# THE DATASHEET OF

# TMS320DM6446BZWT8

# TMS320DM6446
# Digital Media System-on-Chip

**Check for Samples: TMS320DM6446**

## 1 Digital Media System-on-Chip (DMSoC)

### 1.1 Features

- High-Performance Digital Media SoC
  - 513-, 594-, 810-MHz C64x+™ Clock Rates
  - 256.5-, 297-, 405-MHz ARM926EJ-S™ Clock Rates
  - Eight 32-Bit C64x+ Instructions/Cycle
  - 4104, 4752, 6480 C64x+ MIPS
  - Fully Software-Compatible With C64x / ARM9™
  - Extended Temperature Devices Available
- Advanced Very-Long-Instruction-Word (VLIW) TMS320C64x+™ DSP Core
  - Eight Highly Independent Functional Units
    - Six ALUs (32-/40-Bit), Each Supports Single 32-Bit, Dual 16-Bit, or Quad 8-Bit Arithmetic per Clock Cycle
    - Two Multipliers Support Four 16 x 16-Bit Multiplies (32-Bit Results) per Clock Cycle or Eight 8 x 8-Bit Multiplies (16-Bit Results) per Clock Cycle
  - Load-Store Architecture With Non-Aligned Support
  - 64 32-Bit General-Purpose Registers
  - Instruction Packing Reduces Code Size
  - All Instructions Conditional
  - Additional C64x+™ Enhancements
    - Protected Mode Operation
    - Exceptions Support for Error Detection and Program Redirection
    - Hardware Support for Modulo Loop Operation
- C64x+ Instruction Set Features
  - Byte-Addressable (8-/16-/32-/64-Bit Data)
  - 8-Bit Overflow Protection
  - Bit-Field Extract, Set, Clear
  - Normalization, Saturation, Bit-Counting
  - Compact 16-Bit Instructions
  - Additional Instructions to Support Complex Multiplies
- C64x+ L1/L2 Memory Architecture
  - 32K-Byte L1P Program RAM/Cache (Direct Mapped)
  - 80K-Byte L1D Data RAM/Cache (2-Way Set-Associative)
  - 64K-Byte L2 Unified Mapped RAM/Cache (Flexible RAM/Cache Allocation)
- ARM926EJ-S Core
  - Support for 32-Bit and 16-Bit (Thumb® Mode) Instruction Sets
  - DSP Instruction Extensions and Single Cycle MAC
  - ARM® Jazelle® Technology
  - EmbeddedICE-RT™ Logic for Real-Time Debug
- ARM9 Memory Architecture
  - 16K-Byte Instruction Cache
  - 8K-Byte Data Cache
  - 16K-Byte RAM
  - 8K-Byte ROM
- Embedded Trace Buffer™ (ETB11™) With 4KB Memory for ARM9 Debug
- Endianness: Little Endian for ARM and DSP
- Video Imaging Co-Processor (VICP)
- Video Processing Subsystem
  - Front End Provides:
    - CCD and CMOS Imager Interface
    - BT.601/BT.656 Digital YCbCr 4:2:2 (8-/16-Bit) Interface
    - Preview Engine for Real-Time Image Processing
    - Glueless Interface to Common Video Decoders
    - Histogram Module
    - Auto-Exposure, Auto-White Balance and Auto-Focus Module
    - Resize Engine
      - Resize Images From 1/4x to 4x
      - Separate Horizontal/Vertical Control

- Video Processing Subsystem (Continued)
  - Back End Provides:
    - Hardware On-Screen Display (OSD)
    - Four 54-MHz DACs for a Combination of
      - Composite NTSC/PAL Video
      - Luma/Chroma Separate Video (S-video)
      - Component (YPbPr or RGB) Video (Progressive)
    - Digital Output
      - 8-/16-bit YUV or up to 24-Bit RGB
      - HD Resolution
      - Up to 2 Video Windows
- External Memory Interfaces (EMIFs)
  - 32-Bit DDR2 SDRAM Memory Controller With 256M-Byte Address Space (1.8-V I/O)
    - Up to 167-MHz Controller (A-513, -594)
    - Up to 189-MHz Controller (-810)
  - Asynchronous16-Bit-Wide EMIF (EMIFA) With 128M-Byte Address Reach
    - Flash Memory Interfaces
      - NOR (8-/16-Bit-Wide Data)
      - NAND (8-/16-Bit-Wide Data)
- Flash Card Interfaces
  - Multimedia Card (MMC)/Secure Digital (SD) with Secure Data I/O (SDIO)
  - Compact Flash Controller With True IDE Mode
  - SmartMedia
- Enhanced Direct-Memory-Access (EDMA3) Controller (64 Independent Channels)
- Two 64-Bit General-Purpose Timers (Each Configurable as Two 32-Bit Timers)
- One 64-Bit Watch Dog Timer
- Three UARTs (One with RTS and CTS Flow Control)
- One Serial Peripheral Interface (SPI) With Two Chip-Selects
- Master/Slave Inter-Integrated Circuit ($I^2C$ Bus™)
- Audio Serial Port (ASP)
  - I2S
  - AC97 Audio Codec Interface
  - Standard Voice Codec Interface (AIC12)
- 10/100 Mb/s Ethernet MAC (EMAC)
  - IEEE 802.3 Compliant
  - Media Independent Interface (MII)
- VLYNQ™ Interface (FPGA Interface)
- Host Port Interface (HPI) with 16-Bit Multiplexed Address/Data
- USB Port With Integrated 2.0 PHY
  - USB 2.0 High-/Full-Speed (480-Mbps) Client
  - USB 2.0 High-/Full-/Low-Speed Host (Mini-Host, Supporting One External Device)
- Three Pulse Width Modulator (PWM) Outputs
- On-Chip ARM ROM Bootloader (RBL) to Boot From NAND Flash or UART
- ATA/ATAPI I/F (ATA/ATAPI-6 Specification)
- Individual Power-Saving Modes for ARM/DSP
- Flexible PLL Clock Generators
- IEEE-1149.1 (JTAG) Boundary-Scan-Compatible
- Up to 71 General-Purpose I/O (GPIO) Pins (Multiplexed With Other Device Functions)
- 361-Pin Pb-Free BGA Package (ZWT Suffix), 0.8-mm Ball Pitch
- 0.09-μm/6-Level Cu Metal Process (CMOS)
- 3.3-V and 1.8-V I/O, 1.2-V Internal (513, 594)
- 3.3-V and 1.8-V I/O, 1.2-V DAC and USB, 1.3-V Internal (810 *only*)
- Applications:
  - Digital Media
  - Networked Media Encode/Decode
  - Video Imaging

## 1.2 Description

The TMS320DM6446 (also referenced as DM6446) leverages TI's DaVinci™ technology to meet the networked media encode and decode application processing needs of next-generation embedded devices.

The DM6446 enables OEMs and ODMs to quickly bring to market devices featuring robust operating systems support, rich user interfaces, high processing performance, and long battery life through the maximum flexibility of a fully integrated mixed processor solution.

The dual-core architecture of the DM6446 provides benefits of both DSP and Reduced Instruction Set Computer (RISC) technologies, incorporating a high-performance TMS320C64x+™ DSP core and an ARM926EJ-S core.

The ARM926EJ-S is a 32-bit RISC processor core that performs 32-bit or 16-bit instructions and processes 32-bit, 16-bit, or 8-bit data. The core uses pipelining so that all parts of the processor and memory system can operate continuously.

The ARM core incorporates:

- A coprocessor 15 (CP15) and protection module
- Data and program Memory Management Units (MMUs) with table look-aside buffers.
- Separate 16K-byte instruction and 8K-byte data caches. Both are four-way associative with virtual index virtual tag (VIVT).

The TMS320C64x+™ DSPs are the highest-performance fixed-point DSP generation in the TMS320C6000™ DSP platform. It is based on an enhanced version of the second-generation high-performance, advanced very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI), making these DSP cores an excellent choice for digital media applications. The C64x is a code-compatible member of the C6000™ DSP platform. The TMS320C64x+ DSP is an enhancement of the C64x+™ DSP with added functionality and an expanded instruction set.

Any reference to the C64x™ DSP or C64x™ CPU also applies, unless otherwise noted, to the C64x+™ DSP and C64x+™ CPU, respectively.

With performance of up to 6480 million instructions per second (MIPS) at a clock rate of 810 MHz, the C64x+ core offers solutions to high-performance DSP programming challenges. The DSP core possesses the operational flexibility of high-speed controllers and the numerical capability of array processors. The C64x+ DSP core processor has 64 general-purpose registers of 32-bit word length and eight highly independent functional units—two multipliers for a 32-bit result and six arithmetic logic units (ALUs). The eight functional units include instructions to accelerate the performance in video and imaging applications. The DSP core can produce four 16-bit multiply-accumulates (MACs) per cycle for a total of 3240 million MACs per second (MMACS), or eight 8-bit MACs per cycle for a total of 6480 MMACS. For more details on the C64x+ DSP, see the *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* (literature number SPRU732).

The DM6446 also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals similar to the other C6000 DSP platform devices. The DM6446 core uses a two-level cache-based architecture. The Level 1 program cache (L1P) is a 256K-bit direct mapped cache and the Level 1 data cache (L1D) is a 640K-bit 2-way set-associative cache. The Level 2 memory/cache (L2) consists of an 512K-bit memory space that is shared between program and data space. L2 memory can be configured as mapped memory, cache, or combinations of the two.

The peripheral set includes: 2 configurable video ports; a 10/100 Mb/s Ethernet MAC (EMAC) with a Management Data Input/Output (MDIO) module; an inter-integrated circuit (I2C) Bus interface; one audio serial port (ASP); 2 64-bit general-purpose timers each configurable as 2 independent 32-bit timers; 1 64-bit watchdog timer; up to 71-pins of general-purpose input/output (GPIO) with programmable interrupt/event generation modes, multiplexed with other peripherals; 3 UARTs with hardware handshaking support on 1 UART; 3 pulse width modulator (PWM) peripherals; and 2 external memory interfaces: an asynchronous external memory interface (EMIFA) for slower memories/peripherals, and a higher speed synchronous memory interface for DDR2.

The DM6446 device includes a Video Processing Subsystem (VPSS) with two configurable video/imaging peripherals: 1 Video Processing Front-End (VPFE) input used for video capture, 1 Video Processing Back-End (VPBE) output with imaging co-processor (VICP) used for display.

The Video Processing Front-End (VPFE) is comprised of a CCD Controller (CCDC), a Preview Engine (Previewer), Histogram Module, Auto-Exposure/White Balance/Focus Module (H3A), and Resizer. The CCDC is capable of interfacing to common video decoders, CMOS sensors, and Charge Coupled Devices (CCDs). The Previewer is a real-time image processing engine that takes raw imager data from a CMOS sensor or CCD and converts from an RGB Bayer Pattern to YUV4:2:2. The Histogram and H3A modules provide statistical information on the raw color data for use by the DM6446. The Resizer accepts image data for separate horizontal and vertical resizing from 1/4x to 4x in increments of 256/N, where N is between 64 and 1024.

The Video Processing Back-End (VPBE) is comprised of an On-Screen Display Engine (OSD) and a Video Encoder (VENC). The OSD engine is capable of handling 2 separate video windows and 2 separate OSD windows. Other configurations include 2 video windows, 1 OSD window, and 1 attribute window allowing up to 8 levels of alpha blending. The VENC provides four analog DACs that run at 54 MHz, providing a means for composite NTSC/PAL video, S-Video, and/or Component video output. The VENC also provides up to 24 bits of digital output to interface to RGB888 devices. The digital output is capable of 8/16-bit BT.656 output and/or CCIR.601 with separate horizontal and vertical syncs.

The Ethernet Media Access Controller (EMAC) provides an efficient interface between the DM644x and the network. The DM6446 EMAC support both 10Base-T and 100Base-TX, or 10 Mbits/second (Mbps) and 100 Mbps in either half- or full-duplex mode, with hardware flow control and quality of service (QOS) support.

The Management Data Input/Output (MDIO) module continuously polls all 32 MDIO addresses in order to enumerate all PHY devices in the system. Once a PHY candidate has been selected by the ARM, the MDIO module transparently monitors its link state by reading the PHY status register. Link change events are stored in the MDIO module and can optionally interrupt the ARM, allowing the ARM to poll the link status of the device without continuously performing costly MDIO accesses.

