# THE DATASHEET OF

# TMS320C6454BCTZ8

# TMS320C6454
# Fixed-Point Digital Signal Processor

**Check for Samples: TMS320C6454**

## 1 Features

- High-Performance Fixed-Point DSP (C6454)
  - 1.39-, 1.17-, and 1-ns Instruction Cycle Time
  - 720-MHz, 850-MHz, and 1-GHz Clock Rate
  - Eight 32-Bit Instructions/Cycle
  - 8000 MIPS/MMACS (16-Bits)
  - Commercial Temperature [0°C to 90°C]
  - Extended Temperature [-40°C to 105°C]
- TMS320C64x+™ DSP Core
  - Dedicated SPLOOP Instruction
  - Compact Instructions (16-Bit)
  - Instruction Set Enhancements
  - Exception Handling
- TMS320C64x+ Megamodule L1/L2 Memory Architecture:
  - 256K-Bit (32K-Byte) L1P Program Cache [Direct Mapped]
  - 256K-Bit (32K-Byte) L1D Data Cache [2-Way Set-Associative]
  - 8M-Bit (1048K-Byte) L2 Unified Mapped RAM/Cache [Flexible Allocation]
  - 256K-Bit (32K-Byte) L2 ROM
  - Time Stamp Counter
- Endianess: Little Endian, Big Endian
- 64-Bit External Memory Interface (EMIFA)
  - Glueless Interface to Asynchronous Memories (SRAM, Flash, and EEPROM) and Synchronous Memories (SBSRAM, ZBT SRAM)
  - Supports Interface to Standard Sync Devices and Custom Logic (FPGA, CPLD, ASICs, etc.)
  - 32M-Byte Total Addressable External Memory Space
- DDR2 Memory Controller
  - Interfaces to DDR2-533 SDRAM
  - 32-Bit/16-Bit, 533-MHz (data rate) Bus
  - 512M-Byte Total Addressable External Memory Space
- EDMA3 Controller (64 Independent Channels)
- 32-/16-Bit Host-Port Interface (HPI)
- 32-Bit 33-/66-MHz, 3.3-V Peripheral Component Interconnect (PCI) Master/Slave Interface Conforms to *PCI Local Bus Specification* (v2.3)
- One Inter-Integrated Circuit ($I^2C$) Bus
- Two McBSPs
- 10/100/1000 Mb/s Ethernet MAC (EMAC)
  - IEEE 802.3 Compliant
  - Supports Multiple Media Independent Interfaces (MII, GMII, RMII, and RGMII)
  - 8 Independent Transmit (TX) and 8 Independent Receive (RX) Channels
- Two 64-Bit General-Purpose Timers, Configurable as Four 32-Bit Timers
- 16 General-Purpose I/O (GPIO) Pins
- System PLL and PLL Controller
- Secondary PLL and PLL Controller, Dedicated to EMAC and DDR2 Memory Controller
- Advanced Event Triggering (AET) Compatible
- Trace-Enabled Device
- IEEE-1149.1 (JTAG™) Boundary-Scan-Compatible
- 697-Pin Ball Grid Array (BGA) Package (CTZ, GTZ, or ZTZ Suffix), 0.8-mm Ball Pitch
- 0.09-µm/7-Level Cu Metal Process (CMOS)
- 3.3-/1.8-/1.5-V I/Os, 1.25-/1.2-V Internal
- Pin-Compatible with the TMS320C6455 Fixed-Point Digital Signal Processor

## 1.1 CTZ/GTZ/ZTZ BGA Package (Bottom View)

Figure 1-1 shows the TMS320C6454 device 697-pin ball grid array package (bottom view).

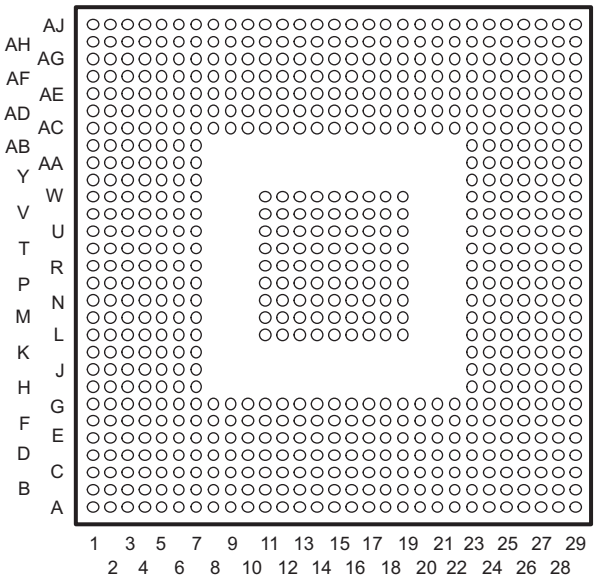



**Figure 1-1. CTZ/GTZ/ZTZ BGA Package (Bottom View)**

## 1.2 Description

The TMS320C64x+™ DSPs (including the TMS320C6454 device) are the highest-performance fixed-point DSP generation in the TMS320C6000™ DSP platform. The C6454 device is based on the third-generation high-performance, advanced VelociTI™ very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI), making these DSPs an excellent choice for applications including video and telecom infrastructure, imaging/medical, and wireless infrastructure (WI). The C64x+™ devices are upward code-compatible from previous devices that are part of the C6000™ DSP platform.

The C6454 device offers a lower cost pin-compatible migration path for C6455 customers who don't need the 2MB of the C6455 or the high-speed interconnect provided by Serial RapidIO. The C6454 device also provides an excellent migration path for existing C6414/6415/6416 customers who require C6454 advanced peripherals; DDR2 at 533 MHz provides 2x performance boost over older SDRAM interface, gigabit Ethernet provides low-cost high-performance ubiquitous packet interface, and 66-MHz PCI (revision 2.3 complaint) provides legacy high-bandwidth interconnect.

Based on 90-nm process technology and with performance of up to 8000 million instructions per second (MIPS) [or 8000 16-bit MMACs per cycle] at a 1-GHz clock rate, the C6454 device offers cost-effective solutions to high-performance DSP programming challenges. The C6454 DSP possesses the operational flexibility of high-speed controllers and the numerical capability of array processors.

The C64x+ DSP core employs eight functional units, two register files, and two data paths. Like the earlier C6000 devices, two of these eight functional units are multipliers or .M units. Each C64x+ .M unit doubles the multiply throughput versus the C64x core by performing four 16-bit x 16-bit multiply-accumulates (MACs) every clock cycle. Thus, eight 16-bit x 16-bit MACs can be executed every cycle on the C64x+ core. At a 1-GHz clock rate, this means 8000 16-bit MMACs can occur every second. Moreover, each multiplier on the C64x+ core can compute one 32-bit x 32-bit MAC or four 8-bit x 8-bit MACs every clock cycle.

The C6454 DSP integrates a large amount of on-chip memory organized as a two-level memory system. The level-1 (L1) program and data memories on the C6454 device are 32KB each. This memory can be configured as mapped RAM, cache, or some combination of the two. When configured as cache, L1 program (L1P) is a direct mapped cache where as L1 data (L1D) is a two-way set associative cache. The level-2 (L2) memory is shared between program and data space and is 1048KB in size. L2 memory can also be configured as mapped RAM, cache, or some combination of the two. The C64x+ Megamodule also has a 32-bit peripheral configuration (CFG) port, an internal DMA (IDMA) controller, a system component with reset/boot control, interrupt/exception control, a power-down control, and a free-running 32-bit timer for time stamp.

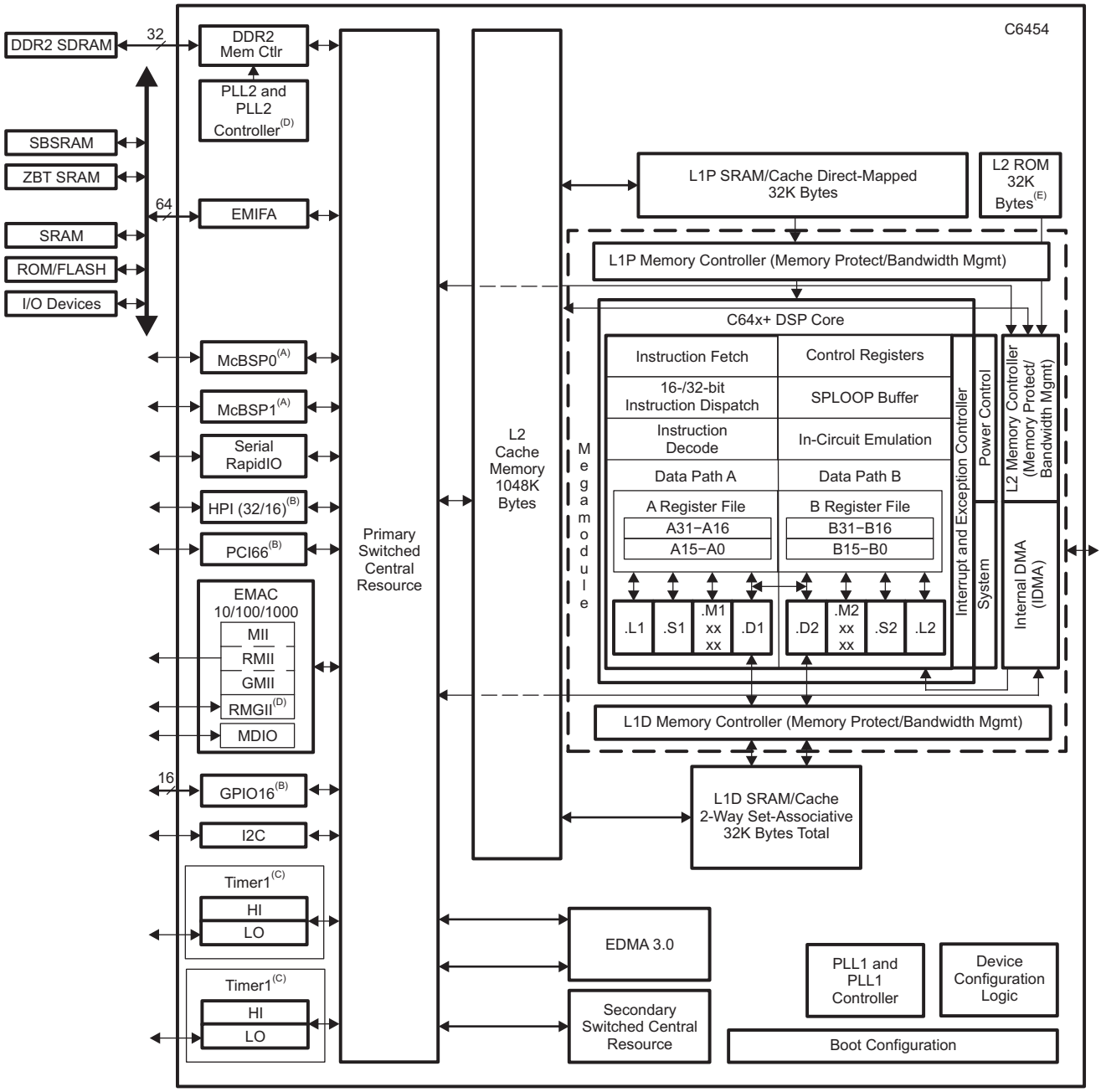
The peripheral set includes: an inter-integrated circuit bus module (I2C); two multichannel buffered serial ports (McBSPs); a user-configurable 16-bit or 32-bit host-port interface (HPI16/HPI32); a peripheral component interconnect (PCI); a 16-pin general-purpose input/output port (GPIO) with programmable interrupt/event generation modes; an 10/100/1000 Ethernet media access controller (EMAC), which provides an efficient interface between the C6454 DSP core processor and the network; a management data input/output (MDIO) module (also part of the EMAC) that continuously polls all 32 MDIO addresses in order to enumerate all PHY devices in the system; a glueless external memory interface (64-bit EMIFA), which is capable of interfacing to synchronous and asynchronous peripherals; and a 32-bit DDR2 SDRAM interface.

The I2C ports on the C6454 device allows the DSP to easily control peripheral devices and communicate with a host processor. In addition, the standard multichannel buffered serial port (McBSP) may be used to communicate with serial peripheral interface (SPI) mode peripheral devices.

The C6454 DSP has a complete set of development tools which includes: a new C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows® debugger interface for visibility into source code execution.

## 1.3 Functional Block Diagram

Figure 1-2 shows the functional block diagram of the C6454 device.




A. McBSPs: Framing Chips - H.100, MVIP, SCSA, T1, E1; AC97 Devices; SPI Devices; Codecs.
B. The PCI peripheral pins are muxed with some of the HPI peripheral pins. For more detailed information, see the *Device Configuration* section.
C. Each of the TIMER peripherals (TIMER1 and TIMER0) is configurable as a 64-bit general-purpose timer, dual 32-bit general-purpose timers, or a watchdog timer.
D. The PLL2 controller also generates clocks for the EMAC.
E. When accessing the internal ROM of the DSP, the CPU frequency must always be less than 750 MHz.

**Figure 1-2. Functional Block Diagram**

**1 Features** ..... 1
1.1 CTZ/GTZ/ZTZ BGA Package (Bottom View) ..... 2
1.2 Description ..... 2
1.3 Functional Block Diagram ..... 4
**Revision History** ..... 6
**2 Device Overview** ..... 7
2.1 Device Characteristics ..... 7
2.2 CPU (DSP Core) Description ..... 8
2.3 Memory Map Summary ..... 11
2.4 Boot Sequence ..... 13
2.5 Pin Assignments ..... 15
2.6 Signal Groups Description ..... 19
2.7 Terminal Functions ..... 24
2.8 Development ..... 48
**3 Device Configuration** ..... 52
3.1 Device Configuration at Device Reset ..... 52
3.2 Peripheral Configuration at Device Reset ..... 54
3.3 Peripheral Selection After Device Reset ..... 55
3.4 Device State Control Registers ..... 57
3.5 Device Status Register Description ..... 67
3.6 JTAG ID (JTAGID) Register Description ..... 69
3.7 Pullup/Pulldown Resistors ..... 70
3.8 Configuration Examples ..... 70
**4 System Interconnect** ..... 73
4.1 Internal Buses, Bridges, and Switch Fabrics ..... 73
4.2 Data Switch Fabric Connections ..... 74
4.3 Configuration Switch Fabric ..... 76
4.4 Bus Priorities ..... 78
**5 C64x+ Megamodule** ..... 79
5.1 Memory Architecture ..... 79
5.2 Memory Protection ..... 81
5.3 Bandwidth Management ..... 82
5.4 Power-Down Control ..... 83
5.5 Megamodule Resets ..... 83
5.6 Megamodule Revision ..... 84
5.7 C64x+ Megamodule Register Descriptions ..... 85
**6 Device Operating Conditions** ..... 93
6.1 Absolute Maximum Ratings Over Operating Case Temperature Range (Unless Otherwise Noted) ..... 93
6.2 Recommended Operating Conditions ..... 93
6.3 Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Case Temperature (Unless Otherwise Noted) ..... 95
**7 C64x+ Peripheral Information and Electrical Specifications** ..... 97
7.1 Parameter Information ..... 97
7.2 Recommended Clock and Control Signal Transition Behavior ..... 99
7.3 Power Supplies ..... 99
7.4 Enhanced Direct Memory Access (EDMA3) Controller ..... 101
7.5 Interrupts ..... 116
7.6 Reset Controller ..... 120
7.7 PLL1 and PLL1 Controller ..... 128
7.8 PLL2 and PLL2 Controller ..... 143
7.9 DDR2 Memory Controller ..... 152
7.10 External Memory Interface A (EMIFA) ..... 154
7.11 I2C Peripheral ..... 165
7.12 Host-Port Interface (HPI) Peripheral ..... 170
7.13 Multichannel Buffered Serial Port (McBSP) ..... 181
7.14 Ethernet MAC (EMAC) ..... 195
7.15 Timers ..... 213
7.16 Peripheral Component Interconnect (PCI) ..... 215
7.17 General-Purpose Input/Output (GPIO) ..... 222
7.18 Emulation Features and Capability ..... 224
**8 Mechanical Data** ..... 226
8.1 Thermal Data ..... 226
8.2 Packaging Information ..... 226

## Revision History

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

This data manual revision history highlights the technical changes made to the document in this revision.

**Scope:** Applicable updates to the C64x device family, specifically relating to the TMS320C6454 device, have been incorporated.

### C6454 DSP Revision History

| SEE | ADDITIONS/MODIFICATIONS/DELETIONS |
|---|---|
| Section 7.7.1.1 | Internal Clocks and Maximum Operating Frequencies:<br>Modified values for SYSCLK2 and SYSCLK3 in fifth paragraph |

# 2 Device Overview

## 2.1 Device Characteristics

Table 2-1, provides an overview of the C6454 DSP. The tables show significant features of the C6454 device, including the capacity of on-chip RAM, the peripherals, the CPU frequency, and the package type with pin count.

**Table 2-1. Characteristics of the C6454 Processor**

| HARDWARE FEATURES | | C6454 |
|---|---|---|
| Peripherals<br>Not all peripherals pins are available at the same time (For more detail, see Section 3, *Device Configuration*). | EMIFA (64-bit bus width)<br>(clock source = AECLKIN or SYSCLK4) | 1 |
| | DDR2 Memory Controller (32-bit bus width) [1.8 V I/O]<br>(clock source = CLKIN2) | 1 |
| | EDMA3 (64 independent channels) [CPU/3 clock rate] | 1 |
| | I2C | 1 |
| | HPI (32- or 16-bit user selectable) | 1 (HPI16 or HPI32) |
| | PCI (32-bit), [66-MHz or 33-MHz] | 1 (PCI66 or PCI33) |
| | McBSPs (internal CPU/6 or external clock source up to 100 Mbps) | 2 |
| | 10/100/1000 Ethernet MAC (EMAC) | 1 |
| | Management Data Input/Output (MDIO) | 1 |
| | 64-Bit Timers (Configurable)<br>(internal clock source = CPU/6 clock frequency) | 2 64-bit ***or*** 4 32-bit |
| | General-Purpose Input/Output Port (GPIO) | 16 |
| On-Chip Memory | Size (Bytes) | 1144K |
| | Organization | 32K-Byte (32KB) L1 Program Memory Controller [SRAM/Cache]<br>32KB Data Memory Controller [SRAM/Cache]<br>1024KB L2 Unified Memory/Cache<br>32KB L2 ROM |
| C64x+ Megamodule Revision ID | Megamodule Revision ID Register (address location: 0181 2000h) | See Section 5.6, *Megamodule Revision* |
| JTAG BSDL_ID | JTAGID register (address location: 0x02A80008) | See Section 3.6, *JTAG ID (JTAGID) Register Description* |
| Frequency | MHz | 720, 850, and 1000 (1 GHz) |
| Cycle Time | ns | 1.39 ns (C6454-720), 1.17 ns (C6454-850),<br>1 ns (C6454 A-1000, -1000) [1-GHz CPU](1) |
| Voltage | Core (V) | 1.25 V (A-1000/-1000)<br>1.2 V (-850/-720) |
| | I/O (V) | 1.5/1.8 [EMAC RGMII], and<br>1.8 and 3.3 V [I/O Supply Voltage] |
| PLL1 and PLL1 Controller Options | CLKIN1 frequency multiplier | Bypass (x1), x15, x20, x25, x30, x32 |
| PLL2 | CLKIN2 frequency multiplier<br>[DDR2 Memory Controller and EMAC support only] | x20 |
| BGA Package | 24 x 24 mm | 697-Pin Flip-Chip Plastic BGA (CTZ)<br>697-Pin Plastic BGA (GTZ)<br>697-Pin Flip-Chip Plastic BGA (ZTZ) |
| Process Technology | µm | 0.09 µm |
| Product Status(2) | Product Preview (PP), Advance Information (AI),<br>or Production Data (PD) | PD |

(1) The extended temperature device's (A-1000) electrical characteristics and ac timings are the same as those for the corresponding commercial temperature devices (-1000).
(2) PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**Table 2-1. Characteristics of the C6454 Processor (continued)**

| HARDWARE FEATURES | | C6454 |
|---|---|---|
| Device Part Numbers | (For more details on the C64x+™ DSP part numbering, see Figure 2-12) | TMS320C6454CTZ7/GTZ7/ZTZ7<br>TMS320C6454CTZ8/GTZ8/ZTZ8<br>TMS320C6454CTZ/GTZ/ZTZ |

## 2.2 CPU (DSP Core) Description

The C64x+ Central Processing Unit (CPU) consists of eight functional units, two register files, and two data paths as shown in Figure 2-1. 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.

The C64x+ CPU extends the performance of the C64x core through enhancements and new features.

Each C64x+ .M unit can perform one of the following each clock cycle: one 32 x 32 bit multiply, two 16 x 16 bit multiplies, two 16 x 32 bit multiplies, 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 for 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 audio and other 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.

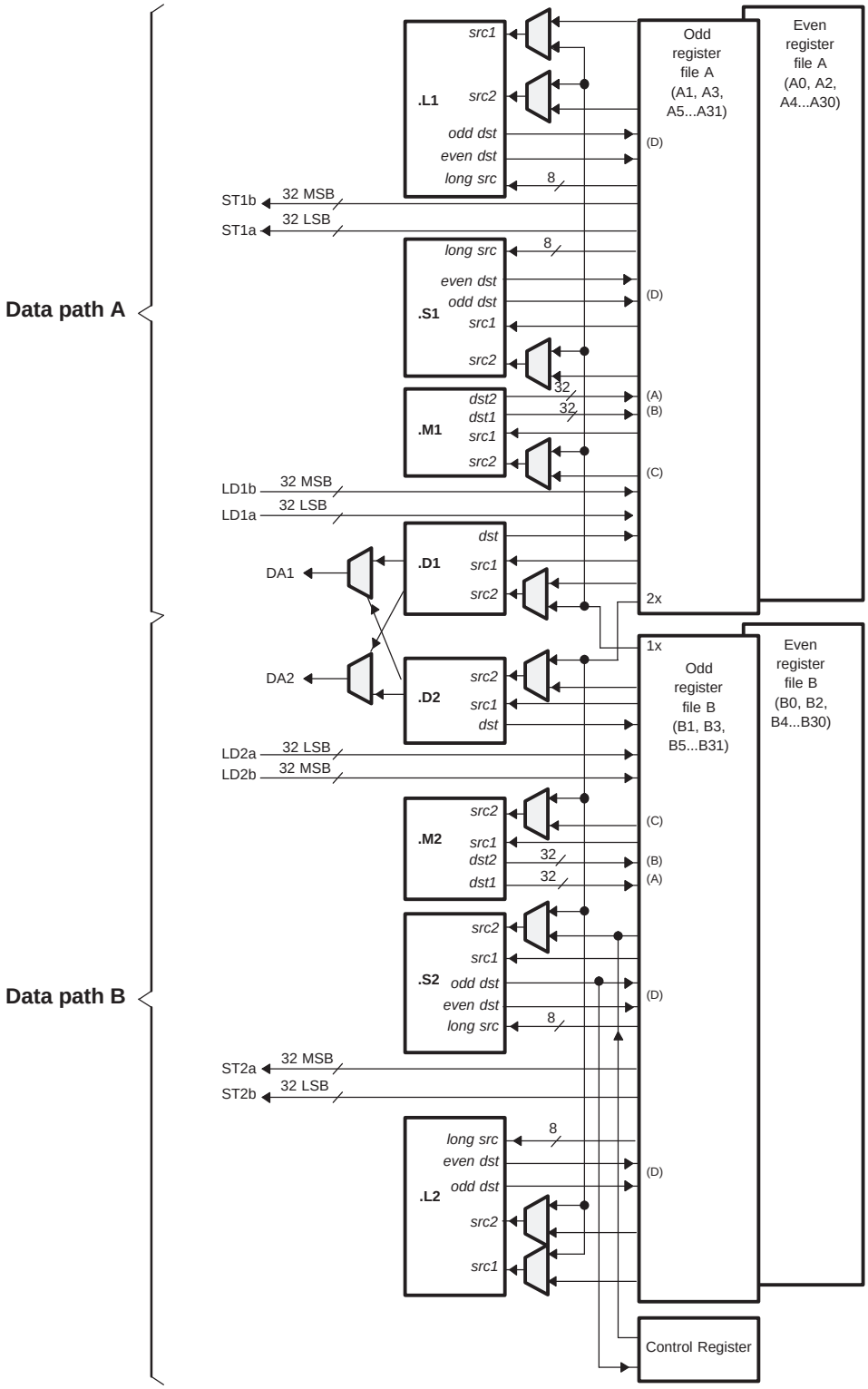
The C64x+ core enhances the .S unit in several ways. In the C64x core, dual 16-bit MIN2 and MAX2 comparisons were only available on the .L units. On the C64x+ core they are also available on the .S unit which increases the performance of algorithms that do searching and sorting. Finally, to increase 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 C64x+ compiler can restrict the code to use certain registers in the register file. This compression is performed by the code generation tools.
- **Instruction Set Enhancements** - As noted above, there are new instructions such as 32-bit multiplications, complex multiplications, packing, sorting, bit manipulation, and 32-bit Galois field multiplication.
- **Exception Handling** - Intended to aid the programmer in isolating bugs. The C64x+ 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 C64x+ CPU and its enhancements over the C64x architecture, see the following documents:

- *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* (literature number SPRU732)
- *TMS320C64x+ DSP Cache User's Guide* (literature number SPRU862)
- *TMS320C64x+ Megamodule Reference Guide* (literature number SPRU871)
- *TMS320C6455 Technical Reference* (literature number SPRU965)
- *TMS320C64x to TMS320C64x+ CPU Migration Guide* (literature number SPRAA84)




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 2-1. TMS320C64x+™ CPU (DSP Core) Data Paths**

## 2.3 Memory Map Summary

Table 2-2 shows the memory map address ranges of the C6454 device. The external memory configuration register address ranges in the C6454 device begin at the hex address location 0x7000 0000 for EMIFA and hex address location 0x7800 0000 for DDR2 Memory Controller.

**Table 2-2. C6454 Memory Map Summary**

| MEMORY BLOCK DESCRIPTION | BLOCK SIZE (BYTES) | HEX ADDRESS RANGE |
|---|---|---|
| Reserved | 1024K | 0000 0000 - 000F FFFF |
| Internal ROM | 32K | 0010 0000 - 0010 7FFF |
| Reserved | 7M - 32K | 0010 8000 - 007F FFFF |
| Internal RAM (L2) [L2 SRAM] | 1M | 0080 0000 - 008F FFFF |
| Reserved | 5M | 0090 0000 - 00DF FFFF |
| L1P SRAM | 32K | 00E0 0000 - 00E0 7FFF |
| Reserved | 1M - 32K | 00E0 8000 - 00EF FFFF |
| L1D SRAM | 32K | 00F0 0000 - 00F0 7FFF |
| Reserved | 1M - 32K | 00F0 8000 - 00FF FFFF |
| Reserved | 8M | 0100 0000 - 017F FFFF |
| C64x+ Megamodule Registers | 4M | 0180 0000 - 01BF FFFF |
| Reserved | 12.5M | 01C0 0000 - 0287 FFFF |
| HPI Control Registers | 256K | 0288 0000 - 028B FFFF |
| McBSP 0 Registers | 256K | 028C 0000 - 028F FFFF |
| McBSP 1 Registers | 256K | 0290 0000 - 0293 FFFF |
| Timer 0 Registers | 256K | 0294 0000 - 0297 FFFF |
| Timer 1 Registers | 128K | 0298 0000 - 0299 FFFF |
| PLL1 Controller (including Reset Controller) Registers | 512 | 029A 0000 - 029A 01FF |
| Reserved | 256K - 512 | 029A 0200 - 029B FFFF |
| PLL2 Controller Registers | 512 | 029C 0000 - 029C 01FF |
| Reserved | 64K | 029C 0200 - 029C FFFF |
| EDMA3 Channel Controller Registers | 32K | 02A0 0000 - 02A0 7FFF |
| Reserved | 96K | 02A0 8000 - 02A1 FFFF |
| EDMA3 Transfer Controller 0 Registers | 32K | 02A2 0000 - 02A2 7FFF |
| EDMA3 Transfer Controller 1 Registers | 32K | 02A2 8000 - 02A2 FFFF |
| EDMA3 Transfer Controller 2 Registers | 32K | 02A3 0000 - 02A3 7FFF |
| EDMA3 Transfer Controller 3 Registers | 32K | 02A3 8000 - 02A3 FFFF |
| Reserved | 256K | 02A4 0000 - 02A7 FFFF |
| Chip-Level Registers | 256K | 02A8 0000 - 02AB FFFF |
| Device State Control Registers | 256K | 02AC 0000 - 02AF FFFF |
| GPIO Registers | 16K | 02B0 0000 - 02B0 3FFF |
| I2C Data and Control Registers | 256K | 02B0 4000 - 02B3 FFFF |
| Reserved | 720K | 02B4 0000 - 02BF FFFF |
| PCI Control Registers | 256K | 02C0 0000 - 02C3 FFFF |
| Reserved | 256K | 02C4 0000 - 02C7 FFFF |
| EMAC Control | 4K | 02C8 0000 - 02C8 0FFF |
| EMAC Control Module Registers | 2K | 02C8 1000 - 02C8 17FF |
| MDIO Control Registers | 2K | 02C8 1800 - 02C8 1FFF |
| EMAC Descriptor Memory | 8K | 02C8 2000 - 02C8 3FFF |
| Reserved | 496K | 02C8 4000 - 02CF FFFF |
| Reserved | 220M | 02D0 0000 - 0FFF FFFF |
| Reserved | 256M | 1000 0000 - 1FFF FFFF |

**Table 2-2. C6454 Memory Map Summary (continued)**

| MEMORY BLOCK DESCRIPTION | BLOCK SIZE (BYTES) | HEX ADDRESS RANGE |
|---|---|---|
| Reserved | 256M | 2000 0000 - 2FFF FFFF |
| McBSP 0 Data | 256 | 3000 0000 - 3000 00FF |
| Reserved | 64M - 256 | 3000 0100 - 33FF FFFF |
| McBSP 1 Data | 256 | 3400 0000 - 3400 00FF |
| Reserved | 64M - 256 | 3400 0100 - 37FF FFFF |
| Reserved | 64M | 3800 0000 - 3BFF FFFF |
| Reserved | 2K | 3C00 0000 - 3C00 07FF |
| Reserved | 16M - 2K | 3C00 0800 - 3CFF FFFF |
| Reserved | 48M | 3D00 0000 - 3FFF FFFF |
| PCI External Memory Space | 256M | 4000 0000 - 4FFF FFFF |
| Reserved | 256M | 5000 0000 - 5FFF FFFF |
| Reserved | 256M | 6000 0000 - 6FFF FFFF |
| EMIFA (EMIF64) Configuration Registers | 128M | 7000 0000 - 77FF FFFF |
| DDR2 Memory Controller Configuration Registers | 128M | 7800 0000 - 7FFF FFFF |
| Reserved | 256M | 8000 0000 - 8FFF FFFF |
| Reserved | 256M | 9000 0000 - 9FFF FFFF |
| EMIFA CE2 - SBSRAM/Async(1) | 8M | A000 0000 - A07F FFFF |
| Reserved | 256M - 8M | A080 0000 - AFFF FFFF |
| EMIFA CE3 - SBSRAM/Async(1) | 8M | B000 0000 - B07F FFFF |
| Reserved | 256M - 8M | B080 0000 - BFFF FFFF |
| EMIFA CE4 - SBSRAM/Async(1) | 8M | C000 0000 - C07F FFFF |
| Reserved | 256M - 8M | C080 0000 - CFFF FFFF |
| EMIFA CE5 - SBSRAM/Async(1) | 8M | D000 0000 - D07F FFFF |
| Reserved | 256M - 8M | D080 0000 - DFFF FFFF |
| DDR2 Memory Controller CE0 - DDR2 SDRAM | 512M | E000 0000 - FFFF FFFF |

(1) The EMIFA $\overline{CE0}$ and $\overline{CE1}$ are ***not*** functionally supported on the C6454 device and, therefore, are ***not*** pinned out.

## 2.4 Boot Sequence

The boot sequence is a process by which the DSP's internal memory is loaded with program and data sections and the DSP's internal registers are programmed with predetermined values. The boot sequence is started automatically after each power-on reset, warm reset, and system reset. For more details on the initiators of these resets, see Section 7.6, *Reset Controller*.

There are several methods by which the memory and register initialization can take place. Each of these methods is referred to as a boot mode. The boot mode to be used is selected at reset through the BOOTMODE[3:0] pins.

Each boot mode can be classified as a hardware boot mode or as a software boot mode. Software boot modes require the use of the on-chip bootloader. The bootloader is DSP code that transfers application code from an external source into internal or external program memory after the DSP is taken out of reset. The bootloader is permanently stored in the internal ROM of the DSP starting at byte address 0010 0000h. Hardware boot modes are carried out by the boot configuration logic. The boot configuration logic is actual hardware that does not require the execution of DSP code. Section 2.4.1, *Boot Modes Supported*, describes each boot mode in more detail.

When accessing the internal ROM of the DSP, the CPU frequency must always be less than 750 MHz. Therefore, when using a software boot mode, care must be taken such that the CPU frequency does not exceed 750 MHz at any point during the boot sequence. After the boot sequence has completed, the CPU frequency can be programmed to the frequency required by the application.

### 2.4.1 Boot Modes Supported

The C6454 device has five boot modes:

- No boot (BOOTMODE[3:0] = 0000b)

  With no boot, the CPU executes directly from the internal L2 SRAM located at address 0x80 0000. Note: device operations is undefined if invalid code is located at address 0x80 0000. This boot mode is a hardware boot mode.

- Host boot (BOOTMODE[3:0] = 0001b and BOOTMODE[3:0] = 0111b)

  If host boot is selected, after reset, the CPU is internally "stalled" while the remainder of the device is released. During this period, an external host can initialize the CPU's memory space as necessary through Host Port Interface (HPI) or the Peripheral Component Interconnect (PCI) interface. Internal configuration registers, such as those that control the EMIF can also be initialized by the host with two exceptions: Device State Control registers (Section 3.4), PLL1 and PLL2 Controller registers (Section 7.7 and Section 7.8) cannot be accessed through any host interface, including HPI and PCI.

  Once the host is finished with all necessary initialization, it must generate a DSP interrupt (DSPINT) to complete the boot process. This transition causes boot configuration logic to bring the CPU out of the "stalled" state. The CPU then begins execution from the internal L2 SRAM located at 0x80 0000. Note that the DSP interrupt is registered in bit 0 (channel 0) of the EDMA Event Register (ER). This event must be cleared by software before triggering transfers on DMA channel 0.

  All memory, with the exceptions previously described, may be written to and read by the host. This allows for the host to verify what it sends to the DSP if required. After the CPU is out of the "stalled" state, the CPU needs to clear the DSPINT, otherwise, no more DSPINTs can be received.

  As previously mentioned, for the C6454 device, the Host Port Interface (HPI) and the Peripheral Component Interconnect (PCI) interface can be used for host boot. To use the HPI for host boot, the PCI_EN pin (Y29) must be low [default] (enabling the HPI peripheral) and BOOTMODE[3:0] must be set to 0001b at device reset. Conversely, to use the PCI interface for host boot, the PCI_EN pin (Y29) must be high (enabling the PCI peripheral) and BOOTMODE[3:0] must be set to 0111b at device reset. For the HPI host boot, the DSP interrupt can be generated through the use of the DSPINT bit in the HPI Control (HPIC) register.

  For the HPI host boot, the CPU is actually held in reset until a DSP interrupt is generated by the host. The DSP interrupt can be generated through the use of the DSPINT bit in the HPI Control (HPIC) register. Since the CPU is held in reset during HPI host boot, it will not respond to emulation software

such as Code Composer Studio.

For the PCI host boot, the CPU is out of reset, but it executes an IDLE instruction until a DSP interrupt is generated by the host. The host can generate a DSP interrupt through the PCI peripheral by setting the DSPINT bit in the Back-End Application Interrupt Enable Set Register (PCIBINTSET) and the Status Set Register (PCISTATSET).

Note that the HPI host boot is a hardware boot mode while the PCI host boot is a software boot mode.

If PCI boot is selected, the on-chip bootloader configures the PLL1 Controller such that CLKIN1 is multiplied by 15. More specifically, PLLM is set to 0Eh (x15) and RATIO is set to 0 (÷1) in the PLL1 Multiplier Control Register (PLLM) and PLL1 Pre-Divider Register (PREDIV), respectively. The CLKIN1 frequency must not be greater than 50 MHz so that the maximum speed of the internal ROM, 750 MHz, is not violated. The CFGGP[2:0] pins must be set to 000b during reset for proper operation of the PCI boot mode.

As mentioned previously, a DSP interrupt must be generated at the end of the host boot process to begin execution of the loaded application. Since the DSP interrupt generated by the HPI and PCI is mapped to the EDMA event DSP_EVT (DMA channel 0), it will get recorded in bit 0 of the EDMA Event Register (ER). This event must be cleared by software before triggering transfers on DMA channel 0.

- EMIFA 8-bit ROM boot (BOOTMODE[3:0] = 0100b)

  After reset, the device will begin executing software out of an Asynchronous 8-bit ROM located in EMIFA CE3 space using the default settings in the EMIFA registers. This boot mode is a hardware boot mode.

- Master I2C boot (BOOTMODE[3:0] = 0101b)

  After reset, the DSP can act as a master to the I2C bus and copy data from an I2C EEPROM or a device acting as an I2C slave to the DSP using a predefined boot table format. The destination address and length are contained within the boot table. This boot mode is a software boot mode.

- Slave I2C boot (BOOTMODE[3:0] = 0110b)

  A Slave I2C boot is also implemented, which programs the DSP as an I2C Slave and simply waits for a Master to send data using a standard boot table format.

  Using the Slave I2C boot, a single DSP or a device acting as an I2C Master can simultaneously boot multiple slave DSPs connected to the same I2C bus. Note that the Master DSP may require booting via an I2C EEPROM before acting as a Master and booting other DSPs.

  The Slave I2C boot is a software boot mode.

### 2.4.2 2nd-Level Bootloaders

Any of the boot modes can be used to download a 2nd-level bootloader. A 2nd-level bootloader allows for any level of customization to current boot methods as well as definition of a completely customized boot. TI offers a few 2nd-level bootloaders, such as an EMAC bootloader, which can be loaded using the Master I2C boot.

## 2.5 Pin Assignments

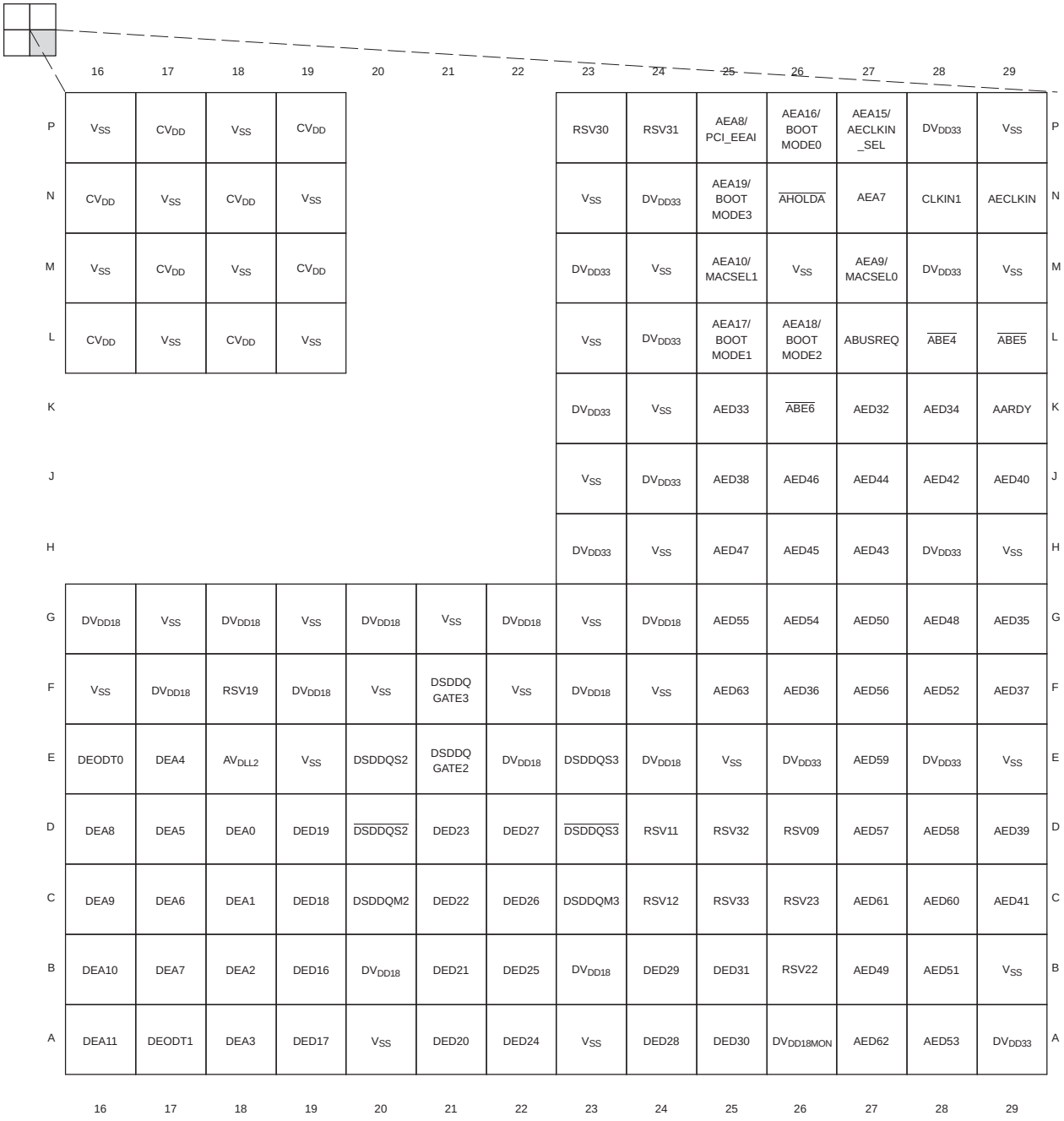
### 2.5.1 Pin Map

Figure 2-2 through Figure 2-5 show the C6454 device pin assigments in four quadrants (A, B, C, and D).

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AJ | $DV_{DD33}$ | GP[5] | FSX0 | CLKS | DR0 | TINPL1 | $DV_{DD33}$ | $V_{SS}$ | TCK | TMS | RSV26 | RSV40 | SYSCLK4/ GP[1] | $V_{SS}$ | $DV_{DD33}$ | AJ |
| AH | $V_{SS}$ | GP[4] | FSR0 | NMI | DR1/ GP[8] | TINPL0 | $\overline{TRST}$ | TDO | TDI | EMU17 | RSV27 | EMU16 | EMU9 | $DV_{DD33}$ | $V_{SS}$ | AH |
| AG | CLKR0 | GP[7] | GP[6] | FSX1/ GP[11] | DX1/ GP[9] | CLKX0 | TOUTL1 | EMU6 | EMU2 | RSV38 | RSV39 | $DV_{DD33}$ | $V_{SS}$ | $\overline{RESET}$ | RSV46 | AG |
| AF | $DV_{DD33}$ | $V_{SS}$ | HD11/ AD11 | CLKR1/ GP[0] | CLKX1/ GP[3] | DX0 | EMU0 | TOUTL0 | EMU4 | EMU3 | EMU8 | EMU7 | EMU14 | $\overline{POR}$ | RSV45 | AF |
| AE | HD22/ AD22 | HD0/ AD0 | HD10/ AD10 | $V_{SS}$ | FSR1/ GP[10] | $DV_{DD33}$ | $V_{SS}$ | $DV_{DD33}$ | EMU15 | EMU12 | EMU1 | EMU5 | EMU18 | $\overline{RESETSTAT}$ | $DV_{DD33}$ | AE |
| AD | HD21/ AD21 | HD25/ AD25 | HD5/ AD5 | HD3/ AD3 | $DV_{DD33}$ | $V_{SS}$ | $DV_{DD33}$ | EMU13 | RSV37 | EMU10 | RSV36 | EMU11 | $V_{SS}$ | $DV_{DD33}$ | $V_{SS}$ | AD |
| AC | HD19/ AD19 | HD13/ AD13 | HD23/ AD23 | HD29/ AD29 | HD27/ AD27 | $DV_{DD33}$ | $V_{SS}$ | $V_{SS}$ | $DV_{DD33}$ | $V_{SS}$ | $DV_{DD33}$ | $V_{SS}$ | $DV_{DD33}$ | $V_{SS}$ | RSV64 | AC |
| AB | HD17/ AD17 | HD15/ AD15 | HD9/ AD9 | HD7/ AD7 | HD1/ AD1 | $V_{SS}$ | $DV_{DD33}$ | | | | | | | | | AB |
| AA | $DV_{DD33}$ | $V_{SS}$ | HD31/ AD31 | HD28/ AD28 | HD30/ AD30 | $DV_{DD33}$ | $V_{SS}$ | | | | | | | | | AA |
| Y | HD26/ AD26 | HD18/ AD18 | HD16/ AD16 | HD6/ AD6 | HD4/ AD4 | $V_{SS}$ | $DV_{DD33}$ | | | | | | | | | Y |
| W | HD24/ AD24 | HD20/ AD20 | RSV03 | HD14/ AD14 | HD8/ AD8 | HD2/ AD2 | $V_{SS}$ | | | | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | W |
| V | $DV_{DD33}$ | $V_{SS}$ | HHWIL/ PCLK | HD12/ AD12 | RSV02 | $V_{SS}$ | $DV_{DD33}$ | | | | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | RSV68 | V |
| U | $\overline{HDS2}$/ $\overline{PCBE1}$ | $\overline{HDS1}$/ $\overline{PSERR}$ | $\overline{HINT}$/ $\overline{PFRAME}$ | HCNTL1/ $\overline{PDEVSEL}$ | HCNTL0/ $\overline{PSTOP}$ | $\overline{HCS}$/ $\overline{PPERR}$ | $V_{SS}$ | | | | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | U |
| T | RSV15 | RSV16 | $\overline{HAS}$/ PPAR | $\overline{HRDY}$/ $\overline{PIRDY}$ | HR/$\overline{W}$/ $\overline{PCBE2}$ | $V_{SS}$ | $DV_{DD33}$ | | | | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | T |
| R | $DV_{DD33}$ | $V_{SS}$ | PIDSEL | $\overline{PGNT}$/ GP[12] | $\overline{PRST}$/ GP[13] | $DV_{DD33}$ | $V_{SS}$ | | | | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | R |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |

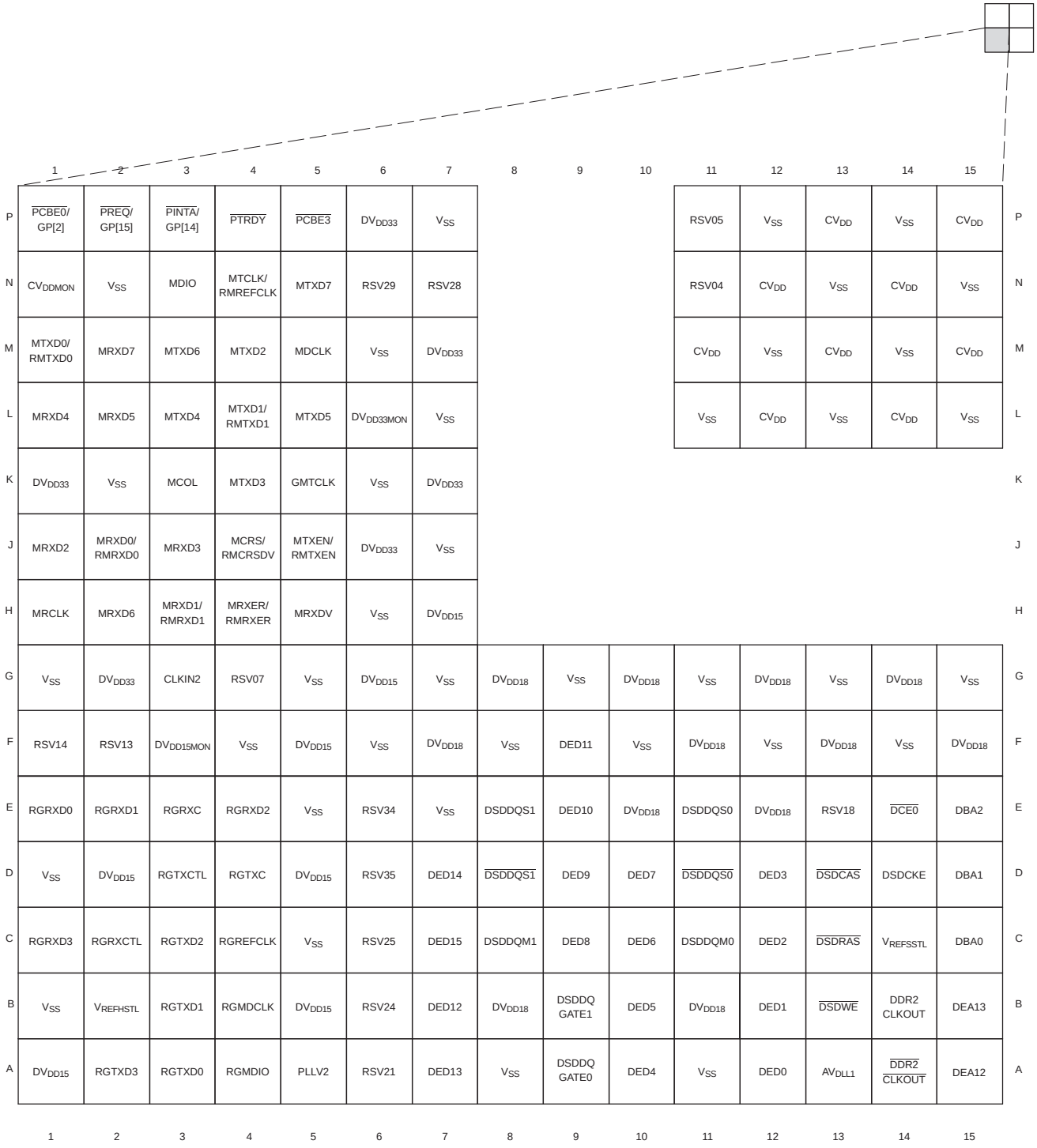
**Figure 2-2. C6454 Pin Map (Bottom View) [Quadrant A]**


**Figure 2-3. C6454 Pin Map (Bottom View) [Quadrant B]**



| | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | | | | RSV30 | RSV31 | AEA8/ PCI_EEAI | AEA16/ BOOT MODE0 | AEA15/ AECLKIN _SEL | $DV_{DD33}$ | $V_{SS}$ | P |
| N | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | | | | $V_{SS}$ | $DV_{DD33}$ | AEA19/ BOOT MODE3 | $\overline{AHOLDA}$ | AEA7 | CLKIN1 | AECLKIN | N |
| M | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | | | | $DV_{DD33}$ | $V_{SS}$ | AEA10/ MACSEL1 | $V_{SS}$ | AEA9/ MACSEL0 | $DV_{DD33}$ | $V_{SS}$ | M |
| L | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | | | | $V_{SS}$ | $DV_{DD33}$ | AEA17/ BOOT MODE1 | AEA18/ BOOT MODE2 | ABUSREQ | $\overline{ABE4}$ | $\overline{ABE5}$ | L |
| K | | | | | | | | $DV_{DD33}$ | $V_{SS}$ | AED33 | $\overline{ABE6}$ | AED32 | AED34 | AARDY | K |
| J | | | | | | | | $V_{SS}$ | $DV_{DD33}$ | AED38 | AED46 | AED44 | AED42 | AED40 | J |
| H | | | | | | | | $DV_{DD33}$ | $V_{SS}$ | AED47 | AED45 | AED43 | $DV_{DD33}$ | $V_{SS}$ | H |
| G | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | AED55 | AED54 | AED50 | AED48 | AED35 | G |
| F | $V_{SS}$ | $DV_{DD18}$ | RSV19 | $DV_{DD18}$ | $V_{SS}$ | DSDDQ GATE3 | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | AED63 | AED36 | AED56 | AED52 | AED37 | F |
| E | DEODT0 | DEA4 | $AV_{DLL2}$ | $V_{SS}$ | DSDDQS2 | DSDDQ GATE2 | $DV_{DD18}$ | DSDDQS3 | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD33}$ | AED59 | $DV_{DD33}$ | $V_{SS}$ | E |
| D | DEA8 | DEA5 | DEA0 | DED19 | $\overline{DSDDQS2}$ | DED23 | DED27 | $\overline{DSDDQS3}$ | RSV11 | RSV32 | RSV09 | AED57 | AED58 | AED39 | D |
| C | DEA9 | DEA6 | DEA1 | DED18 | DSDDQM2 | DED22 | DED26 | DSDDQM3 | RSV12 | RSV33 | RSV23 | AED61 | AED60 | AED41 | C |
| B | DEA10 | DEA7 | DEA2 | DED16 | $DV_{DD18}$ | DED21 | DED25 | $DV_{DD18}$ | DED29 | DED31 | RSV22 | AED49 | AED51 | $V_{SS}$ | B |
| A | DEA11 | DEODT1 | DEA3 | DED17 | $V_{SS}$ | DED20 | DED24 | $V_{SS}$ | DED28 | DED30 | $DV_{DD18MON}$ | AED62 | AED53 | $DV_{DD33}$ | A |
| | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | |

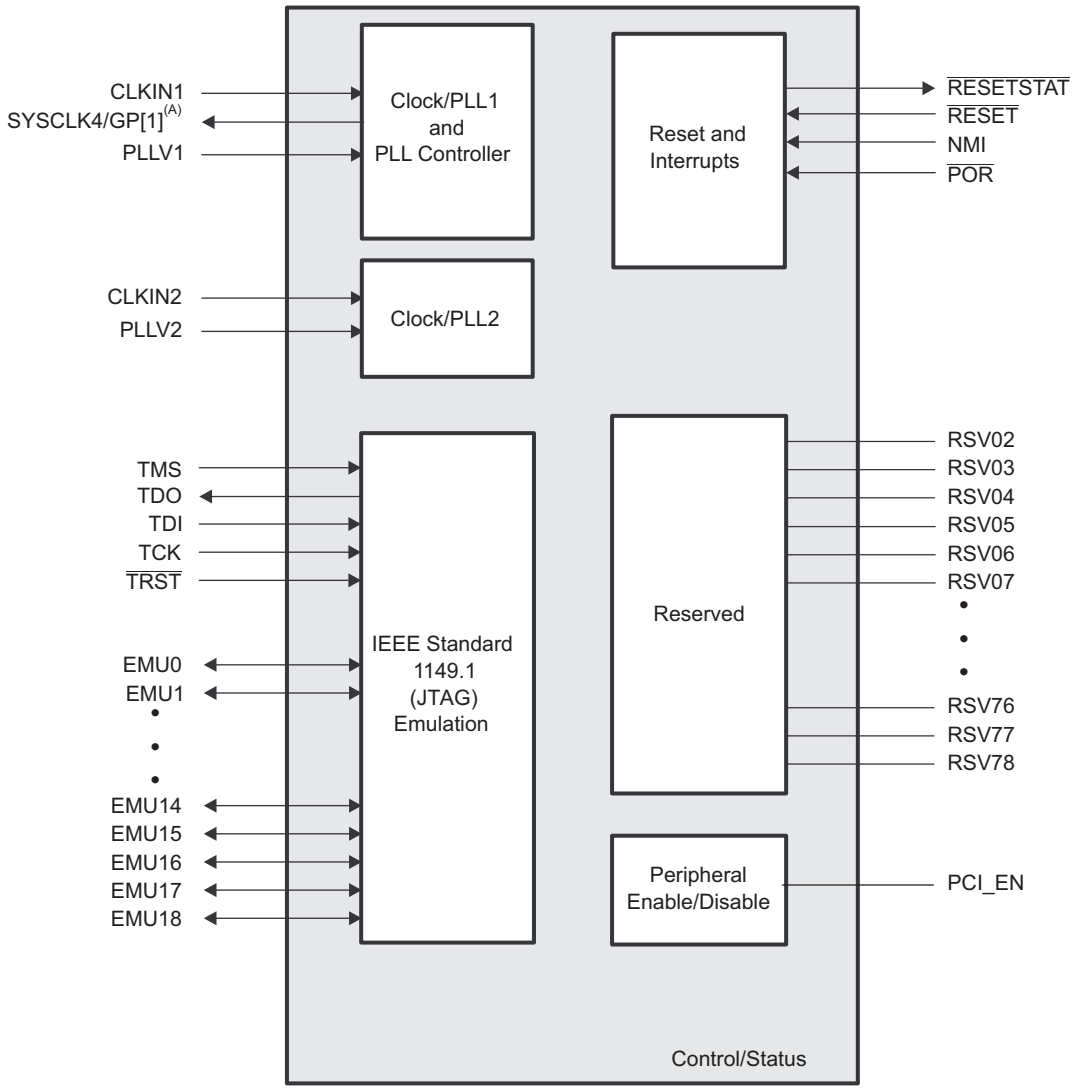
**Figure 2-4. C6454 Pin Map (Bottom View) [Quadrant C]**



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | $\overline{\text{PCBE0}}$/ GP[2] | $\overline{\text{PREQ}}$/ GP[15] | $\overline{\text{PINTA}}$/ GP[14] | $\overline{\text{PTRDY}}$ | $\overline{\text{PCBE3}}$ | $DV_{DD33}$ | $V_{SS}$ | | | | RSV05 | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | P |
| N | $CV_{DDMON}$ | $V_{SS}$ | MDIO | MTCLK/ RMREFCLK | MTXD7 | RSV29 | RSV28 | | | | RSV04 | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | N |
| M | MTXD0/ RMTXD0 | MRXD7 | MTXD6 | MTXD2 | MDCLK | $V_{SS}$ | $DV_{DD33}$ | | | | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | M |
| L | MRXD4 | MRXD5 | MTXD4 | MTXD1/ RMTXD1 | MTXD5 | $DV_{DD33MON}$ | $V_{SS}$ | | | | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | L |
| K | $DV_{DD33}$ | $V_{SS}$ | MCOL | MTXD3 | GMTCLK | $V_{SS}$ | $DV_{DD33}$ | | | | | | | | | K |
| J | MRXD2 | MRXD0/ RMRXD0 | MRXD3 | MCRS/ RMCRSDV | MTXEN/ RMTXEN | $DV_{DD33}$ | $V_{SS}$ | | | | | | | | | J |
| H | MRCLK | MRXD6 | MRXD1/ RMRXD1 | MRXER/ RMRXER | MRXDV | $V_{SS}$ | $DV_{DD15}$ | | | | | | | | | H |
| G | $V_{SS}$ | $DV_{DD33}$ | CLKIN2 | RSV07 | $V_{SS}$ | $DV_{DD15}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | G |
| F | RSV14 | RSV13 | $DV_{DD15MON}$ | $V_{SS}$ | $DV_{DD15}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | DED11 | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | F |
| E | RGRXD0 | RGRXD1 | RGRXC | RGRXD2 | $V_{SS}$ | RSV34 | $V_{SS}$ | DSDDQS1 | DED10 | $DV_{DD18}$ | DSDDQS0 | $DV_{DD18}$ | RSV18 | $\overline{\text{DCE0}}$ | DBA2 | E |
| D | $V_{SS}$ | $DV_{DD15}$ | RGTXCTL | RGTXC | $DV_{DD15}$ | RSV35 | DED14 | $\overline{\text{DSDDQS1}}$ | DED9 | DED7 | $\overline{\text{DSDDQS0}}$ | DED3 | $\overline{\text{DSDCAS}}$ | DSDCKE | DBA1 | D |
| C | RGRXD3 | RGRXCTL | RGTXD2 | RGREFCLK | $V_{SS}$ | RSV25 | DED15 | DSDDQM1 | DED8 | DED6 | DSDDQM0 | DED2 | $\overline{\text{DSDRAS}}$ | $V_{REFSSTL}$ | DBA0 | C |
| B | $V_{SS}$ | $V_{REFHSTL}$ | RGTXD1 | RGMDCLK | $DV_{DD15}$ | RSV24 | DED12 | $DV_{DD18}$ | DSDDQ GATE1 | DED5 | $DV_{DD18}$ | DED1 | $\overline{\text{DSDWE}}$ | DDR2 CLKOUT | DEA13 | B |
| A | $DV_{DD15}$ | RGTXD3 | RGTXD0 | RGMDIO | PLLV2 | RSV21 | DED13 | $V_{SS}$ | DSDDQ GATE0 | DED4 | $V_{SS}$ | DED0 | $AV_{DLL1}$ | $\overline{\text{DDR2}}$ $\overline{\text{CLKOUT}}$ | DEA12 | A |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |

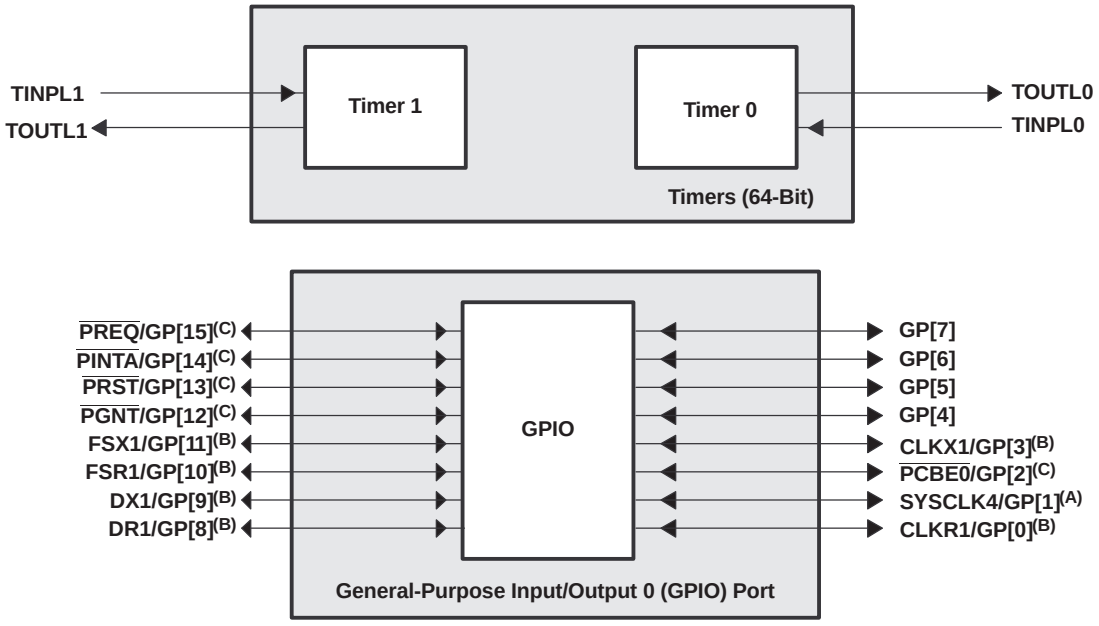
**Figure 2-5. C6454 Pin Map (Bottom View) [Quadrant D]**

## 2.6 Signal Groups Description




A. This pin functions as GP[1] by default. For more details, see Section 3.

**Figure 2-6. CPU and Peripheral Signals**




A. This pin functions as GP[1] by default. For more details, see the *Device Configuration* section of this document.
B. These McBSP1 peripheral pins are muxed with the GPIO peripheral pins and by default these signals function as GPIO peripheral pins. For more details, see the *Device Configuration* section of this document.
C. These PCI peripheral pins are muxed with the GPIO peripheral pins and by default these signals function as GPIO peripheral pins. For more details, see the *Device Configuration* section of this document.

**Figure 2-7. Timers/GPIO Peripheral Signals**

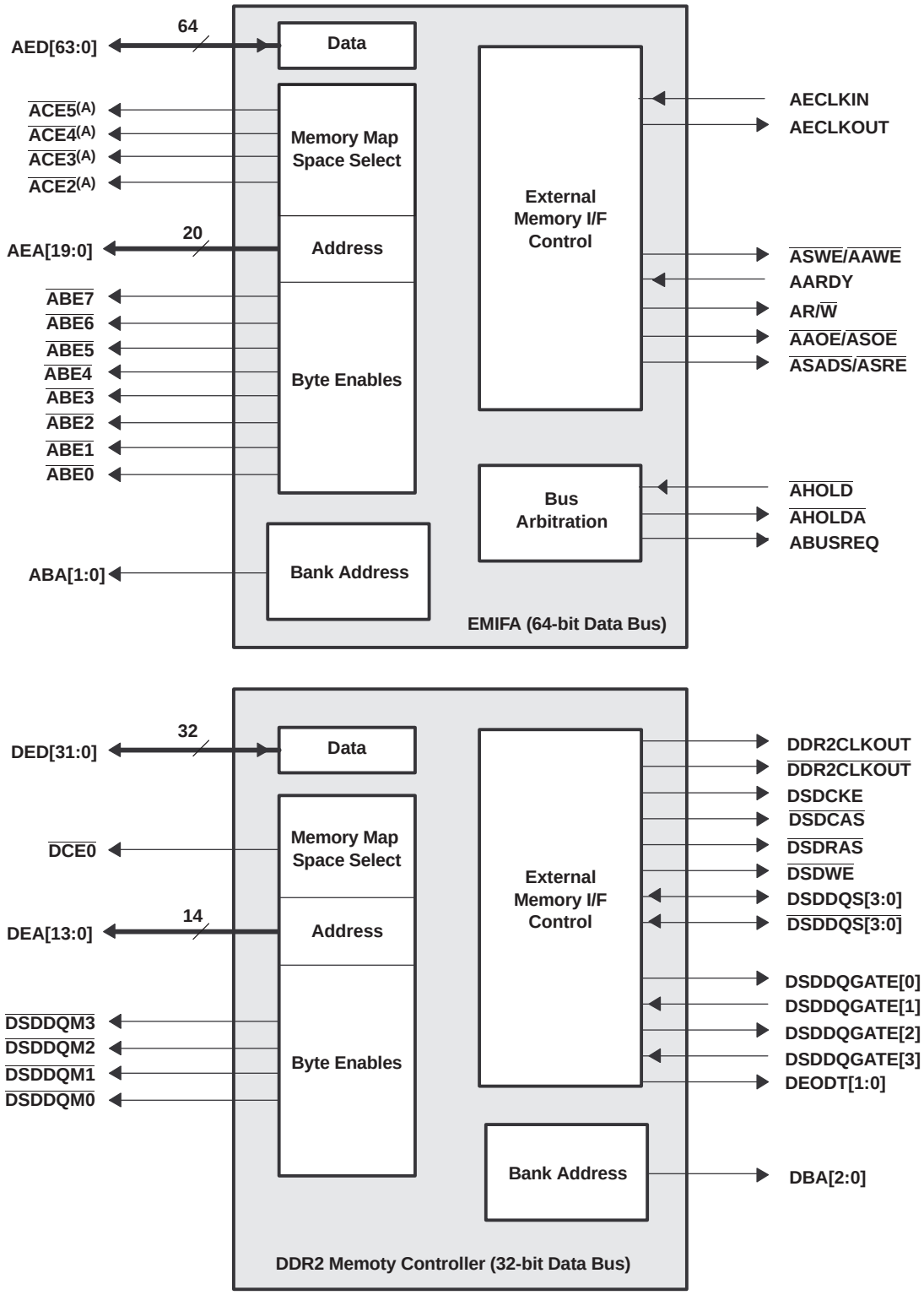
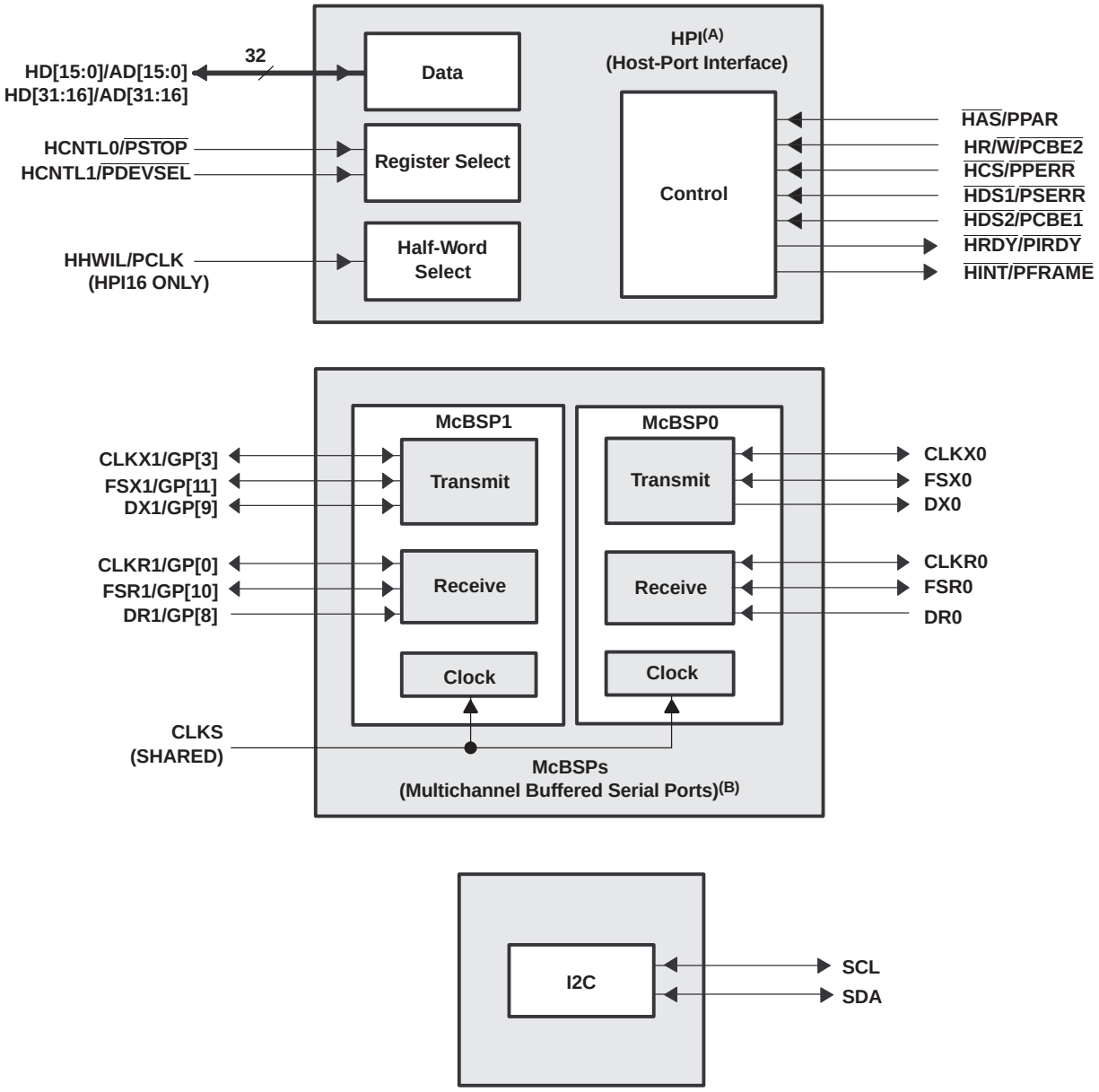


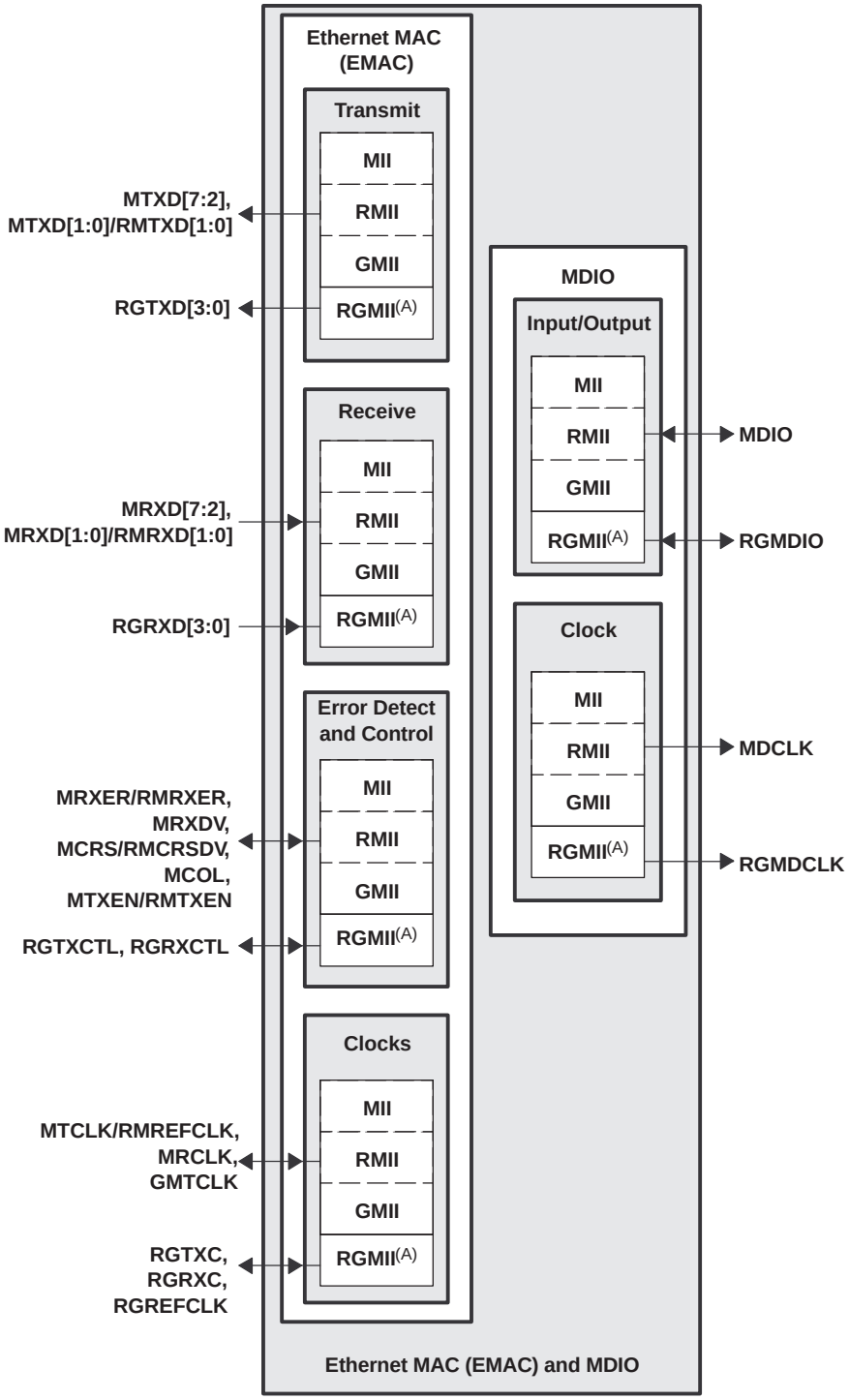

**Figure 2-8. EMIFA and DDR2 Memory Controller Peripheral Signals**




A. These HPI pins are muxed with the PCI peripheral. By default, these pins function as HPI. When the HPI is enabled, the number of HPI pins used depends on the HPI configuration (HPI16 or HPI32). For more details on these muxed pins, see the *Device Configuration* section of this document.

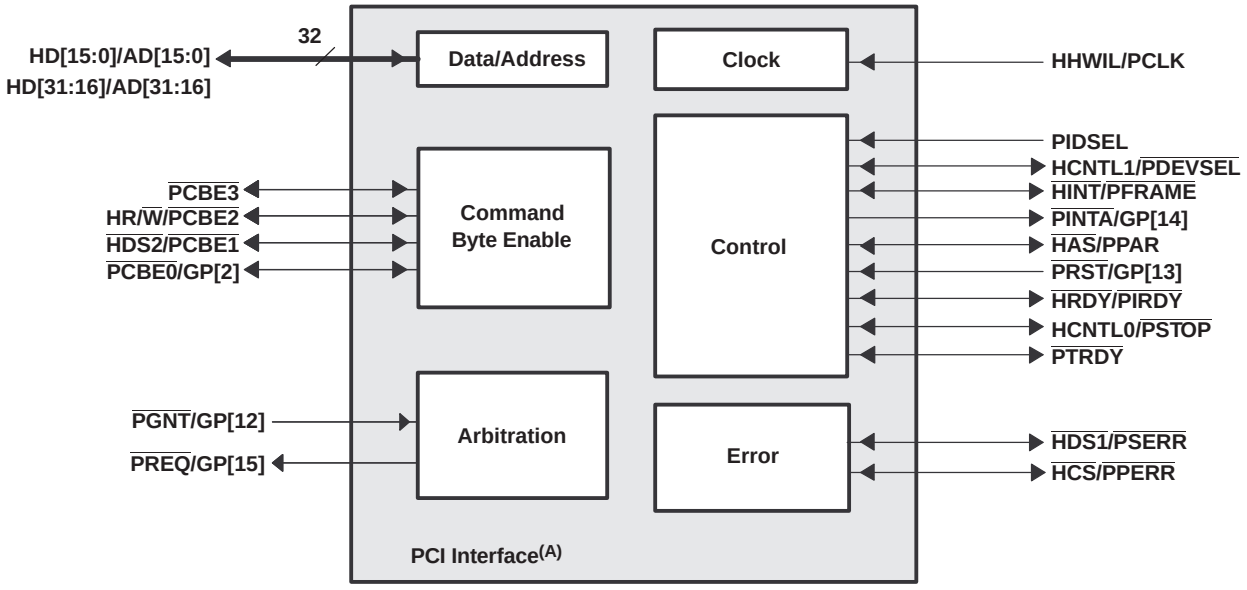
B. These McBSP1 peripheral pins are muxed with the GPIO peripheral pins and by default these signals function as GPIO peripheral pins. For more details, see the *Device Configuration* section of this document.

