI0\_ENA}$ .

(8) In the case where the master SPI is ready with new data before  $\overline{SPI0\_SCS}$  assertion.

(9) This delay can be increased under software control by the register bit field  $SPIDELAY.C2TDELAY[4:0]$ .

(10) If  $\overline{SPI0\_ENA}$  was initially deasserted high and  $SPI0\_CLK$  is delayed.

**Table 6-54. Additional<sup>(1)</sup> SPI0 Slave Timings, 4-Pin Enable Option<sup>(2)</sup> <sup>(3)</sup>**

| NO. |                     |   | MIN  | MAX                           | UNIT                             |    |
|-----|---------------------|---|--|-------------------------------|----------------------------------|----|
| 24  | $t_{d(SPC\_ENAH)S}$ | Delay from final SPI0_CLK edge to slave deasserting SPI0_ENA. | Polarity = 0, Phase = 0, from SPI0_CLK falling | 1.5 P - 3                     | 2.5 P + 18.5                     | ns |
|     |                     |   | Polarity = 0, Phase = 1, from SPI0_CLK falling | $-0.5t_{c(SPC)M} + 1.5 P - 3$ | $-0.5t_{c(SPC)M} + 2.5 P + 18.5$ |    |
|     |                     |   | Polarity = 1, Phase = 0, from SPI0_CLK rising  | 1.5 P - 3                     | 2.5 P + 18.5                     |    |
|     |                     |   | Polarity = 1, Phase = 1, from SPI0_CLK rising  | $-0.5t_{c(SPC)M} + 1.5 P - 3$ | $-0.5t_{c(SPC)M} + 2.5 P + 18.5$ |    |

(1) These parameters are in addition to the general timings for SPI slave modes (Table 6-50).

(2) P = SYSCLK2 period

(3) Figure shows only Polarity = 0, Phase = 0 as an example. Table gives parameters for all four slave clocking modes.

**Table 6-55. Additional<sup>(1)</sup> SPI0 Slave Timings, 4-Pin Chip Select Option<sup>(2)</sup> <sup>(3)</sup>**

| NO. |                        |   | MIN  | MAX                      | UNIT |
|-----|------------------------|---|--|--------------------------|------|
| 25  | $t_{d(SCSL\_SPC)S}$    | Required delay from $\overline{SPI0\_SCS}$ asserted at slave to first SPI0_CLK edge at slave. | 2P   |                          | ns   |
| 26  | $t_{d(SPC\_SCSH)S}$    | Required delay from final SPI0_CLK edge before $\overline{SPI0\_SCS}$ is deasserted.          | Polarity = 0, Phase = 0, from SPI0_CLK falling | $0.5t_{c(SPC)M} + P + 5$ | ns   |
|     |                        |   | Polarity = 0, Phase = 1, from SPI0_CLK falling | P + 5                    |      |
|     |                        |   | Polarity = 1, Phase = 0, from SPI0_CLK rising  | $0.5t_{c(SPC)M} + P + 5$ |      |
|     |                        |   | Polarity = 1, Phase = 1, from SPI0_CLK rising  | P + 5                    |      |
| 27  | $t_{ena(SCSL\_SOMI)S}$ | Delay from master asserting SPI0_SCS to slave driving SPI0_SOMI valid                         |  | P + 18.5                 |      |
| 28  | $t_{dis(SCSH\_SOMI)S}$ | Delay from master deasserting $\overline{SPI0\_SCS}$ to slave 3-stating SPI0_SOMI             |  | P + 18.5                 |      |

(1) These parameters are in addition to the general timings for SPI slave modes (Table 6-50).

(2) P = SYSCLK2 period

(3) Figure shows only Polarity = 0, Phase = 0 as an example. Table gives parameters for all four slave clocking modes.

**Table 6-56. Additional<sup>(1)</sup> SPI0 Slave Timings, 5-Pin Option<sup>(2)</sup> <sup>(3)</sup>**

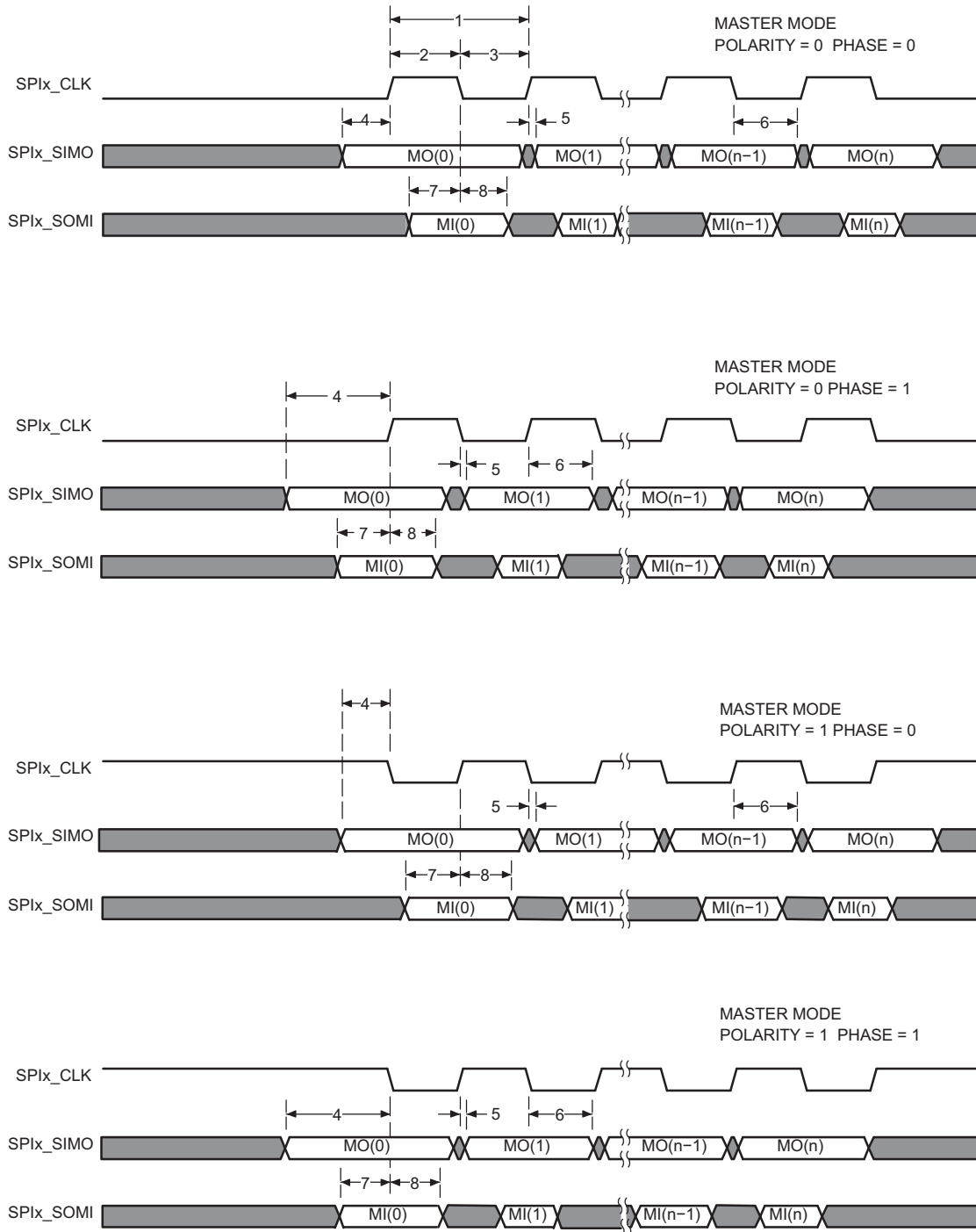
| NO. | PARAMETER              |   | MIN  | MAX                  | UNIT |
|-----|------------------------|---|--|----------------------|------|
| 25  | $t_{d(SCSL\_SPC)S}$    | Required delay from $\overline{SPI0\_SCS}$ asserted at slave to first SPI0_CLK edge at slave.               | 2P   |                      | ns   |
| 26  | $t_{d(SPC\_SCSH)S}$    | Required delay from final SPI0_CLK edge before $\overline{SPI0\_SCS}$ is deasserted.                        | Polarity = 0, Phase = 0, from SPI0_CLK falling | $0.5t_{c(SPC)M} + 5$ | ns   |
|     |                        |   | Polarity = 0, Phase = 1, from SPI0_CLK falling | 5                    |      |
|     |                        |   | Polarity = 1, Phase = 0, from SPI0_CLK rising  | $0.5t_{c(SPC)M} + 5$ |      |
|     |                        |   | Polarity = 1, Phase = 1, from SPI0_CLK rising  | 5                    |      |
| 27  | $t_{ena(SCSL\_SOMI)S}$ | Delay from master asserting $\overline{SPI0\_SCS}$ to slave driving SPI0_SOMI valid                         |  | P + 18.5             | ns   |
| 28  | $t_{dis(SCSH\_SOMI)S}$ | Delay from master deasserting $\overline{SPI0\_SCS}$ to slave 3-stating SPI0_SOMI                           |  | P + 18.5             | ns   |
| 29  | $t_{ena(SCSL\_ENA)S}$  | Delay from master deasserting $\overline{SPI0\_SCS}$ to slave driving SPI0_ENA valid                        |  | 18.5                 | ns   |
| 30  | $t_{dis(SPC\_ENA)S}$   | Delay from final clock receive edge on SPI0_CLK to slave 3-stating or driving high SPI0_ENA. <sup>(4)</sup> | Polarity = 0, Phase = 0, from SPI0_CLK falling | 2.5 P + 18.5         | ns   |
|     |                        |   | Polarity = 0, Phase = 1, from SPI0_CLK rising  | 2.5 P + 18.5         |      |
|     |                        |   | Polarity = 1, Phase = 0, from SPI0_CLK rising  | 2.5 P + 18.5         |      |
|     |                        |   | Polarity = 1, Phase = 1, from SPI0_CLK falling | 2.5 P + 18.5         |      |

(1) These parameters are in addition to the general timings for SPI slave modes (Table 6-50).

(2) P = SYSCLK2 period

(3) Figure shows only Polarity = 0, Phase = 0 as an example. Table gives parameters for all four slave clocking modes.

(4) SPI0\_ENA is driven low after the transmission completes if the SPIINT0.ENABLE\_HIGHZ bit is programmed to 0. Otherwise it is tri-stated. If tri-stated, an external pullup resistor should be used to provide a valid level to the master. This option is useful when tying several SPI slave devices to a single master.



**Figure 6-33. SPI Timings—Master Mode**

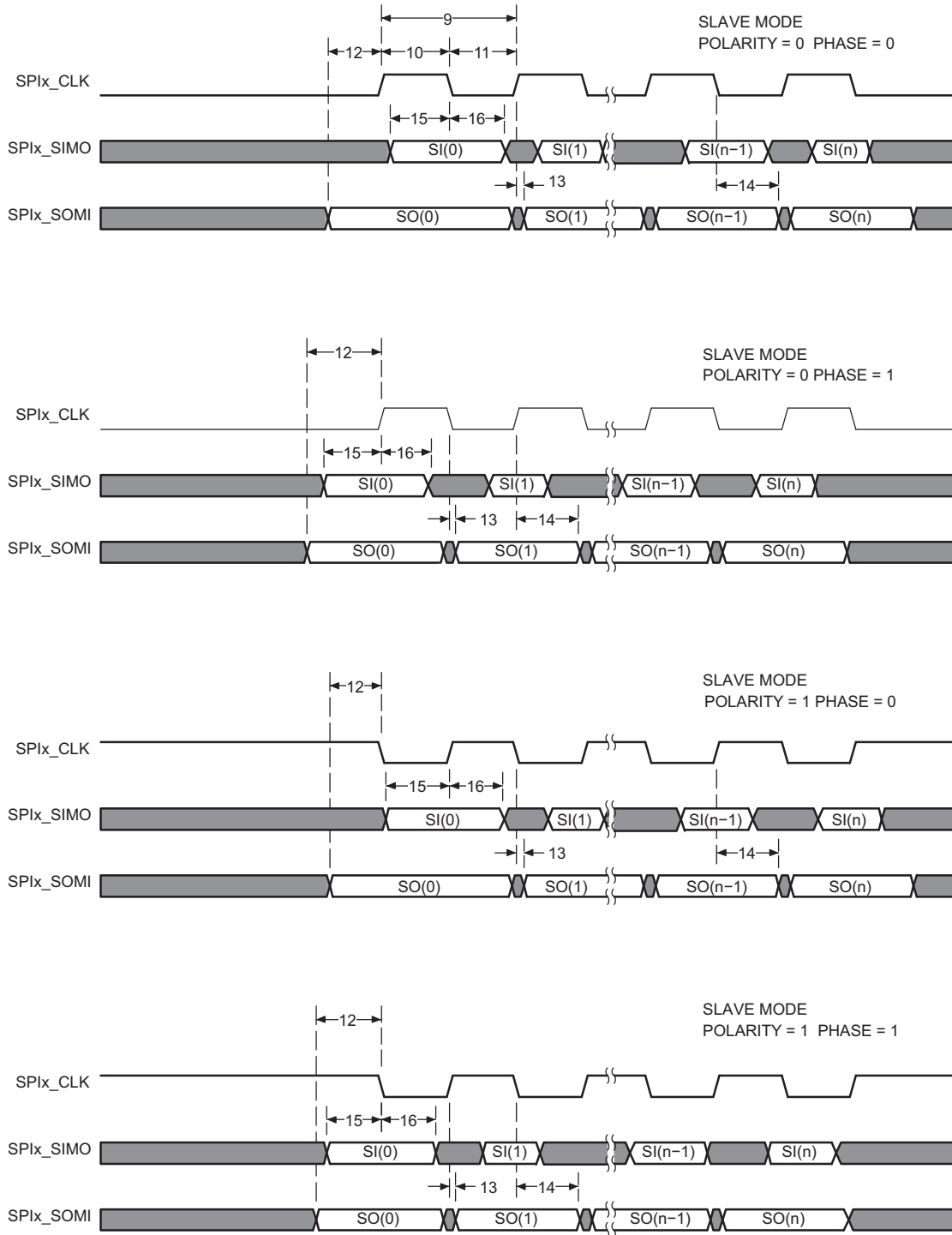
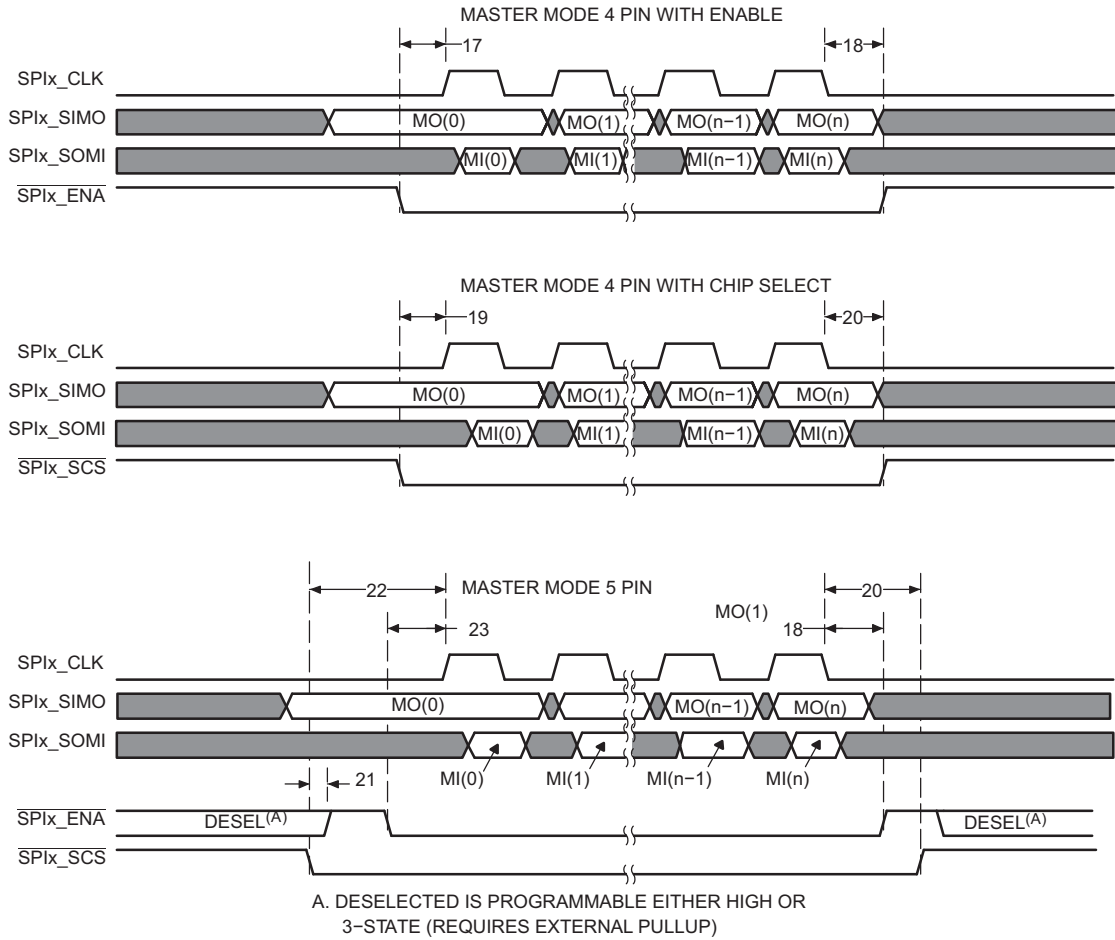


Figure 6-34. SPI Timings—Slave Mode



**Figure 6-35. SPI Timings—Master Mode (4-Pin and 5-Pin)**

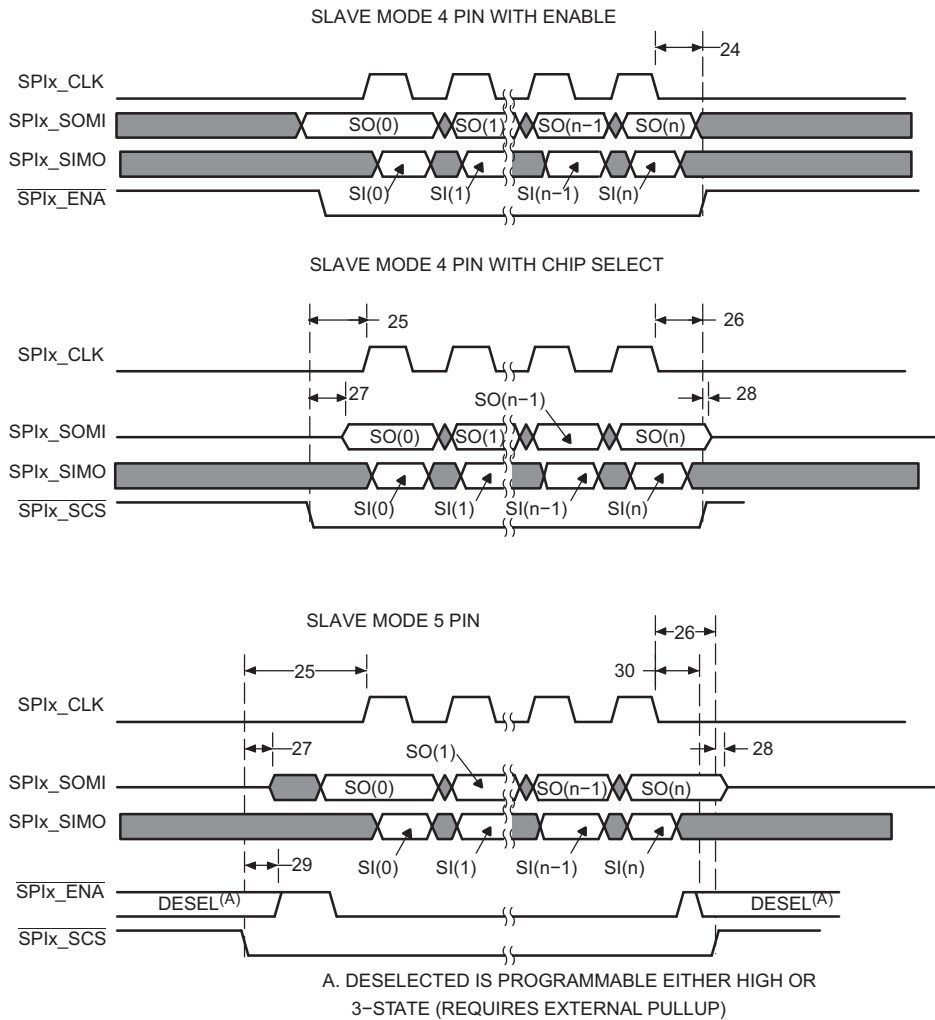


Figure 6-36. SPI Timings—Slave Mode (4-Pin and 5-Pin)

## 6.18 Enhanced Capture (eCAP) Peripheral

The device contains up to three enhanced capture (eCAP) modules. [Figure 6-37](#) shows a functional block diagram of a module. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide* ([SPRUFK9](#)) for more details.