The HPI, I2C, SPI, USB2.0, and VLYNQ ports allow DM6446 to easily control peripheral devices and/or communicate with host processors. The DM6446 also provides multimedia card support, MMC/SD, with SDIO support.

The DM6446 also includes a Video/Imaging Co-processor (VICP) to offload many video and imaging processing tasks from the DSP core, making more DSP MIPS available for common video and imaging algorithms. For more information on the VICP enhanced codecs, such as H.264 and MPEG4, please contact your nearest TI sales representative.

The rich peripheral set provides the ability to control external peripheral devices and communicate with external processors. For details on each of the peripherals, see the related sections later in this document and the associated peripheral reference guides listed in Section 2.8.3.1, *Related Documentation From Texas Instruments*.

The DM6446 has a complete set of development tools for both the ARM and DSP. These include C compilers, a DSP assembly optimizer to simplify programming and scheduling, and a Windows™ debugger interface for visibility into source code execution.

## 1.3 Functional Block Diagram

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

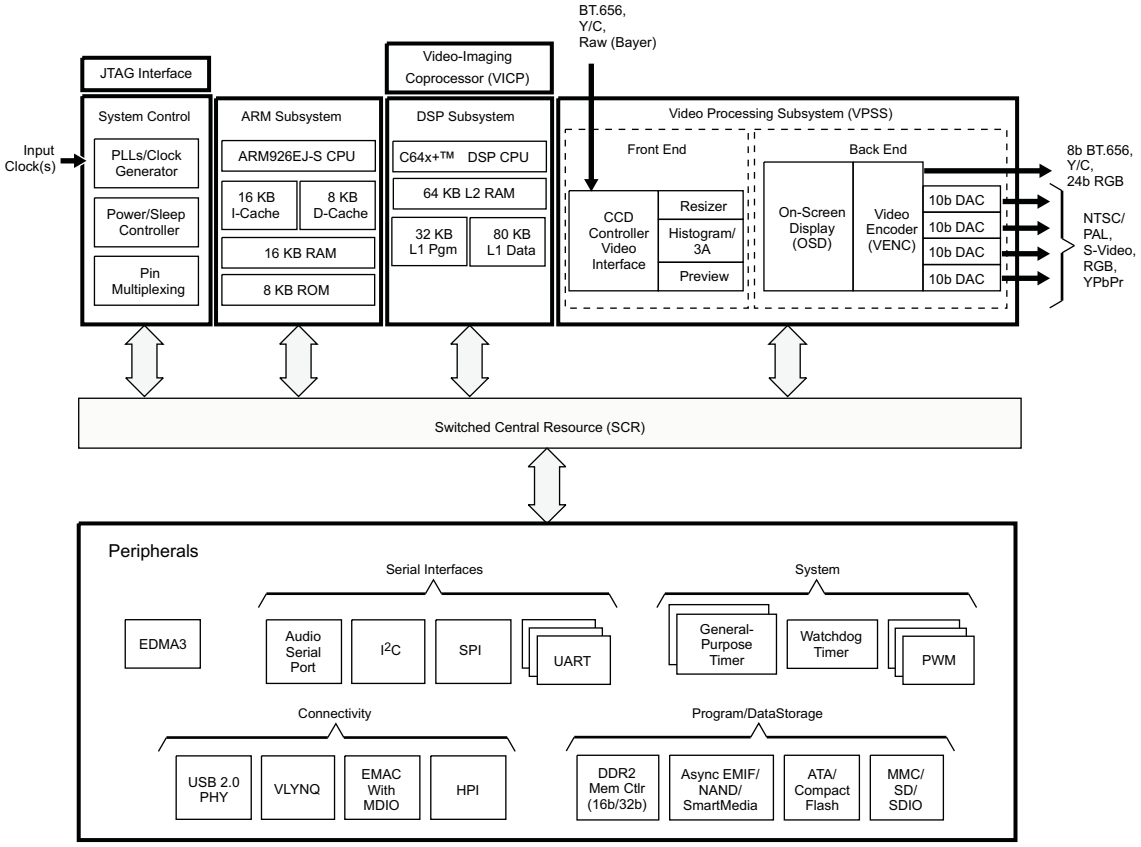



**Figure 1-1. TMS320DM6446 Functional Block Diagram**

## 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 SPRS283G device-specific data manual to make it an SPRS283H revision.

**Scope:** Added information/data on silicon revision 2.3.
Applicable updates to the DM644x device family, specifically relating to the TMS320DM6446 device, have been incorporated.

### TMS320DM6446 Revision History

| SEE | ADDITIONS/MODIFICATIONS/DELETIONS |
|---|---|
| Global | • Added information/data on silicon revision 2.3 |
| Section 2.1 Device Characteristics | Table 2-1, Characteristics of the Processor:<br>• Updated/changed "C64x+ Megamodule Revision" for silicon revision 2.3<br>• Updated/changed "JTAG BSDL_ID" for silicon revision 2.3 |
| Section 2.8.3.1 Related Documentation From Texas Instruments | • Updated/changed list of reference documents |
| Section 3.3.1.1 BOOTCFG Register Description | • Updated/changed the address of the BOOTCFG register from "0x01C4 000A" to "0x01C4 0014" [Update made in Revision G] |
| Section 3.3.3 DSP Boot | • Added Section 3.3.3.1, Host-Boot Mode |
| Section 3.5.1 Switched Central Resource (SCR) Bus Priorities | Figure 3-6, MSTPRI1 Register:<br>• Updated/changed the default value of bits 22:20 from "R-100" to "R/W-100" |
| Section 3.5.4 PINMUX0 Register Description | Figure 3-7, PINMUX0 Register:<br>• Updated/changed the default value of bits 4:0 from "R/W-LLLL" to "R/W-LLLL**L**" |
| Section 6.3.1.3 DM6446 Power and Clock Domains | Figure 6-6, PLL1 and PLL2 Clock Domain Block Diagram:<br>• PLL Controller 2: Updated/changed "PLLDIV1 (**/1**)" to "PLLDIV1 (**/10**)" |
| Section 6.6.3 Clock PLL Electrical Data/Timing (Input and Output Clocks) | Table 6-19, Switching Characteristics Over Recommended Operating Conditions for CLK_OUT1:<br>• Parameter 1 [$t_c$]: Added "ns" in the UNIT column |
| Section 6.10.1.2 EMIFA Electrical Data/Timing | Table 6-35, Switching Characteristics Over Recommended Operating Conditions for Asynchronous Memory Cycles for EMIFA Module:<br>• Parameter 24 [$t_{w(EMWEL)}$]: Added "ns" in the UNIT column |
| Section 6.12.1 MMC/SD/SDIO Peripheral Description(s) | Table 6-43, MMC/SD/SDIO Register Descriptions:<br>• Updated/changed 0x01E1 0064 from "SDIO" to "SDIOCTL (SDIO Control Register)"<br>• Updated/changed 0x01E1 0068 from "SDIO" to "SDIOST0 (SDIO Status Register 0)"<br>• Updated/changed 0x01E1 006C from "SDIO" to "SDIOIEN (SDIO Interrupt Enable Register)"<br>• Updated/changed 0x01E1 0070 from "Reserved" to "SDIOIST (SDIO Interrupt Status Register)" |

**TMS320DM6446 Revision History (continued)**

| SEE | ADDITIONS/MODIFICATIONS/DELETIONS |
|---|---|
| Section 6.15.3 USB2.0 Electrical Data/Timing | Table 6-72, Switching Characteristics Over Recommended Operating Conditions for USB2.0:<br>• Parameter 8, [$t_{w(EOPR)}$]: Added footnote about being accepted as valid EOP |
| Section 6.25.1 JTAG Peripheral Register Description(s) – JTAG ID Register | • Updated/changed "The JTAG ID register is a read-only register ..." paragraph<br>Figure 6-79, JTAG ID Register Description - DM6446 Register Value - 0xXB70 002F:<br>• Updated/changed footnote<br>Table 6-114, JTAG ID Register Selection Bit Descriptions:<br>• Updated/changed DESCRIPTION of Bits 31:28 (VARIANT) |

# 2 Device Overview

## 2.1 Device Characteristics

Table 2-1 provides an overview of the TMS320DM6446 SoC. The table shows significant features of the device, including the capacity of on-chip RAM, peripherals, internal peripheral bus frequency relative to the C64x+ DSP, and the package type with pin count.

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

| HARDWARE FEATURES | | DM6446 |
|---|---|---|
| Peripherals<br>Not all peripherals pins are available at the same time. (For more details, see Section 3, *Device Configurations*). | DDR2 Memory Controller | DDR2 (16/32-bit bus width) |
| | Asynchronous EMIF (EMIFA) | Asynchronous (8/16-bit bus width) RAM, Flash (NOR, NAND) |
| | Flash Cards | Compact Flash<br>MMC/SD with secure data input/output (SDIO)<br>SmartMedia/xD |
| | EDMA3 | 64 independent channels<br>8 QDMA channels |
| | Timers | 2 64-Bit General Purpose (each configurable as 2 separate 32-bit timers)<br>1 64-Bit Watchdog |
| | UART | 3 (one with RTS and CTS flow control) |
| | SPI | 1 (supports 2 slave devices) |
| | $I^2C$ | 1 (Master/Slave) |
| | Audio Serial Port [ASP] | 1 |
| | 10/100 Ethernet MAC with Management Data Input/Output | 1 |
| | VLYNQ | 1 |
| | HPI | 1 (16-bit multiplexed address/data) |
| | General-Purpose Input/Output Port | Up to 71 |
| | PWM | 3 outputs |
| | ATA/CF | 1 (ATA/ATAPI-6) |
| | Configurable Video Ports | 1 Input (VPFE)<br>1 Output (VPBE) |
| | USB 2.0 | High Speed Device<br>High Speed Host |
| | VICP | 1 |
| On-Chip Memory | Size (Bytes) | 160KB RAM, 8KB ROM |
| | Organization | DSP<br>• 32KB L1 Program (L1P)/Cache (up to 32KB)<br>• 80KB L1 Data (L1D)/Cache (up to 32KB)<br>• 64KB Unified Mapped RAM/Cache (L2)<br>ARM<br>• 16KB I-cache<br>• 8KB D-cache<br>• 16KB RAM<br>• 8KB ROM |
| CPU ID + CPU Rev ID | Control Status Register (CSR.[31:16]) | 0x1000 |
| C64x+ Megamodule Revision | Revision ID Register (MM_REVID[15:0])<br>(address location: 0x0181 2000) | 0x0000 (Silicon Revisions 1.3 and earlier)<br>0x0003 (Silicon Revisions 2.1 and later) |
| JTAG BSDL_ID | JTAGID Register<br>(address location: 0x01C4 0028) | 0x0B70 002F (Silicon Revisions 1.3 and earlier)<br>0x1B70 002F (Silicon Revisions 2.1 and later) |

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

| HARDWARE FEATURES | | | DM6446 |
|---|---|---|---|
| CPU Frequency | MHz | DM6446 - 810 | DSP 810 MHz |
| | | | ARM 405 MHz |
| | | DM6446 - 594 | DSP 594 MHz |
| | | | ARM 297 MHz |
| | | DM6446A - 513 | DSP 513 MHz |
| | | | ARM 256.5 MHz |
| Cycle Time | ns | DM6446 - 810 | DSP 1.23 ns |
| | | | ARM 2.47 ns |
| | | DM6446 - 594 | DSP 1.68 ns |
| | | | ARM 3.37 ns |
| | | DM6446A - 513 | DSP 1.95 ns |
| | | | ARM 3.90 ns |
| Voltage | Core (V) | | 1.2 V (-594, A-513) |
| | | | 1.3 V (-810) |
| | I/O (V) | | 1.8 V, 3.3 V (-810, -594, A-513) |
| PLL Options | CLKIN frequency multiplier (27 MHz reference) | | x1 (Bypass), x30 (-810) |
| | | | x1 (Bypass), x22 (-594, A-513) |
| BGA Package | 16 x 16 mm<br>ball finish SnAgCu | | 361-Pin BGA (ZWT) |
| Process Technology | μm | | 0.09 μm |
| Product Status(1) | Product Preview (PP),<br>Advance Information (AI),<br>or Production Data (PD) | | PD |

(1) 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.

## 2.2 Device Compatibility

The ARM926EJ-S RISC CPU is compatible with other ARM9 CPUs from ARM Holdings plc.

The C64x+ DSP core is code-compatible with the C6000™ DSP platform and supports features of the C64x DSP family.