**Figure 2-9. HPI/McBSP/I2C Peripheral Signals**




A. RGMII signals are mutually exclusive to all other EMAC signals.

**Figure 2-10. EMAC/MDIO [MII, GMII, RMII, and RGMII] Peripheral Signals**




A. These PCI pins are muxed with the HPI or GPIO peripherals. By default, these signals function as HPI or GPIO or EMAC. For more details on these muxed pins, see the *Device Configuration* section of this document.

**Figure 2-11. PCI Peripheral Signals**

## 2.7 Terminal Functions

The terminal functions table (Table 2-3) identifies the external signal names, the associated pin (ball) numbers along with the mechanical package designator, the pin type (I, O/Z, or I/O/Z), whether the pin has any internal pullup/pulldown resistors, and a functional pin description. For more detailed information on device configuration, peripheral selection, multiplexed/shared pins, and pullup/pulldown resistors, see Section 3, *Device Configuration*.

**Table 2-3. Terminal Functions**

| SIGNAL NAME | SIGNAL NO. | TYPE(1) | IPD/IPU(2) | DESCRIPTION |
|---|---|---|---|---|
| **CLOCK/PLL CONFIGURATIONS** | | | | |
| CLKIN1 | N28 | I | IPD | Clock Input for PLL1. |
| CLKIN2 | G3 | I | IPD | Clock Input for PLL2. |
| PLLV1 | T29 | A | | 1.8-V I/O supply voltage for PLL1 |
| PLLV2 | A5 | A | | 1.8-V I/O supply voltage for PLL2 |
| SYSCLK4/GP[1](3) | AJ13 | I/O/Z | IPD | SYSCLK4 is the clock output at 1/8 of the device speed (**O/Z**) or this pin can be programmed as the GP1 pin (**I/O/Z**) [default]. |
| **JTAG EMULATION** | | | | |
| TMS | AJ10 | I | IPU | JTAG test-port mode select |
| TDO | AH8 | O/Z | IPU | JTAG test-port data out |
| TDI | AH9 | I | IPU | JTAG test-port data in |
| TCK | AJ9 | I | IPU | JTAG test-port clock |
| $\overline{\text{TRST}}$ | AH7 | I | IPD | JTAG test-port reset. For IEEE 1149.1 JTAG compatibility, see Section 7.18.3.1.1. |
| EMU0(4) | AF7 | I/O/Z | IPU | Emulation pin 0 |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. For most systems, a 1-kΩ resistor can be used to oppose the IPU/IPD. For more detailed information on pullup/pulldown resistors and situations where external pullup/pulldown resistors are required, see Section 3.7, *Pullup/Pulldown Resistors*.
(3) These pins are multiplexed pins. For more details, see Section 3, *Device Configuration*.
(4) The C6454 DSP does not require external pulldown resistors on the EMU0 and EMU1 pins for normal or boundary-scan operation.

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE(1) | IPD/IPU(2) | DESCRIPTION |
|---|---|---|---|---|
| EMU1(4) | AE11 | I/O/Z | IPU | Emulation pin 1 |
| EMU2 | AG9 | I/O/Z | IPU | Emulation pin 2 |
| EMU3 | AF10 | I/O/Z | IPU | Emulation pin 3 |
| EMU4 | AF9 | I/O/Z | IPU | Emulation pin 4 |
| EMU5 | AE12 | I/O/Z | IPU | Emulation pin 5 |
| EMU6 | AG8 | I/O/Z | IPU | Emulation pin 6 |
| EMU7 | AF12 | I/O/Z | IPU | Emulation pin 7 |
| EMU8 | AF11 | I/O/Z | IPU | Emulation pin 8 |
| EMU9 | AH13 | I/O/Z | IPU | Emulation pin 9 |
| EMU10 | AD10 | I/O/Z | IPU | Emulation pin 10 |
| EMU11 | AD12 | I/O/Z | IPU | Emulation pin 11 |
| EMU12 | AE10 | I/O/Z | IPU | Emulation pin 12 |
| EMU13 | AD8 | I/O/Z | IPU | Emulation pin 13 |
| EMU14 | AF13 | I/O/Z | IPU | Emulation pin 14 |
| EMU15 | AE9 | I/O/Z | IPU | Emulation pin 15 |
| EMU16 | AH12 | I/O/Z | IPU | Emulation pin 16 |
| EMU17 | AH10 | I/O/Z | IPU | Emulation pin 17 |
| EMU18 | AE13 | I/O/Z | IPU | Emulation pin 18 |
| **RESETS, INTERRUPTS, AND GENERAL-PURPOSE INPUT/OUTPUTS** | | | | |
| $\overline{\text{RESET}}$ | AG14 | I | | Device reset |
| NMI | AH4 | I | IPD | Nonmaskable interrupt, edge-driven (rising edge)<br>Any noise on the NMI pin may trigger an NMI interrupt; therefore, if the NMI pin is not used, it is recommended that the NMI pin be grounded versus relying on the IPD. |
| $\overline{\text{RESETSTAT}}$ | AE14 | O | | Reset Status pin. The $\overline{\text{RESETSTAT}}$ pin indicates when the device is in reset |
| $\overline{\text{POR}}$ | AF14 | I | | Power on reset. |
| GP[7] | AG2 | I/O/Z | IPD | General-purpose input/output (GPIO) pins (**I/O/Z**). |
| GP[6] | AG3 | I/O/Z | IPD | |
| GP[5] | AJ2 | I/O/Z | IPD | |
| GP[4] | AH2 | I/O/Z | IPD | |
| $\overline{\text{PREQ}}$/ GP[15] | P2 | I/O/Z | | PCI peripheral pins or General-purpose input/output (GPIO) [15:12, 2] pins (**I/O/Z**) [default] |
| $\overline{\text{PINTA}}$ (5)/ GP[14] | P3 | I/O/Z | | |
| $\overline{\text{PRST}}$/ GP[13] | R5 | I/O/Z | | |
| $\overline{\text{PGNT}}$/ GP[12] | R4 | I/O/Z | | PCI bus request (**O/Z**) or GP[15] (**I/O/Z**) [default]<br>PCI interrupt A (**O/Z**) or GP[14] (**I/O/Z**) [default]<br>PCI reset (**I**) or GP[13] (**I/O/Z**) [default]<br>PCI bus grant (**I**) or GP[12] (**I/O/Z**) [default]<br>PCI command/byte enable 0 (**I/O/Z**) or GP[2] (**I/O/Z**) [default] |
| FSX1/GP[11] | AG4 | I/O/Z | IPD | |
| FSR1/GP[10] | AE5 | I/O/Z | IPD | |
| DX1/GP[9] | AG5 | I/O/Z | IPD | |
| DR1/GP[8] | AH5 | I/O/Z | IPD | McBSP1 transmit clock (**I/O/Z**) or GP[3] (**I/O/Z**) [default]<br>McBSP1 receive clock (**I/O/Z**) or GP[0] (**I/O/Z**) [default] |
| CLKX1/GP[3] | AF5 | I/O/Z | IPD | |
| $\overline{\text{PCBE0}}$/ GP[2] | P1 | I/O/Z | | GP[1] pin (**I/O/Z**). SYSCLK4 is the clock output at 1/8 of the device speed (**O/Z**) or this pin can be programmed as a GP[1] pin (**I/O/Z**) [default]. |
| SYSCLK4/GP[1] | AJ13 | O/Z | IPD | |
| CLKR1/GP[0] | AF4 | I/O/Z | IPD | |
| **HOST-PORT INTERFACE (HPI) or PERIPHERAL COMPONENT INTERCONNECT (PCI)** | | | | |
| PCI_EN | Y29 | I | IPD | PCI enable pin. This pin controls the selection (enable/disable) of the HPI and GP[15:8], or PCI peripherals. This pin works in conjunction with the MCBSP**1**_EN (AEA5 pin) to enable/disable other peripherals (for more details, see Section 3, *Device Configuration*). |

(5) These pins function as open-drain outputs when configured as PCI pins.

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE(1) | IPD/IPU(2) | DESCRIPTION |
|---|---|---|---|---|
| HINT/PFRAME | U3 | I/O/Z | | Host interrupt from DSP to host (**O/Z**) or PCI frame (**I/O/Z**) |
| HCNTL1/PDEVSEL | U4 | I/O/Z | | Host control - selects between control, address, or data registers (**I**) [default] or PCI device select (**I/O/Z**) |
| HCNTL0/PSTOP | U5 | I/O/Z | | Host control - selects between control, address, or data registers (**I**) [default] or PCI stop (**I/O/Z**) |
| HHWIL/PCLK | V3 | I/O/Z | | Host half-word select - first or second half-word (not necessarily high or low order)<br>[For HPI16 bus width selection only] (**I**) [default] or PCI clock (**I**) |
| HR/W/PCBE2 | T5 | I/O/Z | | Host read or write select (**I**) [default] or PCI command/byte enable 2 (**I/O/Z**) |
| HAS/PPAR | T3 | I/O/Z | | Host address strobe (**I**) [default] or PCI parity (**I/O/Z**) |
| HCS/PPERR | U6 | I/O/Z | | Host chip select (**I**) [default] or PCI parity error (**I/O/Z**) |
| HDS1/PSERR (5) | U2 | I/O/Z | | Host data strobe 1 (**I**) [default] or PCI system error (**I/O/Z**) |
| HDS2/PCBE1 | U1 | I/O/Z | | Host data strobe 2 (**I**) [default] or PCI command/byte enable 1 (**I/O/Z**) |
| HRDY/PIRDY | T4 | I/O/Z | | Host ready from DSP to host (**O/Z**) [default] or PCI initiator ready (**I/O/Z**) |
| PREQ/ GP[15] | P2 | I/O/Z | | PCI bus request (**O/Z**) or GP[15] (**I/O/Z**) [default] |
| PINTA (6)/ GP[14] | P3 | I/O/Z | | PCI interrupt A (**O/Z**) or GP[14] (**I/O/Z**) default] |
| PRST/ GP[13] | R5 | I/O/Z | | PCI reset (**I**) or GP[13] (**I/O/Z**) [default] |
| PGNT/ GP[12] | R4 | I/O/Z | | or PCI bus grant (**I**) or GP[12] (**I/O/Z**)[default] |
| PCBE0/ GP[2] | P1 | I/O/Z | | PCI command/byte enable 0 (**I/O/Z**) or GP[2] (**I/O/Z**)[default] |
| PCBE3 | P5 | I/O/Z | | PCI command/byte enable 3 (**I/O/Z**). By default, this pin has no function. |
| PIDSEL | R3 | I | | PCI initialization device select (**I**). By default, this pin has no function. |
| PTRDY | P4 | I/O/Z | | PCI target ready (PRTDY) (**I/O/Z**). By default, this pin has no function. |
| HD31/AD31 | AA3 | I/O/Z | | Host-port data [31:16] pin (**I/O/Z**) [default] or PCI data-address bus [31:16] (**I/O/Z**) |
| HD30/AD30 | AA5 | | | |
| HD29/AD29 | AC4 | | | |
| HD28/AD28 | AA4 | | | |
| HD27/AD27 | AC5 | | | |
| HD26/AD26 | Y1 | | | |
| HD25/AD25 | AD2 | | | |
| HD24/AD24 | W1 | | | |
| HD23/AD23 | AC3 | | | |
| HD22/AD22 | AE1 | | | |
| HD21/AD21 | AD1 | | | |
| HD20/AD20 | W2 | | | |
| HD19/AD19 | AC1 | | | |
| HD18/AD18 | Y2 | | | |
| HD17/AD17 | AB1 | | | |
| HD16/AD16 | Y3 | | | |

(6) These pins function as open-drain outputs when configured as PCI pins.

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| HD15/AD15 | AB2 | I/O/Z | | Host-port data [15:0] pin (**I/O/Z**) [default] or PCI data-address bus [15:0] (**I/O/Z**) |
| HD14/AD14 | W4 | | | |
| HD13/AD13 | AC2 | | | |
| HD12/AD12 | V4 | | | |
| HD11/AD11 | AF3 | | | |
| HD10/AD10 | AE3 | | | |
| HD9/AD9 | AB3 | | | |
| HD8/AD8 | W5 | | | |
| HD7/AD7 | AB4 | | | |
| HD6/AD6 | Y4 | | | |
| HD5/AD5 | AD3 | | | |
| HD4/AD4 | Y5 | | | |
| HD3/AD3 | AD4 | | | |
| HD2/AD2 | W6 | | | |
| HD1/AD1 | AB5 | | | |
| HD0/AD0 | AE2 | | | |
| **EMIFA (64-BIT) - CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY** | | | | |
| ABA1/EMIFA_EN | V25 | O/Z | IPD | EMIFA bank address control (ABA[1:0])<br>• Active-low bank selects for the 64-bit EMIFA. When interfacing to 16-bit Asynchronous devices, ABA1 carries bit 1 of the byte address. For an 8-bit Asynchronous interface, ABA[1:0] are used to carry bits 1 and 0 of the byte address |
| ABA0/DDR2_EN | V26 | O/Z | IPD | DDR2 Memory Controller enable (DDR2_EN) [**ABA0**]<br>0 - DDR2 Memory Controller peripheral pins are disabled (default)<br>1 - DDR2 Memory Controller peripheral pins are enabled<br><br>EMIFA enable (EMIFA_EN) [**ABA1**]<br>0 - EMIFA peripheral pins are disabled (default)<br>1 - EMIFA peripheral pins are enabled |
| $\overline{\text{ACE5}}$ | V27 | O/Z | IPU | EMIFA memory space enables<br>• Enabled by bits 28 through 31 of the word address<br>• Only one pin is asserted during any external data access<br>Note: The C6454 device does *not* have $\overline{\text{ACE0}}$ and $\overline{\text{ACE1}}$ pins |
| $\overline{\text{ACE4}}$ | V28 | O/Z | IPU | |
| $\overline{\text{ACE3}}$ | W26 | O/Z | IPU | |
| $\overline{\text{ACE2}}$ | W27 | O/Z | IPU | |
| $\overline{\text{ABE7}}$ | W29 | O/Z | IPU | EMIFA byte-enable control<br>• Decoded from the low-order address bits. The number of address bits or byte enables used depends on the width of external memory.<br>• Byte-write enables for most types of memory. |
| $\overline{\text{ABE6}}$ | K26 | O/Z | IPU | |
| $\overline{\text{ABE5}}$ | L29 | O/Z | IPU | |
| $\overline{\text{ABE4}}$ | L28 | O/Z | IPU | |
| $\overline{\text{ABE3}}$ | AA29 | O/Z | IPU | |
| $\overline{\text{ABE2}}$ | AA28 | O/Z | IPU | |
| $\overline{\text{ABE1}}$ | AA25 | O/Z | IPU | |
| $\overline{\text{ABE0}}$ | AA26 | O/Z | IPU | |
| **EMIFA (64-BIT) - BUS ARBITRATION** | | | | |
| $\overline{\text{AHOLDA}}$ | N26 | O | IPU | EMIFA hold-request-acknowledge to the host |
| $\overline{\text{AHOLD}}$ | R29 | I | IPU | EMIFA hold request from the host |
| ABUSREQ | L27 | O | IPU | EMIFA bus request output |
| **EMIFA (64-BIT) - ASYNCHRONOUS/SYNCHRONOUS MEMORY CONTROL** | | | | |
| AECLKIN | N29 | I | IPD | EMIFA external input clock. The EMIFA input clock (AECLKIN or SYSCLK4 clock) is selected at reset via the pullup/pulldown resistor on the AEA[15] pin. Note: AECLKIN is the default for the EMIFA input clock. |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE(1) | IPD/IPU(2) | DESCRIPTION |
|---|---|---|---|---|
| AECLKOUT | V29 | O/Z | IPD | EMIFA output clock [at EMIFA input clock (AECLKIN or SYSCLK4) frequency] |
| $\overline{\text{AAWE}}$/$\overline{\text{ASWE}}$ | AB25 | O/Z | IPU | Asynchronous memory write-enable/Programmable synchronous interface write-enable |
| AARDY | K29 | I | IPU | Asynchronous memory ready input |
| AR/$\overline{\text{W}}$ | W25 | O/Z | IPU | Asynchronous memory read/write |
| $\overline{\text{AAOE}}$/$\overline{\text{ASOE}}$ | Y28 | O/Z | IPU | Asynchronous/Programmable synchronous memory output-enable |
| $\overline{\text{ASADS}}$/$\overline{\text{ASRE}}$ | R26 | O/Z | IPU | Programmable synchronous address strobe or read-enable<br>• For programmable synchronous interface, the R_ENABLE field in the Chip Select x Configuration Register selects between $\overline{\text{ASADS}}$ and $\overline{\text{ASRE}}$:<br>– If R_ENABLE = 0, then the $\overline{\text{ASADS}}$/$\overline{\text{ASRE}}$ signal functions as the $\overline{\text{ASADS}}$ signal.<br>– If R_ENABLE = 1, then the $\overline{\text{ASADS}}$/$\overline{\text{ASRE}}$ signal functions as the $\overline{\text{ASRE}}$ signal. |
| **EMIFA (64-BIT) - ADDRESS** | | | | |
| AEA19/BOOTMODE3 | N25 | O/Z | IPD | EMIFA external address (word address) (**O/Z**)<br>Controls initialization of the DSP modes at reset (**I**) via pullup/pulldown resistors [For more detailed information, see Section 3, *Device Configuration*.]<br>**Note:** If a configuration pin must be routed out from the device and 3-stated (not driven), the internal pullup/pulldown (IPU/IPD) resistor should not be relied upon; TI recommends the use of an external pullup/pulldown resistor. For more detailed information on pullup/pulldown resistors and situations where external pullup/pulldown resistors are required, see Section 3.7, *Pullup/Pulldown Resistors*.<br>• Boot mode - device boot mode configurations (BOOTMODE[3:0]) [**Note:** the peripheral ***must*** be enabled to use the particular boot mode.]<br>**AEA[19:16]**:<br>0000 - No boot (default mode)<br>0001 - Host boot (HPI)<br>0010 -Reserved<br>0011 - Reserved<br>0100 - EMIFA 8-bit ROM boot<br>0101 - Master I2C boot<br>0110 - Slave I2C boot<br>0111 - Host boot (PCI)<br>1000 thru 1111 - Reserved<br>For more detailed information on the boot modes, see Section 2.4, *Boot Sequence*.<br>CFGGP[2:0] pins must be set to 000b during reset for proper operation of the PCI boot mode.<br>• EMIFA input clock source select<br>Clock mode select for EMIFA (AECLKIN_SEL)<br>**AEA15**:<br>0 - AECLKIN (default mode)<br>1 - SYSCLK4 (CPU/x) Clock Rate. The SYSCLK4 clock rate is software selectable via the Software PLL1 Controller. By default, SYSCLK4 is selected as CPU/8 clock rate.<br>• HPI peripheral bus width (HPI_WIDTH) select<br>[Applies only when HPI is enabled; PCI_EN pin = 0]<br>**AEA14**:<br>0 - HPI operates as an HPI16 (default). (HPI bus is 16 bits wide. HD[15:0] pins are used and the remaining HD[31:16] pins are reserved pins in the Hi-Z state.)<br>1 - HPI operates as an HPI32.<br>• Device Endian mode (LENDIAN)<br>**AEA13**:<br>0 - System operates in Big Endian mode<br>1 - System operates in Little Endian mode(default)<br>**Note:** For proper C6454 device operation, the **AEA12** and **AEA11** pins ***must*** be externally pulled down with a 1-kΩ resistor at device reset. |
| AEA18/BOOTMODE2 | L26 | | | |
| AEA17/BOOTMODE1 | L25 | | | |
| AEA16/BOOTMODE0 | P26 | | | |
| AEA15/AECLKIN_SEL | P27 | | | |
| AEA14/HPI_WIDTH | R25 | | | |
| AEA13/LENDIAN | R27 | O/Z | IPU | |
| AEA12 | R28 | | | |
| AEA11 | T25 | O/Z | IPD | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| AEA10/MACSEL1 | M25 | O/Z | IPD | • EMAC/MDIO interface select bits (MACSEL[1:0])<br>There are two configuration pins — MACSEL[1:0] — to select the EMAC/MDIO interface.<br>**AEA[10:9]**: MACSEL[1:0]<br>00 - 10/100 EMAC/MDIO MII Mode Interface (default)<br>01 - 10/100 EMAC/MDIO RMII Mode Interface<br>10 - 10/100/1000 EMAC/MDIO GMII Mode Interface<br>11 - 10/100/1000 with RGMII Mode Interface<br>[RGMII interface requires a 1.8-V or 1.5-V I/O supply] |
| AEA9/MACSEL0 | M27 | | | |
| AEA8/PCI_EEAI | P25 | | | • PCI I2C EEPROM Auto-Initialization (PCI_EEAI)<br>**AEA8**: PCI auto-initialization via external I2C EEPROM<br>If the PCI peripheral is disabled (PCI_EN pin = 0), this pin must ***not*** be pulled up.<br>0 - PCI auto-initialization through I2C EEPROM is disabled (default).<br>1 - PCI auto-initialization through I2C EEPROM is enabled. |
| AEA7 | N27 | | | |
| AEA6/PCI66 | U27 | | | • PCI Frequency Selection (PCI66)<br>[The PCI peripheral *needs* be enabled (PCI_EN = 1) to use this function]<br>Selects the PCI operating frequency of 66-MHz or 33-MHz PCI operating frequency is selected at reset via the pullup/pulldown resistor on the PCI66 pin:<br>**AEA6**:<br>0 - PCI operates at 33 MHz (default).<br>1 - PCI operates at 66 MHz.<br>Note: If the PCI peripheral is disabled (PCI_EN = 0), this pin must ***not*** be pulled up. |
| AEA5/MCBSP1_EN | U28 | | | • McBSP1 Enable bit (MCBSP1_EN)<br>Selects which function is enabled on the McBSP1/GPIO muxed pins<br>**AEA5**:<br>0 - GPIO pin functions enabled (default).<br>1 - McBSP1 pin functions enabled. |
| AEA4/ SYSCLKOUT_EN | T28 | | | • SYSCLKOUT Enable pin (SYSCLKOUT_EN)<br>Selects which function is enabled on the SYSCLK4/GP[1] muxed pin<br>**AEA4**:<br>0 - GP[1] pin function of the SYSCLK4/GP[1] pin enabled (default).<br>1 - SYSCLK4 pin function of the SYSCLK4/GP[1] pin enabled. |
| AEA3 | T27 | | | • Configuration GPI (CFGGP[2:0]) (**AEA[2:0]**)<br>These pins are latched during reset and their values are shown in the DEVSTAT register. These values can be used by software routines for boot operations.<br>**AEA3**:<br>For proper C6454 device operation, the AEA3 pin ***must*** be pulled down to $V_{SS}$ using a 1-kΩ resistor. |
| AEA2/CFGGP2 | T26 | | | |
| AEA1/CFGGP1 | U26 | | | |
| AEA0/CFGGP0 | U25 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| **EMIFA (64-BIT) - DATA** | | | | |
| AED63 | F25 | I/O/Z | IPU | EMIFA external data |
| AED62 | A27 | | | |
| AED61 | C27 | | | |
| AED60 | C28 | | | |
| AED59 | E27 | | | |
| AED58 | D28 | | | |
| AED57 | D27 | | | |
| AED56 | F27 | | | |
| AED55 | G25 | | | |
| AED54 | G26 | | | |
| AED53 | A28 | | | |
| AED52 | F28 | | | |
| AED51 | B28 | | | |
| AED50 | G27 | | | |
| AED49 | B27 | | | |
| AED48 | G28 | | | |
| AED47 | H25 | | | |
| AED46 | J26 | | | |
| AED45 | H26 | | | |
| AED44 | J27 | | | |
| AED43 | H27 | | | |
| AED42 | J28 | | | |
| AED41 | C29 | | | |
| AED40 | J29 | | | |
| AED39 | D29 | | | |
| AED38 | J25 | | | |
| AED37 | F29 | | | |
| AED36 | F26 | | | |
| AED35 | G29 | | | |
| AED34 | K28 | | | |
| AED33 | K25 | | | |
| AED32 | K27 | | | |
| AED31 | AA27 | | | |
| AED30 | AG29 | | | |
| AED29 | AB29 | | | |
| AED28 | AC27 | | | |
| AED27 | AB28 | | | |
| AED26 | AC26 | | | |
| AED25 | AB27 | | | |
| AED24 | AC25 | | | |
| AED23 | AB26 | | | |
| AED22 | AD28 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| AED21 | AD29 | I/O/Z | IPU | EMIFA external data |
| AED20 | AJ28 | | | |
| AED19 | AF29 | | | |
| AED18 | AH28 | | | |
| AED17 | AE29 | | | |
| AED16 | AG28 | | | |
| AED15 | AF28 | | | |
| AED14 | AH26 | | | |
| AED13 | AE28 | | | |
| AED12 | AE26 | | | |
| AED11 | AD26 | | | |
| AED10 | AF27 | | | |
| AED9 | AG27 | | | |
| AED8 | AD27 | | | |
| AED7 | AE25 | | | |
| AED6 | AJ27 | | | |
| AED5 | AJ26 | | | |
| AED4 | AE27 | | | |
| AED3 | AG25 | | | |
| AED2 | AH27 | | | |
| AED1 | AF25 | | | |
| AED0 | AD25 | | | |
| **DDR2 MEMORY CONTROLLER (32-BIT) - CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY** | | | | |
| $\overline{\text{DCE0}}$ | E14 | O/Z | | DDR2 Memory Controller memory space enable. When the DDR2 Memory Controller is enabled, it always keeps this pin low. |
| DBA2 | E15 | O/Z | | DDR2 Memory Controller bank address control |
| DBA1 | D15 | O/Z | | |
| DBA0 | C15 | O/Z | | |
| DDR2CLKOUT | B14 | O/Z | | DDR2 Memory Controller output clock (CLKIN2 frequency × 10) |
| $\overline{\text{DDR2CLKOUT}}$ | A14 | O/Z | | Negative DDR2 Memory Controller output clock (CLKIN2 frequency × 10) |
| $\overline{\text{DSDCAS}}$ | D13 | O/Z | | DDR2 Memory Controller SDRAM column-address strobe |
| $\overline{\text{DSDRAS}}$ | C13 | O/Z | | DDR2 Memory Controller SDRAM row-address strobe |
| $\overline{\text{DSDWE}}$ | B13 | O/Z | | DDR2 Memory Controller SDRAM write-enable |
| DSDCKE | D14 | O/Z | | DDR2 Memory Controller SDRAM clock-enable (used for self-refresh mode) |
| DEODT1 | A17 | O/Z | | On-die termination signals to external DDR2 SDRAM. These pins should not be connected to the DDR2 SDRAM.<br>**Note:** There are no on-die termination resistors implemented on the C6454 DSP die. |
| DEODT0 | E16 | O/Z | | |
| DSDDQGATE3 | F21 | I | | DDR2 Memory Controller data strobe gate [3:0]<br>For hookup of these signals, see the *Implementing DDR2 PCB Layout on the TMS320C6455/5* application report (literature number SPRAAA7). |
| DSDDQGATE2 | E21 | O/Z | | |
| DSDDQGATE1 | B9 | I | | |
| DSDDQGATE0 | A9 | O/Z | | |
| DSDDQM3 | C23 | O/Z | | DDR2 Memory Controller byte-enable controls<br>• Decoded from the low-order address bits. The number of address bits or byte enables used depends on the width of external memory.<br>• Byte-write enables for most types of memory.<br>• Can be directly connected to SDRAM read and write mask signal (SDQM). |
| DSDDQM2 | C20 | O/Z | | |
| DSDDQM1 | C8 | O/Z | | |
| DSDDQM0 | C11 | O/Z | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| DSDDQS3 | E23 | I/O/Z | | DDR2 Memory Controller data strobe [3:0] positive |
| DSDDQS2 | E20 | I/O/Z | | |
| DSDDQS1 | E8 | I/O/Z | | |
| DSDDQS0 | E11 | I/O/Z | | |
| $\overline{\text{DSDDQS3}}$ | D23 | I/O/Z | | DDR2 data strobe [3:0] negative<br>**Note:** These pins are used to meet AC timings. For more detailed information, see the *Implementing DDR2 PCB Layout on the TMS320C6454/5* application report (literature number SPRAAA7). |
| $\overline{\text{DSDDQS2}}$ | D20 | I/O/Z | | |
| $\overline{\text{DSDDQS1}}$ | D8 | I/O/Z | | |
| $\overline{\text{DSDDQS0}}$ | D11 | I/O/Z | | |
| **DDR2 MEMORY CONTROLLER (32-BIT) - ADDRESS** | | | | |
| DEA13 | B15 | O/Z | | DDR2 Memory Controller external address |
| DEA12 | A15 | | | |
| DEA11 | A16 | | | |
| DEA10 | B16 | | | |
| DEA9 | C16 | | | |
| DEA8 | D16 | | | |
| DEA7 | B17 | | | |
| DEA6 | C17 | | | |
| DEA5 | D17 | | | |
| DEA4 | E17 | | | |
| DEA3 | A18 | | | |
| DEA2 | B18 | | | |
| DEA1 | C18 | | | |
| DEA0 | D18 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| **DDR2 MEMORY CONTROLLER (32-BIT) - DATA** | | | | |
| DED31 | B25 | I/O/Z | | DDR2 Memory Controller external data |
| DED30 | A25 | | | |
| DED29 | B24 | | | |
| DED28 | A24 | | | |
| DED27 | D22 | | | |
| DED26 | C22 | | | |
| DED25 | B22 | | | |
| DED24 | A22 | | | |
| DED23 | D21 | | | |
| DED22 | C21 | | | |
| DED21 | B21 | | | |
| DED20 | A21 | | | |
| DED19 | D19 | | | |
| DED18 | C19 | | | |
| DED17 | A19 | | | |
| DED16 | B19 | | | |
| DED15 | C7 | | | |
| DED14 | D7 | | | |
| DED13 | A7 | | | |
| DED12 | B7 | | | |
| DED11 | F9 | | | |
| DED10 | E9 | | | |
| DED9 | D9 | | | |
| DED8 | C9 | | | |
| DED7 | D10 | | | |
| DED6 | C10 | | | |
| DED5 | B10 | | | |
| DED4 | A10 | | | |
| DED3 | D12 | | | |
| DED2 | C12 | | | |
| DED1 | B12 | | | |
| DED0 | A12 | | | |
| **TIMER 1** | | | | |
| TOUTL1 | AG7 | O/Z | IPD | Timer 1 output pin for lower 32-bit counter |
| TINPL1 | AJ6 | I | IPD | Timer 1 input pin for lower 32-bit counter |
| **TIMER 0** | | | | |
| TOUTL0 | AF8 | O/Z | IPD | Timer 0 output pin for lower 32-bit counter |
| TINPL0 | AH6 | I | IPD | Timer 0 input pin for lower 32-bit counter |
| **INTER-INTEGRATED CIRCUIT (I2C)** | | | | |
| SCL | AG26 | I/O/Z | | I2C clock. When the I2C module is used, use an external pullup resistor. |
| SDA | AF26 | I/O/Z | | I2C data. When I2C is used, ensure there is an external pullup resistor. |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| **MULTICHANNEL BUFFERED SERIAL PORT 1 AND MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP1 and McBSP0)** | | | | |
| CLKS | AJ4 | I | IPD | McBSP external clock source (as opposed to internal) (**I**) [shared by McBSP1 and McBSP0] |
| **MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)** | | | | |
| CLKR1/GP[0] | AF4 | I/O/Z | IPD | McBSP1 receive clock (**I/O/Z**) or GP[0] (**I/O/Z**) [default] |
| FSR1/GP[10] | AE5 | I/O/Z | IPD | McBSP1 receive frame sync (**I/O/Z**) or GP[10] (**I/O/Z**) [default] |
| DR1/GP[8] | AH5 | I/O/Z | IPD | McBSP1 receive data (**I**) or GP[8] (**I/O/Z**) [default] |
| DX1/GP[9] | AG5 | I/O/Z | IPD | McBSP1 transmit data (**O/Z**) or GP[9] (**I/O/Z**) [default] |
| FSX1/GP[11] | AG4 | I/O/Z | IPD | McBSP1 transmit frame sync (**I/O/Z**) or GP[11] (**I/O/Z**) [default] |
| CLKX1/GP[3] | AF5 | I/O/Z | IPD | McBSP1 transmit clock (**I/O/Z**) or GP[3] (**I/O/Z**) [default] |
| **MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)** | | | | |
| CLKR0 | AG1 | I/O/Z | IPU | McBSP0 receive clock (**I/O/Z**) |
| FSR0 | AH3 | I/O/Z | IPD | McBSP0 receive frame sync (**I/O/Z**) |
| DR0 | AJ5 | I | IPD | McBSP0 receive data (**I**) |
| DX0 | AF6 | I/O/Z | IPD | McBSP0 transmit data (**O/Z**) |
| FSX0 | AJ3 | I/O/Z | IPD | McBSP0 transmit frame sync (**I/O/Z**) |
| CLKX0 | AG6 | I/O/Z | IPU | McBSP0 transmit clock (**I/O/Z**) |
| **MANAGEMENT DATA INPUT/OUTPUT (MDIO) FOR MII/RMII/GMII** | | | | |
| MDCLK | M5 | I/O/Z | IPD | MDIO serial clock (MDCLK) for MII/RMII/GMII mode (**O**) |
| MDIO | N3 | I/O/Z | IPU | MDIO serial data (MDIO) for MII/RMII/GMII mode (**I/O**) |
| **MANAGEMENT DATA INPUT/OUTPUT (MDIO) FOR RGMII** | | | | |
| RGMDCLK | B4 | O/Z | | MDIO serial clock (RGMII mode) (RGMDCLK) (**O**) |
| RGMDIO | A4 | I/O/Z | | MDIO serial data (RGMII mode) (RGMDIO) (**I/O**) |
| **ETHERNET MAC (EMAC) [MII/RMII/GMII]** | | | | |
| There are two configuration pins - the MAC_SEL[1:0] (AEA[10:9] pins) - that select one of the four interface modes (MII, RMII, GMII, or RGMII) for the EMAC/MDIO interface. For more detailed information on the EMAC configuration pins, see Section 3, *Device Configuration*. | | | | |
| MRCLK | H1 | I | | This pin is EMAC receive clock (MRCLK) for MII [default] or GMII. MACSEL[1:0] dependent. |
| MCRS/ RMCRSDV | J4 | I/O/Z | | This pin is EMAC carrier sense (MCRS) (**I**) for MII [default] or GMII, or EMAC carrier sense/receive data valid (RMCRSDV) (**I**) for RMII. MACSEL[1:0] dependent. |
| MRXER/ RMRXER | H4 | I | | This pin is EMAC receive error (MRXIR) (**I**) for MII [default], RMII, or GMII. MACSEL[1:0] dependent. |
| MRXDV | H5 | I | | This pin is EMAC MII [default] or GMII receive data valid (MRXDV) (**I**). MACSEL[1:0] dependent. |
| MRXD7 | M2 | I | | EMAC receive data bus for MII [default], RMII, or GMII These pins function as EMAC receive data pins for MII [default], RMII, or GMII (MRXD[x:0]) (**I**). MACSEL[1:0] dependent. |
| MRXD6 | H2 | | | |
| MRXD5 | L2 | | | |
| MRXD4 | L1 | | | |
| MRXD3 | J3 | | | |
| MRXD2 | J1 | | | |
| MRXD1/ RMRXD1 | H3 | | | |
| MRXD0/ RMRXD0 | J2 | | | |
| GMTCLK | K5 | O/Z | | This pin is EMAC GMII transmit clock (GMTCLK) (**O**). MACSEL[1:0] dependent. |
| MTCLK/ RMREFCLK | N4 | I | | This pin is either EMAC MII [default] or GMII transmit clock (MTCLK) (**I**) or the EMAC RMII reference clock (RMREFCLK) (**I**). The EMAC function is controlled by the MACSEL[1:0] (AEA[10:9] pins). For more detailed information, see Section 3, *Device Configuration*. |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| MCOL | K3 | I/O/Z | | This pin is the EMAC collision sense (MCDL) (**I**) for MII [default] or GMII. MACSEL[1:0] dependent. |
| MTXEN/ RMTXEN | J5 | I/O/Z | | This pin is either the EMAC transmit enable (MTXEN) (**O**) for MII [default], RMII, or GMII. MACSEL[1:0] dependent. |
| MTXD7 | N5 | O/Z | | EMAC transmit data bus for MII [default], RMII, or GMII.<br>These pins function as EMAC transmit data pins (MTXD[x:0]) (**O**) for MII, RMII, or GMII. MACSEL[1:0] dependent. |
| MTXD6 | M3 | | | |
| MTXD5 | L5 | | | |
| MTXD4 | L3 | | | |
| MTXD3 | K4 | | | |
| MTXD2 | M4 | | | |
| MTXD1/ RMTXD1 | L4 | | | |
| MTXD0/ RMTXD0 | M1 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| **ETHERNET MAC (EMAC) [RGMII]** | | | | |
| There are two configuration pins - the MAC_SEL[1:0] (AEA[10:9] pins) - that select one of the four interface modes (MII, RMII, GMII, or RGMII) for the EMAC/MDIO interface. For more detailed information on the EMAC configuration pins, see Section 3, *Device Configuration*. | | | | |
| RGREFCLK | C4 | O/Z | | RGMII reference clock (**O**). This 125-MHz reference clock is provided as a convenience. It can be used as a clock source to a PHY, so that the PHY may generate RXC clock to communicate with the EMAC. This clock is stopped while the device is in reset. This pin is available only when RGMII mode is selected ( MACSEL[1:0] =11). |
| RGTXC | D4 | O/Z | | RGMII transmit clock (**O**). This pin is available only when RGMII mode is selected (MACSEL[1:0] =11). |
| RGTXD3 | A2 | O/Z | | RGMII transmit data [3:0] (**O**). This pin is available only when RGMII mode is selected (MACSEL[1:0] =11). |
| RGTXD2 | C3 | | | |
| RGTXD1 | B3 | | | |
| RGTXD0 | A3 | | | |
| RGTXCTL | D3 | O/Z | | RGMII transmit enable (**O**). This pin is available only when RGMII mode is selected (MACSEL[1:0] =11). |
| RGRXC | E3 | I | | RGMII receive clock (**I**). This pin is available only when RGMII mode is selected (MACSEL[1:0] =11). |
| RGRXD3 | C1 | I | | RGMII receive data [3:0] (**I**). This pin is available only when RGMII mode is selected (MACSEL[1:0] =11). |
| RGRXD2 | E4 | I | | |
| RGRXD1 | E2 | I | | |
| RGRXD0 | E1 | I | | |
| RGRXCTL | C2 | I | | RGMII receive control (**I**). This pin is available only when RGMII mode is selected (MACSEL[1:0] =11). |
| **RESERVED FOR TEST** | | | | |
| RSV02 | V5 | | | Reserved. These pins must be connected directly to core supply ($CV_{DD}$) for proper device operation. |
| RSV03 | W3 | | | |
| RSV04 | N11 | | | |
| RSV05 | P11 | | | |
| RSV07 | G4 | I | | Reserved. This pin must be connected directly to 1.5-/1.8-V I/O supply ($DV_{DD15}$) for proper device operation.<br>**NOTE:** If the EMAC RGMII is not used, these pins can be connected directly to ground ($V_{SS}$). |
| RSV09 | D26 | I | | Reserved. This pin must be connected directly to the 1.8-V I/O supply ($DV_{DD18}$) for proper device operation. |
| RSV11 | D24 | | | Reserved. This pin must be connected to ground ($V_{SS}$) via a 200-Ω resistor for proper device operation.<br>**NOTE:** If the DDR2 Memory Controller is not used, the $V_{REFSSTL}$, RSV11, and RSV12 pins can be connected directly to ground ($V_{SS}$) to save power. However, connecting these pins directly to ground will prevent boundary-scan from functioning on the DDR2 Memory Controller pins. To preserve boundary-scan functionality on the DDR2 Memory Controller pins, see Section 7.3.4. |
| RSV12 | C24 | | | Reserved. This pin must be connected to the 1.8-V I/O supply ($DV_{DD18}$) via a 200-Ω resistor for proper device operation.<br>**NOTE:** If the DDR2 Memory Controller is not used, the $V_{REFSSTL}$, RSV11, and RSV12 pins can be connected directly to ground ($V_{SS}$) to save power. However, connecting these pins directly to ground will prevent boundary-scan from functioning on the DDR2 Memory Controller pins. To preserve boundary-scan functionality on the DDR2 Memory Controller pins, see Section 7.3.4. |
| RSV13 | F2 | | | Reserved. This pin must be connected to ground ($V_{SS}$) via a 200-Ω resistor for proper device operation.<br>**NOTE:** If the RGMII mode of the EMAC is not used, the $DV_{DD15}$, $V_{REFHSTL}$, RSV13, and RSV14 pins can be connected to directly ground ($V_{SS}$) to save power. However, connecting these pins directly to ground will prevent boundary-scan from functioning on the RGMII pins of the EMAC. To preserve boundary-scan functionality on the RGMII pins, see Section 7.3.4. |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| RSV14 | F1 | | | Reserved. This pin must be connected to the 1.5/1.8-V I/O supply ($DV_{DD15}$) via a 200-Ω resistor for proper device operation. **NOTE:** If the RGMII mode of the EMAC is not used, the $DV_{DD15}$, $V_{REFHSTL}$, RSV13, and RSV14 pins can be connected to directly ground ($V_{SS}$) to save power. However, connecting these pins directly to ground will prevent boundary-scan from functioning on the RGMII pins of the EMAC. To preserve boundary-scan functionality on the RGMII pins, see Section 7.3.4. |
| RSV15 | T1 | | | Reserved. This pin must be connected via a 39-Ω resistor directly to ground ($V_{SS}$) for proper device operation. The resistor used should have a minimal rating of 1/10 W. |
| RSV16 | T2 | | | Reserved. This pin must be connected via a 20-Ω resistor directly to 3.3-V I/O Supply ($DV_{DD33}$) for proper device operation. The resistor used should have a minimal rating of 1/10 W. |
| RSV17 | AE21 | A | | Reserved. (Leave unconnected, do not connect to power or ground.) |
| RSV18 | E13 | A | | |
| RSV19 | F18 | A | | |
| RSV20 | U29 | A | | |
| RSV21 | A6 | A | | |
| RSV22 | B26 | O | | |
| RSV23 | C26 | O | | |
| RSV24 | B6 | O | | |
| RSV25 | C6 | O | | |
| RSV26 | AJ11 | A | | |
| RSV27 | AH11 | A | | |
| RSV36 | AD11 | I/O/Z | IPU | |
| RSV37 | AD9 | I/O/Z | IPU | |
| RSV38 | AG10 | I/O/Z | IPU | |
| RSV39 | AG11 | I/O/Z | IPU | |
| RSV40 | AJ12 | I/O/Z | IPU | |
| RSV41 | W28 | O/Z | IPU | |
| RSV42 | Y26 | O/Z | IPU | |
| RSV43 | Y25 | O/Z | IPU | |
| RSV44 | Y27 | O/Z | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| RSV45 | AF15 | I | | Reserved. (Leave unconnected, do not connect to power or ground.) |
| RSV46 | AG15 | I | | |
| RSV47 | AF17 | O/Z | | |
| RSV48 | AG18 | O/Z | | |
| RSV49 | AG22 | O/Z | | |
| RSV50 | AF23 | O/Z | | |
| RSV51 | AF18 | O/Z | | |
| RSV52 | AG19 | O/Z | | |
| RSV53 | AG21 | O/Z | | |
| RSV54 | AF22 | O/Z | | |
| RSV55 | AH18 | I | | |
| RSV56 | AJ18 | I | | |
| RSV57 | AJ22 | I | | |
| RSV58 | AH22 | I | | |
| RSV59 | AH17 | I | | |
| RSV60 | AJ19 | I | | |
| RSV61 | AJ21 | I | | |
| RSV62 | AH23 | I | | |
| RSV28 | N7 | A | | Reserved. These pins must be connected directly to $V_{SS}$ for proper device operation. |
| RSV29 | N6 | A | | |
| RSV30 | P23 | A | | |
| RSV31 | P24 | A | | |
| RSV32 | D25 | | | Reserved. This pin must be connected to the 1.8-V I/O supply ($DV_{DD18}$) via a 1-kΩ resistor for proper device operation. |
| RSV33 | C25 | | | Reserved. This pin must be connected directly to ground for proper device operation. |
| RSV34 | E6 | | | Reserved. This pin must be connected to the 1.8-V I/O supply ($DV_{DD18}$) via a 1-kΩ resistor for proper device operation. |
| RSV35 | D6 | | | Reserved. This pin must be connected directly to ground for proper device operation. |
| RSV63 | AD20 | I | | Reserved. These pins must be connected directly to $V_{SS}$ for proper device operation. |
| RSV64 | AC15 | I | | |
| RSV65 | AC17 | I | | |
| RSV66 | AD16 | I | | |
| RSV67 | U16 | I | | |
| RSV68 | V15 | I | | |
| RSV69 | V17 | I | | |
| RSV70 | W16 | I | | |
| RSV71 | W18 | I | | |
| RSV72 | AE17 | I | | |
| RSV73 | AE19 | I | | |
| RSV74 | AE23 | I | | |
| RSV75 | AF20 | I | | |
| RSV76 | AH20 | I | | |
| RSV77 | AJ17 | I | | |
| RSV78 | AJ23 | I | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE(1) | IPD/IPU(2) | DESCRIPTION |
|---|---|---|---|---|
| **SUPPLY VOLTAGE MONITOR PINS** | | | | |
| $CV_{DDMON}$ | N1 | | | Die-side 1.2-V core supply ($CV_{DD}$) voltage monitor pin. The monitor pins indicate the voltage on the die and, therefore, provide the best probe point for voltage monitoring purposes. If the $CV_{DDMON}$ pin is not used, it should be connected directly to the 1.2-V core supply ($CV_{DD}$). |
| $DV_{DD33MON}$ | L6 | | | Die-side 3.3-V I/O supply ($DV_{DD33}$) voltage monitor pin. The monitor pins indicate the voltage on the die and, therefore, provide the best probe point for voltage monitoring purposes. If the $DV_{DD33MON}$ pin is not used, it should be connected directly to the 3.3-V I/O supply ($DV_{DD33}$). |
| $DV_{DD15MON}$ | F3 | I | | Die-side 1.5-/1.8-V I/O supply ($DV_{DD15}$) voltage monitor pin. The monitor pins indicate the voltage on the die and, therefore, provide the best probe point for voltage monitoring purposes. If the $DV_{DD15MON}$ pin is not used, it should be connected directly to the 1.5-/1.8-V I/O supply ($DV_{DD15}$).<br>**NOTE:** If the RGMII mode of the EMAC is not used, the $DV_{DD15}$, $DV_{DD15MON}$, $V_{REFHSTL}$, RSV13, and RSV14 pins can be connected directly to ground ($V_{SS}$) to save power. However, connecting these pins directly to ground will prevent boundary-scan from functioning on the RGMII pins of the EMAC. To preserve boundary-scan functionality on the RGMII pins, see Section 7.3.4. |
| $DV_{DD18MON}$ | A26 | | | Die-side 1.8-V I/O supply ($DV_{DD18}$) voltage monitor pin. The monitor pins indicate the voltage on the die and, therefore, provide the best probe point for voltage monitoring purposes. If the $DV_{DD18MON}$ pin is not used, it should be connected directly to the 1.8-V I/O supply ($DV_{DD18}$). |
| **SUPPLY VOLTAGE PINS** | | | | |
| $V_{REFSSTL}$ | C14 | A | | ($DV_{DD18}$/2)-V reference for SSTL buffer (DDR2 Memory Controller). This input voltage can be generated directly from $DV_{DD18}$ using two 1-kΩ resistors to form a resistor divider circuit.<br>**NOTE:** The DDR2 Memory Controller is not used, the $V_{REFSSTL}$, RSV11, and RSV12 pins can be connected directly to ground ($V_{SS}$) to save power. However, connecting these pins directly to ground will prevent boundary-scan from functioning on the DDR2 Memory Controller pins. To preserve boundary-scan functionality on the DDR2 Memory Controller pins, see Section 7.3.4. |
| $V_{REFHSTL}$ | B2 | A | | ($DV_{DD15}$/2)-V reference for HSTL buffer (EMAC RGMII). $V_{REFHSTL}$ can be generated directly from $DV_{DD15}$ using two 1-kΩ resistors to form a resistor divider circuit.<br>**NOTE:** If the RGMII mode of the EMAC is not used, the $DV_{DD15}$, $V_{REFHSTL}$, RSV13, and RSV14 pins can be connected to directly ground ($V_{SS}$) to save power. However, connecting these pins directly to ground will prevent boundary-scan from functioning on the RGMII pins of the EMAC. To preserve boundary-scan functionality on the RGMII pins, see Section 7.3.4. |
| $AV_{DLL1}$ | A13 | A | | 1.8-V I/O supply voltage. |
| $AV_{DLL2}$ | E18 | | | |
| $DV_{DD15}$ | A1 | S | | 1.8-V or 1.5-V I/O supply voltage for the RGMII function of the EMAC.<br>**NOTE:** If the RGMII mode of the EMAC is not used, the $DV_{DD15}$, $V_{REFHSTL}$, RSV13, and RSV14 pins can be connected to directly ground ($V_{SS}$) to save power. However, connecting these pins directly to ground will prevent boundary-scan from functioning on the RGMII pins of the EMAC. To preserve boundary-scan functionality on the RGMII pins, see Section 7.3.4. |
| | B5 | | | |
| | D2 | | | |
| | D5 | | | |
| | F5 | | | |
| | G6 | | | |
| | H7 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| $DV_{DD18}$ | B8 | S | | 1.8-V I/O supply voltage (DDR2 Memory Controller) |
| | B11 | | | |
| | B20 | | | |
| | B23 | | | |
| | E10 | | | |
| | E12 | | | |
| | E22 | | | |
| | E24 | | | |
| | F7 | | | |
| | F11 | | | |
| | F13 | | | |
| | F15 | | | |
| | F17 | | | |
| | F19 | | | |
| | F23 | | | |
| | G8 | | | |
| | G10 | | | |
| | G12 | | | |
| | G14 | | | |
| | G16 | | | |
| | G18 | | | |
| | G20 | | | |
| | G22 | | | |
| | G24 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| | A29 | | | |
| | E26 | | | |
| | E28 | | | |
| | G2 | | | |
| | H23 | | | |
| | H28 | | | |
| | J6 | | | |
| | J24 | | | |
| | K1 | | | |
| | K7 | | | |
| | K23 | | | |
| | L24 | | | |
| | M7 | | | |
| | M23 | | | |
| | M28 | | | |
| | N24 | | | |
| | P6 | | | |
| | P28 | | | |
| | R1 | | | |
| | R6 | | | |
| | R23 | | | |
| $DV_{DD33}$ | T7 | S | | 3.3-V I/O supply voltage |
| | T24 | | | |
| | U23 | | | |
| | V1 | | | |
| | V7 | | | |
| | V24 | | | |
| | W23 | | | |
| | Y7 | | | |
| | Y24 | | | |
| | AA1 | | | |
| | AA6 | | | |
| | AA23 | | | |
| | AB7 | | | |
| | AB24 | | | |
| | AC6 | | | |
| | AC9 | | | |
| | AC11 | | | |
| | AC13 | | | |
| | AC19 | | | |
| | AC21 | | | |
| | AC23 | | | |
| | AC29 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| $DV_{DD33}$ | AD5 | S | | 3.3-V I/O supply voltage |
| | AD7 | | | |
| | AD14 | | | |
| | AD18 | | | |
| | AD22 | | | |
| | AD24 | | | |
| | AE6 | | | |
| | AE8 | | | |
| | AE15 | | | |
| | AF1 | | | |
| | AF16 | | | |
| | AF24 | | | |
| | AG12 | | | |
| | AG17 | | | |
| | AG23 | | | |
| | AH14 | | | |
| | AH16 | | | |
| | AH24 | | | |
| | AJ1 | | | |
| | AJ7 | | | |
| | AJ15 | | | |
| | AJ25 | | | |
| | AJ29 | | | |
| $CV_{DD}$ | L12 | S | | 1.25-V core supply voltage (-1000 devices)<br>1.2-V core supply voltage (-850 and -720 devices) |
| | L14 | | | |
| | L16 | | | |
| | L18 | | | |
| | M11 | | | |
| | M13 | | | |
| | M15 | | | |
| | M17 | | | |
| | M19 | | | |
| | N12 | | | |
| | N14 | | | |
| | N16 | | | |
| | N18 | | | |
| | P13 | | | |
| | P15 | | | |
| | P17 | | | |
| | P19 | | | |
| | R12 | | | |
| | R14 | | | |
| | R16 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| $CV_{DD}$ | R18 | S | | 1.25-V core supply voltage (-1000 devices)<br>1.2-V core supply voltage (-850 and -720 devices) |
| | T11 | | | |
| | T13 | | | |
| | T15 | | | |
| | T17 | | | |
| | T19 | | | |
| | U12 | | | |
| | U14 | | | |
| | U18 | | | |
| | V11 | | | |
| | V13 | | | |
| | V19 | | | |
| | W12 | | | |
| | W14 | | | |
| **GROUND PINS** | | | | |
| $V_{SS}$ | A8 | GND | | Ground pins |
| | A11 | | | |
| | A20 | | | |
| | A23 | | | |
| | B1 | | | |
| | B29 | | | |
| | C5 | | | |
| | D1 | | | |
| | E5 | | | |
| | E7 | | | |
| | E19 | | | |
| | E25 | | | |
| | E29 | | | |
| | F4 | | | |
| | F6 | | | |
| | F8 | | | |
| | F10 | | | |
| | F12 | | | |
| | F14 | | | |
| | F16 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| $V_{SS}$ | F20 | GND | | Ground pins |
| | F22 | | | |
| | F24 | | | |
| | G1 | | | |
| | G5 | | | |
| | G7 | | | |
| | G9 | | | |
| | G11 | | | |
| | G13 | | | |
| | G15 | | | |
| | G17 | | | |
| | G19 | | | |
| | G21 | | | |
| | G23 | | | |
| | H6 | | | |
| | H24 | | | |
| | H29 | | | |
| | J7 | | | |
| | J23 | | | |
| | K2 | | | |
| | K6 | | | |
| | K24 | | | |
| | L7 | | | |
| | L11 | | | |
| | L13 | | | |
| | L15 | | | |
| | L17 | | | |
| | L19 | | | |
| | L23 | | | |
| | M6 | | | |
| | M12 | | | |
| | M14 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| $V_{SS}$ | M16 | GND | | Ground pins |
| | M18 | | | |
| | M24 | | | |
| | M26 | | | |
| | M29 | | | |
| | N2 | | | |
| | N13 | | | |
| | N15 | | | |
| | N17 | | | |
| | N19 | | | |
| | N23 | | | |
| | P7 | | | |
| | P12 | | | |
| | P14 | | | |
| | P16 | | | |
| | P18 | | | |
| | P29 | | | |
| | R2 | | | |
| | R7 | | | |
| | R11 | | | |
| | R13 | | | |
| | R15 | | | |
| | R17 | | | |
| | R19 | | | |
| | R24 | | | |
| | T6 | | | |
| | T12 | | | |
| | T14 | | | |
| | T16 | | | |
| | T18 | | | |
| | T23 | | | |
| | U7 | | | |
| | U11 | | | |
| | U13 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| $V_{SS}$ | U15 | GND | | Ground pins |
| | U17 | | | |
| | U19 | | | |
| | U24 | | | |
| | V2 | | | |
| | V6 | | | |
| | V12 | | | |
| | V14 | | | |
| | V16 | | | |
| | V18 | | | |
| | V23 | | | |
| | W7 | | | |
| | W11 | | | |
| | W13 | | | |
| | W15 | | | |
| | W17 | | | |
| | W19 | | | |
| | W24 | | | |
| | Y6 | | | |
| | Y23 | | | |
| | AA2 | | | |
| | AA7 | | | |
| | AA24 | | | |
| | AB6 | | | |
| | AB23 | | | |
| | AC7 | | | |
| | AC8 | | | |
| | AC10 | | | |
| | AC12 | | | |
| | AC14 | | | |
| | AC16 | | | |
| | AC18 | | | |

**Table 2-3. Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | IPD/IPU[2] | DESCRIPTION |
|---|---|---|---|---|
| $V_{SS}$ | AC20 | GND | | Ground pins |
| | AC22 | | | |
| | AC24 | | | |
| | AC28 | | | |
| | AD6 | | | |
| | AD13 | | | |
| | AD15 | | | |
| | AD17 | | | |
| | AD19 | | | |
| | AD21 | | | |
| | AD23 | | | |
| | AE4 | | | |
| | AE7 | | | |
| | AE16 | | | |
| | AE18 | | | |
| | AE20 | | | |
| | AE22 | | | |
| | AE24 | | | |
| | AF2 | | | |
| | AF19 | | | |
| | AF21 | | | |
| | AG13 | | | |
| | AG16 | | | |
| | AG20 | | | |
| | AG24 | | | |
| | AH1 | | | |
| | AH15 | | | |
| | AH19 | | | |
| | AH21 | | | |
| | AH25 | | | |
| | AH29 | | | |
| | AJ8 | | | |
| | AJ14 | | | |
| | AJ16 | | | |
| | AJ20 | | | |
| | AJ24 | | | |

## 2.8 Development

### 2.8.1 Development Support

In case the customer would like to develop their own features and software on the C6454 device, TI offers an extensive line of development tools for the TMS320C6000™ DSP 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 C6000™ DSP-based 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 DSP application.

**Hardware Development Tools:** Extended Development System (XDS™) Emulator (supports C6000™ DSP multiprocessor system debug) EVM (Evaluation Module)

### 2.8.2 Device Support

#### 2.8.2.1 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., **TMS**320C6454GTZ). 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:

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

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

**TMS** Fully qualified production device.

Support tool development evolutionary flow:

**TMDX** Development-support product that has not yet completed Texas Instruments internal qualification testing.

**TMDS** Fully qualified development-support product.

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

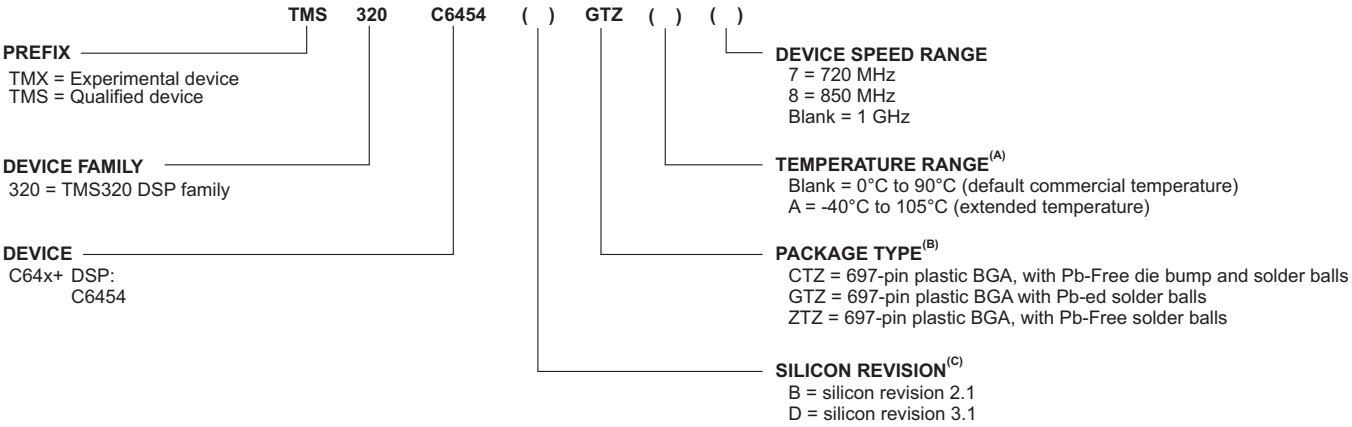
"Developmental product is intended for internal evaluation purposes."

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, GTZ), the temperature range (for example, blank is the default commercial temperature range), and the device speed range, in megahertz (for example, blank is 1000 MHz [1 GHz]). Figure 2-12 provides a legend for reading the complete device name for any TMS320C64x+™ DSP generation member.

For device part numbers and further ordering information for TMS320C6454 in the CTZ/GTZ/ZTZ package type, see the TI website (www.ti.com) or contact your TI sales representative.




A. The extended temperature "A version" devices may have different operating conditions than the commercial temperature devices. For more details, see Section 6.2, *Recommended Operating Conditions*.

B. BGA = Ball Grid Array

C. For silicon revision information, see the *TMS320C6455/54 Digital Signal Processor Silicon Errata* (literature number SPRZ234).

**Figure 2-12. TMS320C64x+™ DSP Device Nomenclature (including the TMS320C6454 DSP)**

### 2.8.2.2 Documentation Support

The following documents describe the TMS320C6454 Fixed-Point Digital Signal Processor. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

**SPRU871** ***TMS320C64x+ DSP Megamodule Reference Guide.*** Describes the TMS320C64x+ 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.

**SPRU732** ***TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide.*** Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C64x and TMS320C64x+ digital signal processors (DSPs) of the TMS320C6000 DSP family. The C64x/C64x+ DSP generation comprises fixed-point devices in the C6000 DSP platform. The C64x+ DSP is an enhancement of the C64x DSP with added functionality and an expanded instruction set.

**SPRAA84** ***TMS320C64x to TMS320C64x+ CPU Migration Guide.*** Describes migrating from the Texas Instruments TMS320C64x digital signal processor (DSP) to the TMS320C64x+ DSP. The objective of this document is to indicate differences between the two cores. Functionality in the devices that is identical is not included.

**SPRU889** ***High-Speed DSP Systems Design Reference Guide.*** Provides recommendations for meeting the many challenges of high-speed DSP system design. These recommendations include information about DSP audio, video, and communications systems for the C5000 and C6000 DSP platforms.

**SPRU971** ***TMS320C645x DSP External Memory Interface (EMIF) User's Guide.*** This document describes the operation of the external memory interface (EMIF) in the TMS320C645x DSPs.

**SPRU970** ***TMS320C645x DSP DDR2 Memory Controller User's Guide.*** This document describes the DDR2 memory controller in the TMS320C645x digital-signal processors (DSPs).

**SPRU969** ***TMS320C645x DSP Host Port Interface (HPI) User's Guide.*** This guide describes the host port interface (HPI) on the TMS320C645x digital signal processors (DSPs). The HPI enables an external host processor (host) to directly access DSP resources (including internal and external memory) using a 16-bit (HPI16) or 32-bit (HPI32) interface.

**SPRUEC6** ***TMS320C645x/C647x Bootloader User's Guide.*** This document describes the features of the on-chip Bootloader provided with the TMS320C645x/C647x digital signal processors (DSPs). Included are descriptions of the available boot modes and any interfacing requirements associated with them, instructions on generating the boot table, and information on the different versions of the Bootloader.

**SPRU966** ***TMS320C645x DSP Enhanced DMA (EDMA3) Controller User's Guide.*** This document describes the Enhanced DMA (EDMA3) Controller on the TMS320C645x digital signal processors (DSPs).

**SPRU580** ***TMS320C6000 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide.*** Describes the operation of the multichannel buffered serial port (McBSP) in the digital signal processors (DSPs) of the TMS320C6000 DSP family. The McBSP consists of a data path and a control path that connect to external devices. Separate pins for transmission and reception communicate data to these external devices. The C6000 CPU communicates to the McBSP using 32-bit-wide control registers accessible via the internal peripheral bus.

**SPRU975** ***TMS320C645x DSP EMAC/MDIO Module User's Guide.*** This document provides a functional description of the Ethernet Media Access Controller (EMAC) and Physical layer (PHY) device Management Data Input/Output (MDIO) module integrated with the TMS320C645x digital signal processors (DSPs).

**SPRUE60** ***TMS320C645x DSP Peripheral Component Interconnect (PCI) User's Guide.*** This document describes the peripheral component interconnect (PCI) port in the TMS320C645x digital signal processors (DSPs). See the PCI Specification revision 2.3 for details on the PCI interface.

**SPRUE56** ***TMS320C645x DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide.*** This document describes the operation of the software-programmable phase-locked loop (PLL) controller in the TMS320C645x digital signal processors (DSPs). The PLL controller offers flexibility and convenience by way of software-configurable multipliers and dividers to modify the input signal internally. The resulting clock outputs are passed to the TMS320C645x DSP core, peripherals, and other modules inside the TMS320C645x digital signal processors (DSPs).

**SPRU974** ***TMS320C645x DSP Inter-Integrated Circuit (I2C) Module User's Guide.*** This document describes the inter-integrated circuit (I2C) module in the TMS320C645x Digital Signal Processor (DSP). The I2C provides an interface between the TMS320C645x device and other devices compliant with Philips Semiconductors Inter-IC bus (I2C-bus) specification version 2.1 and connected by way of an I2C-bus. This document assumes the reader is familiar with the I2C-bus specification.

**SPRU968** ***TMS320C645x DSP 64-Bit Timer User's Guide.*** This document provides an overview of the 64-bit timer in the TMS320C645x digital signal processors (DSPs). The timer can be configured as a general-purpose 64-bit timer, dual general-purpose 32-bit timers, or a watchdog timer. When configured as a dual 32-bit timers, each half can operate in conjunction (chain mode) or independently (unchained mode) of each other.

**SPRU724** ***TMS320C645x DSP General-Purpose Input/Output (GPIO) User's Guide.*** This document describes the general-purpose input/output (GPIO) peripheral in the TMS320C645x digital signal processors (DSPs). The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an input, you can detect the state of the input by reading the state of an internal register. When configured as an output, you can write to an internal register to control the state driven on the output pin.

#### 2.8.2.3 Community Resources

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

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

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

# 3 Device Configuration

On the C6454 device, certain device configurations like boot mode, pin multiplexing, and endianess, are selected at device reset. The status of the peripherals (enabled/disabled) is determined after device reset. By default, the peripherals on the C6454 device are disabled and need to be enabled by software before being used.

## 3.1 Device Configuration at Device Reset

Table 3-1 describes the C6454 device configuration pins. The logic level of the AEA[19:0], ABA[1:0], and PCI_EN pins is latched at reset to determine the device configuration. The logic level on the device configuration pins can be set by using external pullup/pulldown resistors or by using some control device (e.g., FPGA/CPLD) to intelligently drive these pins. When using a control device, care should be taken to ensure there is no contention on the lines when the device is out of reset. The device configuration pins are sampled during reset and are driven after the reset is removed. To avoid contention, the control device should only drive the EMIFA pins when $\overline{\text{RESETSTAT}}$ is low.

**NOTE**

If a configuration pin must be routed out from the device and 3-stated (not driven), the internal pullup/pulldown (IPU/IPD) resistor should not be relied upon; TI recommends the use of an external pullup/pulldown resistor. For more detailed information on pullup/pulldown resistors and situations where external pullup/pulldown resistors are required, see Section 3.7, *Pullup/Pulldown Resistors*.

**Table 3-1. C6454 Device Configuration Pins (AEA[19:0], ABA[1:0], and PCI_EN)**

| CONFIGURATION PIN | NO. | IPD/ IPU(1) | FUNCTIONAL DESCRIPTION |
|---|---|---|---|
| AEA[19:16] | [N25, L26, L25, P26] | IPD | Boot Mode Selections (BOOTMODE [3:0]).<br>These pins select the boot mode for the device.<br>0000 No boot (default mode)<br>0001 Host boot (HPI)<br>0010 Reserved<br>0011 Reserved<br>0100 EMIFA 8-bit ROM boot<br>0101 Master I2C boot<br>0110 Slave I2C boot<br>0111 Host boot (PCI)<br>1000 thru 1111 Reserved<br>If selected for boot, the corresponding peripheral is automatically enabled after device reset. For more detailed information on boot modes, see Section 2.4, *Boot Sequence*.<br>CFGGP[2:0] pins must be set to 000b during reset for proper operation of the PCI boot mode. |
| AEA15 | P27 | IPD | EMIFA input clock source select (AECLKIN_SEL).<br>0 AECLKIN (default mode)<br>1 SYSCLK4 (CPU/x) Clock Rate. The SYSCLK4 clock rate is software selectable via the Software PLL1 Controller. By default, SYSCLK4 is selected as CPU/8 clock rate. |

(1) IPD = Internal pulldown, IPU = Internal pullup. For most systems, a 1-kΩ resistor can be used to oppose the IPU/IPD. For more detailed information on pullup/pulldown resistors and situations where external pullup/pulldown resistors are required, see Section 3.7, *Pullup/Pulldown Resistors*.

**Table 3-1. C6454 Device Configuration Pins (AEA[19:0], ABA[1:0], and PCI_EN) (continued)**

| CONFIGURATION PIN | NO. | IPD/ IPU(1) | FUNCTIONAL DESCRIPTION |
|---|---|---|---|
| AEA14 | R25 | IPD | HPI peripheral bus width select (HPI_WIDTH).<br>0 HPI operates in HPI16 mode (default). HPI bus is 16 bits wide; HD[15:0] pins are used and the remaining HD[31:16] pins are reserved pins in the Hi-Z state.<br>1 HPI operates in HPI32 mode. HPI bus is 32 bits wide; HD[31:0] pins are used.<br>Applies only when HPI function of HPI/PCI multiplexed pins is selected (PCI_EN pin = 0). |
| AEA13 | R27 | IPU | Device Endian mode (LENDIAN).<br>0 System operates in Big Endian mode.<br>1 System operates in Little Endian mode (default). |
| AEA12 | R28 | IPD | For proper C6454 device operation, this pin must be externally pulled down with a 1-kΩ resistor at device reset. |
| AEA11 | T25 | IPD | For proper C6454 device operation, this pin must be externally pulled down with a 1-kΩ resistor at device reset. |
| AEA[10:9] | [M25, M27] | IPD | EMAC Interface Selects (MACSEL[1:0]).<br>These pins select the interface used by the EMAC/MDIO peripheral.<br>00 10/100 EMAC/MDIO with MII Interface [default]<br>01 10/100 EMAC/MDIO with RMII Interface<br>10 10/100/1000 EMAC/MDIO with GMII Interface<br>11 10/100/1000 EMAC/MDIO with RGMII Interface<br>For more detailed information on the MAC_SEL[1:0] control pin selections, see . |
| AEA8 | P25 | IPD | PCI I2C EEPROM Auto-Initialization (PCI_EEAI).<br>PCI auto-initialization via external I2C EEPROM<br>0 PCI auto-initialization through external I2C EEPROM is disabled. The PCI peripheral uses the specified PCI default values (default).<br>1 PCI auto-initialization through external I2C EEPROM is enabled. The PCI peripheral is configured through external I2C EEPROM provided the PCI peripheral pins are enabled (PCI_EN = 1).<br>**Note:** If the PCI pin function is disabled (PCI_EN pin = 0), this pin must ***not*** be pulled up. |
| AEA7 | N27 | IPD | For proper C6454 device operation, do not oppose the IPD on this pin. |
| AEA6 | U27 | IPD | PCI Frequency Selection (PCI66).<br>Selects the operating frequency of the PCI (either 33 MHz or 66 MHz).<br>0 PCI operates at 33 MHz (default)<br>1 PCI operates at 66 MHz<br>**Note:** If the PCI pin function is disabled (PCI_EN pin = 0), this pin must ***not*** be pulled up. |
| AEA5 | U28 | IPD | McBSP1 pin function enable bit (MCBSP1_EN).<br>Selects which function is enabled on the McBSP1/GPIO multiplexed pins.<br>0 GPIO pin function enabled (default). This means all multiplexed McBSP1/GPIO pins function as GPIO pins.<br>1 McBSP1 pin function enabled. This means all multiplexed McBSP1/GPIO pins function as McBSP1 pins. |
| AEA4 | T28 | IPD | SYSCLKOUT Enable bit (SYSCLKOUT_EN).<br>Selects which function is enabled on the SYSCLK4/GP[1] muxed pin.<br>0 GP[1] pin function is enabled (default)<br>1 SYSCLK4 pin function is enabled |
| AEA3 | T27 | IPD | For proper C6454 device operation, the AEA3 pin must be pulled down to $V_{SS}$ using a 1-kΩ resistor. |
| AEA[2:0] | [T26, U26, U25] | IPD | Configuration General-Purpose Inputs (CFGGP[2:0])<br>The value of these pins is latched to the Device Status Register following device reset and is used by the on-chip bootloader for some boot modes. For more information on the boot modes, see Section 2.4, *Boot Sequence*. |

**Table 3-1. C6454 Device Configuration Pins (AEA[19:0], ABA[1:0], and PCI_EN) (continued)**

| CONFIGURATION PIN | NO. | IPD/ IPU(1) | FUNCTIONAL DESCRIPTION |
|---|---|---|---|
| PCI_EN | Y29 | IPD | PCI pin function enable bit (PCI_EN).<br>Selects which function is enabled on the HPI/PCI multiplexed pins.<br>0 HPI pin function enabled (default)<br>This means all multiplexed HPI/PCI pins function as HPI pins.<br>1 PCI pin function enabled<br>This means all multiplexed HPI/PCI pins function as PCI pins. |
| ABA0 | V26 | IPD | DDR2 Memory Controller enable (DDR2_EN).<br>0 DDR2 Memory Controller peripheral pins are disabled (default)<br>1 DDR2 Memory Controller peripheral pins are enabled |
| ABA1 | V25 | IPD | EMIFA enable (EMIFA_EN).<br>0 EMIFA peripheral pins are disabled (default)<br>1 EMIFA peripheral pins are enabled |

## 3.2 Peripheral Configuration at Device Reset

Some C6454 device peripherals share the same pins (internally multiplexed) and are mutually exclusive. Therefore, not all peripherals may be used at the same time. The device configuration pins described in Section 3.1, *Device Configuration at Device Reset*, determine which function is enabled for the multiplexed pins.

Note that when the pin function of a peripheral is disabled at device reset, the peripheral is permanently disabled and cannot be enabled until its pin function is enabled and another device reset is executed. Also, note that enabling the pin function of a peripheral does not enable the corresponding peripheral. All peripherals on the C6454 device are disabled by default, except when used for boot, and must be enabled through software before being used.

Other peripheral options like PCI clock speed and EMAC/MDIO interface mode can also be selected at device reset through the device configuration pins. The configuration selected is also fixed at device reset and cannot be changed until another device reset is executed with a different configuration selected.

The multiply factor of the PLL1 Controller is not selected through the configuration pins. The PLL1 multiply factor is set in software through the PLL1 controller registers after device reset. The PLL2 multiply factor is fixed. For more information, see Section 7.7, *PLL1 and PLL1 Controller*, and Section 7.8, *PLL2 and PLL2 Controller*.

On the C6454 device, the PCI peripheral pins are multiplexed with the HPI pins. The PCI_EN pin selects the function for the HPI/PCI multiplexed pins. The PCI66, PCI_EEAI, and HPI_WIDTH control other functions of the PCI and HPI peripherals. Table 3-2 describes the effect of the PCI_EN, PCI66, PCI_EEAI, and HPI_WIDTH configuration pins.

**Table 3-2. PCI_EN, PCI66, PCI_EEAI, and HPI_WIDTH Peripheral Selection (HPI and PCI)**

| CONFIGURATION PIN SETTING(1) | | | | PERIPHERAL FUNCTION SELECTED | | | |
|---|---|---|---|---|---|---|---|
| PCI_EN PIN [Y29] | PCI66 AEA6 PIN [U27] | PCI_EEAI AEA8 PIN [P25](1) | HPI_WIDTH AEA14 PIN [R25] | HPI DATA LOWER | HPI DATA UPPER | 32-BIT PCI (66-/33-MHz) | PCI AUTO-INIT |
| 0 | 0 | 0 | 0 | Enabled | Hi-Z | Disabled | N/A |
| 0 | 0 | 0 | 1 | Enabled | Enabled | Disabled | N/A |
| 1 | 1 | 1 | X | Disabled | | Enabled (66 MHz) | Enabled (via External I2C EEPROM) |
| 1 | 1 | 0 | X | Disabled | | | Disabled |

(1) PCI_EEAI is latched at reset as a configuration input. If PCI_EEAI is set as one, then default values are loaded from an external I2C EEPROM.

**Table 3-2. PCI_EN, PCI66, PCI_EEAI, and HPI_WIDTH Peripheral Selection (HPI and PCI) (continued)**

| CONFIGURATION PIN SETTING(1) | | | | PERIPHERAL FUNCTION SELECTED | | | |
|---|---|---|---|---|---|---|---|
| PCI_EN PIN [Y29] | PCI66 AEA6 PIN [U27] | PCI_EEAI AEA8 PIN [P25](1) | HPI_WIDTH AEA14 PIN [R25] | HPI DATA LOWER | HPI DATA UPPER | 32-BIT PCI (66-/33-MHz) | PCI AUTO-INIT |
| 1 | 0 | 0 | X | Disabled | | Enabled (33 MHz) | Disabled (default values) |
| 1 | 0 | 1 | X | Disabled | | | Enabled (via External I2C EEPROM) |

The MAC_SEL[1:0] configuration pins (AEA[10:9) control which interface is used by the EMAC/MDIO. Table 3-3 describes the effect of the MACSEL[1:0] configuration pins.

**Table 3-3. MAC_SEL[1:0] Peripheral Selection (EMAC)**

| MAC_SEL[1:0]/ AEA[10:9] PINS [M25, M27] CONFIGURATION PIN SETTING | EMAC/MDIO PERIPHERAL FUNCTION SELECTED |
|---|---|
| 00b | 10/100 EMAC/MDIO with MII Interface [default] |
| 01b | 10/100 EMAC/MDIO with RMII Interface |
| 10b | 10/100/1000 EMAC/MDIO with GMII Interface |
| 11b | 10/100/1000 EMAC/MDIO with RGMII Interface(1) |

(1) RGMII interface requires a 1.5-/1.8-V I/O supply.