Uses for ECAP include:

- Speed measurements of rotating machinery (e.g. toothed sprockets sensed via Hall sensors)
- Elapsed time measurements between position sensor triggers
- Period and duty cycle measurements of Pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

The ECAP module described in this specification includes the following features:

- 32 bit time base
- 4 event time-stamp registers (each 32 bits)
- Edge polarity selection for up to 4 sequenced time-stamp capture events
- Interrupt on either of the 4 events
- Single shot capture of up to 4 event time-stamps
- Continuous mode capture of time-stamps in a 4 deep circular buffer
- Absolute time-stamp capture
- Difference mode time-stamp capture
- All the above resources are dedicated to a single input pin

The eCAP modules are clocked at the SYSCLK2 rate.

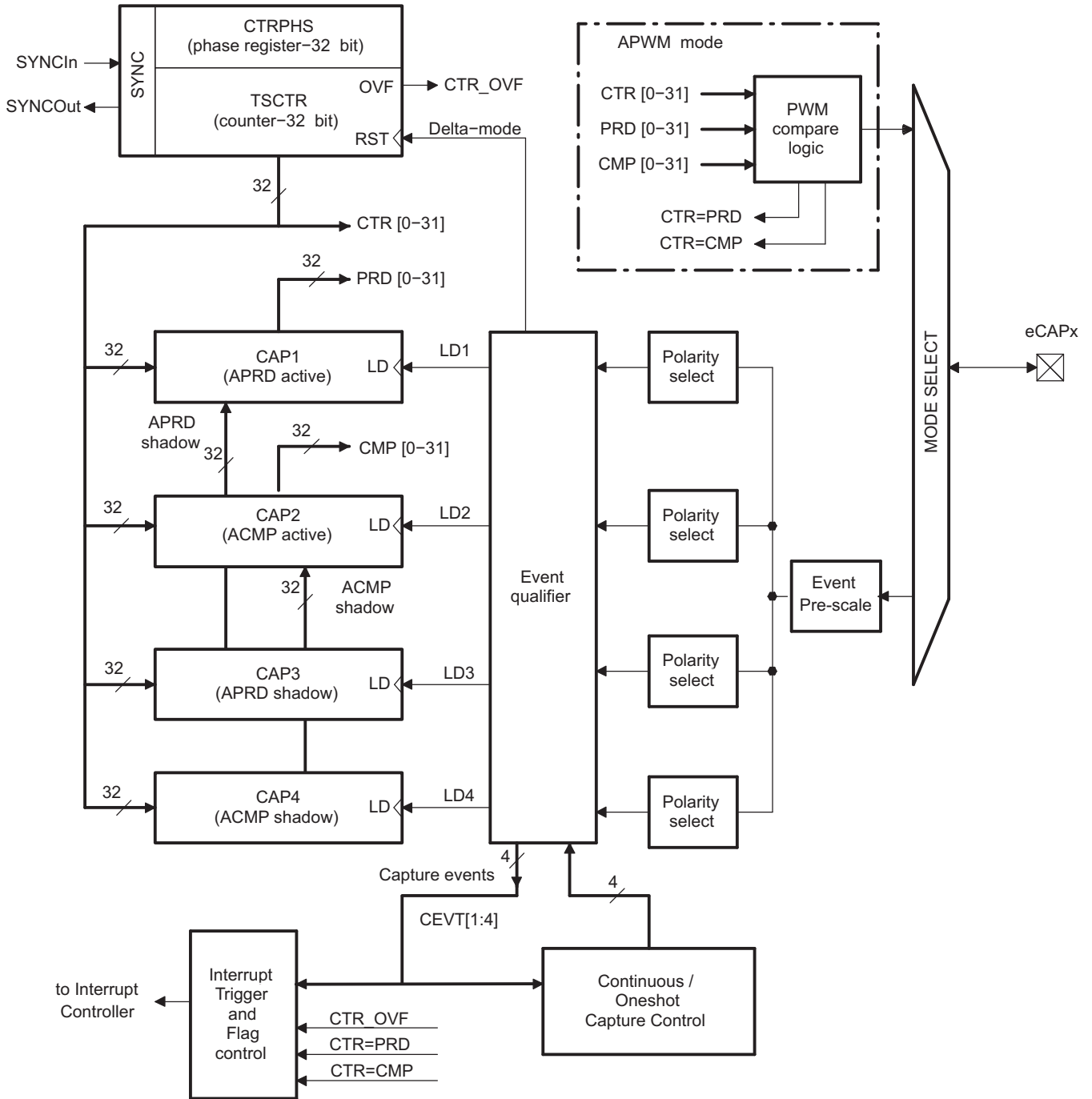


Figure 6-37. eCAP Functional Block Diagram

Table 6-57 is the list of the ECAP registers.

**Table 6-57. ECAPx Configuration Registers**

| ECAP0<br>BYTE ADDRESS | ECAP1<br>BYTE ADDRESS | ECAP2<br>BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                |
|-----------------------|-----------------------|-----------------------|---------------|-------------------------------------|
| 0x01F0 6000           | 0x01F0 7000           | 0x01F0 8000           | TSCTR         | Time-Stamp Counter                  |
| 0x01F0 6004           | 0x01F0 7004           | 0x01F0 8004           | CTRPHS        | Counter Phase Offset Value Register |
| 0x01F0 6008           | 0x01F0 7008           | 0x01F0 8008           | CAP1          | Capture 1 Register                  |
| 0x01F0 600C           | 0x01F0 700C           | 0x01F0 800C           | CAP2          | Capture 2 Register                  |
| 0x01F0 6010           | 0x01F0 7010           | 0x01F0 8010           | CAP3          | Capture 3 Register                  |
| 0x01F0 6014           | 0x01F0 7014           | 0x01F0 8014           | CAP4          | Capture 4 Register                  |
| 0x01F0 6028           | 0x01F0 7028           | 0x01F0 8028           | ECCTL1        | Capture Control Register 1          |
| 0x01F0 602A           | 0x01F0 702A           | 0x01F0 802A           | ECCTL2        | Capture Control Register 2          |
| 0x01F0 602C           | 0x01F0 702C           | 0x01F0 802C           | ECEINT        | Capture Interrupt Enable Register   |
| 0x01F0 602E           | 0x01F0 702E           | 0x01F0 802E           | ECFLG         | Capture Interrupt Flag Register     |
| 0x01F0 6030           | 0x01F0 7030           | 0x01F0 8030           | ECCLR         | Capture Interrupt Clear Register    |
| 0x01F0 6032           | 0x01F0 7032           | 0x01F0 8032           | ECFRC         | Capture Interrupt Force Register    |
| 0x01F0 605C           | 0x01F0 705C           | 0x01F0 805C           | REVID         | Revision ID                         |

Table 6-58 shows the eCAP timing requirement and Table 6-59 shows the eCAP switching characteristics.

**Table 6-58. Enhanced Capture (eCAP) Timing Requirement**

|              |                           | TEST CONDITIONS | MIN           | MAX | UNIT   |
|--------------|---------------------------|-----------------|---------------|-----|--------|
| $t_{w(CAP)}$ | Capture input pulse width | Asynchronous    | $2t_{c(SCO)}$ |     | cycles |
|              |                           | Synchronous     | $2t_{c(SCO)}$ |     | cycles |

**Table 6-59. eCAP Switching Characteristics**

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

### 6.19 Enhanced Quadrature Encoder (eQEP) Peripheral

The device contains up to two enhanced quadrature encoder (eQEP) modules. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

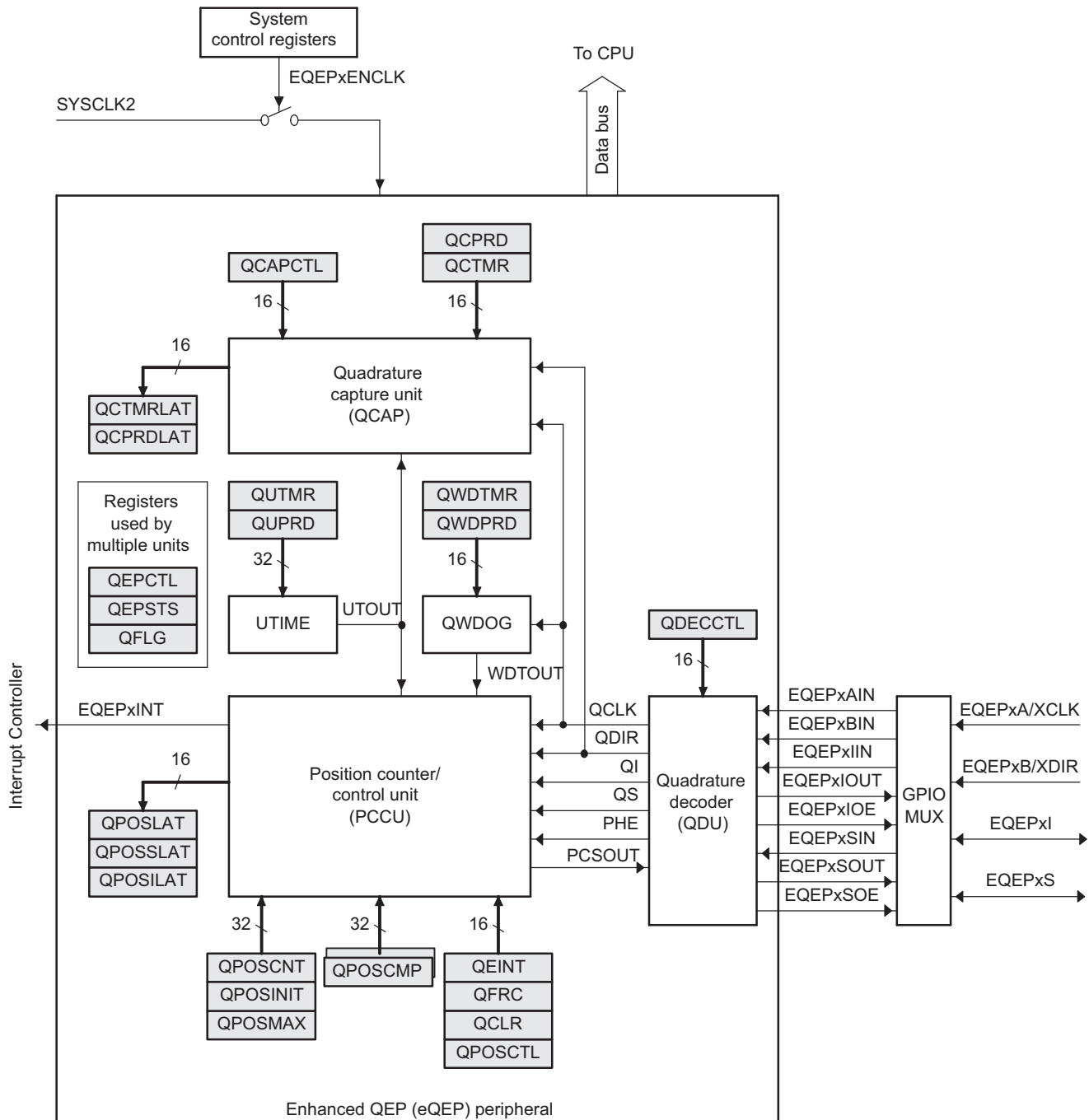


Figure 6-38. eQEP Functional Block Diagram

Table 6-60 is the list of the EQEP registers.

Table 6-61 shows the eQEP timing requirement and Table 6-62 shows the eQEP switching characteristics.

**Table 6-60. EQEP Registers**

| EQEP0<br>BYTE ADDRESS | EQEP1<br>BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                   |
|-----------------------|-----------------------|---------------|--|
| 0x01F0 9000           | 0x01F0 A000           | QPOSCNT       | eQEP Position Counter                  |
| 0x01F0 9004           | 0x01F0 A004           | QPOSINIT      | eQEP Initialization Position Count     |
| 0x01F0 9008           | 0x01F0 A008           | QPOSMAX       | eQEP Maximum Position Count            |
| 0x01F0 900C           | 0x01F0 A00C           | QPOSCMP       | eQEP Position-compare                  |
| 0x01F0 9010           | 0x01F0 A010           | QPOSILAT      | eQEP Index Position Latch              |
| 0x01F0 9014           | 0x01F0 A014           | QPOSSLAT      | eQEP Strobe Position Latch             |
| 0x01F0 9018           | 0x01F0 A018           | QPOSLAT       | eQEP Position Latch                    |
| 0x01F0 901C           | 0x01F0 A01C           | QUTMR         | eQEP Unit Timer                        |
| 0x01F0 9020           | 0x01F0 A020           | QUPRD         | eQEP Unit Period Register              |
| 0x01F0 9024           | 0x01F0 A024           | QWDTMR        | eQEP Watchdog Timer                    |
| 0x01F0 9026           | 0x01F0 A026           | QWDPRD        | eQEP Watchdog Period Register          |
| 0x01F0 9028           | 0x01F0 A028           | QDECCTL       | eQEP Decoder Control Register          |
| 0x01F0 902A           | 0x01F0 A02A           | QEPCTL        | eQEP Control Register                  |
| 0x01F0 902C           | 0x01F0 A02C           | QCAPCTL       | eQEP Capture Control Register          |
| 0x01F0 902E           | 0x01F0 A02E           | QPOSCTL       | eQEP Position-compare Control Register |
| 0x01F0 9030           | 0x01F0 A030           | QEINT         | eQEP Interrupt Enable Register         |
| 0x01F0 9032           | 0x01F0 A032           | QFLG          | eQEP Interrupt Flag Register           |
| 0x01F0 9034           | 0x01F0 A034           | QCLR          | eQEP Interrupt Clear Register          |
| 0x01F0 9036           | 0x01F0 A036           | QFRC          | eQEP Interrupt Force Register          |
| 0x01F0 9038           | 0x01F0 A038           | QEPSTS        | eQEP Status Register                   |
| 0x01F0 903A           | 0x01F0 A03A           | QCTMR         | eQEP Capture Timer                     |
| 0x01F0 903C           | 0x01F0 A03C           | QCPRD         | eQEP Capture Period Register           |
| 0x01F0 903E           | 0x01F0 A03E           | QCTMLAT       | eQEP Capture Timer Latch               |
| 0x01F0 9040           | 0x01F0 A040           | QCPRLAT       | eQEP Capture Period Latch              |
| 0x01F0 905C           | 0x01F0 A05C           | REVID         | eQEP Revision ID                       |

**Table 6-61. Enhanced Quadrature Encoder Pulse (eQEP) Timing Requirements**

|                 |                           | TEST CONDITIONS          | MIN           | MAX | UNIT   |
|-----------------|---------------------------|--------------------------|---------------|-----|--------|
| $t_{w(QEPP)}$   | QEP input period          | Asynchronous/synchronous | $2t_{c(SCO)}$ |     | cycles |
| $t_{w(INDEXH)}$ | QEP Index Input High time | Asynchronous/synchronous | $2t_{c(SCO)}$ |     | cycles |
| $t_{w(INDEXL)}$ | QEP Index Input Low time  | Asynchronous/synchronous | $2t_{c(SCO)}$ |     | cycles |
| $t_{w(STROBH)}$ | QEP Strobe High time      | Asynchronous/synchronous | $2t_{c(SCO)}$ |     | cycles |
| $t_{w(STROBL)}$ | QEP Strobe Input Low time | Asynchronous/synchronous | $2t_{c(SCO)}$ |     | cycles |

**Table 6-62. eQEP Switching Characteristics**

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

## 6.20 Enhanced Pulse Width Modulator (eHRPWM) Modules

The device contains up to three enhanced PWM Modules (eHRPWM). Figure 6-39 shows a block diagram of multiple eHRPWM modules. Figure 4-4 shows the signal interconnections with the eHRPWM. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