## 2.3 ARM Subsystem

The ARM Subsystem is designed to give the ARM926EJ-S (ARM9) master control of the device. In general, the ARM is responsible for configuration and control of the device; including the DSP Subsystem, the VPSS Subsystem, and a majority of the peripherals and external memories.

The ARM Subsystem includes the following features:

- ARM926EJ-S RISC processor
- ARMv5TEJ (32/16-bit) instruction set
- Little endian
- Co-Processor 15 (CP15)
- MMU
- 16KB Instruction cache
- 8KB Data cache
- Write Buffer
- 16KB Internal RAM (32-bit-wide access)
- 8KB Internal ROM (ARM bootloader for non-EMIFA boot options)
- Embedded Trace Module and Embedded Trace Buffer (ETM/ETB)

- ARM Interrupt controller
- PLL Controller
- Power and Sleep Controller (PSC)
- System Module

### 2.3.1 ARM926EJ-S RISC CPU

The ARM Subsystem integrates the ARM926EJ-S processor. The ARM926EJ-S processor is a member of ARM9 family of general-purpose microprocessors. This processor is targeted at multi-tasking applications where full memory management, high performance, low die size, and low power are all important. The ARM926EJ-S processor supports the 32-bit ARM and 16 bit THUMB instruction sets, enabling the user to trade off between high performance and high code density. Specifically, the ARM926EJ-S processor supports the ARMv5TEJ instruction set, which includes features for efficient execution of Java byte codes, providing Java performance similar to Just in Time (JIT) Java interpreter, but without associated code overhead.

The ARM926EJ-S processor supports the ARM debug architecture and includes logic to assist in both hardware and software debug. The ARM926EJ-S processor has a Harvard architecture and provides a complete high performance subsystem, including:

- ARM926EJ -S integer core
- CP15 system control coprocessor
- Memory Management Unit (MMU)
- Separate instruction and data Caches
- Write buffer
- Separate instruction and data Tightly-Coupled Memories (TCMs) [internal RAM] interfaces
- Separate instruction and data AHB bus interfaces
- Embedded Trace Module and Embedded Trace Buffer (ETM/ETB)

For more complete details on the ARM9, refer to the *ARM926EJ-S Technical Reference Manual*, available at http://www.arm.com.

### 2.3.2 CP15

The ARM926EJ-S system control coprocessor (CP15) is used to configure and control instruction and data caches, Tightly-Coupled Memories (TCMs), Memory Management Unit (MMU), and other ARM subsystem functions. The CP15 registers are programmed using the MRC and MCR ARM instructions, when the ARM in a privileged mode such as supervisor or system mode.

### 2.3.3 MMU

The ARM926EJ-S MMU provides virtual memory features required by operating systems such as Linux®, Windows® CE, Ultron®, ThreadX®, etc. A single set of two level page tables stored in main memory is used to control the address translation, permission checks and memory region attributes for both data and instruction accesses. The MMU uses a single unified Translation Lookaside Buffer (TLB) to cache the information held in the page tables. The MMU features are:

- Standard ARM architecture v4 and v5 MMU mapping sizes, domains and access protection scheme.
- Mapping sizes are:
  - 1MB (sections)
  - 64KB (large pages)
  - 4KB (small pages)
  - 1KB (tiny pages)
- Access permissions for large pages and small pages can be specified separately for each quarter of the page (subpage permissions)
- Hardware page table walks

- Invalidate entire TLB, using CP15 register 8
- Invalidate TLB entry, selected by MVA, using CP15 register 8
- Lockdown of TLB entries, using CP15 register 10

### 2.3.4 Caches and Write Buffer

The size of the Instruction Cache is 16KB, Data cache is 8KB. Additionally, the Caches have the following features:

- Virtual index, virtual tag, and addressed using the Modified Virtual Address (MVA)
- Four-way set associative, with a cache line length of eight words per line (32-bytes per line) and with two dirty bits in the Dcache
- Dcache supports write-through and write-back (or copy back) cache operation, selected by memory region using the C and B bits in the MMU translation tables.
- Critical-word first cache refilling
- Cache lockdown registers enable control over which cache ways are used for allocation on a line fill, providing a mechanism for both lockdown, and controlling cache corruption
- Dcache stores the Physical Address TAG (PA TAG) corresponding to each Dcache entry in the TAG RAM for use during the cache line write-backs, in addition to the Virtual Address TAG stored in the TAG RAM. This means that the MMU is not involved in Dcache write-back operations, removing the possibility of TLB misses related to the write-back address.
- Cache maintenance operations provide efficient invalidation of, the entire Dcache or Icache, regions of the Dcache or Icache, and regions of virtual memory.

The write buffer is used for all writes to a noncachable bufferable region, write-through region and write misses to a write-back region. A separate buffer is incorporated in the Dcache for holding write-back for cache line evictions or cleaning of dirty cache lines. The main write buffer has 16-word data buffer and a four-address buffer. The Dcache write-back has eight data word entries and a single address entry.

### 2.3.5 Tightly Coupled Memory (TCM)

ARM internal RAM is provided for storing real-time and performance-critical code/data and the Interrupt Vector table. ARM internal ROM enables non-EMIFA boot options, such as NAND and UART. The RAM and ROM memories interfaced to the ARM926EJ-S via the tightly coupled memory interface that provides for separate instruction and data bus connections. Since the ARM TCM does not allow instructions on the D-TCM bus or data on the I-TCM bus, an arbiter is included so that both data and instructions can be stored in the internal RAM/ROM. The arbiter also allows accesses to the RAM/ROM from extra-ARM sources (e.g., EDMA3 or other masters). The ARM926EJ-S has built-in DMA support for direct accesses to the ARM internal memory from a non-ARM master. Because of the time-critical nature of the TCM link to the ARM internal memory, all accesses from non-ARM devices are treated as DMA transfers.

Instruction and Data accesses are differentiated via accessing different memory map regions, with the instruction region from 0x0000 through 0x7FFF and data from 0x8000 through 0xFFFF. The instruction region at 0x0000 and data region at 0x8000 map to the same physical 16KB TCM RAM. Placing the instruction region at 0x0000 is necessary to allow the ARM Interrupt Vector table to be placed at 0x0000, as required by the ARM architecture. The internal 16-KB RAM is split into two physical banks of 8KB each, which allows simultaneous instruction and data accesses to be accomplished if the code and data are in separate banks.

### 2.3.6 Advanced High-Performance Bus (AHB)

The ARM Subsystem uses the AHB port of the ARM926EJ-S to connect the ARM to the Config bus and the external memories. Arbiters are employed to arbitrate access to the separate D-AHB and I-AHB by the Config Bus and the external memories bus.

### 2.3.7 Embedded Trace Macrocell (ETM) and Embedded Trace Buffer (ETB)

To support real-time trace, the ARM926EJ-S processor provides an interface to enable connection of an Embedded Trace Macrocell (ETM). The ARM926ES-J Subsystem in the DM6446 also includes the Embedded Trace Buffer (ETB). The ETM consists of two parts:

- Trace Port provides real-time trace capability for the ARM9.
- Triggering facilities provide trigger resources, which include address and data comparators, counter, and sequencers.

The DM6446 trace port is not pinned out and is instead only connected to the Embedded Trace Buffer. The ETB has a 4KB buffer memory. ETB enabled debug tools are required to read/interpret the captured trace data.

### 2.3.8 ARM Memory Mapping

The ARM memory map is shown in Section 2.5, *Memory Map Summary*, of this document. The ARM has access to memories shown in the following sections.

#### 2.3.8.1 ARM Internal Memories

The ARM has access to the following ARM internal memories:

- 16KB ARM Internal RAM on TCM interface, logically separated into two 8KB pages to allow simultaneous access on any given cycle if there are separate accesses for code (I-TCM bus) and data (D-TCM) to the different memory regions.
- 8KB ARM Internal ROM

#### 2.3.8.2 External Memories

The ARM has access to the following external memories:

- DDR2 Synchronous DRAM
- Asynchronous EMIF / NOR Flash / NAND Flash
- ATA/CF
- Flash card devices:
  - MMC/SD with SDIO
  - xD
  - SmartMedia

#### 2.3.8.3 DSP Memories

The ARM has access to the following DSP memories:

- L2 RAM
- L1P RAM
- L1D RAM

#### 2.3.8.4 ARM-DSP Integration

DM6446 ARM and DSP integration features are as follows:

- DSP visibility from ARM's memory map, see Section 2.5, *Memory Map Summary*, for details
- Boot Modes for DSP - see Section 3 (*Device Configurations*) and Section 3.3.3 (*DSP Boot*) for details
- ARM control of DSP boot / reset - see Section 3 (*Device Configurations*) and Section 3.3.2 (*ARM Boot*) for details
- ARM control of DSP isolation and powerdown / powerup - see Section 3, *Device Configurations*, for details
- ARM and DSP Interrupts - see Section 6.7.1, *ARM CPU Interrupts*, and Section 6.7.2, *DSP Interrupts*, for details

### 2.3.9 Peripherals

The ARM9 has access to all of the peripherals on the DM6446 device with the exception of the VICP.

### 2.3.10 PLL Controller (PLLC)

The ARM Subsystem includes the PLL Controller. The PLL Controller contains a set of registers for configuring DM6446's two internal PLLs (PLL1 and PLL2). The PLL Controller provides the following configuration and control:

- PLL Bypass Mode
- Set PLL multiplier parameters
- Set PLL divider parameters
- PLL power down
- Oscillator power down

The PLLs are briefly described in this document in Section 6.6, *Clock PLLs*. For more detailed information on the PLLs and PLL Controller register descriptions, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

### 2.3.11 Power and Sleep Controller (PSC)

The ARM Subsystem includes the Power and Sleep Controller (PSC). Through register settings accessible by the ARM9, the PSC provides two levels of power savings: peripheral/module clock gating and power domain shut-off. Brief details on the PSC are given in Section 6.3, *Power Supplies*. For more detailed information and complete register descriptions for the PSC, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

### 2.3.12 ARM Interrupt Controller (AINTC)

The ARM Interrupt Controller (AINTC) accepts device interrupts and maps them to either the ARM's IRQ (interrupt request) or FIQ (fast interrupt request). The ARM Interrupt Controller is briefly described in Section 6.7, *Interrupts*, of this document. For detailed information on the ARM Interrupt Controller, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

### 2.3.13 System Module

The ARM Subsystem includes the System module. The System module consists of a set of registers for configuring and controlling a variety of system functions. For details and register descriptions for the System module, see Section 3, *Device Configurations*, and see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

### 2.3.14 Power Management

DM6446 has several means of managing power consumption. There is extensive use of clock gating, which reduces the power used by global device clocks and individual peripheral clocks. Clock management can be utilized to reduce clock frequencies in order to reduce switching power. For more details on power management techniques, see Section 3 (*Device Configurations*), Section 6 (*Peripheral and Electrical Specifications*), and see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

DM6446 gives the programmer full flexibility to use any and all of the previously mentioned capabilities to customize an optimal power management strategy. Several typical power management scenarios are described in the following sections.

## 2.4 DSP Subsystem

The DSP Subsystem includes the following features:

- C64x+ DSP CPU
- 32KB L1 Program (L1P)/Cache (up to 32KB)
- 80KB L1 Data (L1D)/Cache (up to 32KB)
- 64KB Unified Mapped RAM/Cache (L2)
- Little endian

### *2.4.1 C64x+ DSP CPU 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, one 16 x 32 bit multiply, two 16 x 16 bit multiplies, two 16 x 32 bit multiplies, two 16 x 16 bit multiplies with add/subtract capabilities, four 8 x 8 bit multiplies, four 8 x 8 bit multiplies with add operations, and four 16 x 16 multiplies with add/subtract capabilities (including a complex multiply). There is also support for Galois field multiplication for 8-bit and 32-bit data. Many communications algorithms such as FFTs and modems require complex multiplication. The complex multiply (CMPY) instruction takes 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.