## 3.3 Peripheral Selection After Device Reset

On the C6454 device, peripherals can be in one of several states. These states are listed in Table 3-4.

**Table 3-4. Peripheral States**

| STATE | DESCRIPTION | PERIPHERALS THAT CAN BE IN THIS STATE |
|---|---|---|
| Static powerdown | Peripheral pin function has been completely disabled through the device configuration pins. Peripheral is held in reset and clock is turned off. | HPI<br>PCI<br>McBSP1<br>EMAC/MDIO<br>EMIFA<br>DDR2 Memory Controller |
| Disabled | Peripheral is held in reset and clock is turned off. Default state for all peripherals not in static powerdown mode. | I2C<br>Timer 0<br>Timer 1<br>GPIO<br>EMAC/MDIO<br>McBSP0<br>McBSP1<br>HPI<br>PCI |
| Enabled | Clock to the peripheral is turned on and the peripheral is taken out of reset. | I2C<br>Timer 0<br>Timer 1<br>GPIO<br>MDIO<br>EMAC/MDIO<br>McBSP0<br>McBSP1<br>HPI<br>PCI<br>EMIFA<br>DDR2 Memory Controller |

**Table 3-4. Peripheral States (continued)**

| STATE | DESCRIPTION | PERIPHERALS THAT CAN BE IN THIS STATE |
|---|---|---|
| Enable in progress | Not a user-programmable state. This is an intermediate state when transitioning from an disabled state to an enabled state. | All peripherals that can be in an enabled state. |

Following device reset, all peripherals that are not in the static powerdown state are in the disabled state by default. Peripherals used for boot such as HPI and PCI are enabled automatically following a device reset.

Peripherals are only allowed certain transitions between states (see Figure 3-1).

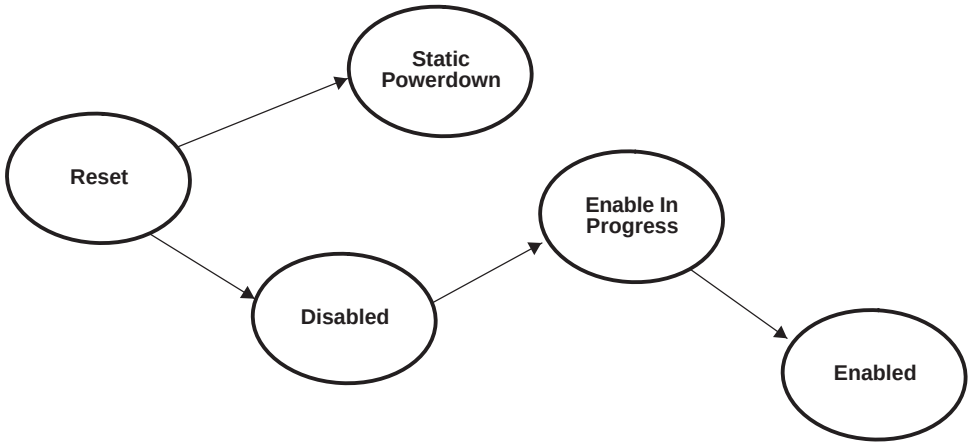



**Figure 3-1. Peripheral Transitions Between States**

Figure 3-2 shows the flow needed to change the state of a given peripheral on the C6454 device.

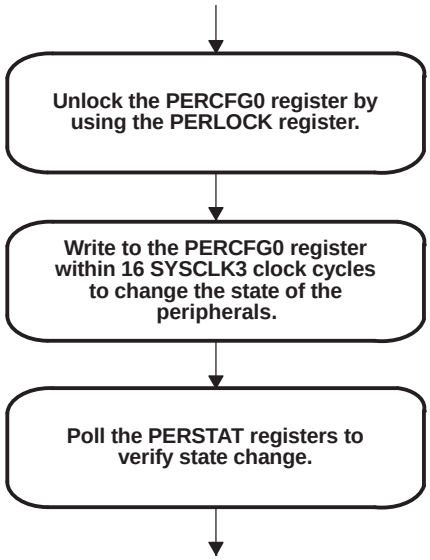



**Figure 3-2. Peripheral State Change Flow**

A 32-bit key (value = 0x0F0A 0B00) must be written to the Peripheral Lock register (PERLOCK) in order to allow access to the PERCFG0 register. Writes to the PERCFG1 register can be done directly without going through the PERLOCK register.

**NOTE**

The instructions that write to the PERLOCK and PERCFG0 registers must be in the same fetch packet if code is being executed from external memory. If the instructions are in different fetch packets, fetching the second instruction from external memory may stall the instruction long enough such that PERCFG0 register will be locked before the instruction is executed.

## 3.4 Device State Control Registers

The C6454 device has a set of registers that are used to control the status of its peripherals. These registers are shown in Table 3-5 and described in the next sections.

**NOTE**

The device state control registers can only be accessed using the CPU or the emulator.

**Table 3-5. Device State Control Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02AC 0000 | - | Reserved |
| 02AC 0004 | PERLOCK | Peripheral Lock Register |
| 02AC 0008 | PERCFG0 | Peripheral Configuration Register 0 |
| 02AC 000C | - | Reserved |
| 02AC 0010 | - | Reserved |
| 02AC 0014 | PERSTAT0 | Peripheral Status Register 0 |
| 02AC 0018 | PERSTAT1 | Peripheral Status Register 1 |
| 02AC 001C - 02AC 001F | - | Reserved |
| 02AC 0020 | EMACCFG | EMAC Configuration Register |
| 02AC 0024 - 02AC 002B | - | Reserved |
| 02AC 002C | PERCFG1 | Peripheral Configuration Register 1 |
| 02AC 0030 - 02AC 0053 | - | Reserved |
| 02AC 0054 | EMUBUFPD | Emulator Buffer Powerdown Register |
| 02AC 0058 | - | Reserved |

### 3.4.1 Peripheral Lock Register Description

When written with correct 32-bit key (0x0F0A0B00), the Peripheral Lock Register (PERLOCK) allows one write to the PERCFG0 register within 16 SYSCLK3 cycles.

**NOTE**

The instructions that write to the PERLOCK and PERCFG0 registers must be in the same fetch packet if code is being executed from external memory. If the instructions are in different fetch packets, fetching the second instruction from external memory may stall the instruction long enough such that PERCFG0 register will be locked before the instruction is executed.

| 31 | 0 |
|---|---|
| LOCKVAL | |
| R/W-F0F0 F0F0 | |

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

**Figure 3-3. Peripheral Lock Register (PERLOCK) - 0x02AC 0004**

**Table 3-6. Peripheral Lock Register (PERLOCK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:0 | LOCKVAL | | When programmed with 0x0F0A 0B00 allows one write to the PERCFG0 register within 16 SYSCLK3 clock cycles. |

### 3.4.2 Peripheral Configuration Register 0 Description

The Peripheral Configuration Register (PERCFG0) is used to change the state of the peripherals. One write is allowed to this register within 16 SYSCLK3 cycles after the correct key is written to the PERLOCK register.

**NOTE**

The instructions that write to the PERLOCK and PERCFG0 registers must be in the same fetch packet if code is being executed from external memory. If the instructions are in different fetch packets, fetching the second instruction from external memory may stall the instruction long enough that the PERCFG0 register is locked before the instruction is executed.

| 31 | | | | | | | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R/W-0 | | | | | | | |
| 23 | | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | PCICTL | Reserved | HPICTL | Reserved | McBSP1CTL |
| R/W-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | McBSP0CTL | Reserved | I2CCTL | Reserved | GPIOCTL | Reserved | TIMER1CTL |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | | | 0 |
| Reserved | TIMER0CTL | Reserved | EMACCTL | Reserved | | | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | | | |

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

**Figure 3-4. Peripheral Configuration Register 0 (PERCFG0) - 0x02AC 0008**

**Table 3-7. Peripheral Configuration Register 0 (PERCFG0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:21 | Reserved | | Reserved. |
| 20 | PCICTL | | Mode control for PCI. This bit defaults to 1 when host boot is used (BOOTMODE[3:0] = 0111b). |
| | | 0 | Set PCI to disabled mode |
| | | 1 | Set PCI to enabled mode |
| 19 | Reserved | | Reserved. |
| 18 | HPICTL | | Mode control for HPI. This bit defaults to 1 when host boot is used (BOOTMODE[3:0] = 0001b). |
| | | 0 | Set HPI to disabled mode |
| | | 1 | Set HPI to enabled mode |
| 17 | Reserved | 1 | Reserved. |
| 16 | McBSP1CTL | | Mode control for McBSP1 |
| | | 0 | Set McBSP1 to disabled mode |
| | | 1 | Set McBSP1 to enabled mode |
| 15 | Reserved | | Reserved. |
| 14 | McBSP0CTL | | Mode control for McBSP0 |
| | | 0 | Set McBSP0 to disabled mode |
| | | 1 | Set McBSP0 to enabled mode |
| 13 | Reserved | | Reserved. |

**Table 3-7. Peripheral Configuration Register 0 (PERCFG0) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 12 | I2CCTL | | Mode control for I2C |
| | | 0 | Set I2C to disabled mode |
| | | 1 | Set I2C to enabled mode |
| 11 | Reserved | | Reserved. |
| 10 | GPIOCTL | | Mode control for GPIO |
| | | 0 | Set GPIO to disabled mode |
| | | 1 | Set GPIO to enabled mode |
| 9 | Reserved | | Reserved. |
| 8 | TIMER1CTL | | Mode control for Timer 1 |
| | | 0 | Set Timer 1 to disabled mode |
| | | 1 | Set Timer 1 to enabled mode |
| 7 | Reserved | | Reserved. |
| 6 | TIMER0CTL | | Mode control for Timer 0 |
| | | 0 | Set Timer 0 to disabled mode |
| | | 1 | Set Timer 0 to enabled mode |
| 5 | Reserved | | Reserved. |
| 4 | EMACCTL | | Mode control for EMAC/MDIO |
| | | 0 | Set EMAC/MDIO to disabled mode |
| | | 1 | Set EMAC/MDIO to enabled mode |
| 3:0 | Reserved | | Reserved. |

### *3.4.3 Peripheral Configuration Register 1 Description*

The Peripheral Configuration Register (PERCFG1) is used to enable the EMIFA and DDR2 Memory Controller. EMIFA and the DDR2 Memory Controller do not have corresponding status bits in the Peripheral Status Registers. The EMIFA and DDR2 Memory Controller peripherals can be used within 16 SYSCLK3 cycles after EMIFACTL and DDR2CTL are set to 1. Once EMIFACTL and DDR2CTL are set to 1, they cannot be set to 0. Note that if the DDR2 Memory Controller and EMIFA are disabled at reset through the device configuration pins (DDR2.EN[ABA0] and EMIFA[ABA1]), they cannot be enabled through the PERCFG1 register.

| 31 | 8 |
|---|---|
| Reserved | |
| R-0x00 | |

| 7 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | DDR2CTL | EMIFACTL |
| R-0x00 | | R/W-0 | R/W-0 |

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

**Figure 3-5. Peripheral Configuration Register 1 (PERCFG1) - 0x02AC 002C**

**Table 3-8. Peripheral Configuration Register 1 (PERCFG1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:2 | Reserved | | Reserved. |
| 1 | DDR2CTL | | Mode Control for DDR2 Memory Controller. Once this bit is set to 1, it cannot be changed to 0. |
| | | 0 | Set DDR2 to disabled |
| | | 1 | Set DDR2 to enabled |

**Table 3-8. Peripheral Configuration Register 1 (PERCFG1) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | EMIFACTL | | Mode control for EMIFA. Once this bit is set to 1, it cannot be changed to 0. This bit defaults to 1 if EMIFA 8-bit ROM boot is used (BOOTMODE[3:0] = 0100b). |
| | | 0 | Set EMIFA to disabled |
| | | 1 | Set EMIFA to enabled |

### 3.4.4 Peripheral Status Registers Description

The Peripheral Status Registers (PERSTAT0 and PERSTAT1) show the status of the C6454 device peripherals.

| 31 | 30 | 29 | 27 | 26 | 24 |
|---|---|---|---|---|---|
| Reserved | | HPISTAT | | McBSP1STAT | |
| R-0 | | R-0 | | R-0 | |

| 23 | 21 | 20 | 18 | 17 | 16 |
|---|---|---|---|---|---|
| McBSP0STAT | | I2CSTAT | | GPIOSTAT | |
| R-0 | | R-0 | | R-0 | |

| 15 | 14 | 12 | 11 | 9 | 8 |
|---|---|---|---|---|---|
| GPIOSTAT | TIMER1STAT | | TIMER0STAT | | EMACSTAT |
| R-0 | R-0 | | R-0 | | R-0 |

| 7 | 6 | 5 | 0 |
|---|---|---|---|
| EMACSTAT | | Reserved | |
| R-0 | | R-0 | |

**LEGEND:** R = Read only; -*n* = value after reset

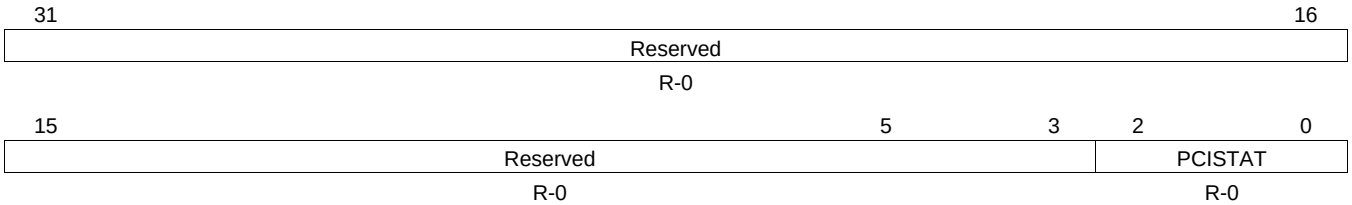
**Figure 3-6. Peripheral Status Register 0 (PERSTAT0) - 0x02AC 0014**

**Table 3-9. Peripheral Status Register 0 (PERSTAT0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:30 | Reserved | | Reserved. |
| 29:27 | HPISTAT | | HPI status |
| | | 000 | HPI is in the disabled state |
| | | 001 | HPI is in the enabled state |
| | | 011 | HPI is in the static powerdown state |
| | | 100 | HPI is in the disable in progress state |
| | | 101 | HPI is in the enable in progress state |
| | | Others | Reserved |
| 26:24 | McBSP1STAT | | McBSP1 status |
| | | 000 | McBSP1 is in the disabled state |
| | | 001 | McBSP1 is in the enabled state |
| | | 011 | McBSP1 is in the static powerdown state |
| | | 100 | McBSP1 is in the disable in progress state |
| | | 101 | McBSP1 is in the enable in progress state |
| | | Others | Reserved |
| 23:21 | McBSP0STAT | | McBSP0 status |
| | | 000 | McBSP0 is in the disabled state |
| | | 001 | McBSP0 is in the enabled state |
| | | 011 | McBSP0 is in the static powerdown state |
| | | 100 | McBSP0 is in the disable in progress state |
| | | 101 | McBSP0 is in the enable in progress state |
| | | Others | Reserved |

**Table 3-9. Peripheral Status Register 0 (PERSTAT0) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 20:18 | I2CSTAT | | I2C status |
| | | 000 | I2C is in the disabled state |
| | | 001 | I2C is in the enabled state |
| | | 011 | I2C is in the static powerdown state |
| | | 100 | I2C is in the disable in progress state |
| | | 101 | I2C is in the enable in progress state |
| | | Others | Reserved |
| 17:15 | GPIOSTAT | | GPIO status |
| | | 000 | GPIO is in the disabled state |
| | | 001 | GPIO is in the enabled state |
| | | 011 | GPIO is in the static powerdown state |
| | | 100 | GPIO is in the disable in progress state |
| | | 101 | GPIO is in the enable in progress state |
| | | Others | Reserved |
| 14:12 | TIMER1STAT | | Timer1 status |
| | | 000 | Timer1 is in the disabled state |
| | | 001 | Timer1 is in the enabled state |
| | | 011 | Timer1 is in the static powerdown state |
| | | 100 | Timer1 is in the disable in progress state |
| | | 101 | Timer1 is in the enable in progress state |
| | | Others | Reserved |
| 11:9 | TIMER0STAT | | Timer0 status |
| | | 000 | Timer0 is in the disabled state |
| | | 001 | Timer0 is in the enabled state |
| | | 011 | Timer0 is in the static powerdown state |
| | | 100 | Timer0 is in the disable in progress state |
| | | 101 | Timer0 is in the enable in progress state |
| | | Others | Reserved |
| 8:6 | EMACSTAT | | EMAC/MDIO status |
| | | 000 | EMAC/MDIO is in the disabled state |
| | | 001 | EMAC/MDIO is in the enabled state |
| | | 011 | EMAC/MDIO is in the static powerdown state |
| | | 100 | EMAC/MDIO is in the disable in progress state |
| | | 101 | EMAC/MDIO is in the enable in progress state |
| | | Others | Reserved |
| 5:0 | Reserved | | Reserved |




**LEGEND:** R = Read only; -*n* = value after reset

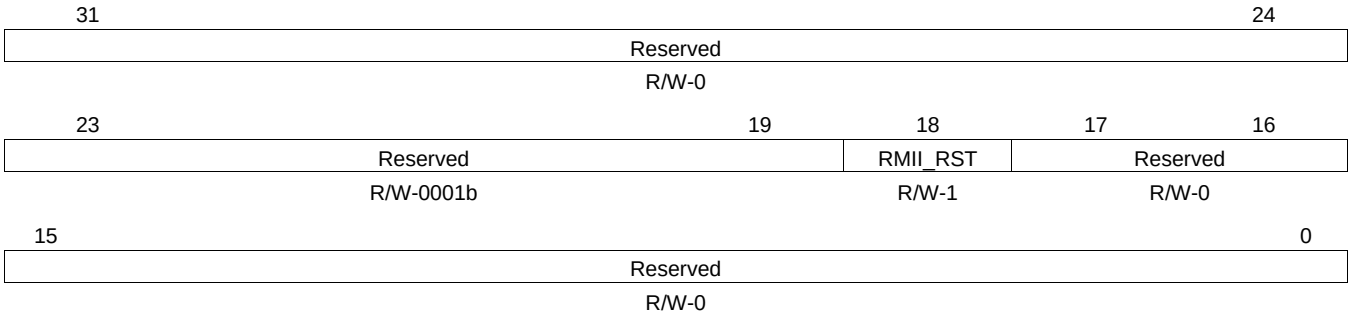
**Figure 3-7. Peripheral Status Register 1 (PERSTAT1) - 0x02AC 0018**

**Table 3-10. Peripheral Status Register 1 (PERSTAT1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:3 | Reserved | | Reserved |
| 2:0 | PCISTAT | | PCI status |
| | | 000 | PCI is in the disabled state |
| | | 001 | PCI is in the enabled state |
| | | 011 | PCI is in the static powerdown state |
| | | 101 | PCI is in the enable in progress state |
| | | Others | Reserved |

### 3.4.5 EMAC Configuration Register (EMACCFG) Description

The EMAC Configuration Register (EMACCFG) is used to assert and deassert the reset of the Reduced Media Independent Interface (RMII) logic of the EMAC. For more details on how to use this register, see Section 7.14, *Ethernet MAC (EMAC)*.




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

**Figure 3-8. EMAC Configuration Register (EMACCFG) - 0x02AC 0020**

**Table 3-11. EMAC Configuration Register (EMACCFG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:19 | Reserved | | Reserved. Writes to this register must keep the default values of these bits. |
| 18 | RMII_RST | | RMII reset bit. This bit is used to reset the RMII logic of the EMAC. |
| | | 0 | RMII logic reset is released. |
| | | 1 | RMII logic reset is asserted. |
| 17:0 | Reserved | | Reserved. Writes to this register must keep this bit as 0. |

### 3.4.6 Emulator Buffer Powerdown Register (EMUBUFPD) Description

The Emulator Buffer Powerdown Register (EMUBUFPD) is used to control the state of the pin buffers of emulator pins EMU[18:2]. These buffers can be powered down if the device trace feature is not needed.

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 1 | 0 |
|---|---|---|
| Reserved | | EMUCTL |
| R-0 | | R/W-0 |

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

**Figure 3-9. Emulator Buffer Powerdown Register (EMUBUFPD) - 0x02AC 0054**

**Table 3-12. Emulator Buffer Powerdown Register (EMUBUFPD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:1 | Reserved | | Reserved |
| 0 | EMUCTL | | Buffer powerdown for EMU[18:2] pins |
| | | 0 | Power-up buffers |
| | | 1 | Power-down buffers |

## 3.5 Device Status Register Description

The device status register depicts the device configuration selected upon device reset. Once set, these bits will remain set until a device reset. For the actual register bit names and their associated bit field descriptions, see Figure 3-10 and Table 3-13.

Note that enabling or disabling peripherals through the Peripheral Configuration Registers (PERCFG0 and PERCFG1) does not affect the DEVSTAT register. To determine the status of peripherals following writes to the PERCFG0 and PERCFG1 registers, read the Peripherals Status Registers (PERSTAT0 and PERSTAT1).

| 31 | | | | | | | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0000 0000 | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | EMIFA_EN | DDR2_EN | PCI_EN | CFGGP2 | CFGGP1 | CFGGP0 | Reserved |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SYSCLKOUT_EN | MCBSP1_EN | PCI66 | Reserved | PCI_EEAI | MAC_SEL1 | MAC_SEL0 | Reserved |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-1 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | LENDIAN | HPI_WIDTH | AECLKINSEL | BOOTMODE3 | BOOTMODE2 | BOOTMODE1 | BOOTMODE0 |
| R-0 | R-1 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

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

**Note:** The default values of the fields in the DEVSTAT register are latched from device configuration pins, as described in Section 3.1, *Device Configuration at Device Reset*. The default values shown here correspond to the setting dictated by the internal pullup or pulldown resistor.

**Figure 3-10. Device Status Register (DEVSTAT) - 0x02A8 0000**

**Table 3-13. Device Status Register (DEVSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:23 | Reserved | | Reserved. Read-only, writes have no effect. |
| 22 | EMIFA_EN | | EMIFA Enable (EMIFA_EN) status bit<br>Shows the status of whether the EMIFA peripheral pins are enabled/disabled. |
| | | 0 | EMIFA peripheral pins are disabled (default) |
| | | 1 | EMIFA peripheral pins are enabled |
| 21 | DDR2_EN | | DDR2 Memory Controller Enable (DDR2_EN) status bit<br>Shows the status of whether the DDR2 Memory Controller peripheral pins are enabled/disabled. |
| | | 0 | DDR2 Memory Controller peripheral pins are disabled (default) |
| | | 1 | DDR2 Memory Controller peripheral pins are enabled |
| 20 | PCI_EN | | PCI Enable (PCI_EN) status bit<br>Shows the status of which function is enabled on the HPI/PCI multiplexed pins. |
| | | 0 | HPI pin functions are enabled (default) |
| | | 1 | PCI pin functions are enabled |
| 19:17 | CFGGP[2:0] | | Used as General-Purpose inputs for configuration purposes.<br>These pins are latched at reset. These values can be used by S/W routines for boot operations. |
| 16 | Reserved | | Reserved. Read-only, writes have no effect. |
| 15 | SYSCLKOUT_EN | | SYSCLKOUT Enable (SYSCLKOUT_EN) status bit<br>Shows the status of which function is enabled on the SYSCLK4/GP[1] muxed pin. |
| | | 0 | GP[1] pin function of the SYSCLK4/GP[1] pin enabled (default) |
| | | 1 | SYSCLK4 pin function of the SYSCLK4/GP[1] pin enabled |

**Table 3-13. Device Status Register (DEVSTAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 14 | MCBSP1_EN | | McBSP1 Enable (MCBSP1_EN) status bit<br>Shows the status of which function is enabled on the McBSP1/GPIO muxed pins. |
| | | 0 | GPIO pin functions enabled (default) |
| | | 1 | McBSP1 pin functions enabled |
| 13 | PCI66 | | PCI Frequency Selection (PCI66) status bit<br>Shows the PCI operating frequency selected at reset. |
| | | 0 | PCI operates at 33 MHz (default) |
| | | 1 | PCI operates at 66 MHz |
| 11 | PCI_EEAI | | PCI I2C EEPROM Auto-Initialization (PCI_EEAI) status bit<br>Shows whether the PCI auto-initialization via external I2C EEPROM is enabled/disabled. |
| | | 0 | PCI auto-initialization through external I2C EEPROM is disabled; the PCI peripheral uses the specified PCI default values (default). |
| | | 1 | PCI auto-initialization through external I2C EEPROM is enabled; the PCI peripheral is configured through external I2C EEPROM provided the PCI peripheral pin is enabled (PCI_EN = 1). |
| 10:9 | MACSEL[1:0] | | EMAC Interface Select (MACSEL[1:0]) status bits<br>Shows which EMAC interface mode has been selected. |
| | | 00 | 10/100 EMAC/MDIO with MII Interface (default) |
| | | 01 | 10/100 EMAC/MDIO with RMII Interface |
| | | 10 | 10/100/1000 EMAC/MDIO with GMII Interface |
| | | 11 | 10/100/1000 EMAC/MDIO with RGMII Mode Interface<br>[RGMII interface requires a 1.8-V or 1.5-V I/O supply] |
| 8:7 | Reserved | | Reserved. Read-only, writes have no effect. |
| 6 | LENDIAN | | Device Endian mode (LENDIAN)<br>Shows the status of whether the system is operating in Big Endian mode or Little Endian mode (default). |
| | | 0 | System is operating in Big Endian mode |
| | | 1 | System is operating in Little Endian mode (default) |
| 5 | HPI_WIDTH | | HPI bus width control bit.<br>Shows the status of whether the HPI bus operates in 32-bit mode or in 16-bit mode (default). |
| | | 0 | HPI operates in 16-bit mode. (default) |
| | | 1 | HPI operates in 32-bit mode |
| 4 | AECLKINSEL | | EMIFA input clock select<br>Shows the status of what clock mode is enabled or disabled for EMIFA. |
| | | 0 | AECLKIN (default mode) |
| | | 1 | SYSCLK4 (CPU/x) Clock Rate. The SYSCLK4 clock rate is software selectable via the PLL1 Controller. By default, SYSCLK4 is selected as CPU/8 clock rate. |
| 3:0 | BOOTMODE[3:0] | | Boot mode configuration bits<br>Shows the status of what device boot mode configuration is operational.<br>BOOTMODE[3:0]<br>[**Note:** if selected for boot, the corresponding peripheral is automatically enabled after device reset.] |
| | | 0000 | No boot (default mode) |
| | | 0001 | Host boot (HPI) |
| | | 0010 | Reserved |
| | | 0011 | Reserved |
| | | 0100 | EMIFA 8-bit ROM boot |
| | | 0101 | Master I2C boot |
| | | 0110 | Slave I2C boot |
| | | 0111 | Host boot (PCI) |
| | | 1000 thru 1111 | Reserved<br><br>For more detailed information on the boot modes, see Section 2.4, *Boot Sequence*. |

## 3.6 JTAG ID (JTAGID) Register Description

The JTAG ID register is a read-only register that identifies to the customer the JTAG/Device ID. For the C6454 device, the JTAG ID register resides at address location 0x02A8 0008. For the actual register bit names and their associated bit field descriptions, see Figure 3-11 and Table 3-14.

| 31 28 | 27 12 | 11 1 | 0 |
|---|---|---|---|
| VARIANT (4-bit) | PART NUMBER (16-bit) | MANUFACTURER (11-bit) | LSB |
| R-n | R-0000 0000 1000 1010b | 0000 0010 111b | R-1 |

**LEGEND:** R = Read only; -*n* = value after reset

**Figure 3-11. JTAG ID (JTAGID) Register - 0x02A8 0008**

**Table 3-14. JTAG ID (JTAGID) Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:28 | VARIANT | | Variant (4-Bit) value. The value of this field depends on the silicon revision being used. For more information, see the *TMS320C6455/54 Digital Signal Processor Silicon Errata* (literature number SPRZ234).<br>**Note:** the VARIANT field may be invalid if no CLKIN1 signal is applied. |
| 27:12 | PART NUMBER | | Part Number (16-Bit) value. C6454 device value: 0000 0000 1000 1010b. |
| 11:1 | MANUFACTURER | | Manufacturer (11-Bit) value. C6454 device value: 0000 0010 111b. |
| 0 | LSB | | LSB. This bit is read as a "1" for the C6454 device. |

## 3.7 Pullup/Pulldown Resistors

Proper board design should ensure that input pins to the C6454 device always be at a valid logic level and not floating. This may be achieved via pullup/pulldown resistors. The C6454 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:

- *Device Configuration Pins*: If the pin is both routed out and 3-stated (not driven), an external pullup/pulldown resistor ***must*** be used, 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 device configuration pins (listed in Table 3-1), 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 configuration pins. In addition, applying external pullup/pulldown resistors on the device 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 $DV_{DD}$ rail.

For most systems, a 1-kΩ 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.

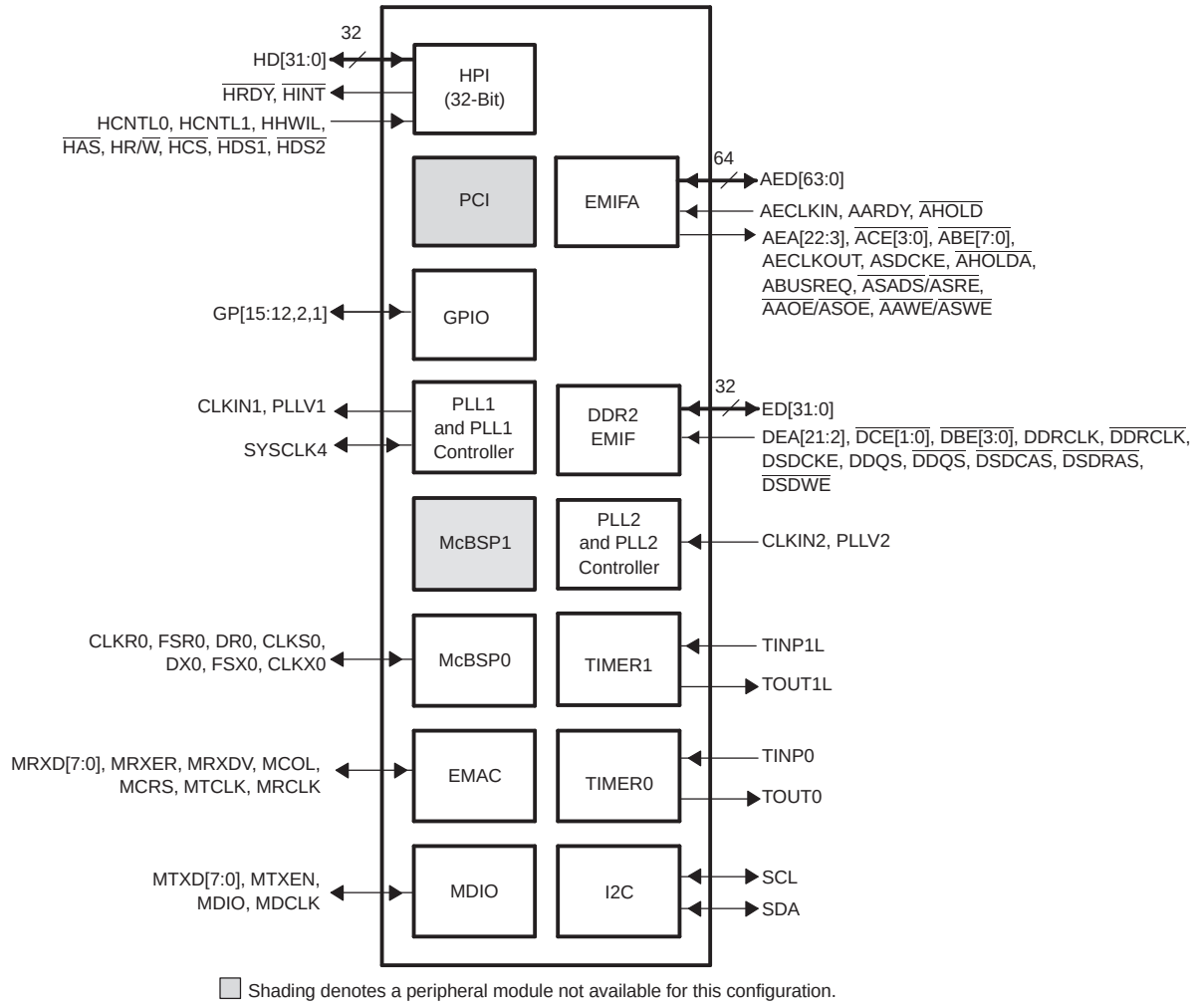
For most systems, a 20-kΩ resistor can be used to compliment the IPU/IPD on the device configuration pins while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.

For more detailed information on input current ($I_I$), and the low-/high-level input voltages ($V_{IL}$ and $V_{IH}$) for the C6454 device, see Section 6.3, *Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Case Temperature*.

To determine which pins on the C6454 device include internal pullup/pulldown resistors, see Table 2-3, *Terminal Functions*.

## 3.8 Configuration Examples

Figure 3-12 and Figure 3-13 illustrate examples of peripheral selections/options that are configurable on the C6454 device.

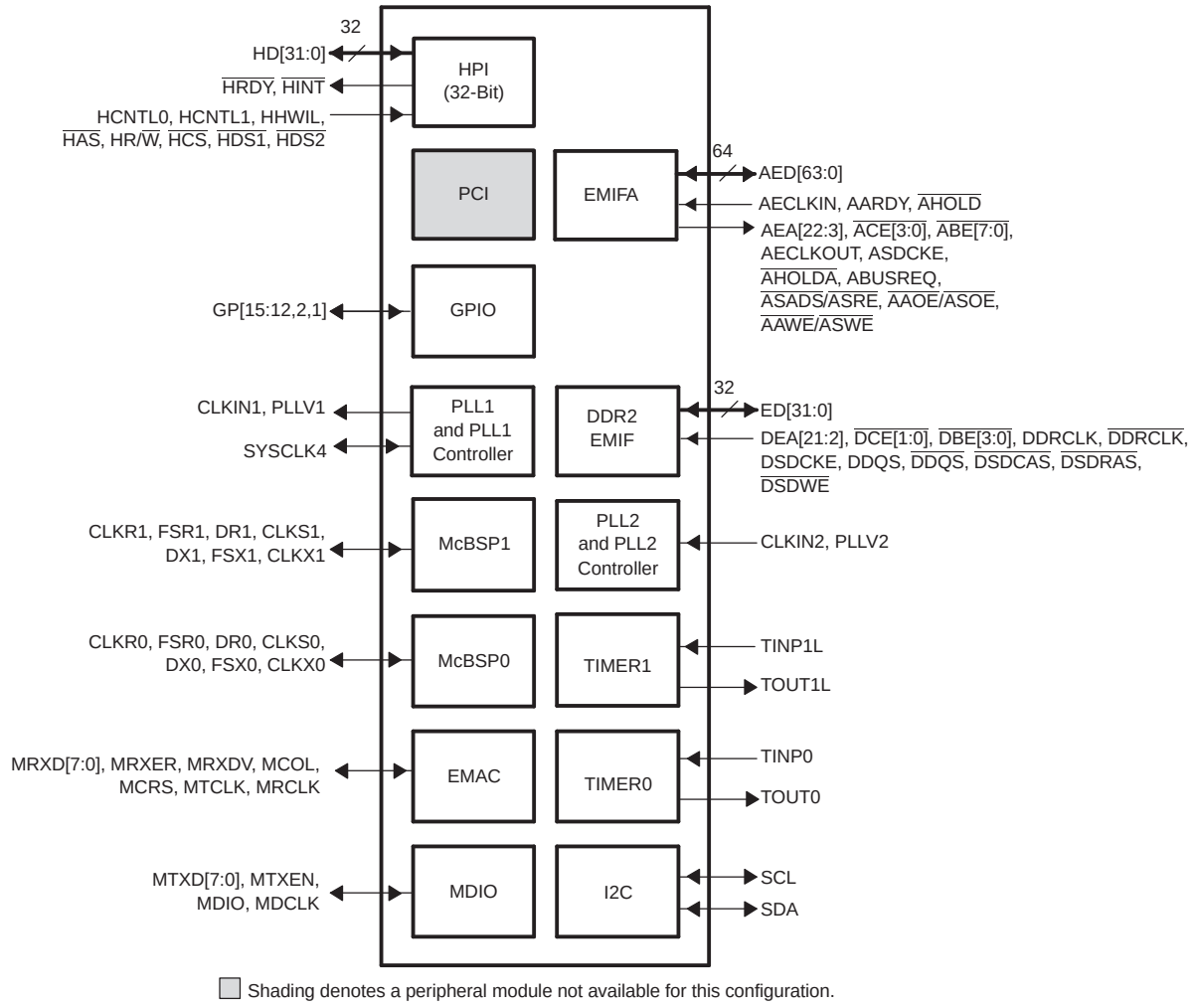



DEVSTAT Register: 0x0061 8161

PCI_EN = 0 (PCI disabled, default)
ABA1 (EMIFA_EN) = 1(EMIFA enabled)
ABA0 (DDR2_EN) = 1 (DDR2 Memory Controller enabled)

AEA[19:16] (BOOTMODE[3:0]) = 0001, (HPI Boot)
AEA[15] (AECLKIN_SEL) = 0, (AECLKIN, default)
AEA[14] (HPI_WIDTH) = 1, (HPI, 32-bit Operation)
AEA[13] (LENDIAN) = IPU, (Little Endian Mode, default)
AEA[12] = 0, (do not oppose IPD)
AEA[11] = 0, (do not oppose IPD)
AEA[10:9] (MACSEL[1:0]) = 00, (10/100 MII Mode)
AEA[8] (PCI_EEAI) = 0, (PCI I2C EEPROM Auto-Init disabled, default)
AEA[7] = 0, (do not oppose IPD)
AEA[6] (PCI66) = 0, (PCI 33 MHz [default, don't care])
AEA[5] (MCBSP1_EN) = 0, (McBSP1 disabled, default)
AEA[4] (SYSCLKOUT_EN) = 1, (SYSCLK4 pin function)
AEA[3] = 0, (do not oppose IPD)
AEA[2:0] (CFGGP[2:0]) = 000, (default)

**Figure 3-12. Configuration Example A (McBSP + HPI32 + I2C + EMIFA + DDR2 Memory Controller + TIMERS + EMAC (MII) + MDIO)**




DEVSTAT Register: 0x0061 C161

PCI_EN = 0 (PCI disabled, default)
ABA1 (EMIFA_EN) = 1(EMIFA enabled)
ABA0 (DDR2_EN) = 1 (DDR2 Memory Controller enabled)

AEA[19:16] (BOOTMODE[3:0]) = 0001, (HPI Boot)
AEA[15] (AECLKIN_SEL) = 0, (AECLKIN, default)
AEA[14] (HPI_WIDTH) = 1, (HPI, 32-bit Operation)
AEA[13] (LENDIAN) = IPU, (Little Endian Mode, default)
AEA[12] = 0, (do not oppose IPD)
AEA[11] = 0, (do not oppose IPD)
AEA[10:9] (MACSEL[1:0]) = 00, (10/100 MII Mode)
AEA[8] (PCI_EEAI) = 0, (PCI I2C EEPROM Auto-Init disabled, default)
AEA[7] = 0, (do not oppose IPD)
AEA[6] (PCI66) = 0, (PCI 33 MHz [default, don't care])
AEA[5] (MCBSP1_EN) = 1, (McBSP1 enabled)
AEA[4] (SYSCLKOUT_EN) = 1, (SYSCLK4 pin function)
AEA[3] = 0, (do not oppose IPD)
AEA[2:0] (CFGGP[2:0]) = 000, (default)

**Figure 3-13. Configuration Example B (2 McBSPs + HPI32 + I2C + EMIFA + DDR2 Memory Controller + TIMERS + EMAC (GMII) + MDIO**

# 4 System Interconnect

On the C6454 device, the C64x+ Megamodule, the EDMA3 transfer controllers, and the system peripherals are interconnected through two switch fabrics. The switch fabrics allow for low-latency, concurrent data transfers between master peripherals and slave peripherals. The switch fabrics also allow for seamless arbitration between the system masters when accessing system slaves.

## 4.1 Internal Buses, Bridges, and Switch Fabrics

Two types of buses exist in the C6454 device: data buses and configuration buses. Some C6454 device peripherals have both a data bus and a configuration bus interface, while others only have one type of interface. Furthermore, the bus interface width and speed varies from peripheral to peripheral.

Configuration buses are mainly used to access the register space of a peripheral and the data buses are used mainly for data transfers. However, in some cases, the configuration bus is also used to transfer data. For example, data is transferred to the McBSP via its configuration bus. Similarly, the data bus can also be used to access the register space of a peripheral. For example, the EMIFA and DDR2 memory controller registers are accessed through their data bus interface.

The C64x+ Megamodule, the EDMA3 traffic controllers, and the various system peripherals can be classified into two categories: masters and slaves. Masters are capable of initiating read and write transfers in the system and do not rely on the EDMA3 for their data transfers. Slaves on the other hand rely on the EDMA3 to perform transfers to and from them. Masters include the EDMA3 traffic controllers and PCI. Slaves include the McBSP and I2C.

The C6454 device contains two switch fabrics through which masters and slaves communicate. The data switch fabric, known as the data switched central resource (SCR), is a high-throughput interconnect mainly used to move data across the system (for more information, see Section 4.2). The data SCR connects masters to slaves via 128-bit data buses running at a SYSCLK2 frequency (SYSCLK2 is generated from PLL1 controller). Peripherals that have a 128-bit data bus interface running at this speed can connect directly to the data SCR; other peripherals require a bridge.

The configuration switch fabric, also known as the configuration switch central resource (SCR) is mainly used by the C64x+ Megamodule to access peripheral registers (for more information, see Section 4.3). The configuration SCR connects C64x+ Megamodule to slaves via 32-bit configuration buses running at a SYSCLK2 frequency (SYSCLK2 is generated from PLL1 controller). As with the data SCR, some peripherals require the use of a bridge to interface to the configuration SCR. Note that the data SCR also connects to the configuration SCR.

Bridges perform a variety of functions:

- Conversion between configuration bus and data bus.
- Width conversion between peripheral bus width and SCR bus width.
- Frequency conversion between peripheral bus frequency and SCR bus frequency.

For example, the EMIFA and DDR2 memory controller require a bridge to convert their 64-bit data bus interface into a 128-bit interface so that they can connect to the data SCR.

Note that some peripherals can be accessed through the data SCR and also through the configuration SCR.

## 4.2 Data Switch Fabric Connections

Figure 4-1 shows the connection between slaves and masters through the data switched central resource (SCR). Masters are shown on the right and slaves on the left. The data SCR connects masters to slaves via 128-bit data buses running at a SYSCLK2 frequency. SYSCLK2 is supplied by the PLL1 controller and is fixed at a frequency equal to the CPU frequency divided by 3.

Some peripherals, like PCI and the C64x+ Megamodule, have both slave and master ports. Note that each EDMA3 transfer controller has an independent connection to the data SCR.

Note that masters can access the configuration SCR through the data SCR. The configuration SCR is described in Section 4.3.

Not all masters on the C6454 DSP may connect to all slaves. Allowed connections are summarized in Table 4-1.

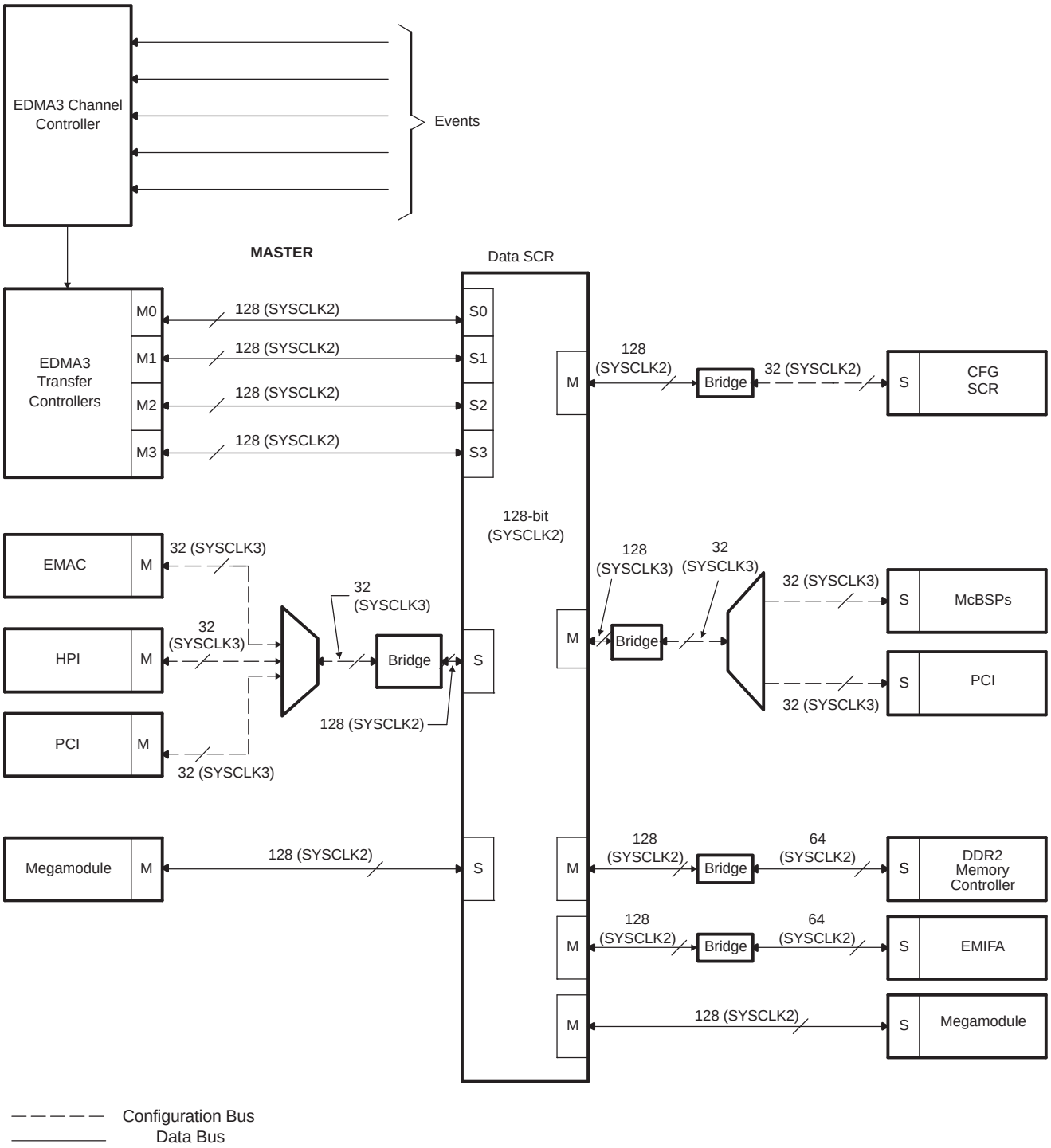



**Figure 4-1. Switched Central Resource Block Diagram**

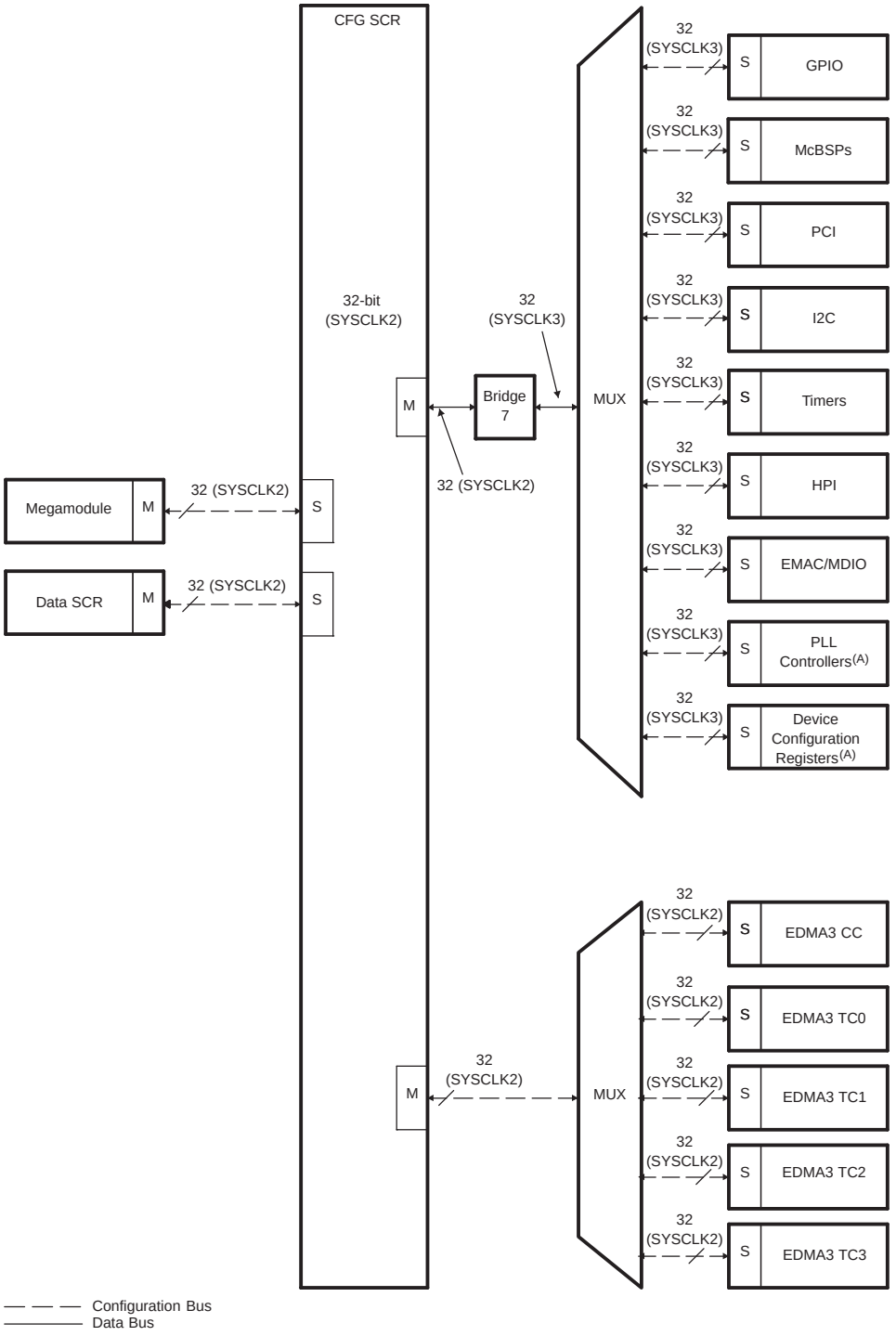
**Table 4-1. SCR Connection Matrix**

| | McBSPs | CONFIGURATION SCR | PCI | DDR2 MEMORY CONTROLLER | EMIFA | MEGAMODULE |
|---|---|---|---|---|---|---|
| **TC0** | N | N | N | Y | Y | Y |
| **TC1** | Y | Y | Y | Y | Y | Y |
| **TC2** | N | N | Y | Y | Y | Y |
| **TC3** | N | N | Y | Y | Y | Y |
| **EMAC** | N | N | N | Y | Y | Y |
| **HPI** | N | Y | N | Y | Y | Y |
| **PCI** | N | Y | N | Y | Y | Y |

## 4.3 Configuration Switch Fabric

Figure 4-2 shows the connection between the C64x+ Megamodule and the configuration switched central resource (SCR). The configuration SCR is mainly used by the C64x+ Megamodule to access peripheral registers. The data SCR also has a connection to the configuration SCR which allows masters to access most peripheral registers. The only registers not accessible by the data SCR through the configuration SCR are the device configuration registers and the PLL1 and PLL2 controller registers; these can only be accessed by the C64x+ Megamodule.

The configuration SCR uses 32-bit configuration buses running at SYSCLK2 frequency. SYSCLK2 is supplied by the PLL1 controller and is fixed at a frequency equal to the CPU frequency divided by 3.




A. Only accessible by the C64x+ Megamodule.
B. All clocks in this figure are generated by the PLL1 controller.

**Figure 4-2. C64x+ Megamodule - SCR Connection**

## 4.4 Bus Priorities

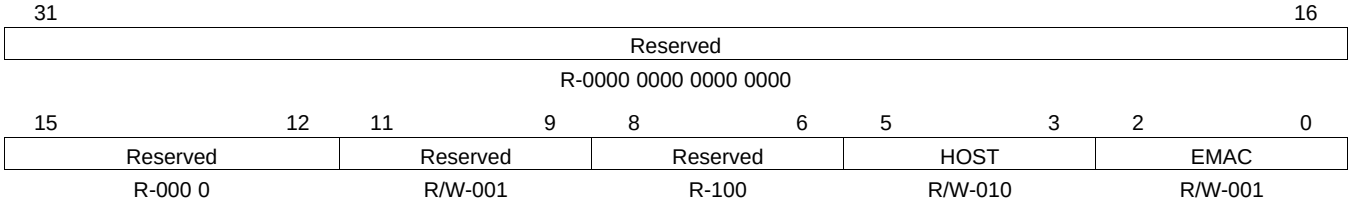
On the C6454 device, bus priority is programmable for each master. The register bit fields and default priority levels for C6454 bus masters are shown in Table 4-2. The priority levels should be tuned to obtain the best system performance for a particular application. Lower values indicate higher priorities. For some masters, the priority values are programmed at the system level by configuring the PRI_ALLOC register. Details on the PRI_ALLOC register are shown in Figure 4-3. The C64x+ megamodule and EDMA masters contain registers that control their own priority values.

The priority is enforced when several masters in the system are vying for the same endpoint. Note that the configuration SCR port on the data SCR is considered a single endpoint meaning priority will be enforced when multiple masters try to access the configuration SCR. Priority is also enforced on the configuration SCR side when a master (through the data SCR) tries to access the same endpoint as the C64x+ megamodule.

In the PRI_ALLOC register, the HOST field applies to the priority of the HPI and PCI peripherals. The EMAC field specifies the priority of the EMAC peripheral.

**Table 4-2. C6454 Default Bus Master Priorities**

| BUS MASTER | DEFAULT PRIORITY LEVEL | PRIORITY CONTROL |
|---|---|---|
| EDMA3TC0 | 0 | QUEPRI.PRIQ0 (EDMA3 register) |
| EDMA3TC1 | 0 | QUEPRI.PRIQ1 (EDMA3 register) |
| EDMA3TC2 | 0 | QUEPRI.PRIQ2 (EDMA3 register) |
| EDMA3TC3 | 0 | QUEPRI.PRIQ3 (EDMA3 register) |
| EMAC | 1 | PRI_ALLOC.EMAC |
| PCI | 2 | PRI_ALLOC.HOST |
| HPI | 2 | PRI_ALLOC.HOST |
| C64x+ Megamodule (MDMA port) | 7 | MDMAARBE.PRI (C64x+ Megamodule Register) |

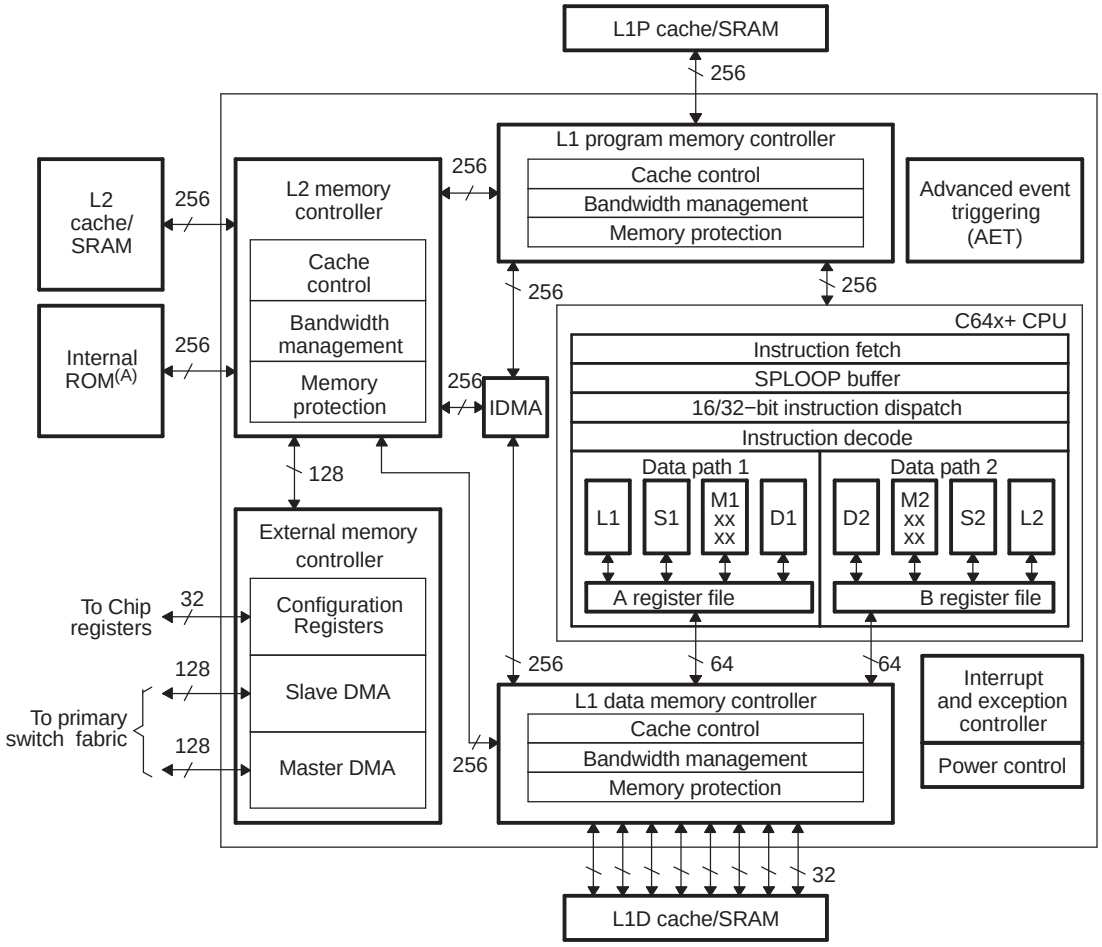



**LEGEND:** R/W = Read/Write; R = Read only; -*n* = value at reset

**Figure 4-3. Priority Allocation Register (PRI_ALLOC)**

# 5 C64x+ Megamodule

The C64x+ Megamodule consists of several components — the C64x+ CPU, the L1 program and data memory controllers, the L2 memory controller, the internal DMA (IDMA), the interrupt controller, power-down controller, and external memory controller. The C64x+ Megamodule also provides support for memory protection (for L1P, L1D, and L2 memories) and bandwidth management (for resources local to the C64x+ Megamodule). Figure 5-1 shows a block diagram of the C64x+ Megamodule.




A. When accessing the internal ROM of the DSP, the CPU frequency must always be less than 750 MHz.

**Figure 5-1. 64x+ Megamodule Block Diagram**

For more detailed information on the TMS320C64x+ Megamodule on the C6454 device, see the *TMS320C64x+ Megamodule Reference Guide* (literature number SPRU871).

## 5.1 Memory Architecture

The TMS320C6454 device contains a 1024KB level-2 memory (L2), a 32KB level-1 program memory (L1P), and a 32KB level-1 data memory (L1D).

The L1P memory configuration for the C6454 device is as follows:

- Region 0 size is 0K bytes (disabled).
- Region 1 size is 32K bytes with no wait states.

The L1D memory configuration for the C6454 device is as follows:

- Region 0 size is 0K bytes (disabled).
- Region 1 size is 32K bytes with no wait states.

L1D is a two-way set-associative cache while L1P is a direct-mapped cache.

The L1P and L1D cache can be reconfigured via software through the L1PMODE field of the L1P Configuration Register (L1PMODE) and the L1DMODE field of the L1D Configuration Register (L1DCFG) of the C64x+ Megamodule. After device reset, L1P and L1D cache are configured as all cache or all SRAM. The on-chip Bootloader changes the reset configuration for L1P and L1D. For more information, see the *TMS320C645x Bootloader User's Guide* (literature number SPRUEC6).

Figure 5-2 and Figure 5-3 show the available SRAM/cache configurations for L1P and L1D, respectively.

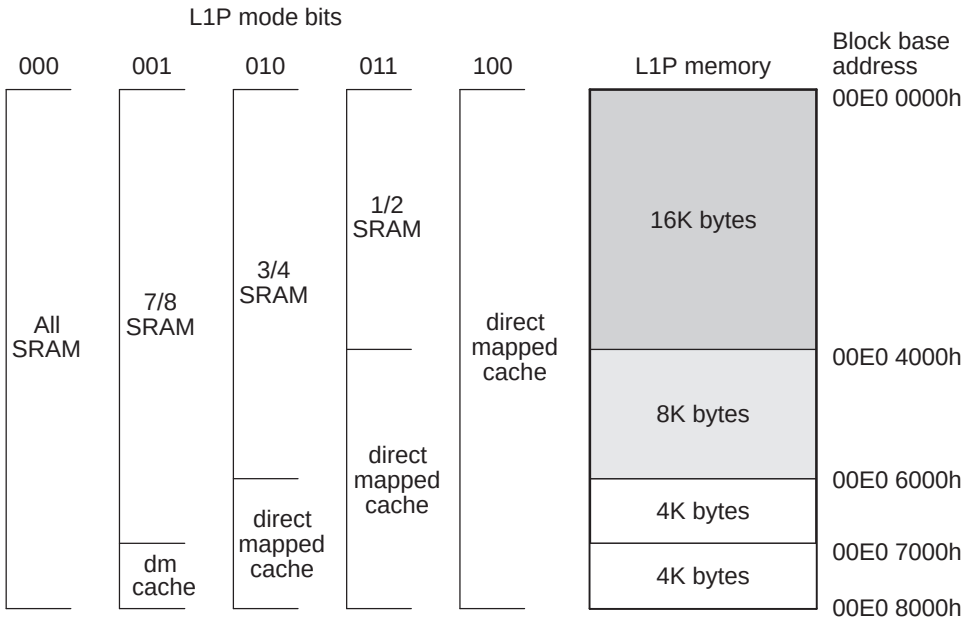



**Figure 5-2. TMS320C6454 L1P Memory Configurations**

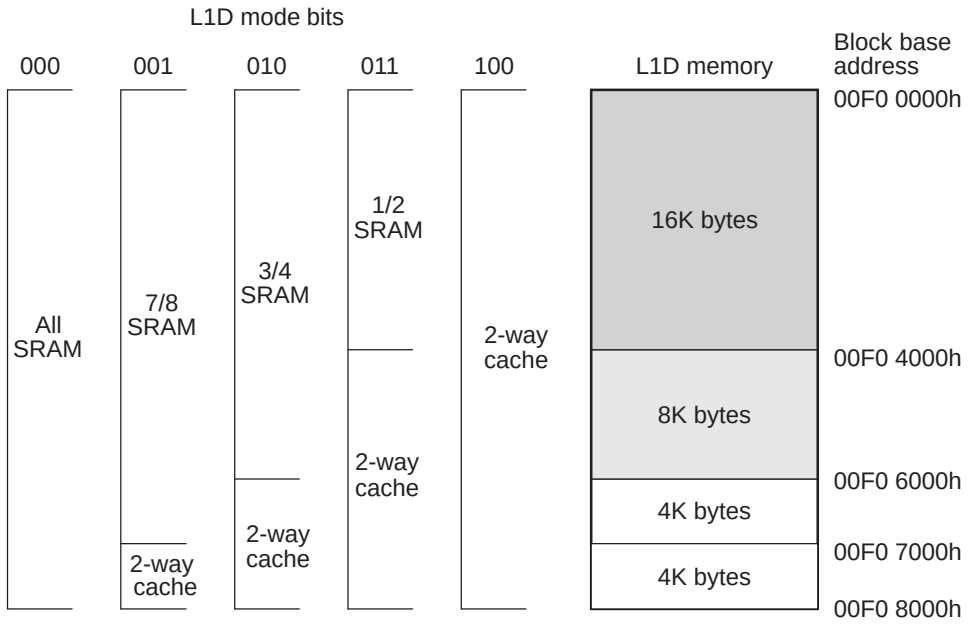



**Figure 5-3. TMS320C6454 L1D Memory Configurations**

The L2 memory configuration for the C6454 device is as follows:

- Port 0 configuration:
  - Memory size is 1024KB
  - Starting address is 0080 0000h
  - 2-cycle latency
  - 4 × 128-bit bank configuration
- Port 1 configuration:
  - Memory size is 32K bytes (this corresponds to the internal ROM)
  - Starting address is 0010 0000h
  - 1-cycle latency
  - 1 × 256-bit bank configuration

L2 memory can be configured as all SRAM or as part 4-way set-associative cache. The amount of L2 memory that is configured as cache is controlled through the L2MODE field of the L2 Configuration Register (L2CFG) of the C64x+ Megamodule. Figure 5-4 shows the available SRAM/cache configurations for L2. By default, L2 is configured as all SRAM after device reset.

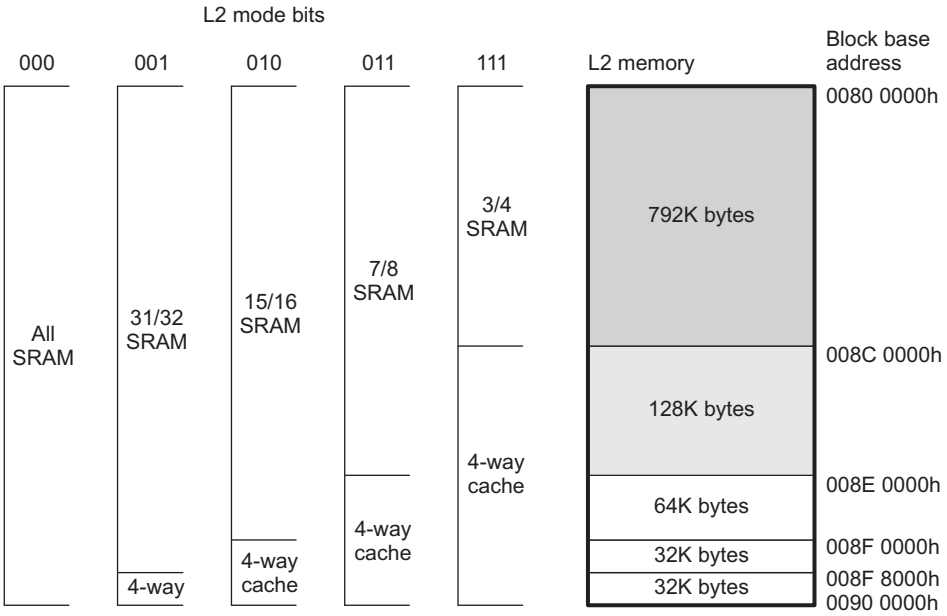



**Figure 5-4. TMS320C6454 L2 Memory Configurations**

For more information on the operation L1 and L2 caches, see the *TMS320C64x+ DSP Cache User's Guide* (literature number SPRU862).

All memory on the C6454 device has a unique location in the memory map (see Table 2-2).

When accessing the internal ROM of the DSP, the CPU frequency must always be less than 750 MHz. Therefore, when using a software boot mode, care must be taken such that the CPU frequency does not exceed 750 MHz at any point during the boot sequence. After the boot sequence has completed, the CPU frequency can be programmed to the frequency required by the application. For more detailed information ont he boot modes, see Section 2.4, *Boot Sequence*.

## 5.2 Memory Protection

Memory protection allows an operating system to define who or what is authorized to access L1D, L1P, and L2 memory. To accomplish this, the L1D, L1P, and L2 memories are divided into pages. There are 16 pages of L1P (2KB each), 16 pages of L1D (2KB each), and 16 pages of L2 (64KB each). The L1D, L1P, and L2 memory controllers in the C64x+ Megamodule are equipped with a set of registers that specify the permissions for each memory page.

Each page may be assigned with fully orthogonal user and supervisor read, write, and execute permissions. Additionally, a page may be marked as either (or both) locally or globally accessible. A local access is a direct CPU access to L1D, L1P, and L2, while a global access is initiated by a DMA (either IDMA or the EDMA3) or by other system masters. Note that EDMA or IDMA transfers programmed by the CPU count as global accesses.

The CPU and the system masters on the C6454 device are all assigned a privilege ID of 0. Therefore it is only possible to specify whether memory pages are locally or globally accessible. The AID0 and LOCAL bits of the memory protection page attribute registers specify the memory page protection scheme, see Table 5-1.

**Table 5-1. Available Memory Page Protection Schemes**

| AID0 Bit | LOCAL Bit | Description |
|---|---|---|
| 0 | 0 | No access to memory page is permitted. |
| 0 | 1 | Only direct access by CPU is permitted. |
| 1 | 0 | Only accesses by system masters and IDMA are permitted (includes EDMA and IDMA accesses initiated by the CPU). |
| 1 | 1 | All accesses permitted |

For more information on memory protection for L1D, L1P, and L2, see the *TMS320C64x+ Megamodule Reference Guide* (literature number SPRU871).

## 5.3 Bandwidth Management

When multiple requestors contend for a single C64x+ Megamodule resource, the conflict is solved by granting access to the highest priority requestor. The following four resources are managed by the Bandwidth Management control hardware:

- Level 1 Program (L1P) SRAM/Cache
- Level 1 Data (L1D) SRAM/Cache
- Level 2 (L2) SRAM/Cache
- Memory-mapped registers configuration bus

The priority level for operations initiated within the C64x+ Megamodule; e.g., CPU-initiated transfers, user-programmed cache coherency operations, and IDMA-initiated transfers, are declared through registers in the C64x+ Megamodule. The priority level for operations initiated outside the C64x+ Megamodule by system peripherals is declared through the Priority Allocation Register (PRI_ALLOC), see Section 4.4. System peripherals with no fields in PRI_ALLOC have their own registers to program their priorities.

More information on the bandwidth management features of the C64x+ Megamodule can be found in the *TMS320C64x+ Megamodule Reference Guide* (literature number SPRU871).

## 5.4 Power-Down Control

The C64x+ Megamodule supports the ability to power-down various parts of the C64x+ Megamodule. The power-down controller (PDC) of the C64x+ Megamodule can be used to power down L1P, the cache control hardware, the CPU, and the entire C64x+ Megamodule. These power-down features can be used to design systems for lower overall system power requirements.

**NOTE**

The C6454 device does not support power-down modes for the L2 memory at this time.

More information on the power-down features of the C64x+ Megamodule can be found in the *TMS320C64x+ Megamodule Reference Guide* (literature number SPRU871).

## 5.5 Megamodule Resets

Table 5-2 shows the reset types supported on the C6454 device and they affect the resetting of the Megamodule, either both globally or just locally.

**Table 5-2. Megamodule Reset (Global or Local)**

| RESET TYPE | GLOBAL MEGAMODULE RESET | LOCAL MEGAMODULE RESET |
|---|---|---|
| Power-On Reset | Y | Y |
| Warm Reset | Y | Y |
| System Reset | Y | Y |
| CPU Reset | N | Y |

For more detailed information on the global and local megamodule resets, see the *TMS320C64x+ Megamodule Reference Guide* (literature number SPRU871) and for more detailed information on device resets, see Section 7.6, *Reset Controller*.

## 5.6 Megamodule Revision

The version and revision of the C64x+ Megamodule can be read from the Megamodule Revision ID Register (MM_REVID) located at address 0181 2000h. The MM_REVID register is shown in Figure 5-5 and described in Table 5-3. The C64x+ Megamodule revision is dependant on the silicon revision being used. For more information, see the *TMS320C6455/54 Digital Signal Processor Silicon Errata* (literature number SPRZ234).

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| VERSION | | REVISION(A) | |
| R-1h | | R-n | |

**LEGEND:** R = Read only; -*n* = value after reset

A. The C64x+ Megamodule revision is dependant on the silicon revision being used. For more information, see the *TMS320C6455/54 Digital Signal Processor Silicon Errata* (literature number SPRZ234).