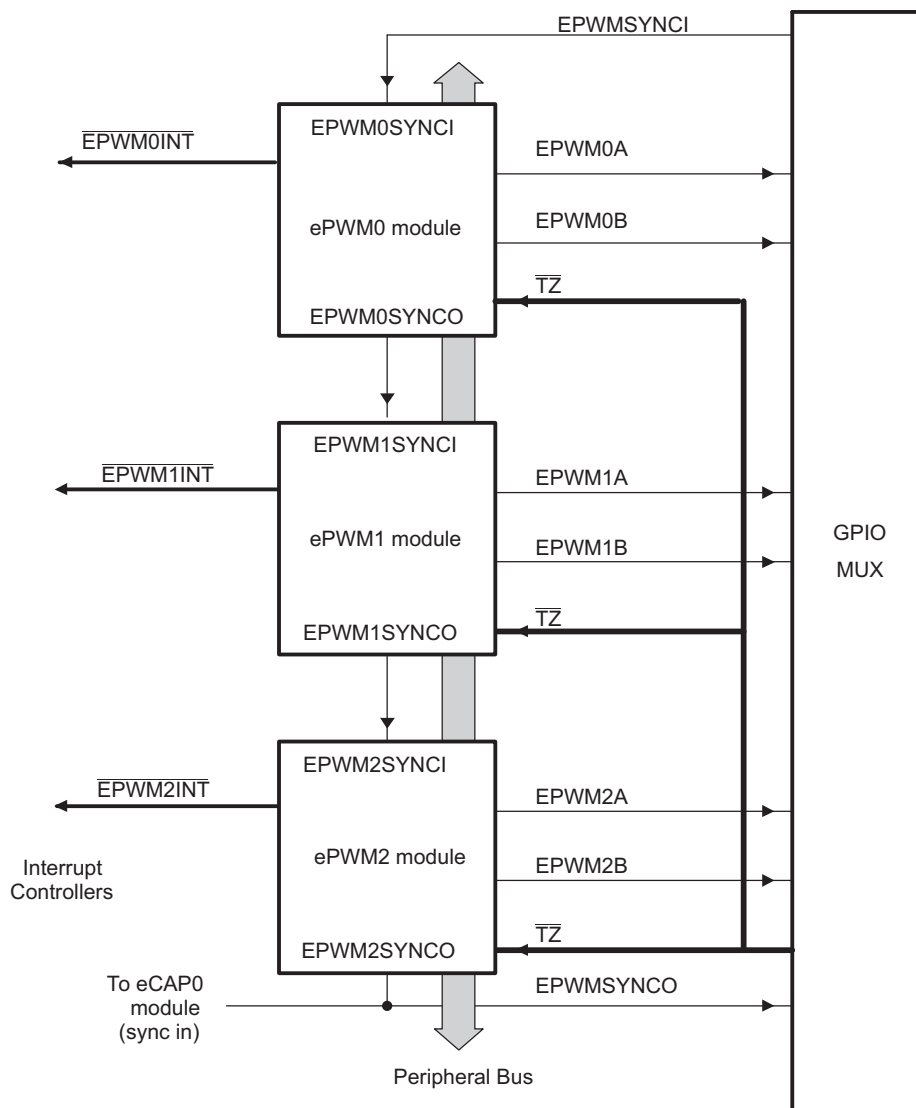


Figure 6-39. Multiple PWM Modules in a C6743 System

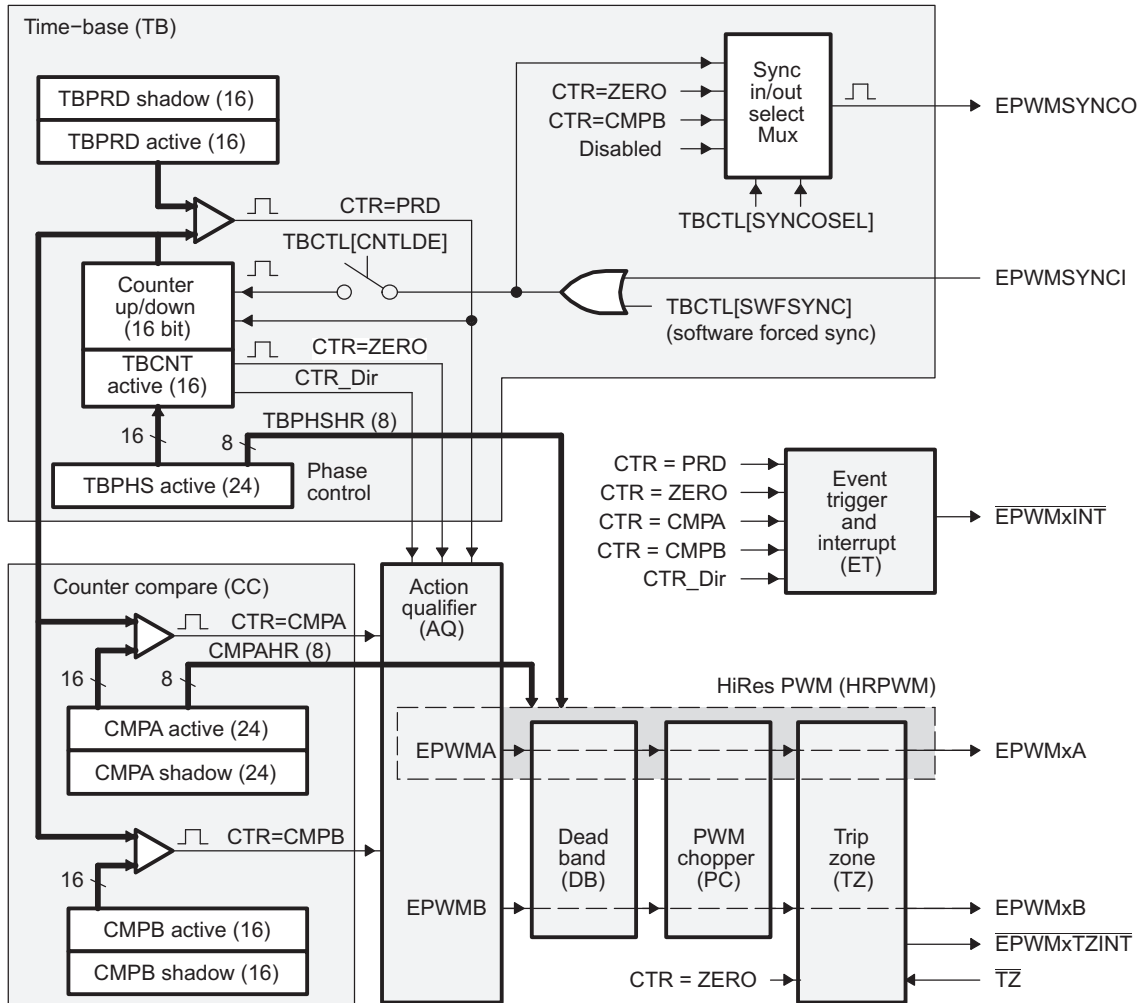


Figure 6-40. eHRPWM Sub-Modules Showing Critical Internal Signal Interconnections

Table 6-63. eHRPWM Module Control and Status Registers Grouped by Submodule

| eHRPWM1<br>BYTE ADDRESS                    | eHRPWM2<br>BYTE ADDRESS | eHRPWM3<br>BYTE ADDRESS | REGISTER<br>NAME | SIZE<br>(x16) | SHADOW | REGISTER DESCRIPTION                                  |
|--|-------------------------|-------------------------|------------------|---------------|--------|---|
| <b>Time-Base Submodule Registers</b>       |                         |                         |                  |               |        |   |
| 0x01F0 0000                                | 0x01F0 2000             | 0x01F0 4000             | TBCTL            | 1             | No     | Time-Base Control Register                            |
| 0x01F0 0002                                | 0x01F0 2002             | 0x01F0 4002             | TBSTS            | 1             | No     | Time-Base Status Register                             |
| 0x01F0 0004                                | 0x01F0 2004             | 0x01F0 4004             | TBPHSHR          | 1             | No     | Extension for HRPWM Phase Register<br>(1)             |
| 0x01F0 0006                                | 0x01F0 2006             | 0x01F0 4006             | TBPHS            | 1             | No     | Time-Base Phase Register                              |
| 0x01F0 0008                                | 0x01F0 2008             | 0x01F0 4008             | TBCNT            | 1             | No     | Time-Base Counter Register                            |
| 0x01F0 000A                                | 0x01F0 200A             | 0x01F0 400A             | TBPRD            | 1             | Yes    | Time-Base Period Register                             |
| <b>Counter-Compare Submodule Registers</b> |                         |                         |                  |               |        |   |
| 0x01F0 000E                                | 0x01F0 200E             | 0x01F0 400E             | CMPCTL           | 1             | No     | Counter-Compare Control Register                      |
| 0x01F0 0010                                | 0x01F0 2010             | 0x01F0 4010             | CMPAHR           | 1             | No     | Extension for HRPWM Counter-Compare A Register<br>(1) |
| 0x01F0 0012                                | 0x01F0 2012             | 0x01F0 4012             | CMPA             | 1             | Yes    | Counter-Compare A Register                            |
| 0x01F0 0014                                | 0x01F0 2014             | 0x01F0 4014             | CMPB             | 1             | Yes    | Counter-Compare B Register                            |

(1) These registers are only available on eHRPWM instances that include the high-resolution PWM (HRPWM) extension; otherwise, these locations are reserved.

**Table 6-63. eHRPWM Module Control and Status Registers Grouped by Submodule (continued)**

| eHRPWM1<br>BYTE ADDRESS                                | eHRPWM2<br>BYTE ADDRESS | eHRPWM3<br>BYTE ADDRESS | REGISTER<br>NAME | SIZE<br>(×16) | SHADOW | REGISTER DESCRIPTION                                      |
|--|-------------------------|-------------------------|------------------|---------------|--------|---|
| <b>Action-Qualifier Submodule Registers</b>            |                         |                         |                  |               |        |   |
| 0x01F0 0016  | 0x01F0 2016             | 0x01F0 4016             | AQCTLA           | 1             | No     | Action-Qualifier Control Register for Output A (eHRPWMxA) |
| 0x01F0 0018  | 0x01F0 2018             | 0x01F0 4018             | AQCTLB           | 1             | No     | Action-Qualifier Control Register for Output B (eHRPWMxB) |
| 0x01F0 001A  | 0x01F0 201A             | 0x01F0 401A             | AQSFR            | 1             | No     | Action-Qualifier Software Force Register                  |
| 0x01F0 001C  | 0x01F0 201C             | 0x01F0 401C             | AQCSFR           | 1             | Yes    | Action-Qualifier Continuous S/W Force Register Set        |
| <b>Dead-Band Generator Submodule Registers</b>         |                         |                         |                  |               |        |   |
| 0x01F0 001E  | 0x01F0 201E             | 0x01F0 401E             | DBCTL            | 1             | No     | Dead-Band Generator Control Register                      |
| 0x01F0 0020  | 0x01F0 2020             | 0x01F0 4020             | DBRED            | 1             | No     | Dead-Band Generator Rising Edge Delay Count Register      |
| 0x01F0 0022  | 0x01F0 2022             | 0x01F0 4022             | DBFED            | 1             | No     | Dead-Band Generator Falling Edge Delay Count Register     |
| <b>PWM-Chopper Submodule Registers</b>                 |                         |                         |                  |               |        |   |
| 0x01F0 003C  | 0x01F0 203C             | 0x01F0 403C             | PCCTL            | 1             | No     | PWM-Chopper Control Register                              |
| <b>Trip-Zone Submodule Registers</b>                   |                         |                         |                  |               |        |   |
| 0x01F0 0024  | 0x01F0 2024             | 0x01F0 4024             | TZSEL            | 1             | No     | Trip-Zone Select Register                                 |
| 0x01F0 0028  | 0x01F0 2028             | 0x01F0 4028             | TZCTL            | 1             | No     | Trip-Zone Control Register                                |
| 0x01F0 002A  | 0x01F0 202A             | 0x01F0 402A             | TZEINT           | 1             | No     | Trip-Zone Enable Interrupt Register                       |
| 0x01F0 002C  | 0x01F0 202C             | 0x01F0 402C             | TZFLG            | 1             | No     | Trip-Zone Flag Register                                   |
| 0x01F0 002E  | 0x01F0 202E             | 0x01F0 402E             | TZCLR            | 1             | No     | Trip-Zone Clear Register                                  |
| 0x01F0 0030  | 0x01F0 2030             | 0x01F0 4030             | TZFRC            | 1             | No     | Trip-Zone Force Register                                  |
| <b>Event-Trigger Submodule Registers</b>               |                         |                         |                  |               |        |   |
| 0x01F0 0032  | 0x01F0 2032             | 0x01F0 4032             | ETSEL            | 1             | No     | Event-Trigger Selection Register                          |
| 0x01F0 0034  | 0x01F0 2034             | 0x01F0 4034             | ETPS             | 1             | No     | Event-Trigger Pre-Scale Register                          |
| 0x01F0 0036  | 0x01F0 2036             | 0x01F0 4036             | ETFLG            | 1             | No     | Event-Trigger Flag Register                               |
| 0x01F0 0038  | 0x01F0 2038             | 0x01F0 4038             | ETCLR            | 1             | No     | Event-Trigger Clear Register                              |
| 0x01F0 003A  | 0x01F0 203A             | 0x01F0 403A             | ETFRC            | 1             | No     | Event-Trigger Force Register                              |
| <b>High-Resolution PWM (HRPWM) Submodule Registers</b> |                         |                         |                  |               |        |   |
| 0x01F0 1040  | 0x01F0 3040             | 0x01F0 5040             | HRCNFG           | 1             | No     | HRPWM Configuration Register <sup>(1)</sup>               |

### 6.20.1 Enhanced Pulse Width Modulator (eHRPWM) Timing

PWM refers to PWM outputs on eHRPWM1-6. Table 6-64 shows the PWM timing requirements and Table 6-65, switching characteristics.

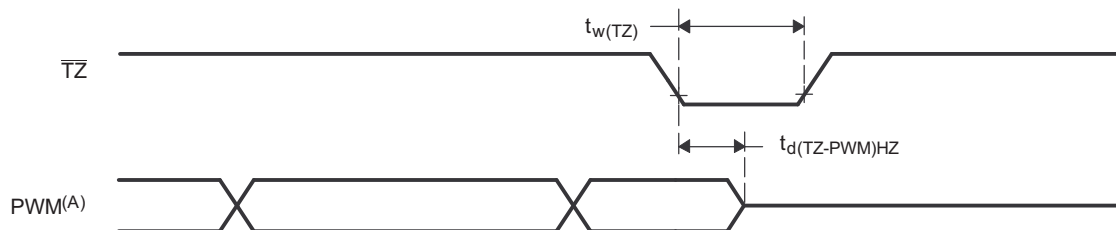
**Table 6-64. eHRPWM Timing Requirements**

|                       |                        | TEST CONDITIONS | MIN                  | MAX | UNIT   |
|-----------------------|------------------------|-----------------|----------------------|-----|--------|
| $t_{w(\text{SYNIN})}$ | Sync input pulse width | Asynchronous    | $2t_{c(\text{SCO})}$ |     | cycles |
|                       |                        | Synchronous     | $2t_{c(\text{SCO})}$ |     | cycles |

**Table 6-65. eHRPWM Switching Characteristics**

| PARAMETER                       | TEST CONDITIONS                                  | MIN   | MAX | UNIT   |
|---------------------------------|--|---|-----|--------|
| $t_{w(\text{PWM})}$             | Pulse duration, PWMx output high/low             | 20  |     | ns     |
| $t_{w(\text{SYNCOUT})}$         | Sync output pulse width                          | $8t_{c(\text{SCO})}$                          |     | cycles |
| $t_{d(\text{PWM})\text{tza}}$   | Delay time, trip input active to PWM forced high | no pin load; no additional programmable delay | 25  | ns     |
|                                 | Delay time, trip input active to PWM forced low  |   |     |        |
| $t_{d(\text{TZ-PWM})\text{HZ}}$ | Delay time, trip input active to PWM Hi-Z        |   | 20  | ns     |

### 6.20.2 Trip-Zone Input Timing



- A. PWM refers to all the PWM pins in the device. The state of the PWM pins after  $\overline{\text{TZ}}$  is taken high depends on the PWM recovery software.

**Figure 6-41. PWM Hi-Z Characteristics**

**Table 6-66. Trip-Zone input Timing Requirements**

|                    |  | TEST CONDITIONS | MIN                  | MAX | UNIT   |
|--------------------|--|-----------------|----------------------|-----|--------|
| $t_{w(\text{TZ})}$ | Pulse duration, $\overline{\text{TZ}}$ input low | Asynchronous    | $1t_{c(\text{SCO})}$ |     | cycles |
|                    |  | Synchronous     | $2t_{c(\text{SCO})}$ |     | cycles |

Table 6-67 shows the high-resolution PWM switching characteristics.

**Table 6-67. High Resolution PWM Characteristics at SYSCLKOUT = (60 - 100 MHz)**

| PARAMETER   | MIN | TYP | MAX | UNIT |
|---|-----|-----|-----|------|
| Micro Edge Positioning (MEP) step size <sup>(1)</sup> |     | 200 |     | ps   |

- (1) MEP step size will increase with low voltage and high temperature, and decrease with high voltage and cold temperature.

## 6.21 Timers

The timers support the following features:

- Configurable as single 64-bit timer or two 32-bit timers
- Period timeouts generate interrupts, DMA events or external pin events
- 8 32-bit compare registers
- Compare matches generate interrupt events
- Capture capability
- 64-bit Watchdog capability (Timer64P1 only)

[Table 6-68](#) lists the timer registers. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

**Table 6-68. Timer Registers**

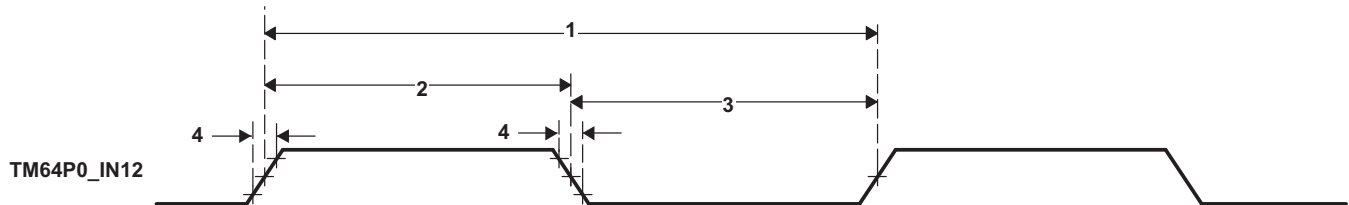
| TIMER64P 0  | TIMER64P 1  | REGISTER NAME | REGISTER DESCRIPTION                        |
|-------------|-------------|---------------|---|
| 0x01C2 0000 | 0x01C2 1000 | REV           | Revision Register                           |
| 0x01C2 0004 | 0x01C2 1004 | EMUMGT        | Emulation Management Register               |
| 0x01C2 0008 | 0x01C2 1008 | GPINTGPEN     | GPIO Interrupt and GPIO Enable Register     |
| 0x01C2 000C | 0x01C2 100C | GPDATGPDIR    | GPIO Data and GPIO Direction Register       |
| 0x01C2 0010 | 0x01C2 1010 | TIM12         | Timer Counter Register 12                   |
| 0x01C2 0014 | 0x01C2 1014 | TIM34         | Timer Counter Register 34                   |
| 0x01C2 0018 | 0x01C2 1018 | PRD12         | Timer Period Register 12                    |
| 0x01C2 001C | 0x01C2 101C | PRD34         | Timer Period Register 34                    |
| 0x01C2 0020 | 0x01C2 1020 | TCR           | Timer Control Register                      |
| 0x01C2 0024 | 0x01C2 1024 | TGCR          | Timer Global Control Register               |
| 0x01C2 0028 | 0x01C2 1028 | WDTCR         | Watchdog Timer Control Register             |
| 0x01C2 0034 | 0x01C2 1034 | REL12         | Timer Reload Register 12                    |
| 0x01C2 0038 | 0x01C2 1038 | REL34         | Timer Reload Register 34                    |
| 0x01C2 003C | 0x01C2 103C | CAP12         | Timer Capture Register 12                   |
| 0x01C2 0040 | 0x01C2 1040 | CAP34         | Timer Capture Register 34                   |
| 0x01C2 0044 | 0x01C2 1044 | INTCTLSTAT    | Timer Interrupt Control and Status Register |
| 0x01C2 0060 | 0x01C2 1060 | CMP0          | Compare Register 0                          |
| 0x01C2 0064 | 0x01C2 1064 | CMP1          | Compare Register 1                          |
| 0x01C2 0068 | 0x01C2 1068 | CMP2          | Compare Register 2                          |
| 0x01C2 006C | 0x01C2 106C | CMP3          | Compare Register 3                          |
| 0x01C2 0070 | 0x01C2 1070 | CMP4          | Compare Register 4                          |
| 0x01C2 0074 | 0x01C2 1074 | CMP5          | Compare Register 5                          |
| 0x01C2 0078 | 0x01C2 1078 | CMP6          | Compare Register 6                          |
| 0x01C2 007C | 0x01C2 107C | CMP7          | Compare Register 7                          |

### 6.21.1 Timer Electrical Data/Timing

**Table 6-69. Timing Requirements for Timer Input<sup>(1)(2)</sup> (see Figure 6-42)**

| NO. |  | MIN   | MAX                        | UNIT |
|-----|--|-------|----------------------------|------|
| 1   | $t_{c(TM64Px\_IN12)}$ Cycle time, TM64Px_IN12    | 4P    |                            | ns   |
| 2   | $t_{w(TINPH)}$ Pulse duration, TM64Px_IN12 high  | 0.45C | 0.55C                      | ns   |
| 3   | $t_{w(TINPL)}$ Pulse duration, TM64Px_IN12 low   | 0.45C | 0.55C                      | ns   |
| 4   | $t_t(TM64Px\_IN12)$ Transition time, TM64Px_IN12 |       | 0.25P or 10 <sup>(3)</sup> | ns   |

- (1) P = OSCIN cycle time in ns.
- (2) C = TM64P0\_IN12 cycle time in ns.
- (3) Whichever is smaller. P = The period of the applied signal. Maintaining transition times as fast as possible is recommended to improve noise immunity on input signals.