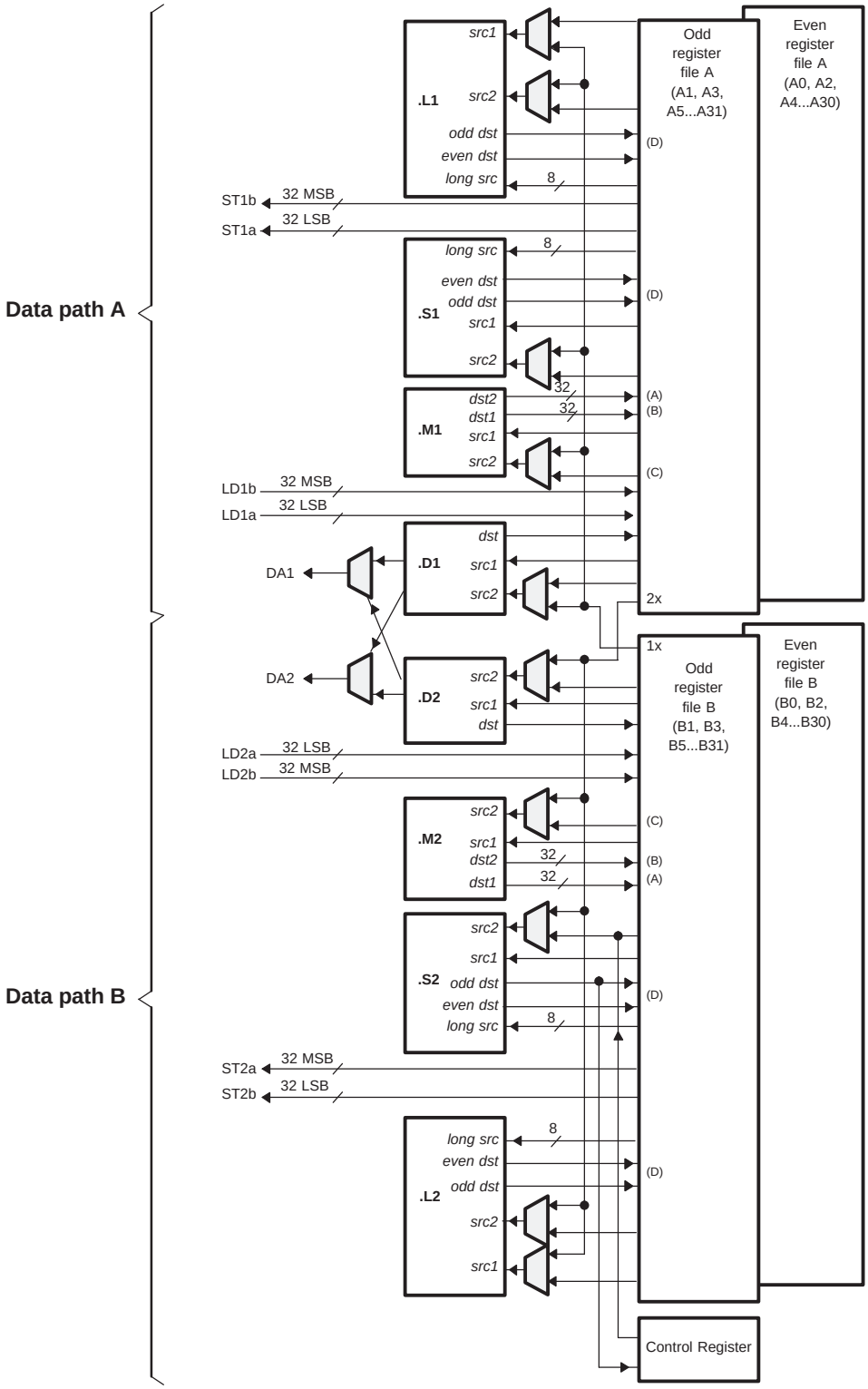
The C64x+ core enhances the .S unit in several ways. In the C64x core, dual 16-bit MIN2 and MAX2 comparisons were only available on the .L units. On the C64x+ core they are also available on the .S unit which increases the performance of algorithms that do searching and sorting. Finally, to increase 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 Enhancement** - As noted above, there are new instructions such as 32-bit multiplications, complex multiplications, packing, sorting, bit manipulation, and 32-bit Galois field multiplication.
- **Exceptions Handling** - Intended to aid the programmer in isolating bugs. The 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 Technical Overview* (literature number SPRU395)




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

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

### 2.4.2 DSP Memory Mapping

The DSP memory map is shown in Section 2.5, *Memory Map Summary*. Configuration of the control registers for DDR2, EMIFA, and ARM Internal RAM is supported by the ARM. The DSP has access to memories shown in the following sections.

#### 2.4.2.1 ARM Internal Memories

The DSP has access to the 16KB ARM Internal RAM on the ARM D-TCM interface (i.e., data only).

#### 2.4.2.2 External Memories

The DSP has access to the following External memories:

- DDR2 Synchronous DRAM
- Asynchronous EMIF / NOR Flash

#### 2.4.2.3 DSP Internal Memories

The DSP has access to the following DSP memories:

- L2 RAM
- L1P RAM
- L1D RAM

#### 2.4.2.4 C64x+ CPU

The C64x+ core uses a two-level cache-based architecture. The Level 1 Program cache (L1P) is 32 KB direct mapped cache and the Level 1 Data cache (L1D) is 80 KB 2-way set associated cache. The Level 2 memory/cache (L2) consists of a 64 KB memory space that is shared between program and data space. L2 memory can be configured as mapped memory, cache, or a combination of both.

Table 2-2 shows a memory map of the C64x+ CPU cache registers for the device.

**Table 2-2. C64x+ Cache Registers**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x0184 0000 | L2CFG | L2 Cache configuration register |
| 0x0184 0020 | L1PCFG | L1P Size Cache configuration register |
| 0x0184 0024 | L1PCC | L1P Freeze Mode Cache configuration register |
| 0x0184 0040 | L1DCFG | L1D Size Cache configuration register |
| 0x0184 0044 | L1DCC | L1D Freeze Mode Cache configuration register |
| 0x0184 0048 - 0x0184 0FFC | - | Reserved |
| 0x0184 1000 | EDMAWEIGHT | L2 EDMA3 access control register |
| 0x0184 1004 - 0x0184 1FFC | - | Reserved |
| 0x0184 2000 | L2ALLOC0 | L2 allocation register 0 |
| 0x0184 2004 | L2ALLOC1 | L2 allocation register 1 |
| 0x0184 2008 | L2ALLOC2 | L2 allocation register 2 |
| 0x0184 200C | L2ALLOC3 | L2 allocation register 3 |
| 0x0184 2010 - 0x0184 3FFF | - | Reserved |
| 0x0184 4000 | L2WBAR | L2 writeback base address register |
| 0x0184 4004 | L2WWC | L2 writeback word count register |
| 0x0184 4010 | L2WIBAR | L2 writeback invalidate base address register |
| 0x0184 4014 | L2WIWC | L2 writeback invalidate word count register |
| 0x0184 4018 | L2IBAR | L2 invalidate base address register |
| 0x0184 401C | L2IWC | L2 invalidate word count register |
| 0x0184 4020 | L1PIBAR | L1P invalidate base address register |
| 0x0184 4024 | L1PIWC | L1P invalidate word count register |

**Table 2-2. C64x+ Cache Registers (continued)**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x0184 4030 | L1DWIBAR | L1D writeback invalidate base address register |
| 0x0184 4034 | L1DWIWC | L1D writeback invalidate word count register |
| 0x0184 4038 | - | Reserved |
| 0x0184 4040 | L1DWBAR | L1D Block Writeback |
| 0x0184 4044 | L1DWWC | L1D Block Writeback |
| 0x0184 4048 | L1DIBAR | L1D invalidate base address register |
| 0x0184 404C | L1DIWC | L1D invalidate word count register |
| 0x0184 4050 - 0x0184 4FFF | - | Reserved |
| 0x0184 5000 | L2WB | L2 writeback all register |
| 0x0184 5004 | L2WBINV | L2 writeback invalidate all register |
| 0x0184 5008 | L2INV | L2 Global Invalidate without writeback |
| 0x0184 500C - 0x0184 5027 | - | Reserved |
| 0x0184 5028 | L1PINV | L1P Global Invalidate |
| 0x0184 502C - 0x0184 5039 | - | Reserved |
| 0x0184 5040 | L1DWB | L1D Global Writeback |
| 0x0184 5044 | L1DWBINV | L1D Global Writeback with Invalidate |
| 0x0184 5048 | L1DINV | L1D Global Invalidate without writeback |
| 0x0184 8000 - 0x0184 8004 | MAR0 - MAR1 | Reserved 0x0000 0000 - 0x01FF FFFF |
| 0x0184 8008 - 0x0184 8024 | MAR2 - MAR9 | Memory Attribute Registers for EMIFA 0x0200 0000 - 0x09FF FFFF |
| 0x0184 8028 - 0x0184 802C | MAR10 - MAR11 | Reserved 0x0A00 0000 - 0x0BFF FFFF |
| 0x0184 8030 - 0x0184 803C | MAR12 - MAR15 | Memory Attribute Registers for VLYNQ 0x0C00 0000 - 0x0FFF FFFF |
| 0x0184 8040 - 0x0184 8104 | MAR16 - MAR65 | Reserved 0x1000 0000 - 0x41FF FFFF |
| 0x0184 8108 - 0x0184 813C | MAR66 - MAR79 | Memory Attribute Registers for EMIFA/VLYNQ Shadow 0x4200 0000 - 0x4FFF FFFF |
| 0x0184 8140- 0x0184 81FC | MAR80 - MAR127 | Reserved 0x5000 0000 - 0x7FFF FFFF |
| 0x0184 8200 - 0x0184 823C | MAR128 - MAR143 | Memory Attribute Registers for DDR2 0x8000 0000 - 0x8FFF FFFF |
| 0x0184 8240 - 0x0184 83FC | MAR144 - MAR255 | Reserved 0x9000 0000 - 0xFFFF FFFF |

### 2.4.3 Peripherals

The DSP has controllability for the following peripherals:

- VICP
- EDMA3
- ASP
- 2 Timers (Timer 0 and Timer1) that can each be configured as 1 64-bit or 2 32-bit timers

### 2.4.4 DSP Interrupt Controller

The DSP Interrupt Controller accepts device interrupts and appropriately maps them to the DSP's available interrupts. The DSP Interrupt Controller is briefly described in Section 6.7, *Interrupts*, of this document. For more detailed on the DSP Interrupt Controller, see the *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* (literature number SPRU732).

## 2.5 Memory Map Summary

Table 2-3 shows the memory map address ranges of the device. Table 2-4 depicts the expanded map of the Configuration Space (0x0180 0000 through 0x0FFF FFFF). The device has multiple on-chip memories associated with its two processors and various subsystems. To help simplify software development a unified memory map is used where possible to maintain a consistent view of device resources across all bus masters.