**Figure 5-5. Megamodule Revision ID Register (MM_REVID) [Hex Address: 0181 2000h]**

**Table 5-3. Megamodule Revision ID Register (MM_REVID) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:16 | VERSION | 1h | Version of the C64x+ Megamodule implemented on the device. This field is always read as 1h. |
| 15:0 | REVISION | | Revision of the C64x+ Megamodule version implemented on the device. The C64x+ Megamodule revision is dependant on the silicon revision being used. For more information, see the *TMS320C6455/54 Digital Signal Processor Silicon Errata* (literature number SPRZ234). |

## 5.7 C64x+ Megamodule Register Descriptions

**Table 5-4. Megamodule Interrupt Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0180 0000 | EVTFLAG0 | Event Flag Register 0 (Events [31:0]) |
| 0180 0004 | EVTFLAG1 | Event Flag Register 1 |
| 0180 0008 | EVTFLAG2 | Event Flag Register 2 |
| 0180 000C | EVTFLAG3 | Event Flag Register 3 |
| 0180 0010 - 0180 001C | - | Reserved |
| 0180 0020 | EVTSET0 | Event Set Register 0 (Events [31:0]) |
| 0180 0024 | EVTSET1 | Event Set Register 1 |
| 0180 0028 | EVTSET2 | Event Set Register 2 |
| 0180 002C | EVTSET3 | Event Set Register 3 |
| 0180 0030 - 0180 003C | - | Reserved |
| 0180 0040 | EVTCLR0 | Event Clear Register 0 (Events [31:0]) |
| 0180 0044 | EVTCLR1 | Event Clear Register 1 |
| 0180 0048 | EVTCLR2 | Event Clear Register 2 |
| 0180 004C | EVTCLR3 | Event Clear Register 3 |
| 0180 0050 - 0180 007C | - | Reserved |
| 0180 0080 | EVTMASK0 | Event Mask Register 0 (Events [31:0]) |
| 0180 0084 | EVTMASK1 | Event Mask Register 1 |
| 0180 0088 | EVTMASK2 | Event Mask Register 2 |
| 0180 008C | EVTMASK3 | Event Mask Register 3 |
| 0180 0090 - 0180 009C | - | Reserved |
| 0180 00A0 | MEVTFLAG0 | Masked Event Flag Status Register 0 (Events [31:0]) |
| 0180 00A4 | MEVTFLAG1 | Masked Event Flag Status Register 1 |
| 0180 00A8 | MEVTFLAG2 | Masked Event Flag Status Register 2 |
| 0180 00AC | MEVTFLAG3 | Masked Event Flag Status Register 3 |
| 0180 00B0 - 0180 00BC | - | Reserved |
| 0180 00C0 | EXPMASK0 | Exception Mask Register 0 (Events [31:0]) |
| 0180 00C4 | EXPMASK1 | Exception Mask Register 1 |
| 0180 00C8 | EXPMASK2 | Exception Mask Register 2 |
| 0180 00CC | EXPMASK3 | Exception Mask Register 3 |
| 0180 00D0 - 0180 00DC | - | Reserved |
| 0180 00E0 | MEXPFLAG0 | Masked Exception Flag Register 0 |
| 0180 00E4 | MEXPFLAG1 | Masked Exception Flag Register 1 |
| 0180 00E8 | MEXPFLAG2 | Masked Exception Flag Register 2 |
| 0180 00EC | MEXPFLAG3 | Masked Exception Flag Register 3 |
| 0180 00F0 - 0180 00FC | - | Reserved |
| 0180 0100 | - | Reserved |
| 0180 0104 | INTMUX1 | Interrupt Multiplexor Register 1 |
| 0180 0108 | INTMUX2 | Interrupt Multiplexor Register 2 |
| 0180 010C | INTMUX3 | Interrupt Multiplexor Register 3 |
| 0180 0110 - 0180 013C | - | Reserved |
| 0180 0140 | AEGMUX0 | Advanced Event Generator Mux Register 0 |
| 0180 0144 | AEGMUX1 | Advanced Event Generator Mux Register 1 |
| 0180 0148 - 0180 017C | - | Reserved |
| 0180 0180 | INTXSTAT | Interrupt Exception Status Register |
| 0180 0184 | INTXCLR | Interrupt Exception Clear Register |

**Table 5-4. Megamodule Interrupt Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0180 0188 | INTDMASK | Dropped Interrupt Mask Register |
| 0180 0188 - 0180 01BC | - | Reserved |
| 0180 01C0 | EVTASRT | Event Asserting Register |
| 0180 01C4 - 0180 FFFF | - | Reserved |

**Table 5-5. Megamodule Powerdown Control Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0181 0000 | PDCCMD | Power-down controller command register |
| 0181 0004 - 0181 1FFF | - | Reserved |

**Table 5-6. Megamodule Revision Register**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0181 2000 | MM_REVID | Megamodule Revision ID Register |
| 0181 2004 - 0181 2FFF | - | Reserved |

**Table 5-7. Megamodule IDMA Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0182 0000 | IDMA0STAT | IDMA Channel 0 Status Register |
| 0182 0004 | IDMA0MASK | IDMA Channel 0 Mask Register |
| 0182 0008 | IMDA0SRC | IDMA Channel 0 Source Address Register |
| 0182 000C | IDMA0DST | IDMA Channel 0 Destination Address Register |
| 0182 0010 | IDMA0CNT | IDMA Channel 0 Count Register |
| 0182 0014 - 0182 00FC | - | Reserved |
| 0182 0100 | IDMA1STAT | IDMA Channel 1 Status Register |
| 0182 0104 | - | Reserved |
| 0182 0108 | IMDA1SRC | IDMA Channel 1 Source Address Register |
| 0182 010C | IDMA1DST | IDMA Channel 1 Destination Address Register |
| 0182 0110 | IDMA1CNT | IDMA Channel 1 Count Register |
| 0182 0114 - 0182 017C | - | Reserved |
| 0182 0180 | - | Reserved |
| 0182 0184 - 0182 01FF | - | Reserved |

**Table 5-8. Megamodule Cache Configuration Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0184 0000 | L2CFG | L2 Cache Configuration Register |
| 0184 0004 - 0184 001F | - | Reserved |
| 0184 0020 | L1PCFG | L1P Configuration Register |
| 0184 0024 | L1PCC | L1P Cache Control Register |
| 0184 0028 - 0184 003F | - | Reserved |
| 0184 0040 | L1DCFG | L1D Configuration Register |
| 0184 0044 | L1DCC | L1D Cache Control Register |
| 0184 0048 - 0184 0FFF | - | Reserved |
| 0184 1000 - 0184 104F | - | See Table 5-10, *CPU Megamodule Bandwidth Management Registers* |
| 0184 1050 - 0184 3FFF | - | Reserved |
| 0184 4000 | L2WBAR | L2 Writeback Base Address Register - for Block Writebacks |
| 0184 4004 | L2WWC | L2 Writeback Word Count Register |
| 0184 4008 - 0184 400C | - | Reserved |
| 0184 4010 | L2WIBAR | L2 Writeback and Invalidate Base Address Register - for Block Writebacks |
| 0184 4014 | L2WIWC | L2 Writeback and Invalidate word count register |
| 0184 4018 | L2IBAR | L2 Invalidate Base Address Register |
| 0184 401C | L2IWC | L2 Invalidate Word Count Register |
| 0184 4020 | L1PIBAR | L1P Invalidate Base Address Register |
| 0184 4024 | L1PIWC | L1P Invalidate Word Count Register |
| 0184 4030 | L1DWIBAR | L1D Writeback and Invalidate Base Address Register |
| 0184 4034 | L1DWIWC | L1D Writeback and Invalidate Word Count Register |
| 0184 4038 | - | Reserved |
| 0184 4040 | L1DWBAR | L1D Writeback Base Address Register - for Block Writebacks |
| 0184 4044 | L1DWWC | L1D Writeback Word Count Register |
| 0184 4048 | L1DIBAR | L1D Invalidate Base Address Register |
| 0184 404C | L1DIWC | L1D Invalidate Word Count Register |
| 0184 4050 - 0184 4FFF | - | Reserved |
| 0184 5000 | L2WB | L2 Global Writeback Register |
| 0184 5004 | L2WBINV | L2 Global Writeback and Invalidate Register |
| 0184 5008 | L2INV | L2 Global Invalidate Register |
| 0184 500C - 0184 5024 | - | Reserved |
| 0184 5028 | L1PINV | L1P Global Invalidate Register |
| 0184 502C - 0184 503C | - | Reserved |
| 0184 5040 | L1DWB | L1D Global Writeback Register |
| 0184 5044 | L1DWBINV | L1D Global Writeback and Invalidate Register |
| 0184 5048 | L1DINV | L1D Global Invalidate Register |
| 0184 504C - 0184 7FFF | - | Reserved |
| 0184 8000 - 0184 81FC | MAR0 to MAR127 | Reserved |
| 0184 8200 - 0184 823C | MAR128 to MAR143 | Reserved |
| 0184 8240 - 0184 827C | MAR144 to MAR159 | Reserved |
| 0184 8280 | MAR160 | Controls EMIFA CE2 Range A000 0000 - A0FF FFFF |
| 0184 8284 | MAR161 | Controls EMIFA CE2 Range A100 0000 - A1FF FFFF |
| 0184 8288 | MAR162 | Controls EMIFA CE2 Range A200 0000 - A2FF FFFF |
| 0184 828C | MAR163 | Controls EMIFA CE2 Range A300 0000 - A3FF FFFF |
| 0184 8290 | MAR164 | Controls EMIFA CE2 Range A400 0000 - A4FF FFFF |

**Table 5-8. Megamodule Cache Configuration Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0184 8294 | MAR165 | Controls EMIFA CE2 Range A500 0000 - A5FF FFFF |
| 0184 8298 | MAR166 | Controls EMIFA CE2 Range A600 0000 - A6FF FFFF |
| 0184 829C | MAR167 | Controls EMIFA CE2 Range A700 0000 - A7FF FFFF |
| 0184 82A0 | MAR168 | Controls EMIFA CE2 Range A800 0000 - A8FF FFFF |
| 0184 82A4 | MAR169 | Controls EMIFA CE2 Range A900 0000 - A9FF FFFF |
| 0184 82A8 | MAR170 | Controls EMIFA CE2 Range AA00 0000 - AAFF FFFF |
| 0184 82AC | MAR171 | Controls EMIFA CE2 Range AB00 0000 - ABFF FFFF |
| 0184 82B0 | MAR172 | Controls EMIFA CE2 Range AC00 0000 - ACFF FFFF |
| 0184 82B4 | MAR173 | Controls EMIFA CE2 Range AD00 0000 - ADFF FFFF |
| 0184 82B8 | MAR174 | Controls EMIFA CE2 Range AE00 0000 - AEFF FFFF |
| 0184 82BC | MAR175 | Controls EMIFA CE2 Range AF00 0000 - AFFF FFFF |
| 0184 82C0 | MAR176 | Controls EMIFA CE3 Range B000 0000 - B0FF FFFF |
| 0184 82C4 | MAR177 | Controls EMIFA CE3 Range B100 0000 - B1FF FFFF |
| 0184 82C8 | MAR178 | Controls EMIFA CE3 Range B200 0000 - B2FF FFFF |
| 0184 82CC | MAR179 | Controls EMIFA CE3 Range B300 0000 - B3FF FFFF |
| 0184 82D0 | MAR180 | Controls EMIFA CE3 Range B400 0000 - B4FF FFFF |
| 0184 82D4 | MAR181 | Controls EMIFA CE3 Range B500 0000 - B5FF FFFF |
| 0184 82D8 | MAR182 | Controls EMIFA CE3 Range B600 0000 - B6FF FFFF |
| 0184 82DC | MAR183 | Controls EMIFA CE3 Range B700 0000 - B7FF FFFF |
| 0184 82E0 | MAR184 | Controls EMIFA CE3 Range B800 0000 - B8FF FFFF |
| 0184 82E4 | MAR185 | Controls EMIFA CE3 Range B900 0000 - B9FF FFFF |
| 0184 82E8 | MAR186 | Controls EMIFA CE3 Range BA00 0000 - BAFF FFFF |
| 0184 82EC | MAR187 | Controls EMIFA CE3 Range BB00 0000 - BBFF FFFF |
| 0184 82F0 | MAR188 | Controls EMIFA CE3 Range BC00 0000 - BCFF FFFF |
| 0184 82F4 | MAR189 | Controls EMIFA CE3 Range BD00 0000 - BDFF FFFF |
| 0184 82F8 | MAR190 | Controls EMIFA CE3 Range BE00 0000 - BEFF FFFF |
| 0184 82FC | MAR191 | Controls EMIFA CE3 Range BF00 0000 - BFFF FFFF |
| 0184 8300 | MAR192 | Controls EMIFA CE4 Range C000 0000 - C0FF FFFF |
| 0184 8304 | MAR193 | Controls EMIFA CE4 Range C100 0000 - C1FF FFFF |
| 0184 8308 | MAR194 | Controls EMIFA CE4 Range C200 0000 - C2FF FFFF |
| 0184 830C | MAR195 | Controls EMIFA CE4 Range C300 0000 - C3FF FFFF |
| 0184 8310 | MAR196 | Controls EMIFA CE4 Range C400 0000 - C4FF FFFF |
| 0184 8314 | MAR197 | Controls EMIFA CE4 Range C500 0000 - C5FF FFFF |
| 0184 8318 | MAR198 | Controls EMIFA CE4 Range C600 0000 - C6FF FFFF |
| 0184 831C | MAR199 | Controls EMIFA CE4 Range C700 0000 - C7FF FFFF |
| 0184 8320 | MAR200 | Controls EMIFA CE4 Range C800 0000 - C8FF FFFF |
| 0184 8324 | MAR201 | Controls EMIFA CE4 Range C900 0000 - C9FF FFFF |
| 0184 8328 | MAR202 | Controls EMIFA CE4 Range CA00 0000 - CAFF FFFF |
| 0184 832C | MAR203 | Controls EMIFA CE4 Range CB00 0000 - CBFF FFFF |
| 0184 8330 | MAR204 | Controls EMIFA CE4 Range CC00 0000 - CCFF FFFF |
| 0184 8334 | MAR205 | Controls EMIFA CE4 Range CD00 0000 - CDFF FFFF |
| 0184 8338 | MAR206 | Controls EMIFA CE4 Range CE00 0000 - CEFF FFFF |
| 0184 833C | MAR207 | Controls EMIFA CE4 Range CF00 0000 - CFFF FFFF |
| 0184 8340 | MAR208 | Controls EMIFA CE5 Range D000 0000 - D0FF FFFF |
| 0184 8344 | MAR209 | Controls EMIFA CE5 Range D100 0000 - D1FF FFFF |
| 0184 8348 | MAR210 | Controls EMIFA CE5 Range D200 0000 - D2FF FFFF |
| 0184 834C | MAR211 | Controls EMIFA CE5 Range D300 0000 - D3FF FFFF |

**Table 5-8. Megamodule Cache Configuration Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0184 8350 | MAR212 | Controls EMIFA CE5 Range D400 0000 - D4FF FFFF |
| 0184 8354 | MAR213 | Controls EMIFA CE5 Range D500 0000 - D5FF FFFF |
| 0184 8358 | MAR214 | Controls EMIFA CE5 Range D600 0000 - D6FF FFFF |
| 0184 835C | MAR215 | Controls EMIFA CE5 Range D700 0000 - D7FF FFFF |
| 0184 8360 | MAR216 | Controls EMIFA CE5 Range D800 0000 - D8FF FFFF |
| 0184 8364 | MAR217 | Controls EMIFA CE5 Range D900 0000 - D9FF FFFF |
| 0184 8368 | MAR218 | Controls EMIFA CE5 Range DA00 0000 - DAFF FFFF |
| 0184 836C | MAR219 | Controls EMIFA CE5 Range DB00 0000 - DBFF FFFF |
| 0184 8370 | MAR220 | Controls EMIFA CE5 Range DC00 0000 - DCFF FFFF |
| 0184 8374 | MAR221 | Controls EMIFA CE5 Range DD00 0000 - DDFF FFFF |
| 0184 8378 | MAR222 | Controls EMIFA CE5 Range DE00 0000 - DEFF FFFF |
| 0184 837C | MAR223 | Controls EMIFA CE5 Range DF00 0000 - DFFF FFFF |
| 0184 8380 | MAR224 | Controls DDR2 CE0 Range E000 0000 - E0FF FFFF |
| 0184 8384 | MAR225 | Controls DDR2 CE0 Range E100 0000 - E1FF FFFF |
| 0184 8388 | MAR226 | Controls DDR2 CE0 Range E200 0000 - E2FF FFFF |
| 0184 838C | MAR227 | Controls DDR2 CE0 Range E300 0000 - E3FF FFFF |
| 0184 8390 | MAR228 | Controls DDR2 CE0 Range E400 0000 - E4FF FFFF |
| 0184 8394 | MAR229 | Controls DDR2 CE0 Range E500 0000 - E5FF FFFF |
| 0184 8398 | MAR230 | Controls DDR2 CE0 Range E600 0000 - E6FF FFFF |
| 0184 839C | MAR231 | Controls DDR2 CE0 Range E700 0000 - E7FF FFFF |
| 0184 83A0 | MAR232 | Controls DDR2 CE0 Range E800 0000 - E8FF FFFF |
| 0184 83A4 | MAR233 | Controls DDR2 CE0 Range E900 0000 - E9FF FFFF |
| 0184 83A8 | MAR234 | Controls DDR2 CE0 Range EA00 0000 - EAFF FFFF |
| 0184 83AC | MAR235 | Controls DDR2 CE0 Range EB00 0000 - EBFF FFFF |
| 0184 83B0 | MAR236 | Controls DDR2 CE0 Range EC00 0000 - ECFF FFFF |
| 0184 83B4 | MAR237 | Controls DDR2 CE0 Range ED00 0000 - EDFF FFFF |
| 0184 83B8 | MAR238 | Controls DDR2 CE0 Range EE00 0000 - EEFF FFFF |
| 0184 83BC | MAR239 | Controls DDR2 CE0 Range EF00 0000 - EFFF FFFF |
| 0184 83C0 -0184 83FC | MAR240 to MAR255 | Reserved |

**Table 5-9. Megamodule L1/L2 Memory Protection Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0184 A000 | L2MPFAR | L2 memory protection fault address register |
| 0184 A004 | L2MPFSR | L2 memory protection fault status register |
| 0184 A008 | L2MPFCR | L2 memory protection fault command register |
| 0184 A00C - 0184 A0FF | - | Reserved |
| 0184 A100 | L2MPLK0 | L2 memory protection lock key bits [31:0] |
| 0184 A104 | L2MPLK1 | L2 memory protection lock key bits [63:32] |
| 0184 A108 | L2MPLK2 | L2 memory protection lock key bits [95:64] |
| 0184 A10C | L2MPLK3 | L2 memory protection lock key bits [127:96] |
| 0184 A110 | L2MPLKCMD | L2 memory protection lock key command register |
| 0184 A114 | L2MPLKSTAT | L2 memory protection lock key status register |
| 0184 A118 - 0184 A1FF | - | Reserved |
| 0184 A200 | L2MPPA0 | L2 memory protection page attribute register 0 |
| 0184 A204 | L2MPPA1 | L2 memory protection page attribute register 1 |
| 0184 A208 | L2MPPA2 | L2 memory protection page attribute register 2 |

**Table 5-9. Megamodule L1/L2 Memory Protection Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0184 A20C | L2MPPA3 | L2 memory protection page attribute register 3 |
| 0184 A210 | L2MPPA4 | L2 memory protection page attribute register 4 |
| 0184 A214 | L2MPPA5 | L2 memory protection page attribute register 5 |
| 0184 A218 | L2MPPA6 | L2 memory protection page attribute register 6 |
| 0184 A21C | L2MPPA7 | L2 memory protection page attribute register 7 |
| 0184 A220 | L2MPPA8 | L2 memory protection page attribute register 8 |
| 0184 A224 | L2MPPA9 | L2 memory protection page attribute register 9 |
| 0184 A228 | L2MPPA10 | L2 memory protection page attribute register 10 |
| 0184 A22C | L2MPPA11 | L2 memory protection page attribute register 11 |
| 0184 A230 | L2MPPA12 | L2 memory protection page attribute register 12 |
| 0184 A234 | L2MPPA13 | L2 memory protection page attribute register 13 |
| 0184 A238 | L2MPPA14 | L2 memory protection page attribute register 14 |
| 0184 A23C | L2MPPA15 | L2 memory protection page attribute register 15 |
| 0184 A240 | L2MPPA16 | L2 memory protection page attribute register 16 |
| 0184 A244 | L2MPPA17 | L2 memory protection page attribute register 17 |
| 0184 A248 | L2MPPA18 | L2 memory protection page attribute register 18 |
| 0184 A24C | L2MPPA19 | L2 memory protection page attribute register 19 |
| 0184 A250 | L2MPPA20 | L2 memory protection page attribute register 20 |
| 0184 A254 | L2MPPA21 | L2 memory protection page attribute register 21 |
| 0184 A258 | L2MPPA22 | L2 memory protection page attribute register 22 |
| 0184 A25C | L2MPPA23 | L2 memory protection page attribute register 23 |
| 0184 A260 | L2MPPA24 | L2 memory protection page attribute register 24 |
| 0184 A264 | L2MPPA25 | L2 memory protection page attribute register 25 |
| 0184 A268 | L2MPPA26 | L2 memory protection page attribute register 26 |
| 0184 A26C | L2MPPA27 | L2 memory protection page attribute register 27 |
| 0184 A270 | L2MPPA28 | L2 memory protection page attribute register 28 |
| 0184 A274 | L2MPPA29 | L2 memory protection page attribute register 29 |
| 0184 A278 | L2MPPA30 | L2 memory protection page attribute register 30 |
| 0184 A27C | L2MPPA31 | L2 memory protection page attribute register 31 |
| 0184 A280 - 0184 A2FC(1) | - | Reserved |
| 0184 0300 - 0184 A3FF | - | Reserved |
| 0184 A400 | L1PMPFAR | L1 program (L1P) memory protection fault address register |
| 0184 A404 | L1PMPFSR | L1P memory protection fault status register |
| 0184 A408 | L1PMPFCR | L1P memory protection fault command register |
| 0184 A40C - 0184 A4FF | - | Reserved |
| 0184 A500 | L1PMPLK0 | L1P memory protection lock key bits [31:0] |
| 0184 A504 | L1PMPLK1 | L1P memory protection lock key bits [63:32] |
| 0184 A508 | L1PMPLK2 | L1P memory protection lock key bits [95:64] |
| 0184 A50C | L1PMPLK3 | L1P memory protection lock key bits [127:96] |
| 0184 A510 | L1PMPLKCMD | L1P memory protection lock key command register |
| 0184 A514 | L1PMPLKSTAT | L1P memory protection lock key status register |
| 0184 A518 - 0184 A5FF | - | Reserved |
| 0184 A600 - 0184 A63C(2) | - | Reserved |
| 0184 A640 | L1PMPPA16 | L1P memory protection page attribute register 16 |

(1) These addresses correspond to the L2 memory protection page attribute registers 32-63 (L2MPPA32-L2MPPA63) of the C64x+ megamaodule. These registers are not supported for the C6454 device.
(2) These addresses correspond to the L1P memory protection page attribute registers 0-15 (L1PMPPA0-L1PMPPA15) of the C64x+ megamaodule. These registers are not supported for the C6454 device.

**Table 5-9. Megamodule L1/L2 Memory Protection Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0184 A644 | L1PMPPA17 | L1P memory protection page attribute register 17 |
| 0184 A648 | L1PMPPA18 | L1P memory protection page attribute register 18 |
| 0184 A64C | L1PMPPA19 | L1P memory protection page attribute register 19 |
| 0184 A650 | L1PMPPA20 | L1P memory protection page attribute register 20 |
| 0184 A654 | L1PMPPA21 | L1P memory protection page attribute register 21 |
| 0184 A658 | L1PMPPA22 | L1P memory protection page attribute register 22 |
| 0184 A65C | L1PMPPA23 | L1P memory protection page attribute register 23 |
| 0184 A660 | L1PMPPA24 | L1P memory protection page attribute register 24 |
| 0184 A664 | L1PMPPA25 | L1P memory protection page attribute register 25 |
| 0184 A668 | L1PMPPA26 | L1P memory protection page attribute register 26 |
| 0184 A66C | L1PMPPA27 | L1P memory protection page attribute register 27 |
| 0184 A670 | L1PMPPA28 | L1P memory protection page attribute register 28 |
| 0184 A674 | L1PMPPA29 | L1P memory protection page attribute register 29 |
| 0184 A678 | L1PMPPA30 | L1P memory protection page attribute register 30 |
| 0184 A67C | L1PMPPA31 | L1P memory protection page attribute register 31 |
| 0184 A680 - 0184 ABFF | - | Reserved |
| 0184 AC00 | L1DMPFAR | L1 data (L1D) memory protection fault address register |
| 0184 AC04 | L1DMPFSR | L1D memory protection fault status register |
| 0184 AC08 | L1DMPFCR | L1D memory protection fault command register |
| 0184 AC0C - 0184 ACFF | - | Reserved |
| 0184 AD00 | L1DMPLK0 | L1D memory protection lock key bits [31:0] |
| 0184 AD04 | L1DMPLK1 | L1D memory protection lock key bits [63:32] |
| 0184 AD08 | L1DMPLK2 | L1D memory protection lock key bits [95:64] |
| 0184 AD0C | L1DMPLK3 | L1D memory protection lock key bits [127:96] |
| 0184 AD10 | L1DMPLKCMD | L1D memory protection lock key command register |
| 0184 AD14 | L1DMPLKSTAT | L1D memory protection lock key status register |
| 0184 AD18 - 0184 ADFF | - | Reserved |
| 0184 AE00 - 0184 AE3C(3) | - | Reserved |
| 0184 AE40 | L1DMPPA16 | L1D memory protection page attribute register 16 |
| 0184 AE44 | L1DMPPA17 | L1D memory protection page attribute register 17 |
| 0184 AE48 | L1DMPPA18 | L1D memory protection page attribute register 18 |
| 0184 AE4C | L1DMPPA19 | L1D memory protection page attribute register 19 |
| 0184 AE50 | L1DMPPA20 | L1D memory protection page attribute register 20 |
| 0184 AE54 | L1DMPPA21 | L1D memory protection page attribute register 21 |
| 0184 AE58 | L1DMPPA22 | L1D memory protection page attribute register 22 |
| 0184 AE5C | L1DMPPA23 | L1D memory protection page attribute register 23 |
| 0184 AE60 | L1DMPPA24 | L1D memory protection page attribute register 24 |
| 0184 AE64 | L1DMPPA25 | L1D memory protection page attribute register 25 |
| 0184 AE68 | L1DMPPA26 | L1D memory protection page attribute register 26 |
| 0184 AE6C | L1DMPPA27 | L1D memory protection page attribute register 27 |
| 0184 AE70 | L1DMPPA28 | L1D memory protection page attribute register 28 |
| 0184 AE74 | L1DMPPA29 | L1D memory protection page attribute register 29 |
| 0184 AE78 | L1DMPPA30 | L1D memory protection page attribute register 30 |
| 0184 AE7C | L1DMPPA31 | L1D memory protection page attribute register 31 |
| 0184 AE80 - 0185 FFFF | - | Reserved |

(3) These addresses correspond to the L1D memory protection page attribute registers 0-15 (L1DMPPA0-L1DMPPA15) of the C64x+ megamaodule. These registers are not supported for the C6454 device.

**Table 5-10. CPU Megamodule Bandwidth Management Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0182 0200 | EMCCPUARBE | EMC CPU Arbitration Control Register |
| 0182 0204 | EMCIDMAARBE | EMC IDMA Arbitration Control Register |
| 0182 0208 | EMCSDMAARBE | EMC Slave DMA Arbitration Control Register |
| 0182 020C | EMCMDMAARBE | EMC Master DMA Arbitration Control Resgiter |
| 0182 0210 - 0182 02FF | - | Reserved |
| 0184 1000 | L2DCPUARBU | L2D CPU Arbitration Control Register |
| 0184 1004 | L2DIDMAARBU | L2D IDMA Arbitration Control Register |
| 0184 1008 | L2DSDMAARBU | L2D Slave DMA Arbitration Control Register |
| 0184 100C | L2DUCARBU | L2D User Coherence Arbitration Control Resgiter |
| 0184 1010 - 0184 103F | - | Reserved |
| 0184 1040 | L1DCPUARBD | L1D CPU Arbitration Control Register |
| 0184 1044 | L1DIDMAARBD | L1D IDMA Arbitration Control Register |
| 0184 1048 | L1DSDMAARBD | L1D Slave DMA Arbitration Control Register |
| 0184 104C | L1DUCARBD | L1D User Coherence Arbitration Control Resgiter |

**Table 5-11. Device Configuration Registers (Chip-Level Registers)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 02A8 0000 | DEVSTAT | Device Status Register | Read-only. Provides status of the user's device configuration on reset. |
| 02A8 0004 | PRI_ALLOC | Priority Allocation Register | Sets priority for Master peripherals |
| 02A8 0008 | JTAGID | JTAG and BSDL Identification Register | Read-only. Provides 32-bit JTAG ID of the device. |
| 02A8 000C - 02AB FFFF | - | Reserved | |
| 02AC 0000 | - | Reserved | |
| 02AC 0004 | PERLOCK | Peripheral Lock Register | |
| 02AC 0008 | PERCFG0 | Peripheral Configuration Register 0 | |
| 02AC 000C | - | Reserved | |
| 02AC 0010 | - | Reserved | |
| 02AC 0014 | PERSTAT0 | Peripheral Status Register 0 | |
| 02AC 0018 | PERSTAT1 | Peripheral Status Register 1 | |
| 02AC 001C - 02AC 001F | - | Reserved | |
| 02AC 0020 | EMACCFG | EMAC Configuration Register | |
| 02AC 0024 - 02AC 002B | - | Reserved | |
| 02AC 002C | PERCFG1 | Peripheral Configuration Register 1 | |
| 02AC 0030 - 02AC 0053 | - | Reserved | |
| 02AC 0054 | EMUBUFPD | Emulator Buffer Powerdown Register | |
| 02AC 0058 | - | Reserved | |

# 6 Device Operating Conditions

## 6.1 Absolute Maximum Ratings Over Operating Case Temperature Range (Unless Otherwise Noted)(1)

| | | |
|---|---|---|
| Supply voltage range: | $CV_{DD}$ (2) | -0.5 V to 1.5 V |
| | $DV_{DD33}$ (2) | -0.5 V to 4.2 V |
| | $DV_{DD15}$, $DV_{DD18}$, $AV_{DLL1}$, $AV_{DLL2}$ (2) | -0.5 V to 2.5 V |
| | PLLV1, PLLV2(2) | -0.5 V to 2.5 V |
| Input voltage ($V_I$) range: | 3.3-V pins (except PCI-capable pins) | -0.5 V to $DV_{DD33}$ + 0.5 V |
| | PCI-capable pins | -0.5 V to $DV_{DD33}$ + 0.5 V |
| | RGMII pins | -0.5 V to 2.5 V |
| | DDR2 memory controller pins | -0.5 V to 2.5 V |
| Output voltage ($V_O$) range: | 3.3-V pins (except PCI-capable pins) | -0.5 V to $DV_{DD33}$ + 0.5 V |
| | PCI-capable pins | -0.5 V to $DV_{DD33}$ + 0.5 V |
| | RGMII pins | -0.5 V to 2.5 V |
| | DDR2 memory controller pins | -0.5 V to 2.5 V |
| Operating case temperature range, $T_C$: | (default) | 0°C to 90°C |
| | (A version) [A-1000 device] | -40°C to 105°C |
| Storage temperature range, $T_{stg}$ | | -65°C to 150°C |

(1) Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "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 $V_{SS}$.

## 6.2 Recommended Operating Conditions

| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $CV_{DD}$ | Supply voltage, Core | A-1000/-1000 | 1.2125 | 1.25 | 1.2875 | V |
| | | -850<br>-720 | 1.1640 | 1.20 | 1.2360 | V |
| $DV_{DD33}$ | Supply voltage, I/O | | 3.14 | 3.3 | 3.46 | V |
| $DV_{DD18}$ | Supply voltage, I/O | | 1.71 | 1.8 | 1.89 | V |
| $AV_{DLL1}$ | Supply voltage, I/O | | 1.71 | 1.8 | 1.89 | V |
| $AV_{DLL2}$ | Supply voltage, I/O | | 1.71 | 1.8 | 1.89 | V |
| $V_{REFSSTL}$ | Reference voltage | | $0.49DV_{DD18}$ | $0.50DV_{DD18}$ | $0.51DV_{DD18}$ | V |
| $DV_{DD15}$ | Supply voltage, I/O [required only for EMAC RGMII] | 1.8-V operation | 1.71 | 1.8 | 1.89 | V |
| | | 1.5-V operation | 1.43 | 1.5 | 1.57 | V |
| $V_{REFHSTL}$ | Reference voltage | 1.8-V operation | 0.855 | 0.9 | 0.945 | V |
| | | 1.5-V operation | 0.713 | 0.75 | 0.787 | V |
| PLLV1, PLLV2 | Supply voltage, PLL | | 1.71 | 1.8 | 1.89 | V |
| $V_{SS}$ | Supply ground | | 0 | 0 | 0 | V |

## Recommended Operating Conditions *(continued)*

| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_{IH}$ | High-level input voltage | 3.3 V pins (except PCI-capable and I2C pins) | 2 | | | V |
| | | PCI-capable pins(1) | $0.5DV_{DD33}$ | | $DV_{DD33}$ + 0.5 | V |
| | | I2C pins | $0.7DV_{DD33}$ | | | V |
| | | RGMII pins | $V_{REFHSTL}$ + 0.10 | | $DV_{DD15}$ + 0.30 | V |
| | | DDR2 memory controller pins (DC) | $V_{REFSSTL}$ + 0.125 | | $DV_{DD18}$ + 0.3 | V |
| $V_{IL}$ | Low-level input voltage | 3.3 V pins (except PCI-capable and I2C pins) | | | 0.8 | V |
| | | PCI-capable pins(1) | -0.5 | | $0.3DV_{DD33}$ | V |
| | | I2C pins | 0 | | $0.3DV_{DD33}$ | V |
| | | RGMII pins | -0.3 | | $V_{REFHSTL}$ - 0.1 | V |
| | | DDR2 memory controller pins (DC) | -0.3 | | $V_{REFSSTL}$ - 0.125 | V |
| $V_{OS}$ | Maximum voltage during overshoot/undershoot (PCI-capable pins)(2) (3) | I/O $V_{DD}$ = 1.5 V (EMAC RGMII) | -0.450 | | 1.950 | V |
| | | I/O $V_{DD}$ = 1.8 V (EMAC RGMII, DDR2) | -0.540 | | 2.340 | |
| | | I/O $V_{DD}$ = 3.3 V (except PCI-capable pins) | -1.000 | | 4.300 | |
| $T_C$ | Operating case temperature | commercial temperature | 0 | | 90 | °C |
| | | extended temperature | -40 | | 105 | |

(1) These rated numbers are from the *PCI Local Bus Specification* (version 2.3). The DC specifications and AC specifications are defined in Table 4-3 and Table 4-4, respectively, of the *PCI Local Bus Specification*.
(2) PCI-capable pins can withstand a maximum overshoot/undershoot for up to 11 ns as required by the *PCI Local Bus Specification* (version 2.3).
(3) Duration of overshoot/undershoot must not exceed 30% of the cycle period.

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

| PARAMETER | | | TEST CONDITIONS(1) | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $V_{OH}$ | High-level output voltage | 3.3-V pins (except PCI-capable and I2C pins) | $DV_{DD33}$ = MIN, $I_{OH}$ = MAX | $0.8DV_{DD33}$ | | | V |
| | | PCI-capable pins(2) | $I_{OH}$ = -0.5 mA, $DV_{DD33}$ = 3.3 V | $0.9DV_{DD33}$ | | | V |
| | | RGMII pins | | $DV_{DD15}$ - 0.4 | | | V |
| | | DDR2 memory controller pins | | 1.4 | | | V |
| $V_{OL}$ | Low-level output voltage | 3.3-V pins (except PCI-capable and I2C pins) | $DV_{DD33}$ = MIN, $I_{OL}$ = MAX | | | $0.22DV_{DD33}$ | V |
| | | PCI-capable pins(2) | $I_{OL}$ = 1.5 mA, $DV_{DD33}$ = 3.3 V | | | $0.1DV_{DD33}$ | V |
| | | I2C pins | Pulled up to 3.3 V, 3 mA sink current | | | 0.4 | V |
| | | RGMII pins | | | | 0.4 | V |
| | | DDR2 memory controller pins | | | | 0.4 | V |
| $I_I$ (3) | Input current [DC] | 3.3-V pins (except PCI-capable and I2C pins) | $V_I = V_{SS}$ to $DV_{DD33}$, pins without internal pullup or pulldown resistor | -1 | | 1 | uA |
| | | | $V_I = V_{SS}$ to $DV_{DD33}$, pins with internal pullup resistor | 50 | 100 | 400 | uA |
| | | | $V_I = V_{SS}$ to $DV_{DD33}$, pins with internal pulldown resistor | -400 | -100 | -50 | uA |
| | | I2C pins | $0.1DV_{DD33} \le V_I \le 0.9DV_{DD33}$ | -10 | | 10 | uA |
| | | PCI-capable pins(4) | | -1000 | | 1000 | uA |
| | | RGMII pins | | | | 0.4 | V |
| $I_{OH}$ | High-level output current [DC] | AECLKOUT, CLKR1/GP[0], CLKX1/GP[3], SYSCLK4/GP[1], EMU[18:0], CLKR0, CLKX0 | | | | -8 | mA |
| | | EMIF pins (except AECLKOUT), NMI, TOUT0L, TINP0L, TOUT1L, TINP1L, PCI_EN, EMAC-capable pins (except RGMII pins), RESETSTAT, McBSP-capable pins (except CLKR1/GP[0], CLKX1/GP[3], CLKR0, CLKX0), GP[7:4], and TDO | | | | -4 | mA |
| | | PCI-capable pins(2) | | | | -0.5 | mA |
| | | RGMII pins | | | | -8 | mA |
| | | DDR2 memory controller pins | | | | 4 | mA |

(1) For test conditions shown as MIN, MAX, or NOM, use the appropriate value specified in the recommended operating conditions table.
(2) These rated numbers are from the *PCI Local Bus Specification* (version 2.3). The DC specifications and AC specifications are defined in Table 4-3 and Table 4-4, respectively, of the *PCI Local Bus Specification*.
(3) $I_I$ applies to input-only pins and bi-directional pins. For input-only pins, $I_I$ indicates the input leakage current. For bi-directional pins, $I_I$ includes input leakage current and off-state (hi-Z) output leakage current.
(4) PCI input leakage currents include Hi-Z output leakage for all bidirectional buffers with 3-state outputs.

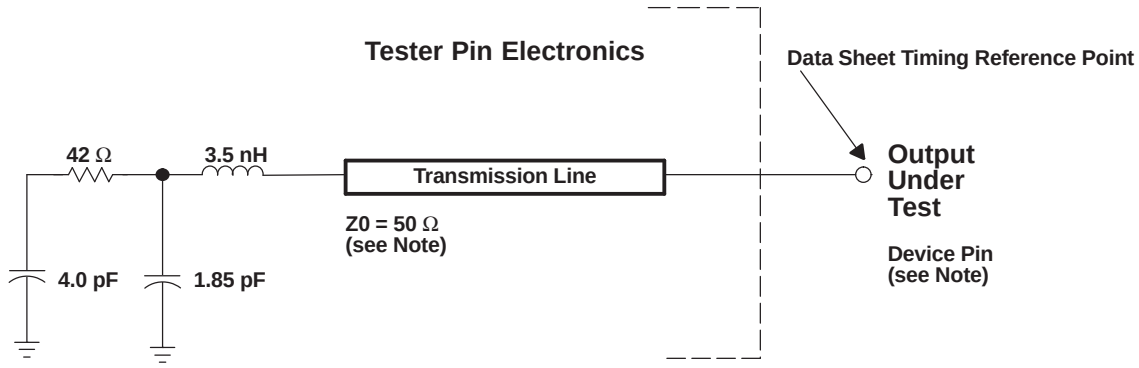
## Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Case Temperature (Unless Otherwise Noted) *(continued)*

| PARAMETER | | | TEST CONDITIONS[1] | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $I_{OL}$ | Low-level output current [DC] | AECLKOUT, CLKR1/GP[0], CLKX1/GP[3], SYSCLK4/GP[1], EMU[18:0], CLKR0, CLKX0 | | | | 8 | mA |
| | | EMIF pins (except AECLKOUT), NMI, TOUT0L, TINP0L, TOUTP1L, TINP1L, PCI_EN, EMAC-capable pins (except RGMII pins), $\overline{\text{RESETSTAT}}$, McBSP-capable pins (except CLKR1/GP[0], CLKX1/GP[3], CLKR0, CLKX0), GP[7:4], and TDO | | | | 4 | mA |
| | | PCI-capable pins[2] | | | | 1.5 | mA |
| | | RGMII pins | | | | 8 | mA |
| | | DDR2 memory controller pins | | | | -4 | mA |
| $I_{OZ}$ [5] | Off-state output current [DC] | 3.3-V pins | $V_O = DV_{DD33}$ or 0 V | -20 | | 20 | uA |
| $P_{CDD}$ | Core supply power[6] | | $CV_{DD}$ = 1.25 V, CPU frequency = 1000 MHz | | 1.66 | | W |
| | | | $CV_{DD}$ = 1.2 V, CPU frequency = 850 MHz | | 1.41 | | W |
| | | | $CV_{DD}$ = 1.2 V, CPU frequency = 720 MHz | | 1.29 | | W |
| $P_{DDD}$ | I/O supply power[6] | | $DV_{DD33}$ = 3.3 V, $DV_{DD18}$ = 1.8 V, PLLV1 = PLLV2 = $AV_{DLL1}$ = $AV_{DLL2}$ = 1.8 V, CPU frequency = 1000 MHz | | 0.53 | | W |
| | | | $DV_{DD33}$ = 3.3 V, $DV_{DD18}$ = 1.8 V, PLLV1 = PLLV2 = $AV_{DLL1}$ = $AV_{DLL2}$ = 1.8 V, CPU frequency = 850 MHz | | 0.53 | | W |
| | | | $DV_{DD33}$ = 3.3 V, $DV_{DD18}$ = 1.8 V, PLLV1 = PLLV2 = $AV_{DLL1}$ = $AV_{DLL2}$ = 1.8 V, CPU frequency = 720 MHz | | 0.52 | | W |
| $C_i$ | Input capacitance | | | | | 10 | pF |
| $C_o$ | Output capacitance | | | | | 10 | pF |

(5) $I_{OZ}$ applies to output-only pins, indicating off-state (hi-Z) output leakage current.
(6) Assumes the following conditions: 60% CPU utilization; DDR2 at 50% utilization (250 MHz), 50% writes, 32 bits, 50% bit switching; two 2-MHz McBSPs at 100% utilization, 50% switching; two 75-MHz Timers at 100% utilization; device configured for HPI32 mode with pull-up resistors on HPI pins; room temperature (25°C). The actual current draw is highly application-dependent. For more details on core and I/O activity, see the *TMS320C6455/54 Power Consumption Summary* application report (literature number SPRAAE8).

# 7 C64x+ Peripheral Information and Electrical Specifications

## 7.1 Parameter Information




NOTE: 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 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) 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.

**Figure 7-1. 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.

### *7.1.1 3.3-V Signal Transition Levels*

All input and output timing parameters are referenced to 1.5 V for both "0" and "1" logic levels.

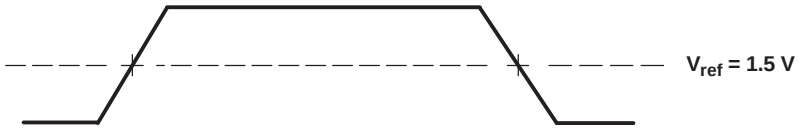



**Figure 7-2. Input and Output Voltage Reference Levels for AC Timing Measurements**

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.

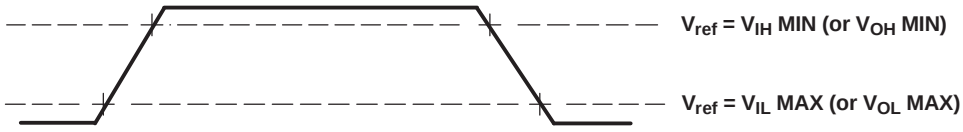



**Figure 7-3. Rise and Fall Transition Time Voltage Reference Levels**

### *7.1.2 3.3-V Signal Transition Rates*

All timings are tested with an input edge rate of 4 volts per nanosecond (4 V/ns).

### 7.1.3 Timing Parameters and Board Routing Analysis

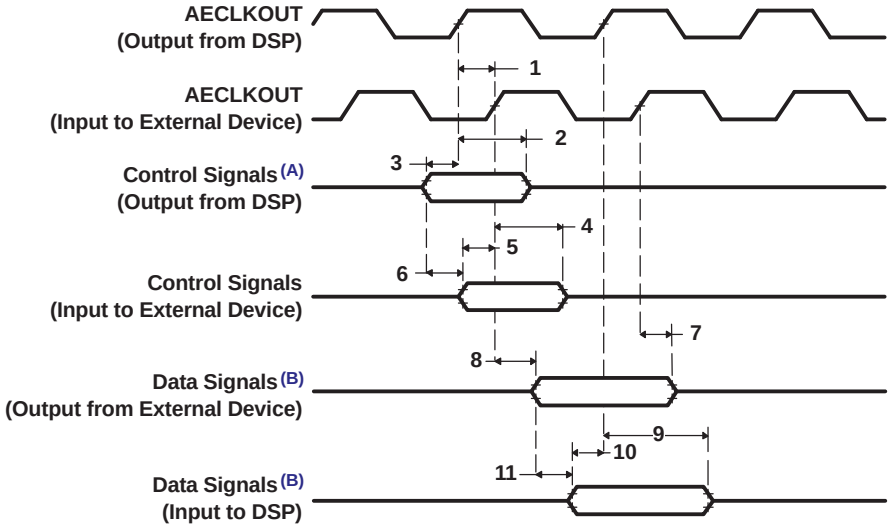
The timing parameter values specified in this data sheet do *not* include delays by board routings. As a good board design practice, such delays must *always* be taken into account. Timing values may be adjusted by increasing/decreasing such delays. TI recommends utilizing the available I/O buffer information specification (IBIS) models to analyze the timing characteristics correctly. To properly use IBIS models to attain accurate timing analysis for a given system, see the *Using IBIS Models for Timing Analysis* application report (literature number SPRA839). If needed, external logic hardware such as buffers may be used to compensate any timing differences.

For inputs, timing is most impacted by the round-trip propagation delay from the DSP to the external device and from the external device to the DSP. This round-trip delay tends to negatively impact the input setup time margin, but also tends to improve the input hold time margins (see Table 7-1 and Figure 7-4).

Figure 7-4 represents a general transfer between the DSP and an external device. The figure also represents board route delays and how they are perceived by the DSP and the external device.

**Table 7-1. Board-Level Timing Example (see Figure 7-4)**

| NO. | DESCRIPTION |
|---|---|
| 1 | Clock route delay |
| 2 | Minimum DSP hold time |
| 3 | Minimum DSP setup time |
| 4 | External device hold time requirement |
| 5 | External device setup time requirement |
| 6 | Control signal route delay |
| 7 | External device hold time |
| 8 | External device access time |
| 9 | DSP hold time requirement |
| 10 | DSP setup time requirement |
| 11 | Data route delay |




A. Control signals include data for Writes.

B. Data signals are generated during Reads from an external device.

**Figure 7-4. Board-Level Input/Output Timings**

## 7.2 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.

## 7.3 Power Supplies

### *7.3.1 Power-Supply Sequencing*

TI recommends the power-supply sequence shown in Figure 7-5. After the $DV_{DD33}$ supply is stable, the remaining power supplies can be powered up at the same time as $CV_{DD}$ as long as their supply voltage never exceeds the $CV_{DD}$ voltage during powerup. Some TI power-supply devices include features that facilitate power sequencing; for example, Auto-Track or Slow-Start/Enable features. For more information, visit www.ti.com/dsppower.

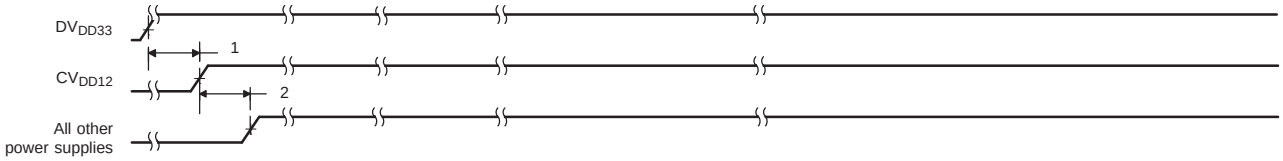



**Figure 7-5. Power-Supply Sequence**

**Table 7-2. Timing Requirements for Power-Supply Sequence**

| NO. | | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(DVDD33-CVDD12)}$ | Setup time, $DV_{DD33}$ supply stable before $CV_{DD12}$ supply stable | 0.5 | 200 | ms |
| 2 | $t_{su(CVDD12-ALLSUP)}$ | Setup time, $CV_{DD12}$ supply stable before all other supplies stable | 0 | 200 | ms |

### *7.3.2 Power-Supply Decoupling*

In order to properly decouple the supply planes from system noise, place as many capacitors (caps) as possible close to the DSP. These caps need to be close to the DSP, no more than 1.25 cm maximum distance to be effective. Physically smaller caps are better, such as 0402, but need to be evaluated from a yield/manufacturing point-of-view. Parasitic inductance limits the effectiveness of the decoupling capacitors, therefore physically smaller capacitors should be used while maintaining the largest available capacitance value. As with the selection of any component, verification of capacitor availability over the product's production lifetime should be considered.

### *7.3.3 Power-Down Operation*

One of the power goals for the C6454 device is to reduce power dissipation due to unused peripherals. There are different ways to power down peripherals on the C6454 device.

Some peripherals can be statically powered down at device reset through the device configuration pins (see Section 3.1, *Device Configuration at Device Reset*). Once in a static power-down state, the peripheral is held in reset and its clock is turned off. Peripherals cannot be enabled once they are in a static power-down state. To take a peripheral out of the static power-down state, a device reset must be executed with a different configuration pin setting.

After device reset, all peripherals on the C6454 device are in a disabled state and must be enabled by software before being used. It is possible to enable only the peripherals needed by the application while keeping the rest disabled. Note that peripherals in a disabled state are held in reset with their clocks gated. For more information on how to enable peripherals, see Section 3.3, *Peripheral Selection After Device Reset*.

Peripherals used for booting, like I2C and HPI, are automatically enabled after device reset. It is not possible to disable these peripherals after the boot process is complete.

The C64x+ Megamodule also allows for software-driven power-down management for all of the C64x+ megamodule components through its Power-Down Controller (PDC). The CPU can power-down part or the entire C64x+ megamodule through the power-down controller based on its own execution thread or in response to an external stimulus from a host or global controller. More information on the power-down features of the C64x+ Megamodule can be found in the *TMS320C64x+ Megamodule Reference Guide* (literature number SPRU871).

### 7.3.4 Preserving Boundary-Scan Functionality on RGMII and DDR2 Memory Pins

When the RGMII mode of the EMAC is not used, the $DV_{DD15}$, $DV_{DD15MON}$, $V_{REFHSTL}$, RSV13, and RSV14 pins can be connected directly to ground ($V_{SS}$) to save power. However, this will prevent boundary-scan from functioning on the RGMII pins of the EMAC. To preserve boundary-scan functionality on the RGMII pins, $DV_{DD15}$, $V_{REFHSTL}$, RSV14, and RSV13 should be connected as follows:

- $DV_{DD15}$ and $DV_{DD15MON}$ - connect these pins to the 1.8-V I/O supply ($DV_{DD18}$).
- $V_{REFHSTL}$ - connect to a voltage of $DV_{DD18}$/2. The $DV_{DD18}$/2 voltage can be generated directly from the $DV_{DD18}$ supply using two 1-kΩ resistors to form a resistor divider circuit.
- RSV13 - connect this pin to ground ($V_{SS}$) via a 200-Ω resistor.
- RSV14 - connect this pin to the 1.8-V I/O supply ($DV_{DD18}$) via a 200-Ω resistor.

Similarly, when the DDR2 Memory Controller is not used, the $V_{REFSSTL}$, RSV11, and RSV12 pins can be connected directly to ground ($V_{SS}$) to save power. However, this will prevent boundary-scan from functioning on the DDR2 Memory Controller pins. To preserve boundary-scan functionality on the DDR2 Memory Controller pins, $V_{REFSSTL}$, RSV11, and RSV12 should be connected as follows:

- $V_{REFSSTL}$ - connect to a voltage of $DV_{DD18}$/2. The $DV_{DD18}$/2 voltage can be generated directly from the $DV_{DD18}$ supply using two 1-kΩ resistors to form a resistor divider circuit.
- RSV11 - connect this pin to ground ($V_{SS}$) via a 200-Ω resistor.
- RSV12 - connect this pin to the 1.8-V I/O supply ($DV_{DD18}$) via a 200-Ω resistor.

## 7.4 Enhanced Direct Memory Access (EDMA3) Controller

The primary purpose of the EDMA3 is to service user-programmed data transfers between two memory-mapped slave endpoints on the device. The EDMA3 services software-driven paging transfers (e.g., data movement between external memory and internal memory), performs sorting or subframe extraction of various data structures, services event driven peripherals such as the McBSP, and offloads data transfers from the device CPU.

The EDMA3 includes the following features:

- Fully orthogonal transfer description
  - 3 transfer dimensions: array (multiple bytes), frame (multiple arrays), and block (multiple frames)
  - Single event can trigger transfer of array, frame, or entire block
  - Independent indexes on source and destination
- Flexible transfer definition:
  - Increment or FIFO transfer addressing modes
  - Linking mechanism allows for ping-pong buffering, circular buffering, and repetitive/continuous transfers, all with no CPU intervention
  - Chaining allows multiple transfers to execute with one event
- 256 PaRAM entries
  - Used to define transfer context for channels
  - Each PaRAM entry can be used as a DMA entry, QDMA entry, or link entry
- 64 DMA channels
  - Manually triggered (CPU writes to channel controller register), external event triggered, and chain triggered (completion of one transfer triggers another)
- 4 Quick DMA (QDMA) channels
  - Used for software-driven transfers
  - Triggered upon writing to a single PaRAM set entry
- 4 transfer controllers/event queues with programmable system-level priority
- Interrupt generation for transfer completion and error conditions
- Memory protection support
  - Active memory protection for accesses to PaRAM and registers
- Debug visibility
  - Queue watermarking/threshold allows detection of maximum usage of event queues
  - Error and status recording to facilitate debug

Each of the transfer controllers has a direct connection to the switched central resource (SCR).

**NOTE**

Although the transfer controllers are directly connected to the SCR, they can only access certain device resources. For example, only transfer controller 1 (TC1) can access the McBSPs. lists the device resources that can be accessed by each of the transfer controllers.

### 7.4.1 EDMA3 Device-Specific Information

The EDMA supports two addressing modes: constant addressing and increment addressing mode. On the C6454 DSP, constant addressing mode is not supported by any peripheral or internal memory. For more information on these two addressing modes, see the *TMS320C645x DSP Enhanced DMA (EDMA3) Controller User's Guide* (literature number SPRU966).

A DSP interrupt must be generated at the end of an HPI or PCI boot operation to begin execution of the loaded application. Since the DSP interrupt generated by the HPI and PCI is mapped to the EDMA event DSP_EVT (DMA channel 0), it will get recorded in bit 0 of the EDMA Event Register (ER). This event must be cleared by software before triggering transfers on DMA channel 0. The EDMA3 on the C6454 DSP supports active memory protection, but it does not support proxied memory protection.

### 7.4.2 EDMA3 Channel Synchronization Events

The EDMA3 supports up to 64 DMA channels that can be used to service system peripherals and to move data between system memories. DMA channels can be triggered by synchronization events generated by system peripherals. Table 7-3 lists the source of the synchronization event associated with each of the DMA channels. On the C6454 device, the association of each synchronization event and DMA channel is fixed and cannot be reprogrammed.

For more detailed information on the EDMA3 module and how EDMA3 events are enabled, captured, processed, prioritized, linked, chained, and cleared, etc., see the *TMS320C645x DSP Enhanced DMA (EDMA3) Controller User's Guide* (literature number SPRU966).

**Table 7-3. C6454 EDMA3 Channel Synchronization Events(1)**

| EDMA CHANNEL | BINARY | EVENT NAME | EVENT DESCRIPTION |
|---|---|---|---|
| 0(2) | 000 0000 | DSP_EVT | HPI/PCI-to-DSP event |
| 1 | 000 0001 | TEVTLO0 | Timer 0 lower counter event |
| 2 | 000 0010 | TEVTHI0 | Timer 0 high counter event |
| 3 | 000 0011 | - | None |
| 4 | 000 0100 | - | None |
| 5 | 000 0101 | - | None |
| 6 | 000 0110 | - | None |
| 7 | 000 0111 | - | None |
| 8 | 000 1000 | - | None |
| 9 | 000 1001 | - | None |
| 10 | 000 1010 | - | None |
| 11 | 000 1011 | - | None |
| 12 | 000 1100 | XEVT0 | McBSP0 transmit event |
| 13 | 000 1101 | REVT0 | McBSP0 receive event |
| 14 | 000 1110 | XEVT1 | McBSP1 transmit event |
| 15 | 000 1111 | REVT1 | McBSP1 receive event |
| 16 | 001 0000 | TEVTLO1 | Timer 1 lower counter event |
| 17 | 001 0001 | TEVTHI1 | Timer 1 high counter event |
| 18-43 | - | - | None |
| 44 | 010 1100 | ICREVT | I2C receive event |
| 45 | 010 1101 | ICXEVT | I2C transmit event |
| 46-47 | - | - | None |

(1) In addition to the events shown in this table, each of the 64 channels can also be synchronized with the transfer completion or alternate transfer completion events. For more detailed information on EDMA event-transfer chaining, see the *TMS320C645x DSP Enhanced DMA (EDMA3) Controller User's Guide* (literature number SPRU966).

(2) HPI boot and PCI boot are terminated using a DSP interrupt. The DSP interrupt is registered in bit 0 (channel 0) of the EDMA Event Register (ER). This event must be cleared by software before triggering transfers on DMA channel 0.

**Table 7-3. C6454 EDMA3 Channel Synchronization Events[1] (continued)**

| EDMA CHANNEL | BINARY | EVENT NAME | EVENT DESCRIPTION |
|---|---|---|---|
| 48 | 011 0000 | GPINT0 | GPIO event 0 |
| 49 | 011 0001 | GPINT1 | GPIO event 1 |
| 50 | 011 0010 | GPINT2 | GPIO event 2 |
| 51 | 011 0011 | GPINT3 | GPIO event 3 |
| 52 | 011 0100 | GPINT4 | GPIO event 4 |
| 53 | 011 0101 | GPINT5 | GPIO event 5 |
| 54 | 011 0110 | GPINT6 | GPIO event 6 |
| 55 | 011 0111 | GPINT7 | GPIO event 7 |
| 56 | 011 1000 | GPINT8 | GPIO event 8 |
| 57 | 011 1001 | GPINT9 | GPIO event 9 |
| 58 | 011 1010 | GPINT10 | GPIO event 10 |
| 59 | 011 1011 | GPINT11 | GPIO event 11 |
| 60 | 011 1100 | GPINT12 | GPIO event 12 |
| 61 | 011 1101 | GPINT13 | GPIO event 13 |
| 62 | 011 1110 | GPINT14 | GPIO event 14 |
| 63 | 011 1111 | GPINT15 | GPIO event 15 |

### 7.4.3 EDMA3 Peripheral Register Descriptions

**Table 7-4. EDMA3 Channel Controller Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 0000 | PID | Peripheral ID Register |
| 02A0 0004 | CCCFG | EDMA3CC Configuration Register |
| 02A0 0008 - 02A0 00FC | - | Reserved |
| 02A0 0100 | DCHMAP0 | DMA Channel 0 Mapping Register |
| 02A0 0104 | DCHMAP1 | DMA Channel 1 Mapping Register |
| 02A0 0108 | DCHMAP2 | DMA Channel 2 Mapping Register |
| 02A0 010C | DCHMAP3 | DMA Channel 3 Mapping Register |
| 02A0 0110 | DCHMAP4 | DMA Channel 4 Mapping Register |
| 02A0 0114 | DCHMAP5 | DMA Channel 5 Mapping Register |
| 02A0 0118 | DCHMAP6 | DMA Channel 6 Mapping Register |
| 02A0 011C | DCHMAP7 | DMA Channel 7 Mapping Register |
| 02A0 0120 | DCHMAP8 | DMA Channel 8 Mapping Register |
| 02A0 0124 | DCHMAP9 | DMA Channel 9 Mapping Register |
| 02A0 0128 | DCHMAP10 | DMA Channel 10 Mapping Register |
| 02A0 012C | DCHMAP11 | DMA Channel 11 Mapping Register |
| 02A0 0130 | DCHMAP12 | DMA Channel 12 Mapping Register |
| 02A0 0134 | DCHMAP13 | DMA Channel 13 Mapping Register |
| 02A0 0138 | DCHMAP14 | DMA Channel 14 Mapping Register |
| 02A0 013C | DCHMAP15 | DMA Channel 15 Mapping Register |
| 02A0 0140 | DCHMAP16 | DMA Channel 16 Mapping Register |
| 02A0 0144 | DCHMAP17 | DMA Channel 17 Mapping Register |
| 02A0 0148 | DCHMAP18 | DMA Channel 18 Mapping Register |
| 02A0 014C | DCHMAP19 | DMA Channel 19 Mapping Register |
| 02A0 0150 | DCHMAP20 | DMA Channel 20 Mapping Register |
| 02A0 0154 | DCHMAP21 | DMA Channel 21 Mapping Register |

**Table 7-4. EDMA3 Channel Controller Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 0158 | DCHMAP22 | DMA Channel 22 Mapping Register |
| 02A0 015C | DCHMAP23 | DMA Channel 23 Mapping Register |
| 02A0 0160 | DCHMAP24 | DMA Channel 24 Mapping Register |
| 02A0 0164 | DCHMAP25 | DMA Channel 25 Mapping Register |
| 02A0 0168 | DCHMAP26 | DMA Channel 26 Mapping Register |
| 02A0 016C | DCHMAP27 | DMA Channel 27 Mapping Register |
| 02A0 0170 | DCHMAP28 | DMA Channel 28 Mapping Register |
| 02A0 0174 | DCHMAP29 | DMA Channel 29 Mapping Register |
| 02A0 0178 | DCHMAP30 | DMA Channel 30 Mapping Register |
| 02A0 017C | DCHMAP31 | DMA Channel 31 Mapping Register |
| 02A0 0180 | DCHMAP32 | DMA Channel 32 Mapping Register |
| 02A0 0184 | DCHMAP33 | DMA Channel 33 Mapping Register |
| 02A0 0188 | DCHMAP34 | DMA Channel 34 Mapping Register |
| 02A0 018C | DCHMAP35 | DMA Channel 35 Mapping Register |
| 02A0 0190 | DCHMAP36 | DMA Channel 36 Mapping Register |
| 02A0 0194 | DCHMAP37 | DMA Channel 37 Mapping Register |
| 02A0 0198 | DCHMAP38 | DMA Channel 38 Mapping Register |
| 02A0 019C | DCHMAP39 | DMA Channel 39 Mapping Register |
| 02A0 01A0 | DCHMAP40 | DMA Channel 40 Mapping Register |
| 02A0 01A4 | DCHMAP41 | DMA Channel 41 Mapping Register |
| 02A0 01A8 | DCHMAP42 | DMA Channel 42 Mapping Register |
| 02A0 01AC | DCHMAP43 | DMA Channel 43 Mapping Register |
| 02A0 01B0 | DCHMAP44 | DMA Channel 44 Mapping Register |
| 02A0 01B4 | DCHMAP45 | DMA Channel 45 Mapping Register |
| 02A0 01B8 | DCHMAP46 | DMA Channel 46 Mapping Register |
| 02A0 01BC | DCHMAP47 | DMA Channel 47 Mapping Register |
| 02A0 01C0 | DCHMAP48 | DMA Channel 48 Mapping Register |
| 02A0 01C4 | DCHMAP49 | DMA Channel 49 Mapping Register |
| 02A0 01C8 | DCHMAP50 | DMA Channel 50 Mapping Register |
| 02A0 01CC | DCHMAP51 | DMA Channel 51 Mapping Register |
| 02A0 01D0 | DCHMAP52 | DMA Channel 52 Mapping Register |
| 02A0 01D4 | DCHMAP53 | DMA Channel 53 Mapping Register |
| 02A0 01D8 | DCHMAP54 | DMA Channel 54 Mapping Register |
| 02A0 01DC | DCHMAP55 | DMA Channel 55 Mapping Register |
| 02A0 01E0 | DCHMAP56 | DMA Channel 56 Mapping Register |
| 02A0 01E4 | DCHMAP57 | DMA Channel 57 Mapping Register |
| 02A0 01E8 | DCHMAP58 | DMA Channel 58 Mapping Register |
| 02A0 01EC | DCHMAP59 | DMA Channel 59 Mapping Register |
| 02A0 01F0 | DCHMAP60 | DMA Channel 60 Mapping Register |
| 02A0 01F4 | DCHMAP61 | DMA Channel 61 Mapping Register |
| 02A0 01F8 | DCHMAP62 | DMA Channel 62 Mapping Register |
| 02A0 01FC | DCHMAP63 | DMA Channel 63 Mapping Register |
| 02A0 0200 | QCHMAP0 | QDMA Channel 0 Mapping Register |
| 02A0 0204 | QCHMAP1 | QDMA Channel 1 Mapping Register |
| 02A0 0208 | QCHMAP2 | QDMA Channel 2 Mapping Register |
| 02A0 020C | QCHMAP3 | QDMA Channel 3 Mapping Register |
| 02A0 0210 - 02A0 021C | - | Reserved |

**Table 7-4. EDMA3 Channel Controller Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 0220 - 02A0 023C | - | Reserved |
| 02A0 0240 | DMAQNUM0 | DMA Queue Number Register 0 |
| 02A0 0244 | DMAQNUM1 | DMA Queue Number Register 1 |
| 02A0 0248 | DMAQNUM2 | DMA Queue Number Register 2 |
| 02A0 024C | DMAQNUM3 | DMA Queue Number Register 3 |
| 02A0 0250 | DMAQNUM4 | DMA Queue Number Register 4 |
| 02A0 0254 | DMAQNUM5 | DMA Queue Number Register 5 |
| 02A0 0258 | DMAQNUM6 | DMA Queue Number Register 6 |
| 02A0 025C | DMAQNUM7 | DMA Queue Number Register 7 |
| 02A0 0260 | QDMAQNUM | QDMA Queue Number Register |
| 02A0 0264 - 02A0 0280 | - | Reserved |
| 02A0 0284 | QUEPRI | Queue Priority Register |
| 02A0 0288 - 02A0 02FC | - | Reserved |
| 02A0 0300 | EMR | Event Missed Register |
| 02A0 0304 | EMRH | Event MissedRegister High |
| 02A0 0308 | EMCR | Event Missed Clear Register |
| 02A0 030C | EMCRH | Event Missed Clear Register High |
| 02A0 0310 | QEMR | QDMA Event Missed Register |
| 02A0 0314 | QEMCR | QDMA Event Missed Clear Register |
| 02A0 0318 | CCERR | EDMA3CC Error Register |
| 02A0 031C | CCERRCLR | EDMA3CC Error Clear Register |
| 02A0 0320 | EEVAL | Error Evaluate Register |
| 02A0 0324 - 02A0 033C | - | Reserved |
| 02A0 0340 | DRAE0 | DMA Region Access Enable Register for Region 0 |
| 02A0 0344 | DRAEH0 | DMA Region Access Enable Register High for Region 0 |
| 02A0 0348 | DRAE1 | DMA Region Access Enable Register for Region 1 |
| 02A0 034C | DRAEH1 | DMA Region Access Enable Register High for Region 1 |
| 02A0 0350 | DRAE2 | DMA Region Access Enable Register for Region 2 |
| 02A0 0354 | DRAEH2 | DMA Region Access Enable Register High for Region 2 |
| 02A0 0358 | DRAE3 | DMA Region Access Enable Register for Region 3 |
| 02A0 035C | DRAEH3 | DMA Region Access Enable Register High for Region 3 |
| 02A0 0360 | DRAE4 | DMA Region Access Enable Register for Region 4 |
| 02A0 0364 | DRAEH4 | DMA Region Access Enable Register High for Region 4 |
| 02A0 0368 | DRAE5 | DMA Region Access Enable Register for Region 5 |
| 02A0 036C | DRAEH5 | DMA Region Access Enable Register High for Region 5 |
| 02A0 0370 | DRAE6 | DMA Region Access Enable Register for Region 6 |
| 02A0 0374 | DRAEH6 | DMA Region Access Enable Register High for Region 6 |
| 02A0 0378 | DRAE7 | DMA Region Access Enable Register for Region 7 |
| 02A0 037C | DRAEH7 | DMA Region Access Enable Register High for Region 7 |
| 02A0 0380 | QRAE0 | QDMA Region Access Enable Register for Region 0 |
| 02A0 0384 | QRAE1 | QDMA Region Access Enable Register for Region 1 |
| 02A0 0388 | QRAE2 | QDMA Region Access Enable Register for Region 2 |
| 02A0 038C | QRAE3 | QDMA Region Access Enable Register for Region 3 |
| 02A0 0390 - 02A0 039C | - | Reserved |
| 02A0 0400 | Q0E0 | Event Queue 0 Entry Register 0 |
| 02A0 0404 | Q0E1 | Event Queue 0 Entry Register 1 |
| 02A0 0408 | Q0E2 | Event Queue 0 Entry Register 2 |

**Table 7-4. EDMA3 Channel Controller Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 040C | Q0E3 | Event Queue 0 Entry Register 3 |
| 02A0 0410 | Q0E4 | Event Queue 0 Entry Register 4 |
| 02A0 0414 | Q0E5 | Event Queue 0 Entry Register 5 |
| 02A0 0418 | Q0E6 | Event Queue 0 Entry Register 6 |
| 02A0 041C | Q0E7 | Event Queue 0 Entry Register 7 |
| 02A0 0420 | Q0E8 | Event Queue 0 Entry Register 8 |
| 02A0 0424 | Q0E9 | Event Queue 0 Entry Register 9 |
| 02A0 0428 | Q0E10 | Event Queue 0 Entry Register 10 |
| 02A0 042C | Q0E11 | Event Queue 0 Entry Register 11 |
| 02A0 0430 | Q0E12 | Event Queue 0 Entry Register 12 |
| 02A0 0434 | Q0E13 | Event Queue 0 Entry Register 13 |
| 02A0 0438 | Q0E14 | Event Queue 0 Entry Register 14 |
| 02A0 043C | Q0E15 | Event Queue 0 Entry Register 15 |
| 02A0 0440 | Q1E0 | Event Queue 1 Entry Register 0 |
| 02A0 0444 | Q1E1 | Event Queue 1 Entry Register 1 |
| 02A0 0448 | Q1E2 | Event Queue 1 Entry Register 2 |
| 02A0 044C | Q1E3 | Event Queue 1 Entry Register 3 |
| 02A0 0450 | Q1E4 | Event Queue 1 Entry Register 4 |
| 02A0 0454 | Q1E5 | Event Queue 1 Entry Register 5 |
| 02A0 0458 | Q1E6 | Event Queue 1 Entry Register 6 |
| 02A0 045C | Q1E7 | Event Queue 1 Entry Register 7 |
| 02A0 0460 | Q1E8 | Event Queue 1 Entry Register 8 |
| 02A0 0464 | Q1E9 | Event Queue 1 Entry Register 9 |
| 02A0 0468 | Q1E10 | Event Queue 1 Entry Register 10 |
| 02A0 046C | Q1E11 | Event Queue 1 Entry Register 11 |
| 02A0 0470 | Q1E12 | Event Queue 1 Entry Register 12 |
| 02A0 0474 | Q1E13 | Event Queue 1 Entry Register 13 |
| 02A0 0478 | Q1E14 | Event Queue 1 Entry Register 14 |
| 02A0 047C | Q1E15 | Event Queue 1 Entry Register 15 |
| 02A0 0480 | Q2E0 | Event Queue 2 Entry Register 0 |
| 02A0 0484 | Q2E1 | Event Queue 2 Entry Register 1 |
| 02A0 0488 | Q2E2 | Event Queue 2 Entry Register 2 |
| 02A0 048C | Q2E3 | Event Queue 2 Entry Register 3 |
| 02A0 0490 | Q2E4 | Event Queue 2 Entry Register 4 |
| 02A0 0494 | Q2E5 | Event Queue 2 Entry Register 5 |
| 02A0 0498 | Q2E6 | Event Queue 2 Entry Register 6 |
| 02A0 049C | Q2E7 | Event Queue 2 Entry Register 7 |
| 02A0 04A0 | Q2E8 | Event Queue 2 Entry Register 8 |
| 02A0 04A4 | Q2E9 | Event Queue 2 Entry Register 9 |
| 02A0 04A8 | Q2E10 | Event Queue 2 Entry Register 10 |
| 02A0 04AC | Q2E11 | Event Queue 2 Entry Register 11 |
| 02A0 04B0 | Q2E12 | Event Queue 2 Entry Register 12 |
| 02A0 04B4 | Q2E13 | Event Queue 2 Entry Register 13 |
| 02A0 04B8 | Q2E14 | Event Queue 2 Entry Register 14 |
| 02A0 04BC | Q2E15 | Event Queue 2 Entry Register 15 |
| 02A0 04C0 | Q3E0 | Event Queue 3 Entry Register 0 |
| 02A0 04C4 | Q3E1 | Event Queue 3 Entry Register 1 |

**Table 7-4. EDMA3 Channel Controller Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 04C8 | Q3E2 | Event Queue 3 Entry Register 2 |
| 02A0 04CC | Q3E3 | Event Queue 3 Entry Register 3 |
| 02A0 04D0 | Q3E4 | Event Queue 3 Entry Register 4 |
| 02A0 04D4 | Q3E5 | Event Queue 3 Entry Register 5 |
| 02A0 04D8 | Q3E6 | Event Queue 3 Entry Register 6 |
| 02A0 04DC | Q3E7 | Event Queue 3 Entry Register 7 |
| 02A0 04E0 | Q3E8 | Event Queue 3 Entry Register 8 |
| 02A0 04E4 | Q3E9 | Event Queue 3 Entry Register 9 |
| 02A0 04E8 | Q3E10 | Event Queue 3 Entry Register 10 |
| 02A0 04EC | Q3E11 | Event Queue 3 Entry Register 11 |
| 02A0 04F0 | Q3E12 | Event Queue 3 Entry Register 12 |
| 02A0 04F4 | Q3E13 | Event Queue 3 Entry Register 13 |
| 02A0 04F8 | Q3E14 | Event Queue 3 Entry Register 14 |
| 02A0 04FC | Q3E15 | Event Queue 3 Entry Register 15 |
| 02A0 0500 - 02A0 051C | - | Reserved |
| 02A0 0520 - 02A0 05FC | - | Reserved |
| 02A0 0600 | QSTAT0 | Queue Status Register 0 |
| 02A0 0604 | QSTAT1 | Queue Status Register 1 |
| 02A0 0608 | QSTAT2 | Queue Status Register 2 |
| 02A0 060C | QSTAT3 | Queue Status Register 3 |
| 02A0 0610 - 02A0 061C | - | Reserved |
| 02A0 0620 | QWMTHRA | Queue Watermark Threshold A Register |
| 02A0 0624 - 02A0 063C | - | Reserved |
| 02A0 0640 | CCSTAT | EDMA3CC Status Register |
| 02A0 0644 - 02A0 06FC | - | Reserved |
| 02A0 0700 - 02A0 07FC | - | Reserved |
| 02A0 0800 | MPFAR | Memory Protection Fault Address Register |
| 02A0 0804 | MPFSR | Memory Protection Fault Status Register |
| 02A0 0808 | MPFCR | Memory Protection Fault Command Register |
| 02A0 080C | MPPA0 | Memory Protection Page Attribute Register 0 |
| 02A0 0810 | MPPA1 | Memory Protection Page Attribute Register 1 |
| 02A0 0814 | MPPA2 | Memory Protection Page Attribute Register 2 |
| 02A0 0818 | MPPA3 | Memory Protection Page Attribute Register 3 |
| 02A0 081C | MPPA4 | Memory Protection Page Attribute Register 4 |
| 02A0 0820 | MPPA5 | Memory Protection Page Attribute Register 5 |
| 02A0 0824 | MPPA6 | Memory Protection Page Attribute Register 6 |
| 02A0 0828 | MPPA7 | Memory Protection Page Attribute Register 7 |
| 02A0 082C - 02A0 0FFC | - | Reserved |
| 02A0 1000 | ER | Event Register |
| 02A0 1004 | ERH | Event Register High |
| 02A0 1008 | ECR | Event Clear Register |
| 02A0 100C | ECRH | Event Clear Register High |
| 02A0 1010 | ESR | Event Set Register |
| 02A0 1014 | ESRH | Event Set Register High |
| 02A0 1018 | CER | Chained Event Register |
| 02A0 101C | CERH | Chained Event Register High |
| 02A0 1020 | EER | Event Enable Register |

**Table 7-4. EDMA3 Channel Controller Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 1024 | EERH | Event Enable Register High |
| 02A0 1028 | EECR | Event Enable Clear Register |
| 02A0 102C | EECRH | Event Enable Clear Register High |
| 02A0 1030 | EESR | Event Enable Set Register |
| 02A0 1034 | EESRH | Event Enable Set Register High |
| 02A0 1038 | SER | Secondary Event Register |
| 02A0 103C | SERH | Secondary Event Register High |
| 02A0 1040 | SECR | Secondary Event Clear Register |
| 02A0 1044 | SECRH | Secondary Event Clear Register High |
| 02A0 1048 - 02A0 104C | - | Reserved |
| 02A0 1050 | IER | Interrupt Enable Register |
| 02A0 1054 | IERH | Interrupt Enable High Register |
| 02A0 1058 | IECR | Interrupt Enable Clear Register |
| 02A0 105C | IECRH | Interrupt Enable Clear High Register |
| 02A0 1060 | IESR | Interrupt Enable Set Register |
| 02A0 1064 | IESRH | Interrupt Enable Set High Register |
| 02A0 1068 | IPR | Interrupt Pending Register |
| 02A0 106C | IPRH | Interrupt Pending High Register |
| 02A0 1070 | ICR | Interrupt Clear Register |
| 02A0 1074 | ICRH | Interrupt Clear High Register |
| 02A0 1078 | IEVAL | Interrupt Evaluate Register |
| 02A0 107C | - | Reserved |
| 02A0 1080 | QER | QDMA Event Register |
| 02A0 1084 | QEER | QDMA Event Enable Register |
| 02A0 1088 | QEECR | QDMA Event Enable Clear Register |
| 02A0 108C | QEESR | QDMA Event Enable Set Register |
| 02A0 1090 | QSER | QDMA Secondary Event Register |
| 02A0 1094 | QSECR | QDMA Secondary Event Clear Register |
| 02A0 1098 - 02A0 1FFF | - | Reserved |
| **Shadow Region 0 Channel Registers** | | |
| 02A0 2000 | ER | Event Register |
| 02A0 2004 | ERH | Event Register High |
| 02A0 2008 | ECR | Event Clear Register |
| 02A0 200C | ECRH | Event Clear Register High |
| 02A0 2010 | ESR | Event Set Register |
| 02A0 2014 | ESRH | Event Set Register High |
| 02A0 2018 | CER | Chained Event Register |
| 02A0 201C | CERH | Chained Event Register Hig |
| 02A0 2020 | EER | Event Enable Register |
| 02A0 2024 | EERH | Event Enable Register High |
| 02A0 2028 | EECR | Event Enable Clear Register |
| 02A0 202C | EECRH | Event Enable Clear Register High |
| 02A0 2030 | EESR | Event Enable Set Register |
| 02A0 2034 | EESRH | Event Enable Set Register High |
| 02A0 2038 | SER | Secondary Event Register |
| 02A0 203C | SERH | Secondary Event Register High |
| 02A0 2040 | SECR | Secondary Event Clear Register |

**Table 7-4. EDMA3 Channel Controller Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 2044 | SECRH | Secondary Event Clear Register High |
| 02A0 2048 - 02A0 204C | - | Reserved |
| 02A0 2050 | IER | Interrupt Enable Register |
| 02A0 2054 | IERH | Interrupt Enable Register High |
| 02A0 2058 | IECR | Interrupt Enable Clear Register |
| 02A0 205C | IECRH | Interrupt Enable Clear Register High |
| 02A0 2060 | IESR | Interrupt Enable Set Register |
| 02A0 2064 | IESRH | Interrupt Enable Set Register High |
| 02A0 2068 | IPR | Interrupt Pending Register |
| 02A0 206C | IPRH | Interrupt Pending Register High |
| 02A0 2070 | ICR | Interrupt Clear Register |
| 02A0 2074 | ICRH | Interrupt Clear Register High |
| 02A0 2078 | IEVAL | Interrupt Evaluate Register |
| 02A0 207C | - | Reserved |
| 02A0 2080 | QER | QDMA Event Register |
| 02A0 2084 | QEER | QDMA Event Enable Register |
| 02A0 2088 | QEECR | QDMA Event Enable Clear Register |
| 02A0 208C | QEESR | QDMA Event Enable Set Register |
| 02A0 2090 | QSER | QDMA Secondary Event Register |
| 02A0 2094 | QSECR | QDMA Secondary Event Clear Register |
| 02A0 2098 - 02A0 23FF | - | Reserved |
| 02A0 2400 - 02A0 2497 | - | Shadow Region 2 Channel Registers |
| 02A0 2498 - 02A0 25FF | - | Reserved |
| 02A0 2600 - 02A0 2697 | - | Shadow Region 3 Channel Registers |
| 02A0 2698 - 02A0 27FF | - | Reserved |
| 02A0 2800 - 02A0 2897 | - | Shadow Region 4 Channel Registers |
| 02A0 2898 - 02A0 29FF | - | Reserved |
| 02A0 2A00 - 02A0 2A97 | - | Shadow Region 5 Channel Registers |
| 02A0 2A98 - 02A0 2BFF | - | Reserved |
| 02A0 2C00 - 02A0 2C97 | - | Shadow Region 6 Channel Registers |
| 02A0 2C98 - 02A0 2DFF | - | Reserved |
| 02A0 2E00 - 02A0 2E97 | - | Shadow Region 7 Channel Registers |
| 02A0 2E98 - 02A0 2FFF | - | Reserved |

**Table 7-5. EDMA3 Parameter RAM(1)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 4000 - 02A0 401F | - | Parameter Set 0 |
| 02A0 4020 - 02A0 403F | - | Parameter Set 1 |
| 02A0 4040 - 02A0 405F | - | Parameter Set 2 |
| 02A0 4060 - 02A0 407F | - | Parameter Set 3 |
| 02A0 4080 - 02A0 409F | - | Parameter Set 4 |
| 02A0 40A0 - 02A0 40BF | - | Parameter Set 5 |
| 02A0 40C0 - 02A0 40DF | - | Parameter Set 6 |
| 02A0 40E0 - 02A0 40FF | - | Parameter Set 7 |
| 02A0 4100 - 02A0 411F | - | Parameter Set 8 |

(1) The C6454 device has 256 EDMA3 parameter sets total. Each parameter set can be used as a DMA entry, a QDMA entry, or a link entry.