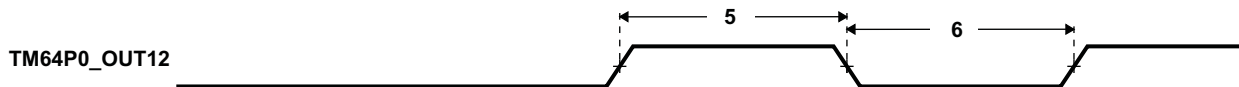


**Figure 6-42. Timer Timing**

**Table 6-70. Switching Characteristics Over Recommended Operating Conditions for Timer Output<sup>(1)</sup>**

| NO. | PARAMETER  | MIN | MAX | UNIT |
|-----|--|-----|-----|------|
| 5   | $t_{w(TOUTH)}$ Pulse duration, TM64P0_OUT12 high | 4P  |     | ns   |
| 6   | $t_{w(TOURL)}$ Pulse duration, TM64P0_OUT12 low  | 4P  |     | ns   |

- (1) P = OSCIN cycle time in ns.



**Figure 6-43. Timer Timing**

## 6.22 Inter-Integrated Circuit Serial Ports (I2C0, I2C1)

### 6.22.1 I2C Device-Specific Information

Having two I2C modules on the device simplifies system architecture, since one module may be used by the DSP to control local peripherals ICs (DACs, ADCs, etc.) while the other may be used to communicate with other controllers in a system or to implement a user interface. Figure 6-44 is the block diagram of the I2C Module. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

Each I2C port supports:

- Compatible with Philips® I2C Specification Revision 2.1 (January 2000)
- Fast Mode up to 400 Kbps (no fail-safe I/O buffers)
- Noise Filter to Remove Noise 50 ns or less
- Seven- and Ten-Bit Device Addressing Modes
- Master (Transmit/Receive) and Slave (Transmit/Receive) Functionality
- Events: DMA, Interrupt, or Polling
- General-Purpose I/O Capability if not used as I2C

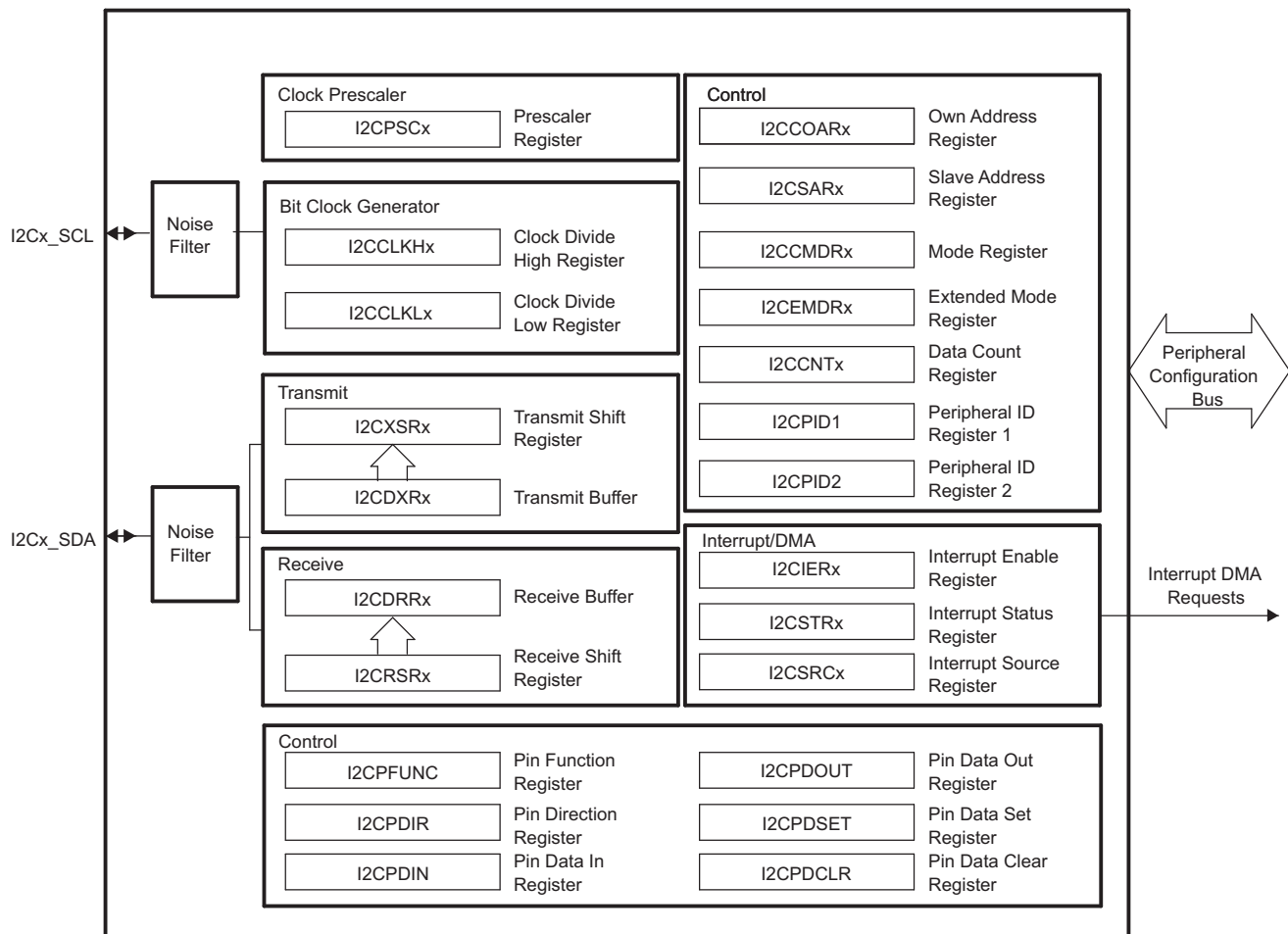


Figure 6-44. I2C Module Block Diagram

## 6.22.2 I2C Peripheral Registers Description(s)

Table 6-71 is the list of the I2C registers.

**Table 6-71. Inter-Integrated Circuit (I2C) Registers**

| I2C0<br>BYTE ADDRESS | I2C1<br>BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                   |
|----------------------|----------------------|---------------|--|
| 0x01C2 2000          | 0x01E2 8000          | ICOAR         | I2C Own Address Register               |
| 0x01C2 2004          | 0x01E2 8004          | ICIMR         | I2C Interrupt Mask Register            |
| 0x01C2 2008          | 0x01E2 8008          | ICSTR         | I2C Interrupt Status Register          |
| 0x01C2 200C          | 0x01E2 800C          | ICCLKL        | I2C Clock Low-Time Divider Register    |
| 0x01C2 2010          | 0x01E2 8010          | ICCLKH        | I2C Clock High-Time Divider Register   |
| 0x01C2 2014          | 0x01E2 8014          | ICCNT         | I2C Data Count Register                |
| 0x01C2 2018          | 0x01E2 8018          | ICDRR         | I2C Data Receive Register              |
| 0x01C2 201C          | 0x01E2 801C          | ICSAR         | I2C Slave Address Register             |
| 0x01C2 2020          | 0x01E2 8020          | ICDXR         | I2C Data Transmit Register             |
| 0x01C2 2024          | 0x01E2 8024          | ICMDR         | I2C Mode Register                      |
| 0x01C2 2028          | 0x01E2 8028          | ICIVR         | I2C Interrupt Vector Register          |
| 0x01C2 202C          | 0x01E2 802C          | ICEMDR        | I2C Extended Mode Register             |
| 0x01C2 2030          | 0x01E2 8030          | ICPSC         | I2C Prescaler Register                 |
| 0x01C2 2034          | 0x01E2 8034          | REVID1        | I2C Revision Identification Register 1 |
| 0x01C2 2038          | 0x01E2 8038          | REVID2        | I2C Revision Identification Register 2 |
| 0x01C2 2048          | 0x01E2 8048          | ICPFUNC       | I2C Pin Function Register              |
| 0x01C2 204C          | 0x01E2 804C          | ICPDIR        | I2C Pin Direction Register             |
| 0x01C2 2050          | 0x01E2 8050          | ICPDIN        | I2C Pin Data In Register               |
| 0x01C2 2054          | 0x01E2 8054          | ICPDOUT       | I2C Pin Data Out Register              |
| 0x01C2 2058          | 0x01E2 8058          | ICPDSET       | I2C Pin Data Set Register              |
| 0x01C2 205C          | 0x01E2 805C          | ICPDCLR       | I2C Pin Data Clear Register            |

## 6.22.3 I2C Electrical Data/Timing

### 6.22.3.1 Inter-Integrated Circuit (I2C) Timing

Table 6-72 and Table 6-73 assume testing over recommended operating conditions (see Figure 6-45 and Figure 6-46).

**Table 6-72. I2C Input Timing Requirements**

| NO. |                     |   | MIN           | MAX | UNIT    |
|-----|---------------------|---|---------------|-----|---------|
| 1   | $t_{c(SCL)}$        | Cycle time, I2Cx_SCL                          | Standard Mode | 10  | $\mu$ s |
|     |                     |   | Fast Mode     | 2.5 |         |
| 2   | $t_{su(SCLH-SDAL)}$ | Setup time, I2Cx_SCL high before I2Cx_SDA low | Standard Mode | 4.7 | $\mu$ s |
|     |                     |   | Fast Mode     | 0.6 |         |
| 3   | $t_{h(SCLL-SDAL)}$  | Hold time, I2Cx_SCL low after I2Cx_SDA low    | Standard Mode | 4   | $\mu$ s |
|     |                     |   | Fast Mode     | 0.6 |         |
| 4   | $t_{w(SCLL)}$       | Pulse duration, I2Cx_SCL low                  | Standard Mode | 4.7 | $\mu$ s |
|     |                     |   | Fast Mode     | 1.3 |         |
| 5   | $t_{w(SCLH)}$       | Pulse duration, I2Cx_SCL high                 | Standard Mode | 4   | $\mu$ s |
|     |                     |   | Fast Mode     | 0.6 |         |
| 6   | $t_{su(SDA-SCLH)}$  | Setup time, I2Cx_SDA before I2Cx_SCL high     | Standard Mode | 250 | ns      |
|     |                     |   | Fast Mode     | 100 |         |
| 7   | $t_{h(SDA-SCLL)}$   | Hold time, I2Cx_SDA after I2Cx_SCL low        | Standard Mode | 0   | $\mu$ s |
|     |                     |   | Fast Mode     | 0   |         |

**Table 6-72. I2C Input Timing Requirements (continued)**

| NO. |                     |  | MIN           | MAX           | UNIT          |
|-----|---------------------|--|---------------|---------------|---------------|
| 8   | $t_{w(SDAH)}$       | Pulse duration, I2Cx_SDA high                  | Standard Mode | 4.7           | $\mu\text{s}$ |
|     |                     |  | Fast Mode     | 1.3           |               |
| 9   | $t_{r(SDA)}$        | Rise time, I2Cx_SDA                            | Standard Mode | 1000          | ns            |
|     |                     |  | Fast Mode     | $20 + 0.1C_b$ |               |
| 10  | $t_{r(SCL)}$        | Rise time, I2Cx_SCL                            | Standard Mode | 1000          | ns            |
|     |                     |  | Fast Mode     | $20 + 0.1C_b$ |               |
| 11  | $t_{f(SDA)}$        | Fall time, I2Cx_SDA                            | Standard Mode | 300           | ns            |
|     |                     |  | Fast Mode     | $20 + 0.1C_b$ |               |
| 12  | $t_{f(SCL)}$        | Fall time, I2Cx_SCL                            | Standard Mode | 300           | ns            |
|     |                     |  | Fast Mode     | $20 + 0.1C_b$ |               |
| 13  | $t_{su(SCLH-SDAH)}$ | Setup time, I2Cx_SCL high before I2Cx_SDA high | Standard Mode | 4             | $\mu\text{s}$ |
|     |                     |  | Fast Mode     | 0.6           |               |
| 14  | $t_{w(SP)}$         | Pulse duration, spike (must be suppressed)     | Standard Mode | N/A           | ns            |
|     |                     |  | Fast Mode     | 0             |               |
| 15  | $C_b$               | Capacitive load for each bus line              | Standard Mode | 400           | pF            |
|     |                     |  | Fast Mode     | 400           |               |

**Table 6-73. I2C Switching Characteristics<sup>(1)</sup>**

| NO. | PARAMETER           |   | MIN           | MAX | UNIT          |
|-----|---------------------|---|---------------|-----|---------------|
| 16  | $t_c(SCL)$          | Cycle time, I2Cx_SCL                            | Standard Mode | 10  | $\mu\text{s}$ |
|     |                     |   | Fast Mode     | 2.5 |               |
| 17  | $t_{su(SCLH-SDAL)}$ | Setup time, I2Cx_SCL high before I2Cx_SDA low   | Standard Mode | 4.7 | $\mu\text{s}$ |
|     |                     |   | Fast Mode     | 0.6 |               |
| 18  | $t_h(SDAL-SCLL)$    | Hold time, I2Cx_SCL low after I2Cx_SDA low      | Standard Mode | 4   | $\mu\text{s}$ |
|     |                     |   | Fast Mode     | 0.6 |               |
| 19  | $t_w(SCLL)$         | Pulse duration, I2Cx_SCL low                    | Standard Mode | 4.7 | $\mu\text{s}$ |
|     |                     |   | Fast Mode     | 1.3 |               |
| 20  | $t_w(SCLH)$         | Pulse duration, I2Cx_SCL high                   | Standard Mode | 4   | $\mu\text{s}$ |
|     |                     |   | Fast Mode     | 0.6 |               |
| 21  | $t_{su(SDAV-SCLH)}$ | Setup time, I2Cx_SDA valid before I2Cx_SCL high | Standard Mode | 250 | ns            |
|     |                     |   | Fast Mode     | 100 |               |
| 22  | $t_h(SCLL-SDAV)$    | Hold time, I2Cx_SDA valid after I2Cx_SCL low    | Standard Mode | 0   | $\mu\text{s}$ |
|     |                     |   | Fast Mode     | 0   |               |
| 23  | $t_w(SDAH)$         | Pulse duration, I2Cx_SDA high                   | Standard Mode | 4.7 | $\mu\text{s}$ |
|     |                     |   | Fast Mode     | 1.3 |               |
| 28  | $t_{su(SCLH-SDAH)}$ | Setup time, I2Cx_SCL high before I2Cx_SDA high  | Standard Mode | 4   | $\mu\text{s}$ |
|     |                     |   | Fast Mode     | 0.6 |               |

(1) I2C must be configured correctly to meet the timings in [Table 6-73](#).

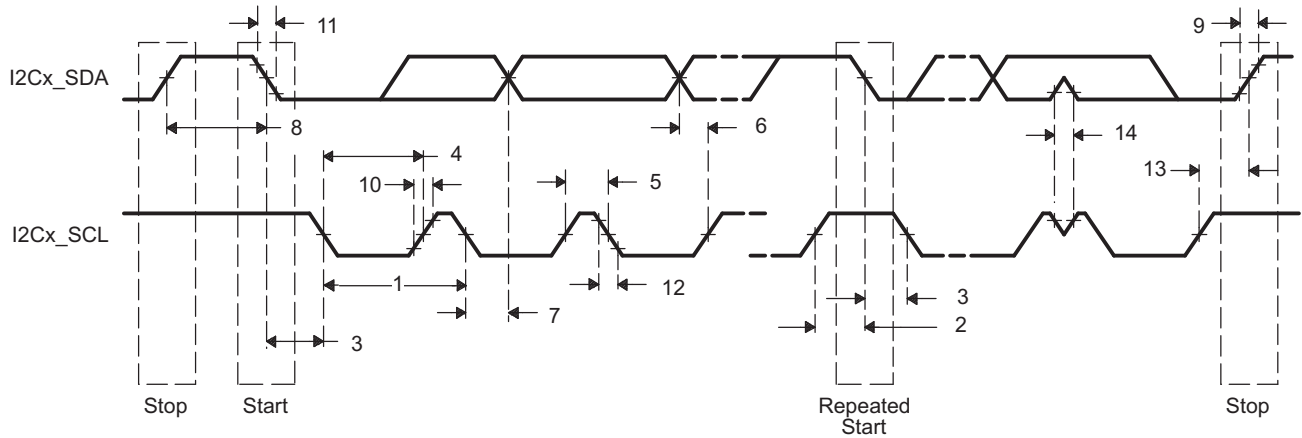


Figure 6-45. I2C Receive Timings

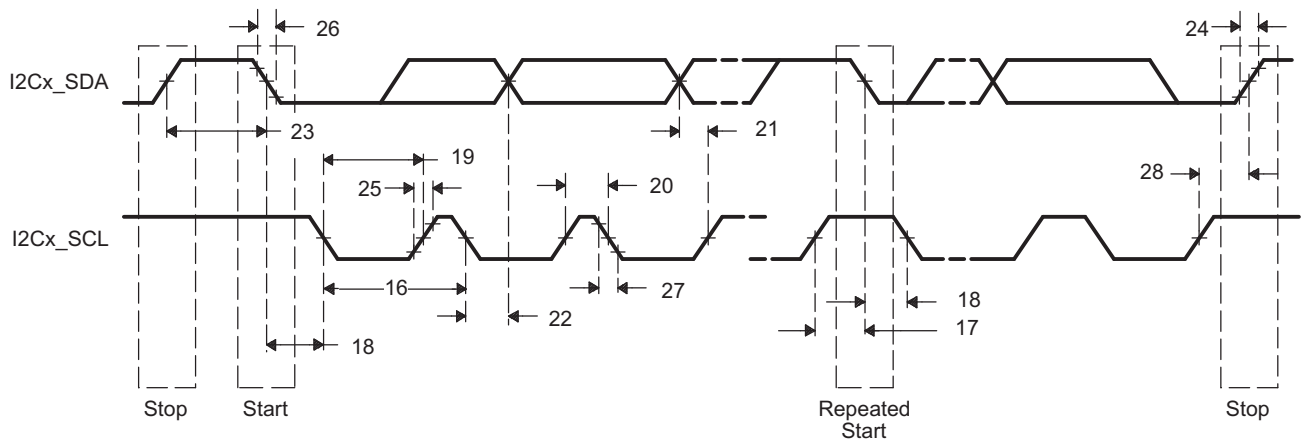


Figure 6-46. I2C Transmit Timings

## 6.23 Universal Asynchronous Receiver/Transmitter (UART)

The device has 2 UART peripherals. Each UART has the following features:

- 16-byte storage space for both the transmitter and receiver FIFOs
- Autoflow control signals (CTS, RTS) on UART0 only.
- 1, 4, 8, or 14 byte selectable receiver FIFO trigger level for autoflow control and DMA
- DMA signaling capability for both received and transmitted data
- Programmable auto-rts and auto-cts for autoflow control
- Programmable Baud Rate up to 3Mbaud
- Programmable Oversampling Options of x13 and x16
- Frequency pre-scale values from 1 to 65,535 to generate appropriate baud rates
- Prioritized interrupts
- Programmable serial data formats
  - 5, 6, 7, or 8-bit characters
  - Even, odd, or no parity bit generation and detection
  - 1, 1.5, or 2 stop bit generation
- False start bit detection
- Line break generation and detection
- Internal diagnostic capabilities
  - Loopback controls for communications link fault isolation
  - Break, parity, overrun, and framing error simulation

The UART registers are listed in [Section 6.23.1](#). See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

### 6.23.1 UART Peripheral Registers Description(s)

[Table 6-74](#) is the list of UART registers.