## Table 2-3. Memory Map Summary

| START ADDRESS | END ADDRESS | SIZE (Bytes) | ARM | C64x+ | EDMA3/ PERIPHERAL | HPI | VPSS |
|---|---|---|---|---|---|---|---|
| 0x0000 0000 | 0x0000 1FFF | 8K | ARM RAM0 (Instruction) | Reserved | Reserved | Reserved | Reserved |
| 0x0000 2000 | 0x0000 3FFF | 8K | ARM RAM1 (Instruction) | | | | |
| 0x0000 4000 | 0x0000 5FFF | 8K | ARM ROM (Instruction) | | | | |
| 0x0000 6000 | 0x0000 7FFF | 8K | Reserved | | | | |
| 0x0000 8000 | 0x0000 9FFF | 8K | ARM RAM0 (Data) | | ARM RAM0 | ARM RAM0 | |
| 0x0000 A000 | 0x0000 BFFF | 8K | ARM RAM1 (Data) | | ARM RAM1 | ARM RAM1 | |
| 0x0000 C000 | 0x0000 DFFF | 8K | ARM ROM (Data) | | ARM ROM | ARM ROM | |
| 0x0000 E000 | 0x0000 FFFF | 8K | Reserved | | Reserved | Reserved | |
| 0x0001 0000 | 0x000F FFFF | 960K | | | | | |
| 0x0010 0000 | 0x001F FFFF | 1M | | VICP | | | |
| 0x0020 0000 | 0x007F FFFF | 6M | | Reserved | | | |
| 0x0080 0000 | 0x0080 FFFF | 64K | | L2 RAM/Cache | | | |
| 0x0081 0000 | 0x00E0 7FFF | 6112K | | Reserved | | | |
| 0x00E0 8000 | 0x00E0 FFFF | 32K | | L1P Cache | | | |
| 0x00E1 0000 | 0x00F0 3FFF | 976K | | Reserved | | | |
| 0x00F0 4000 | 0x00F0 FFFF | 48K | | L1D RAM | | | |
| 0x00F1 0000 | 0x00F1 7FFF | 32K | | L1D Cache | | | |
| 0x00F1 8000 | 0x017F FFFF | 9120K | | Reserved | | | |
| 0x0180 0000 | 0x01BB FFFF | 3840K | | CFG Space | | | |
| 0x01BC 0000 | 0x01BC 0FFF | 4K | ARM ETB Memory | | | | |
| 0x01BC 1000 | 0x01BC 17FF | 2K | ARM ETB Registers | | | | |
| 0x01BC 1800 | 0x01BC 18FF | 256 | ARM IceCrusher | | | | |
| 0x01BC 1900 | 0x01BF FFFF | 255744 | Reserved | | | | |
| 0x01C0 0000 | 0x01FF FFFF | 4M | CFG Bus Peripherals | CFG Bus Peripherals | CFG Bus Peripherals | CFG Bus Peripherals(1) | |
| 0x0200 0000 | 0x09FF FFFF | 128M | EMIFA (Code and Data) | EMIFA (Data) | EMIFA (Data) | Reserved | |
| 0x0A00 0000 | 0x0BFF FFFF | 32M | Reserved | Reserved | Reserved | | |
| 0x0C00 0000 | 0x0FFF FFFF | 64M | VLYNQ (Remote) | | VLYNQ (Remote) | | |
| 0x1000 0000 | 0x1000 7FFF | 32K | Reserved | | Reserved | | |
| 0x1000 8000 | 0x1000 9FFF | 8K | | ARM RAM0 | ARM RAM0 | | |
| 0x1000 A000 | 0x1000 BFFF | 8K | | ARM RAM1 | ARM RAM1 | | |
| 0x1000 C000 | 0x1000 DFFF | 8K | | ARM ROM | ARM ROM | | |
| 0x1000 E000 | 0x1000 FFFF | 8K | | Reserved | Reserved | | |
| 0x1001 0000 | 0x110F FFFF | 17344K | | | | | |
| 0x1110 0000 | 0x111F FFFF | 1M | | | | | |
| 0x1120 0000 | 0x117F FFFF | 6M | | | | | |
| 0x1180 0000 | 0x1180 FFFF | 64K | L2 RAM/Cache | L2 RAM/Cache | L2 RAM/Cache | | |
| 0x1181 0000 | 0x11E0 7FFF | 6112K | Reserved | Reserved | Reserved | | |
| 0x11E0 8000 | 0x11E0 FFFF | 32K | L1P Cache | L1P Cache | L1P Cache | | |
| 0x11E1 0000 | 0x11F0 3FFF | 976K | Reserved | Reserved | Reserved | | |
| 0x11F0 4000 | 0x11F0 FFFF | 48K | L1D RAM | L1D RAM | L1D RAM | | |
| 0x11F1 0000 | 0x11F1 7FFF | 32K | L1D RAM/Cache | L1D RAM/Cache | L1D RAM/Cache | | |
| 0x11F1 8000 | 0x1FFF FFFF | 241M-32K | Reserved | Reserved | Reserved | | |
| 0x2000 0000 | 0x2000 7FFF | 32K | DDR2 Control Registers | DDR2 Control Registers | DDR2 Control Registers | DDR2 Control Registers | |
| 0x2000 8000 | 0x41FF FFFF | 544M-32k | Reserved | Reserved | Reserved | Reserved | |
| 0x4200 0000(2) | 0x4FFF FFFF | 224M | Reserved | EMIFA/VLYNQ Shadow | EMIFA/VLYNQ Shadow | | |
| 0x5000 0000 | 0x7FFF FFFF | 768M | Reserved | Reserved | Reserved | | |
| 0x8000 0000 | 0x8FFF FFFF | 256M | DDR2 | DDR2 | DDR2 | DDR2 | DDR2 |
| 0x9000 0000 | 0xFFFF FFFF | 1792M | Reserved | Reserved | Reserved | Reserved | Reserved |

(1) HPI's access to the configuration bus peripherals is limited to the power and sleep controller registers, PLL1 and PLL2 registers, and HPI configuration registers.

(2) EMIFA shadow memory started a 0x4200 0000 is physically the same memory as location 0x0200 0000. Memory range 0x200 0000 through 0x09FF FFFF should only be used by C64x+ for data accesses. Memory range 0x4200 0000 through 0x4FFF FFFF can be used by C64x+ for both code execution and data accesses.

**Table 2-4. Configuration Memory Map Summary**

| START ADDRESS | END ADDRESS | SIZE (Bytes) | ARM/EDMA3 | C64x+ |
|---|---|---|---|---|
| 0x0180 0000 | 0x0180 FFFF | 64K | Reserved | C64x+ Interrupt Controller |
| 0x0181 0000 | 0x0181 0FFF | 4K | | C64x+ Powerdown Controller |
| 0x0181 1000 | 0x0181 1FFF | 4K | | C64x+ Security ID |
| 0x0181 2000 | 0x0181 2FFF | 4K | | C64x+ Revision ID |
| 0x0182 0000 | 0x0182 FFFF | 64K | | C64x+ EMC |
| 0x0183 0000 | 0x0183 FFFF | 64K | | Reserved |
| 0x0184 0000 | 0x0184 FFFF | 64K | | C64x+ Memory System |
| 0x0185 0000 | 0x0187 FFFF | 192K | | Reserved |
| 0x0188 0000 | 0x01BB FFFF | 3328K | | Reserved |
| 0x01BC 0000 | 0x01BC 00FF | 256 | ARM ETB Memory | Reserved |
| 0x01BC 0100 | 0x01BC 01FF | 256 | | Pin Manager and Trace |
| 0x01BC 0200 | 0x01BC 0FFF | 3.5K | | Reserved |
| 0x01BC 1000 | 0x01BC 17FF | 2K | ARM ETB Registers | |
| 0x01BC 1800 | 0x01BC 18FF | 256 | ARM Ice Crusher | |
| 0x01BC 1900 | 0x01BF FFFF | 255744 | Reserved | |
| 0x01C0 0000 | 0x01C0 FFFF | 64K | EDMA3 CC | EDMA3 CC |
| 0x01C1 0000 | 0x01C1 03FF | 1K | EDMA3 TC0 | EDMA3 TC0 |
| 0x01C1 0400 | 0x01C1 07FF | 1K | EDMA3 TC1 | EDMA3 TC1 |
| 0x01C1 8800 | 0x01C1 9FFF | 6K | Reserved | Reserved |
| 0x01C1 A000 | 0x01C1 FFFF | 24K | | |
| 0x01C2 0000 | 0x01C2 03FF | 1K | UART0 | |
| 0x01C2 0400 | 0x01C2 07FF | 1K | UART1 | |
| 0x01C2 0800 | 0x01C2 0BFF | 1K | UART2 | |
| 0x01C2 0C00 | 0x01C2 0FFF | 1K | Reserved | |
| 0x01C2 1000 | 0x01C2 13FF | 1K | I2C | |
| 0x01C2 1400 | 0x01C2 17FF | 1K | Timer0 | Timer0 |
| 0x01C2 1800 | 0x01C2 1BFF | 1K | Timer1 | Timer1 |
| 0x01C2 1C00 | 0x01C2 1FFF | 1K | Timer2 (Watchdog) | Reserved |
| 0x01C2 2000 | 0x01C2 23FF | 1K | PWM0 | |
| 0x01C2 2400 | 0x01C2 27FF | 1K | PWM1 | |
| 0x01C2 2800 | 0x01C2 2BFF | 1K | PWM2 | |
| 0x01C2 2C00 | 0x01C3 FFFF | 117K | Reserved | |
| 0x01C4 0000 | 0x01C4 07FF | 2K | System Module | System Module |
| 0x01C4 0800 | 0x01C4 0BFF | 1K | PLL Controller 1 | Reserved |
| 0x01C4 0C00 | 0x01C4 0FFF | 1K | PLL Controller 2 | |
| 0x01C4 1000 | 0x01C4 1FFF | 4K | Power and Sleep Controller | Power and Sleep Controller |
| 0x01C4 2000 | 0x01C4 202F | 48 | Reserved | Reserved |
| 0x01C4 2030 | 0x01C4 2033 | 4 | DDR2 VTP Reg | DDR2 VTP Reg |
| 0x01C4 2034 | 0x01C4 23FF | 1K - 52 | Reserved | Reserved |
| 0x01C4 2400 | 0x01C4 7FFF | 23K | | |
| 0x01C4 8000 | 0x01C4 83FF | 1K | ARM Interrupt Controller | |
| 0x01C4 8400 | 0x01C5 FFFF | 95K | Reserved | |
| 0x01C6 0000 | 0x01C6 3FFF | 16K | | |
| 0x01C6 4000 | 0x01C6 5FFF | 8K | USB2.0 Registers / RAM | |
| 0x01C6 6000 | 0x01C6 67FF | 2K | ATA/CF | |
| 0x01C6 6800 | 0x01C6 6FFF | 2K | SPI | |
| 0x01C6 7000 | 0x01C6 77FF | 2K | GPIO | |

**Table 2-4. Configuration Memory Map Summary (continued)**

| START ADDRESS | END ADDRESS | SIZE (Bytes) | ARM/EDMA3 | C64x+ |
|---|---|---|---|---|
| 0x01C6 7800 | 0x01C6 7FFF | 2K | HPI | HPI |
| 0x01C6 8000 | 0x01C6 FFFF | 32K | Reserved | Reserved |
| 0x01C7 0000 | 0x01C7 3FFF | 16K | VPSS Registers | |
| 0x01C7 4000 | 0x01C7 FFFF | 48K | Reserved | |
| 0x01C8 0000 | 0x01C8 0FFF | 4K | EMAC Control Registers | |
| 0x01C8 1000 | 0x01C8 1FFF | 4K | EMAC Control Module Registers | |
| 0x01C8 2000 | 0x01C8 3FFF | 8K | EMAC Control Module RAM | |
| 0x01C8 4000 | 0x01C8 47FF | 2K | MDIO Control Registers | |
| 0x01C8 4800 | 0x01C8 4FFF | 2K | Reserved | |
| 0x01C8 5000 | 0x01CB FFFF | 236K | | |
| 0x01CC 0000 | 0x01CD FFFF | 128K | VICP | VICP |
| 0x01CE 0000 | 0x01CF FFFF | 128K | Reserved | Reserved |
| 0x01D0 0000 | 0x01DF FFFF | 1M | | |
| 0x01E0 0000 | 0x01E0 0FFF | 4K | EMIFA Control | |
| 0x01E0 1000 | 0x01E0 1FFF | 4K | VLYNQ Control Registers | |
| 0x01E0 2000 | 0x01E0 3FFF | 8K | ASP | ASP |
| 0x01E0 4000 | 0x01E0 FFFF | 48K | Reserved | Reserved |
| 0x01E1 0000 | 0x01E1 FFFF | 64K | MMC/SD/SDIO | |
| 0x01E2 0000 | 0x01E3 FFFF | 128K | Reserved | |
| 0x01E4 0000 | 0x01FF FFFF | 1792K | | |
| 0x0200 0000 | 0x03FF FFFF | 32M | EMIFA Data/Code (CS2) | EMIFA Data (CS2) |
| 0x0400 0000 | 0x05FF FFFF | 32M | EMIFA Data/Code (CS3) | EMIFA Data (CS3) |
| 0x0600 0000 | 0x07FF FFFF | 32M | EMIFA Data/Code (CS4) | EMIFA Data (CS4) |
| 0x0800 0000 | 0x09FF FFFF | 32M | EMIFA Data/Code (CS5) | EMIFA Data (CS5) |
| 0x0A00 0000 | 0x0BFF FFFF | 32M | Reserved | Reserved |
| 0x0C00 0000 | 0x0FFF FFFF | 64M | VLYNQ (Remote) | |