**Table 7-5. EDMA3 Parameter RAM[(1)] (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A0 4120 - 02A0 413F | - | Parameter Set 9 |
| ... | | ... |
| 02A0 47E0 - 02A0 47FF | - | Parameter Set 63 |
| 02A0 4800 - 02A0 481F | - | Parameter Set 64 |
| 02A0 4820 - 02A0 483F | - | Parameter Set 65 |
| ... | | ... |
| 02A0 5FC0 - 02A0 5FDF | - | Parameter Set 254 |
| 02A0 5FE0 - 02A0 5FFF | - | Parameter Set 255 |

**Table 7-6. EDMA3 Transfer Controller 0 Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A2 0000 | PID | Peripheral Identification Register |
| 02A2 0004 | TCCFG | EDMA3TC Configuration Register |
| 02A2 0008 - 02A2 00FC | - | Reserved |
| 02A2 0100 | TCSTAT | EDMA3TC Channel Status Register |
| 02A2 0104 - 02A2 011C | - | Reserved |
| 02A2 0120 | ERRSTAT | Error Register |
| 02A2 0124 | ERREN | Error Enable Register |
| 02A2 0128 | ERRCLR | Error Clear Register |
| 02A2 012C | ERRDET | Error Details Register |
| 02A2 0130 | ERRCMD | Error Interrupt Command Register |
| 02A2 0134 - 02A2 013C | - | Reserved |
| 02A2 0140 | RDRATE | Read Rate Register |
| 02A2 0144 - 02A2 023C | - | Reserved |
| 02A2 0240 | SAOPT | Source Active Options Register |
| 02A2 0244 | SASRC | Source Active Source Address Register |
| 02A2 0248 | SACNT | Source Active Count Register |
| 02A2 024C | SADST | Source Active Destination Address Register |
| 02A2 0250 | SABIDX | Source Active Source B-Index Register |
| 02A2 0254 | SAMPPRXY | Source Active Memory Protection Proxy Register |
| 02A2 0258 | SACNTRLD | Source Active Count Reload Register |
| 02A2 025C | SASRCBREF | Source Active Source Address B-Reference Register |
| 02A2 0260 | SADSTBREF | Source Active Destination Address B-Reference Register |
| 02A2 0264 - 02A2 027C | - | Reserved |
| 02A2 0280 | DFCNTRLD | Destination FIFO Set Count Reload |
| 02A2 0284 | DFSRCBREF | Destination FIFO Set Destination Address B Reference Register |
| 02A2 0288 | DFDSTBREF | Destination FIFO Set Destination Address B Reference Register |
| 02A2 028C - 02A2 02FC | - | Reserved |
| 02A2 0300 | DFOPT0 | Destination FIFO Options Register 0 |
| 02A2 0304 | DFSRC0 | Destination FIFO Source Address Register 0 |
| 02A2 0308 | DFCNT0 | Destination FIFO Count Register 0 |
| 02A2 030C | DFDST0 | Destination FIFO Destination Address Register 0 |
| 02A2 0310 | DFBIDX0 | Destination FIFO BIDX Register 0 |
| 02A2 0314 | DFMPPRXY0 | Destination FIFO Memory Protection Proxy Register 0 |
| 02A2 0318 - 02A2 033C | - | Reserved |
| 02A2 0340 | DFOPT1 | Destination FIFO Options Register 1 |
| 02A2 0344 | DFSRC1 | Destination FIFO Source Address Register 1 |

**Table 7-6. EDMA3 Transfer Controller 0 Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A2 0348 | DFCNT1 | Destination FIFO Count Register 1 |
| 02A2 034C | DFDST1 | Destination FIFO Destination Address Register 1 |
| 02A2 0350 | DFBIDX1 | Destination FIFO BIDX Register 1 |
| 02A2 0354 | DFMPPRXY1 | Destination FIFO Memory Protection Proxy Register 1 |
| 02A2 0358 - 02A2 037C | - | Reserved |
| 02A2 0380 | DFOPT2 | Destination FIFO Options Register 2 |
| 02A2 0384 | DFSRC2 | Destination FIFO Source Address Register 2 |
| 02A2 0388 | DFCNT2 | Destination FIFO Count Register 2 |
| 02A2 038C | DFDST2 | Destination FIFO Destination Address Register 2 |
| 02A2 0390 | DFBIDX2 | Destination FIFO BIDX Register 2 |
| 02A2 0394 | DFMPPRXY2 | Destination FIFO Memory Protection Proxy Register 2 |
| 02A2 0398 - 02A2 03BC | - | Reserved |
| 02A2 03C0 | DFOPT3 | Destination FIFO Options Register 3 |
| 02A2 03C4 | DFSRC3 | Destination FIFO Source Address Register 3 |
| 02A2 03C8 | DFCNT3 | Destination FIFO Count Register 3 |
| 02A2 03CC | DFDST3 | Destination FIFO Destination Address Register 3 |
| 02A2 03D0 | DFBIDX3 | Destination FIFO BIDX Register 3 |
| 02A2 03D4 | DFMPPRXY3 | Destination FIFO Memory Protection Proxy Register 3 |
| 02A2 03D8 - 02A2 7FFF | - | Reserved |

**Table 7-7. EDMA3 Transfer Controller 1 Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A2 8000 | PID | Peripheral Identification Register |
| 02A2 8004 | TCCFG | EDMA3TC Configuration Register |
| 02A2 8008 - 02A2 80FC | - | Reserved |
| 02A2 8100 | TCSTAT | EDMA3TC Channel Status Register |
| 02A2 8104 - 02A2 811C | - | Reserved |
| 02A2 8120 | ERRSTAT | Error Register |
| 02A2 8124 | ERREN | Error Enable Register |
| 02A2 8128 | ERRCLR | Error Clear Register |
| 02A2 812C | ERRDET | Error Details Register |
| 02A2 8130 | ERRCMD | Error Interrupt Command Register |
| 02A2 8134 - 02A2 813C | - | Reserved |
| 02A2 8140 | RDRATE | Read Rate Register |
| 02A2 8144 - 02A2 823C | - | Reserved |
| 02A2 8240 | SAOPT | Source Active Options Register |
| 02A2 8244 | SASRC | Source Active Source Address Register |
| 02A2 8248 | SACNT | Source Active Count Register |
| 02A2 824C | SADST | Source Active Destination Address Register |
| 02A2 8250 | SABIDX | Source Active Source B-Index Register |
| 02A2 8254 | SAMPPRXY | Source Active Memory Protection Proxy Register |
| 02A2 8258 | SACNTRLD | Source Active Count Reload Register |
| 02A2 825C | SASRCBREF | Source Active Source Address B-Reference Register |
| 02A2 8260 | SADSTBREF | Source Active Destination Address B-Reference Register |
| 02A2 8264 - 02A2 827C | - | Reserved |
| 02A2 8280 | DFCNTRLD | Destination FIFO Set Count Reload |
| 02A2 8284 | DFSRCBREF | Destination FIFO Set Destination Address B Reference Register |

**Table 7-7. EDMA3 Transfer Controller 1 Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A2 8288 | DFDSTBREF | Destination FIFO Set Destination Address B Reference Register |
| 02A2 828C - 02A2 82FC | - | Reserved |
| 02A2 8300 | DFOPT0 | Destination FIFO Options Register 0 |
| 02A2 8304 | DFSRC0 | Destination FIFO Source Address Register 0 |
| 02A2 8308 | DFCNT0 | Destination FIFO Count Register 0 |
| 02A2 830C | DFDST0 | Destination FIFO Destination Address Register 0 |
| 02A2 8310 | DFBIDX0 | Destination FIFO BIDX Register 0 |
| 02A2 8314 | DFMPPRXY0 | Destination FIFO Memory Protection Proxy Register 0 |
| 02A2 8318 - 02A2 833C | - | Reserved |
| 02A2 8340 | DFOPT1 | Destination FIFO Options Register 1 |
| 02A2 8344 | DFSRC1 | Destination FIFO Source Address Register 1 |
| 02A2 8348 | DFCNT1 | Destination FIFO Count Register 1 |
| 02A2 834C | DFDST1 | Destination FIFO Destination Address Register 1 |
| 02A2 8350 | DFBIDX1 | Destination FIFO BIDX Register 1 |
| 02A2 8354 | DFMPPRXY1 | Destination FIFO Memory Protection Proxy Register 1 |
| 02A2 8358 - 02A2 837C | - | Reserved |
| 02A2 8380 | DFOPT2 | Destination FIFO Options Register 2 |
| 02A2 8384 | DFSRC2 | Destination FIFO Source Address Register 2 |
| 02A2 8388 | DFCNT2 | Destination FIFO Count Register 2 |
| 02A2 838C | DFDST2 | Destination FIFO Destination Address Register 2 |
| 02A2 8390 | DFBIDX2 | Destination FIFO BIDX Register 2 |
| 02A2 8394 | DFMPPRXY2 | Destination FIFO Memory Protection Proxy Register 2 |
| 02A2 8398 - 02A2 83BC | - | Reserved |
| 02A2 83C0 | DFOPT3 | Destination FIFO Options Register 3 |
| 02A2 83C4 | DFSRC3 | Destination FIFO Source Address Register 3 |
| 02A2 83C8 | DFCNT3 | Destination FIFO Count Register 3 |
| 02A2 83CC | DFDST3 | Destination FIFO Destination Address Register 3 |
| 02A2 83D0 | DFBIDX3 | Destination FIFO BIDX Register 3 |
| 02A2 83D4 | DFMPPRXY3 | Destination FIFO Memory Protection Proxy Register 3 |
| 02A2 83D8 - 02A2 FFFF | - | Reserved |

**Table 7-8. EDMA3 Transfer Controller 2 Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A3 0000 | PID | Peripheral Identification Register |
| 02A3 0004 | TCCFG | EDMA3TC Configuration Register |
| 02A3 0008 - 02A3 00FC | - | Reserved |
| 02A3 0100 | TCSTAT | EDMA3TC Channel Status Register |
| 02A3 0104 - 02A3 011C | - | Reserved |
| 02A3 0120 | ERRSTAT | Error Register |
| 02A3 0124 | ERREN | Error Enable Register |
| 02A3 0128 | ERRCLR | Error Clear Register |
| 02A3 012C | ERRDET | Error Details Register |
| 02A3 0130 | ERRCMD | Error Interrupt Command Register |
| 02A3 0134 - 02A3 013C | - | Reserved |
| 02A3 0140 | RDRATE | Read Rate Register |
| 02A3 0144 - 02A3 023C | - | Reserved |
| 02A3 0240 | SAOPT | Source Active Options Register |

**Table 7-8. EDMA3 Transfer Controller 2 Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A3 0244 | SASRC | Source Active Source Address Register |
| 02A3 0248 | SACNT | Source Active Count Register |
| 02A3 024C | SADST | Source Active Destination Address Register |
| 02A3 0250 | SABIDX | Source Active Source B-Index Register |
| 02A3 0254 | SAMPPRXY | Source Active Memory Protection Proxy Register |
| 02A3 0258 | SACNTRLD | Source Active Count Reload Register |
| 02A3 025C | SASRCBREF | Source Active Source Address B-Reference Register |
| 02A3 0260 | SADSTBREF | Source Active Destination Address B-Reference Register |
| 02A3 0264 - 02A3 027C | - | Reserved |
| 02A3 0280 | DFCNTRLD | Destination FIFO Set Count Reload |
| 02A3 0284 | DFSRCBREF | Destination FIFO Set Destination Address B Reference Register |
| 02A3 0288 | DFDSTBREF | Destination FIFO Set Destination Address B Reference Register |
| 02A3 028C - 02A3 02FC | - | Reserved |
| 02A3 0300 | DFOPT0 | Destination FIFO Options Register 0 |
| 02A3 0304 | DFSRC0 | Destination FIFO Source Address Register 0 |
| 02A3 0308 | DFCNT0 | Destination FIFO Count Register 0 |
| 02A3 030C | DFDST0 | Destination FIFO Destination Address Register 0 |
| 02A3 0310 | DFBIDX0 | Destination FIFO BIDX Register 0 |
| 02A3 0314 | DFMPPRXY0 | Destination FIFO Memory Protection Proxy Register 0 |
| 02A3 0318 - 02A3 033C | - | Reserved |
| 02A3 0340 | DFOPT1 | Destination FIFO Options Register 1 |
| 02A3 0344 | DFSRC1 | Destination FIFO Source Address Register 1 |
| 02A3 0348 | DFCNT1 | Destination FIFO Count Register 1 |
| 02A3 034C | DFDST1 | Destination FIFO Destination Address Register 1 |
| 02A3 0350 | DFBIDX1 | Destination FIFO BIDX Register 1 |
| 02A3 0354 | DFMPPRXY1 | Destination FIFO Memory Protection Proxy Register 1 |
| 02A3 0358 - 02A3 037C | - | Reserved |
| 02A3 0380 | DFOPT2 | Destination FIFO Options Register 2 |
| 02A3 0384 | DFSRC2 | Destination FIFO Source Address Register 2 |
| 02A3 0388 | DFCNT2 | Destination FIFO Count Register 2 |
| 02A3 038C | DFDST2 | Destination FIFO Destination Address Register 2 |
| 02A3 0390 | DFBIDX2 | Destination FIFO BIDX Register 2 |
| 02A3 0394 | DFMPPRXY2 | Destination FIFO Memory Protection Proxy Register 2 |
| 02A3 0398 - 02A3 03BC | - | Reserved |
| 02A3 03C0 | DFOPT3 | Destination FIFO Options Register 3 |
| 02A3 03C4 | DFSRC3 | Destination FIFO Source Address Register 3 |
| 02A3 03C8 | DFCNT3 | Destination FIFO Count Register 3 |
| 02A3 03CC | DFDST3 | Destination FIFO Destination Address Register 3 |
| 02A3 03D0 | DFBIDX3 | Destination FIFO BIDX Register 3 |
| 02A3 03D4 | DFMPPRXY3 | Destination FIFO Memory Protection Proxy Register 3 |
| 02A3 03D8 - 02A3 7FFF | - | Reserved |

**Table 7-9. EDMA3 Transfer Controller 3 Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A3 8000 | PID | Peripheral Identification Register |
| 02A3 8004 | TCCFG | EDMA3TC Configuration Register |
| 02A3 8008 - 02A3 80FC | - | Reserved |

**Table 7-9. EDMA3 Transfer Controller 3 Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A3 8100 | TCSTAT | EDMA3TC Channel Status Register |
| 02A3 8104 - 02A3 811C | - | Reserved |
| 02A3 8120 | ERRSTAT | Error Register |
| 02A3 8124 | ERREN | Error Enable Register |
| 02A3 8128 | ERRCLR | Error Clear Register |
| 02A3 812C | ERRDET | Error Details Register |
| 02A3 8130 | ERRCMD | Error Interrupt Command Register |
| 02A3 8134 - 02A3 813C | - | Reserved |
| 02A3 8140 | RDRATE | Read Rate Register |
| 02A3 8144 - 02A3 823C | - | Reserved |
| 02A3 8240 | SAOPT | Source Active Options Register |
| 02A3 8244 | SASRC | Source Active Source Address Register |
| 02A3 8248 | SACNT | Source Active Count Register |
| 02A3 824C | SADST | Source Active Destination Address Register |
| 02A3 8250 | SABIDX | Source Active Source B-Index Register |
| 02A3 8254 | SAMPPRXY | Source Active Memory Protection Proxy Register |
| 02A3 8258 | SACNTRLD | Source Active Count Reload Register |
| 02A3 825C | SASRCBREF | Source Active Source Address B-Reference Register |
| 02A3 8260 | SADSTBREF | Source Active Destination Address B-Reference Register |
| 02A3 8264 - 02A3 827C | - | Reserved |
| 02A3 8280 | DFCNTRLD | Destination FIFO Set Count Reload |
| 02A3 8284 | DFSRCBREF | Destination FIFO Set Destination Address B Reference Register |
| 02A3 8288 | DFDSTBREF | Destination FIFO Set Destination Address B Reference Register |
| 02A3 828C - 02A3 82FC | - | Reserved |
| 02A3 8300 | DFOPT0 | Destination FIFO Options Register 0 |
| 02A3 8304 | DFSRC0 | Destination FIFO Source Address Register 0 |
| 02A3 8308 | DFCNT0 | Destination FIFO Count Register 0 |
| 02A3 830C | DFDST0 | Destination FIFO Destination Address Register 0 |
| 02A3 8310 | DFBIDX0 | Destination FIFO BIDX Register 0 |
| 02A3 8314 | DFMPPRXY0 | Destination FIFO Memory Protection Proxy Register 0 |
| 02A3 8318 - 02A3 833C | - | Reserved |
| 02A3 8340 | DFOPT1 | Destination FIFO Options Register 1 |
| 02A3 8344 | DFSRC1 | Destination FIFO Source Address Register 1 |
| 02A3 8348 | DFCNT1 | Destination FIFO Count Register 1 |
| 02A3 834C | DFDST1 | Destination FIFO Destination Address Register 1 |
| 02A3 8350 | DFBIDX1 | Destination FIFO BIDX Register 1 |
| 02A3 8354 | DFMPPRXY1 | Destination FIFO Memory Protection Proxy Register 1 |
| 02A3 8358 - 02A3 837C | - | Reserved |
| 02A3 8380 | DFOPT2 | Destination FIFO Options Register 2 |
| 02A3 8384 | DFSRC2 | Destination FIFO Source Address Register 2 |
| 02A3 8388 | DFCNT2 | Destination FIFO Count Register 2 |
| 02A3 838C | DFDST2 | Destination FIFO Destination Address Register 2 |
| 02A3 8390 | DFBIDX2 | Destination FIFO BIDX Register 2 |
| 02A3 8394 | DFMPPRXY2 | Destination FIFO Memory Protection Proxy Register 2 |
| 02A3 8398 - 02A3 83BC | - | Reserved |
| 02A3 83C0 | DFOPT3 | Destination FIFO Options Register 3 |
| 02A3 83C4 | DFSRC3 | Destination FIFO Source Address Register 3 |

**Table 7-9. EDMA3 Transfer Controller 3 Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02A3 83C8 | DFCNT3 | Destination FIFO Count Register 3 |
| 02A3 83CC | DFDST3 | Destination FIFO Destination Address Register 3 |
| 02A3 83D0 | DFBIDX3 | Destination FIFO BIDX Register 3 |
| 02A3 83D4 | DFMPPRXY3 | Destination FIFO Memory Protection Proxy Register 3 |
| 02A3 83D8 - 02A3 FFFF | - | Reserved |

## 7.5 Interrupts

### 7.5.1 Interrupt Sources and Interrupt Controller

The CPU interrupts on the C6454 device are configured through the C64x+ Megamodule Interrupt Controller. The interrupt controller allows for up to 128 system events to be programmed to any of the twelve CPU interrupt inputs (CPUINT4 - CPUINT15), the CPU exception input (EXCEP), or the advanced emulation logic. The 128 system events consist of both internally-generated events (within the megamodule) and chip-level events. Table 7-10 shows the mapping of system events. For more information on the Interrupt Controller, see the *TMS320C64x+ Megamodule Reference Guide* (literature number SPRU871).

**Table 7-10. C6454 System Event Mapping**

| EVENT NUMBER | INTERRUPT EVENT | DESCRIPTION |
|---|---|---|
| 0(1) | EVT0 | Output of event combiner 0 in interrupt controller, for events 1 - 31. |
| 1(1) | EVT1 | Output of event combiner 1 in interrupt controller, for events 32 - 63. |
| 2(1) | EVT2 | Output of event combiner 2 in interrupt controller, for events 64 - 95. |
| 3(1) | EVT3 | Output of event combiner 3 in interrupt controller, for events 96 - 127. |
| 4 - 8 | Reserved | Reserved. These system events are not connected and, therefore, not used. |
| 9(1) | EMU_DTDMA | EMU interrupt for:<br>1. Host scan access<br>2. DTDMA transfer complete<br>3. AET interrupt |
| 10 | None | This system event is not connected and, therefore, not used. |
| 11(1) | EMU_RTDXRX | EMU real-time data exchange (RTDX) receive complete |
| 12(1) | EMU_RTDXTX | EMU RTDX transmit complete |
| 13(1) | IDMA0 | IDMA channel 0 interrupt |
| 14(1) | IDMA1 | IDMA channel 1 interrupt |
| 15 | DSPINT | HPI/PCI-to-DSP interrupt |
| 16 | I2CINT | I2C interrupt |
| 17 | MACINT | Ethernet MAC interrupt |
| 18 | AEASYNCERR | EMIFA error interrupt |
| 19 - 23 | Reserved | Reserved. This system event is not connected and, therefore, not used. |
| 24 | EDMA3CC_GINT | EDMA3 channel global completion interrupt |
| 25 - 39 | Reserved | Reserved. These system events are not connected and, therefore, not used. |
| 40 | RINT0 | McBSP0 receive interrupt |
| 41 | XINT0 | McBSP0 transmit interrupt |
| 42 | RINT1 | McBSP1 receive interrupt |
| 43 | XINT1 | McBSP1 transmit interrupt |
| 44 - 50 | Reserved | Reserved. These system events are not connected and, therefore, not used. |
| 51 | GPINT0 | GPIO interrupt |
| 52 | GPINT1 | GPIO interrupt |
| 53 | GPINT2 | GPIO interrupt |
| 54 | GPINT3 | GPIO interrupt |
| 55 | GPINT4 | GPIO interrupt |
| 56 | GPINT5 | GPIO interrupt |
| 57 | GPINT6 | GPIO interrupt |

(1) This system event is generated from within the C64x+ megamodule.

**Table 7-10. C6454 System Event Mapping (continued)**

| EVENT NUMBER | INTERRUPT EVENT | DESCRIPTION |
|---|---|---|
| 58 | GPINT7 | GPIO interrupt |
| 59 | GPINT8 | GPIO interrupt |
| 60 | GPINT9 | GPIO interrupt |
| 61 | GPINT10 | GPIO interrupt |
| 62 | GPINT11 | GPIO interrupt |
| 63 | GPINT12 | GPIO interrupt |
| 64 | GPINT13 | GPIO interrupt |
| 65 | GPINT14 | GPIO interrupt |
| 66 | GPINT15 | GPIO interrupt |
| 67 | TINTLO0 | Timer 0 lower counter interrupt |
| 68 | TINTHI0 | Timer 0 higher counter interrupt |
| 69 | TINTLO1 | Timer 1 lower counter interrupt |
| 70 | TINTHI1 | Timer 1 higher counter interrupt |
| 71 | EDMA3CC_INT0 | EDMA3CC completion interrupt - Mask0 |
| 72 | EDMA3CC_INT1 | EDMA3CC completion interrupt - Mask1 |
| 73 | EDMA3CC_INT2 | EDMA3CC completion interrupt - Mask2 |
| 74 | EDMA3CC_INT3 | EDMA3CC completion interrupt - Mask3 |
| 75 | EDMA3CC_INT4 | EDMA3CC completion interrupt - Mask4 |
| 76 | EDMA3CC_INT5 | EDMA3CC completion interrupt - Mask5 |
| 77 | EDMA3CC_INT6 | EDMA3CC completion interrupt - Mask6 |
| 78 | EDMA3CC_INT7 | EDMA3CC completion interrupt - Mask7 |
| 79 | EDMA3CC_ERRINT | EDMA3CC error interrupt |
| 80 | Reserved | Reserved. This system event is not connected and, therefore, not used. |
| 81 | EDMA3TC0_ERRINT | EDMA3TC0 error interrupt |
| 82 | EDMA3TC1_ERRINT | EDMA3TC1 error interrupt |
| 83 | EDMA3TC2_ERRINT | EDMA3TC2 error interrupt |
| 84 | EDMA3TC3_ERRINT | EDMA3TC3 error interrupt |
| 85 - 95 | Reserved | Reserved. These system events are not connected and, therefore, not used. |
| 96 (2) | INTERR | Interrupt Controller dropped CPU interrupt event |
| 97 (2) | EMC_IDMAERR | EMC invalid IDMA parameters |
| 98 - 99 | Reserved | Reserved. These system events are not connected and, therefore, not used. |
| 100 (2) | EFIINTA | EFI interrupt from side A |
| 101 (2) | EFIINTB | EFI interrupt from side B |
| 102 - 112 | Reserved | Reserved. These system events are not connected and, therefore, not used. |
| 113 (2) | L1P_ED1 | L1P single bit error detected during DMA read |
| 114 - 115 | Reserved | Reserved. These system events are not connected and, therefore, not used. |
| 116 (2) | L2_ED1 | L2 single bit error detected |
| 117 (2) | L2_ED2 | L2 two bit error detected |
| 118 (2) | PDC_INT | Powerdown sleep interrupt |
| 119 | Reserved | Reserved. This system event is not connected and, therefore, not used. |
| 120 (2) | L1P_CMPA | L1P CPU memory protection fault |
| 121 (2) | L1P_DMPA | L1P DMA memory protection fault |

(2) This system event is generated from within the C64x+ megamodule.

**Table 7-10. C6454 System Event Mapping (continued)**

| EVENT NUMBER | INTERRUPT EVENT | DESCRIPTION |
|---|---|---|
| 122[2] | L1D_CMPA | L1D CPU memory protection fault |
| 123[2] | L1D_DMPA | L1D DMA memory protection fault |
| 124[2] | L2_CMPA | L2 CPU memory protection fault |
| 125[2] | L2_DMPA | L2 DMA memory protection fault |
| 126[2] | IDMA_CMPA | IDMA CPU memory protection fault |
| 127[2] | IDMA_BUSERR | IDMA bus error interrupt |

### 7.5.2 External Interrupts Electrical Data/Timing

**Table 7-11. Timing Requirements for External Interrupts(1)**

(see Figure 7-6)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(NMIL)}$ | Width of the NMI interrupt pulse low | 6P | | ns |
| 2 | $t_{w(NMIH)}$ | Width of the NMI interrupt pulse high | 6P | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.

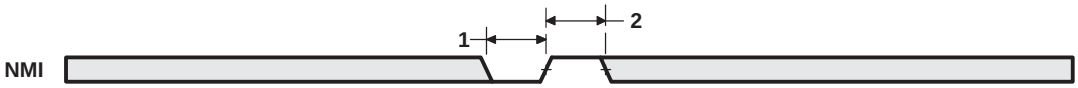



**Figure 7-6. NMI Interrupt Timing**

## 7.6 Reset Controller

The reset controller detects the different type of resets supported on the C6454 device and manages the distribution of those resets throughout the device.

The C6454 device has several types of resets: power-on reset, warm reset, system reset, and CPU reset. Table 7-12 explains further the types of reset, the reset initiator, and the effects of each reset on the chip. For more information on the effects of each reset on the PLL controllers and their clocks, see Section 7.6.7, *Reset Electrical Data/Timing*.

**Table 7-12. Reset Types**

| TYPE | INITIATOR | EFFECT(s) |
|---|---|---|
| Power-on Reset | $\overline{\text{POR}}$ pin | Resets the entire chip including the test and emulation logic. |
| Warm Reset | $\overline{\text{RESET}}$ pin | Resets everything except for the test and emulation logic and PLL2. Emulator stays alive during Warm Reset. |
| System Reset | Emulator | A system reset maintains memory contents and does not reset the test and emulation circuitry. The device configuration pins are also not re-latched and the state of the peripherals is also not affected. (1) |
| CPU Local Reset | HPI/PCI | CPU local reset. |

(1) On the C6454 device, peripherals can be in one of several states. These states are listed in Table 3-4.

### *7.6.1 Power-on Reset ($\overline{\text{POR}}$ Pin)*

Power-on Reset is initiated by the $\overline{\text{POR}}$ pin and is used to reset the entire chip, including the test and emulation logic. Power-on Reset is also referred to as a cold reset since the device usually goes through a power-up cycle. During power-up, the $\overline{\text{POR}}$ pin must be asserted (driven low) until the power supplies have reached their normal operating conditions. Note that a device power-up cycle is not required to initiate a Power-on Reset.

The following sequence must be followed during a Power-on Reset:

1. Wait for all power supplies to reach normal operating conditions while keeping the $\overline{\text{POR}}$ pin asserted (driven low).

   While $\overline{\text{POR}}$ is asserted, all pins will be set to high-impedance. After the $\overline{\text{POR}}$ pin is deasserted (driven high), all Z group pins, low group pins, and high group pins are set to their reset state and will remain at their reset state until the otherwise configured by their respective peripheral. All peripherals, except those selected for boot purposes, are disabled after a Power-on Reset and must be enabled through the Device State Control registers; for more details, see Section 3.3, *Peripheral Selection After Device Reset*.
2. Once all the power supplies are within valid operating conditions, the $\overline{\text{POR}}$ pin must remain asserted (low) for a minimum of 256 CLKIN2 cycles. The PLL1 controller input clock, CLKIN1, and the PCI input clock, PCLK, must also be valid during this time. PCLK is only needed if the PCI module is being used. If the DDR2 memory controller and the EMAC peripheral are not needed, CLKIN2 can be tied low and, in this case, the $\overline{\text{POR}}$ pin must remain asserted (low) for a minimum of 256 CLKIN1 cycles after all power supplies have reached valid operating conditions.

   Within the low period of the $\overline{\text{POR}}$ pin, the following happens:
   - The reset signals flow to the entire chip (including the test and emulation logic), resetting modules that use reset asynchronously.
   - The PLL1 controller clocks are started at the frequency of the system reference clock. The clocks are propagated throughout the chip to reset modules that use reset synchronously. By default, PLL1 is in reset and unlocked.
   - The PLL2 controller clocks are started at the frequency of the system reference clock. PLL2 is held in reset. Since the PLL2 controller always operates in PLL mode, the system reference clock and all the system clocks are invalid at this point.
   - The $\overline{\text{RESETSTAT}}$ pin stays asserted (low), indicating the device is in reset.
3. The $\overline{\text{POR}}$ pin may now be deasserted (driven high).

When the $\overline{\text{POR}}$ pin is deasserted, the configuration pin values are latched and the PLL controllers change their system clocks to their default divide-down values. PLL2 is taken out of reset and automatically starts its locking sequence. Other device initialization is also started.

4. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin is deasserted (driven high). By this time, PLL2 has already completed its locking sequence and is outputting a valid clock. The system clocks of both PLL controllers are allowed to finish their current cycles and then paused for 10 cycles of their respective system reference clocks. After the pause the system clocks are restarted at their default divide-by settings.
5. The device is now out of reset, device execution begins as dictated by the selected boot mode (see Section 2.4, *Boot Sequence*).

**NOTE**

To most of the device, reset is de-asserted only when the $\overline{\text{POR}}$ and $\overline{\text{RESET}}$ pins are both de-asserted (driven high). Therefore, in the sequence described above, if the $\overline{\text{RESET}}$ pin is held low past the low period of the $\overline{\text{POR}}$ pin, most of the device will remain in reset. The only exception being that PLL2 is taken out of reset as soon as $\overline{\text{POR}}$ is de-asserted (driven high), regardless of the state of the $\overline{\text{RESET}}$ pin. The $\overline{\text{RESET}}$ pin should not be tied together with the $\overline{\text{POR}}$ pin.

### 7.6.2 Warm Reset ($\overline{\text{RESET}}$ Pin)

A Warm Reset has the same effects as a Power-on Reset, except that in this case, the test and emulation logic and PLL2 are not reset.

The following sequence must be followed during a Warm Reset:

1. Hold the $\overline{\text{RESET}}$ pin low for a minimum of 24 CLKIN1 cycles. Within the minimum 24 CLKIN1 cycles.

   Within the low period of the $\overline{\text{RESET}}$ pin, the following happens:
   - The Z group pins, low group pins, and the high group pins are set to their reset state with one exception:

     The PCI pins are not affected by warm reset if the PCI module was enabled before $\overline{\text{RESET}}$ went low. In this case, PCI pins stay at whatever their value was before $\overline{\text{RESET}}$ went low.
   - The reset signals flow to the entire chip (excluding the test and emulation logic), resetting modules that use reset asynchronously.
   - The PLL1 controller is reset thereby switching back to bypass mode and resetting all its registers to their default values. PLL1 is placed in reset and loses lock. The PLL1 controller clocks start running at the frequency of the system reference clock. The clocks are propagated throughout the chip to reset modules that use reset synchronously.
   - The PLL2 controller is reset thereby resetting all its registers to their default values. The PLL2 controller clocks start running at the frequency of the system reference clock. PLL2 is not reset, therefore it remains locked.
   - The $\overline{\text{RESETSTAT}}$ pin becomes active (low), indicating the device is in reset.
2. The $\overline{\text{RESET}}$ pin may now be released (driven inactive high).

   When the $\overline{\text{RESET}}$ pin is released, the configuration pin values are latched and the PLL controllers immediately change their system clocks to their default divide-down values. Other device initialization is also started.
3. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin goes inactive (high). All system clocks are allowed to finish their current cycles and then paused for 10 cycles of their respective system reference clocks. After the pause the system clocks are restarted at their default divide-by settings.
4. The device is now out of reset, device execution begins as dictated by the selected boot mode (see Section 2.4, *Boot Sequence*).

**NOTE**

The $\overline{\text{POR}}$ pin should be held inactive (high) throughout the Warm Reset sequence. Otherwise, if $\overline{\text{POR}}$ is activated (brought low), the minimum $\overline{\text{POR}}$ pulse width must be met. The $\overline{\text{RESET}}$ pin should not be tied together with the $\overline{\text{POR}}$ pin.

### 7.6.3 System Reset

The emulator initiates a System Reset via the ICEPick module. This ICEPick-initiated reset is non-maskable. To invoke the maximum reset via the ICEPick module, the user can perform the following from the Code Composer Studio™ menu: Debug → Advanced Resets → System Reset.

The following memory contents are maintained during a System Reset:

- DDR2 Memory Controller: The DDR2 Memory Controller registers are ***not*** reset. In addition, the DDR2 SDRAM memory content is retained if the user places the DDR2 SDRAM in self-refresh mode before invoking the System Reset.
- EMIFA: The contents of the memory connected to the EMIFA are retained. The EMIFA registers are ***not*** reset.

Test, emulation, and clock logic are unaffected. The device configuration pins are also not re-latched and the state of the peripherals (see Table 3-4) is not affected.

During a System Reset, the following happens:

1. The System Reset is initiated by the emulator.

   During this time, the following happens:
   – The reset signals flow to the entire chip resetting all the modules on chip except the test and emulation logic.
   – The PLL controllers are ***not*** reset. Internal system clocks are unaffected. If PLL1/PLL2 were locked before the System Reset, they remain locked.
   – The $\overline{\text{RESETSTAT}}$ pin goes low to indicate an internal reset is being generated.
2. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin is deasserted (driven high). In addition, the PLL controllers pause their system clocks for about 10 cycles.

   At this point:
   – The state of the peripherals before the System Reset is not changed. For example, if McBSP0 was in the enabled state before System Reset, it will remain in the enabled state after System Reset.
   – The I/O pins are controlled as dictated by the DEVSTAT register.
   – The DDR2 Memory Controller and EMIFA registers retain their previous values. Only the DDR2 Memory Controller and EMIFA state machines are reset by the System Reset.
   – The PLL controllers are operating in the mode prior to System Reset. System clocks are unaffected.

The boot sequence is started after the system clocks are restarted. Since the configuration pins (including the BOOTMODE[3:0] pins) are not latched with a System Reset, the previous values, as shown in the DEVSTAT register, are used to select the boot mode.

### 7.6.4 CPU Reset

A CPU Reset is initiated by the HPI or PCI peripheral. This reset only affects the CPU. During a PCI-initiated CPU Reset, the PCI pins are set to their reset state. With the exception of the $\overline{\text{HRDY}}$/$\overline{\text{PIRDY}}$ pin, the PCI pins have a reset state of high-impedance; the $\overline{\text{HRDY}}$/$\overline{\text{PIRDY}}$ pin goes high during reset.

### 7.6.5 Reset Priority

If any of the above reset sources occur simultaneously, the PLLCTRL only processes the highest priority reset request. The rest request priorities are as follows (high to low):

- Power-on Reset

- Warm Reset
- System Reset
- CPU Reset

### 7.6.6 Reset Controller Register

The reset type status (RSTYPE) register (029A 00E4) is the only register for the reset controller. This register falls in the same memory range as the PLL1 controller registers [029A 0000 - 029A 01FF] (see Table 7-18).

#### 7.6.6.1 Reset Type Status Register Description

The rest type status (RSTYPE) register latches the cause of the last reset. If multiple reset sources occur simultaneously, this register latches the highest priority reset source. The reset type status register is shown in Figure 7-7 and described in Table 7-13.

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | SRST | Rsvd | WRST | POR |
| R-0 | | R-0 | R-0 | R-0 | R-0 |

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

**Figure 7-7. Reset Type Status Register (RSTYPE) [Hex Address: 029A 00E4]**

**Table 7-13. Reset Type Status Register (RSTYPE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:4 | Reserved | | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 3 | SRST | | System reset. |
| | | 0 | System Reset was not the last reset to occur. |
| | | 1 | System Reset was the last reset to occur. |
| 2 | Reserved | | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 1 | WRST | | Warm reset. |
| | | 0 | Warm Reset was not the last reset to occur. |
| | | 1 | Warm Reset was the last reset to occur. |
| 0 | POR | | Power-on reset. |
| | | 0 | Power-on Reset was not the last reset to occur. |
| | | 1 | Power-on Reset was the last reset to occur. |

### 7.6.7 Reset Electrical Data/Timing

**Table 7-14. Timing Requirements for Reset(1) (2) (3)**

(see Figure 7-8 and Figure 7-9)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 5 | $t_{w(POR)}$ | Pulse duration, $\overline{POR}$ low | 256D(4) | | ns |
| 6 | $t_{w(RESET)}$ | Pulse duration, $\overline{RESET}$ low | 24C | | ns |
| 7 | $t_{su(boot)}$ | Setup time, boot mode and configuration pins valid before $\overline{POR}$ high or $\overline{RESET}$ high(5) | 6P | | ns |
| 8 | $t_{h(boot)}$ | Hold time, boot mode and configuration pins valid after $\overline{POR}$ high or $\overline{RESET}$ high(5) | 6P | | ns |

(1) C = 1/CLKIN1 clock frequency in ns.
(2) D = 1/CLKIN2 clock frequency in ns.
(3) P = 1/CPU clock frequency in nanoseconds (ns). Note that after power-on reset and warm reset the CPU frequency is equal to the CLKIN1 frequency divided by three due to the PLL1 controller being reset (see Section 7.6, *Reset Controller*).
(4) If CLKIN2 is not used, $t_{w(POR)}$ must be measured in terms of CLKIN1 cycles; otherwise, use CLKIN2 cycles.
(5) AEA[19:0], ABA[1:0], and PCI_EN are the boot configuration pins during device reset. **Note:** If a configuration pin must be routed out from the device and 3-stated (not driven), the internal pullup/pulldown (IPU/IPD) resistor should not be relied upon; TI recommends the use of an external pullup/pulldown resistor. For more detailed information on pullup/pulldown resistors and situations where external pullup/pulldown resistors are required, see Section 3.7, *Pullup/Pulldown Resistors*.

**Table 7-15. Switching Characteristics Over Recommended Operating Conditions During Reset(1)**

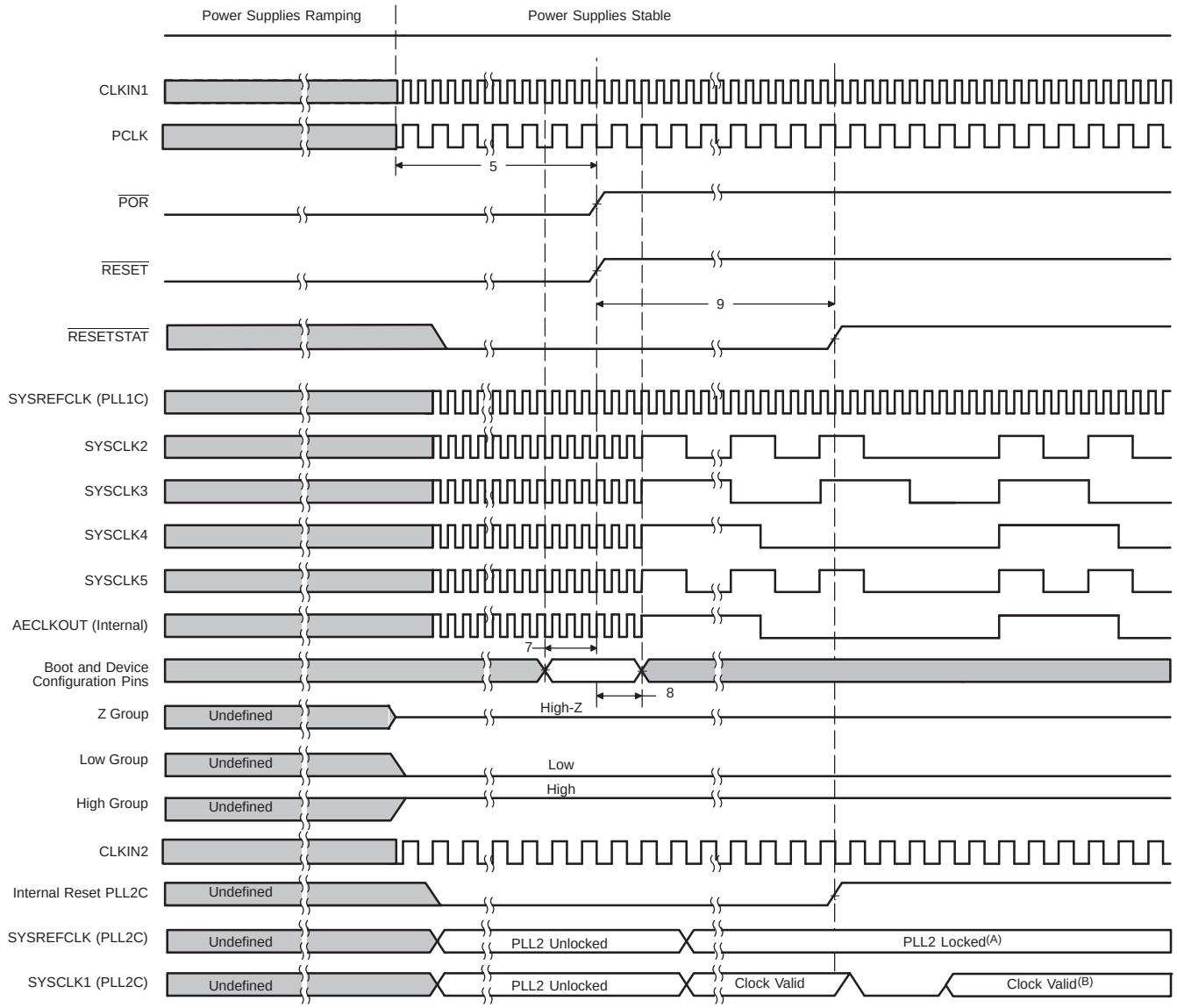
(see Figure 7-9)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 9 | $t_{d(PORH-RSTATH)}$ | Delay time, $\overline{POR}$ high AND $\overline{RESET}$ high to $\overline{RESETSTAT}$ high | | 15000C | ns |

(1) C = 1/CLKIN1 clock frequency in ns.

For Figure 7-8, note the following:

- Z group consists of: all I/O/Z and O/Z pins, except for Low and High group pins. Pins become high impedance as soon as their respective power supply has reached normal operating coditions. Pins remain in high impedance until configured otherwise by their respective peripheral.
- Low group consists of: MTXD0/RMTXD0, MTXD1/RMTXD1, MTXD2/RMTXD2, MTXD3/RMTXD3, MTXD4/RMTXD4, and MTXEN/RMTXEN. Pins become low as soon as their respective power supply has reached normal operating conditions. Pins remain low until configured otherwise by their respective peripheral.
- High group consists of: $\overline{AHOLD}$, ABUSREQ, and $\overline{HRDY}$/$\overline{PIRDY}$. Pins become high as soon as their respective power supply has reached normal operating conditions. Pins remain high until configured otherwise by their respective peripheral. The ABUSREQ pin remains high until the EMIFA is enabled through the PERCFG1 register. Once the EMIFA is enabled, the ABUSREQ pin is driven to its inactive state (driven low).
- All peripherals must be enable through software following a Power-on Reset; for more details, see Section 7.6.1, *Power-on Reset*.
- For power-supply sequence requirements, see Section 7.3.1, *Power-Supply Sequencing*.




A. SYSREFCLK of the PLL2 controller runs at CLKIN2 ×10.

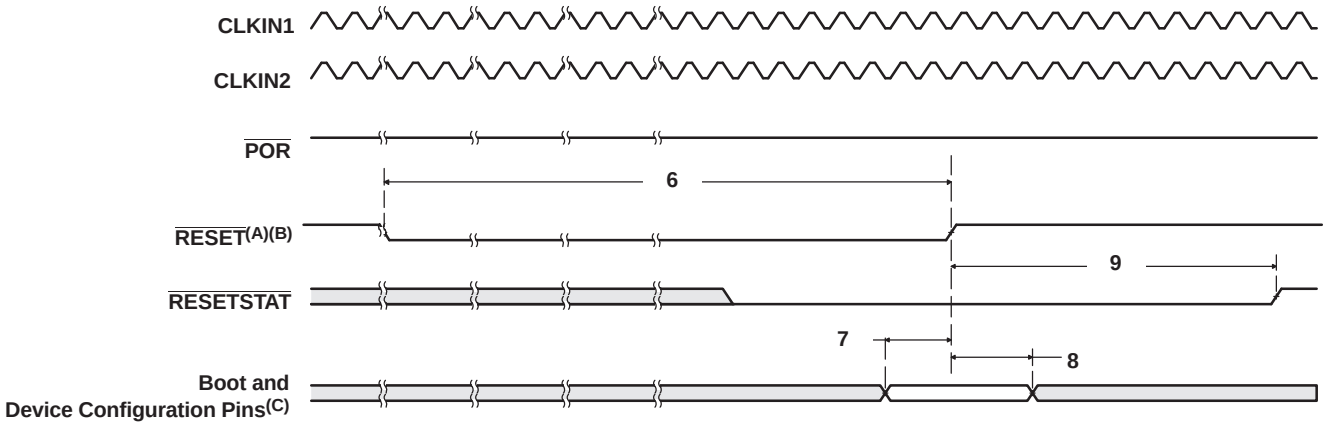
B. SYSCLK1 of PLL2 controller runs at SYSREFCLK/2 (default).

C. Power supplies, CLKIN1, CLKIN2 (if used), and PCLK (if used) must be stable before the start of $t_{w(POR)}$.

D. Do not tie the $\overline{\text{RESET}}$ and $\overline{\text{POR}}$ pins together.

E. The $\overline{\text{RESET}}$ pin can be brought high after the $\overline{\text{POR}}$ pin has been brought high. In this case, the $\overline{\text{RESET}}$ pin must be held low for a minimum of $t_{w(RESET)}$ after the $\overline{\text{POR}}$ pin has been brought high.

**Figure 7-8. Power-Up Timing**




A. RESET should only be used after device has been powered up. For more details on the use of the RESET pin, see Section 7.6, *Reset Controller*.

B. A reset signal is generated internally during a Warm Reset. This internal reset signal has the same effect as the RESET pin during a Warm Reset.

C. Boot and Device Configurations Inputs (during reset) include: AEA[19:0], ABA[1:0], and PCI_EN.

**Figure 7-9. Warm Reset Timing**

## 7.7 PLL1 and PLL1 Controller

The primary PLL controller generates the input clock to the C64x+ megamodule (including the CPU) as well as most of the system peripherals such as the multichannel buffered serial ports (McBSPs) and the external memory interface (EMIF).

As shown in Figure 7-10, the PLL1 controller features a software-programmable PLL multiplier controller (PLLM) and five dividers (PREDIV, D2, D3, D4, and D5). The PLL1 controller uses the device input clock CLKIN1 to generate a system reference clock (SYSREFCLK) and four system clocks (SYSCLK2, SYSCLK3, SYSCLK4, and SYSCLK5).

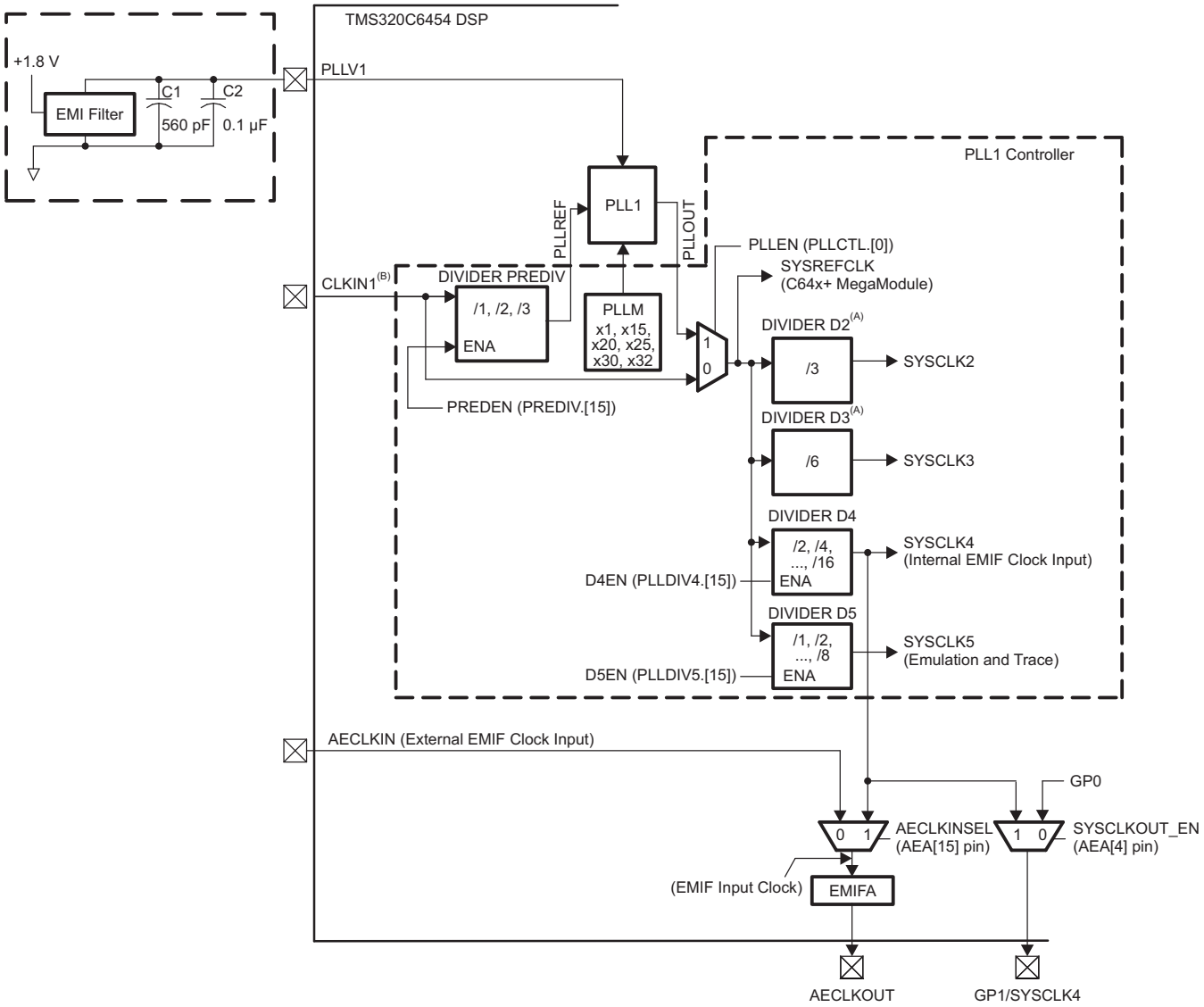
PLL1 power is supplied externally via the PLL1 power-supply pin (PLLV1). An external EMI filter circuit must be added to PLLV1, as shown in Figure 7-10. The 1.8-V supply of the EMI filter must be from the same 1.8-V power plane supplying the I/O power-supply pin, $DV_{DD18}$. TI requires EMI filter manufacturer Murata, part number NFM18CC222R1C3 or NFM18CC223R1C3.

All PLL external components (C1, C2, and the EMI Filter) must be placed as close to the C64x+ DSP device as possible. For the best performance, TI recommends that all the PLL external components be on a single side of the board without jumpers, switches, or components other than the ones shown. For reduced PLL jitter, maximize the spacing between switching signals and the PLL external components (C1, C2, and the EMI Filter).

The minimum CLKIN1 rise and fall times should also be observed. For the input clock timing requirements, see Section 7.7.4, *PLL1 Controller Input and Output Clock Electrical Data/Timing*.

**CAUTION**

The PLL controller module as described in the *TMS320C645x DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (literature number SPRUE56) includes a superset of features, some of which are not supported on the C6454 DSP. The following sections describe the features that are supported; it should be assumed that any feature not included in these sections is not supported by the C6454 DSP.




A. DIVIDER D2 and DIVIDER D3 are always enabled.
B. CLKIN1 is a 3.3-V signal.

**Figure 7-10. PLL1 and PLL1 Controller**

## *7.7.1 PLL1 Controller Device-Specific Information*

### 7.7.1.1 Internal Clocks and Maximum Operating Frequencies

As shown in Figure 7-10, the PLL1 controller generates several internal clocks including the system reference clock (SYSREFCLK), and the system clocks (SYSCLK2/3/4/5). The high-frequency clock signal SYSREFCLK is directly used to clock the C64x+ megamodule (including the CPU) and also serves as a reference clock for the rest of the DSP system.

Dividers D2, D3, D4, and D5 divide the high-frequency clock SYSREFCLK to generate SYSCLK2, SYSCLK3, SYSCLK4, and SYSCLK5, respectively. The system clocks are used to clock different portions of the DSP:

- SYSCLK2 is used to clock the switched central resources (SCRs), EDMA3, as well as the data bus interfaces of the EMIFA and DDR2 Memory Controller.
- SYSCLK3 clocks the PCI, HPI, McBSP, GPIO, TIMER, and I2C peripherals, as well as the configuration bus of the PLL2 Controller.

- SYSCLK4 is used as the internal clock for the EMIFA. It is also used to clock other logic within the DSP.
- SYSCLK5 clocks the emulation and trace logic of the DSP.

The divider ratio bits of dividers D2 and D3 are fixed at ÷3 and ÷6, respectively. The divider ratio bits of dividers D4 and D5 are programmable through the PLL controller divider registers PLLDIV4 and PLLDIV5, respectively.

The PLL multiplier controller (PLLM) and the dividers (D4 and D5) must be programmed after reset. **There is no hardware CLKMODE selection on the C6454 device.**

Since the divider ratio bits for dividers D2 and D3 are fixed, the frequency of SYSCLK2 and SYSCLK3 is tied to the frequency of SYSREFCLK. However, the frequency of SYSCLK4 and SYSCLK5 depends on the configuration of dividers D4 and D5. For example, with PLLM in the PLL1 multiply control register set to 10011b (x20 mode) and a 50-MHz CLKIN1 input, the PLL output PLLOUT is set to 1000 MHz and SYSCLK2 and SYSCLK3 run at 333 MHz and 166 MHz, respectively. Divider D4 can be programmed through the PLLDIV4 register to divide SYSREFCLK by 10 such that SYSCLK4 and, hence, the EMIF internal clock, runs at 120 MHz.

All hosts (HPI, PCI, etc.) must hold off accesses to the DSP while the frequency of its internal clocks is changing. A mechanism must be in place such that the DSP notifies the host when the PLL configuration has completed.

Note that there is a minimum and maximum operating frequency for PLLREF, PLLOUT, SYSCLK4, and SYSCLK5. The PLL1 Controller must not be configured to exceed any of these constraints (certain combinations of external clock input, internal dividers, and PLL multiply ratios might not be supported). For the PLL clocks input and output frequency ranges, see Table 7-16.

**Table 7-16. PLL1 Clock Frequency Ranges**

| CLOCK SIGNAL | MIN | MAX | UNIT |
|---|---|---|---|
| CLKIN1 | | 66.6 | MHz |
| PLLREF (PLLEN = 1)(1) | 33.3 | 66.6 | MHz |
| PLLOUT(1) | 400 | 1000 | MHz |
| SYSCLK4 | 25 | 166 | MHz |
| SYSCLK5 | | 333 | MHz |

(1) Only applies when the PLL1 Controller is set to PLL mode (PLLEN = 1 in the PLLCTL register).

### 7.7.1.2 PLL1 Controller Operating Modes

The PLL1 controller has two modes of operation: bypass mode and PLL mode. The mode of operation is determined by the PLLEN bit of the PLL control register (PLLCTL). In PLL mode, SYSREFCLK is generated from the device input clock CLKIN1 using the divider PREDIV and the PLL multiplier PLLM. In bypass mode, CLKIN1 is fed directly to SYSREFCLK.

All hosts (HPI, PCI, etc.) must hold off accesses to the DSP while the frequency of its internal clocks is changing. A mechanism must be in place such that the DSP notifies the host when the PLL configuration has completed.

### 7.7.1.3 PLL1 Stabilization, Lock, and Reset Times

The PLL stabilization time is the amount of time that must be allotted for the internal PLL regulators to become stable after device powerup. The PLL should not be operated until this stabilization time has expired.

The PLL reset time is the amount of wait time needed when resetting the PLL (writing PLLRST = 1), in order for the PLL to properly reset, before bringing the PLL out of reset (writing PLLRST = 0). For the PLL1 reset time value, see Table 7-17.

The PLL lock time is the amount of time needed from when the PLL is taken out of reset (PLLRST = 1 with PLLEN = 0) to when to when the PLL controller can be switched to PLL mode (PLLEN = 1). The PLL1 lock time is given in Table 7-17.

**Table 7-17. PLL1 Stabilization, Lock, and Reset Times**

| | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|
| PLL stabilization time | 150 | | | µs |
| PLL lock time | | | 2000*C (1) | ns |
| PLL reset time | 128*C (1) | | | ns |

(1) C = CLKIN1 cycle time in ns. For example, when CLKIN1 frequency is 50 MHz, use C = 20 ns.

### 7.7.2 PLL1 Controller Memory Map

The memory map of the PLL1 controller is shown in Table 7-18. Note that only registers documented here are accessible on the TMS320C6454 device. Other addresses in the PLL1 controller memory map should not be modified.

**Table 7-18. PLL1 Controller Registers (Including Reset Controller)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 029A 0000 - 029A 00E3 | - | Reserved |
| 029A 00E4 | RSTYPE | Reset Type Status Register (Reset Controller) |
| 029A 00E8 - 029A 00FF | - | Reserved |
| 029A 0100 | PLLCTL | PLL Control Register |
| 029A 0104 | - | Reserved |
| 029A 0108 | - | Reserved |
| 029A 010C | - | Reserved |
| 029A 0110 | PLLM | PLL Multiplier Control Register |
| 029A 0114 | PREDIV | PLL Pre-Divider Control Register |
| 029A 0118 | - | Reserved |
| 029A 011C | - | Reserved |
| 029A 0120 | - | Reserved |
| 029A 0124 | - | Reserved |
| 029A 0128 | - | Reserved |
| 029A 012C | - | Reserved |
| 029A 0130 | - | Reserved |
| 029A 0134 | - | Reserved |
| 029A 0138 | PLLCMD | PLL Controller Command Register |
| 029A 013C | PLLSTAT | PLL Controller Status Register |
| 029A 0140 | ALNCTL | PLL Controller Clock Align Control Register |
| 029A 0144 | DCHANGE | PLLDIV Ratio Change Status Register |
| 029A 0148 | - | Reserved |
| 029A 014C | - | Reserved |
| 029A 0150 | SYSTAT | SYSCLK Status Register |
| 029A 0154 | - | Reserved |
| 029A 0158 | - | Reserved |
| 029A 015C | - | Reserved |
| 029A 0160 | PLLDIV4 | PLL Controller Divider 4 Register |
| 029A 0164 | PLLDIV5 | PLL Controller Divider 5 Register |
| 029A 0168 - 029B FFFF | - | Reserved |

### 7.7.3 PLL1 Controller Register Descriptions

This section provides a description of the PLL1 controller registers. For details on the operation of the PLL controller module, see the *TMS320C645x DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (literature number SPRUE56).

**NOTE:** The PLL1 controller registers can only be accessed using the CPU or the emulator.

Not all of the registers documented in the *TMS320C645x DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (literature number SPRUE56) are supported on the TMS320C6454 DSP. Only those registers documented in this section are supported. Furthermore, only the bits within the registers described here are supported. You should not write to any reserved memory location or change the value of reserved bits.

#### 7.7.3.1 PLL1 Control Register

The PLL control register (PLLCTL) is shown in Figure 7-11 and described in Table 7-19.

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | Rsvd | Rsvd | Reserved | | PLLRST | Rsvd | PLL PWRDN | PLLEN |
| R-0 | | R/W-0 | R-1 | R/W-0 | | R/W-1 | R-0 | R/W-0 | R/W-0 |

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

**Figure 7-11. PLL1 Control Register (PLLCTL) [Hex Address: 029A 0100]**

**Table 7-19. PLL1 Control Register (PLLCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:8 | Reserved | | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 7 | Reserved | | Reserved. Writes to this register must keep this bit as 0. |
| 6 | Reserved | | Reserved. The reserved bit location is always read as 1. A value written to this field has no effect. |
| 5:4 | Reserved | | Reserved. Writes to this register must keep this bit as 0. |
| 3 | PLLRST | | PLL reset bit |
| | | 0 | PLL reset is released |
| | | 1 | PLL reset is asserted |
| 2 | Reserved | | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 1 | PLLPWRDN | | PLL power-down mode select bit |
| | | 0 | PLL is operational |
| | | 1 | PLL is placed in power-down state, i.e., all analog circuitry in the PLL is turned-off |
| 0 | PLLEN | | PLL enable bit |
| | | 0 | Bypass mode. Divider PREDIV and PLL are bypassed. All the system clocks (SYSCLK*n*) are divided down directly from input reference clock. |
| | | 1 | PLL mode. Divider PREDIV and PLL are not bypassed. PLL output path is enabled. All the system clocks (SYSCLK*n*) are divided down from PLL output. |

### 7.7.3.2 PLL Multiplier Control Register

The PLL multiplier control register (PLLM) is shown in Figure 7-12 and described in Table 7-20. The PLLM register defines the input reference clock frequency multiplier in conjunction with the PLL divider ratio bits (RATIO) in the PLL controller pre-divider register (PREDIV).

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | PLLM | |
| R-0 | | R/W-0h | |

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

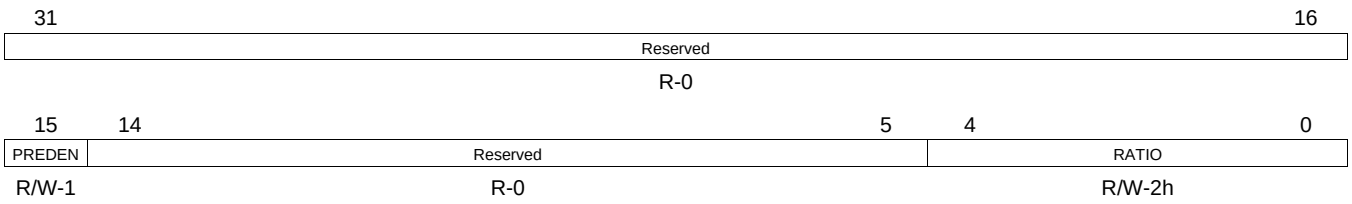
**Figure 7-12. PLL Multiplier Control Register (PLLM) [Hex Address: 029A 0110]**

**Table 7-20. PLL Multiplier Control Register (PLLM) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:5 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4:0 | PLLM | | PLL multiplier bits. Defines the frequency multiplier of the input reference clock in conjunction with the PLL divider ratio bits (RATIO) in PREDIV. |
| | | 0h | x1 multiplier rate |
| | | Eh | x15 multiplier rate |
| | | 13h | x20 multiplier rate |
| | | 18h | x25 multiplier rate |
| | | 1Dh | x30 multiplier rate |
| | | 1Fh | x32 multiplier rate |

### 7.7.3.3 PLL Pre-Divider Control Register

The PLL pre-divider control register (PREDIV) is shown in Figure 7-13 and described in Table 7-21.




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

**Figure 7-13. PLL Pre-Divider Control Register (PREDIV) [Hex Address: 029A 0114]**

**Table 7-21. PLL Pre-Divider Control Register (PREDIV) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:16 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 15 | PREDEN | | Pre-divider enable bit. |
| | | 0 | Pre-divider is disabled. No clock output. |
| | | 1 | Pre-divider is enabled. |
| 14:5 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4:0 | RATIO | 0-1Fh | Divider ratio bits. |
| | | 0 | ÷1. Divide frequency by 1. |
| | | 1h | ÷2. Divide frequency by 2. |
| | | 2h | ÷3. Divide frequency by 3. |
| | | 3h-1Fh | Reserved, do not use. |

### 7.7.3.4 PLL Controller Divider 4 Register

The PLL controller divider 4 register (PLLDIV4) is shown in Figure 7-14 and described in Table 7-22.

Besides being used as the EMIFA internal clock, SYSCLK4 is also used in other parts of the system. Disabling this clock will cause unpredictable system behavior. Therefore, the PLLDIV4 register should never be used to disable SYSCLK4.

| 31 | | 16 |
|---|---|---|
| Reserved | | |
| R-0 | | |

| 15 | 14 | 5 | 4 | 0 |
|---|---|---|---|---|
| D4EN | Reserved | | RATIO | |
| R/W-1 | R-0 | | R/W-3 | |

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

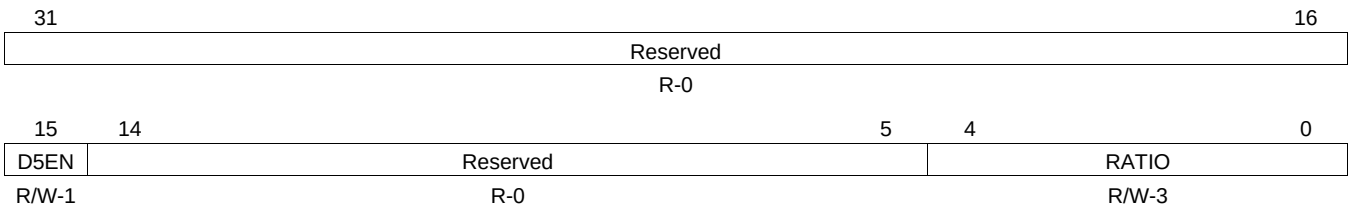
**Figure 7-14. PLL Controller Divider 4 Register (PLLDIV4) [Hex Address: 029A 0160]**

**Table 7-22. PLL Controller Divider 4 Register (PLLDIV4) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:16 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 15 | D4EN | | Divider 4 enable bit. |
| | | 0 | Divider 4 is disabled. No clock output. |
| | | 1 | Divider 4 is enabled. |
| 14:5 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4:0 | RATIO | 0-1Fh | Divider ratio bits. |
| | | 0 | ÷2. Divide frequency by 2. |
| | | 1h | ÷4. Divide frequency by 4. |
| | | 2h | ÷6. Divide frequency by 6. |
| | | 3h | ÷8. Divide frequency by 8. |
| | | 4h-7h | ÷10 to ÷16. Divide frequency by 10 to divide frequency by 16. |
| | | 8h-1Fh | Reserved, do not use. |

### 7.7.3.5 PLL Controller Divider 5 Register

The PLL controller divider 5 register (PLLDIV5) is shown in Figure 7-15 and described in Table 7-23.




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

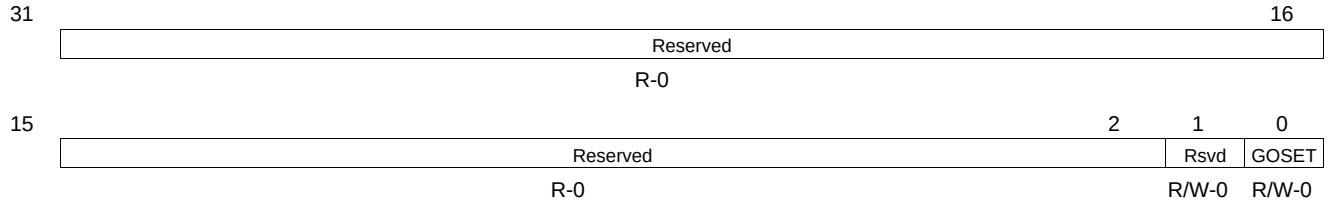
**Figure 7-15. PLL Controller Divider 5 Register (PLLDIV5) [Hex Address: 029A 0164]**

**Table 7-23. PLL Controller Divider 5 Register (PLLDIV5) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:16 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 15 | D5EN | | Divider 5 enable bit. |
| | | 0 | Divider 5 is disabled. No clock output. |
| | | 1 | Divider 5 is enabled. |
| 14:5 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4:0 | RATIO | 0-1Fh | Divider ratio bits. |
| | | 0 | ÷1. Divide frequency by 1. |
| | | 1h | ÷2. Divide frequency by 2. |
| | | 2h | ÷3. Divide frequency by 3. |
| | | 3h | ÷4. Divide frequency by 4. |
| | | 4h-7h | ÷5 to ÷8. Divide frequency by 5 to divide frequency by 8. |
| | | 8h-1Fh | Reserved, do not use. |

### 7.7.3.6 PLL Controller Command Register

The PLL controller command register (PLLCMD) contains the command bit for GO operation. PLLCMD is shown in Figure 7-16 and described in Table 7-24.




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

**Figure 7-16. PLL Controller Command Register (PLLCMD) [Hex Address: 029A 0138]**

**Table 7-24. PLL Controller Command Register (PLLCMD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:2 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 1 | Reserved | | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 0 | GOSET | | GO operation command for SYSCLK rate change and phase alignment. Before setting this bit to 1 to initiate a GO operation, check the GOSTAT bit in the PLLSTAT register to ensure all previous GO operations have completed. |
| | | 0 | No effect. Write of 0 clears bit to 0. |
| | | 1 | Initiates GO operation. Write of 1 initiates GO operation. Once set, GOSET remains set but further writes of 1 can initiate the GO operation. |

### 7.7.3.7 PLL Controller Status Register

The PLL controller status register (PLLSTAT) shows the PLL controller status. PLLSTAT is shown in Figure 7-17 and described in Table 7-25.

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | 1 | 0 |
| Reserved | | | | | | | | | | | | | | | GOSTAT |
| R-0 | | | | | | | | | | | | | | | R-0 |

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

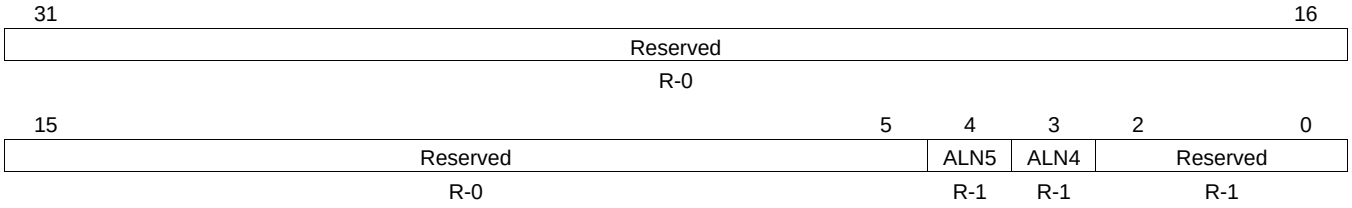
**Figure 7-17. PLL Controller Status Register (PLLSTAT) [Hex Address: 029A 013C]**

**Table 7-25. PLL Controller Status Register (PLLSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 0 | GOSTAT | | GO operation status. |
| | | 0 | GO operation is not in progress. SYSCLK divide ratios are not being changed. |
| | | 1 | GO operation is in progress. SYSCLK divide ratios are being changed. |

### 7.7.3.8 PLL Controller Clock Align Control Register

The PLL controller clock align control register (ALNCTL) is shown in Figure 7-18 and described in Table 7-26.




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

**Figure 7-18. PLL Controller Clock Align Control Register (ALNCTL) [Hex Address: 029A 0140]**

**Table 7-26. PLL Controller Clock Align Control Register (ALNCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:5 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4:3 | ALN*n* | | SYSCLK*n* alignment. Do not change the default values of these fields. |
| | | 0 | Do not align SYSCLK*n* to other SYSCLKs during GO operation. If SYS*n* in DCHANGE is set to 1, SYSCLK*n* switches to the new ratio immediately after the GOSET bit in PLLCMD is set. |
| | | 1 | Align SYSCLK*n* to other SYSCLKs selected in ALNCTL when the GOSET bit in PLLCMD is set. The SYSCLK*n* ratio is set to the ratio programmed in the RATIO bit in PLLDIV*n*. |
| 2:0 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |

### 7.7.3.9 PLLDIV Ratio Change Status Register

Whenever a different ratio is written to the PLLDIV*n* registers, the PLLCTRL flags the change in the PLLDIV ratio change status registers (DCHANGE). During the GO operation, the PLL controller will only change the divide ratio of the SYSCLKs with the bit set in DCHANGE. Note that changed clocks will be automatically aligned to other clocks. The PLLDIV divider ratio change status register is shown in Figure 7-19 and described in Table 7-27.

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|
| Reserved | | SYS5 | SYS4 | Reserved | |
| R-0 | | R-0 | R-0 | R-0 | |

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

**Figure 7-19. PLLDIV Divider Ratio Change Status Register (DCHANGE) [Hex Address: 029A 0144]**

**Table 7-27. PLLDIV Divider Ratio Change Status Register (DCHANGE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:5 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4 | SYS5 | | Identifies when the SYSCLK5 divide ratio has been modified. |
| | | 0 | SYSCLK5 ratio has not been modified. When GOSET is set, SYSCLK5 will not be affected. |
| | | 1 | SYSCLK5 ratio has been modified. When GOSET is set, SYSCLK5 will change to the new ratio. |
| 3 | SYS4 | | Identifies when the SYSCLK4 divide ratio has been modified. |
| | | 0 | SYSCLK4 ratio has not been modified. When GOSET is set, SYSCLK4 will not be affected. |
| | | 1 | SYSCLK4 ratio has been modified. When GOSET is set, SYSCLK4 will change to the new ratio. |
| 2:0 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |

### 7.7.3.10 SYSCLK Status Register

The SYSCLK status register (SYSTAT) shows the status of the system clocks (SYSCLK*n*). SYSTAT is shown in Figure 7-20 and described in Table 7-28.

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | SYS5ON | SYS4ON | SYS3ON | SYS2ON | Reserved |
| R-0 | | R-1 | R-1 | R-1 | R-1 | R-1 |

**LEGEND:** R = Read only; -*n* = value after reset

**Figure 7-20. SYSCLK Status Register (SYSTAT) [Hex Address: 029A 0150]**

**Table 7-28. SYSCLK Status Register (SYSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:4 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4:1 | SYS*n*ON | | SYSCLK*n* on status. |
| | | 0 | SYSCLK*n* is gated. |
| | | 1 | SYSCLK*n* is on. |
| 0 | Reserved | 1 | Reserved. The reserved bit location is always read as 1. A value written to this field has no effect. |

### 7.7.4 PLL1 Controller Input and Output Clock Electrical Data/Timing

**Table 7-29. Timing Requirements for CLKIN1 Devices(1) (2) (3)**

(see Figure 7-21)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | PLL MODES x1 (Bypass), x15, x20, x25, x30, x32 | | |
| | | | MIN | MAX | |
| 1 | $t_{c(CLKIN1)}$ | Cycle time, CLKIN1(4) | 15 | 30.3 | ns |
| 2 | $t_{w(CLKIN1H)}$ | Pulse duration, CLKIN1 high | 0.4C | | ns |
| 3 | $t_{w(CLKIN1L)}$ | Pulse duration, CLKIN1 low | 0.4C | | ns |
| 4 | $t_{t(CLKIN1)}$ | Transition time, CLKIN1 | | 1.2 | ns |
| 5 | $t_{J(CLKIN1)}$ | Period jitter (peak-to-peak), CLKIN1 | | 100 | ps |

(1) The reference points for the rise and fall transitions are measured at 3.3 V $V_{IL}$ MAX and $V_{IH}$ MIN.
(2) For more details on the PLL multiplier factors (x1 [BYPASS], x 15, x20, x25, x30, x32), see Section 7.7.1.2, *PLL1 Controller Operating Modes*.
(3) C = **CLKIN1** cycle time in ns. For example, when CLKIN1 frequency is 50 MHz, use C = 20 ns.
(4) The PLL1 multiplier factors (x1 [BYPASS], x 15, x20, x25, x30, x32) further limit the MIN and MAX values for $t_{c(CLKIN1)}$. For more detailed information on these limitations, see Section 7.7.1.1, *Internal Clocks and Maximum Operating Frequencies*.