**Table 6-74. UART Registers**

| UART0<br>BYTE ADDRESS | UART2<br>BYTE ADDRESS | REGISTER NAME | REGISTER DESCRIPTION                          |
|-----------------------|-----------------------|---------------|---|
| 0x01C4 2000           | 0x01D0 D000           | RBR           | Receiver Buffer Register (read only)          |
| 0x01C4 2000           | 0x01D0 D000           | THR           | Transmitter Holding Register (write only)     |
| 0x01C4 2004           | 0x01D0 D004           | IER           | Interrupt Enable Register                     |
| 0x01C4 2008           | 0x01D0 D008           | IIR           | Interrupt Identification Register (read only) |
| 0x01C4 2008           | 0x01D0 D008           | FCR           | FIFO Control Register (write only)            |
| 0x01C4 200C           | 0x01D0 D00C           | LCR           | Line Control Register                         |
| 0x01C4 2010           | 0x01D0 D010           | MCR           | Modem Control Register                        |
| 0x01C4 2014           | 0x01D0 D014           | LSR           | Line Status Register                          |
| 0x01C4 2020           | 0x01D0 D020           | DLL           | Divisor LSB Latch                             |
| 0x01C4 2024           | 0x01D0 D024           | DLH           | Divisor MSB Latch                             |
| 0x01C4 2028           | 0x01D0 D028           | REVID1        | Revision Identification Register 1            |
| 0x01C4 2030           | 0x01D0 D030           | PWREMU_MGMT   | Power and Emulation Management Register       |
| 0x01C4 2034           | 0x01D0 D034           | MDR           | Mode Definition Register                      |

### 6.23.2 UART Electrical Data/Timing

**Table 6-75. Timing Requirements for UARTx Receive<sup>(1)</sup> (see Figure 6-47)**

| NO. |                     |   | MIN   | MAX   | UNIT  |
|-----|---------------------|---|-------|-------|-------|
| 4   | $t_w(\text{URXDB})$ | Pulse duration, receive data bit (RXDn) | 0.96U | 1.05U | MBaud |
| 5   | $t_w(\text{URXSB})$ | Pulse duration, receive start bit       | 0.96U | 1.05U | ns    |

(1) U = UART baud time = 1/programmed baud rate.

**Table 6-76. Switching Characteristics Over Recommended Operating Conditions for UARTx Transmit<sup>(1)</sup> (see Figure 6-47)**

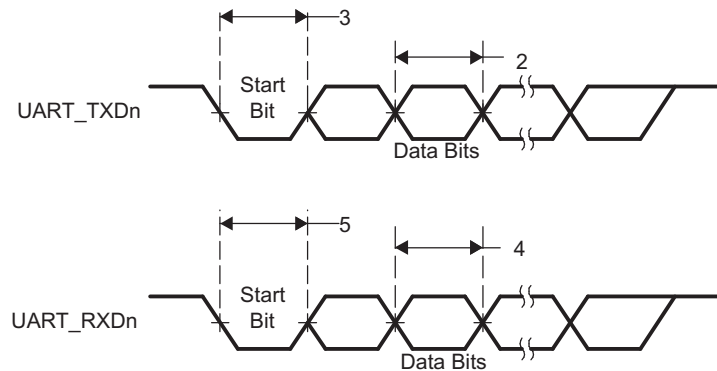
| NO. | PARAMETER           | MIN   | MAX                   | UNIT  |
|-----|---------------------|-------|-----------------------|-------|
| 1   | $f_{(\text{baud})}$ |       | D/E <sup>(3)(4)</sup> | MBaud |
| 2   | $t_w(\text{UTXDB})$ | U - 2 | U + 2                 | ns    |
| 3   | $t_w(\text{UTXSB})$ | U - 2 | U + 2                 | ns    |

(1) U = UART baud time = 1/programmed baud rate.

(2) Baud rate is not indicative of data rate. Actual data rate will be limited by system factors such as EDMA loading, EMIF loading, system frequency, etc.

(3) D = UART input clock in MHz. The UART(s) input clock source is PLL0\_SYSCLK2.

(4) E = UART divisor x UART sampling rate. The UART divisor is set through the UART divisor latch registers (DLL and DLH). The UART sampling rate is set through the over-sampling mode select bit (OSM\_SEL) of the UART mode definition register (MDR).



**Figure 6-47. UART Transmit/Receive Timing**

## 6.24 Power and Sleep Controller (PSC)

The Power and Sleep Controllers (PSC) are responsible for managing transitions of system power on/off, clock on/off, resets (device level and module level). It is used primarily to provide granular power control for on chip modules (peripherals and CPU). A PSC module consists of a Global PSC (GPSC) and a set of Local PSCs (LPSCs). The GPSC contains memory mapped registers, PSC interrupts, a state machine for each peripheral/module it controls. An LPSC is associated with every module that is controlled by the PSC and provides clock and reset control. See the *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide (SPRUFK9)* for more details.

The PSC includes the following features:

- Provides a software interface to:
  - Control module clock enable/disable
  - Control module reset
  - Control CPU local reset
- Supports IcePick emulation features: power, clock and reset

### 6.24.1 PSC Peripheral Registers Description(s)

**Table 6-77. Power and Sleep Controller (PSC) Registers**

| PSC0<br>BYTE ADDRESS      | PSC1<br>BYTE ADDRESS      | REGISTER NAME    | REGISTER DESCRIPTION   |
|---------------------------|---------------------------|------------------|--|
| 0x01C1 0000               | 0x01E2 7000               | REVID            | Peripheral Revision and Class Information Register   |
| 0x01C1 0018               | 0x01E2 7018               | INTEVAL          | Interrupt Evaluation Register  |
| 0x01C1 0040               | 0x01E2 7040               | MERRPR0          | Module Error Pending Register 0 (module 0-15) (PSC0)<br>Module Error Pending Register 0 (module 0-31) (PSC1) |
| 0x01C1 0050               | 0x01E2 7050               | MERRCR0          | Module Error Clear Register 0 (module 0-15) (PSC0)<br>Module Error Clear Register 0 (module 0-31) (PSC1)     |
| 0x01C1 0060               | 0x01E2 7060               | PERRPR           | Power Error Pending Register   |
| 0x01C1 0068               | 0x01E2 7068               | PERRCR           | Power Error Clear Register   |
| 0x01C1 0120               | 0x01E2 7120               | PTCMD            | Power Domain Transition Command Register   |
| 0x01C1 0128               | 0x01E2 7128               | PTSTAT           | Power Domain Transition Status Register  |
| 0x01C1 0200               | 0x01E2 7200               | PDSTAT0          | Power Domain 0 Status Register   |
| 0x01C1 0204               | 0x01E2 7204               | PDSTAT1          | Power Domain 1 Status Register   |
| 0x01C1 0300               | 0x01E2 7300               | PDCTL0           | Power Domain 0 Control Register  |
| 0x01C1 0304               | 0x01E2 7304               | PDCTL1           | Power Domain 1 Control Register  |
| 0x01C1 0400               | 0x01E2 7400               | PDCFG0           | Power Domain 0 Configuration Register  |
| 0x01C1 0404               | 0x01E2 7404               | PDCFG1           | Power Domain 1 Configuration Register  |
| 0x01C1 0800 - 0x01C1 083C | 0x01E2 7800 - 0x01E2 787C | MDSTAT0-MDSTAT15 | Module Status <i>n</i> Register (modules 0-15) (PSC0)  |
|                           |                           | MDSTAT0-MDSTAT31 | Module Status <i>n</i> Register (modules 0-31) (PSC1)  |
| 0x01C1 0A00 - 0x01C1 0A3C | 0x01E2 7A00 - 0x01E2 7A7C | MDCTL0-MDCTL15   | Module Control <i>n</i> Register (modules 0-15) (PSC0)   |
|                           |                           | MDCTL0-MDCTL31   | Module Control <i>n</i> Register (modules 0-31) (PSC1)   |

## 6.24.2 Power Domain and Module Topology

The device includes two PSC modules.

Each PSC module controls clock states for several on the on chip modules, controllers and interconnect components. [Table 6-78](#) and [Table 6-79](#) lists the set of peripherals/modules that are controlled by the PSC, the power domain they are associated with, the LPSC assignment and the default (power-on reset) module states. The module states and terminology are defined in [Section 6.24.2.2](#).

**Table 6-78. PSC0 Default Module Configuration**

| LPSC NUMBER | MODULE NAME                      | POWER DOMAIN   | DEFAULT MODULE STATE | AUTO SLEEP/WAKE ONLY |
|-------------|----------------------------------|----------------|----------------------|----------------------|
| 0           | EDMA3 Channel Controller         | AlwaysON (PD0) | SwRstDisable         | —                    |
| 1           | EDMA3 Transfer Controller 0      | AlwaysON (PD0) | SwRstDisable         | —                    |
| 2           | EDMA3 Transfer Controller 1      | AlwaysON (PD0) | SwRstDisable         | —                    |
| 3           | EMIFA (BR7)                      | AlwaysON (PD0) | SwRstDisable         | —                    |
| 4           | SPI 0                            | AlwaysON (PD0) | SwRstDisable         | —                    |
| 5           | MMC/SD 0                         | AlwaysON (PD0) | SwRstDisable         | —                    |
| 8           | —                                | —              | —                    | —                    |
| 9           | UART 0                           | AlwaysON (PD0) | SwRstDisable         | —                    |
| 10          | SCR0<br>(Br 0, Br 1, Br 2, Br 8) | AlwaysON (PD0) | Enable               | Yes                  |
| 11          | SCR1<br>(Br 4)                   | AlwaysON (PD0) | Enable               | Yes                  |
| 12          | SCR2<br>(Br 3, Br 5, Br 6)       | AlwaysON (PD0) | Enable               | Yes                  |
| 13          | PRUSS                            | AlwaysON (PD0) | SwRstDisable-        | —                    |
| 15          | DSP                              | PD_DSP (PD1)   | Enable               | —                    |

**Table 6-79. PSC1 Default Module Configuration**

| LPSC NUMBER | MODULE NAME             | POWER DOMAIN   | DEFAULT MODULE STATE | AUTO SLEEP/WAKE ONLY |
|-------------|-------------------------|----------------|----------------------|----------------------|
| 0-2         | —                       | —              | —                    | —                    |
| 3           | GPIO                    | AlwaysON (PD0) | SwRstDisable         | —                    |
| 4           | —                       | —              | —                    | —                    |
| 5           | EMAC                    | AlwaysON (PD0) | SwRstDisable         | —                    |
| 6           | EMIFB (Br 20)           | AlwaysON (PD0) | SwRstDisable         | —                    |
| 7           | McASP0 ( + McASP0 FIFO) | AlwaysON (PD0) | SwRstDisable         | —                    |
| 8           | McASP1 ( + McASP1 FIFO) | AlwaysON (PD0) | SwRstDisable         | —                    |
| 9-10        | —                       | —              | —                    | —                    |
| 11          | I2C 1                   | AlwaysON (PD0) | SwRstDisable         | —                    |
| 12          | —                       | —              | —                    | —                    |
| 13          | UART 2                  | AlwaysON (PD0) | SwRstDisable         | —                    |
| 14-16       | —                       | —              | —                    | —                    |
| 17          | eHRPWM0/1/2             | AlwaysON (PD0) | SwRstDisable         | —                    |
| 18-19       | —                       | —              | —                    | —                    |
| 20          | ECAP0/1/2               | AlwaysON (PD0) | SwRstDisable         | —                    |
| 21          | EQEP0/1                 | AlwaysON (PD0) | SwRstDisable         | —                    |
| 22-23       | —                       | —              | —                    | —                    |
| 24          | SCR8<br>(Br 15)         | AlwaysON (PD0) | Enable               | Yes                  |

**Table 6-79. PSC1 Default Module Configuration (continued)**

| LPSC NUMBER | MODULE NAME      | POWER DOMAIN   | DEFAULT MODULE STATE | AUTO SLEEP/WAKE ONLY |
|-------------|------------------|----------------|----------------------|----------------------|
| 25          | SCR7<br>(Br 12)  | AlwaysON (PD0) | Enable               | Yes                  |
| 26          | SCR12<br>(Br 18) | AlwaysON (PD0) | Enable               | Yes                  |
| 27-31       | —                | —              | —                    | —                    |

### 6.24.2.1 Power Domain States

A power domain can only be in one of the two states: ON or OFF, defined as follows:

- ON: power to the domain is on
- OFF: power to the domain is off

In the device, for both PSC0 and PSC1, the Always ON domain, or PD0 power domain, is always in the ON state when the chip is powered-on. This domain is not programmable to OFF state.

- On PSC0 PD1/PD\_DSP Domain: Controls the sleep state for DSP L1 and L2 Memories

### 6.24.2.2 Module States

The PSC defines several possible states for a module. These states are essentially a combination of the module reset asserted or de-asserted and module clock on/enabled or off/disabled. The module states are defined in [Table 6-80](#).

**Table 6-80. Module States**

| MODULE STATE | MODULE RESET | MODULE CLOCK | MODULE STATE DEFINITION   |
|--------------|--------------|--------------|---|
| Enable       | De-asserted  | On           | A module in the enable state has its module reset de-asserted and it has its clock on. This is the normal operational state for a given module  |
| Disable      | De-asserted  | Off          | A module in the disabled state has its module reset de-asserted and it has its module clock off. This state is typically used for disabling a module clock to save power. The device is designed in full static CMOS, so when you stop a module clock, it retains the module's state. When the clock is restarted, the module resumes operating from the stopping point.  |
| SyncReset    | Asserted     | On           | A module state in the SyncReset state has its module reset asserted and it has its clock on. Generally, software is not expected to initiate this state   |
| SwRstDisable | Asserted     | Off          | A module in the SwResetDisable state has its module reset asserted and it has its clock disabled. After initial power-on, several modules come up in the SwRstDisable state. Generally, software is not expected to initiate this state   |
| Auto Sleep   | De-asserted  | Off          | A module in the Auto Sleep state also has its module reset de-asserted and its module clock disabled, similar to the Disable state. However this is a special state, once a module is configured in this state by software, it can "automatically" transition to "Enable" state whenever there is an internal read/write request made to it, and after servicing the request it will "automatically" transition into the sleep state (with module reset re de-asserted and module clock disabled), without any software intervention. The transition from sleep to enabled and back to sleep state has some cycle latency associated with it. It is not envisioned to use this mode when peripherals are fully operational and moving data. |
| Auto Wake    | De-asserted  | Off          | A module in the Auto Wake state also has its module reset de-asserted and its module clock disabled, similar to the Disable state. However this is a special state, once a module is configured in this state by software, it will "automatically" transition to "Enable" state whenever there is an internal read/write request made to it, and will remain in the "Enabled" state from then on (with module reset re de-asserted and module clock on), without any software intervention. The transition from sleep to enabled state has some cycle latency associated with it. It is not envisioned to use this mode when peripherals are fully operational and moving data.   |

## 6.25 Programmable Real-Time Unit Subsystem (PRUSS)

The Programmable Real-Time Unit Subsystem (PRUSS) consists of

- Two Programmable Real-Time Units (PRU0 and PRU1) and their associated memories
- An Interrupt Controller (INTC) for handling system interrupt events. The INTC also supports posting events back to the device level host CPU.
- A Switched Central Resource (SCR) for connecting the various internal and external masters to the resources inside the PRUSS.

The two PRUs can operate completely independently or in coordination with each other. The PRUs can also work in coordination with the device level host CPU. This is determined by the nature of the program which is loaded into the PRUs instruction memory. Several different signaling mechanisms are available between the two PRUs and the device level host CPU.

The PRUs are optimized for performing embedded tasks that require manipulation of packed memory mapped data structures, handling of system events that have tight realtime constraints and interfacing with systems external to the device.

The PRUSS comprises various distinct addressable regions. Externally the subsystem presents a single 64Kbyte range of addresses. The internal interconnect bus (also called switched central resource, or SCR) of the PRUSS decodes accesses for each of the individual regions. The PRUSS memory map is documented in [Table 6-81](#) and in [Table 6-82](#). Note that these two memory maps are implemented inside the PRUSS and are local to the components of the PRUSS.

**Table 6-81. Programmable Real-Time Unit Subsystem (PRUSS) Local Instruction Space Memory Map**

| BYTE ADDRESS              | PRU0                 | PRU1                 |
|---------------------------|----------------------|----------------------|
| 0x0000 0000 - 0x0000 0FFF | PRU0 Instruction RAM | PRU1 Instruction RAM |

**Table 6-82. Programmable Real-Time Unit Subsystem (PRUSS) Local Data Space Memory Map**

| BYTE ADDRESS              | PRU0                      | PRU1                      |
|---------------------------|---------------------------|---------------------------|
| 0x0000 0000 - 0x0000 01FF | Data RAM 0 <sup>(1)</sup> | Data RAM 1 <sup>(1)</sup> |
| 0x0000 0200 - 0x0000 1FFF | Reserved                  | Reserved                  |
| 0x0000 2000 - 0x0000 21FF | Data RAM 1 <sup>(1)</sup> | Data RAM 0 <sup>(1)</sup> |
| 0x0000 2200 - 0x0000 3FFF | Reserved                  | Reserved                  |
| 0x0000 4000 - 0x0000 6FFF | INTC Registers            | INTC Registers            |
| 0x0000 7000 - 0x0000 73FF | PRU0 Control Registers    | PRU0 Control Registers    |
| 0x0000 7400 - 0x0000 77FF | Reserved                  | Reserved                  |
| 0x0000 7800 - 0x0000 7BFF | PRU1 Control Registers    | PRU1 Control Registers    |
| 0x0000 7C00 - 0xFFFF FFFF | Reserved                  | Reserved                  |

- (1) Note that PRU0 accesses Data RAM0 at address 0x0000 0000, also PRU1 accesses Data RAM1 at address 0x0000 0000. Data RAM0 is intended to be the primary data memory for PRU0 and Data RAM1 is intended to be the primary data memory for PRU1. However for passing information between PRUs, each PRU can access the data ram of the 'other' PRU through address 0x0000 2000.