## 2.6 Pin Assignments

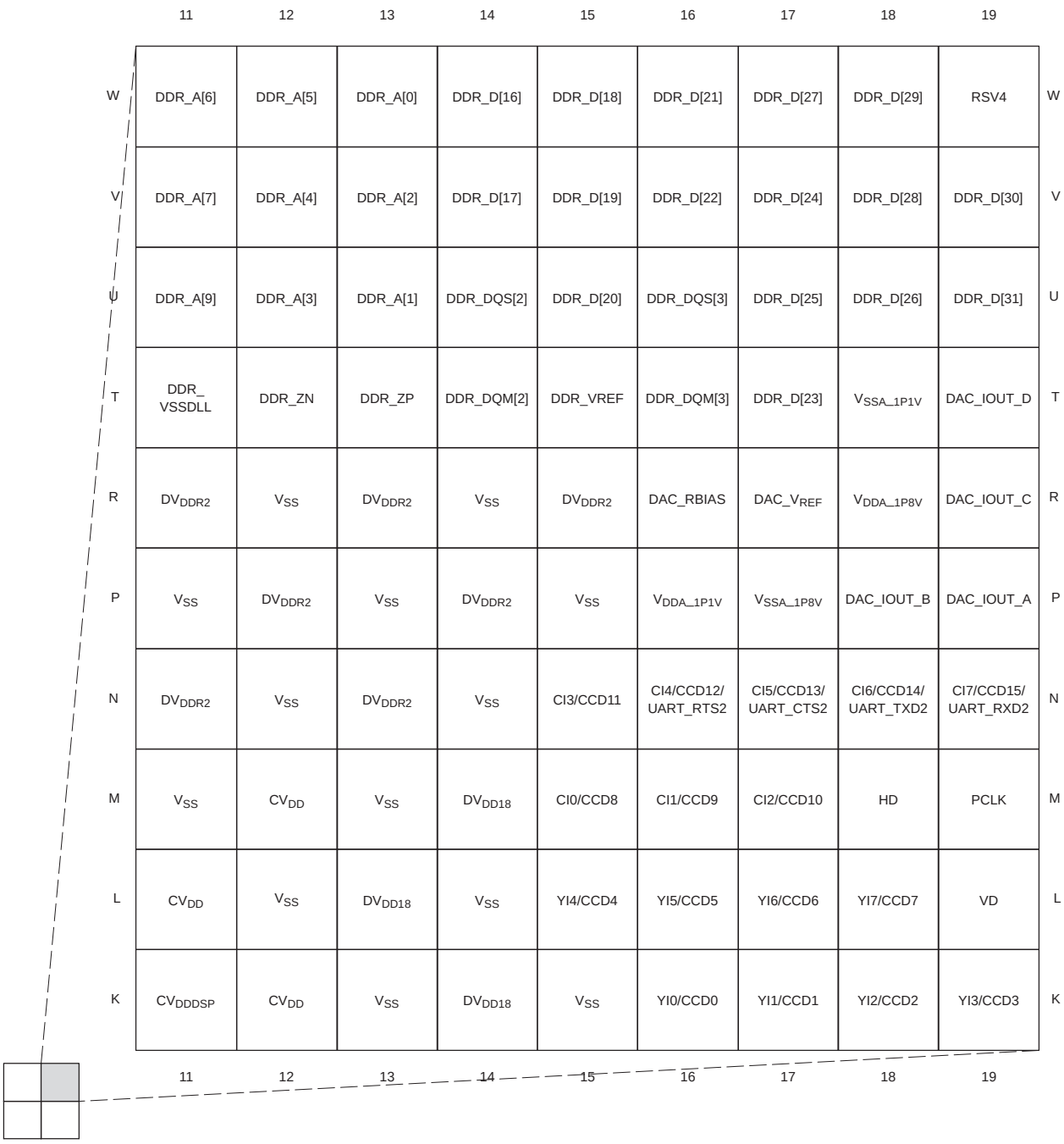
Extensive use of pin multiplexing is used to accommodate the largest number of peripheral functions in the smallest possible package. Pin multiplexing is controlled using a combination of hardware configuration at device reset and software programmable register settings. For more information on pin muxing, see Section 3.5.2, *Multiplexed Pin Configurations*, of this document.

### 2.6.1 Pin Map (Bottom View)

Figure 2-2 through Figure 2-5 show the bottom view of the package pin assignments in four quadrants (A, B, C, and D).

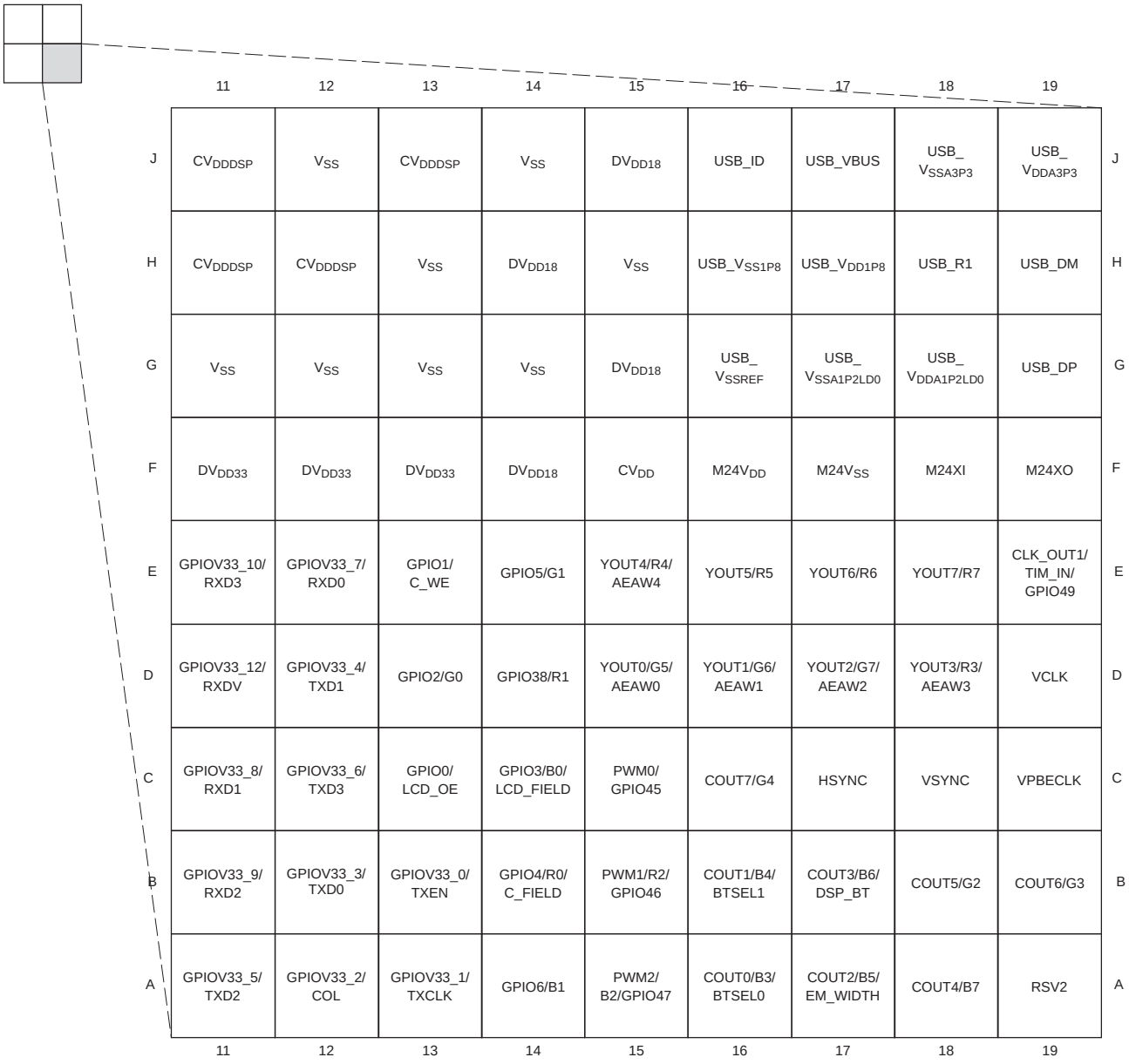
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| W | RSV3 | DDR_D[4] | DDR_D[7] | DDR_D[9] | DDR_D[12] | DDR_D[14] | DDR_CLK0 | $\overline{\text{DDR\_CLK0}}$ | DDR_A[12] | DDR_A[11] | W |
| V | DDR_D[2] | DDR_D[3] | DDR_D[6] | DDR_D[8] | DDR_D[11] | DDR_D[13] | DDR_D[15] | DDR_CKE | DDR_BS[1] | DDR_A[8] | V |
| U | DDR_D[0] | DDR_D[1] | DDR_D[5] | DDR_DQS[0] | DDR_D[10] | DDR_DQS[1] | $\overline{\text{DDR\_RAS}}$ | DDR_BS[0] | DDR_BS[2] | DDR_A[10] | U |
| T | $\overline{\text{EM\_CS5}}$/ GPIO8/ VLYNQ_ CLOCK | $\overline{\text{EM\_CS4}}$/ GPIO9/ VLYNQ_ SCRUN | EM_A[21]/ GPIO10/ VLYNQ_TXD0 | DDR_ DQM[0] | $DV_{DDR2}$ | DDR_ DQM[1] | $\overline{\text{DDR\_CAS}}$ | $\overline{\text{DDR\_WE}}$ | $\overline{\text{DDR\_CS}}$ | DDR_VDDDLL | T |
| R | EM_A[12]/ GPIO19 | EM_A[17]/ GPIO14/ VLYNQ_TXD2 | EM_A[20]/ GPIO11/ VLYNQ_RXD0 | EM_A[19]/ GPIO12/ VLYNQ_TXD1 | EM_A[16]/ GPIO15/ VLYNQ_RXD2 | $V_{SS}$ | $V_{SS}$ | RSV7 | $DV_{DDR2}$ | $V_{SS}$ | R |
| P | EM_A[10]/ GPIO21 | EM_A[11]/ GPIO20 | EM_A[15]/ GPIO16/ VLYNQ_TXD3 | EM_A[14]/ GPIO17/ VLYNQ_RXD3 | EM_A[18]/ GPIO13/ VLYNQ_RXD1 | $DV_{DDR2}$ | $V_{SS}$ | $DV_{DDR2}$ | $V_{SS}$ | $DV_{DDR2}$ | P |
| N | EM_A[6]/ GPIO25 | EM_A[7]/ GPIO24 | EM_A[8]/ GPIO23 | EM_A[13]/ GPIO18 | $DV_{DD18}$ | $V_{SS}$ | $DV_{DDR2}$ | $V_{SS}$ | $DV_{DDR2}$ | $V_{SS}$ | N |
| M | MXO | $PLLV_{DD18}$ | RSV24 | EM_A[9]/ GPIO22 | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $CV_{DD}$ | M |
| L | MXI/CLKIN | $MXV_{SS}$ | RSV6 | $\overline{\text{RESET}}$ | $MXV_{DD}$ | $V_{SS}$ | $DV_{DD18}$ | $CV_{DD}$ | $CV_{DD}$ | $CV_{DD}$ | L |
| K | CLK_OUT0/ GPIO48 | EM_A[3]/ GPIO28 | EM_A[5]/ GPIO26 | EM_A[4]/ GPIO27 | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $CV_{DDDSP}$ | $CV_{DDDSP}$ | $CV_{DD}$ | K |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |

**Figure 2-2. Pin Map [Quadrant A]**



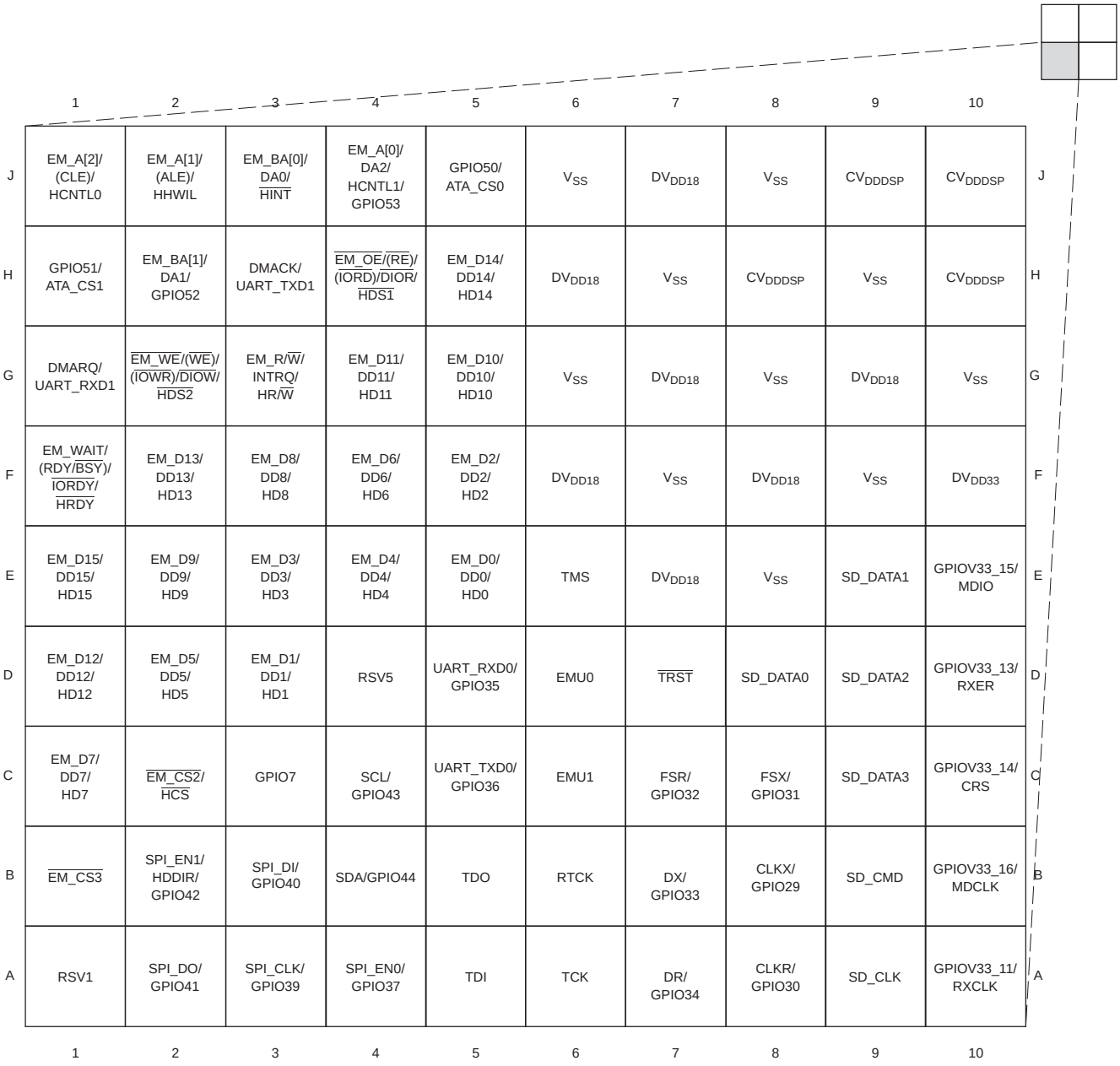
| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | |
|---|---|---|---|---|---|---|---|---|---|---|
| W | DDR_A[6] | DDR_A[5] | DDR_A[0] | DDR_D[16] | DDR_D[18] | DDR_D[21] | DDR_D[27] | DDR_D[29] | RSV4 | W |
| V | DDR_A[7] | DDR_A[4] | DDR_A[2] | DDR_D[17] | DDR_D[19] | DDR_D[22] | DDR_D[24] | DDR_D[28] | DDR_D[30] | V |
| U | DDR_A[9] | DDR_A[3] | DDR_A[1] | DDR_DQS[2] | DDR_D[20] | DDR_DQS[3] | DDR_D[25] | DDR_D[26] | DDR_D[31] | U |
| T | DDR_ VSSDLL | DDR_ZN | DDR_ZP | DDR_DQM[2] | DDR_VREF | DDR_DQM[3] | DDR_D[23] | $V_{SSA\_1P1V}$ | DAC_IOUT_D | T |
| R | $DV_{DDR2}$ | $V_{SS}$ | $DV_{DDR2}$ | $V_{SS}$ | $DV_{DDR2}$ | DAC_RBIAS | DAC_$V_{REF}$ | $V_{DDA\_1P8V}$ | DAC_IOUT_C | R |
| P | $V_{SS}$ | $DV_{DDR2}$ | $V_{SS}$ | $DV_{DDR2}$ | $V_{SS}$ | $V_{DDA\_1P1V}$ | $V_{SSA\_1P8V}$ | DAC_IOUT_B | DAC_IOUT_A | P |
| N | $DV_{DDR2}$ | $V_{SS}$ | $DV_{DDR2}$ | $V_{SS}$ | CI3/CCD11 | CI4/CCD12/ UART_RTS2 | CI5/CCD13/ UART_CTS2 | CI6/CCD14/ UART_TXD2 | CI7/CCD15/ UART_RXD2 | N |
| M | $V_{SS}$ | $CV_{DD}$ | $V_{SS}$ | $DV_{DD18}$ | CI0/CCD8 | CI1/CCD9 | CI2/CCD10 | HD | PCLK | M |
| L | $CV_{DD}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | YI4/CCD4 | YI5/CCD5 | YI6/CCD6 | YI7/CCD7 | VD | L |
| K | $CV_{DDDSP}$ | $CV_{DD}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | YI0/CCD0 | YI1/CCD1 | YI2/CCD2 | YI3/CCD3 | K |
| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | |

**Figure 2-3. Pin Map [Quadrant B]**



| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | |
|---|---|---|---|---|---|---|---|---|---|---|
| J | $CV_{DDDSP}$ | $V_{SS}$ | $CV_{DDDSP}$ | $V_{SS}$ | $DV_{DD18}$ | USB_ID | USB_VBUS | USB_ $V_{SSA3P3}$ | USB_ $V_{DDA3P3}$ | J |
| H | $CV_{DDDSP}$ | $CV_{DDDSP}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | USB_$V_{SS1P8}$ | USB_$V_{DD1P8}$ | USB_R1 | USB_DM | H |
| G | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $DV_{DD18}$ | USB_ $V_{SSREF}$ | USB_ $V_{SSA1P2LD0}$ | USB_ $V_{DDA1P2LD0}$ | USB_DP | G |
| F | $DV_{DD33}$ | $DV_{DD33}$ | $DV_{DD33}$ | $DV_{DD18}$ | $CV_{DD}$ | $M24V_{DD}$ | $M24V_{SS}$ | M24XI | M24XO | F |
| E | GPIOV33_10/ RXD3 | GPIOV33_7/ RXD0 | GPIO1/ C_WE | GPIO5/G1 | YOUT4/R4/ AEAW4 | YOUT5/R5 | YOUT6/R6 | YOUT7/R7 | CLK_OUT1/ TIM_IN/ GPIO49 | E |
| D | GPIOV33_12/ RXDV | GPIOV33_4/ TXD1 | GPIO2/G0 | GPIO38/R1 | YOUT0/G5/ AEAW0 | YOUT1/G6/ AEAW1 | YOUT2/G7/ AEAW2 | YOUT3/R3/ AEAW3 | VCLK | D |
| C | GPIOV33_8/ RXD1 | GPIOV33_6/ TXD3 | GPIO0/ LCD_OE | GPIO3/B0/ LCD_FIELD | PWM0/ GPIO45 | COUT7/G4 | HSYNC | VSYNC | VPBECLK | C |
| B | GPIOV33_9/ RXD2 | GPIOV33_3/ TXD0 | GPIOV33_0/ TXEN | GPIO4/R0/ C_FIELD | PWM1/R2/ GPIO46 | COUT1/B4/ BTSEL1 | COUT3/B6/ DSP_BT | COUT5/G2 | COUT6/G3 | B |
| A | GPIOV33_5/ TXD2 | GPIOV33_2/ COL | GPIOV33_1/ TXCLK | GPIO6/B1 | PWM2/ B2/GPIO47 | COUT0/B3/ BTSEL0 | COUT2/B5/ EM_WIDTH | COUT4/B7 | RSV2 | A |
| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | |

**Figure 2-4. Pin Map [Quadrant C]**



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| J | EM_A[2]/ (CLE)/ HCNTL0 | EM_A[1]/ (ALE)/ HHWIL | EM_BA[0]/ DA0/ $\overline{\text{HINT}}$ | EM_A[0]/ DA2/ HCNTL1/ GPIO53 | GPIO50/ ATA_CS0 | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $CV_{DDDSP}$ | $CV_{DDDSP}$ | J |
| H | GPIO51/ ATA_CS1 | EM_BA[1]/ DA1/ GPIO52 | DMACK/ UART_TXD1 | $\overline{\text{EM\_OE}}$/($\overline{\text{RE}}$)/ ($\overline{\text{IORD}}$)/$\overline{\text{DIOR}}$/ $\overline{\text{HDS1}}$ | EM_D14/ DD14/ HD14 | $DV_{DD18}$ | $V_{SS}$ | $CV_{DDDSP}$ | $V_{SS}$ | $CV_{DDDSP}$ | H |
| G | DMARQ/ UART_RXD1 | $\overline{\text{EM\_WE}}$/($\overline{\text{WE}}$)/ ($\overline{\text{IOWR}}$)/$\overline{\text{DIOW}}$/ $\overline{\text{HDS2}}$ | EM_R/$\overline{\text{W}}$/ INTRQ/ HR/$\overline{\text{W}}$ | EM_D11/ DD11/ HD11 | EM_D10/ DD10/ HD10 | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | G |
| F | EM_WAIT/ (RDY/$\overline{\text{BSY}}$)/ $\overline{\text{IORDY}}$/ $\overline{\text{HRDY}}$ | EM_D13/ DD13/ HD13 | EM_D8/ DD8/ HD8 | EM_D6/ DD6/ HD6 | EM_D2/ DD2/ HD2 | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD18}$ | $V_{SS}$ | $DV_{DD33}$ | F |
| E | EM_D15/ DD15/ HD15 | EM_D9/ DD9/ HD9 | EM_D3/ DD3/ HD3 | EM_D4/ DD4/ HD4 | EM_D0/ DD0/ HD0 | TMS | $DV_{DD18}$ | $V_{SS}$ | SD_DATA1 | GPIOV33_15/ MDIO | E |
| D | EM_D12/ DD12/ HD12 | EM_D5/ DD5/ HD5 | EM_D1/ DD1/ HD1 | RSV5 | UART_RXD0/ GPIO35 | EMU0 | $\overline{\text{TRST}}$ | SD_DATA0 | SD_DATA2 | GPIOV33_13/ RXER | D |
| C | EM_D7/ DD7/ HD7 | $\overline{\text{EM\_CS2}}$/ $\overline{\text{HCS}}$ | GPIO7 | SCL/ GPIO43 | UART_TXD0/ GPIO36 | EMU1 | FSR/ GPIO32 | FSX/ GPIO31 | SD_DATA3 | GPIOV33_14/ CRS | C |
| B | $\overline{\text{EM\_CS3}}$ | SPI_EN1/ HDDIR/ GPIO42 | SPI_DI/ GPIO40 | SDA/GPIO44 | TDO | RTCK | DX/ GPIO33 | CLKX/ GPIO29 | SD_CMD | GPIOV33_16/ MDCLK | B |
| A | RSV1 | SPI_DO/ GPIO41 | SPI_CLK/ GPIO39 | SPI_EN0/ GPIO37 | TDI | TCK | DR/ GPIO34 | CLKR/ GPIO30 | SD_CLK | GPIOV33_11/ RXCLK | A |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |

**Figure 2-5. Pin Map [Quadrant D]**

## 2.7 Terminal Functions

The terminal functions tables (Table 2-5 through Table 2-30) identify the external signal names, the associated pin (ball) numbers along with the mechanical package designator, the pin type, whether the pin has any internal pullup or pulldown resistors, and a functional pin description. For more detailed information on device configuration, peripheral selection, multiplexed/shared pins, see Section 3, *Device Configurations*, of this data manual.