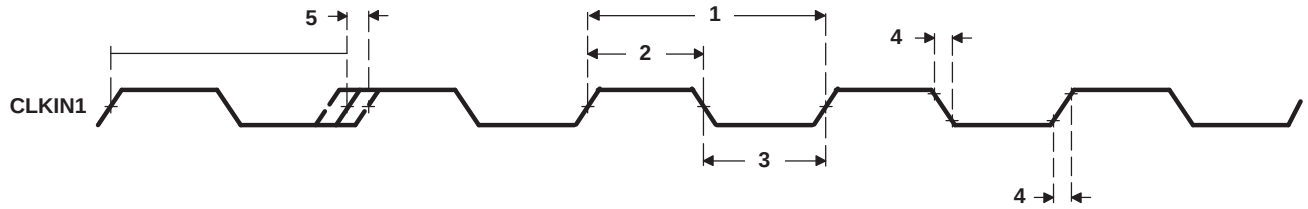



**Figure 7-21. CLKIN1 Timing**

**Table 7-30. Switching Characteristics Over Recommended Operating Conditions for SYSCLK4 [CPU/8 - CPU/12](1) (2)**

(see Figure 7-22)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 2 | $t_{w(CKO3H)}$ | Pulse duration, SYSCLK4 high | 4P - 0.7 | 6P + 0.7 | ns |
| 3 | $t_{w(CKO3L)}$ | Pulse duration, SYSCLK4 low | 4P - 0.7 | 6P + 0.7 | ns |
| 4 | $t_{t(CKO3)}$ | Transition time, SYSCLK4 | | 1 | ns |

(1) The reference points for the rise and fall transitions are measured at 3.3 V $V_{OL}$ MAX and $V_{OH}$ MIN.
(2) P = 1/CPU clock frequency in nanoseconds (ns)

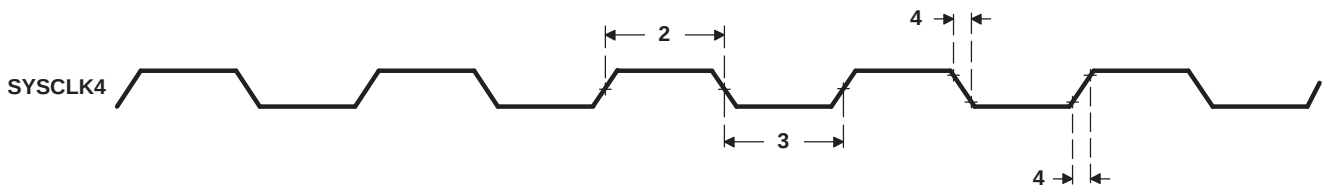



**Figure 7-22. SYSCLK4 Timing**

## 7.8 PLL2 and PLL2 Controller

The secondary PLL controller generates interface clocks for the Ethernet media access controller (EMAC) and the DDR2 memory controller.

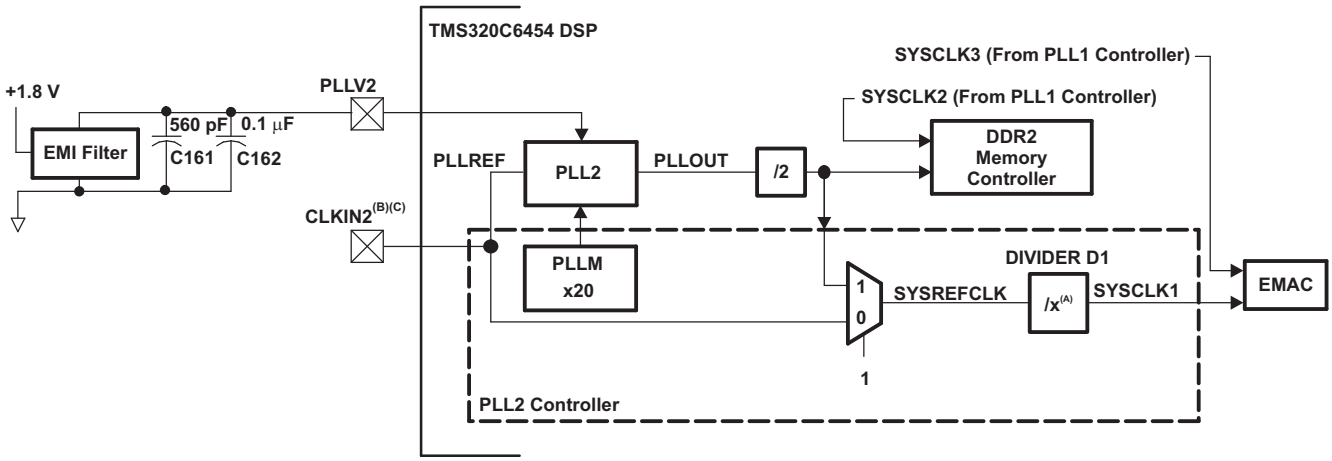
As shown in Figure 7-23, the PLL2 controller features a PLL multiplier controller and one divider (D1). The PLL multiplier is fixed to a x20 multiplier rate and the divider D1 can be programmed to a ÷2 or ÷5 mode.

PLL2 power is supplied externally via the PLL2 power supply (PLLV2). An external PLL filter circuit must be added to PLLV2 as shown in Figure 7-23. The 1.8-V supply for the EMI filter must be from the same 1.8-V power plane supplying the I/O power-supply pin, $DV_{DD18}$. TI requires EMI filter manufacturer Murata, part number NFM18CC222R1C3 or NFM18CC223R1C3.

All PLL external components (C161, C162, and the EMI Filter) should be placed as close to the C64x+ DSP device as possible. For the best performance, TI requires that all the PLL external components be on a single side of the board without jumpers, switches, or components other than the ones shown. For reduced PLL jitter, maximize the spacing between switching signals and the PLL external components (C161, C162, and the EMI Filter). The minimum CLKIN2 rise and fall times should also be observed. For the input clock timing requirements, see Section 7.8.4, *PLL2 Controller Input Clock Electrical Data/Timing*.

**CAUTION**

The PLL controller module as described in the *TMS320C645x DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (literature number SPRUE56) includes a superset of features, some of which are not supported on the C6454 DSP. The following sections describe the features that are supported; it should be assumed that any feature not included in these sections is not supported by the C6454 DSP.




A. /x must be programmed to /2 for GMII (default) and to /5 for RGMII.
B. If EMAC is enabled with RGMII, or GMII, CLKIN2 frequency must be 25 MHz.
C. CLKIN2 is a 3.3-V signal.

**Figure 7-23. PLL2 Block Diagram**

### 7.8.1 PLL2 Controller Device-Specific Information

#### 7.8.1.1 Internal Clocks and Maximum Operating Frequencies

As shown in Figure 7-23, the output of PLL2, PLLOUT, is divided by 2 and directly fed to the DDR2 memory controller. This clock is used by the DDR2 memory controller to generate DDR2CLKOUT and $\overline{\text{DDR2CLKOUT}}$. Note that, internally, the data bus interface of the DDR2 memory controller is clocked by SYSCLK2 of the PLL1 controller.

The PLLOUT/2 clock is also fed back into the PLL2 controller where it becomes SYSREFCLK. Divider D1 of the PLL2 controller generates SYSCLK1 for the Ethernet media access controller (EMAC). The EMAC uses SYSCLK1 to generate the necessary clock for each of its interfaces. Divider D1 should be programmed to ÷2 mode [default] when using the Gigabit Media Independent Interface (GMII) mode and to ÷5 mode when using the Reduce Gigabit Media Independent Interface (RGMII). Divider D1 is software programmable and, if necessary, must be programmed after device reset to ÷5 when the RGMII mode of the EMAC is used. Note that, internally, the data bus interface of the EMAC is clocked by SYSCLK3 of the PLL2 controller.

Note that there is a minimum and maximum operating frequency for PLLREF, PLLOUT, and SYSCLK1. The clock generator must not be configured to exceed any of these constraints. For the PLL clocks input and output frequency ranges, see Table 7-31. Also, when EMAC is enabled with RGMII or GMII, CLKIN2 must be 25 MHz.

**Table 7-31. PLL2 Clock Frequency Ranges**

| CLOCK SIGNAL | MIN | MAX | UNIT |
|---|---|---|---|
| PLLREF (PLLEN = 1) | 12.5 | 26.7 | MHz |
| PLLOUT | 250 | 533 | MHz |
| SYSCLK1 (1) | 50 | 125 | MHz |

(1) SYSCLK1 restriction applies only when the EMAC is enabled and the RGMII or GMII modes are used. SYSCLK1 must be programmed to 125 MHz when the GMII mode is used and to 50 MHz when the RGMII mode is used.

#### 7.8.1.2 PLL2 Controller Operating Modes

Unlike the PLL1 controller which can operate in bypass and a PLL mode, the PLL2 controller only operates in PLL mode. In this mode, SYSREFCLK is generated outside the PLL2 controller by dividing the output of PLL2 by two.

The PLL2 controller is affected by power-on reset and warm reset. During these resets the PLL2 controller registers get reset to their default values. The internal clocks of the PLL2 controller are also affected as described in Section 7.6, *Reset Controller*.

PLL2 is only unlocked during the power-up sequence (see Section 7.6, *Reset Controller* ) and is locked by the time the RESETSTAT pin goes high. It does not lose lock during any of the other resets.

### 7.8.2 PLL2 Controller Memory Map

The memory map of the PLL2 controller is shown in Table 7-32. Note that only registers documented here are accessible on the TMS320C6454 device. Other addresses in the PLL2 controller memory map should not be modified.

**Table 7-32. PLL2 Controller Registers**

| HEX ADDRESS RANGE | ACRONYM | DESCRIPTION |
|---|---|---|
| 029C 0000 - 029C 0114 | - | Reserved |
| 029C 0118 | PLLDIV1 | PLL Controller Divider 1 Register |
| 029C 011C - 029C 0134 | - | Reserved |
| 029C 0138 | PLLCMD | PLL Controller Command Register |
| 029C 013C | PLLSTAT | PLL Controller Status Register |
| 029C 0140 | ALNCTL | PLL Controller Clock Align Control Register |
| 029C 0144 | DCHANGE | PLLDIV Ratio Change Status Register |
| 029C 0148 | - | Reserved |
| 029C 014C | - | Reserved |
| 029C 0150 | SYSTAT | SYSCLK Status Register |
| 029C 0154 - 029C 0190 | - | Reserved |
| 029C 0194 - 029C 01FF | - | Reserved |
| 029C 0200 - 029C FFFF | - | Reserved |

### 7.8.3 PLL2 Controller Register Descriptions

This section provides a description of the PLL2 controller registers. For details on the operation of the PLL controller module, see the *TMS320C645x DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (literature number SPRUE56).

**NOTE:** The PLL2 controller registers can only be accessed using the CPU or the emulator.

Not all of the registers documented in the *TMS320C645x DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (literature number SPRUE56) are supported on the TMS320C6454 device. Only those registers documented in this section are supported. Furthermore, only the bits within the registers described here are supported. You should not write to any reserved memory location or change the value of reserved bits.

### 7.8.3.1 PLL Controller Divider 1 Register

The PLL controller divider 1 register (PLLDIV1) is shown in Figure 7-24 and described in Table 7-33.

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | | | | | | | | | | | | | 5 | 4 | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D1EN | Reserved | | | | | | | | | | | | | | RATIO | | | | | | | | | |
| R/W-1 | R-0 | | | | | | | | | | | | | | R/W-1 | | | | | | | | | |

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

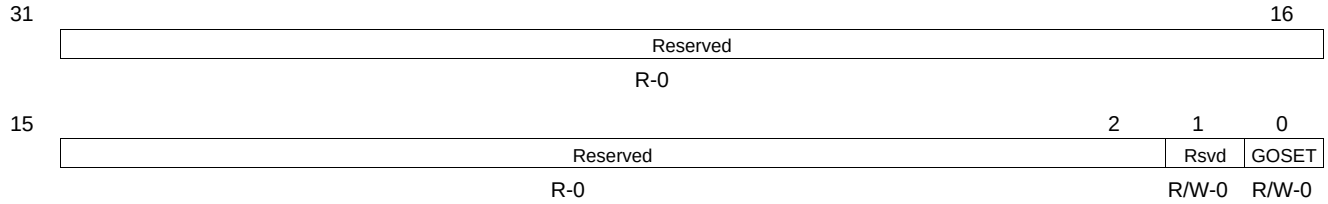
**Figure 7-24. PLL Controller Divider 1 Register (PLLDIV1) [Hex Address: 029C 0118]**

**Table 7-33. PLL Controller Divider 1 Register (PLLDIV1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:16 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 15 | D1EN | | Divider D1 enable bit. |
| | | 0 | Divider D1 is disabled. No clock output. |
| | | 1 | Divider D1 is enabled. |
| 14:5 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4:0 | RATIO | 0-1Fh | Divider ratio bits. |
| | | 1h | ÷2. Divide frequency by 2. |
| | | 4h | ÷5. Divide frequency by 5. |
| | | Others | Reserved |

### 7.8.3.2 PLL Controller Command Register

The PLL controller command register (PLLCMD) contains the command bit for GO operation. PLLCMD is shown in Figure 7-25 and described in Table 7-34.




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

**Figure 7-25. PLL Controller Command Register (PLLCMD) [Hex Address: 029C 0138]**

**Table 7-34. PLL Controller Command Register (PLLCMD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:2 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 1 | Reserved | | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 0 | GOSET | | GO operation command for SYSCLK rate change and phase alignment. Before setting this bit to 1 to initiate a GO operation, check the GOSTAT bit in the PLLSTAT register to ensure all previous GO operations have completed. |
| | | 0 | No effect. Write of 0 clears bit to 0. |
| | | 1 | Initiates GO operation. Write of 1 initiates GO operation. Once set, GOSET remains set but further writes of 1 can initiate the GO operation. |

### 7.8.3.3 PLL Controller Status Register

The PLL controller status register (PLLSTAT) shows the PLL controller status. PLLSTAT is shown in Figure 7-26 and described in Table 7-35.

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | 1 | 0 |
| Reserved | | | | | | | | | | | | | | | GOSTAT |
| R-0 | | | | | | | | | | | | | | | R-0 |

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

**Figure 7-26. PLL Controller Status Register (PLLSTAT) [Hex Address: 029C 013C]**

**Table 7-35. PLL Controller Status Register (PLLSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 0 | GOSTAT | | GO operation status. |
| | | 0 | Go operation is not in progress. SYSCLK divide ratios are not being changed. |
| | | 1 | GO operation is in progress. SYSCLK divide ratios are being changed. |

### 7.8.3.4 PLL Controller Clock Align Control Register

The PLL controller clock align control register (ALNCTL) is shown in Figure 7-27 and described in Table 7-36.

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | 1 | 0 |
| Reserved | | | | | | | | | | | | | | | ALN1 |
| R-0 | | | | | | | | | | | | | | | R/W-1 |

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

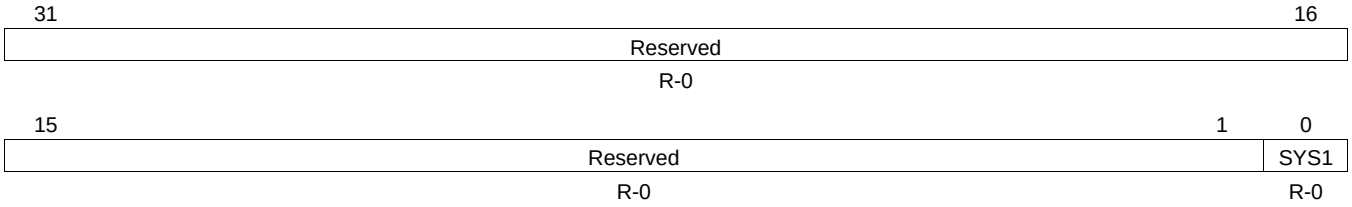
**Figure 7-27. PLL Controller Clock Align Control Register (ALNCTL) [Hex Address: 029C 0140]**

**Table 7-36. PLL Controller Clock Align Control Register (ALNCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 0 | ALN1 | | SYSCLK1 alignment. Do not change the default values of these fields. |
| | | 0 | Do not align SYSCLK1 during GO operation. If SYS1 in DCHANGE is set to 1, SYSCLK1 switches to the new ratio immediately after the GOSET bit in PLLCMD is set. |
| | | 1 | Align SYSCLK1 when the GOSET bit in PLLCMD is set. The SYSCLK1 ratio is set to the ratio programmed in the RATIO bit in PLLDIV1. |

#### 7.8.3.5 PLLDIV Ratio Change Status Register

Whenever a different ratio is written to the PLLDIV1 register, the PLLCTRL flags the change in the DCHANGE status register. During the GO operation, the PLL controller will only change the divide ratio SYSCLK1 if SYS1 in DCHANGE is 1. The PLLDIV divider ratio change status register is shown in Figure 7-28 and described in Table 7-37.




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

**Figure 7-28. PLLDIV Divider Ratio Change Status Register (DCHANGE) [Hex Address: 029C 0144]**

**Table 7-37. PLLDIV Divider Ratio Change Status Register (DCHANGE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 0 | SYS1 | | SYSCLK1 divide ratio has been modified. SYSCLK1 ratio will be modified during GO operation. |
| | | 0 | SYSCLK1 ratio has not been modified. When GOSET is set, SYSCLK1 will not be affected. |
| | | 1 | SYSCLK1 ratio has been modified. When GOSET is set, SYSCLK1 will change to the new ratio. |

### 7.8.3.6 SYSCLK Status Register

The SYSCLK status register (SYSTAT) shows the status of the system clock (SYSCLK1). SYSTAT is shown in Figure 7-29 and described in Table 7-38.

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | SYS1ON |
| R-0 | | R-1 |

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

**Figure 7-29. SYSCLK Status Register [Hex Address: 029C 0150]**

**Table 7-38. SYSCLK Status Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31:1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 0 | SYS1ON | | SYSCLK1 on status. |
| | | 0 | SYSCLK1 is gated. |
| | | 1 | SYSCLK1 is on. |

### 7.8.4 PLL2 Controller Input Clock Electrical Data/Timing

**Table 7-39. Timing Requirements for CLKIN2 (1) (2) (3)**

(see Figure 7-30)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(CLKIN2)}$ | Cycle time, CLKIN2 | 37.5 | 80 | ns |
| 2 | $t_{w(CLKIN2H)}$ | Pulse duration, CLKIN2 high | 0.4C | | ns |
| 3 | $t_{w(CLKIN2L)}$ | Pulse duration, CLKIN2 low | 0.4C | | ns |
| 4 | $t_{t(CLKIN2)}$ | Transition time, CLKIN2 | | 1.2 | ns |
| 5 | $t_{J(CLKIN2)}$ | Period jitter (peak-to-peak), CLKIN2 | | 100 | ps |

(1) The reference points for the rise and fall transitions are measured at 3.3 V $V_{IL}$ MAX and $V_{IH}$ MIN.
(2) C = **CLKIN2** cycle time in ns. For example, when CLKIN2 frequency is 25 MHz, use C = 40 ns.
(3) If EMAC is enabled with RGMII or GMII, CLKIN2 cycle time must be 40 ns (25 MHz).

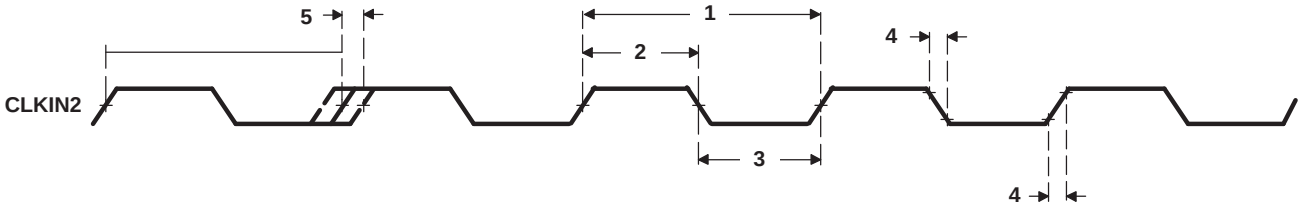



**Figure 7-30. CLKIN2 Timing**

## 7.9 DDR2 Memory Controller

The 32-bit, 533-MHz (data rate) DDR2 Memory Controller bus of the C6454 device is used to interface to JESD79-2A standard-compliant DDR2 SDRAM devices. The DDR2 external bus only interfaces to DDR2 SDRAM devices (up to 512 MB); it does not share the bus with any other types of peripherals. The decoupling of DDR2 memories from other devices both simplifies board design and provides I/O concurrency from a second external memory interface, EMIFA.

The internal data bus clock frequency and DDR2 bus clock frequency directly affect the maximum throughput of the DDR2 bus. The clock frequency of the DDR2 bus is equal to the CLKIN2 frequency multiplied by 10. The internal data bus clock frequency of the DDR2 Memory Controller is fixed at a divide-by-three ratio of the CPU frequency. The maximum DDR2 throughput is determined by the smaller of the two bus frequencies. For example, if the internal data bus frequency is 333 MHz (CPU frequency is 1 GHz) and the DDR2 bus frequency is 267 MHz (CLKIN2 frequency is 26.7 MHz), the maximum data rate achievable by the DDR2 memory controller is 2.1 Gbytes/sec. The DDR2 bus is designed to sustain a maximum throughput of up to 2.1 Gbytes/sec at a 533-MHz data rate (267-MHz clock rate), as long as data requests are pending in the DDR2 Memory Controller.

### *7.9.1 DDR2 Memory Controller Device-Specific Information*

The approach to specifying interface timing for the DDR2 memory bus is different than on other interfaces such as EMIF, HPI, and McBSP. For these other interfaces the device timing was specified in terms of data manual specifications and I/O buffer information specification (IBIS) models.

For the C6454 DDR2 memory bus, the approach is to specify compatible DDR2 devices and provide the printed circuit board (PCB) solution and guidelines directly to the user. Texas Instruments (TI) has performed the simulation and system characterization to ensure all DDR2 interface timings in this solution are met. The complete DDR2 system solution is documented in the *Implementing DDR2 PCB Layout on the TMS320C6455/C6454* application report (literature number SPRAAA7).

**TI *only* supports designs that follow the board design guidelines outlined in the SPRAAA7 application report.**

The DDR2 Memory Controller pins must be enabled by setting the DDR2_EN configuration pin (ABA0) high during device reset. For more details, see Section 3.1, *Device Configuration at Device Reset*.

The ODT[1:0] pins of the memory controller must be left unconnected. The ODT pins on the DDR2 memory device(s) must be connected to ground.

The DDR2 memory controller on the C6454 device supports the following memory topologies:

- A 32-bit wide configuration interfacing to two 16-bit wide DDR2 SDRAM devices.
- A 16-bit wide configuration interfacing to a single 16-bit wide DDR2 SDRAM device.

A race condition may exist when certain masters write data to the DDR2 memory controller. For example, if master A passes a software message via a buffer in external memory and does not wait for indication that the write completes, when master B attempts to read the software message, then the master B read may bypass the master A write and, thus, master B may read stale data and, therefore, receive an incorrect message.

Some master peripherals (e.g., EDMA3 transfer controllers) will always wait for the write to complete before signaling an interrupt to the system, thus avoiding this race condition. For masters that do not have hardware guarantee of write-read ordering, it may be necessary to guarantee data ordering via software.

If master A does not wait for indication that a write is complete, it must perform the following workaround:

1. Perform the required write.
2. Perform a dummy write to the DDR2 memory controller module ID and revision register.
3. Perform a dummy read to the DDR2 memory controller module ID and revision register.
4. Indicate to master B that the data is ready to be read after completion of the read in step 3. The completion of the read in step 3 ensures that the previous write was done.

### 7.9.2 DDR2 Memory Controller Peripheral Register Descriptions

**Table 7-40. DDR2 Memory Controller Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 7800 0000 | MIDR | DDR2 Memory Controller Module and Revision Register |
| 7800 0004 | DMCSTAT | DDR2 Memory Controller Status Register |
| 7800 0008 | SDCFG | DDR2 Memory Controller SDRAM Configuration Register |
| 7800 000C | SDRFC | DDR2 Memory Controller SDRAM Refresh Control Register |
| 7800 0010 | SDTIM1 | DDR2 Memory Controller SDRAM Timing 1 Register |
| 7800 0014 | SDTIM2 | DDR2 Memory Controller SDRAM Timing 2 Register |
| 7800 0018 | - | Reserved |
| 7800 0020 | BPRIO | DDR2 Memory Controller Burst Priority Register |
| 7800 0024 - 7800 004C | - | Reserved |
| 7800 0050 - 7800 0078 | - | Reserved |
| 7800 007C - 7800 00BC | - | Reserved |
| 7800 00C0 - 7800 00E0 | - | Reserved |
| 7800 00E4 | DMCCTL | DDR2 Memory Controller Control Register |
| 7800 00E8 - 7800 00FC | - | Reserved |
| 7800 0100 - 7FFF FFFF | - | Reserved |

### 7.9.3 DDR2 Memory Controller Electrical Data/Timing

The *Implementing DDR2 PCB Layout on the TMS320C6455/C6454* application report (literature number SPRAAA7) specifies a complete DDR2 interface solution for the C6454 device as well as a list of compatible DDR2 devices. TI has performed the simulation and system characterization to ensure all DDR2 interface timings in this solution are met; therefore, no electrical data/timing information is supplied here for this interface.

**TI *only* supports designs that follow the board design guidelines outlined in the SPRAAA7 application report.**

## 7.10 External Memory Interface A (EMIFA)

The EMIFA can interface to a variety of external devices or ASICs, including:

- Pipelined and flow-through Synchronous-Burst SRAM (SBSRAM)
- ZBT (Zero Bus Turnaround) SRAM and Late Write SRAM
- Synchronous FIFOs
- Asynchronous memory, including SRAM, ROM, and Flash

### *7.10.1 EMIFA Device-Specific Information*

Timing analysis must be done to verify all AC timings are met. TI recommends utilizing I/O buffer information specification (IBIS) to analyze all AC timings.

To properly use IBIS models to attain accurate timing analysis for a given system, see the *Using IBIS Models for Timing Analysis* application report (literature number SPRA839).

To maintain signal integrity, serial termination resistors should be inserted into all EMIF output signal lines (for the EMIF output signals, see Table 2-3, *Terminal Functions*).

A race condition may exist when certain masters write data to the EMIFA. For example, if master A passes a software message via a buffer in external memory and does not wait for indication that the write completes, when master B attempts to read the software message, then the master B read may bypass the master A write and, thus, master B may read stale data and, therefore, receive an incorrect message.

Some master peripherals (e.g., EDMA3 transfer controllers) will always wait for the write to complete before signaling an interrupt to the system, thus avoiding this race condition. For masters that do not have hardware guarantee of write-read ordering, it may be necessary to guarantee data ordering via software.

If master A does not wait for indication that a write is complete, it must perform the following workaround:

1. Perform the required write.
2. Perform a dummy write to the EMIFA module ID and revision register.
3. Perform a dummy read to the EMIFA module ID and revision register.
4. Indicate to master B that the data is ready to be read after completion of the read in step 3. The completion of the read in step 3 ensures that the previous write was done.

### 7.10.2 EMIFA Peripheral Register Descriptions

**Table 7-41. EMIFA Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 7000 0000 | MIDR | Module ID and Revision Register |
| 7000 0004 | STAT | Status Register |
| 7000 0008 | - | Reserved |
| 7000 000C - 7000 001C | - | Reserved |
| 7000 0020 | BURST_PRIO | Burst Priority Register |
| 7000 0024 - 7000 004C | - | Reserved |
| 7000 0050 - 7000 007C | - | Reserved |
| 7000 0080 | CE2CFG | EMIFA CE2 Configuration Register |
| 7000 0084 | CE3CFG | EMIFA CE3 Configuration Register |
| 7000 0088 | CE4CFG | EMIFA CE4 Configuration Register |
| 7000 008C | CE5CFG | EMIFA CE5 Configuration Register |
| 7000 0090 - 7000 009C | - | Reserved |
| 7000 00A0 | AWCC | EMIFA Async Wait Cycle Configuration Register |
| 7000 00A4 - 7000 00BC | - | Reserved |
| 7000 00C0 | INTRAW | EMIFA Interrupt RAW Register |
| 7000 00C4 | INTMSK | EMIFA Interrupt Masked Register |
| 7000 00C8 | INTMSKSET | EMIFA Interrupt Mask Set Register |
| 7000 00CC | INTMSKCLR | EMIFA Interrupt Mask Clear Register |
| 7000 00D0 - 7000 00DC | - | Reserved |
| 7000 00E0 - 77FF FFFF | - | Reserved |

### 7.10.3 EMIFA Electrical Data/Timing

**Table 7-42. Timing Requirements for AECLKIN for EMIFA(1) (2)**

(see Figure 7-31)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(EKI)}$ | Cycle time, AECLKIN | 6(3) | 40 | ns |
| 2 | $t_{w(EKIH)}$ | Pulse duration, AECLKIN high | 2.7 | | ns |
| 3 | $t_{w(EKIL)}$ | Pulse duration, AECLKIN low | 2.7 | | ns |
| 4 | $t_{t(EKI)}$ | Transition time, AECLKIN | | 2 | ns |
| 5 | $t_{J(EKI)}$ | Period Jitter, AECLKIN | | 0.02E(4) | ns |

(1) The reference points for the rise and fall transitions are measured at $V_{IL}$ MAX and $V_{IH}$ MIN.
(2) E = the EMIF input clock (AECLKIN or SYSCLK4) period in ns for EMIFA.
(3) Minimum AECLKIN cycle times *must* be met, even when AECLKIN is generated by an internal clock source. Minimum AECLKIN times are based on internal logic speed; the maximum useable speed of the EMIF may be lower due to AC timing requirements.
(4) This timing only applies when AECLKIN is used for EMIFA.

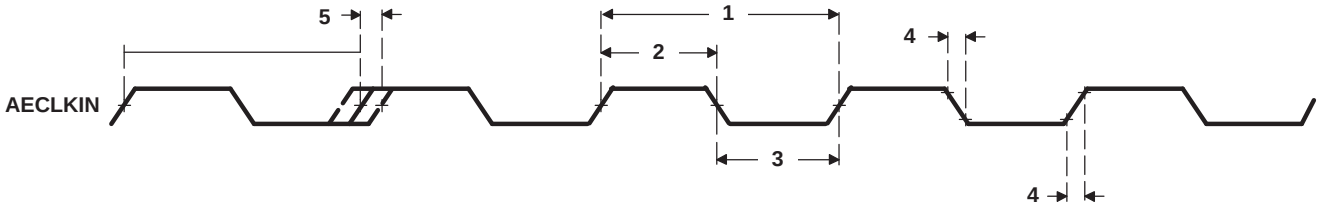



**Figure 7-31. AECLKIN Timing for EMIFA**

**Table 7-43. Switching Characteristics Over Recommended Operating Conditions for AECLKOUT for the EMIFA Module(1) (2) (3)**

(see Figure 7-32)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(EKO)}$ | Cycle time, AECLKOUT | E - 0.7 | E + 0.7 | ns |
| 2 | $t_{w(EKOH)}$ | Pulse duration, AECLKOUT high | EH - 0.7 | EH + 0.7 | ns |
| 3 | $t_{w(EKOL)}$ | Pulse duration, AECLKOUT low | EL - 0.7 | EL + 0.7 | ns |
| 4 | $t_{t(EKO)}$ | Transition time, AECLKOUT | | 1 | ns |
| 5 | $t_{d(EKIH-EKOH)}$ | Delay time, AECLKIN high to AECLKOUT high | 1 | 8 | ns |
| 6 | $t_{d(EKIL-EKOL)}$ | Delay time, AECLKIN low to AECLKOUT low | 1 | 8 | ns |

(1) E = the EMIF input clock (AECLKIN or SYSCLK4) period in ns for EMIFA.
(2) The reference points for the rise and fall transitions are measured at $V_{OL}$ MAX and $V_{OH}$ MIN.
(3) EH is the high period of E (EMIF input clock period) in ns and EL is the low period of E (EMIF input clock period) in ns for EMIFA.

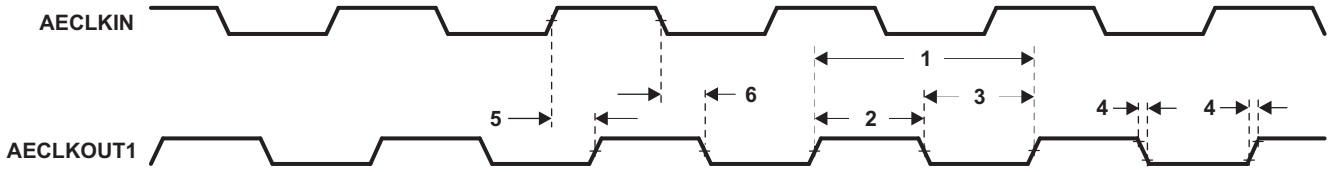



**Figure 7-32. AECLKOUT Timing for the EMIFA Module**

### 7.10.3.1 Asynchronous Memory Timing

**Table 7-44. Timing Requirements for Asynchronous Memory Cycles for EMIFA Module(1) (2) (3)**

(see Figure 7-33 and Figure 7-34)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{su(EDV-A\ OEH)}$ | Setup time, AEDx valid before $\overline{AAOE}$ high | 6.5 | | ns |
| 4 | $t_{h(AOEH-EDV)}$ | Hold time, AEDx valid after $\overline{AAOE}$ high | 0 | | ns |
| 5 | $t_{su(ARDY-EKOH)}$ | Setup time, AARDY valid before AECLKOUT low | 1 | | ns |
| 6 | $t_{h(EKOH-ARDY)}$ | Hold time, AARDY valid after AECLKOUT low | 2 | | ns |
| 7 | $t_{w(ARDY\ )}$ | Pulse width, AARDY assertion and deassertion | 2E + 5 | | ns |
| 8 | $t_{d(ARDY-HOLD)}$ | Delay time, from AARDY sampled deasserted on AECLKOUT falling to beginning of programmed hold period | | 4E | ns |
| 9 | $t_{su(ARDY-HOLD)}$ | Setup time, before end of programmed strobe period by which AARDY should be asserted in order to insert extended strobe wait states. | 2E | | ns |

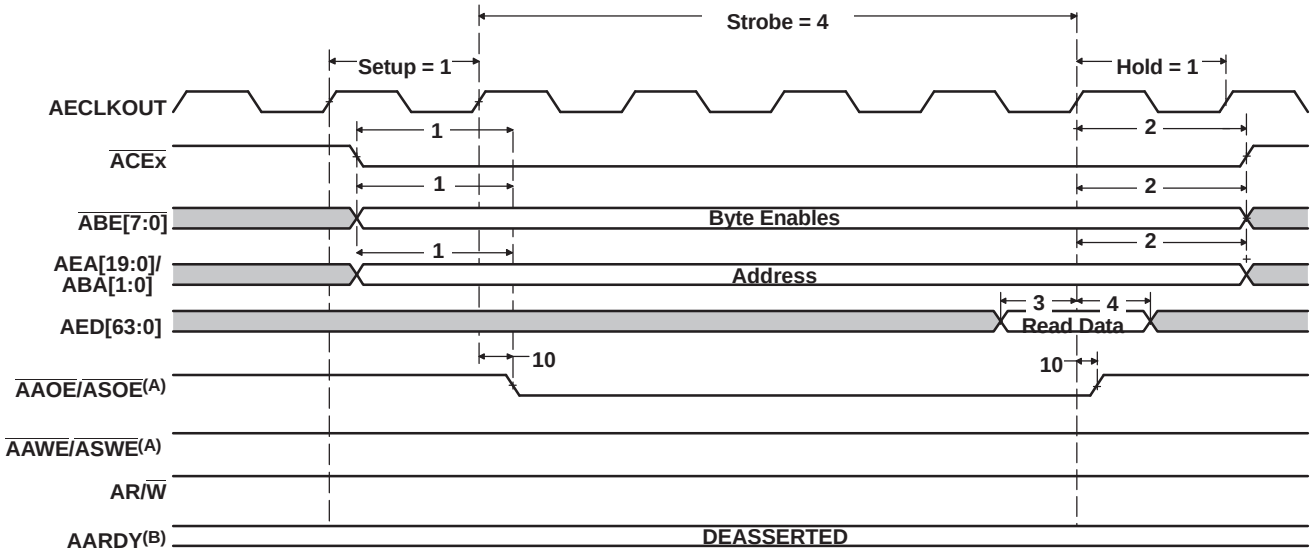
(1) E = AECLKOUT period in ns for EMIFA
(2) To ensure data setup time, simply program the strobe width wide enough.
(3) AARDY is internally synchronized. To use AARDY as an asynchronous input, the pulse width of the AARDY signal should be at least 2E to ensure setup and hold time is met.

**Table 7-45. Switching Characteristics Over Recommended Operating Conditions for Asynchronous Memory Cycles for EMIFA Module(1) (2) (3)**

(see Figure 7-33 and Figure 7-34)

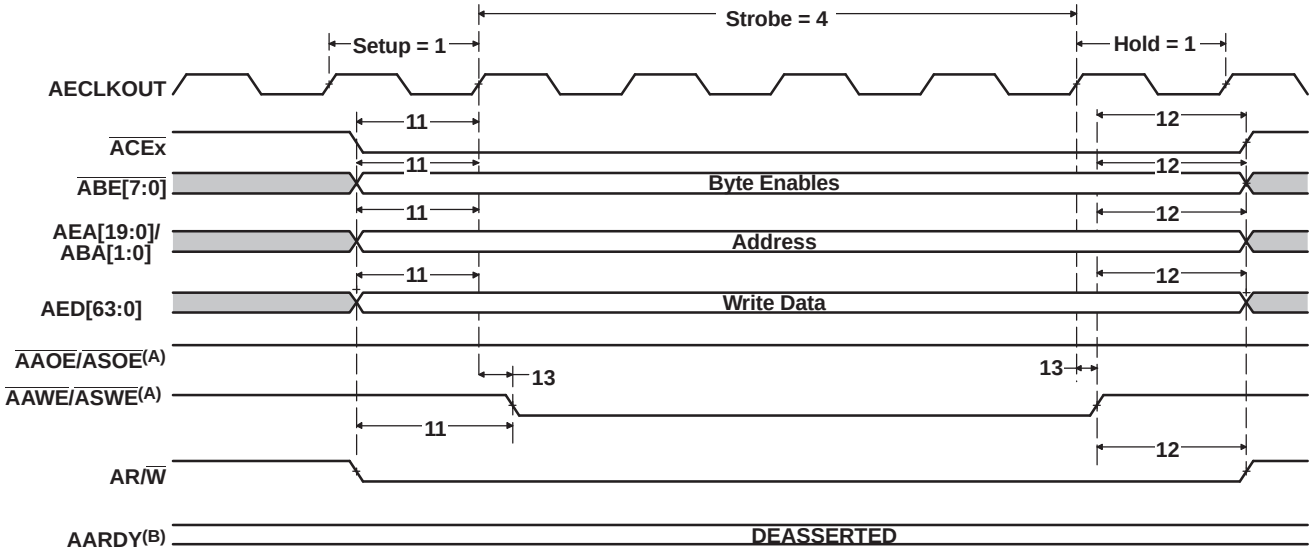
| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{osu(SELV-AOEL)}$ | Output setup time, select signals valid to $\overline{AAOE}$ low | RS * E - 1.5 | | ns |
| 2 | $t_{oh(AOEH-SELIV)}$ | Output hold time, $\overline{AAOE}$ high to select signals invalid | RS * E - 1.9 | | ns |
| 10 | $t_{d(EKOH-AOEV)}$ | Delay time, AECLKOUT high to $\overline{AAOE}$ valid | 1 | 7 | ns |
| 11 | $t_{osu(SELV-AWEL)}$ | Output setup time, select signals valid to $\overline{AAWE}$ low | WS * E - 1.7 | | ns |
| 12 | $t_{oh(AWEH-SELIV)}$ | Output hold time, $\overline{AAWE}$ high to select signals invalid | WH * E - 1.8 | | ns |
| 13 | $t_{d(EKOH-AWEV)}$ | Delay time, AECLKOUT high to $\overline{AAWE}$ valid | 1.3 | 7.1 | ns |

(1) E = AECLKOUT period in ns for EMIFA
(2) RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed via the EMIFA CE Configuration registers (CEnCFG).
(3) Select signals for EMIFA include: $\overline{ACEx}$, $\overline{ABE}[7:0]$, AEA[19:0], ABA[1:0]; and for EMIFA writes, also include AR/$\overline{W}$, AED[63:0].




A $\overline{AAOE}$/$\overline{ASOE}$ and $\overline{AAWE}$/$\overline{ASWE}$ operate as $\overline{AAOE}$ (identified under select signals) and $\overline{AAWE}$, respectively, during asynchronous memory accesses.
B Polarity of the AARDY signal is programmable through the AP field of the EMIFA Async Wait Cycle Configuration register (AWCC).

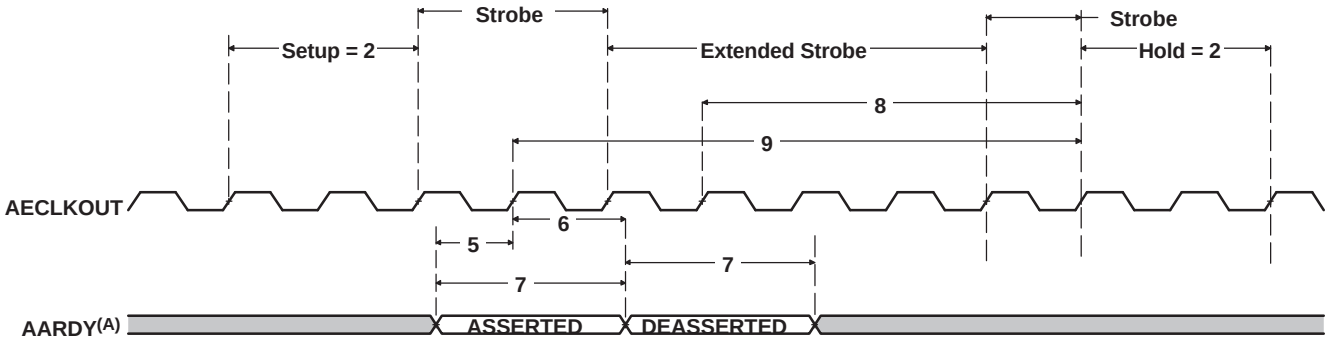
**Figure 7-33. Asynchronous Memory Read Timing for EMIFA**




A AAOE/ASOE and AAWE/ASWE operate as AAOE (identified under select signals) and AAWE, respectively, during asynchronous memory accesses.

B Polarity of the AARDY signal is programmable through the AP field of the EMIFA Async Wait Cycle Configuration register (AWCC).

**Figure 7-34. Asynchronous Memory Write Timing for EMIFA**




A Polarity of the AARDY signal is programmable through the AP field of the EMIFA Async Wait Cycle Configuration register (AWCC).

**Figure 7-35. AARDY Timing**

### 7.10.3.2 Programmable Synchronous Interface Timing

**Table 7-46. Timing Requirements for Programmable Synchronous Interface Cycles for EMIFA Module**

(see Figure 7-36)

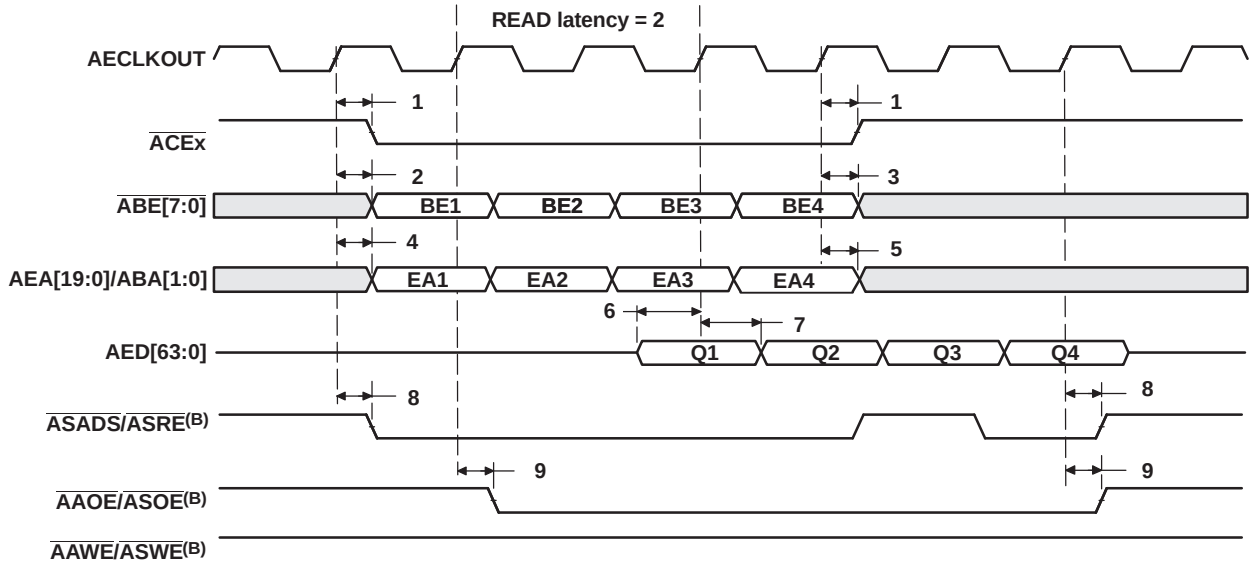
| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 6 | $t_{su(EDV-EKOH)}$ | Setup time, read AEDx valid before AECLKOUT high | 2 | | ns |
| 7 | $t_{h(EKOH-EDV)}$ | Hold time, read AEDx valid after AECLKOUT high | 1.5 | | ns |

**Table 7-47. Switching Characteristics Over Recommended Operating Conditions for Programmable Synchronous Interface Cycles for EMIFA Module(1)**

(see Figure 7-36 through Figure 7-38)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(EKOH-CEV)}$ | Delay time, AECLKOUT high to $\overline{ACE}$x valid | 1.3 | 4.9 | ns |
| 2 | $t_{d(EKOH-BEV)}$ | Delay time, AECLKOUT high to $\overline{ABE}$x valid | | 4.9 | ns |
| 3 | $t_{d(EKOH-BEIV)}$ | Delay time, AECLKOUT high to $\overline{ABE}$x invalid | 1.3 | | ns |
| 4 | $t_{d(EKOH-EAV)}$ | Delay time, AECLKOUT high to AEAx valid | | 4.9 | ns |
| 5 | $t_{d(EKOH-EAIV)}$ | Delay time, AECLKOUT high to AEAx invalid | 1.3 | | ns |
| 8 | $t_{d(EKOH-ADSV)}$ | Delay time, AECLKOUT high to $\overline{ASADS}$/$\overline{ASRE}$ valid | 1.3 | 4.9 | ns |
| 9 | $t_{d(EKOH-OEV)}$ | Delay time, AECLKOUT high to $\overline{ASOE}$ valid | 1.3 | 4.9 | ns |
| 10 | $t_{d(EKOH-EDV)}$ | Delay time, AECLKOUT high to AEDx valid | | 4.9 | ns |
| 11 | $t_{d(EKOH-EDIV)}$ | Delay time, AECLKOUT high to AEDx invalid | 1.3 | | ns |
| 12 | $t_{d(EKOH-WEV)}$ | Delay time, AECLKOUT high to $\overline{ASWE}$ valid | 1.3 | 4.9 | ns |

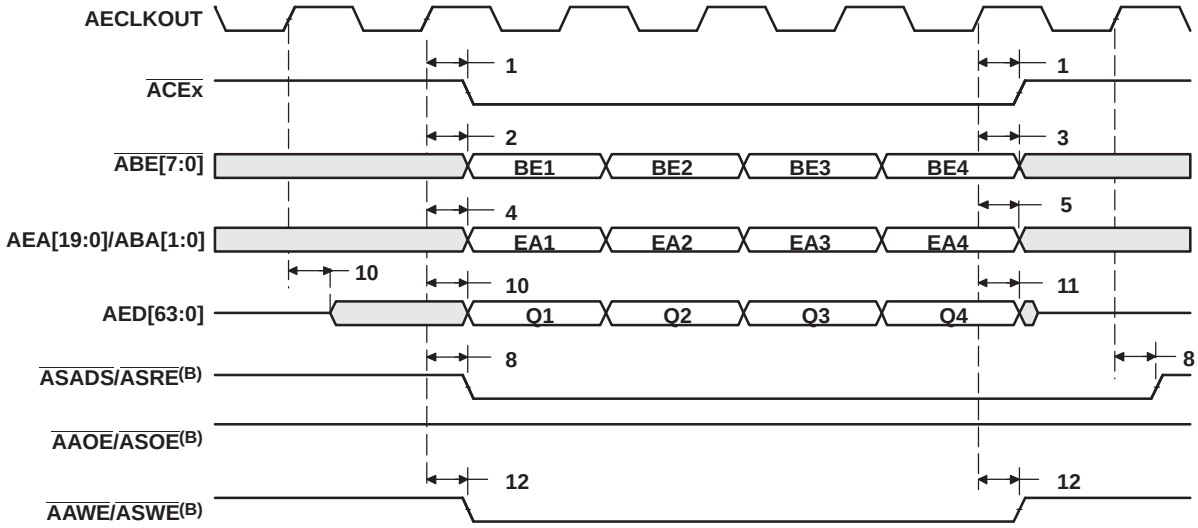
(1) The following parameters are programmable via the EMIFA CE Configuration registers (CEnCFG):
- Read latency (R_LTNCY): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (W_LTNCY): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{ACE}$x assertion length (CE_EXT): For standard SBSRAM or ZBT SRAM interface, $\overline{ACE}$x goes inactive after the final command has been issued (CE_EXT = 0). For synchronous FIFO interface with glue, $\overline{ACE}$x is active when $\overline{ASOE}$ is active (CE_EXT = 1).
- Function of $\overline{ASADS}$/$\overline{ASRE}$ (R_ENABLE): For standard SBSRAM or ZBT SRAM interface, $\overline{ASADS}$/$\overline{ASRE}$ acts as $\overline{ASADS}$ with deselect cycles (R_ENABLE = 0). For FIFO interface, $\overline{ASADS}$/$\overline{ASRE}$ acts as $\overline{ASRE}$ with NO deselect cycles (R_ENABLE = 1).




A The following parameters are programmable via the EMIFA Chip Select n Configuration Register (CESECn):
- Read latency (R_LTNCY): 1-, 2-, or 3-cycle read latency
- Write latency (W_LTNCY): 0-, 1-, 2-, or 3-cycle write latency
- ACEx assertion length (CE_EXT): For standard SBSRAM or ZBT SRAM interface, ACEx goes inactive after the final command has been issued (CE_EXT = 0). For synchronous FIFO interface, ACEx is active when ASOE is active (CE_EXT = 1).
- Function of ASADS/ASRE (R_ENABLE): For standard SBSRAM or ZBT SRAM interface, ASADS/ASRE acts as ASADS with deselect cycles (R_ENABLE = 0). For FIFO interface, ASADS/ASRE acts as SRE with NO deselect cycles (R_ENABLE = 1).
- In this figure R_LTNCY = 2, CE_EXT = 0, R_ENABLE = 0, and SSEL = 1.

B AAOE/ASOE, and AAWE/ASWE operate as ASOE, and ASWE, respectively, during programmable synchronous interface accesses.

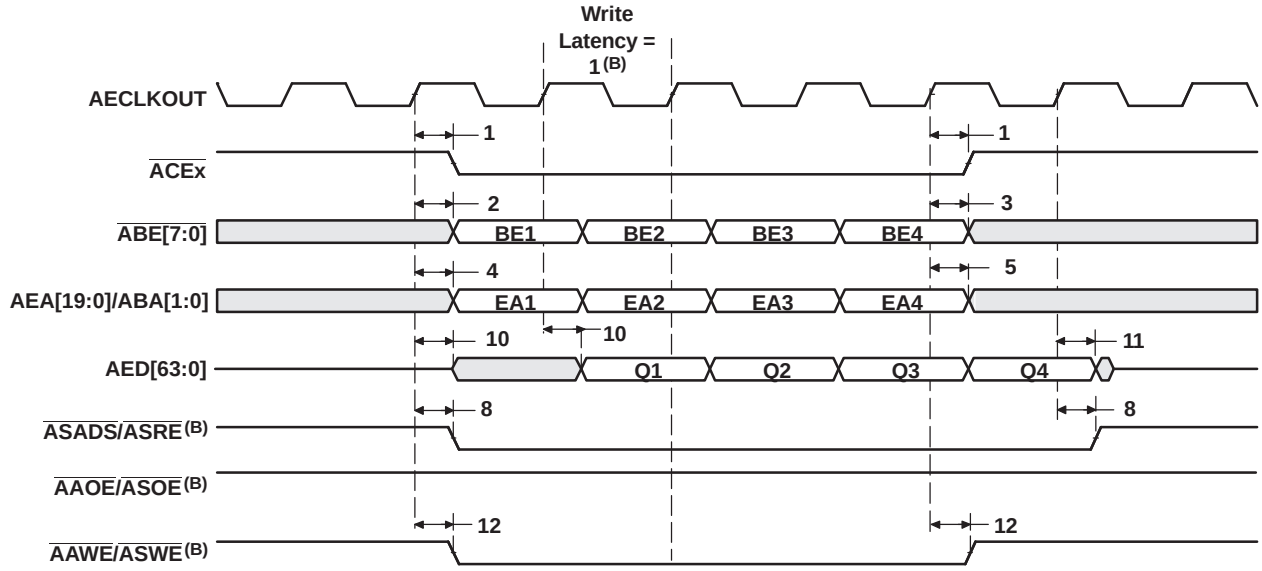
**Figure 7-36. Programmable Synchronous Interface Read Timing for EMIFA (With Read Latency = 2)(A)**




A The following parameters are programmable via the EMIFA Chip Select n Configuration Register (CESECn):
- Read latency (R_LTNCY): 1-, 2-, or 3-cycle read latency
- Write latency (W_LTNCY): 0-, 1-, 2-, or 3-cycle write latency
- ACEx assertion length (CE_EXT): For standard SBSRAM or ZBT SRAM interface, ACEx goes inactive after the final command has been issued (CE_EXT = 0). For synchronous FIFO interface, ACEx is active when ASOE is active (CE_EXT = 1).
- Function of ASADS/ASRE (R_ENABLE): For standard SBSRAM or ZBT SRAM interface, ASADS/ASRE acts as ASADS with deselect cycles (R_ENABLE = 0). For FIFO interface, ASADS/ASRE acts as SRE with NO deselect cycles (R_ENABLE = 1).
- In this figure W_LTNCY = 0, CE_EXT = 0, R_ENABLE = 0, and SSEL = 1.

B AAOE/ASOE, and AAWE/ASWE operate as ASOE, and ASWE, respectively, during programmable synchronous interface accesses.

**Figure 7-37. Programmable Synchronous Interface Write Timing for EMIFA (With Write Latency = 0)(A)**




A The following parameters are programmable via the EMIFA Chip Select n Configuration Register (CESECn):
- Read latency (R_LTNCY): 1-, 2-, or 3-cycle read latency
- Write latency (W_LTNCY): 0-, 1-, 2-, or 3-cycle write latency
- ACEx assertion length (CE_EXT): For standard SBSRAM or ZBT SRAM interface, ACEx goes inactive after the final command has been issued (CE_EXT = 0). For synchronous FIFO interface, ACEx is active when ASOE is active (CE_EXT = 1).
- Function of ASADS/ASRE (R_ENABLE): For standard SBSRAM or ZBT SRAM interface, ASADS/ASRE acts as ASADS with deselect cycles (R_ENABLE = 0). For FIFO interface, ASADS/ASRE acts as SRE with NO deselect cycles (R_ENABLE = 1).
- In this figure W_LTNCY = 1, CE_EXT = 0, R_ENABLE = 0, and SSEL = 1.

B AAOE/ASOE, and AAWE/ASWE operate as ASOE, and ASWE, respectively, during programmable synchronous interface accesses.

**Figure 7-38. Programmable Synchronous Interface Write Timing for EMIFA (With Write Latency = 1) (A)**

### 7.10.4 HOLD/HOLDA Timing

**Table 7-48. Timing Requirements for the HOLD/HOLDA Cycles for EMIFA Module(1)**

(see Figure 7-39)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{h(HOLDAL-HOLDL)}$ | Hold time, HOLD low after HOLDA low | E | | ns |

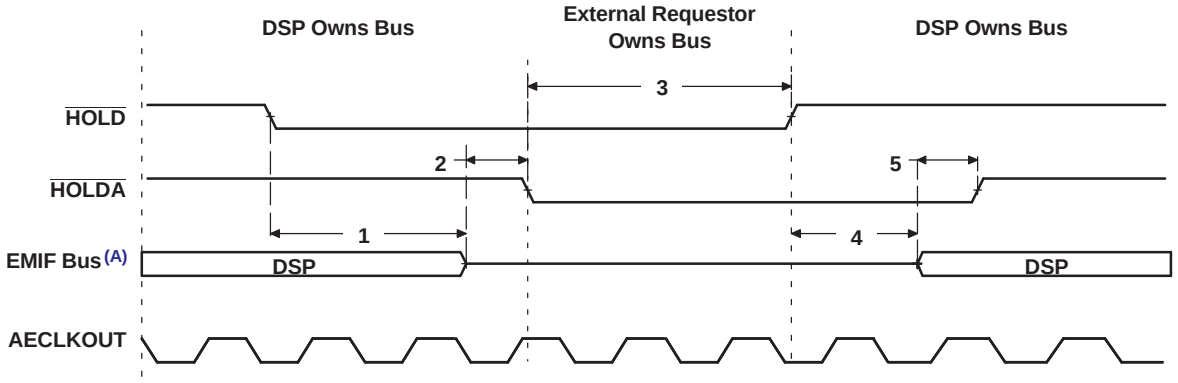
(1) E = the EMIF input clock (ECLKIN) period in ns for EMIFA.

**Table 7-49. Switching Characteristics Over Recommended Operating Conditions for the HOLD/HOLDA Cycles for EMIFA Module(1) (2)**

(see Figure 7-39)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(HOLDL-EMHZ)}$ | Delay time, HOLD low to EMIFA Bus high impedance | 2E | (3) | ns |
| 2 | $t_{d(EMHZ-HOLDAL)}$ | Delay time, EMIF Bus high impedance to HOLDA low | 0 | 2E | ns |
| 4 | $t_{d(HOLDH-EMLZ)}$ | Delay time, HOLD high to EMIF Bus low impedance | 2E | 7E | ns |
| 5 | $t_{d(EMLZ-HOLDAH)}$ | Delay time, EMIFA Bus low impedance to HOLDA high | 0 | 2E | ns |

(1) E = the EMIF input clock (ECLKIN) period in ns for EMIFA.
(2) EMIFA Bus consists of: ACE[5:2], ABE[7:0], AED[63:0], AEA[19:0], ABA[1:0], AR/W, ASADS/ASRE, AAOE/ ASOE, and AAWE/ASWE.
(3) All pending EMIF transactions are allowed to complete before HOLDA is asserted. If no bus transactions are occurring, then the minimum delay time can be achieved.




A. EMIFA Bus consists of: ACE[5:2], ABE[7:0], AED[63:0], AEA[19:0], ABA[1:0], AR/W, ASADS/ASRE, AAOE/ ASOE, and AAWE/ASWE.

**Figure 7-39. HOLD/HOLDA Timing for EMIFA**

### 7.10.5 BUSREQ Timing

**Table 7-50. Switching Characteristics Over Recommended Operating Conditions for the BUSREQ Cycles for EMIFA Module**

(see Figure 7-40)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(AEKOH-ABUSRV)}$ | Delay time, AECLKOUT high to ABUSREQ valid | 1 | 5.5 | ns |

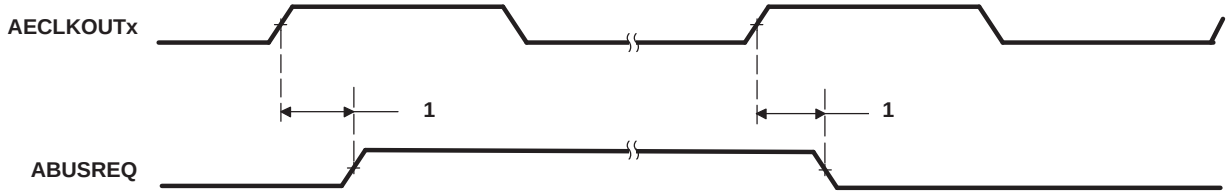



**Figure 7-40. BUSREQ Timing for EMIFA**

## 7.11 I2C Peripheral

The inter-integrated circuit (I2C) module provides an interface between a C64x+ DSP and other devices compliant with Philips Semiconductors Inter-IC bus (I2C bus™) specification version 2.1 and connected by way of an I2C bus. External components attached to this 2-wire serial bus can transmit/receive up to 8-bit data to/from the DSP through the I2C module.

### *7.11.1 I2C Device-Specific Information*

The C6454 device includes an I2C peripheral module (I2C). NOTE: when using the I2C module, ensure there are external pullup resistors on the SDA and SCL pins.

The I2C modules on the C6454 device may be used by the DSP to control local peripherals ICs (DACs, ADCs, etc.) or may be used to communicate with other controllers in a system or to implement a user interface.

The 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
- 7- and 10-Bit Device Addressing Modes
- Multi-Master (Transmit/Receive) and Slave (Transmit/Receive) Functionality
- Events: DMA, Interrupt, or Polling
- Slew-Rate Limited Open-Drain Output Buffers

Figure 7-41 is a block diagram of the I2C module.

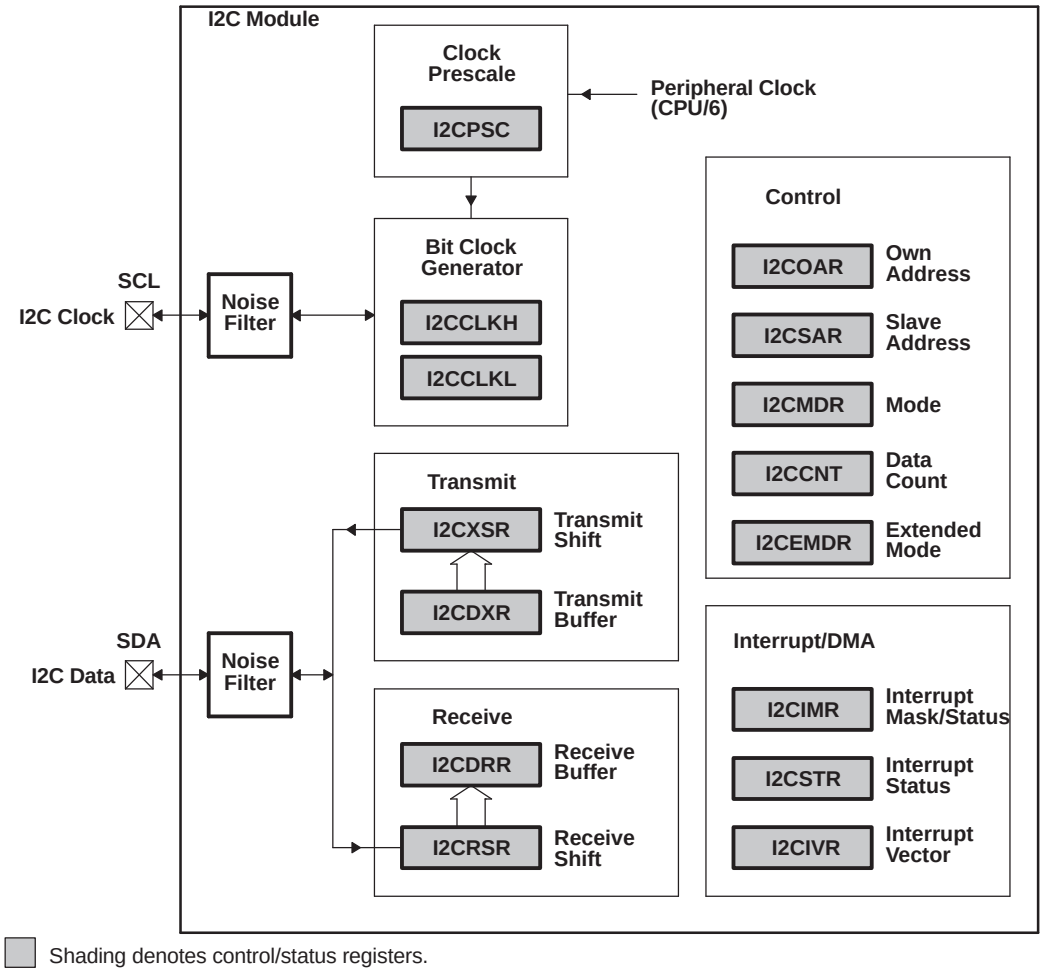



**Figure 7-41. I2C Module Block Diagram**

### 7.11.2 I2C Peripheral Register Descriptions

**Table 7-51. I2C Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02B0 4000 | ICOAR | I2C own address register |
| 02B0 4004 | ICIMR | I2C interrupt mask/status register |
| 02B0 4008 | ICSTR | I2C interrupt status register |
| 02B0 400C | ICCLKL | I2C clock low-time divider register |
| 02B0 4010 | ICCLKH | I2C clock high-time divider register |
| 02B0 4014 | ICCNT | I2C data count register |
| 02B0 4018 | ICDRR | I2C data receive register |
| 02B0 401C | ICSAR | I2C slave address register |
| 02B0 4020 | ICDXR | I2C data transmit register |
| 02B0 4024 | ICMDR | I2C mode register |
| 02B0 4028 | ICIVR | I2C interrupt vector register |
| 02B0 402C | ICEMDR | I2C extended mode register |
| 02B0 4030 | ICPSC | I2C prescaler register |
| 02B0 4034 | ICPID1 | I2C peripheral identification register 1 [Value: 0x0000 0105] |
| 02B0 4038 | ICPID2 | I2C peripheral identification register 2 [Value: 0x0000 0005] |
| 02B0 403C - 02B0 405C | - | Reserved |
| 02B0 4060 - 02B3 407F | - | Reserved |
| 02B0 4080 - 02B3 FFFF | - | Reserved |

### 7.11.3 I2C Electrical Data/Timing

**Table 7-52. Timing Requirements for I2C Timings(1)**

(see Figure 7-42)

| NO. | | | -720 -850 A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | STANDARD MODE | | FAST MODE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(SCL)}$ | Cycle time, SCL | 10 | | 2.5 | | µs |
| 2 | $t_{su(SCLH-SDAL)}$ | Setup time, SCL high before SDA low (for a repeated START condition) | 4.7 | | 0.6 | | µs |
| 3 | $t_{h(SCLL-SDAL)}$ | Hold time, SCL low after SDA low (for a START and a repeated START condition) | 4 | | 0.6 | | µs |
| 4 | $t_{w(SCLL)}$ | Pulse duration, SCL low | 4.7 | | 1.3 | | µs |
| 5 | $t_{w(SCLH)}$ | Pulse duration, SCL high | 4 | | 0.6 | | µs |
| 6 | $t_{su(SDAV-SDLH)}$ | Setup time, SDA valid before SCL high | 250 | | 100(2) | | ns |
| 7 | $t_{h(SDA-SDLL)}$ | Hold time, SDA valid after SCL low (For $I^2C$ bus™ devices) | 0(3) | | 0(3) | 0.9(4) | µs |
| 8 | $t_{w(SDAH)}$ | Pulse duration, SDA high between STOP and START conditions | 4.7 | | 1.3 | | µs |
| 9 | $t_{r(SDA)}$ | Rise time, SDA | | 1000 | 20 + 0.1$C_b$ (5) | 300 | ns |
| 10 | $t_{r(SCL)}$ | Rise time, SCL | | 1000 | 20 + 0.1$C_b$ (5) | 300 | ns |
| 11 | $t_{f(SDA)}$ | Fall time, SDA | | 300 | 20 + 0.1$C_b$ (5) | 300 | ns |
| 12 | $t_{f(SCL)}$ | Fall time, SCL | | 300 | 20 + 0.1$C_b$ (5) | 300 | ns |
| 13 | $t_{su(SCLH-SDAH)}$ | Setup time, SCL high before SDA high (for STOP condition) | 4 | | 0.6 | | µs |
| 14 | $t_{w(SP)}$ | Pulse duration, spike (must be suppressed) | | | 0 | 50 | ns |
| 15 | $C_b$ (5) | Capacitive load for each bus line | | 400 | | 400 | pF |

(1) The I2C pins SDA and SCL do not feature fail-safe I/O buffers. These pins could potentially draw current when the device is powered down.
(2) A Fast-mode $I^2C$-bus™ device can be used in a Standard-mode $I^2C$-bus™ system, but the requirement $t_{su(SDA-SCLH)}$ ≥250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line $t_r$ max + $t_{su(SDA-SCLH)}$ = 1000 + 250 = 1250 ns (according to the Standard-mode $I^2C$-Bus Specification) before the SCL line is released.
(3) A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the $V_{IHmin}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.
(4) The maximum $t_{h(SDA-SCLL)}$ has only to be met if the device does not stretch the low period [$t_{w(SCLL)}$] of the SCL signal.
(5) $C_b$ = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

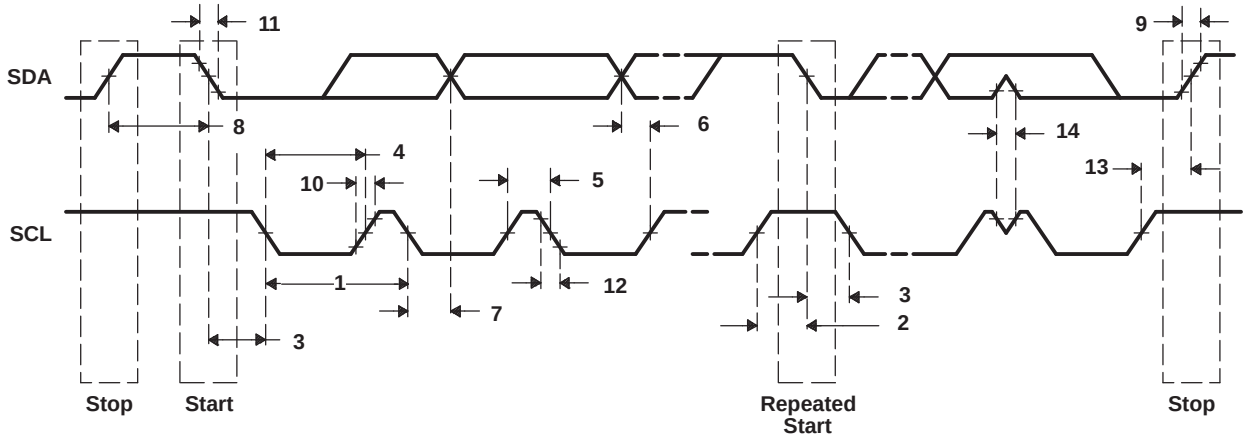



**Figure 7-42. I2C Receive Timings**

### Table 7-53. Switching Characteristics for I2C Timings[1]

(see Figure 7-43)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | STANDARD MODE | | FAST MODE | | |
| | | | MIN | MAX | MIN | MAX | |
| 16 | $t_{c(SCL)}$ | Cycle time, SCL | 10 | | 2.5 | | µs |
| 17 | $t_{d(SCLH-SDAL)}$ | Delay time, SCL high to SDA low (for a repeated START condition) | 4.7 | | 0.6 | | µs |
| 18 | $t_{d(SDAL-SCLL)}$ | Delay time, SDA low to SCL low (for a START and a repeated START condition) | 4 | | 0.6 | | µs |
| 19 | $t_{w(SCLL)}$ | Pulse duration, SCL low | 4.7 | | 1.3 | | µs |
| 20 | $t_{w(SCLH)}$ | Pulse duration, SCL high | 4 | | 0.6 | | µs |
| 21 | $t_{d(SDAV-SDLH)}$ | Delay time, SDA valid to SCL high | 250 | | 100 | | ns |
| 22 | $t_{v(SDLL-SDAV)}$ | Valid time, SDA valid after SCL low (for I2C bus devices) | 0 | | 0 | 0.9 | µs |
| 23 | $t_{w(SDAH)}$ | Pulse duration, SDA high between STOP and START conditions | 4.7 | | 1.3 | | µs |
| 24 | $t_{r(SDA)}$ | Rise time, SDA | | 1000 | $20 + 0.1C_b$ [2] | 300 | ns |
| 25 | $t_{r(SCL)}$ | Rise time, SCL | | 1000 | $20 + 0.1C_b$ [2] | 300 | ns |
| 26 | $t_{f(SDA)}$ | Fall time, SDA | | 300 | $20 + 0.1C_b$ [2] | 300 | ns |
| 27 | $t_{f(SCL)}$ | Fall time, SCL | | 300 | $20 + 0.1C_b$ [2] | 300 | ns |
| 28 | $t_{d(SCLH-SDAH)}$ | Delay time, SCL high to SDA high (for STOP condition) | 4 | | 0.6 | | µs |
| 29 | $C_p$ | Capacitance for each I2C pin | | 10 | | 10 | pF |

(1) $C_b$ = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.
(2) $C_b$ = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

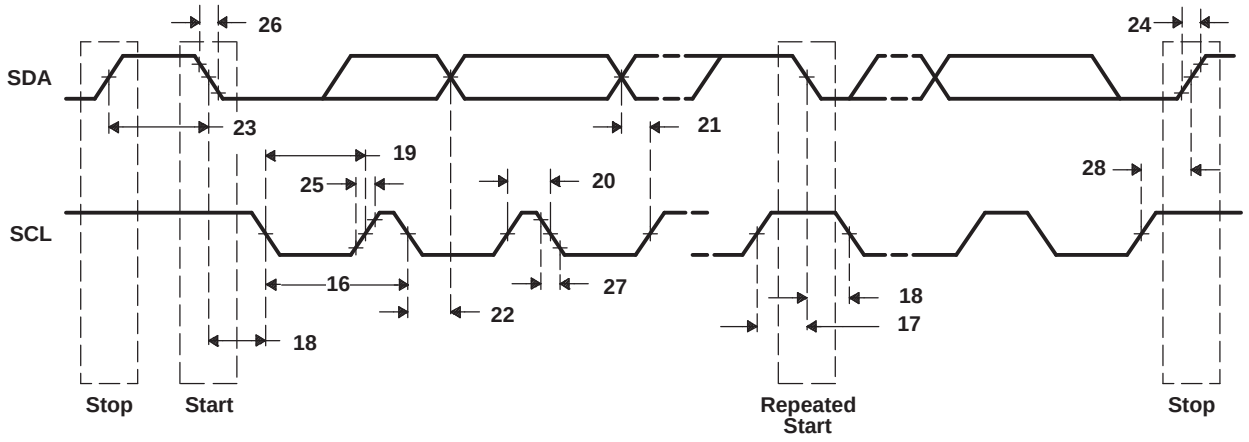



**Figure 7-43. I2C Transmit Timings**

## 7.12 Host-Port Interface (HPI) Peripheral

### 7.12.1 HPI Device-Specific Information

The C6454 device includes a user-configurable 16-bit or 32-bit Host-port interface (HPI16/HPI32). The AEA14 pin controls the HPI_WIDTH, allowing the user to configure the HPI as a 16-bit or 32-bit peripheral.

Software handshaking via the HRDY bit of the Host Port Control Register (HPIC) is not supported on the C6454 device.

An HPI boot is terminated using a DSP interrupt. The DSP interrupt is registered in bit 0 (channel 0) of the EDMA Event Register (ER). This event must be cleared by software before triggering transfers on DMA channel 0.

### 7.12.2 HPI Peripheral Register Descriptions

**Table 7-54. HPI Control Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 0288 0000 | - | Reserved | |
| 0288 0004 | PWREMU_MGMT | HPI power and emulation management register | The CPU has read/write access to the PWREMU_MGMT register; the Host does not have any access to this register. |
| 0288 0008 - 0288 0024 | - | Reserved | |
| 0288 0028 | - | Reserved | |
| 0288 002C | - | Reserved | |
| 0288 0030 | HPIC | HPI control register | The Host and the CPU have read/write access to the HPIC register.(1) |
| 0288 0034 | HPIA (HPIAW)(2) | HPI address register (Write) | The Host has read/write access to the HPIA registers. The CPU has only read access to the HPIA registers. |
| 0288 0038 | HPIA (HPIAR)(2) | HPI address register (Read) | |
| 0288 000C - 028B 007F | - | Reserved | |
| 0288 0080 - 028B FFFF | - | Reserved | |

(1) The CPU can write 1 to the $\overline{\text{HINT}}$ bit to generate an interrupt to the host and it can write 1 to the DSPINT bit to clear/acknowledge an interrupt from the host.