The global view of the PRUSS internal memories and control ports is documented in [Table 6-83](#). The offset addresses of each region are implemented inside the PRUSS but the global device memory mapping places the PRUSS slave port in the address range 0x01C3 0000-0x01C3 FFFF. The PRU0 and PRU1 can use either the local or global addresses to access their internal memories, but using the local addresses will provide access time several cycles faster than using the global addresses. This is because when accessing via the global address the access needs to be routed through the switch fabric outside PRUSS and back in through the PRUSS slave port.

**Table 6-83. Programmable Real-Time Unit Subsystem (PRUSS) Global Memory Map**

| BYTE ADDRESS              | REGION                 |
|---------------------------|------------------------|
| 0x01C3 0000 - 0x01C3 01FF | Data RAM 0             |
| 0x01C3 0200 - 0x01C3 1FFF | Reserved               |
| 0x01C3 2000 - 0x01C3 21FF | Data RAM 1             |
| 0x01C3 2200 - 0x01C3 3FFF | Reserved               |
| 0x01C3 4000 - 0x01C3 6FFF | INTC Registers         |
| 0x01C3 7000 - 0x01C3 73FF | PRU0 Control Registers |
| 0x01C3 7400 - 0x01C3 77FF | PRU0 Debug Registers   |
| 0x01C3 7800 - 0x01C3 7BFF | PRU1 Control Registers |
| 0x01C3 7C00 - 0x01C3 7FFF | PRU1 Debug Registers   |
| 0x01C3 8000 - 0x01C3 8FFF | PRU0 Instruction RAM   |
| 0x01C3 9000 - 0x01C3 BFFF | Reserved               |
| 0x01C3 C000 - 0x01C3 CFFF | PRU1 Instruction RAM   |
| 0x01C3 D000 - 0x01C3 FFFF | Reserved               |

Each of the PRUs can access the rest of the device memory (including memory mapped peripheral and configuration registers) using the global memory space addresses

### 6.25.1 PRUSS Register Descriptions

**Table 6-84. Programmable Real-Time Unit Subsystem (PRUSS) Control / Status Registers**

| PRU0 BYTE ADDRESS       | PRU1 BYTE ADDRESS         | ACRONYM              | REGISTER DESCRIPTION                                |
|-------------------------|---------------------------|----------------------|---|
| 0x01C3 7000             | 0x01C3 7800               | CONTROL              | PRU Control Register                                |
| 0x01C3 7004             | 0x01C3 7804               | STATUS               | PRU Status Register                                 |
| 0x01C3 7008             | 0x01C3 7808               | WAKEUP               | PRU Wakeup Enable Register                          |
| 0x01C3 700C             | 0x01C3 780C               | CYCLCNT              | PRU Cycle Count                                     |
| 0x01C3 7010             | 0x01C3 7810               | STALLCNT             | PRU Stall Count                                     |
| 0x01C3 7020             | 0x01C3 7820               | CONTABBLKIDX0        | PRU Constant Table Block Index Register 0           |
| 0x01C3 7028             | 0x01C3 7828               | CONTABPROPTR0        | PRU Constant Table Programmable Pointer Register 0  |
| 0x01C3 702C             | 0x01C3 782C               | CONTABPROPTR1        | PRU Constant Table Programmable Pointer Register 1  |
| 0x01C37400 - 0x01C3747C | 0x01C3 7C00 - 0x01C3 7C7C | INTGPR0 – INTGPR31   | PRU Internal General Purpose Register 0 (for Debug) |
| 0x01C37480 - 0x01C374FC | 0x01C3 7C80 - 0x01C3 7CFC | INTCTER0 – INTCTER31 | PRU Internal General Purpose Register 0 (for Debug) |

**Table 6-85. Programmable Real-Time Unit Subsystem Interrupt Controller (PRUSS INTC) Registers**

| BYTE ADDRESS | ACRONYM        | REGISTER DESCRIPTION                           |
|--------------|----------------|--|
| 0x01C3 4000  | REVID          | Revision ID Register                           |
| 0x01C3 4004  | CONTROL        | Control Register                               |
| 0x01C3 4010  | GLBLEN         | Global Enable Register                         |
| 0x01C3 401C  | GLBLNSTLVL     | Global Nesting Level Register                  |
| 0x01C3 4020  | STATIDXSET     | System Interrupt Status Indexed Set Register   |
| 0x01C3 4024  | STATIDXCLR     | System Interrupt Status Indexed Clear Register |
| 0x01C3 4028  | ENIDXSET       | System Interrupt Enable Indexed Set Register   |
| 0x01C3 402C  | ENIDXCLR       | System Interrupt Enable Indexed Clear Register |
| 0x01C3 4034  | HSTINTENIDXSET | Host Interrupt Enable Indexed Set Register     |
| 0x01C3 4038  | HSTINTENIDXCLR | Host Interrupt Enable Indexed Clear Register   |

**Table 6-85. Programmable Real-Time Unit Subsystem Interrupt Controller (PRUSS INTC)  
Registers (continued)**

| BYTE ADDRESS              | ACRONYM                            | REGISTER DESCRIPTION                             |
|---------------------------|------------------------------------|--|
| 0x01C3 4080               | GLBLPRIIDX                         | Global Prioritized Index Register                |
| 0x01C3 4200               | STATSETINT0                        | System Interrupt Status Raw/Set Register 0       |
| 0x01C3 4204               | STATSETINT1                        | System Interrupt Status Raw/Set Register 1       |
| 0x01C3 4280               | STATCLRINT0                        | System Interrupt Status Enabled/Clear Register 0 |
| 0x01C3 4284               | STATCLRINT1                        | System Interrupt Status Enabled/Clear Register 1 |
| 0x01C3 4300               | ENABLESET0                         | System Interrupt Enable Set Register 0           |
| 0x01C3 4304               | ENABLESET1                         | System Interrupt Enable Set Register 1           |
| 0x01C3 4380               | ENABLECLR0                         | System Interrupt Enable Clear Register 0         |
| 0x01C3 4384               | ENABLECLR1                         | System Interrupt Enable Clear Register 1         |
| 0x01C3 4400 - 0x01C3 4440 | CHANMAP0 - CHANMAP15               | Channel Map Registers 0-15                       |
| 0x01C3 4800 - 0x01C3 4808 | HOSTMAP0 - HOSTMAP2                | Host Map Register 0-2                            |
| 0x01C3 4900 - 0x01C3 4928 | HOSTINTPRIIDX0 -<br>HOSTINTPRIIDX9 | Host Interrupt Prioritized Index Registers 0-9   |
| 0x01C3 4D00               | POLARITY0                          | System Interrupt Polarity Register 0             |
| 0x01C3 4D04               | POLARITY1                          | System Interrupt Polarity Register 1             |
| 0x01C3 4D80               | TYPE0                              | System Interrupt Type Register 0                 |
| 0x01C3 4D84               | TYPE1                              | System Interrupt Type Register 1                 |
| 0x01C3 5100 - 0x01C3 5128 | HOSTINTNSTLVL0-<br>HOSTINTNSTLVL9  | Host Interrupt Nesting Level Registers 0-9       |
| 0x01C3 5500               | HOSTINTEN                          | Host Interrupt Enable Register                   |

## 6.26 Emulation Logic

The debug capabilities and features for DSP are as shown below.

### DSP:

- Basic Debug
  - Execution Control
  - System Visibility
- Real-Time Debug
  - Interrupts serviced while halted
  - Low/non-intrusive system visibility while running
- Advanced Debug
  - Global Start
  - Global Stop
  - Specify targeted memory level(s) during memory accesses
  - HSRTDX (High Speed Real Time Data eXchange)
- Advanced System Control
  - Subsystem reset via debug
  - Peripheral notification of debug events
  - Cache-coherent debug accesses
- Analysis Actions
  - Stop program execution
  - Generate debug interrupt
  - Benchmarking with counters
  - External trigger generation
  - Debug state machine state transition
  - Combinational and Sequential event generation
- Analysis Events
  - Program event detection
  - Data event detection
  - External trigger Detection
  - System event detection (i.e. cache miss)
  - Debug state machine state detection
- Analysis Configuration
  - Application access
  - Debugger access

**Table 6-86. DSP Debug Features**

| CATEGORY    | HARDWARE FEATURE    | AVAILABILITY  |
|-------------|---------------------|---|
| Basic Debug | Software breakpoint | Unlimited   |
|             | Hardware breakpoint | Up to 10 HWBPs, including:<br>4 precise <sup>(1)</sup> HWBPs inside DSP core and one of them is associated with a counter.<br>2 imprecise <sup>(1)</sup> HWBPs from AET.<br>4 imprecise <sup>(1)</sup> HWBPs from AET which are shared for watch point. |

(1) Precise hardware breakpoints will halt the processor immediately prior to the execution of the selected instruction. Imprecise breakpoints will halt the processor some number of cycles after the selected instruction depending on device conditions.

**Table 6-86. DSP Debug Features (continued)**

| CATEGORY | HARDWARE FEATURE           | AVAILABILITY  |
|----------|----------------------------|---|
| Analysis | Watch point                | Up to 4 watch points, which are shared with HWBPs, and can also be used as 2 watch points with data (32 bits) |
|          | Watch point with Data      | Up to 2, Which can also be used as 4 watch points.  |
|          | Counters/timers            | 1x64-bits (cycle only) + 2x32-bits (water marke counters)   |
|          | External Event Trigger In  | 1   |
|          | External Event Trigger Out | 1   |

### 6.26.1 JTAG Port Description

The device target debug interface uses the five standard IEEE 1149.1(JTAG) signals ( $\overline{\text{TRST}}$ , TCK, TMS, TDI, and TDO).

$\overline{\text{TRST}}$  holds the debug and boundary scan logic in reset (normal DSP operation) when pulled low (its default state). Since  $\overline{\text{TRST}}$  has an internal pull-down resistor, this ensures that at power up the device functions in its normal (non-test) operation mode if  $\overline{\text{TRST}}$  is not connected. Otherwise,  $\overline{\text{TRST}}$  should be driven inactive by the emulator or boundary scan controller. Boundary scan test cannot be performed while the  $\overline{\text{TRST}}$  pin is pulled low.

**Table 6-87. JTAG Port Description**

| PIN                      | TYPE | NAME             | DESCRIPTION   |
|--------------------------|------|------------------|---|
| $\overline{\text{TRST}}$ | I    | Test Logic Reset | When asserted (active low) causes all test and debug logic in the device to be reset along with the IEEE 1149.1 interface |
| TCK                      | I    | Test Clock       | This is the test clock used to drive an IEEE 1149.1 TAP state machine and logic.  |
| TMS                      | I    | Test Mode Select | Directs the next state of the IEEE 1149.1 test access port state machine  |
| TDI                      | I    | Test Data Input  | Scan data input to the device   |
| TDO                      | O    | Test Data Output | Scan data output of the device  |
| EMU0                     | I/O  | Emulation 0      | Channel 0 trigger + HSRTDX  |

### 6.26.2 Scan Chain Configuration Parameters

Table 6-88 shows the TAP configuration details required to configure the router/emulator for this device.

**Table 6-88. Router TAP Configuration**

| ROUTER PORT ID | DEFAULT TAP | TAP NAME | TAP IR LENGTH |
|----------------|-------------|----------|---------------|
| 17             | No          | C674x    | 38            |

The router is revision C and has a 6-bit IR length.

### 6.26.3 JTAG 1149.1 Boundary Scan Considerations

To use boundary scan, the following sequence should be followed:

- Execute a valid reset sequence and exit reset
- Wait at least 6000 OSCIN clock cycles
- Enter boundary scan mode using the JTAG pins

No specific value is required on the EMU0 pin for boundary scan testing. If  $\overline{\text{TRST}}$  is not driven by the boundary scan tool or tester,  $\overline{\text{TRST}}$  should be externally pulled high during boundary scan testing.

## 6.27 IEEE 1149.1 JTAG

The JTAG <sup>(1)</sup> interface is used for BSDL testing and emulation of the device.

The device requires that both  $\overline{\text{TRST}}$  and  $\overline{\text{RESET}}$  be asserted upon power up to be properly initialized. While  $\overline{\text{RESET}}$  initializes the device,  $\overline{\text{TRST}}$  initializes the device's emulation logic. Both resets are required for proper operation.

While both  $\overline{\text{TRST}}$  and  $\overline{\text{RESET}}$  need to be asserted upon power up, only  $\overline{\text{RESET}}$  needs to be released for the device to boot properly.  $\overline{\text{TRST}}$  may be asserted indefinitely for normal operation, keeping the JTAG port interface and device's emulation logic in the reset state.

$\overline{\text{TRST}}$  only needs to be released when it is necessary to use a JTAG controller to debug the device or exercise the device's boundary scan functionality.

$\overline{\text{RESET}}$  must be released only in order for boundary-scan JTAG to read the variant field of IDCODE correctly. Other boundary-scan instructions work correctly independent of current state of  $\overline{\text{RESET}}$ .

For maximum reliability, the device includes an internal pulldown (IPD) on the  $\overline{\text{TRST}}$  pin to ensure that  $\overline{\text{TRST}}$  will always be asserted upon power up and the device's internal emulation logic will always be properly initialized.

JTAG controllers from Texas Instruments actively drive  $\overline{\text{TRST}}$  high. However, some third-party JTAG controllers may not drive  $\overline{\text{TRST}}$  high but expect the use of a pullup resistor on  $\overline{\text{TRST}}$ .

When using this type of JTAG controller, assert  $\overline{\text{TRST}}$  to initialize the device after powerup and externally drive  $\overline{\text{TRST}}$  high before attempting any emulation or boundary scan operations.

**Note:** The sequencing of all the JTAG signals must follow the IEEE.1149.1 JTAG standard.

### 6.27.1 JTAG Peripheral Register Description(s) – JTAG ID Register

Table 6-89. JTAG ID Register

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME                | COMMENTS  |
|-------------------|---------|------------------------------|---|
| 0x01C4 0028       | JTAGID  | JTAG Identification Register | Read-only. Provides 32-bit JTAG ID of the device. |

(1) IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

The JTAG ID register is a read-only register that identifies to the customer the JTAG/Device ID. The register hex value each silicon revision is:

- 0x0B79 F02F for silicon revision 1.0.
- 0x8B79 F02F for silicon revision 1.1.
- 0x9B79 F02F for silicon revision 3.0, 2.1, and 2.0.

For the actual register bit names and their associated bit field descriptions, see [Figure 6-48](#) and [Table 6-90](#).

Figure 6-48. JTAG ID (DEVIDR0) Register Description - Register Value

| 31              | 28 | 27 | 12                    | 11 | 1                     | 0 |
|-----------------|----|----|-----------------------|----|-----------------------|---|
| VARIANT (4-bit) |    |    | PART NUMBER (16-bit)  |    | MANUFACTURER (11-bit) |   |
| R-xxxx          |    |    | R-1011 0111 1101 1111 |    | R-0000 0010 111       |   |
|                 |    |    |                       |    | LSB                   |   |
|                 |    |    |                       |    | R-1                   |   |

LEGEND: R = Read, W = Write, n = value after reset

**Table 6-90. JTAG ID Register Selection Bit Descriptions**

| BIT   | NAME         | DESCRIPTION                     |
|-------|--------------|---------------------------------|
| 31:28 | VARIANT      | Variant (4-bit) value           |
| 27:12 | PART NUMBER  | Part Number (16-Bit) value      |
| 11-1  | MANUFACTURER | Manufacturer (11-Bit) value     |
| 0     | LSB          | LSB. This bit is read as a "1". |

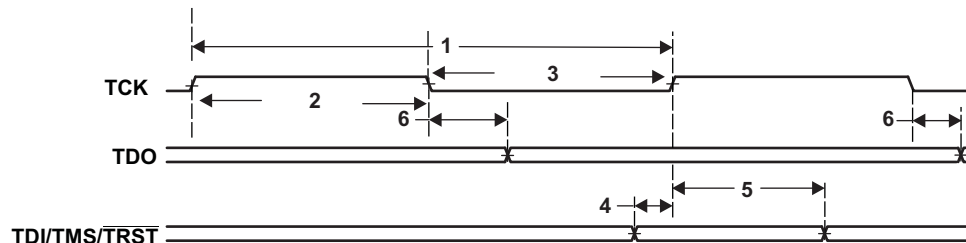
### 6.27.2 JTAG Test-Port Electrical Data/Timing

**Table 6-91. Timing Requirements for JTAG Test Port (see Figure 6-49)**

| No. | PARAMETER                  |   | MIN | MAX | UNIT |
|-----|----------------------------|---|-----|-----|------|
| 1   | $t_c(\text{TCK})$          | Cycle time, TCK   | 40  |     | ns   |
| 2   | $t_w(\text{TCKH})$         | Pulse duration, TCK high  | 16  |     | ns   |
| 3   | $t_w(\text{TCKL})$         | Pulse duration, TCK low   | 16  |     | ns   |
| 4   | $t_{su}(\text{TDIV-TCKH})$ | Setup time, TDI/TMS/ $\overline{\text{TRST}}$ valid before TCK high | 4   |     | ns   |
| 5   | $t_h(\text{TCKH-TDIV})$    | Hold time, TDI/TMS/ $\overline{\text{TRST}}$ valid after TCK high   | 4   |     | ns   |

**Table 6-92. Switching Characteristics Over Recommended Operating Conditions for JTAG Test Port (see Figure 6-49)**

| No. | PARAMETER               |                                  | MIN | MAX | UNIT |
|-----|-------------------------|----------------------------------|-----|-----|------|
| 6   | $t_d(\text{TCKL-TDOV})$ | Delay time, TCK low to TDO valid |     | 15  | ns   |

**Figure 6-49. JTAG Test-Port Timing**

## 7 Device and Documentation Support

### 7.1 Device Support

#### 7.1.1 Development Support

TI offers an extensive line of development tools for the TMS320C6743 platform, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules. The tool's support documentation is electronically available within the Code Composer Studio™ Integrated Development Environment (IDE).

The following products support development of TMS320C6743 applications:

##### Software Development Tools:

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

##### Hardware Development Tools:

Extended Development System (XDS™) Emulator

For a complete listing of development-support tools for the device, visit the Texas Instruments web site on the Worldwide Web at [www.ti.com](http://www.ti.com) uniform resource locator (URL). For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

#### 7.1.2 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all DSP devices and support tools. Each DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (e.g., TMS320C6743). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

|            |   |
|------------|---|
| <b>TMX</b> | Experimental device that is not necessarily representative of the final device's electrical specifications.                           |
| <b>TMP</b> | Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification. |
| <b>TMS</b> | Fully-qualified production device.  |

Support tool development evolutionary flow:

|             |  |
|-------------|--|
| <b>TMDX</b> | Development-support product that has not yet completed Texas Instruments internal qualification testing. |
| <b>TMDS</b> | Fully qualified development-support product.   |

TMX and TMP devices and TMDX development-support tools are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

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

Predictions show that prototype devices (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.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, ZWT), the temperature range (for example, "Blank" is the commercial temperature range), and the device speed range in megahertz (for example, "Blank" is the default).

Figure 7-1 provides a legend for reading the complete device name for any TMS320C674x member.

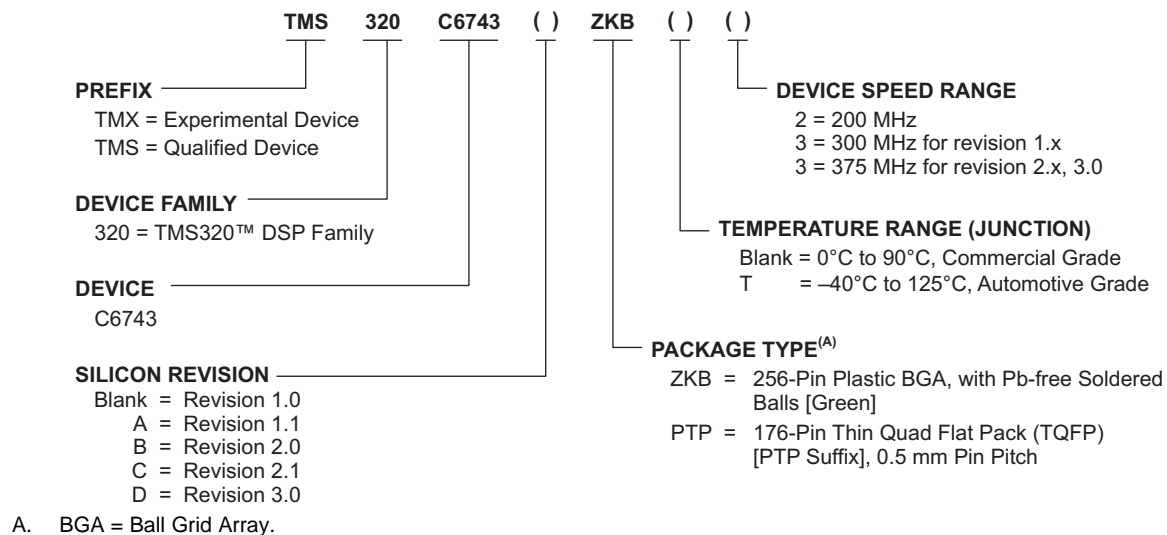


Figure 7-1. Device Nomenclature

## 7.2 Documentation Support

The following documents describe the TMS320C6743 Low-power digital signal processor. Copies of these documents are available on the Internet at [www.ti.com](http://www.ti.com). *Tip:* Enter the literature number in the search box provided at [www.ti.com](http://www.ti.com).

### DSP Reference Guides

**SPRUG82** *TMS320C674x DSP Cache User's Guide.* Explains the fundamentals of memory caches and describes how the two-level cache-based internal memory architecture in the TMS320C674x digital signal processor (DSP) can be efficiently used in DSP applications. Shows how to maintain coherence with external memory, how to use DMA to reduce memory latencies, and how to optimize your code to improve cache efficiency. The internal memory architecture in the C674x DSP is organized in a two-level hierarchy consisting of a dedicated program cache (L1P) and a dedicated data cache (L1D) on the first level. Accesses by the CPU to these first level caches can complete without CPU pipeline stalls. If the data requested by the CPU is not contained in cache, it is fetched from the next lower memory level, L2 or external memory.

**SPRUF8** *TMS320C674x DSP CPU and Instruction Set Reference Guide.* Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C674x digital signal processors (DSPs). The C674x DSP is an enhancement of the C64x+ and C67x+ DSPs with added functionality and an expanded instruction set.

- SPRU186** ***TMS320C6000 Assembly Language Tools User's Guide.*** Describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the TMS320C6000 platform of devices (including the C64x+, C67x+, and C674x generations).
- SPRU187** ***TMS320C6000 Optimizing Compiler User's Guide.*** Describes the TMS320C6000 C compiler and the assembly optimizer. This C compiler accepts ANSI standard C source code and produces assembly language source code for the TMS320C6000 platform of devices (including the C64x+, C67x+, and C674x generations). The assembly optimizer helps you optimize your assembly code.
- SPRUH90** ***TMS320C6743 DSP Technical Reference Manual.*** Describes the System-on-Chip (SoC) and each peripheral in the device.
- SPRUFK5** ***TMS320C674x DSP Megamodule Reference Guide.*** Describes the TMS320C674x digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.
- SPRUFK9** ***TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide.*** Provides an overview and briefly describes the peripherals available on TMS320C6743.

### 7.3 Support Resources

[TI E2E™ support forums](#) are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is 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](#).

### 7.4 Trademarks

TI E2E is a trademark of Texas Instruments.

All other trademarks are the property of their respective owners.

### 7.5 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.

### 7.6 Glossary

[TI Glossary](#) This glossary lists and explains terms, acronyms, and definitions.

## 8 Mechanical Packaging and Orderable Information

This section describes the orderable part numbers, packaging options, materials, thermal and mechanical parameters.

### 8.1 Thermal Data for ZKB

The following table(s) show the thermal resistance characteristics for the PBGA–ZKB mechanical package.

**Table 8-1. Thermal Resistance Characteristics (PBGA Package) [ZKB]**

| NO. | CHARACTERISTIC   |                         | °C/W <sup>(1)</sup> | °C/W <sup>(2)</sup> | AIR FLOW (m/s) <sup>(3)</sup> |
|-----|------------------|-------------------------|---------------------|---------------------|-------------------------------|
| 1   | R $\theta_{JC}$  | Junction-to-case        | 12.8                | 13.5                | N/A                           |
| 2   | R $\theta_{JB}$  | Junction-to-board       | 15.1                | 19.7                | N/A                           |
| 3   | R $\theta_{JA}$  | Junction-to-free air    | 24.5                | 33.8                | 0.00                          |
| 4   | R $\theta_{JMA}$ | Junction-to-moving air  | 21.9                | 30                  | 0.50                          |
| 5   |                  |                         | 21.1                | 28.7                | 1.00                          |
| 6   |                  |                         | 20.4                | 27.4                | 2.00                          |
| 7   |                  |                         | 19.6                | 26                  | 4.00                          |
| 8   | Psi $_{JT}$      | Junction-to-package top | 0.6                 | 0.8                 | 0.00                          |
| 9   |                  |                         | 0.8                 | 1                   | 0.50                          |
| 10  |                  |                         | 0.9                 | 1.2                 | 1.00                          |
| 11  |                  |                         | 1.1                 | 1.4                 | 2.00                          |
| 12  |                  |                         | 1.3                 | 1.8                 | 4.00                          |
| 13  | Psi $_{JB}$      | Junction-to-board       | 14.9                | 19.1                | 0.00                          |
| 14  |                  |                         | 14.4                | 18.2                | 0.50                          |
| 15  |                  |                         | 14.4                | 18                  | 1.00                          |
| 16  |                  |                         | 14.3                | 17.7                | 2.00                          |
| 17  |                  |                         | 14.1                | 17.4                | 4.00                          |

- (1) These measurements were conducted in a JEDEC defined 2S2P system and will change based on environment as well as application. For more information, see these EIA/JEDEC standards – EIA/JESD51-2, *Integrated Circuits Thermal Test Method Environment Conditions - Natural Convection (Still Air)* and JESD51-7, *High Effective Thermal Conductivity Test Board for Leadless Surface Mount Packages*. Power dissipation of 1W and ambient temp of 70C assumed. PCB with 2oz (70um) top and bottom copper thickness and 1.5oz (50um) inner copper thickness
- (2) Simulation data, using the same model but with 1oz (35um) top and bottom copper thickness and 0.5oz (18um) inner copper thickness. Power dissipation of 1W and ambient temp of 70C assumed.
- (3) m/s = meters per second

## 8.2 Thermal Data for PTP

The following table(s) show the thermal resistance characteristics for the HTQFP–PTP mechanical package.

**Table 8-2. Thermal Resistance Characteristics (HTQFP Package) [PTP]**

| NO. | CHARACTERISTIC                          | °C/W <sup>(1)</sup> | °C/W <sup>(2)</sup> | °C/W <sup>(3)</sup> | °C/W <sup>(4)</sup> | AIR FLOW (m/s) <sup>(5)</sup> |
|-----|---|---------------------|---------------------|---------------------|---------------------|-------------------------------|
| 1   | R $\theta_{JC}$ Junction-to-case        | 7.8                 | 9.4                 | 8.6                 | 10.1                | N/A                           |
| 2   | R $\theta_{JB}$ Junction-to-board       | 6.2                 | 9.9                 | 7.1                 | 10.6                | N/A                           |
| 3   | R $\theta_{JA}$ Junction-to-free air    | 21.3                | 27.9                | 23.2                | 30.6                | 0.00                          |
| 4   | R $\theta_{JMA}$ Junction-to-moving air | 14.3                | 20.2                |                     | 22.6                | 0.50                          |
| 5   |   | 13.1                | 18.6                |                     | 21.0                | 1.00                          |
| 6   |   | 12.1                | 17.4                |                     | 19.6                | 2.00                          |
| 7   |   | 11.2                | 16.2                |                     | 18.2                | 4.00                          |
| 8   |   | 0.5                 | 0.7                 |                     | 0.8                 | 0.00                          |
| 9   | Psi $_{JT}$ Junction-to-package top     | 0.6                 | 0.9                 |                     | 1.0                 | 0.50                          |
| 10  |   | 0.7                 | 1.0                 |                     | 1.1                 | 1.00                          |
| 11  |   | 0.8                 | 1.1                 |                     | 1.3                 | 2.00                          |
| 12  |   | 1.0                 | 1.3                 |                     | 1.5                 | 4.00                          |
| 13  |   | 6.3                 | 9.5                 |                     | 10.8                | 0.00                          |
| 14  | Psi $_{JB}$ Junction-to-board           | 5.9                 | 8.8                 |                     | 9.9                 | 0.50                          |
| 15  |   | 5.9                 | 8.7                 |                     | 9.8                 | 1.00                          |
| 16  |   | 5.8                 | 8.6                 |                     | 9.7                 | 2.00                          |
| 17  |   | 5.8                 | 8.5                 |                     | 9.6                 | 4.00                          |

- (1) Simulation data, using a model of a JEDEC defined 2S2P system with a 12mmx12mm copper pad on the top and bottom copper layers connected with an 8x8 thermal via array and soldered to the package thermal pad. Power dissipation of 1W assumed, 70C Ambient temp assumed. Signal layer copper coverage 20%, inner layer copper coverage 90%. Actual performance will change based on environment as well as application. For more information, see these EIA/JEDEC standards – EIA/JESD51-2, *Integrated Circuits Thermal Test Method Environment Conditions - Natural Convection (Still Air)* and JESD51-7, *High Effective Thermal Conductivity Test Board for Leaded Surface Mount Packages*.
- (2) Simulation data, using the same model but with 1oz (35um) top and bottom copper thickness and 0.5oz (18um) inner copper thickness. Power dissipation of 1W and ambient temp of 70C assumed.
- (3) Simulation data, 1S1P PCB model with 12x12mm copper pad on the top layer soldered to device thermal pad and connected to the bottom copper layer (90% copper) with an 8x8 thermal via array. Power dissipation of 1W and ambient temp of 70C assumed. Copper thickness 2oz (70um) top and bottom.
- (4) Simulation data, 1S1P PCB model with 12x12mm copper pad on the top layer soldered to device thermal pad and connected to the bottom copper layer (90% copper) with an 8x8 thermal via array. Power dissipation of 1W and ambient temp of 70C assumed. Copper thickness 1oz (35um) top and bottom.
- (5) m/s = meters per second

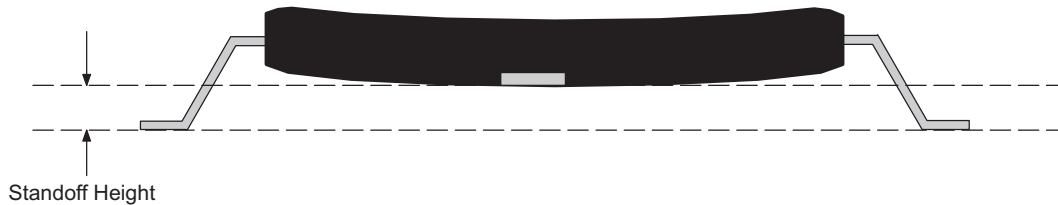
## 8.3 Supplementary Information About the 176-pin PTP PowerPAD™ Package

This section highlights a few important details about the 176-pin PTP PowerPAD™ package. Texas Instruments' *PowerPAD Thermally Enhanced Package Technical Brief (SLMA002)* should be consulted when creating a PCB footprint for this device.

### 8.3.1 Standoff Height

As illustrated in [Figure 8-1](#), the standoff height specification for this device (between 0.050 mm and 0.150 mm) is measured from the seating plane established by the three lowest package pins to the **lowest** point on the package body. Due to warpage, the lowest point on the package body is located in the center of the package at the exposed thermal pad.

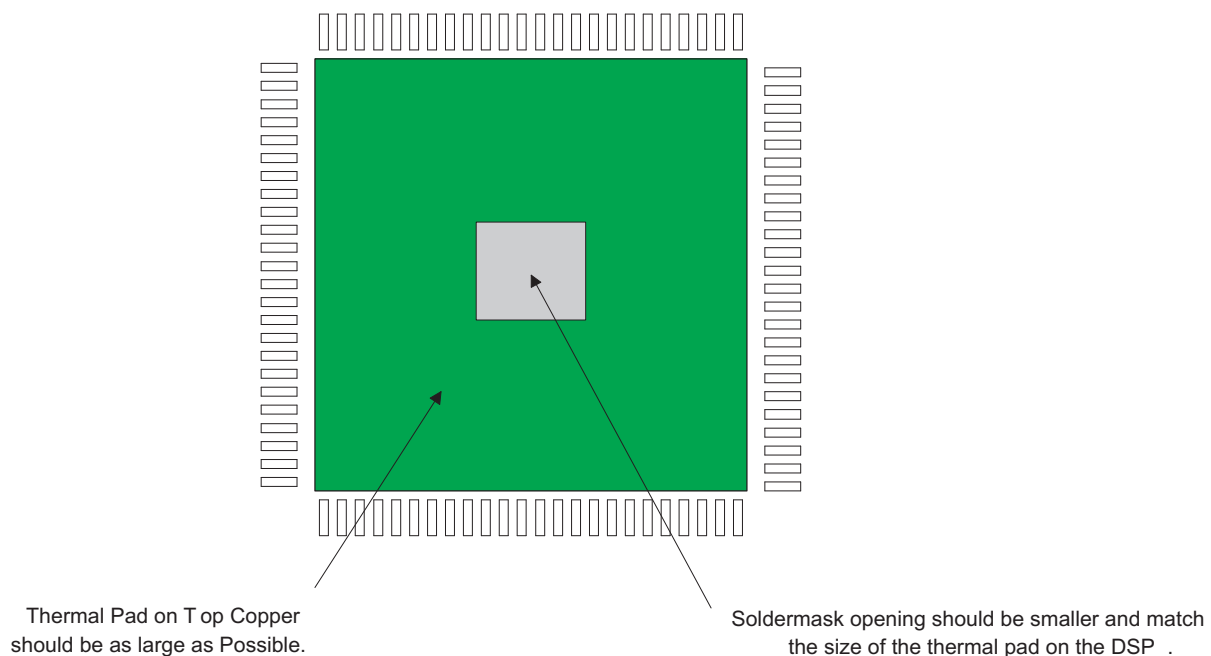
Using this definition of standoff height provides the correct result for determining the correct solder paste thickness. According to TI's *PowerPAD Thermally Enhanced Package Technical Brief (SLMA002)*, the recommended range of solder paste thickness for this package is between 0.152 mm and 0.178 mm.



**Figure 8-1. Standoff Height Measurement on 176-pin PTP Package**

### 8.3.2 PowerPAD™ PCB Footprint

In general, for proper thermal performance, the thermal pad under the package body should be as large as possible. However, the soldermask opening for the PowerPAD™ should be sized to match the pad size on the 176-pin PTP package; as illustrated in [Figure 8-2](#).






**Figure 8-2. Soldermask Opening Should Match Size of DSP Thermal Pad**

## 8.4 Mechanical Drawings

This section contains mechanical drawings for the ZKB Plastic Ball Grid Array package and the PTP Thin Quad Flat Pack package. Additionally, for the PTP package a detailed drawing of the actual thermal pad dimensions as well as a recommended PCB footprint are provided.

**PACKAGING INFORMATION**

| Orderable Device  | Status<br>(1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan<br>(2) | Lead finish/<br>Ball material<br>(6) | MSL Peak Temp<br>(3) | Op Temp (°C) | Device Marking<br>(4/5) | Samples   |
|-------------------|---------------|--------------|-----------------|------|-------------|-----------------|--------------------------------------|----------------------|--------------|-------------------------|---|
| TMS320C6743DPTP2  | ACTIVE        | HLQFP        | PTP             | 176  | 40          | RoHS & Green    | NIPDAU                               | Level-4-260C-72 HR   | 0 to 90      | TMS320<br>C6743DPTP2    |  |
| TMS320C6743DPTP3  | ACTIVE        | HLQFP        | PTP             | 176  | 40          | RoHS & Green    | NIPDAU                               | Level-4-260C-72 HR   | 0 to 90      | TMS320<br>C6743DPTP3    |  |
| TMS320C6743DZKBT3 | ACTIVE        | BGA          | ZKB             | 256  | 90          | RoHS & Green    | SNAGCU                               | Level-3-260C-168 HR  | -40 to 125   | TMS320<br>C6743DZKBT3   |  |

(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.