(2) There are two 32-bit HPIA registers: HPIAR for read operations and HPIAW for write operations. The HPI can be configured such that HPIAR and HPIAW act as a single 32-bit HPIA (single-HPIA mode) or as two separate 32-bit HPIAs (dual-HPIA mode) from the perspective of the host. The CPU can access HPIAW and HPIAR independently. For details about the HPIA registers and their modes, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

### 7.12.3 HPI Electrical Data/Timing

**Table 7-55. Timing Requirements for Host-Port Interface Cycles(1) (2)**

(see Figure 7-44 through Figure 7-51)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 9 | $t_{su(HASL-HSTBL)}$ | Setup time, $\overline{HAS}$ low before $\overline{HSTROBE}$ low | 5 | | ns |
| 10 | $t_{h(HSTBL-HASL)}$ | Hold time, $\overline{HAS}$ low after $\overline{HSTROBE}$ low | 2 | | ns |
| 11 | $t_{su(SELV-HASL)}$ | Setup time, select signals(3) valid before $\overline{HAS}$ low | 5 | | ns |
| 12 | $t_{h(HASL-SELV)}$ | Hold time, select signals(3) valid after $\overline{HAS}$ low | 5 | | ns |
| 13 | $t_{w(HSTBL)}$ | Pulse duration, $\overline{HSTROBE}$ low | 15 | | ns |
| 14 | $t_{w(HSTBH)}$ | Pulse duration, $\overline{HSTROBE}$ high between consecutive accesses | 2M | | ns |
| 15 | $t_{su(SELV-HSTBL)}$ | Setup time, select signals(3) valid before $\overline{HSTROBE}$ low | 5 | | ns |
| 16 | $t_{h(HSTBL-SELV)}$ | Hold time, select signals(3) valid after $\overline{HSTROBE}$ low | 5 | | ns |
| 17 | $t_{su(HDV-HSTBH)}$ | Setup time, host data valid before $\overline{HSTROBE}$ high | 5 | | ns |
| 18 | $t_{h(HSTBH-HDV)}$ | Hold time, host data valid after $\overline{HSTROBE}$ high | 1 | | ns |
| 37 | $t_{su(HCSL-HSTBL)}$ | Setup time, $\overline{HCS}$ low before $\overline{HSTROBE}$ low | 0 | | ns |
| 38 | $t_{h(HRDYL-HSTBL)}$ | Hold time, $\overline{HSTROBE}$ low after $\overline{HRDY}$ low. $\overline{HSTROBE}$ should not be inactivated until $\overline{HRDY}$ is active (low); otherwise, HPI writes will not complete properly. | 1.1 | | ns |

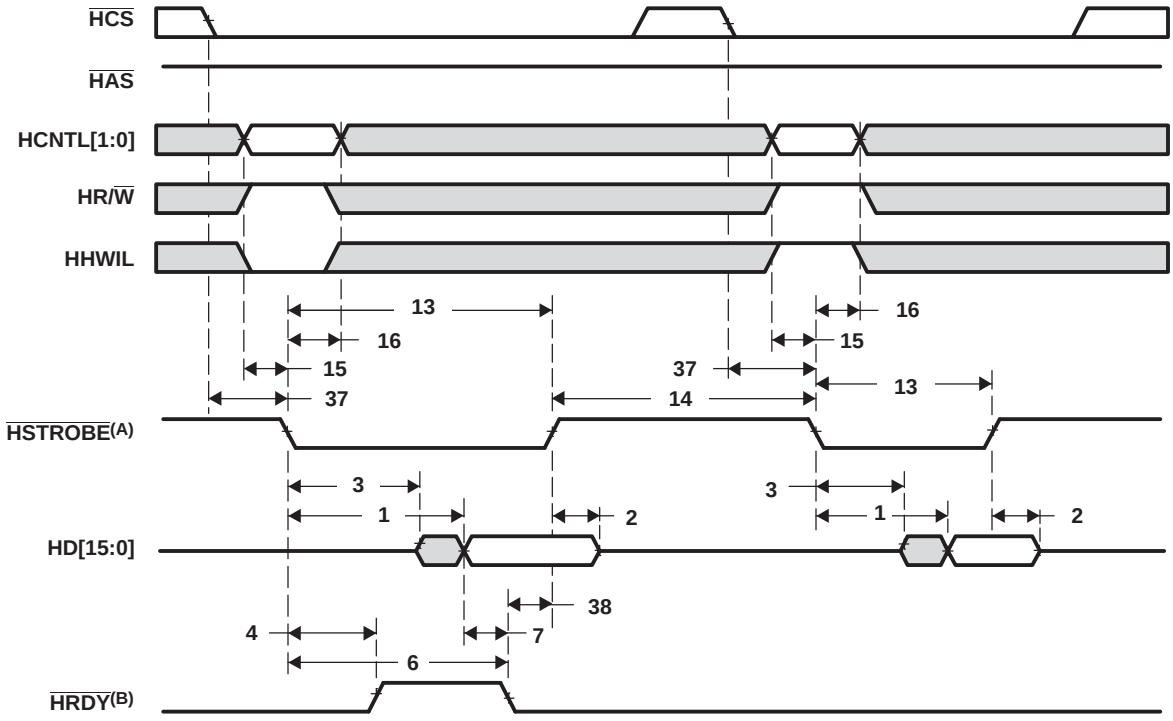
(1) $\overline{HSTROBE}$ refers to the following logical operation on $\overline{HCS}$, $\overline{HDS1}$, and $\overline{HDS2}$: [NOT($\overline{HDS1}$ XOR $\overline{HDS2}$)] OR $\overline{HCS}$.
(2) M = SYSCLK3 period = 6/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use M = 6 ns.
(3) Select signals include: HCNTL[1:0] and HR/$\overline{W}$. For HPI16 mode only, select signals also include HHWIL.

### Table 7-56. Switching Characteristics for Host-Port Interface Cycles(1) (2)

(see Figure 7-44 through Figure 7-51)

| NO. | PARAMETER | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 1 | $t_{d(HSTBL-HDV)}$ | Delay time, HSTROBE low to DSP data valid | Case 1. HPIC or HPIA read | 5 | 15 | ns |
| | | | Case 2. HPID read with no auto-increment(3) | | 9 * M + 20 | |
| | | | Case 3. HPID read with auto-increment and read FIFO initially empty(3) | | 9 * M + 20 | |
| | | | Case 4. HPID read with auto-increment and data previously prefetched into the read FIFO | 5 | 15 | |
| 2 | $t_{dis(HSTBH-HDV)}$ | Disable time, HD high-impedance from HSTROBE high | | 1 | 4 | ns |
| 3 | $t_{en(HSTBL-HD)}$ | Enable time, HD driven from HSTROBE low | | 3 | 15 | ns |
| 4 | $t_{d(HSTBL-HRDYH)}$ | Delay time, HSTROBE low to HRDY high | | | 12 | ns |
| 5 | $t_{d(HSTBH-HRDYH)}$ | Delay time, HSTROBE high to HRDY high | | | 12 | ns |
| 6 | $t_{d(HSTBL-HRDYL)}$ | Delay time, HSTROBE low to HRDY low | Case 1. HPID read with no auto-increment(3) | | 10 * M + 20 | ns |
| | | | Case 2. HPID read with auto-increment and read FIFO initially empty(3) | | 10 * M + 20 | |
| 7 | $t_{d(HDV-HRDYL)}$ | Delay time, HD valid to HRDY low | | 0 | | ns |
| 34 | $t_{d(DSH-HRDYL)}$ | Delay time, HSTROBE high to HRDY low | Case 1. HPIA write(3) | | 5 * M + 20 | ns |
| | | | Case 2. HPID write with no auto-increment(3) | | 5 * M + 20 | |
| 35 | $t_{d(HSTBL-HRDYL)}$ | Delay time, HSTROBE low to HRDY low for HPIA write and FIFO not empty(3) | | | 40 * M + 20 | ns |
| 36 | $t_{d(HASL-HRDYH)}$ | Delay time, HAS low to HRDY high | | | 12 | ns |

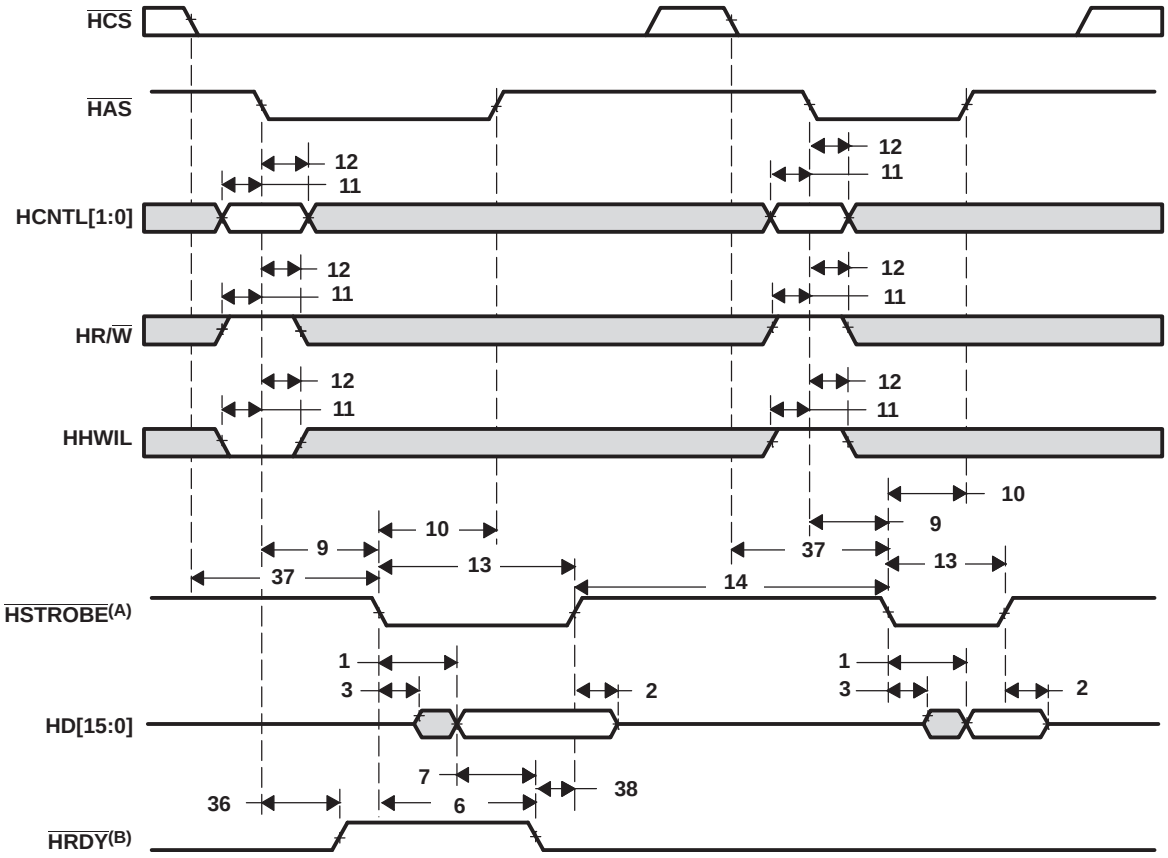
(1) M = SYSCLK3 period = 6/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use M = 6 ns.
(2) HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.
(3) Assumes the HPI is accessing L2/L1 memory and no other master is accessing the same memory location.




A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

B. Depending on the type of write or read operation (HPID without auto-incrementing; HPIA, HPIC, or HPID with auto-incrementing) and the state of the FIFO, transitions on HRDY may or may not occur. For more detailed information on the HPI peripheral, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

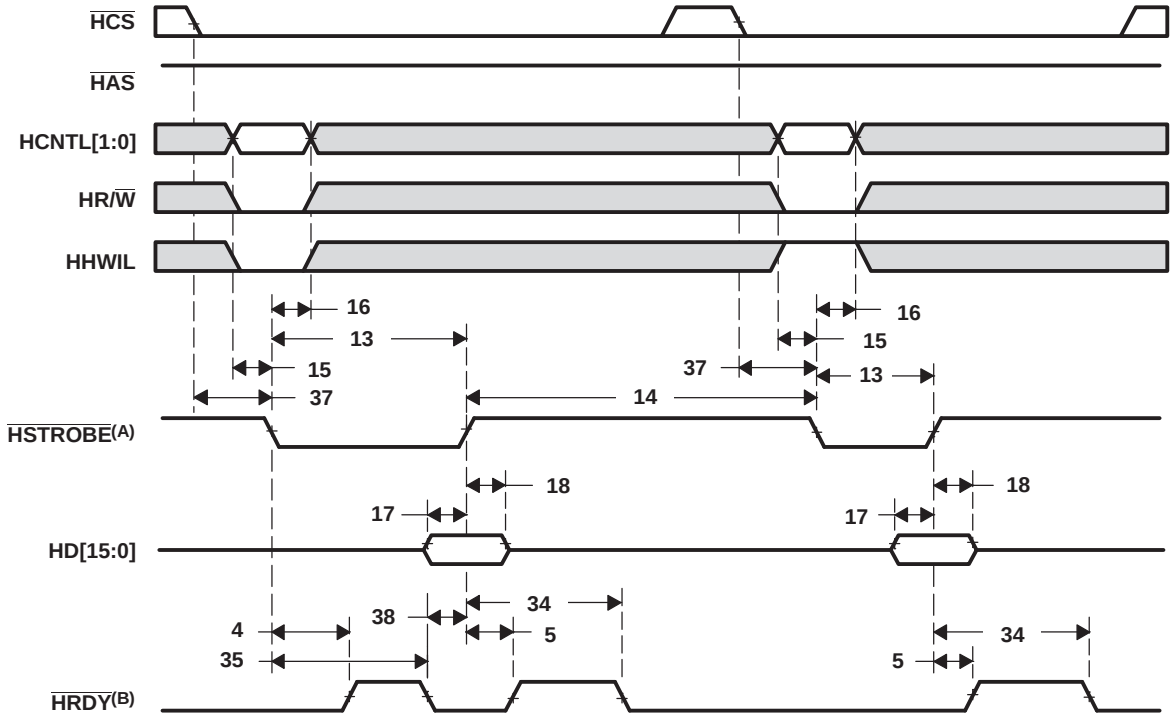
**Figure 7-44. HPI16 Read Timing (HAS Not Used, Tied High)**




A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

B. Depending on the type of write or read operation (HPID without auto-incrementing; HPIA, HPIC, or HPID with auto-incrementing) and the state of the FIFO, transitions on HRDY may or may not occur. For more detailed information on the HPI peripheral, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

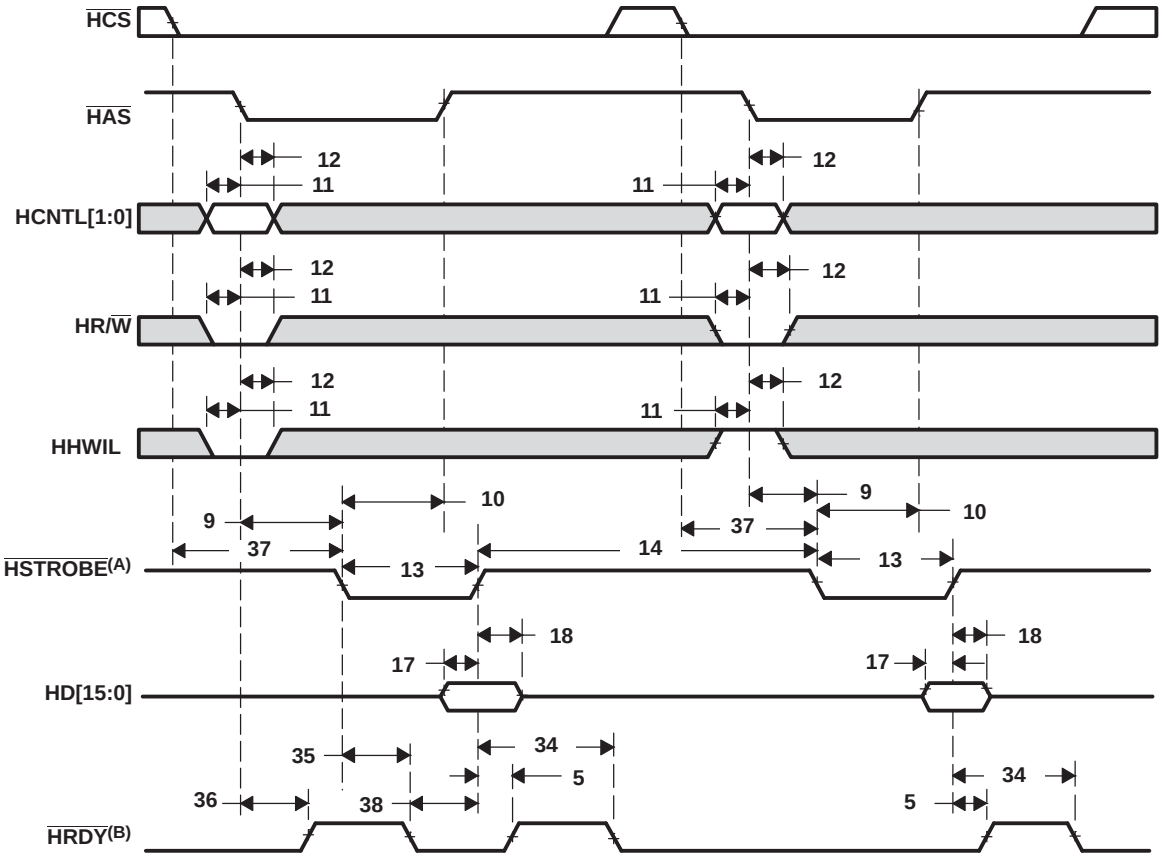
**Figure 7-45. HPI16 Read Timing (HAS Used)**




A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

B. Depending on the type of write or read operation (HPID without auto-incrementing; HPIA, HPIC, or HPID with auto-incrementing) and the state of the FIFO, transitions on HRDY may or may not occur. For more detailed information on the HPI peripheral, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

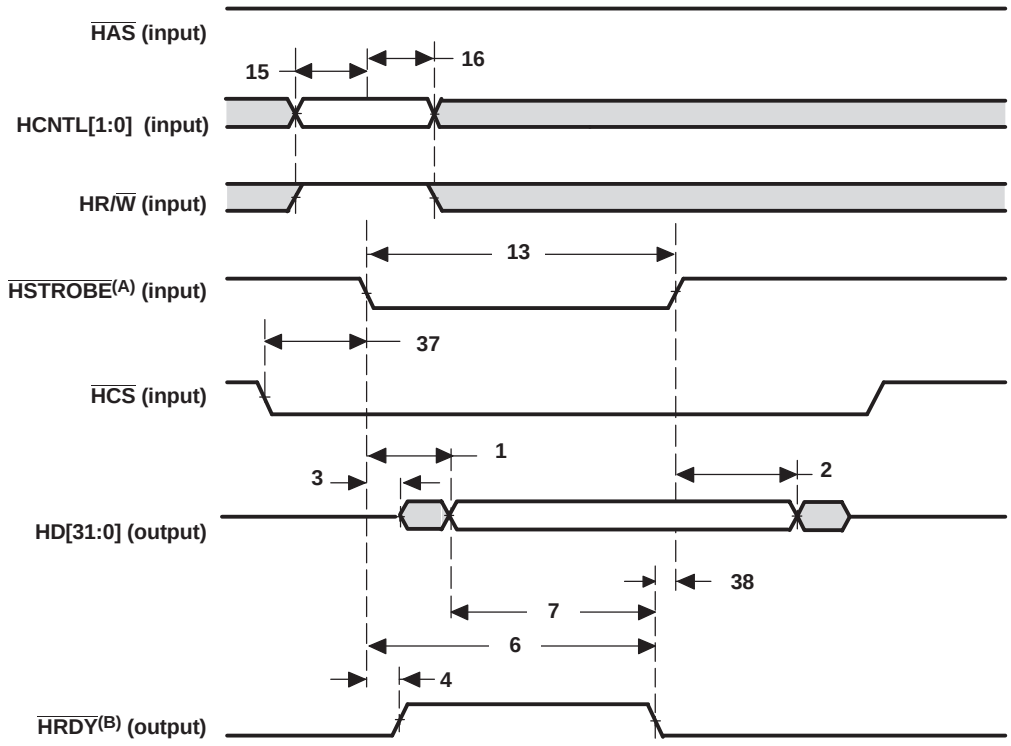
**Figure 7-46. HPI16 Write Timing (HAS Not Used, Tied High)**




A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

B. Depending on the type of write or read operation (HPID without auto-incrementing; HPIA, HPIC, or HPID with auto-incrementing) and the state of the FIFO, transitions on HRDY may or may not occur. For more detailed information on the HPI peripheral, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

**Figure 7-47. HPI16 Write Timing (HAS Used)**

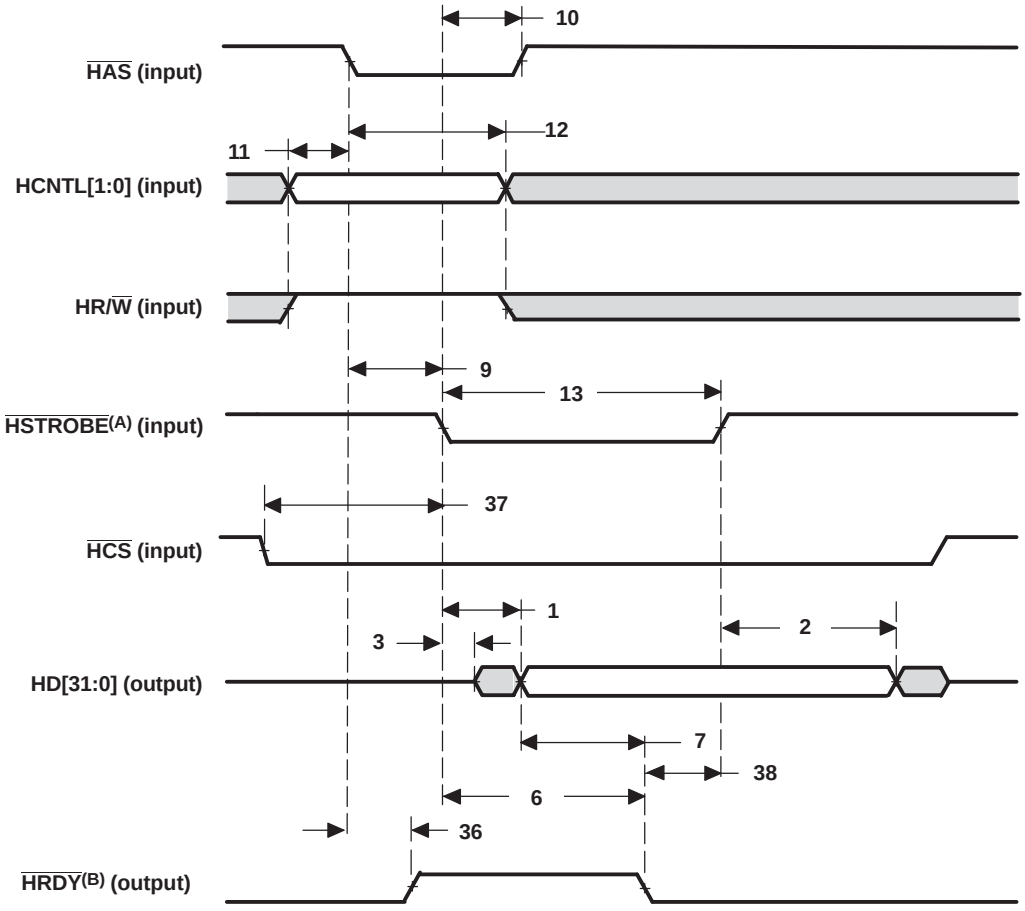



A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

B. Depending on the type of write or read operation (HPID without auto-incrementing; HPIA, HPIC, or HPID with auto-incrementing) and the state of the FIFO, transitions on HRDY may or may not occur. For more detailed information on the HPI peripheral, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

C. The timing $t_{w(HSTBH)}$, HSTROBE high pulse duration, must be met between consecutive HPI accesses in HPI32 mode.

**Figure 7-48. HPI32 Read Timing (HAS Not Used, Tied High)**

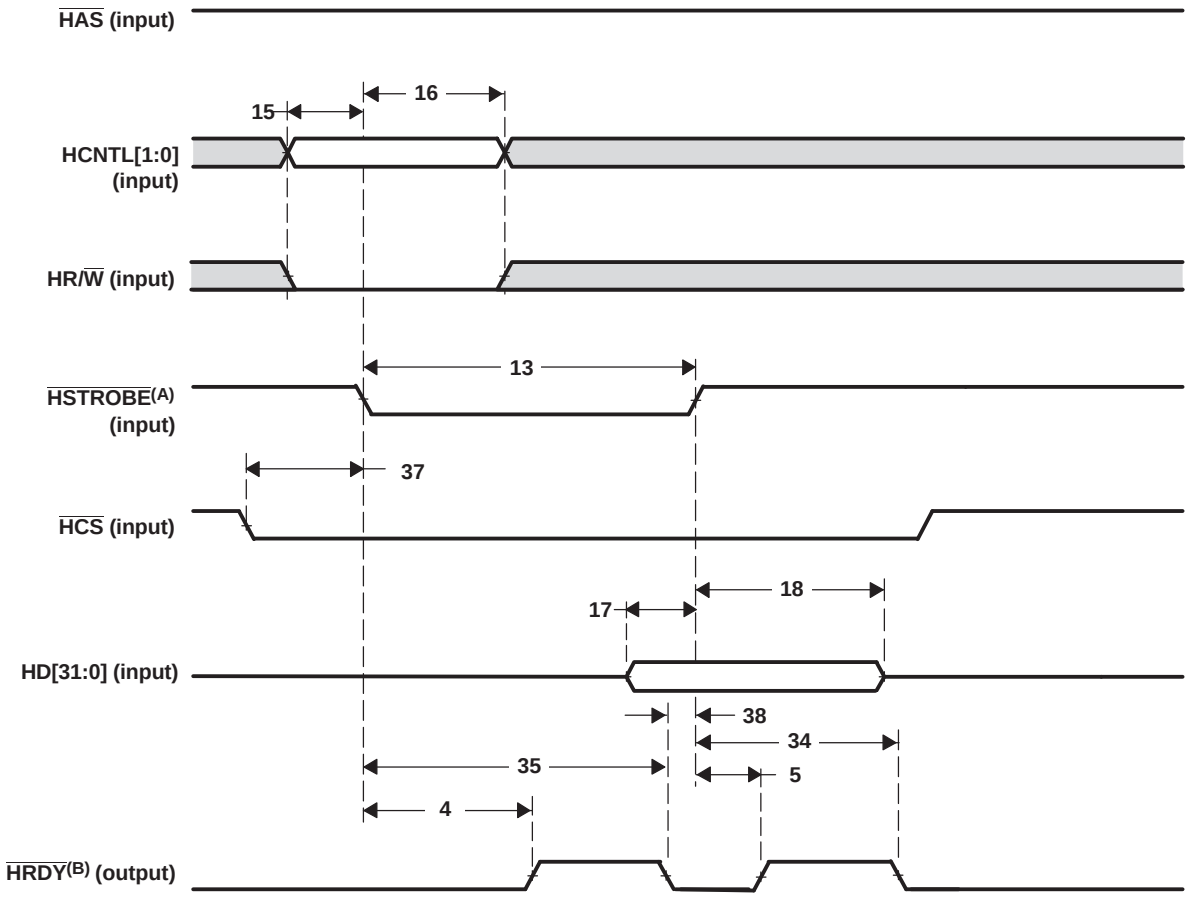



A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

B. Depending on the type of write or read operation (HPID without auto-incrementing; HPIA, HPIC, or HPID with auto-incrementing) and the state of the FIFO, transitions on HRDY may or may not occur. For more detailed information on the HPI peripheral, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

C. The timing $t_{w(HSTBH)}$, HSTROBE high pulse duration, must be met between consecutive HPI accesses in HPI32 mode.

**Figure 7-49. HPI32 Read Timing (HAS Used)**

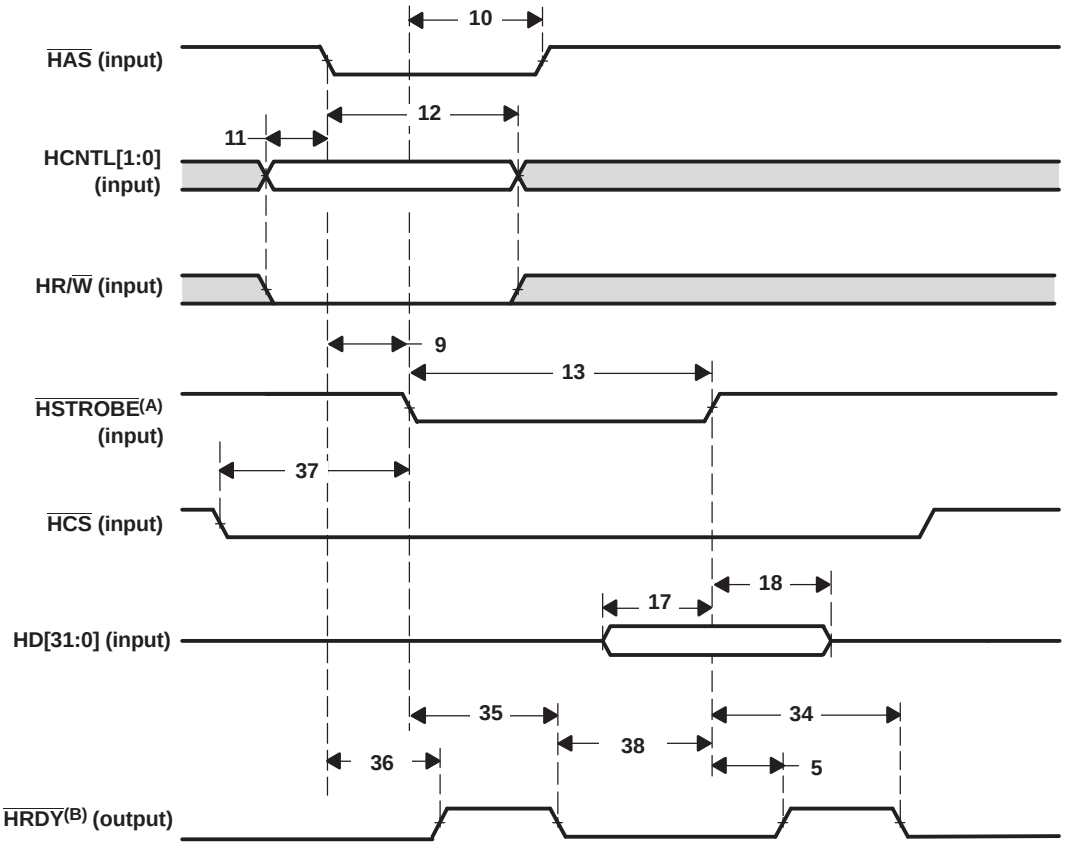



A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

B. Depending on the type of write or read operation (HPID without auto-incrementing; HPIA, HPIC, or HPID with auto-incrementing) and the state of the FIFO, transitions on HRDY may or may not occur. For more detailed information on the HPI peripheral, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

C. The timing $t_{w(HSTBH)}$, HSTROBE high pulse duration, must be met between consecutive HPI accesses in HPI32 mode.

**Figure 7-50. HPI32 Write Timing (HAS Not Used, Tied High)**




A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

B. Depending on the type of write or read operation (HPID without auto-incrementing; HPIA, HPIC, or HPID with auto-incrementing) and the state of the FIFO, transitions on HRDY may or may not occur. For more detailed information on the HPI peripheral, see the *TMS320C645x DSP Host Port Interface (HPI) User's Guide* (literature number SPRU969).

C. The timing $t_{w(HSTBH)}$, HSTROBE high pulse duration, must be met between consecutive HPI accesses in HPI32 mode.

**Figure 7-51. HPI32 Write Timing (HAS Used)**

## 7.13 Multichannel Buffered Serial Port (McBSP)

The McBSP provides these functions:

- Full-duplex communication
- Double-buffered data registers, which allow a continuous data stream
- Independent framing and clocking for receive and transmit
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected analog-to-digital (A/D) and digital-to-analog (D/A) devices
- External shift clock or an internal, programmable frequency shift clock for data transfer

For more detailed information on the McBSP peripheral, see the *TMS320C6000 DSP Multichannel Buffered Serial Port ( McBSP) Reference Guide* (literature number SPRU580).

### 7.13.1 McBSP Device-Specific Information

The CLKS signal is shared by both McBSP0 and McBSP1 on this device. Also, the CLKGDV field of the Sample Rate Generator Register (SRGR) must always be set to a value of 1 or greater.

The McBSP Data Receive Register (DRR) and Data Transmit Register (DXR) can be accessed through two separate busses: a configuration bus and a data bus. Both paths can be used by the CPU and the EDMA. The data bus should be used to service the McBSP as this path provides better performance. However, since the data path shares a bridge with the PCI peripheral (see Figure 4-1), the configuration path should be used in cases where this peripheral is being used to avoid any performance degradation. Note that the PCI peripheral consists of an independent master and slave. Performance degradation is only a concern when this peripheral is used to initiate transactions on the external bus.

### 7.13.2 McBSP Peripheral Register Descriptions

**Table 7-57. McBSP 0 Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 028C 0000 | DRR0 | McBSP0 Data Receive Register via Configuration Bus | The CPU and EDMA3 controller can only read this register; they cannot write to it. |
| 3000 0000 | DRR0 | McBSP0 Data Receive Register via EDMA3 Bus | |
| 028C 0004 | DXR0 | McBSP0 Data Transmit Register via Configuration Bus | |
| 3000 0010 | DXR0 | McBSP0 Data Transmit Register via EDMA Bus | |
| 028C 0008 | SPCR0 | McBSP0 Serial Port Control Register | |
| 028C 000C | RCR0 | McBSP0 Receive Control Register | |
| 028C 0010 | XCR0 | McBSP0 Transmit Control Register | |
| 028C 0014 | SRGR0 | McBSP0 Sample Rate Generator register | |
| 028C 0018 | MCR0 | McBSP0 Multichannel Control Register | |
| 028C 001C | RCERE00 | McBSP0 Enhanced Receive Channel Enable Register 0 Partition A/B | |
| 028C 0020 | XCERE00 | McBSP0 Enhanced Transmit Channel Enable Register 0 Partition A/B | |
| 028C 0024 | PCR0 | McBSP0 Pin Control Register | |
| 028C 0028 | RCERE10 | McBSP0 Enhanced Receive Channel Enable Register 1 Partition C/D | |
| 028C 002C | XCERE10 | McBSP0 Enhanced Transmit Channel Enable Register 1 Partition C/D | |
| 028C 0030 | RCERE20 | McBSP0 Enhanced Receive Channel Enable Register 2 Partition E/F | |
| 028C 0034 | XCERE20 | McBSP0 Enhanced Transmit Channel Enable Register 2 Partition E/F | |
| 028C 0038 | RCERE30 | McBSP0 Enhanced Receive Channel Enable Register 3 Partition G/H | |
| 028C 003C | XCERE30 | McBSP0 Enhanced Transmit Channel Enable Register 3 Partition G/H | |
| 028C 0040 - 028F FFFF | - | Reserved | |

**Table 7-58. McBSP 1 Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 0290 0000 | DRR1 | McBSP1 Data Receive Register via Configuration Bus | The CPU and EDMA controller can only read this register; they cannot write to it. |
| 3400 0000 | DRR1 | McBSP1 Data Receive Register via EDMA bus | |
| 0290 0004 | DXR1 | McBSP1 Data Transmit Register via configuration bus | |
| 3400 0010 | DXR1 | McBSP1 Data Transmit Register via EDMA bus | |
| 0290 0008 | SPCR1 | McBSP1 serial port control register | |
| 0290 000C | RCR1 | McBSP1 Receive Control Register | |
| 0290 0010 | XCR1 | McBSP1 Transmit Control Register | |
| 0290 0014 | SRGR1 | McBSP1 sample rate generator register | |
| 0290 0018 | MCR1 | McBSP1 multichannel control register | |
| 0290 001C | RCERE01 | McBSP1 Enhanced Receive Channel Enable Register 0 Partition A/B | |
| 0290 0020 | XCERE01 | McBSP1 Enhanced Transmit Channel Enable Register 0 Partition A/B | |
| 0290 0024 | PCR1 | McBSP1 Pin Control Register | |
| 0290 0028 | RCERE11 | McBSP1 Enhanced Receive Channel Enable Register 1 Partition C/D | |
| 0290 002C | XCERE11 | McBSP1 Enhanced Transmit Channel Enable Register 1 Partition C/D | |
| 0290 0030 | RCERE21 | McBSP1 Enhanced Receive Channel Enable Register 2 Partition E/F | |
| 0290 0034 | XCERE21 | McBSP1 Enhanced Transmit Channel Enable Register 2 Partition E/F | |
| 0290 0038 | RCERE31 | McBSP1 Enhanced Receive Channel Enable Register 3 Partition G/H | |
| 0290 003C | XCERE31 | McBSP1 Enhanced Transmit Channel Enable Register 3 Partition G/H | |
| 0290 0040 - 0293 FFFF | - | Reserved | |

### 7.13.3 McBSP Electrical Data/Timing

**Table 7-59. Timing Requirements for McBSP(1)**

(see Figure 7-52)

| NO. | | | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X ext | 6P or 10(2) (3) | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X ext | $0.5t_{c(CKRX)}$ - 1(4) | | ns |
| 5 | $t_{su(FRH-CKRL)}$ | Setup time, external FSR high before CLKR low | CLKR int | 9 | | ns |
| | | | CLKR ext | 1.3 | | |
| 6 | $t_{h(CKRL-FRH)}$ | Hold time, external FSR high after CLKR low | CLKR int | 6 | | ns |
| | | | CLKR ext | 3 | | |
| 7 | $t_{su(DRV-CKRL)}$ | Setup time, DR valid before CLKR low | CLKR int | 8 | | ns |
| | | | CLKR ext | 0.9 | | |
| 8 | $t_{h(CKRL-DRV)}$ | Hold time, DR valid after CLKR low | CLKR int | 3 | | ns |
| | | | CLKR ext | 3.1 | | |
| 10 | $t_{su(FXH-CKXL)}$ | Setup time, external FSX high before CLKX low | CLKX int | 9 | | ns |
| | | | CLKX ext | 1.3 | | |
| 11 | $t_{h(CKXL-FXH)}$ | Hold time, external FSX high after CLKX low | CLKX int | 6 | | ns |
| | | | CLKX ext | 3 | | |

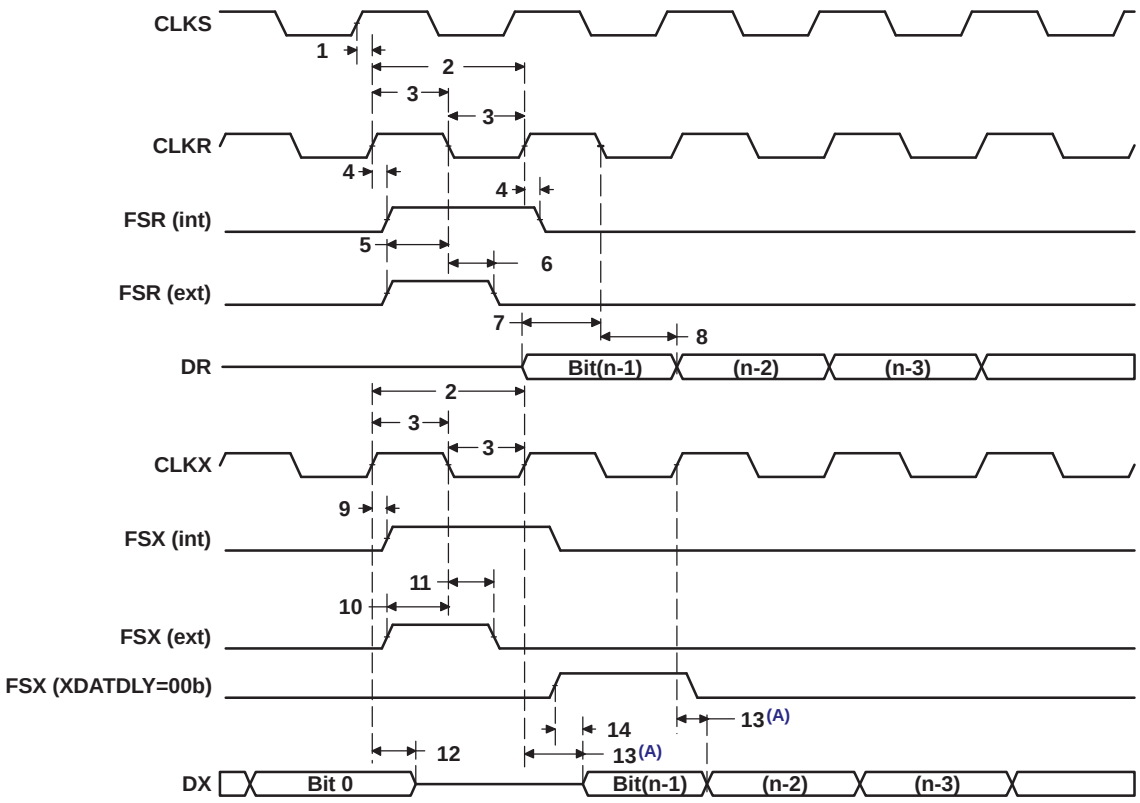
(1) CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
(2) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(3) Use whichever value is greater. Minimum CLKR/X cycle times *must* be met, even when CLKR/X is generated by an internal clock source. The minimum CLKR/X cycle times are based on internal logic speed; the maximum usable speed may be lower due to EDMA limitations and AC timing requirements.
(4) This parameter applies to the maximum McBSP frequency. Operate serial clocks (CLKR/X) in the reasonable range of 40/60 duty cycle.

### Table 7-60. Switching Characteristics Over Recommended Operating Conditions for McBSP[1] [2]

(see Figure 7-52)

| NO. | PARAMETER | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 1 | $t_{d(CKSH-CKRXH)}$ | Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input[3] | | 1.4 | 10 | ns |
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X int | 6P or 10[4] [5] [6] | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X int | C - 1[7] | C + 1[7] | ns |
| 4 | $t_{d(CKRH-FRV)}$ | Delay time, CLKR high to internal FSR valid | CLKR int | -2.1 | 3.3 | ns |
| 9 | $t_{d(CKXH-FXV)}$ | Delay time, CLKX high to internal FSX valid | CLKX int | -1.7 | 3 | ns |
| | | | CLKX ext | 1.7 | 9 | |
| 12 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | CLKX int | -3.9 | 4 | ns |
| | | | CLKX ext | 2.1 | 9 | |
| 13 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | CLKX int | -3.9 + D1[8] | 4 + D2[8] | ns |
| | | | CLKX ext | 2.1 + D1[8] | 9 + D2[8] | |
| 14 | $t_{d(FXH-DXV)}$ | Delay time, FSX high to DX valid<br>ONLY applies when in data delay 0 (XDATDLY = 00b) mode | FSX int | -2.3 + D1[9] | 5.6 + D2[9] | ns |
| | | | FSX ext | 1.9 + D1[9] | 9 + D2[9] | |

(1) CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
(2) Minimum delay times also represent minimum output hold times.
(3) The CLKS signal is shared by both McBSP0 and McBSP1 on this device.
(4) Minimum CLKR/X cycle times must be met, even when CLKR/X is generated by an internal clock source. Minimum CLKR/X cycle times are based on internal logic speed; the maximum usable speed may be lower due to EDMA limitations and AC timing requirements.
(5) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(6) Use whichever value is greater.
(7) C = H or L
S = sample rate generator input clock = 6P if CLKSM = 1 (P = 1/CPU clock frequency)
S = sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
H = (CLKGDV + 1)/2 * S if CLKGDV is odd
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
L = (CLKGDV + 1)/2 * S if CLKGDV is odd
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the maximum limit (see (4) above).
(8) Extra delay from CLKX high to DX valid **applies *only* to the first data bit of a device**, if and only if DXENA = 1 in SPCR.
if DXENA = 0, then D1 = D2 = 0
if DXENA = 1, then D1 = 6P, D2 = 12P
(9) Extra delay from FSX high to DX valid **applies *only* to the first data bit of a device**, if and only if DXENA = 1 in SPCR.
if DXENA = 0, then D1 = D2 = 0
if DXENA = 1, then D1 = 6P, D2 = 12P




A. Parameter No. 13 applies to the first data bit *only* when XDATDLY ≠ 0.

B. The CLKS signal is shared by both McBSP0 and McBSP1 on this device.

**Figure 7-52. McBSP Timing(B)**

**Table 7-61. Timing Requirements for FSR When GSYNC = 1**

(see Figure 7-53)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(FRH-CKSH)}$ | Setup time, FSR high before CLKS high | 4 | | ns |
| 2 | $t_{h(CKSH-FRH)}$ | Hold time, FSR high after CLKS high | 4 | | ns |

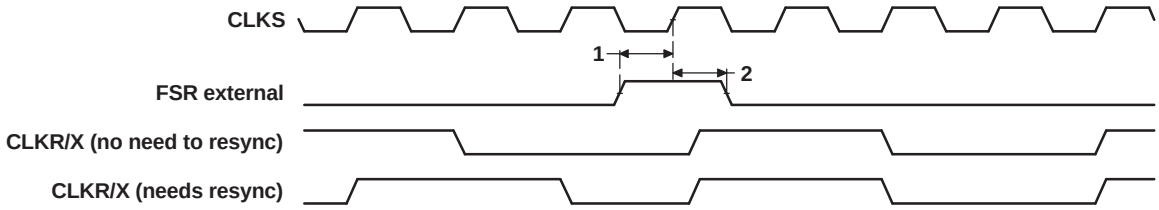



**Figure 7-53. FSR Timing When GSYNC = 1**

**Table 7-62. Timing Requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0 (1) (2)**

(see Figure 7-54)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXL)}$ | Setup time, DR valid before CLKX low | 12 | | 2 - 18P | | ns |
| 5 | $t_{h(CKXL-DRV)}$ | Hold time, DR valid after CLKX low | 4 | | 5 + 36P | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) For all SPI Slave modes, CLKG is programmed as 1/6 of the CPU clock by setting CLKSM = CLKGDV = 1.

**Table 7-63. Switching Characteristics Over Recommended Operating Conditions for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0(1) (2)**

(see Figure 7-54)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER(3) | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low(4) | T - 2 | T + 3 | | | ns |
| 2 | $t_{d(FXL-CKXH)}$ | Delay time, FSX low to CLKX high(5) | L - 2 | L + 3 | | | ns |
| 3 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | -2 | 4 | 18P + 2.8 | 30P + 17 | ns |
| 6 | $t_{dis(CKXL-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX low | L - 2 | L + 3 | | | ns |
| 7 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | | | 6P + 3 | 18P + 17 | ns |
| 8 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | 12P + 2 | 24P + 17 | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) For all SPI Slave modes, CLKG is programmed as 1/6 of the CPU clock by setting CLKSM = CLKGDV = 1.
(3) S = Sample rate generator input clock = 6P if CLKSM = 1 (P = 1/CPU clock frequency)
S = Sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
H = (CLKGDV + 1)/2 * S if CLKGDV is odd
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
L = (CLKGDV + 1)/2 * S if CLKGDV is odd
(4) FSRP = FSXP = 1. As a SPI Master, FSX is inverted to provide active-low slave-enable output. As a Slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for Master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for Slave McBSP
(5) FSX should be low before the rising edge of clock to enable Slave devices and then begin a SPI transfer at the rising edge of the Master clock (CLKX).

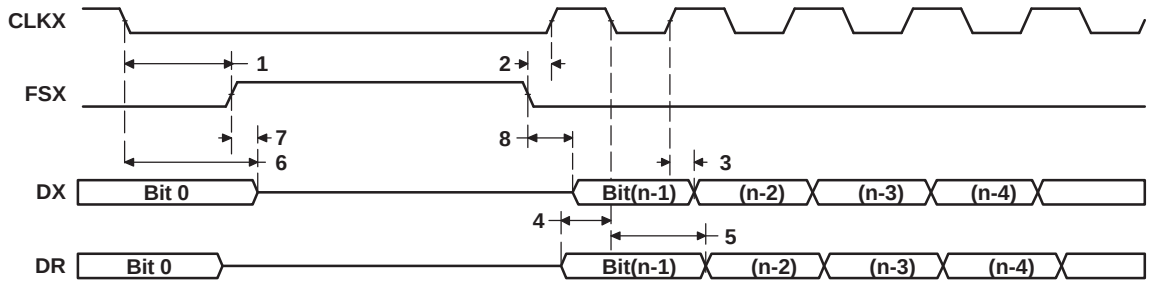



**Figure 7-54. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0**

**Table 7-64. Timing Requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0 (1) (2)**

(see Figure 7-55)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXH)}$ | Setup time, DR valid before CLKX high | 12 | | 2 - 18P | | ns |
| 5 | $t_{h(CKXH-DRV)}$ | Hold time, DR valid after CLKX high | 4 | | 5 + 36P | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) For all SPI Slave modes, CLKG is programmed as 1/6 of the CPU clock by setting CLKSM = CLKGDV = 1.

**Table 7-65. Switching Characteristics Over Recommended Operating Conditions for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0[1] [2]**

(see Figure 7-55)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER[3] | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low[4] | L - 2 | L + 3 | | | ns |
| 2 | $t_{d(FXL-CKXH)}$ | Delay time, FSX low to CLKX high[5] | T - 2 | T + 3 | | | ns |
| 3 | $t_{d(CKXL-DXV)}$ | Delay time, CLKX low to DX valid | -2 | 4 | 18P + 2.8 | 30P + 17 | ns |
| 6 | $t_{dis(CKXL-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX low | -2 | 4 | 18P + 3 | 30P + 17 | ns |
| 7 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | H - 2 | H + 4 | 12P + 2 | 24P + 17 | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) For all SPI Slave modes, CLKG is programmed as 1/6 of the CPU clock by setting CLKSM = CLKGDV = 1.
(3) S = Sample rate generator input clock = 6P if CLKSM = 1 (P = 1/CPU clock frequency)
S = Sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
H = (CLKGDV + 1)/2 * S if CLKGDV is odd
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
L = (CLKGDV + 1)/2 * S if CLKGDV is odd
(4) FSRP = FSXP = 1. As a SPI Master, FSX is inverted to provide active-low slave-enable output. As a Slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for Master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for Slave McBSP
(5) FSX should be low before the rising edge of clock to enable Slave devices and then begin a SPI transfer at the rising edge of the Master clock (CLKX).

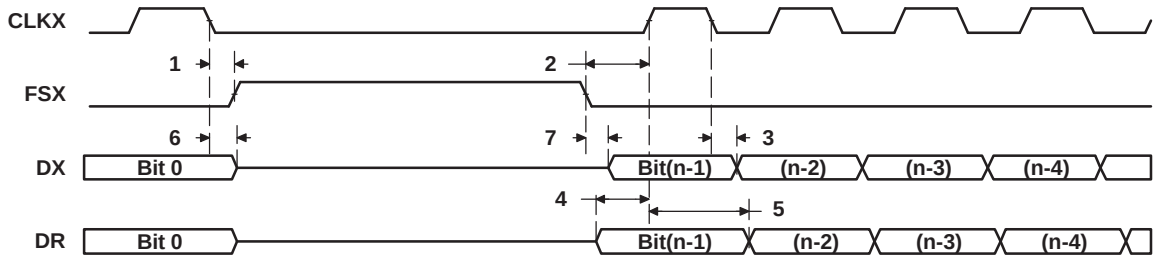



**Figure 7-55. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0**

**Table 7-66. Timing Requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1[1] [2]**

(see Figure 7-56)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXH)}$ | Setup time, DR valid before CLKX high | 12 | | 2 - 18P | | ns |
| 5 | $t_{h(CKXH-DRV)}$ | Hold time, DR valid after CLKX high | 4 | | 5 + 36P | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) For all SPI Slave modes, CLKG is programmed as 1/6 of the CPU clock by setting CLKSM = CLKGDV = 1.

**Table 7-67. Switching Characteristics Over Recommended Operating Conditions for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1(1) (2)**

(see Figure 7-56)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER(3) | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high(4) | T - 2 | T + 3 | | | ns |
| 2 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low(5) | H - 2 | H + 3 | | | ns |
| 3 | $t_{d(CKXL-DXV)}$ | Delay time, CLKX low to DX valid | -2 | 4 | 18P + 2.8 | 30P + 17 | ns |
| 6 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | H - 2 | H + 3 | | | ns |
| 7 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | | | 6P + 3 | 18P + 17 | ns |
| 8 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | 12P + 2 | 24P + 17 | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) For all SPI Slave modes, CLKG is programmed as 1/6 of the CPU clock by setting CLKSM = CLKGDV = 1.
(3) S = Sample rate generator input clock = 6P if CLKSM = 1 (P = 1/CPU clock frequency)
S = Sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
H = (CLKGDV + 1)/2 * S if CLKGDV is odd
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
L = (CLKGDV + 1)/2 * S if CLKGDV is odd
(4) FSRP = FSXP = 1. As a SPI Master, FSX is inverted to provide active-low slave-enable output. As a Slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for Master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for Slave McBSP
(5) FSX should be low before the rising edge of clock to enable Slave devices and then begin a SPI transfer at the rising edge of the Master clock (CLKX).

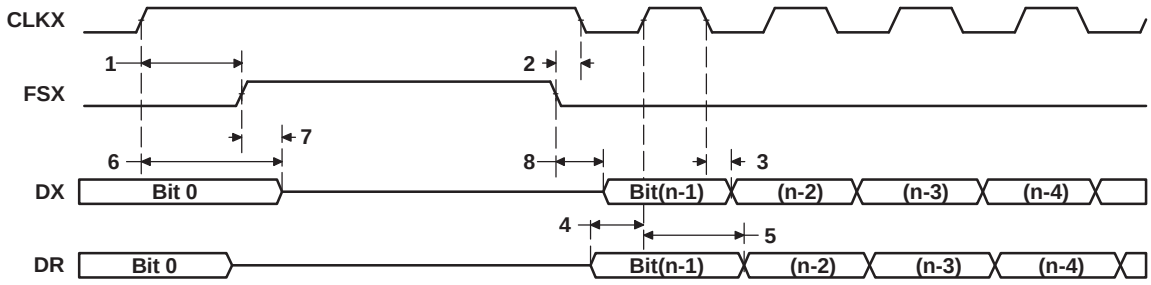



**Figure 7-56. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1**

**Table 7-68. Timing Requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1[1] [2]**

(see Figure 7-57)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXH)}$ | Setup time, DR valid before CLKX high | 12 | | 2 - 18P | | ns |
| 5 | $t_{h(CKXH-DRV)}$ | Hold time, DR valid after CLKX high | 4 | | 5 + 36P | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) For all SPI Slave modes, CLKG is programmed as 1/6 of the CPU clock by setting CLKSM = CLKGDV = 1.

**Table 7-69. Switching Characteristics Over Recommended Operating Conditions for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1[1] [2]**

(see Figure 7-57)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER[3] | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high[4] | H - 2 | H + 3 | | | ns |
| 2 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low[5] | T - 2 | T + 1 | | | ns |
| 3 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | -2 | 4 | 18P + 2.8 | 30P + 17 | ns |
| 6 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | -2 | 4 | 18P + 3 | 30P + 17 | ns |
| 7 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | L - 2 | L + 4 | 12P + 2 | 24P + 17 | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) For all SPI Slave modes, CLKG is programmed as 1/6 of the CPU clock by setting CLKSM = CLKGDV = 1.
(3) S = Sample rate generator input clock = 6P if CLKSM = 1 (P = 1/CPU clock frequency)
S = Sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
H = (CLKGDV + 1)/2 * S if CLKGDV is odd
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
L = (CLKGDV + 1)/2 * S if CLKGDV is odd
(4) FSRP = FSXP = 1. As a SPI Master, FSX is inverted to provide active-low slave-enable output. As a Slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for Master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for Slave McBSP
(5) FSX should be low before the rising edge of clock to enable Slave devices and then begin a SPI transfer at the rising edge of the Master clock (CLKX).

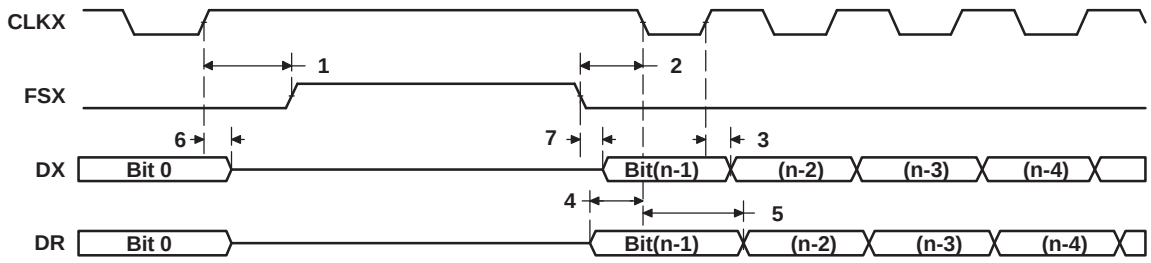



**Figure 7-57. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1**

## 7.14 Ethernet MAC (EMAC)

The Ethernet Media Access Controller (EMAC) module provides an efficient interface between the C6454 DSP core processor and the networked community. The EMAC supports 10Base-T (10 Mbits/second [Mbps]), and 100BaseTX (100 Mbps), in either half- or full-duplex mode, and 1000BaseT (1000 Mbps) in full-duplex mode, with hardware flow control and quality-of-service (QOS) support.

The EMAC module conforms to the IEEE 802.3-2002 standard, describing the “Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer” specifications. The IEEE 802.3 standard has also been adopted by ISO/IEC and re-designated as ISO/IEC 8802-3:2000(E).

Deviation from this standard, the EMAC module does not use the Transmit Coding Error signal MTXER. Instead of driving the error pin when an underflow condition occurs on a transmitted frame, the EMAC will intentionally generate an incorrect checksum by inverting the frame CRC, so that the transmitted frame will be detected as an error by the network.

The EMAC control module is the main interface between the device core processor, the MDIO module, and the EMAC module. The relationship between these three components is shown in Figure 7-58. The EMAC control module contains the necessary components to allow the EMAC to make efficient use of device memory, plus it controls device interrupts. The EMAC control module incorporates 8K-bytes of internal RAM to hold EMAC buffer descriptors. The relationship between these three components is shown in Figure 7-58.

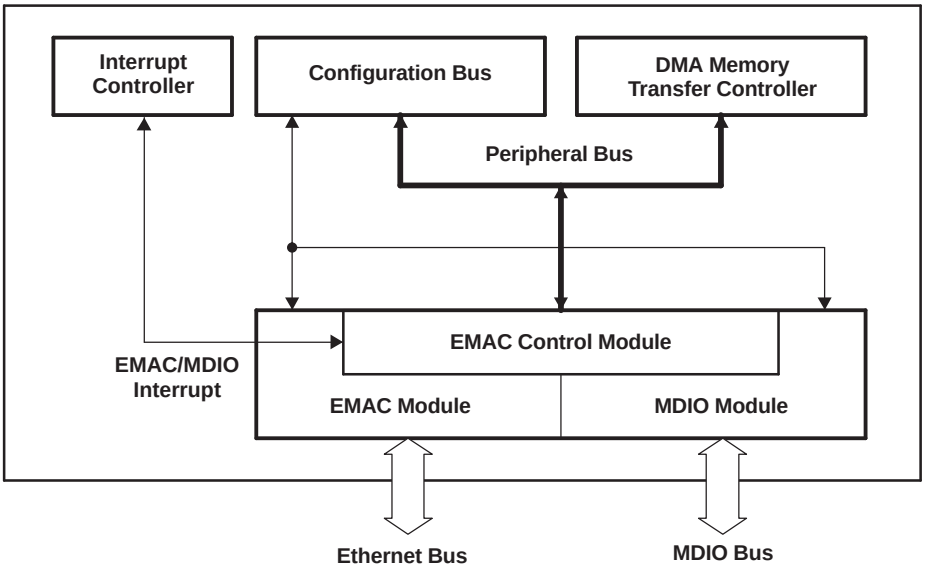



**Figure 7-58. EMAC, MDIO, and EMAC Control Modules**

For more detailed information on the EMAC/MDIO, see the *TMS320C645x DSP EMAC/MDIO Module Reference Guide* (literature number SPRU975).

### 7.14.1 EMAC Device-Specific Information

#### Interface Modes

The EMAC module on the TMS320C6454 device supports four interface modes: Media Independent Interface (MII), Reduced Media Independent Interface (RMII), Gigabit Media Independent Interface (GMII), and Reduced Gigabit Media Independent Interface (RGMII). The MII and GMII interface modes are defined in the IEEE 802.3-2002 standard.

The RGMII mode of the EMAC conforms to the Reduced Gigabit Media Independent Interface (RGMII) Specification (version 2.0). The RGMII mode implements the same functionality as the GMII mode, but with a reduced number of pins. Data and control information is transmitted and received using both edges of the transmit and receive clocks (TXC and RXC).

**Note:** The EMAC internally delays the transmit clock (TXC) with respect to the transmit data and control pins. Therefore, the EMAC conforms to the RGMII-ID operation of the RGMII specification. However, the EMAC does not delay the receive clock (RXC); this signal must be delayed with respect to the receive data and control pins outside of the DSP.

The RMII mode of the EMAC conforms to the RMII Specification (revision 1.2), as written by the RMII Consortium. As the name implies, the Reduced Media Independent Interface (RMII) mode is a reduced pin count version of the MII mode.

#### Interface Mode Select

The EMAC uses the same pins for the MII, GMII, and RMII modes. Standalone pins are included for the RGMII mode due to specific voltage requirements. Only one mode can be used at a time. The mode used is selected at device reset based on the MACSEL[1:0] configuration pins (for more detailed information, see Section 3, *Device Configuration*). Table 7-70 shows which multiplexed pins are used in the MII, GMII, and RMII modes on the EMAC. For a detailed description of these pin functions, see Table 2-3, *Terminal Functions*.

**Table 7-70. EMAC/MDIO Multiplexed Pins (MII, RMII, and GMII Modes)**

| BALL NUMBER | DEVICE PIN NAME | MII (MAC_SEL = 00b) | RMII (MAC_SEL = 01b) | GMII (MAC_SEL = 10b) |
|---|---|---|---|---|
| J2 | MRXD0/RMRXD0 | MRXD0 | RMRXD0 | MRXD0 |
| H3 | MRXD1/RMRXD1 | MRXD1 | RMRXD1 | MRXD1 |
| J1 | MRXD2 | MRXD2 | | MRXD2 |
| J3 | MRXD3 | MRXD3 | | MRXD3 |
| L1 | MRXD4 | | | MRXD4 |
| L2 | MRXD5 | | | MRXD5 |
| H2 | MRXD6 | | | MRXD6 |
| M2 | MRXD7 | | | MRXD7 |
| | | | | |
| M1 | MTXD0/RMTXD0 | MTXD0 | RMTXD0 | MTXD0 |
| L4 | MTXD1/RMTXD1 | MTXD1 | RMTXD1 | MTXD1 |
| M4 | MTXD2 | MTXD2 | | MTXD2 |
| K4 | MTXD3 | MTXD3 | | MTXD3 |
| L3 | MTXD4 | | | MTXD4 |
| L5 | MTXD5 | | | MTXD5 |
| M3 | MTXD6 | | | MTXD6 |
| N5 | MTXD7 | | | MTXD7 |
| | | | | |
| H4 | MRXER/RMRXER | MRXER | RMRXER | MRXER |
| H5 | MRXDV | MRXDV | | MRXDV |
| J5 | MTXEN/RMTXEN | MTXEN | RMTXEN | MTXEN |
| J4 | MCRS/RMCRSDV | MCRS | RMCRSDV | MCRS |
| K3 | MCOL | MCOL | | MCOL |
| | | | | |
| K5 | GMTCLK | | | GMTCLK |
| H1 | MRCLK | MRCLK | | MRCLK |
| N4 | MTCLK/REFCLK | MTCLK | RMREFCLK | MTCLK |
| | | | | |
| N3 | GMDIO | MDIO | MDIO | MDIO |
| M5 | GMDCLK | MDCLK | MDCLK | MDCLK |

### *Using the RMII Mode of the EMAC*

The Ethernet Media Access Controller (EMAC) contains logic that allows it to communicate using the Reduced Media Independent Interface (RMII) protocol. This logic must be taken out of reset before being used. To use the RMII mode of the EMAC follow these steps:

1. Enable the EMAC/MDIO through the Device State Control Registers.
   - Unlock the PERCFG0 register by writing 0x0F0A 0B00 to the PERLOCK register.
   - Set bit 4 in the PERCFG0 register within 16 SYSCLK3 clock cycles to enable the EMAC/MDIO.
   - Poll the PERSTAT0 register to verify state change.
2. Initialize the EMAC/MDIO as needed.
3. Release the RMII logic from reset by clearing the RMII_RST bit of the EMAC Configuration Register (see Section 3.4.5).

As described in the previous section, the RMII mode of the EMAC must be selected by setting MACSEL[1:0] = 01b at device reset.

### *Interface Mode Clocking*

The on-chip PLL2 and PLL2 Controller generate the clocks to the EMAC module in RGMII or GMII mode. When the EMAC is enabled with these modes, the input clock to the PLL2 Controller (CLKIN2) must have a 25-MHz frequency. For more information, see Section 7.8, *PLL2 and PLL2 Controller*.

The EMAC uses SYSCLK1 of the PLL2 Controller to generate the necessary clocks for the GMII and RGMII modes. When these modes are used, the frequency of CLKIN2 must be 25 MHz. Also, divider D1 should be programmed to ÷2 mode [default] when using the GMII mode and to ÷5 mode when using the RGMII mode. Divider D1 is software programmable and, if necessary, must be programmed after device reset to ÷5 when the RGMII mode of the EMAC is used.

### 7.14.2 EMAC Peripheral Register Descriptions

**Table 7-71. Ethernet MAC (EMAC) Control Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02C8 0000 | TXIDVER | Transmit Identification and Version Register |
| 02C8 0004 | TXCONTROL | Transmit Control Register |
| 02C8 0008 | TXTEARDOWN | Transmit Teardown Register |
| 02C8 000F | - | Reserved |
| 02C8 0010 | RXIDVER | Receive Identification and Version Register |
| 02C8 0014 | RXCONTROL | Receive Control Register |
| 02C8 0018 | RXTEARDOWN | Receive Teardown Register |
| 02C8 001C | - | Reserved |
| 02C8 0020 - 02C8 007C | - | Reserved |
| 02C8 0080 | TXINTSTATRAW | Transmit Interrupt Status (Unmasked) Register |
| 02C8 0084 | TXINTSTATMASKED | Transmit Interrupt Status (Masked) Register |
| 02C8 0088 | TXINTMASKSET | Transmit Interrupt Mask Set Register |
| 02C8 008C | TXINTMASKCLEAR | Transmit Interrupt Mask Clear Register |
| 02C8 0090 | MACINVECTOR | MAC Input Vector Register |
| 02C8 0094 - 02C8 009C | - | Reserved |
| 02C8 00A0 | RXINTSTATRAW | Receive Interrupt Status (Unmasked) Register |
| 02C8 00A4 | RXINTSTATMASKED | Receive Interrupt Status (Masked) Register |
| 02C8 00A8 | RXINTMASKSET | Receive Interrupt Mask Set Register |
| 02C8 00AC | RXINTMASKCLEAR | Receive Interrupt Mask Clear Register |
| 02C8 00B0 | MACINTSTATRAW | MAC Interrupt Status (Unmasked) Register |
| 02C8 00B4 | MACINTSTATMASKED | MAC Interrupt Status (Masked) Register |
| 02C8 00B8 | MACINTMASKSET | MAC Interrupt Mask Set Register |
| 02C8 00BC | MACINTMASKCLEAR | MAC Interrupt Mask Clear Register |
| 02C8 00C0 - 02C8 00FC | - | Reserved |
| 02C8 0100 | RXMBPENABLE | Receive Multicast/Broadcast/Promiscuous Channel Enable Register |
| 02C8 0104 | RXUNICASTSET | Receive Unicast Enable Set Register |
| 02C8 0108 | RXUNICASTCLEAR | Receive Unicast Clear Register |
| 02C8 010C | RXMAXLEN | Receive Maximum Length Register |
| 02C8 0110 | RXBUFFEROFFSET | Receive Buffer Offset Register |
| 02C8 0114 | RXFILTERLOWTHRESH | Receive Filter Low Priority Frame Threshold Register |
| 02C8 0118 - 02C8 011C | - | Reserved |
| 02C8 0120 | RX0FLOWTHRESH | Receive Channel 0 Flow Control Threshold Register |
| 02C8 0124 | RX1FLOWTHRESH | Receive Channel 1 Flow Control Threshold Register |
| 02C8 0128 | RX2FLOWTHRESH | Receive Channel 2 Flow Control Threshold Register |
| 02C8 012C | RX3FLOWTHRESH | Receive Channel 3 Flow Control Threshold Register |
| 02C8 0130 | RX4FLOWTHRESH | Receive Channel 4 Flow Control Threshold Register |
| 02C8 0134 | RX5FLOWTHRESH | Receive Channel 5 Flow Control Threshold Register |
| 02C8 0138 | RX6FLOWTHRESH | Receive Channel 6 Flow Control Threshold Register |
| 02C8 013C | RX7FLOWTHRESH | Receive Channel 7 Flow Control Threshold Register |
| 02C8 0140 | RX0FREEBUFFER | Receive Channel 0 Free Buffer Count Register |
| 02C8 0144 | RX1FREEBUFFER | Receive Channel 1 Free Buffer Count Register |
| 02C8 0148 | RX2FREEBUFFER | Receive Channel 2 Free Buffer Count Register |
| 02C8 014C | RX3FREEBUFFER | Receive Channel 3 Free Buffer Count Register |
| 02C8 0150 | RX4FREEBUFFER | Receive Channel 4 Free Buffer Count Register |
| 02C8 0154 | RX5FREEBUFFER | Receive Channel 5 Free Buffer Count Register |

**Table 7-71. Ethernet MAC (EMAC) Control Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02C8 0158 | RX6FREEBUFFER | Receive Channel 6 Free Buffer Count Register |
| 02C8 015C | RX7FREEBUFFER | Receive Channel 7 Free Buffer Count Register |
| 02C8 0160 | MACCONTROL | MAC Control Register |
| 02C8 0164 | MACSTATUS | MAC Status Register |
| 02C8 0168 | EMCONTROL | Emulation Control Register |
| 02C8 016C | FIFOCONTROL | FIFO Control Register (Transmit and Receive) |
| 02C8 0170 | MACCONFIG | MAC Configuration Register |
| 02C8 0174 | SOFTRESET | Soft Reset Register |
| 02C8 0178 - 02C8 01CC | - | Reserved |
| 02C8 01D0 | MACSRCADDRLO | MAC Source Address Low Bytes Register (Lower 32-bits) |
| 02C8 01D4 | MACSRCADDRHI | MAC Source Address High Bytes Register (Upper 32-bits) |
| 02C8 01D8 | MACHASH1 | MAC Hash Address Register 1 |
| 02C8 01DC | MACHASH2 | MAC Hash Address Register 2 |
| 02C8 01E0 | BOFFTEST | Back Off Test Register |
| 02C8 01E4 | TPACETEST | Transmit Pacing Algorithm Test Register |
| 02C8 01E8 | RXPAUSE | Receive Pause Timer Register |
| 02C8 01EC | TXPAUSE | Transmit Pause Timer Register |
| 02C8 01F0 - 02C8 01FC | - | Reserved |
| 02C8 0200 - 02C8 02FC | (see Table 7-72) | EMAC Statistics Registers |
| 02C8 0300 - 02C8 03FC | - | Reserved |
| 02C8 0400 - 02C8 04FC | - | Reserved |
| 02C8 0500 | MACADDRLO | MAC Address Low Bytes Register (used in receive address matching) |
| 02C8 0504 | MACADDRHI | MAC Address High Bytes Register (used in receive address matching) |
| 02C8 0508 | MACINDEX | MAC Index Register |
| 02C8 050C - 02C8 05FC | - | Reserved |
| 02C8 0600 | TX0HDP | Transmit Channel 0 DMA Head Descriptor Pointer Register |
| 02C8 0604 | TX1HDP | Transmit Channel 1 DMA Head Descriptor Pointer Register |
| 02C8 0608 | TX2HDP | Transmit Channel 2 DMA Head Descriptor Pointer Register |
| 02C8 060C | TX3HDP | Transmit Channel 3 DMA Head Descriptor Pointer Register |
| 02C8 0610 | TX4HDP | Transmit Channel 4 DMA Head Descriptor Pointer Register |
| 02C8 0614 | TX5HDP | Transmit Channel 5 DMA Head Descriptor Pointer Register |
| 02C8 0618 | TX6HDP | Transmit Channel 6 DMA Head Descriptor Pointer Register |
| 02C8 061C | TX7HDP | Transmit Channel 7 DMA Head Descriptor Pointer Register |
| 02C8 0620 | RX0HDP | Receive Channel 0 DMA Head Descriptor Pointer Register |
| 02C8 0624 | RX1HDP | Receive Channel 1 DMA Head Descriptor Pointer Register |
| 02C8 0628 | RX2HDP | Receive Channel 2 DMA Head Descriptor Pointer Register |
| 02C8 062C | RX3HDP | Receive Channel 3 DMA Head Descriptor Pointer Register |
| 02C8 0630 | RX4HDP | Receive Channel 4 DMA Head Descriptor Pointer Register |
| 02C8 0634 | RX5HDP | Receive Channel 5 DMA Head Descriptor Pointer Register |
| 02C8 0638 | RX6HDP | Receive Channel 6 DMA Head Descriptor Pointer Register |
| 02C8 063C | RX7HDP | Receive Channel 7 DMA Head Descriptor Pointer Register |
| 02C8 0640 | TX0CP | Transmit Channel 0 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0644 | TX1CP | Transmit Channel 1 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0648 | TX2CP | Transmit Channel 2 Completion Pointer (Interrupt Acknowledge) Register |

**Table 7-71. Ethernet MAC (EMAC) Control Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02C8 064C | TX3CP | Transmit Channel 3 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0650 | TX4CP | Transmit Channel 4 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0654 | TX5CP | Transmit Channel 5 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0658 | TX6CP | Transmit Channel 6 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 065C | TX7CP | Transmit Channel 7 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0660 | RX0CP | Receive Channel 0 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0664 | RX1CP | Receive Channel 1 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0668 | RX2CP | Receive Channel 2 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 066C | RX3CP | Receive Channel 3 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0670 | RX4CP | Receive Channel 4 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0674 | RX5CP | Receive Channel 5 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0678 | RX6CP | Receive Channel 6 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 067C | RX7CP | Receive Channel 7 Completion Pointer (Interrupt Acknowledge) Register |
| 02C8 0680 - 02C8 06FC | - | Reserved |
| 02C8 0700 - 02C8 077C | - | Reserved<br>was State RAM Test Access Registers<br>Processor Read and Write Access to Head Descriptor Pointers and Interrupt Acknowledge Registers |
| 02C8 0780 - 02C8 0FFF | - | Reserved |

**Table 7-72. EMAC Statistics Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02C8 0200 | RXGOODFRAMES | Good Receive Frames Register |
| 02C8 0204 | RXBCASTFRAMES | Broadcast Receive Frames Register<br>(Total number of good broadcast frames received) |
| 02C8 0208 | RXMCASTFRAMES | Multicast Receive Frames Register<br>(Total number of good multicast frames received) |
| 02C8 020C | RXPAUSEFRAMES | Pause Receive Frames Register |
| 02C8 0210 | RXCRCERRORS | Receive CRC Errors Register (Total number of frames received with CRC errors) |
| 02C8 0214 | RXALIGNCODEERRORS | Receive Alignment/Code Errors Register<br>(Total number of frames received with alignment/code errors) |
| 02C8 0218 | RXOVERSIZED | Receive Oversized Frames Register<br>(Total number of oversized frames received) |
| 02C8 021C | RXJABBER | Receive Jabber Frames Register<br>(Total number of jabber frames received) |
| 02C8 0220 | RXUNDERSIZED | Receive Undersized Frames Register<br>(Total number of undersized frames received) |
| 02C8 0224 | RXFRAGMENTS | Receive Frame Fragments Register |
| 02C8 0228 | RXFILTERED | Filtered Receive Frames Register |
| 02C8 022C | RXQOSFILTERED | Received QOS Filtered Frames Register |

**Table 7-72. EMAC Statistics Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02C8 0230 | RXOCTETS | Receive Octet Frames Register (Total number of received bytes in good frames) |
| 02C8 0234 | TXGOODFRAMES | Good Transmit Frames Register (Total number of good frames transmitted) |
| 02C8 0238 | TXBCASTFRAMES | Broadcast Transmit Frames Register |
| 02C8 023C | TXMCASTFRAMES | Multicast Transmit Frames Register |
| 02C8 0240 | TXPAUSEFRAMES | Pause Transmit Frames Register |
| 02C8 0244 | TXDEFERRED | Deferred Transmit Frames Register |
| 02C8 0248 | TXCOLLISION | Transmit Collision Frames Register |
| 02C8 024C | TXSINGLECOLL | Transmit Single Collision Frames Register |
| 02C8 0250 | TXMULTICOLL | Transmit Multiple Collision Frames Register |
| 02C8 0254 | TXEXCESSIVECOLL | Transmit Excessive Collision Frames Register |
| 02C8 0258 | TXLATECOLL | Transmit Late Collision Frames Register |
| 02C8 025C | TXUNDERRUN | Transmit Underrun Error Register |
| 02C8 0260 | TXCARRIERSENSE | Transmit Carrier Sense Errors Register |
| 02C8 0264 | TXOCTETS | Transmit Octet Frames Register |
| 02C8 0268 | FRAME64 | Transmit and Receive 64 Octet Frames Register |
| 02C8 026C | FRAME65T127 | Transmit and Receive 65 to 127 Octet Frames Register |
| 02C8 0270 | FRAME128T255 | Transmit and Receive 128 to 255 Octet Frames Register |
| 02C8 0274 | FRAME256T511 | Transmit and Receive 256 to 511 Octet Frames Register |
| 02C8 0278 | FRAME512T1023 | Transmit and Receive 512 to 1023 Octet Frames Register |
| 02C8 027C | FRAME1024TUP | Transmit and Receive 1024 to 1518 Octet Frames Register |
| 02C8 0280 | NETOCTETS | Network Octet Frames Register |
| 02C8 0284 | RXSOFOVERRUNS | Receive FIFO or DMA Start of Frame Overruns Register |
| 02C8 0288 | RXMOFOVERRUNS | Receive FIFO or DMA Middle of Frame Overruns Register |
| 02C8 028C | RXDMAOVERRUNS | Receive DMA Start of Frame and Middle of Frame Overruns Register |
| 02C8 0290 - 02C8 02FC | - | Reserved |

**Table 7-73. EMAC Control Module Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02C8 1000 | - | Reserved |
| 02C8 1004 | EWCTL | EMAC Control Module Interrupt Control Register |
| 02C8 1008 | EWINTTCNT | EMAC Control Module Interrupt Timer Count Register |
| 02C8 100C - 02C8 17FF | - | Reserved |

**Table 7-74. EMAC Descriptor Memory**

| HEX ADDRESS RANGE | ACRONYM | DESCRIPTION |
|---|---|---|
| 02C8 2000 - 02C8 3FFF | - | EMAC Descriptor Memory |

## 7.14.3 EMAC Electrical Data/Timing

### 7.14.3.1 EMAC MII and GMII Electrical Data/Timing

**Table 7-75. Timing Requirements for MRCLK - MII and GMII Operation**

(see Figure 7-59)

| NO. | | | -720 -850 A-1000/-1000 | | | | | | UNIT |
|---|---|---|---|---|---|---|---|---|---|
| | | | 1000 Mbps (GMII Only) | | 100 Mbps | | 10 Mbps | | |
| | | | MIN | MAX | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(MRCLK)}$ | Cycle time, MRCLK | 8 | | 40 | | 400 | | ns |
| 2 | $t_{w(MRCLKH)}$ | Pulse duration, MRCLK high | 2.8 | | 14 | | 140 | | ns |
| 3 | $t_{w(MRCLKL)}$ | Pulse duration, MRCLK low | 2.8 | | 14 | | 140 | | ns |
| 4 | $t_{t(MRCLK)}$ | Transition time, MRCLK | | 1 | | 3 | | 3 | ns |

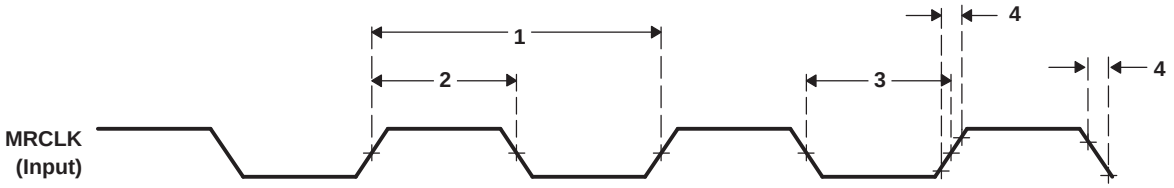



**Figure 7-59. MRCLK Timing (EMAC - Receive) [MII and GMII Operation]**

**Table 7-76. Timing Requirements for MTCLK - MII and GMII Operation**

(see Figure 7-60)

| NO. | | | -720 -850 A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | 100 Mbps | | 10 Mbps | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(MTCLK)}$ | Cycle time, MTCLK | 40 | | 400 | | ns |
| 2 | $t_{w(MTCLKH)}$ | Pulse duration, MTCLK high | 14 | | 140 | | ns |
| 3 | $t_{w(MTCLKL)}$ | Pulse duration, MTCLK low | 14 | | 140 | | ns |
| 4 | $t_{t(MTCLK)}$ | Transition time, MTCLK | | 3 | | 3 | ns |

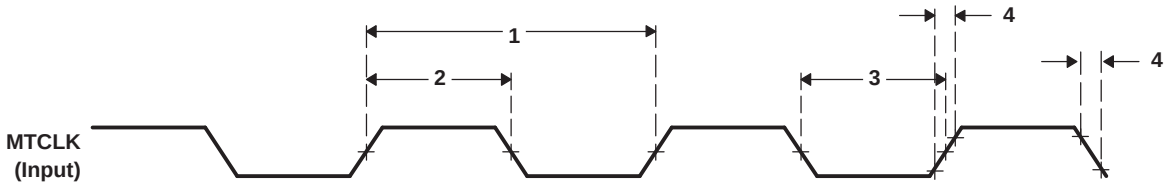



**Figure 7-60. MTCLK Timing (EMAC - Transmit) [MII and GMII Operation]**

### Table 7-77. Switching Characteristics Over Recommended Operating Conditions for GMTCLK - GMII Operation

(see Figure 7-61)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | 1000 Mbps | | |
| | | | MIN | MAX | |
| 1 | $t_{c(GMTCLK)}$ | Cycle time, GMTCLK | 8 | | ns |
| 2 | $t_{w(GMTCLKH)}$ | Pulse duration, GMTCLK high | 2.8 | | ns |
| 3 | $t_{w(GMTCLKL)}$ | Pulse duration, GMTCLK low | 2.8 | | ns |
| 4 | $t_{t(GMTCLK)}$ | Transition time, GMTCLK | | 1 | ns |

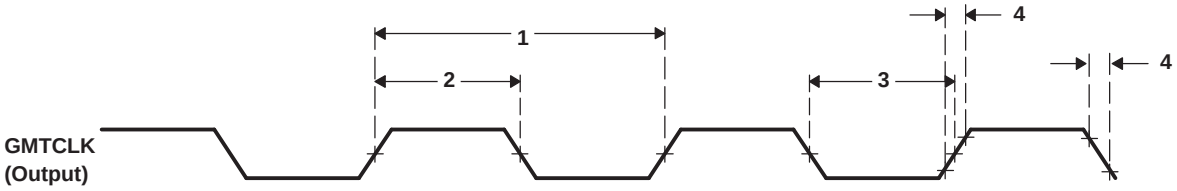



**Figure 7-61. GMTCLK Timing (EMAC - Transmit) [GMII Operation]**

### Table 7-78. Timing Requirements for EMAC MII and GMII Receive 10/100/1000 Mbit/s(1)

(see Figure 7-62)

| NO. | | | -720 -850 A-1000/-1000 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | 1000 Mbps | | 100/10 Mbps | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{su(MRXD-MRCLKH)}$ | Setup time, receive selected signals valid before MRCLK high | 2 | | 8 | | ns |
| 2 | $t_{h(MRCLKH-MRXD)}$ | Hold time, receive selected signals valid after MRCLK high | 0 | | 8 | | ns |

(1) For **MII**, Receive selected signals include: MRXD[3:0], MRXDV, and MRXER. For **GMII**, Receive selected signals include: MRXD[7:0], MRXDV, and MRXER.

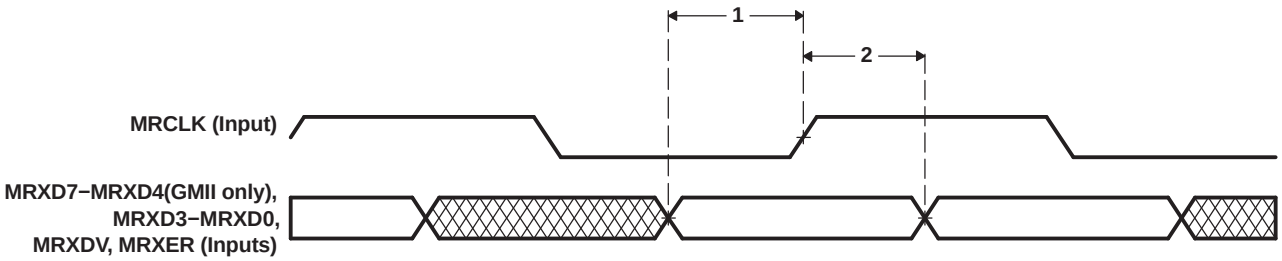



**Figure 7-62. EMAC Receive Interface Timing [MII and GMII Operation]**

**Table 7-79. Switching Characteristics Over Recommended Operating Conditions for EMAC MII and GMII Transmit 10/100 Mbit/s(1)**

(see Figure 7-63)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000<br>100/10 Mbps | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(MTCLKH-MTXD)}$ | Delay time, MTCLK high to transmit selected signals valid | 5 | 25 | ns |

(1) For **MII**, Transmit selected signals include: MTXD[3:0] and MTXEN. For **GMII**, Transmit selected signals include: GMTXD[7:0] and MTXEN.

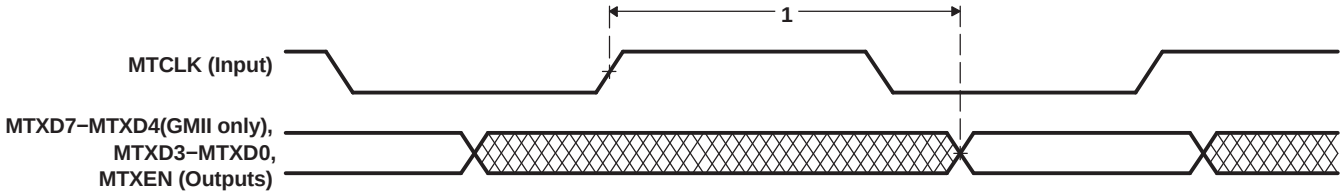



**Figure 7-63. EMAC Transmit Interface Timing [MII and GMII Operation]**

**Table 7-80. Switching Characteristics Over Recommended Operating Conditions for EMAC GMII Transmit 1000 Mbit/s(1)**

(see Figure 7-64)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000<br>1000 Mbps | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(GMTCLKH-MTXD)}$ | Delay time, GMTCLK high to transmit selected signals valid | 0.5 | 5 | ns |

(1) For **GMII**, Transmit selected signals include: GMTXD[7:0] and MTXEN.

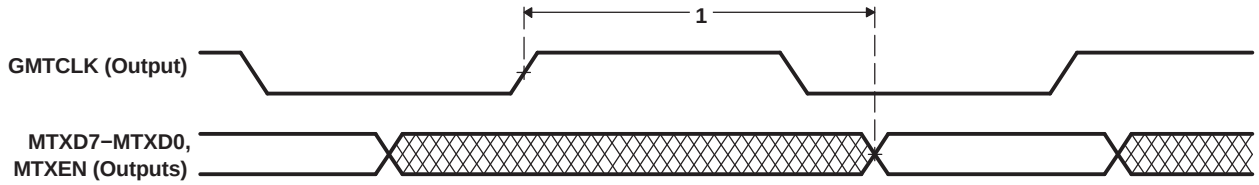



**Figure 7-64. EMAC Transmit Interface Timing [GMII Operation]**

### 7.14.3.2 EMAC RMII Electrical Data/Timing

The RMREFCLK pin is used to source a clock to the EMAC when it is configured for RMII operation. The RMREFCLK frequency should be 50 MHz ±50 PPM with a duty cycle between 35% and 65%, inclusive.

**Table 7-81. Timing Requirements for RMREFCLK - RMII Operation**

(see Figure 7-65)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(RMREFCLKH)}$ | Pulse duration, RMREFCLK high | 7 | 13 | ns |
| 2 | $t_{w(RMREFCLKL)}$ | Pulse duration, RMREFCLK low | 7 | 13 | ns |
| 3 | $t_{t(RMREFCLK)}$ | Transition time, RMREFCLK | | 2 | ns |

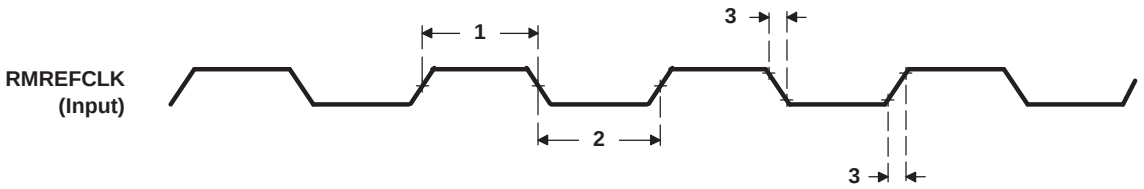



**Figure 7-65. RMREFCLK Timing**

**Table 7-82. Switching Characteristics Over Recommended Operating Conditions for EMAC RMII Transmit 10/100 Mbit/s(1)**

(see Figure 7-66)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | 1000 Mbps | | |
| | | | MIN | MAX | |
| 1 | $t_{d(RMREFCLKH-RMTXD)}$ | Delay time, RMREFCLK high to transmit selected signals valid | 3 | 10 | ns |

(1) For RMII, transmit selected signals include: RMTXD[1:0] and RMTXEN.

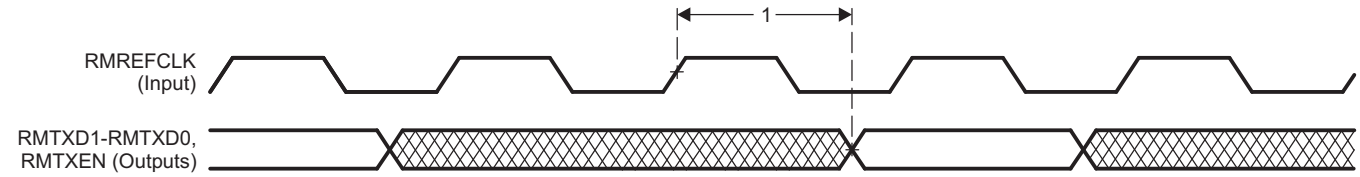



**Figure 7-66. EMAC Transmit Interface Timing [RMII Operation]**

**Table 7-83. Timing Requirements for EMAC RMII Input Receive for 100 Mbps(1)**

(see Figure 7-67)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(RMRXD-RMREFCLK)}$ | Setup time, receive selected signals valid before RMREFCLK (at DSP) high/low | 4.0 | | ns |
| 2 | $t_{h(RMREFCLK-RMRXD)}$ | Hold time, receive selected signals valid after RMREFCLK (at DSP) high/low | 2.0 | | ns |

(1) For RMII, receive selected signals include: RMRXD[1:0], RMRXER, and RMCRSDV.

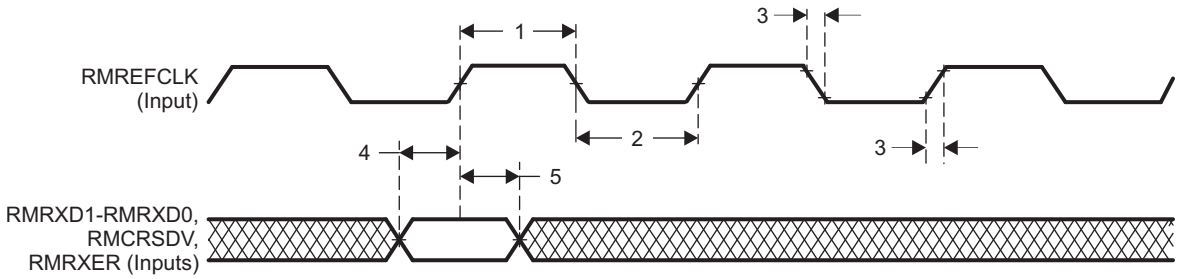



**Figure 7-67. EMAC Receive Interface Timing [RMII Operation]**

### 7.14.3.3 EMAC RGMII Electrical Data/Timing

An extra clock signal, RGREFCLK, running at 125 MHz is included as a convenience to the user. Note that this reference clock is ***not*** a free-running clock. This should only be used by an external device if it does not expect a valid clock during device reset.

**Table 7-84. Switching Characteristics Over Recommended Operating Conditions for EMAC RGREFCLK - RGMII Operation**

(see Figure 7-68)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(RGFCLK)}$ | Cycle time, RGREFCLK | 8 - 0.8 | 8 + 0.8 | ns |
| 2 | $t_{w(RGFCLKH)}$ | Pulse duration, RGREFCLK high | 3.2 | 4.8 | ns |
| 3 | $t_{w(RGFCLKL)}$ | Pulse duration, RGREFCLK low | 3.2 | 4.8 | ns |
| 4 | $t_{t(RGFCLK)}$ | Transition time, RGREFCLK | | 0.75 | ns |

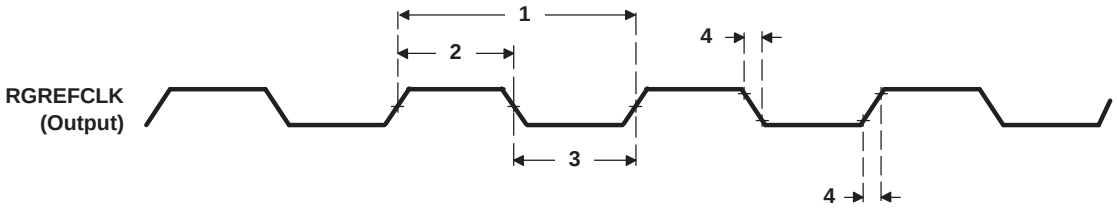



**Figure 7-68. RGREFCLK Timing**

**Table 7-85. Timing Requirements for RGRXC - RGMII Operation**

(see Figure 7-69)

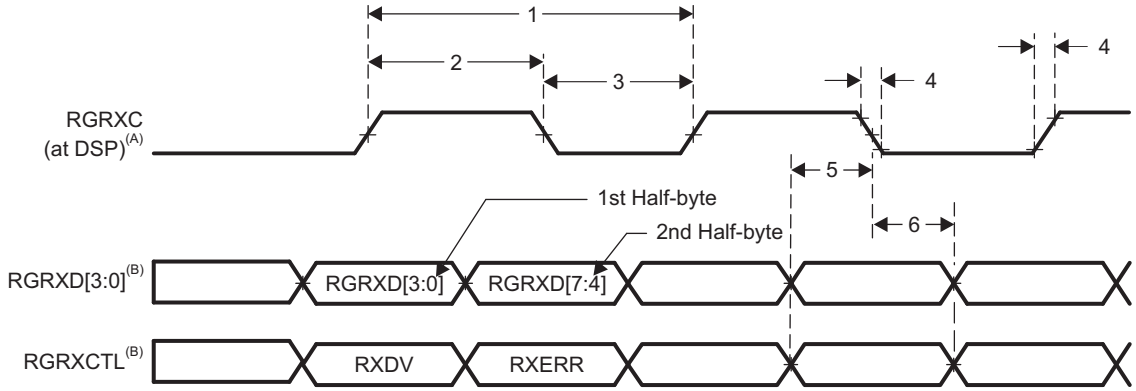
| NO. | | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 1 | $t_{c(RGRXC)}$ | Cycle time, RGRXC | 10 Mbps | 360 | 440 | ns |
| | | | 100 Mbps | 36 | 44 | |
| | | | 1000 Mbps | 7.2 | 8.8 | |
| 2 | $t_{w(RGRXCH)}$ | Pulse duration, RGRXC high | 10 Mbps | 0.40*$t_{c(RGRXC)}$ | 0.60*$t_{c(RGRXC)}$ | ns |
| | | | 100 Mbps | 0.40*$t_{c(RGRXC)}$ | 0.60*$t_{c(RGRXC)}$ | |
| | | | 1000 Mbps | 0.45*$t_{c(RGRXC)}$ | 0.55*$t_{c(RGRXC)}$ | |
| 3 | $t_{w(RGRXCL)}$ | Pulse duration, RGRXC low | 10 Mbps | 0.40*$t_{c(RGRXC)}$ | 0.60*$t_{c(RGRXC)}$ | ns |
| | | | 100 Mbps | 0.40*$t_{c(RGRXC)}$ | 0.60*$t_{c(RGRXC)}$ | |
| | | | 1000 Mbps | 0.45*$t_{c(RGRXC)}$ | 0.55*$t_{c(RGRXC)}$ | |
| 4 | $t_{t(RGRXC)}$ | Transition time, RGRXC | 10 Mbps | | 0.75 | ns |
| | | | 100 Mbps | | 0.75 | |
| | | | 1000 Mbps | | 0.75 | |

**Table 7-86. Timing Requirements for EMAC RGMII Input Receive for 10/100/1000 Mbps(1)**

(see Figure 7-69)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 5 | $t_{su(RGRXD-RGRXCH)}$ | Setup time, receive selected signals valid before RGRXC (at DSP) high/low | 1.0 | | ns |
| 6 | $t_{h(RGRXCH-RGRXD)}$ | Hold time, receive selected signals valid after RGRXC (at DSP) high/low | 1.0 | | ns |

(1) For RGMII, receive selected signals include: RGRXD[3:0] and RGRXCTL.




A. RGRXC must be externally delayed relative to the data and control pins.

B. Data and control information is received using both edges of the clocks. RGRXD[3:0] carries data bits 3-0 on the rising edge of RGRXC and data bits 7-4 on the falling edge of RGRXC. Similarly, RGRXCTL carries RXDV on rising edge of RGRXC and RXERR on falling edge.

**Figure 7-69. EMAC Receive Interface Timing [RGMII Operation]**

**Table 7-87. Switching Characteristics Over Recommended Operating Conditions for RGTXC - RGMII Operation for 10/100/1000 Mbit/s**

(see Figure 7-70)

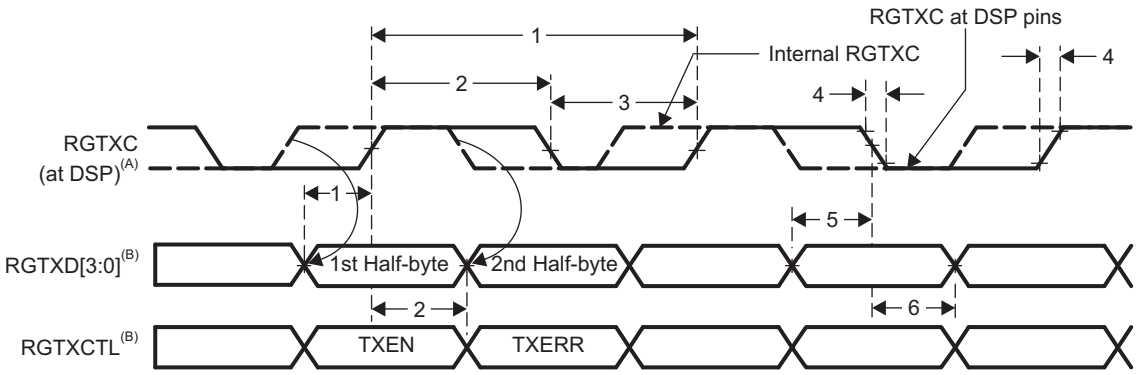
| NO. | | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 1 | $t_{c(RGTXC)}$ | Cycle time, RGTXC | 10 Mbps | 360 | 440 | ns |
| | | | 100 Mbps | 36 | 44 | |
| | | | 1000 Mbps | 7.2 | 8.8 | |
| 2 | $t_{w(RGTXCH)}$ | Pulse duration, RGTXC high | 10 Mbps | $0.40*t_{c(RGTXC)}$ | $0.60*t_{c(RGTXC)}$ | ns |
| | | | 100 Mbps | $0.40*t_{c(RGTXC)}$ | $0.60*t_{c(RGTXC)}$ | |
| | | | 1000 Mbps | $0.45*t_{c(RGTXC)}$ | $0.55*t_{c(RGTXC)}$ | |
| 3 | $t_{w(RGTXCL)}$ | Pulse duration, RGTXC low | 10 Mbps | $0.40*t_{c(RGTXC)}$ | $0.60*t_{c(RGTXC)}$ | ns |
| | | | 100 Mbps | $0.40*t_{c(RGTXC)}$ | $0.60*t_{c(RGTXC)}$ | |
| | | | 1000 Mbps | $0.45*t_{c(RGTXC)}$ | $0.55*t_{c(RGTXC)}$ | |
| 4 | $t_{t(RGTXC)}$ | Transition time, RGTXC | 10 Mbps | | 0.75 | ns |
| | | | 100 Mbps | | 0.75 | |
| | | | 1000 Mbps | | 0.75 | |

**Table 7-88. Switching Characteristics Over Recommended Operating Conditions for EMAC RGMII Transmit(1)**

(see Figure 7-70)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 5 | $t_{su(RGTXD-RGTXCH)}$ | Setup time, transmit selected signals valid before RGTXC (at DSP) high/low | 1.2 | | ns |
| 6 | $t_{h(RGTXCH-RGTXD)}$ | Hold time, transmit selected signals valid after RGTXC (at DSP) high/low | 1.2 | | |

(1) For RGMII, transmit selected signals include: RGTXD[3:0] and RGTXCTL.




A. RGTXC is delayed internally before being driven to the RGTXC pin.

B. Data and control information is transmitted using both edges of the clocks. RGTXD[3:0] carries data bits 3-0 on the rising edge of RGTXC and data bits 7-4 on the falling edge of RGTXC. Similarly, RGTXCTL carries TXEN on rising edge of RGTXC and TXERR of falling edge.

**Figure 7-70. EMAC Transmit Interface Timing [RGMII Operation]**

### 7.14.4 Management Data Input/Output (MDIO)

The Management Data Input/Output (MDIO) module implements the 802.3 serial management interface to interrogate and controls up to 32 Ethernet PHY(s) connected to the device, using a shared two-wire bus. Application 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.

The EMAC control module is the main interface between the device core processor, the MDIO module, and the EMAC module. The relationship between these three components is shown in Figure 7-58.

The MDIO uses the same pins for the MII, GMII, and RMII modes. Standalone pins are included for the RGMII mode due to specific voltage requirements. Only one mode can be used at a time. The mode used is selected at device reset based on the MACSEL[1:0] configuration pins (for more detailed information, see Section 3, *Device Configuration*). Table 7-70 above shows which multiplexed pin are used in the MII, GMII, and RMII modes on the MDIO.

For more detailed information on the EMAC/MDIO, see the *TMS320C645x DSP EMAC/MDIO Module Reference Guide* (literature number SPRU975).

#### 7.14.4.1 MDIO Device-Specific Information

##### *Clocking Information*

The MDIO clock is based on a divide-down of the SYSCLK3 (from the PLL1 controller) and is specified to run up to 2.5 MHz, although typical operation is 1.0 MHz. Since the peripheral clock frequency is variable, the application software or driver controls the divide-down amount.

#### 7.14.4.2 MDIO Peripheral Register Descriptions

**Table 7-89. MDIO Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02C8 1800 | VERSION | MDIO Version Register |
| 02C8 1804 | CONTROL | MDIO Control Register |
| 02C8 1808 | ALIVE | MDIO PHY Alive Status Register |
| 02C8 180C | LINK | MDIO PHY Link Status Register |
| 02C8 1810 | LINKINTRAW | MDIO Link Status Change Interrupt (Unmasked) Register |
| 02C8 1814 | LINKINTMASKED | MDIO Link Status Change Interrupt (Masked) Register |
| 02C8 1818 - 02C8 181C | - | Reserved |
| 02C8 1820 | USERINTRAW | MDIO User Command Complete Interrupt (Unmasked) Register |
| 02C8 1824 | USERINTMASKED | MDIO User Command Complete Interrupt (Masked) Register |
| 02C8 1828 | USERINTMASKSET | MDIO User Command Complete Interrupt Mask Set Register |
| 02C8 182C | USERINTMASKCLEAR | MDIO User Command Complete Interrupt Mask Clear Register |
| 02C8 1830 - 02C8 187C | - | Reserved |
| 02C8 1880 | USERACCESS0 | MDIO User Access Register 0 |
| 02C8 1884 | USERPHYSEL0 | MDIO User PHY Select Register 0 |
| 02C8 1888 | USERACCESS1 | MDIO User Access Register 1 |
| 02C8 188C | USERPHYSEL1 | MDIO User PHY Select Register 1 |
| 02C8 1890 - 02C8 1FFF | - | Reserved |

### 7.14.4.3 MDIO Electrical Data/Timing

**Table 7-90. Timing Requirements for MDIO Input (R)(G)MII**

(see Figure 7-71)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(MDCLK)}$ | Cycle time, MDCLK | 400 | | ns |
| 2a | $t_{w(MDCLK)}$ | Pulse duration, MDCLK high | 180 | | ns |
| 2b | $t_{w(MDCLK)}$ | Pulse duration, MDCLK low | 180 | | ns |
| 3 | $t_{t(MDCLK)}$ | Transition time, MDCLK | | 5 | ns |
| 4 | $t_{su(MDIO-MDCLKH)}$ | Setup time, MDIO data input valid before MDCLK high | 10 | | ns |
| 5 | $t_{h(MDCLKH-MDIO)}$ | Hold time, MDIO data input valid after MDCLK high | 10 | | ns |

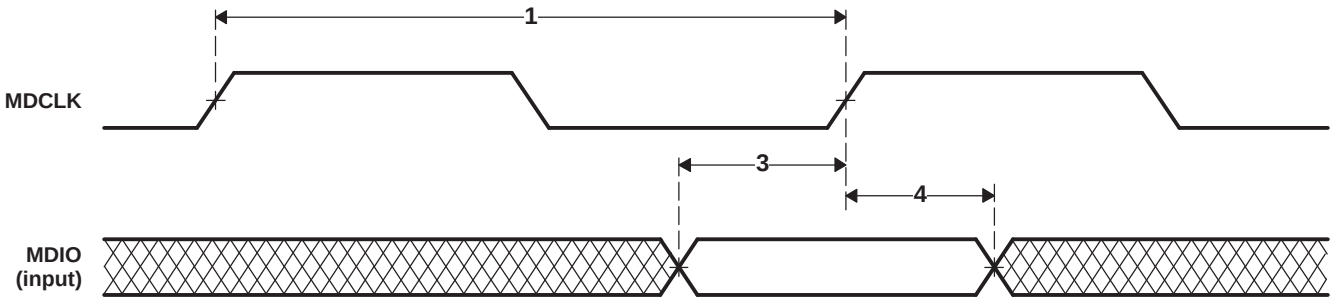



**Figure 7-71. MDIO Input Timing**

**Table 7-91. Switching Characteristics Over Recommended Operating Conditions for MDIO Output**

(see Figure 7-72)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 7 | $t_{d(MDCLKL-MDIO)}$ | Delay time, MDCLK low to MDIO data output valid | | 100 | ns |

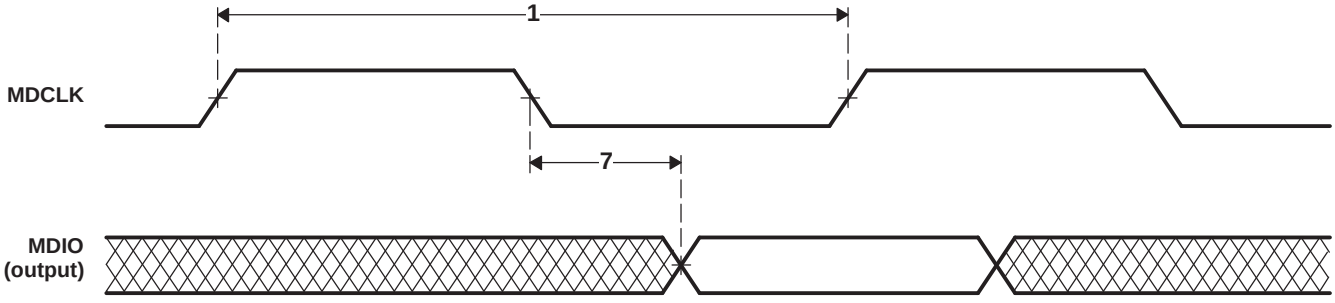



**Figure 7-72. MDIO Output Timing**

## 7.15 Timers

The timers can be used to: time events, count events, generate pulses, interrupt the CPU, and send synchronization events to the EDMA3 channel controller.

### 7.15.1 Timers Device-Specific Information

The C6454 device has two general-purpose timers, Timer0 and Timer1, each of which can be configured as a general-purpose timer or a watchdog timer. When configured as a general-purpose timer, each timer can be programmed as a 64-bit timer or as two separate 32-bit timers.

Each timer is made up of two 32-bit counters: a high counter and a low counter. The timer pins, TINPLx and TOUTLx are connected to the low counter. The high counter does not have any external device pins.

### 7.15.2 Timers Peripheral Register Descriptions

**Table 7-92. Timer 0 Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 0294 0000 | - | Reserved | |
| 0294 0004 | EMUMGT_CLKSPD0 | Timer 0 Emulation Management/Clock Speed Register | |
| 0294 0008 | - | Reserved | |
| 0294 000C | - | Reserved | |
| 0294 0010 | CNTLO0 | Timer 0 Counter Register Low | |
| 0294 0014 | CNTHI0 | Timer 0 Counter Register High | |
| 0294 0018 | PRDLO0 | Timer 0 Period Register Low | |
| 0294 001C | PRDHI0 | Timer 0 Period Register High | |
| 0294 0020 | TCR0 | Timer 0 Control Register | |
| 0294 0024 | TGCR0 | Timer 0 Global Control Register | |
| 0294 0028 | WDTCR0 | Timer 0 Watchdog Timer Control Register | |
| 0294 002C | - | Reserved | |
| 0294 0030 | - | Reserved | |
| 0294 0034 - 0297 FFFF | - | Reserved | |

**Table 7-93. Timer 1 Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 0298 0000 | - | Reserved | |
| 0298 0004 | EMUMGT_CLKSPD1 | Timer 1 Emulation Management/Clock Speed Register | |
| 0298 0008 | - | Reserved | |
| 0298 000C | - | Reserved | |
| 0298 0010 | CNTLO1 | Timer 1 Counter Register Low | |
| 0298 0014 | CNTHI1 | Timer 1 Counter Register High | |
| 0298 0018 | PRDLO1 | Timer 1 Period Register Low | |
| 0298 001C | PRDHI1 | Timer 1 Period Register High | |
| 0298 0020 | TCR1 | Timer 1 Control Register | |
| 0298 0024 | TGCR1 | Timer 1 Global Control Register | |
| 0298 0028 | WDTCR1 | Timer 1 Watchdog Timer Control Register | |
| 0298 002C | - | Reserved | |
| 0298 0030 | - | Reserved | |
| 0298 0034 - 0299 FFFF | - | Reserved | |

### 7.15.3 Timers Electrical Data/Timing

**Table 7-94. Timing Requirements for Timer Inputs(1)**

(see Figure 7-73)

| NO. | | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(TINPH)}$ | Pulse duration, TINPLx high | 12P | | ns |
| 2 | $t_{w(TINPL)}$ | Pulse duration, TINPLx low | 12P | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.

**Table 7-95. Switching Characteristics Over Recommended Operating Conditions for Timer Outputs(1)**

(see Figure 7-73)

| NO. | PARAMETER | | -720 -850 A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{w(TOUTH)}$ | Pulse duration, TOUTLx high | 12P - 3 | | ns |
| 4 | $t_{w(TOUTL)}$ | Pulse duration, TOUTLx low | 12P - 3 | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.

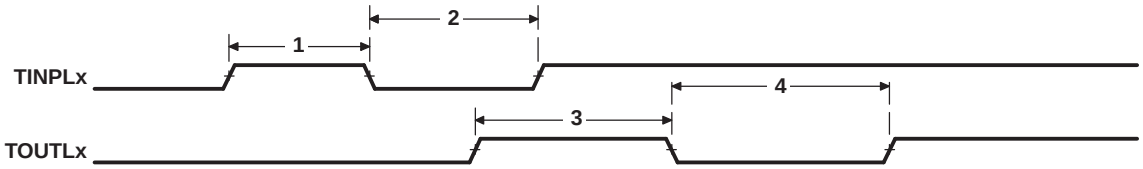



**Figure 7-73. Timer Timing**

## 7.16 Peripheral Component Interconnect (PCI)

The C6454 DSP supports connections to a PCI backplane via the integrated PCI master/slave bus interface. The PCI port interfaces to DSP internal resources via the data switched central resource. The data switched central resource is described in more detail in Section 4.

For more detailed information on the PCI port peripheral module, see the *TMS320C645x DSP Peripheral Component Interconnect (PCI) User's Guide* (literature number SPRUE60).

### *7.16.1 PCI Device-Specific Information*

The PCI peripheral on the C6454 DSP conforms to the *PCI Local Bus Specification* (version 2.3). The PCI peripheral can act both as a PCI bus master and as a target. It supports PCI bus operation of speeds up to 66 MHz and uses a 32-bit data/address bus.

On the C6454 device, the pins of the PCI peripheral are multiplexed with the pins of the HPI and GPIO peripherals. PCI functionality for these pins is controlled (enabled/disabled) by the PCI_EN pin (Y29). The maximum speed of the PCI, 33 MHz or 66 MHz, is controlled through the PCI66 pin (U27). For more detailed information on the peripheral control, see Section 3, *Device Configuration*.

The C6454 device provides an initialization mechanism through which the default values for some of the PCI configuration registers can be read from an I2C EEPROM. Table 7-96 shows the registers which can be initialized through the PCI auto-initialization. Also shown is the default value of these registers when PCI auto-initialization is not used. PCI auto-initialization is controlled (enabled/disabled) through the PCI_EEAI pin (P25). For more information on this feature, see the *TMS320C645x DSP Peripheral Component Interconnect (PCI) User's Guide* (literature number SPRUE60) and the *TMS320C645x Bootloader User's Guide* (literature number SPRUEC6).

**Table 7-96. Default Values for PCI Configuration Registers**

| REGISTER | DEFAULT VALUE |
|---|---|
| Vendor ID/Device ID Register (PCIVENDEV) | 104C B000h |
| Class Code/Revision ID Register (PCICLREV) | 0000 0001h |
| Subsystem Vendor ID/Subsystem ID Register (PCISUBID) | 0000 0000h |
| Max Latency/Min Grant/Interrupt Pin/Interrupt Line Register (PCILGINT) | 0000 0100h |

The on-chip Bootloader supports a host boot which allows an external PCI device to load application code into the DSP's memory space. The PCI boot is terminated when the Host generates a DSP interrupt. The Host can generate a DSP interrupt through the PCI peripheral by setting the DSPINT bit in the Back-End Application Interrupt Enable Set Register (PCIBINTSET) and the Status Set Register (PCISTATSET). For more information on the boot sequence of the C6454 DSP, see Section 2.4.

**NOTE**

After the host boot is complete, the DSP interrupt is registered in bit 0 (channel 0) of the EDMA Event Register (ER). This event must be cleared by software before triggering transfers on DMA channel 0.

### 7.16.2 PCI Peripheral Register Descriptions

**Table 7-97. PCI Configuration Registers**

| PCI HOST ACCESS HEX ADDRESS OFFSET | ACRONYM | PCI HOST ACCESS REGISTER NAME |
|---|---|---|
| 0x00 | PCIVENDEV | Vendor ID/Device ID |
| 0x04 | PCICSR | Command/Status |
| 0x08 | PCICLREV | Class Code/Revision ID |
| 0x0C | PCICLINE | BIST/Header Type/Latency Timer/Cacheline Size |
| 0x10 | PCIBAR0 | Base Address 0 |
| 0x14 | PCIBAR1 | Base Address 1 |
| 0x18 | PCIBAR2 | Base Address 2 |
| 0x1C | PCIBAR3 | Base Address 3 |
| 0x20 | PCIBAR4 | Base Address 4 |
| 0x24 | PCIBAR5 | Base Address 5 |
| 0x28 - 0x2B | - | Reserved |
| 0x2C | PCISUBID | Subsystem Vendor ID/Subsystem ID |
| 0x30 | - | Reserved |
| 0x34 | PCICPBPTR | Capabilities Pointer |
| 0x38 - 0x3B | - | Reserved |
| 0x3C | PCILGINT | Max Latency/Min Grant/Interrupt Pin/Interrupt Line |
| 0x40 - 0x7F | - | Reserved |

**Table 7-98. PCI Back-End Configuration Registers**

| DSP ACCESS HEX ADDRESS RANGE | ACRONYM | DSP ACCESS REGISTER NAME |
|---|---|---|
| 02C0 0000 - 02C0 000F | - | Reserved |
| 02C0 0010 | PCISTATSET | PCI Status Set Register |
| 02C0 0014 | PCISTATCLR | PCI Status Clear Register |
| 02C0 0018 - 02C0 001F | - | Reserved |
| 02C0 0020 | PCIHINTSET | PCI Host Interrupt Enable Set Register |
| 02C0 0024 | PCIHINTCLR | PCI Host Interrupt Enable Clear Register |
| 02C0 0028 - 02C0 002F | - | Reserved |
| 02C0 0030 | PCIBINTSET | PCI Back End Application Interrupt Enable Set Register |
| 02C0 0034 | PCIBINTCLR | PCI Back End Application Interrupt Enable Clear Register |
| 02C0 0038 | PCIBCLKMGT | PCI Back End Application Clock Management Register |
| 02C0 003C - 02C0 00FF | - | Reserved |
| 02C0 0100 | PCIVENDEVMIR | PCI Vendor ID/Device ID Mirror Register |
| 02C0 0104 | PCICSRMIR | PCI Command/Status Mirror Register |
| 02C0 0108 | PCICLREVMIR | PCI Class Code/Revision ID Mirror Register |
| 02C0 010C | PCICLINEMIR | PCI BIST/Header Type/Latency Timer/Cacheline Size Mirror Register |
| 02C0 0110 | PCIBAR0MSK | PCI Base Address Mask Register 0 |
| 02C0 0114 | PCIBAR1MSK | PCI Base Address Mask Register 1 |
| 02C0 0118 | PCIBAR2MSK | PCI Base Address Mask Register 2 |
| 02C0 011C | PCIBAR3MSK | PCI Base Address Mask Register 3 |
| 02C0 0120 | PCIBAR4MSK | PCI Base Address Mask Register 4 |
| 02C0 0124 | PCIBAR5MSK | PCI Base Address Mask Register 5 |
| 02C0 0128 - 02C0 012B | - | Reserved |
| 02C0 012C | PCISUBIDMIR | PCI Subsystem Vendor ID/Subsystem ID Mirror Register |
| 02C0 0130 | - | Reserved |

**Table 7-98. PCI Back-End Configuration Registers (continued)**

| DSP ACCESS HEX ADDRESS RANGE | ACRONYM | DSP ACCESS REGISTER NAME |
|---|---|---|
| 02C0 0134 | PCICPBPTRMIR | PCI Capabilities Pointer Mirror Register |
| 02C0 0138 - 02C0 013B | - | Reserved |
| 02C0 013C | PCILGINTMIR | PCI Max Latency/Min Grant/Interrupt Pin/Interrupt Line Mirror Register |
| 02C0 0140 - 02C0 017F | - | Reserved |
| 02C0 0180 | PCISLVCNTL | PCI Slave Control Register |
| 02C0 0184 - 02C0 01BF | - | Reserved |
| 02C0 01C0 | PCIBAR0TRL | PCI Slave Base Address 0 Translation Register |
| 02C0 01C4 | PCIBAR1TRL | PCI Slave Base Address 1 Translation Register |
| 02C0 01C8 | PCIBAR2TRL | PCI Slave Base Address 2 Translation Register |
| 02C0 01CC | PCIBAR3TRL | PCI Slave Base Address 3 Translation Register |
| 02C0 01D0 | PCIBAR4TRL | PCI Slave Base Address 4 Translation Register |
| 02C0 01D4 | PCIBAR5TRL | PCI Slave Base Address 5 Translation Register |
| 02C0 01D8 - 02C0 01DF | - | Reserved |
| 02C0 01E0 | PCIBAR0MIR | PCI Base Address Register 0 Mirror Register |
| 02C0 01E4 | PCIBAR1MIR | PCI Base Address Register 1 Mirror Register |
| 02C0 01E8 | PCIBAR2MIR | PCI Base Address Register 2 Mirror Register |
| 02C0 01EC | PCIBAR3MIR | PCI Base Address Register 3 Mirror Register |
| 02C0 01F0 | PCIBAR4MIR | PCI Base Address Register 4 Mirror Register |
| 02C0 01F4 | PCIBAR5MIR | PCI Base Address Register 5 Mirror Register |
| 02C0 01F8 - 02C0 02FF | - | Reserved |
| 02C0 0300 | PCIMCFGDAT | PCI Master Configuration/IO Access Data Register |
| 02C0 0304 | PCIMCFGADR | PCI Master Configuration/IO Access Address Register |
| 02C0 0308 | PCIMCFGCMD | PCI Master Configuration/IO Access Command Register |
| 02C0 030C - 02C0 030F | - | Reserved |
| 02C0 0310 | PCIMSTCFG | PCI Master Configuration Register |

**Table 7-99. DSP-to-PCI Address Translation Registers**

| DSP ACCESS HEX ADDRESS RANGE | ACRONYM | DSP ACCESS REGISTER NAME |
|---|---|---|
| 02C0 0314 | PCIADDSUB0 | PCI Address Substitute 0 Register |
| 02C0 0318 | PCIADDSUB1 | PCI Address Substitute 1 Register |
| 02C0 031C | PCIADDSUB2 | PCI Address Substitute 2 Register |
| 02C0 0320 | PCIADDSUB3 | PCI Address Substitute 3 Register |
| 02C0 0324 | PCIADDSUB4 | PCI Address Substitute 4 Register |
| 02C0 0328 | PCIADDSUB5 | PCI Address Substitute 5 Register |
| 02C0 032C | PCIADDSUB6 | PCI Address Substitute 6 Register |
| 02C0 0330 | PCIADDSUB7 | PCI Address Substitute 7 Register |
| 02C0 0334 | PCIADDSUB8 | PCI Address Substitute 8 Register |
| 02C0 0338 | PCIADDSUB9 | PCI Address Substitute 9 Register |
| 02C0 033C | PCIADDSUB10 | PCI Address Substitute 10 Register |
| 02C0 0340 | PCIADDSUB11 | PCI Address Substitute 11 Register |
| 02C0 0344 | PCIADDSUB12 | PCI Address Substitute 12 Register |
| 02C0 0348 | PCIADDSUB13 | PCI Address Substitute 13 Register |
| 02C0 034C | PCIADDSUB14 | PCI Address Substitute 14 Register |
| 02C0 0350 | PCIADDSUB15 | PCI Address Substitute 15 Register |
| 02C0 0354 | PCIADDSUB16 | PCI Address Substitute 16 Register |
| 02C0 0358 | PCIADDSUB17 | PCI Address Substitute 17 Register |

**Table 7-99. DSP-to-PCI Address Translation Registers (continued)**

| DSP ACCESS HEX ADDRESS RANGE | ACRONYM | DSP ACCESS REGISTER NAME |
|---|---|---|
| 02C0 035C | PCIADDSUB18 | PCI Address Substitute 18 Register |
| 02C0 0360 | PCIADDSUB19 | PCI Address Substitute 19 Register |
| 02C0 0364 | PCIADDSUB20 | PCI Address Substitute 20 Register |
| 02C0 0368 | PCIADDSUB21 | PCI Address Substitute 21 Register |
| 02C0 036C | PCIADDSUB22 | PCI Address Substitute 22 Register |
| 02C0 0370 | PCIADDSUB23 | PCI Address Substitute 23 Register |
| 02C0 0374 | PCIADDSUB24 | PCI Address Substitute 24 Register |
| 02C0 0378 | PCIADDSUB25 | PCI Address Substitute 25 Register |
| 02C0 037C | PCIADDSUB26 | PCI Address Substitute 26 Register |
| 02C0 0380 | PCIADDSUB27 | PCI Address Substitute 27 Register |
| 02C0 0384 | PCIADDSUB28 | PCI Address Substitute 28 Register |
| 02C0 0388 | PCIADDSUB29 | PCI Address Substitute 29 Register |
| 02C0 038C | PCIADDSUB30 | PCI Address Substitute 30 Register |
| 02C0 0390 | PCIADDSUB31 | PCI Address Substitute 31 Register |

**Table 7-100. PCI Hook Configuration Registers**

| DSP ACCESS HEX ADDRESS RANGE | ACRONYM | DSP ACCESS REGISTER NAME |
|---|---|---|
| 02C0 0394 | PCIVENDEVPRG | PCI Vendor ID and Device ID Program Register |
| 02C0 0398 | PCICMDSTATPRG | PCI Command and Status Program Register |
| 02C0 039C | PCICLREVPRG | PCI Class Code and Revision ID Program Register |
| 02C0 03A0 | PCISUBIDPRG | PCI Subsystem Vendor ID and Subsystem ID Program Register |
| 02C0 03A4 | PCIMAXLGPRG | PCI Max Latency and Min Grant Program Register |
| 02C0 03A8 | PCILRSTREG | PCI LRESET Register |
| 02C0 03AC | PCICFGDONE | PCI Configuration Done Register |
| 02C0 03B0 | PCIBAR0MPRG | PCI Base Address Mask Register 0 Program Register |
| 02C0 03B4 | PCIBAR1MPRG | PCI Base Address Mask Register 1 Program Register |
| 02C0 03B8 | PCIBAR2MPRG | PCI Base Address Mask Register 2 Program Register |
| 02C0 03BC | PCIBAR3MPRG | PCI Base Address Mask Register 3 Program Register |
| 02C0 03C0 | PCIBAR4MPRG | PCI Base Address Mask Register 4 Program Register |
| 02C0 03C4 | PCIBAR5MPRG | PCI Base Address Mask Register 5 Program Register |
| 02C0 03C8 | PCIBAR0PRG | PCI Base Address Register 0 Program Register |
| 02C0 03CC | PCIBAR1PRG | PCI Base Address Register 1 Program Register |
| 02C0 03D0 | PCIBAR2PRG | PCI Base Address Register 2 Program Register |
| 02C0 03D4 | PCIBAR3PRG | PCI Base Address Register 3 Program Register |
| 02C0 03D8 | PCIBAR4PRG | PCI Base Address Register 4 Program Register |
| 02C0 03DC | PCIBAR5PRG | PCI Base Address Register 5 Program Register |
| 02C0 03E0 | PCIBAR0TRLPRG | PCI Base Address Translation Register 0 Program Register |
| 02C0 03E4 | PCIBAR1TRLPRG | PCI Base Address Translation Register 1 Program Register |
| 02C0 03E8 | PCIBAR2TRLPRG | PCI Base Address Translation Register 2 Program Register |
| 02C0 03EC | PCIBAR3TRLPRG | PCI Base Address Translation Register 3 Program Register |
| 02C0 03F0 | PCIBAR4TRLPRG | PCI Base Address Translation Register 4 Program Register |
| 02C0 03F4 | PCIBAR5TRLPRG | PCI Base Address Translation Register 5 Program Register |
| 02C0 03F8 | PCIBASENPRG | PCI Base En Prog Register |
| 02C0 03FC - 02C0 03FF | - | Reserved |

**Table 7-101. PCI External Memory Space**

| HEX ADDRESS OFFSET | ACRONYM | REGISTER NAME |
|---|---|---|
| 4000 0000 - 407F FFFF | - | PCI Master Window 0 |
| 4080 0000 - 40FF FFFF | - | PCI Master Window 1 |
| 4100 0000 - 417F FFFF | - | PCI Master Window 2 |
| 4180 0000 - 41FF FFFF | - | PCI Master Window 3 |
| 4200 0000 - 427F FFFF | - | PCI Master Window 4 |
| 4280 0000 - 42FF FFFF | - | PCI Master Window 5 |
| 4300 0000 - 437F FFFF | - | PCI Master Window 6 |
| 4380 0000 - 43FF FFFF | - | PCI Master Window 7 |
| 4400 0000 - 447F FFFF | - | PCI Master Window 8 |
| 4480 0000 - 44FF FFFF | - | PCI Master Window 9 |
| 4500 0000 - 457F FFFF | - | PCI Master Window 10 |
| 4580 0000 - 45FF FFFF | - | PCI Master Window 11 |
| 4600 0000 - 467F FFFF | - | PCI Master Window 12 |
| 4680 0000 - 46FF FFFF | - | PCI Master Window 13 |
| 4700 0000 - 477F FFFF | - | PCI Master Window 14 |
| 4780 0000 - 47FF FFFF | - | PCI Master Window 15 |

**Table 7-101. PCI External Memory Space (continued)**

| HEX ADDRESS OFFSET | ACRONYM | REGISTER NAME |
|---|---|---|
| 4800 0000 - 487F FFFF | - | PCI Master Window 16 |
| 4880 0000 - 48FF FFFF | - | PCI Master Window 17 |
| 4900 0000 - 497F FFFF | - | PCI Master Window 18 |
| 4980 0000 - 49FF FFFF | - | PCI Master Window 19 |
| 4A00 0000 - 4A7F FFFF | - | PCI Master Window 20 |
| 4A80 0000 - 4AFF FFFF | - | PCI Master Window 21 |
| 4B00 0000 - 4B7F FFFF | - | PCI Master Window 22 |
| 4B80 0000 - 4BFF FFFF | - | PCI Master Window 23 |
| 4C00 0000 - 4C7F FFFF | - | PCI Master Window 24 |
| 4C80 0000 - 4CFF FFFF | - | PCI Master Window 25 |
| 4D00 0000 - 4D7F FFFF | - | PCI Master Window 26 |
| 4D80 0000 - 4DFF FFFF | - | PCI Master Window 27 |
| 4E00 0000 - 4E7F FFFF | - | PCI Master Window 28 |
| 4E80 0000 - 4EFF FFFF | - | PCI Master Window 29 |
| 4F00 0000 - 4F7F FFFF | - | PCI Master Window 30 |
| 4F80 0000 - 4FFF FFFF | - | PCI Master Window 31 |

### 7.16.3 PCI Electrical Data/Timing

Texas Instruments (TI) has performed the simulation and system characterization to ensure that the PCI peripheral meets all AC timing specifications as required by the *PCI Local Bus Specification* (version 2.3). The AC timing specifications are not reproduced here. For more information on the AC timing specifications, see section 4.2.3, Timing Specification (33 MHz timing), and section 7.6.4, Timing Specification (66 MHz timing), of the *PCI Local Bus Specification* (version 2.3). Note that the C6454 PCI peripheral only supports 3.3-V signaling.

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

### 7.17.1 GPIO Device-Specific Information

On the C6454 device, the GPIO peripheral pins GP[15:8] and GP[3:0] are muxed with the PCI and McBSP1 peripheral pins and the SYSCLK4 signal. For more detailed information on device/peripheral configuration and the C6454 device pin muxing, see Section 3, *Device Configuration*.

### 7.17.2 GPIO Peripheral Register Descriptions

**Table 7-102. GPIO Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 02B0 0008 | BINTEN | GPIO interrupt per bank enable register |
| 02B0 000C | - | Reserved |
| 02B0 0010 | DIR | GPIO Direction Register |
| 02B0 0014 | OUT_DATA | GPIO Output Data register |
| 02B0 0018 | SET_DATA | GPIO Set Data register |
| 02B0 001C | CLR_DATA | GPIO Clear Data Register |
| 02B0 0020 | IN_DATA | GPIO Input Data Register |
| 02B0 0024 | SET_RIS_TRIG | GPIO Set Rising Edge Interrupt Register |
| 02B0 0028 | CLR_RIS_TRIG | GPIO Clear Rising Edge Interrupt Register |
| 02B0 002C | SET_FAL_TRIG | GPIO Set Falling Edge Interrupt Register |
| 02B0 0030 | CLR_FAL_TRIG | GPIO Clear Falling Edge Interrupt Register |
| 02B0 008C | - | Reserved |
| 02B0 0090 - 02B0 00FF | - | Reserved |
| 02B0 0100 - 02B0 3FFF | - | Reserved |

### 7.17.3 GPIO Electrical Data/Timing

**Table 7-103. Timing Requirements for GPIO Inputs(1) (2)**

(see Figure 7-74)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(GPIH)}$ | Pulse duration, GPIx high | 12P | | ns |
| 2 | $t_{w(GPIL)}$ | Pulse duration, GPIx low | 12P | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) The pulse width given is sufficient to generate a CPU interrupt or an EDMA event. However, if a user wants to have the DSP recognize the GPIx changes through software polling of the GPIO register, the GPIx duration must be extended to at least 24P to allow the DSP enough time to access the GPIO register through the CFGBUS.

**Table 7-104. Switching Characteristics Over Recommended Operating Conditions for GPIO Outputs(1)**

(see Figure 7-74)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{w(GPOH)}$ | Pulse duration, GPOx high | 36P - 8(2) | | ns |
| 4 | $t_{w(GPOL)}$ | Pulse duration, GPOx low | 36P - 8(2) | | ns |

(1) P = 1/CPU clock frequency in ns. For example, when running parts at 1000 MHz, use P = 1 ns.
(2) 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.

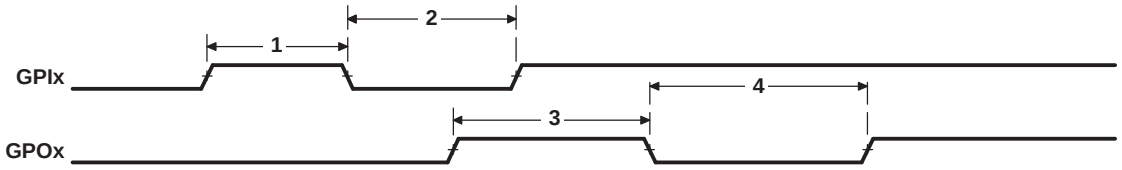



**Figure 7-74. GPIO Port Timing**

## 7.18 Emulation Features and Capability

### 7.18.1 Advanced Event Triggering (AET)

The C6454 device supports Advanced Event Triggering (AET). This capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- **Hardware Program Breakpoints:** specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- **Data Watchpoints:** specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- **Counters:** count the occurrence of an event or cycles for performance monitoring.
- **State Sequencing:** allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

For more information on AET, see the following documents:

*Using Advanced Event Triggering to Find and Fix Intermittent Real-Time Bugs* application report (literature number SPRA753)

*Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor Systems* application report (literature number SPRA387)

### 7.18.2 Trace

The C6454 device supports Trace. Trace is a debug technology that provides a detailed, historical account of application code execution, timing, and data accesses. Trace collects, compresses, and exports debug information for analysis. Trace works in real-time and does not impact the execution of the system.

For more information on board design guidelines for Trace Advanced Emulation, see the *Emulation and Trace Headers Technical Reference Manual* (literature number SPRU655).

### 7.18.3 IEEE 1149.1 JTAG

#### 7.18.3.1 JTAG Device-Specific Information

##### 7.18.3.1.1 IEEE 1149.1 JTAG Compatibility Statement

For maximum reliability, the C6454 DSP 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 DSP's internal emulation logic will always be properly initialized when this pin is not routed out. 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 an external pullup resistor on $\overline{\text{TRST}}$. When using this type of JTAG controller, assert $\overline{\text{TRST}}$ to initialize the DSP after powerup and externally drive $\overline{\text{TRST}}$ high before attempting any emulation or boundary scan operations.

### 7.18.4 JTAG Peripheral Register Descriptions

### 7.18.5 JTAG Electrical Data/Timing

**Table 7-105. Timing Requirements for JTAG Test Port**

(see Figure 7-75)

| NO. | | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(TCK)}$ | Cycle time, TCK | 35 | | ns |
| 3 | $t_{su(TDIV-TCKH)}$ | Setup time, TDI/TMS/$\overline{\text{TRST}}$ valid before TCK high | 6 | | ns |
| 4 | $t_{h(TCKH-TDIV)}$ | Hold time, TDI/TMS/$\overline{\text{TRST}}$ valid after TCK high | 9 | | ns |

**Table 7-106. Switching Characteristics Over Recommended Operating Conditions for JTAG Test Port**

(see Figure 7-75)

| NO. | PARAMETER | | -720<br>-850<br>A-1000/-1000 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 2 | $t_{d(TCKL-TDOV)}$ | Delay time, TCK low to TDO valid | -3 | 11 | ns |

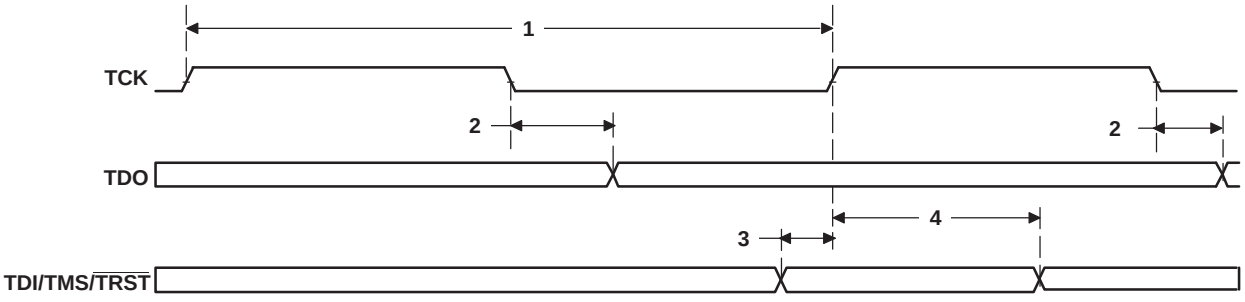



**Figure 7-75. JTAG Test-Port Timing**

# 8 Mechanical Data

## 8.1 Thermal Data

Table 8-1 shows the thermal resistance characteristics for the PBGA - CTZ/GTZ/ZTZ mechanical package.

**Table 8-1. Thermal Resistance Characteristics (S-PBGA Package) [CTZ/GTZ/ZTZ]**

| NO. | | | °C/W | AIR FLOW (m/s)(1) |
|---|---|---|---|---|
| 1 | $R\Theta_{JC}$ | Junction-to-case | 1.45 | N/A |
| 2 | $R\Theta_{JB}$ | Junction-to-board | 8.34 | N/A |
| 3 | $R\Theta_{JA}$ | Junction-to-free air | 16.1 | 0.00 |
| 4 | | | 13.0 | 1.0 |
| 5 | | | 11.9 | 2.0 |
| 6 | | | 10.7 | 3.0 |
| 7 | $Psi_{JT}$ | Junction-to-package top | 0.37 | 0.00 |
| | | | 0.89 | 1.0 |
| | | | 1.01 | 1.5 |
| | | | 1.17 | 3.00 |
| 8 | $Psi_{JB}$ | Junction-to-board | 7.6 | 0.00 |
| | | | 6.7 | 1.0 |
| | | | 6.4 | 1.5 |
| | | | 5.8 | 3.00 |

(1) m/s = meters per second

## 8.2 Packaging Information

The following packaging information reflects the most current released data available for the designated device(s). This data is subject to change without notice and without revision of this document.

## PACKAGING INFORMATION

| Orderable Device | Status (1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan (2) | Lead finish/ Ball material (6) | MSL Peak Temp (3) | Op Temp (°C) | Device Marking (4/5) | Samples |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TMS320C6454BCTZ7 | ACTIVE | FCBGA | CTZ | 697 | 44 | RoHS & Green | SNAGCU | Level-4-245C-72HR | 0 to 90 | TMS @2005 TI 320C6454CTZ 7 | Samples |
| TMS320C6454BCTZ8 | ACTIVE | FCBGA | CTZ | 697 | 44 | RoHS & Green | SNAGCU | Level-4-245C-72HR | 0 to 90 | TMS @2005 TI 320C6454BCTZ 8 | Samples |
| TMS320C6454BCTZA | ACTIVE | FCBGA | CTZ | 697 | 44 | RoHS & Green | SNAGCU | Level-4-245C-72HR | -40 to 105 | TMS @2005 TI 320C6454CTZ A1GHZ | Samples |
| TMS320C6454BGTZA | ACTIVE | FCBGA | GTZ | 697 | 44 | Non-RoHS & Green | SNPB | Level-4-220C-72 HR | -40 to 105 | TMS (@2005 TI, A1GHZ) 320C6454 GTZ | Samples |

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

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
**RoHS Exempt:** TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
**Green:** TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

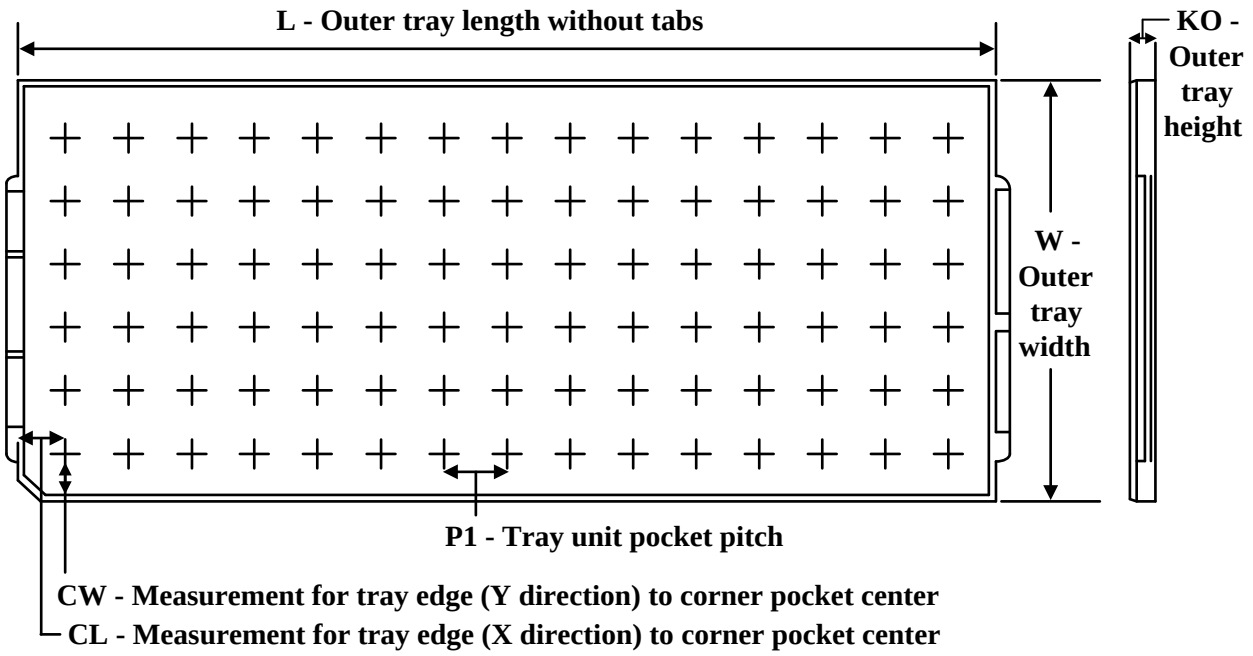
(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

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

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

(6) Lead 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.

## TRAY

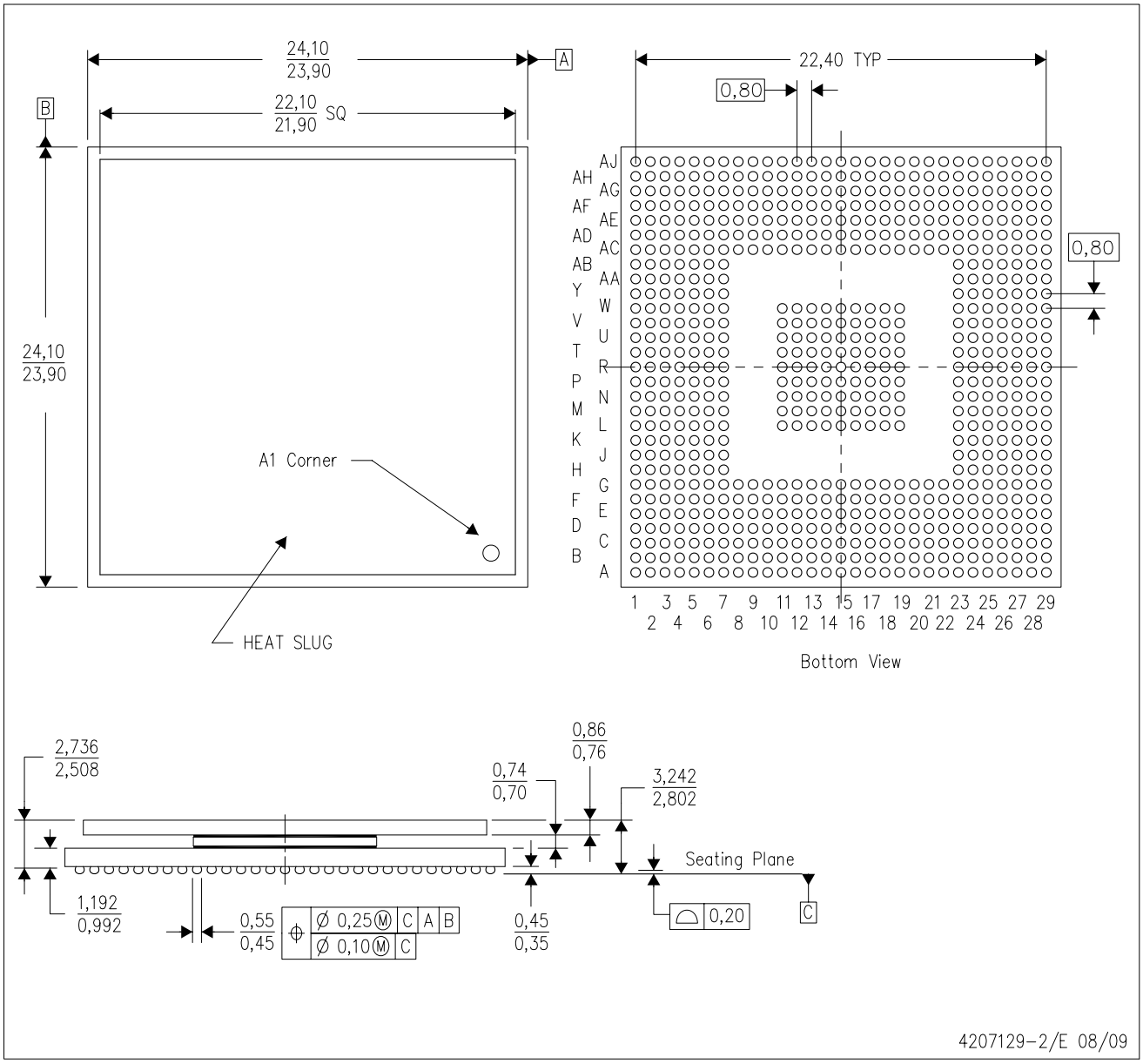



Chamfer on Tray corner indicates Pin 1 orientation of packed units.

*All dimensions are nominal

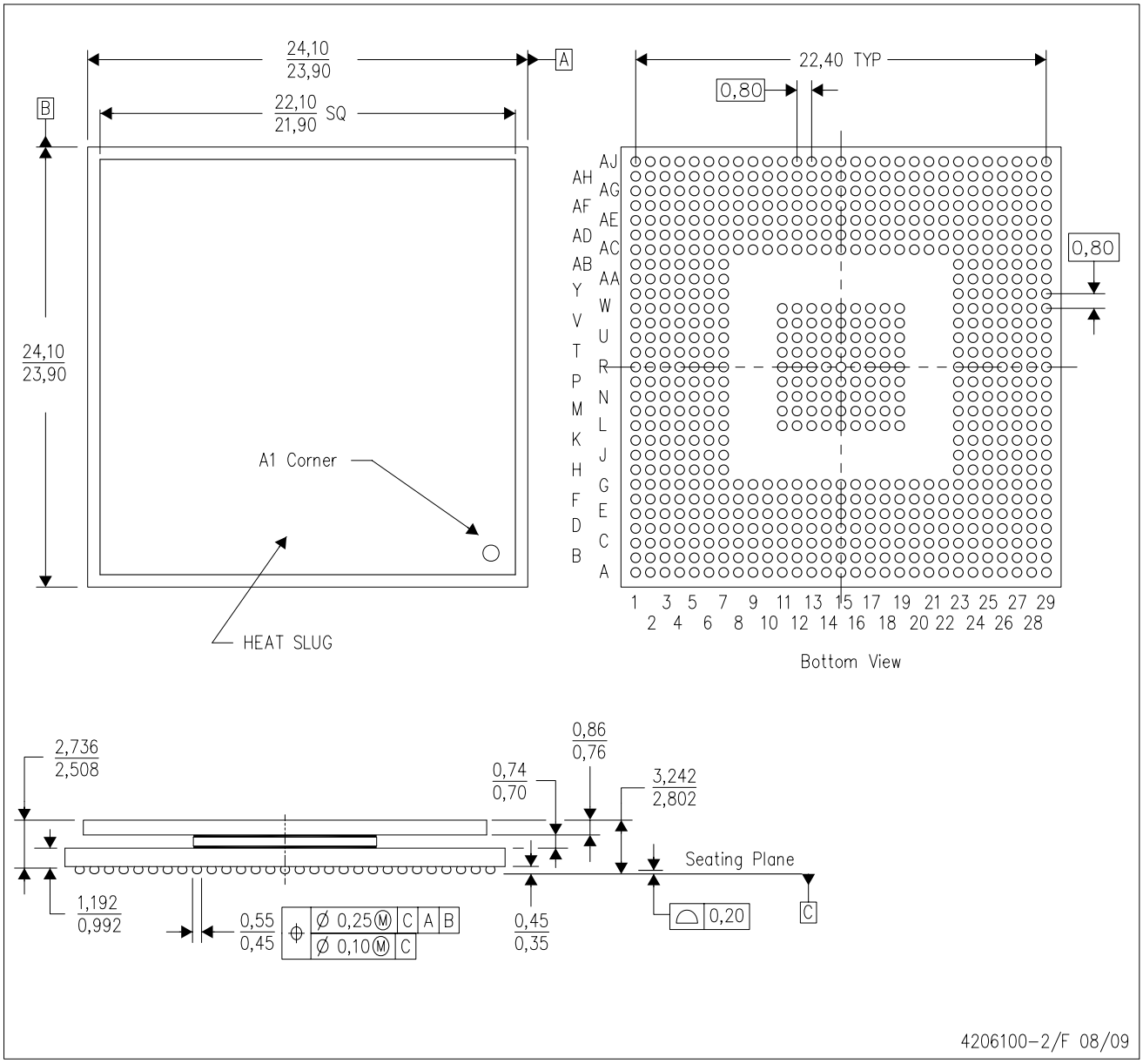
| Device | Package Name | Package Type | Pins | SPQ | Unit array matrix | Max temperature (°C) | L (mm) | W (mm) | K0 (µm) | P1 (mm) | CL (mm) | CW (mm) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMS320C6454BCTZ7 | CTZ | FCBGA | 697 | 44 | 4 X 11 | 150 | 315 | 135.9 | 7620 | 27.5 | 20 | 26.7 |
| TMS320C6454BCTZ8 | CTZ | FCBGA | 697 | 44 | 4 X 11 | 150 | 315 | 135.9 | 7620 | 27.5 | 20 | 26.7 |
| TMS320C6454BCTZA | CTZ | FCBGA | 697 | 44 | 4 X 11 | 150 | 315 | 135.9 | 7620 | 27.5 | 20 | 26.7 |
| TMS320C6454BGTZA | GTZ | FCBGA | 697 | 44 | 4 X 11 | 150 | 315 | 135.9 | 7620 | 27.5 | 20 | 26.7 |

CTZ (S–PBGA–N697) PLASTIC BALL GRID ARRAY



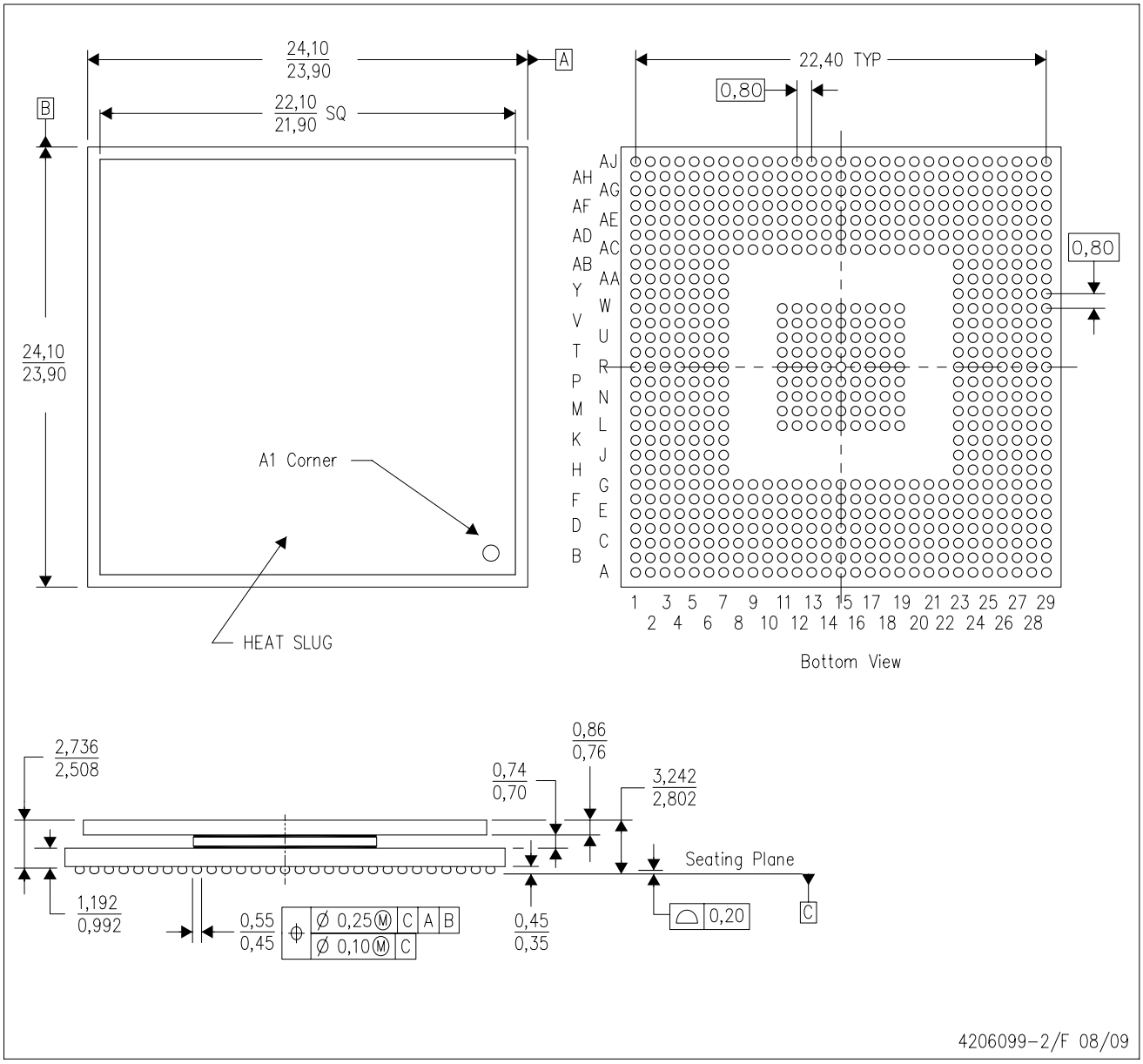

NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package with heat slug (HSL).
D. Flip chip application only.
E. Pb–free die bump and solder ball.

ZTZ (S–PBGA–N697) PLASTIC BALL GRID ARRAY




NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package with heat slug (HSL).
D. Flip chip application only.
E. This is a Pb–free solder ball design.

GTZ (S–PBGA–N697) PLASTIC BALL GRID ARRAY




NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package with heat slug (HSL).
D. Flip chip application only.
E. This is leaded solder ball design.

# 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 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

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

Click below to explore more details on WIN SOURCE:

- View TMS320C6454BCTZ8 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