**OBSELETE:** 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.

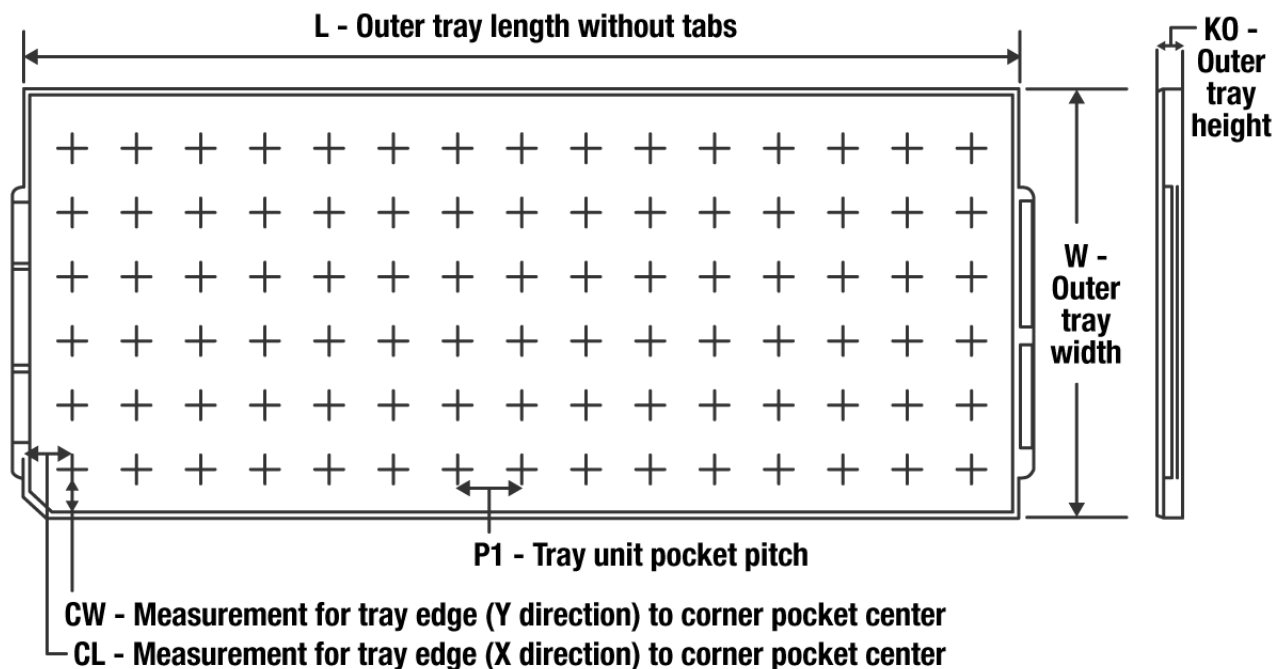
(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(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 finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material 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.

**TRAY**


Chamfer on Tray corner indicates Pin 1 orientation of packed units.

\*All dimensions are nominal

| Device            | Package Name | Package Type | Pins | SPQ | Unit array matrix | Max temperature (°C) | L (mm) | W (mm) | K0 (µm) | P1 (mm) | CL (mm) | CW (mm) |
|-------------------|--------------|--------------|------|-----|-------------------|----------------------|--------|--------|---------|---------|---------|---------|
| TMS320C6743DPTP3  | PTP          | HLQFP        | 176  | 40  | 4x10              | 150                  | 315    | 135.9  | 7620    | 20.7    | 30.4    | 20.7    |
| TMS320C6743DZKBT3 | ZKB          | BGA          | 256  | 90  | 6 X 15            | 150                  | 315    | 135.9  | 7620    | 19.5    | 21      | 19.2    |



## GENERIC PACKAGE VIEW

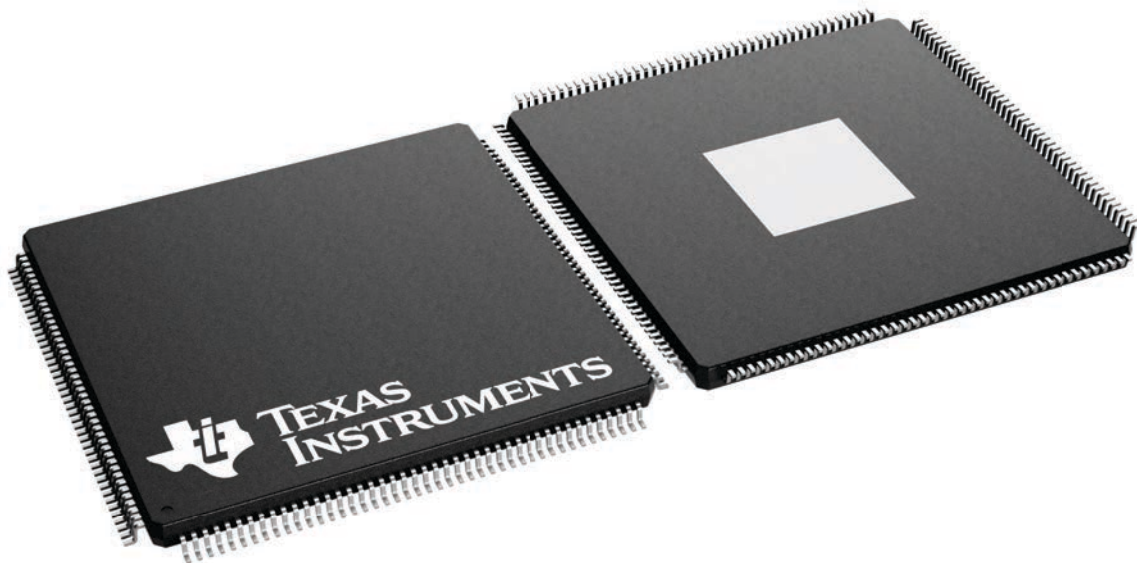
**PTP 176**

**HLQFP - 1.6 mm max height**

24 x 24, 0.5 mm pitch

PLASTIC QUAD FLATPACK

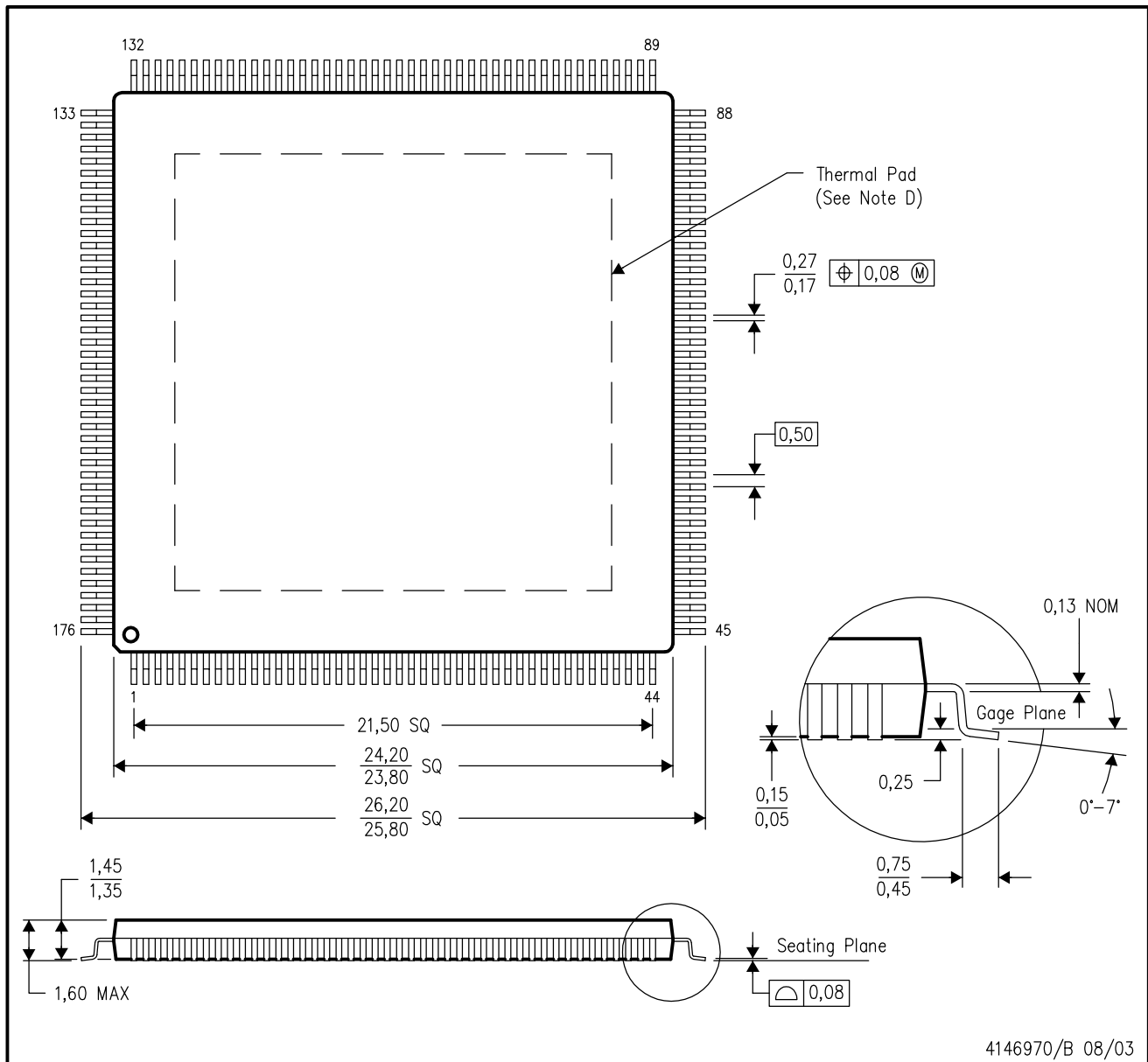
This image is a representation of the package family, actual package may vary.  
Refer to the product data sheet for package details.



4226435/A

PTP (S-PQFP-G176)

PowerPAD™ PLASTIC QUAD FLATPACK



- NOTES:
- All linear dimensions are in millimeters.
  - This drawing is subject to change without notice.
  - Body dimensions do not include mold flash or protrusion.
  - This package is designed to be soldered to a thermal pad on the board. Refer to Technical Brief, PowerPad Thermally Enhanced Package, Texas Instruments Literature No. SLMA002 for information regarding recommended board layout. This document is available at [www.ti.com](http://www.ti.com) <<http://www.ti.com>>.
  - Falls within JEDEC MO-026

PowerPAD is a trademark of Texas Instruments.

PTP (S-PQFP-G176)

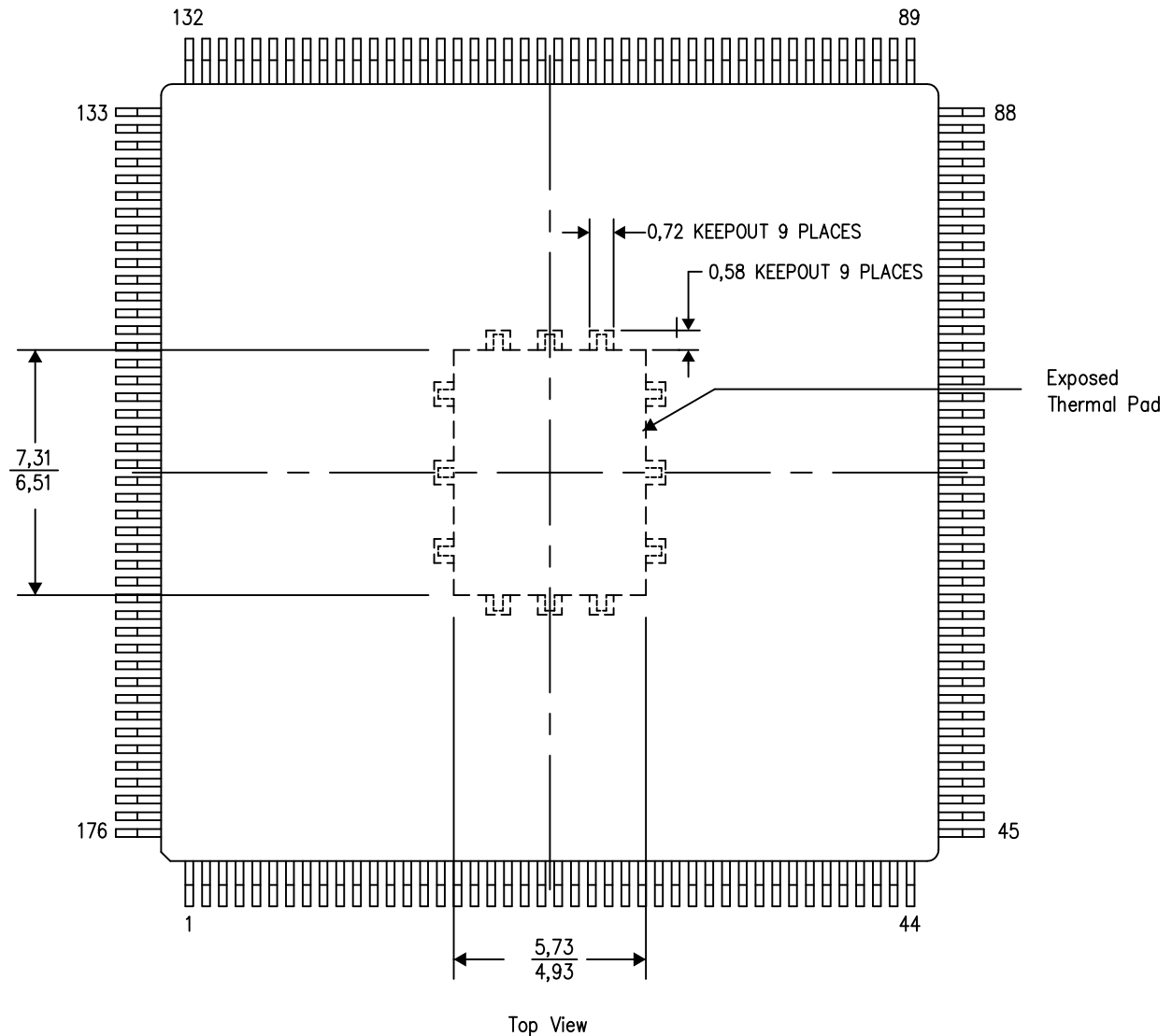
PowerPAD™ PLASTIC QUAD FLATPACK

**THERMAL INFORMATION**

This PowerPAD™ package incorporates an exposed thermal pad that is designed to be attached to a printed circuit board (PCB). The thermal pad must be soldered directly to the PCB. After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For additional information on the PowerPAD package and how to take advantage of its heat dissipating abilities, refer to Technical Brief, PowerPAD Thermally Enhanced Package, Texas Instruments Literature No. SLMA002 and Application Brief, PowerPAD Made Easy, Texas Instruments Literature No. SLMA004. Both documents are available at [www.ti.com](http://www.ti.com).

The exposed thermal pad dimensions for this package are shown in the following illustration.



Exposed Thermal Pad Dimensions

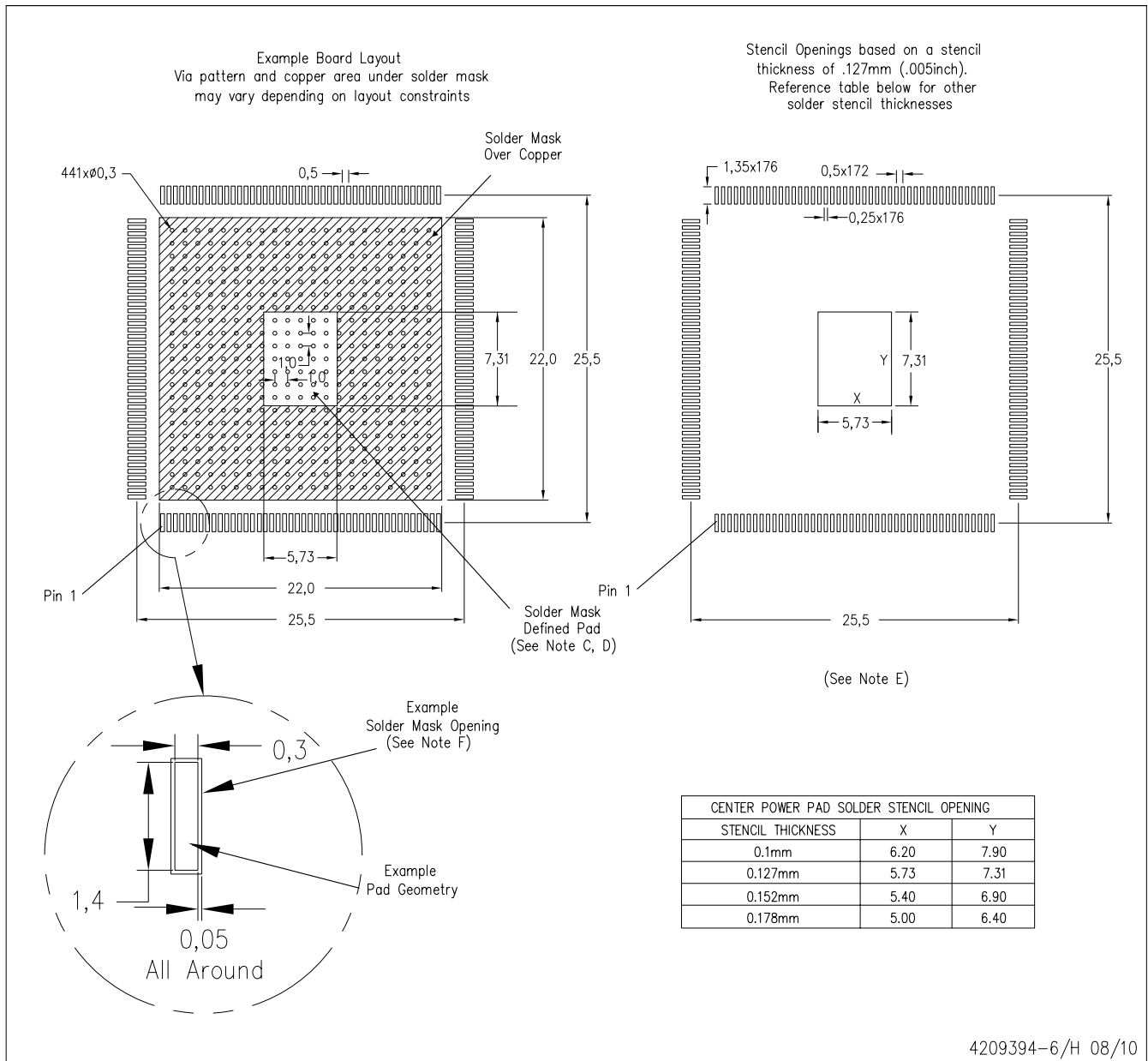
4209350-6/F 04/13

NOTE: All linear dimensions are in millimeters

PowerPAD is a trademark of Texas Instruments

PTP (S-PQFP-G176)

PowerPAD™ PLASTIC QUAD FLATPACK



- NOTES:
- All linear dimensions are in millimeters.
  - This drawing is subject to change without notice.
  - Customers should place a note on the circuit board fabrication drawing not to alter the center solder mask defined pad.
  - This package is designed to be soldered to a thermal pad on the board. Refer to Technical Brief, PowerPad Thermally Enhanced Package, Texas Instruments Literature No. SLMA002, SLMA004, and also the Product Data Sheets for specific thermal information, via requirements, and recommended board layout. These documents are available at [www.ti.com](http://www.ti.com) <<http://www.ti.com>>. Publication IPC-7351 is recommended for alternate designs.
  - Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Example stencil design based on a 50% volumetric metal load solder paste. Refer to IPC-7525 for other stencil recommendations.
  - Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

PowerPAD is a trademark of Texas Instruments.

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2022, Texas Instruments Incorporated


## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

 [View TMS320C6743DPTP3 on WIN SOURCE](#)

 [Texas Instruments](#) Information

## Optimize Your Supply Chain with WIN SOURCE Solutions

-  Global Sourcing Solution
-  Obsolete Management
-  Cost Control Management
-  Shortage Management
-  Alternative Solution
-  Excess Inventory Management