**Table 2-5. BOOT Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **BOOT** | | | | |
| COUT0/ B3/ BTSEL0 | A16 | I/O/Z | IPD $DV_{DD18}$ | These pins are multiplexed between ARM boot mode and the VPBE. At reset, the boot mode inputs BTSEL0 and BTSEL1 are sampled to determine the ARM boot configuration. See below for the boot modes set by these inputs. See Section 3.3, *Bootmode*, for more details. After reset, these are video encoder outputs COUT0 and COUT1, or RGB666/888 Blue output data bits 3 and 4 B3/B4. |
| COUT1/ B4/ BTSEL1 | B16 | I/O/Z | IPD $DV_{DD18}$ | **BTSEL1** / **BTSEL0** / **ARM Boot Mode**<br>0 / 0 / ARM ROM Boot (NAND, SPI) [default]<br>0 / 1 / ARM EMIFA Boot (NOR)<br>1 / 0 / ARM ROM Boot (HPI)<br>1 / 1 / ARM ROM Boot (UART0) |
| COUT2/ B5/ EM_WIDTH | A17 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between EMIFA and the VPBE. At reset, the input state is sampled to set the EMIFA data bus width (EM_WIDTH). For an 8-bit-wide EMIFA data bus, EM_WIDTH = 0. For a 16-bit-wide EMIFA data bus, EM_WIDTH = 1. After reset, it is video encoder output COUT2 or RGB666/888 Blue output data bit 5 B5. |
| COUT3/ B6/ DSP_BT | B17 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between DSP boot and the VPBE. At reset, the input state is sampled to set the DSP boot source DSP_BT. The DSP is booted by the ARM when DSP_BT=0. The DSP boots from EMIFA when DSP_BT=1. After reset, it is video encoder output COUT3 or RGB666/888 Blue data bit 6 output B6. |
| YOUT0/ G5/ AEAW0 | D15 | I/O/Z | IPD $DV_{DD18}$ | These pins are multiplexed between EMIFA and the VPBE. At reset, the input states of AEAW[4:0] are sampled to set the EMIFA address bus width. See Section 3.4.2, *Peripheral Selection at Device Reset*, for details.<br>After reset, these are video encoder outputs YOUT[0:4] or RGB666/888 Red and Green data bit outputs G5, G6, G7, R3, and R4. |
| YOUT1/ G6/ AEAW1 | D16 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT2/ G7/ AEAW2 | D17 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT3/ R3/ AEAW3 | D18 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT4/ R4/ AEAW4 | E15 | I/O/Z | IPD $DV_{DD18}$ | |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(3) Specifies the operating I/O supply voltage for each signal

### Table 2-6. Oscillator/PLL Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **OSCILLATOR, PLL** | | | | |
| MXI/CLKIN | L1 | I | $DV_{DD18}$ | Crystal input MXI for MX oscillator (system oscillator, typically 27 MHz). If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, this is the external oscillator clock input. |
| MXO | M1 | O | $DV_{DD18}$ | Crystal output for MX oscillator. If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, MXO should be left as a No Connect. |
| $MXV_{DD}$ | L5 | S | (3) | 1.8-V power supply for MX oscillator. If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, $MXV_{DD}$ should still be connected to the 1.8-V power supply. |
| $MXV_{SS}$ | L2 | GND | (3) | Ground for MX oscillator. If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, $MXV_{SS}$ should still be connected to ground. |
| M24XI | F18 | I | $DV_{DD18}$ | Crystal input for M24 oscillator (24 MHz for USB). If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, this is the external oscillator clock input. When the USB peripheral ***is not*** used, M24XI should be left as a No Connect. |
| M24XO | F19 | O | $DV_{DD18}$ | Crystal output for M24 oscillator. If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, M24XO should be left as a No Connect. When the USB peripheral ***is not*** used, M24XO should be left as a No Connect. |
| $M24V_{DD}$ | F16 | S | (3) | 1.8-V power supply for M24 oscillator. If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, $M24V_{DD}$ should still be connected to the 1.8-V power supply. When the USB peripheral ***is not*** used, $M24V_{DD}$ should be connected to the 1.8-V power supply. |
| $M24V_{SS}$ | F17 | GND | (3) | Ground for M24 oscillator. If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, $M24V_{SS}$ should still be connected to ground. When the USB peripheral ***is not*** used, $M24V_{SS}$ should be connected to ground. |
| $PLLV_{DD18}$ | M2 | S | (3) | 1.8-V power supply for PLLs (system). |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal
(3) For more information, see Section 5.2, *Recommended Operating Conditions*.

### Table 2-7. Clock Generator Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **CLOCK GENERATOR** | | | | |
| CLK_OUT0/ GPIO48 | K1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between the PLL1 clock generator and GPIO. For the PLL1 clock generator, it is clock output CLK_OUT0. This is configurable for 13.5 MHz or 27 MHz clock outputs. |
| CLK_OUT1/ TIM_IN/ GPIO49 | E19 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between the USB clock generator, timer, and GPIO. For the USB clock generator, it is clock output CLK_OUT1. This is configurable for 12 MHz or 24 MHz clock outputs. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

### Table 2-8. RESET and JTAG Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **RESET** | | | | |
| $\overline{\text{RESET}}$ | L4 | I | IPU $DV_{DD18}$ | This is the active low global reset input. |
| **JTAG** | | | | |
| TMS | E6 | I | IPU $DV_{DD18}$ | JTAG test-port mode select input |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(3) Specifies the operating I/O supply voltage for each signal

**Table 2-8. RESET and JTAG Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| TDO | B5 | O/Z | – $DV_{DD18}$ | JTAG test-port data output |
| TDI | A5 | I | IPU $DV_{DD18}$ | JTAG test-port data input |
| TCK | A6 | I | IPU $DV_{DD18}$ | JTAG test-port clock input |
| RTCK | B6 | O/Z | – $DV_{DD18}$ | JTAG test-port return clock output |
| $\overline{TRST}$ | D7 | I | IPD $DV_{DD18}$ | JTAG test-port reset. For IEEE 1149.1 JTAG compatibility, see the IEEE 1149.1 JTAG compatibility statement portion of this data manual (Section 6.25, *IEEE 1149.1 JTAG*). |
| EMU1 | C6 | I/O/Z | IPU $DV_{DD18}$ | Emulation pin 1 |
| EMU0 | D6 | I/O/Z | IPU $DV_{DD18}$ | Emulation pin 0 |

**Table 2-9. EMIFA Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **EMIFA BOOT CONFIGURATION** | | | | |
| COUT2/ B5/ EM_WIDTH | A17 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between EMIFA and the VPBE. At reset, the input state is sampled to set the EMIFA data bus width (EM_WIDTH). For an 8-bit-wide EMIFA data bus, EM_WIDTH = 0. For a 16-bit-wide EMIFA data bus, EM_WIDTH = 1. After reset, it is video encoder output COUT2 or RGB666/888 Blue output data bit 5 B5. |
| COUT3/ B6/ DSP_BT | B17 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between DSP boot and the VPBE. At reset, the input state is sampled to set the DSP boot source DSP_BT. The DSP is booted by the ARM when DSP_BT=0. The DSP boots from EMIFA when DSP_BT=1. After reset, it is video encoder output COUT3 or RGB666/888 Blue data bit 6 output B6. |
| YOUT0/ G5/ AEAW0 | D15 | I/O/Z | IPD $DV_{DD18}$ | These pins are multiplexed between EMIFA and the VPBE. At reset, the input states of AEAW[4:0] are sampled to set the EMIFA address bus width. See Section 3.4.2, *Peripheral Selection at Device Reset*, for details. After reset, these are video encoder outputs YOUT[0:4] or RGB666/888 Red and Green data bit outputs G5, G6, G7, R3, and R4. |
| YOUT1/ G6/ AEAW1 | D16 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT2/ G7/ AEAW2 | D17 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT3/ R3/ AEAW3 | D18 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT4/ R4/ AEAW4 | E15 | I/O/Z | IPD $DV_{DD18}$ | |
| **EMIFA FUNCTIONAL PINS: ASYNC / NOR** | | | | |
| $\overline{EM\_CS2}$/ $\overline{HCS}$ | C2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and $\overline{HPI}$. For EMIFA, this pin is Chip Select 2 output $\overline{EM\_CS2}$ for use with asynchronous memories (i.e., NOR flash) or NAND flash. This is the chip select for the default boot and ROM boot modes. |
| $\overline{EM\_CS3}$ | B1 | I/O/Z | $DV_{DD18}$ | For EMIFA, this pin is Chip Select 3 output $\overline{EM\_CS3}$ for use with asynchronous memories (i.e., NOR flash) or NAND flash. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(3) Specifies the operating I/O supply voltage for each signal

**Table 2-9. EMIFA Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE[1] | OTHER[2] [3] | DESCRIPTION |
|---|---|---|---|---|
| EM_CS4/ GPIO9/ VLYNQ_SCRUN | T2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is Chip Select 4 output EM_CS4 for use with asynchronous memories (i.e., NOR flash) or NAND flash. |
| EM_CS5/ GPIO8/ VLYNQ_CLOCK | T1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is Chip Select 5 output EM_CS5 for use with asynchronous memories (i.e., NOR flash) or NAND flash. |
| EM_R/W/ INTRQ/ HR/W | G3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, and HPI. For EMIFA, it is read/write output EM_R/W. |
| EM_WAIT/ (RDY/BSY)/ IORDY/ HRDY | F1 | I/O/Z | IPU $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For EMIFA, it is wait state extension input EM_WAIT. |
| EM_OE/ (RE)/ (IORD)/ DIOR/ HDS1 | H4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For EMIFA, it is output enable output EM_OE. |
| EM_WE (WE) (IOWR)/ DIOW/ HDS2 | G2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For NAND/SmartMedia/xD or EMIFA, it is write enable output EM_WE. |
| EM_BA[0]/ DA0/ HINT | J3 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, and HPI. For EMIFA, this is the Bank Address 0 output (EM_BA[0]). When connected to an 8-bit asynchronous memory, this pin is the lowest order bit of the byte address. When connected to a 16-bit asynchronous memory, this pin has the same function as EMIF address pin 22 (EM_A[22]). |
| EM_BA[1]/ DA1/ GPIO52 | H2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, and GPIO. For EMIFA, this is the Bank Address 1 output EM_BA[1]. When connected to a 16 bit asynchronous memory this pin is the lowest order bit of the 16-bit word address. When connected to an 8-bit asynchronous memory, this pin is the 2nd bit of the address. |
| EM_A[21]/ GPIO10/ VLYNQ_TXD0 | T3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is address bit 21 output EM_A[21]. |
| EM_A[20]/ GPIO11/ VLYNQ_RXD0 | R3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is address bit 20 output EM_A[20]. |
| EM_A[19]/ GPIO12/ VLYNQ_TXD1 | R4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is address bit 19 output EM_A[19]. |
| EM_A[18]/ GPIO13/ VLYNQ_RXD1 | P5 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is address bit 18 output EM_A[18]. |
| EM_A[17]/ GPIO14/ VLYNQ_TXD2 | R2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is address bit 17 output EM_A[17]. |
| EM_A[16]/ GPIO15/ VLYNQ_RXD2 | R5 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is address bit 16 output EM_A[16]. |
| EM_A[15]/ GPIO16/ VLYNQ_TXD3 | P3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is address bit 15 output EM_A[15]. |
| EM_A[14]/ GPIO17/ VLYNQ_RXD3 | P4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is address bit 14 output EM_A[14]. |

**Table 2-9. EMIFA Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | OTHER[2] [3] | DESCRIPTION |
|---|---|---|---|---|
| EM_A[13]/ GPIO18 | N4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 13 output EM_A[13]. |
| EM_A[12]/ GPIO19 | R1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 12 output EM_A[12]. |
| EM_A[11]/ GPIO20 | P2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 11 output EM_A[11]. |
| EM_A[10]/ GPIO21 | P1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 10 output EM_A[10]. |
| EM_A[9]/ GPIO22 | M4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 9 output EM_A[9]. |
| EM_A[8]/ GPIO23 | N3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 8 output EM_A[8]. |
| EM_A[7]/ GPIO24 | N2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 7 output EM_A[7]. |
| EM_A[6]/ GPIO25 | N1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 6 output EM_A[6]. |
| EM_A[5]/ GPIO26 | K3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 5 output EM_A[5]. |
| EM_A[4]/ GPIO27 | K4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 4 output EM_A[4]. |
| EM_A[3]/ GPIO28 | K2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and GPIO. For EMIFA, it is address bit 3 output EM_A[3]. |
| EM_A[2]/ (CLE)/ HCNTL0 | J1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and HPI. For EMIFA, this pin is the EM_A[2] address line. |
| EM_A[1]/ (ALE)/ HHWIL | J2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia.xD) and HPI. |
| EM_A[0]/ DA2/ HCNTL1/ GPIO53 | J4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, HPI, and GPIO. For EMIFA, this is Address output EM_A[0], which is the least significant bit on a 32-bit word address. When connected to a 16-bit asynchronous memory, this pin is the 2nd bit of the address. For an 8-bit asynchronous memory, this pin is the 3rd bit of the address. |

**Table 2-9. EMIFA Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | OTHER[2] [3] | DESCRIPTION |
|---|---|---|---|---|
| EM_D0/ DD0/ HD0 | E5 | I/O/Z | $DV_{DD18}$ | These pins are multiplexed between EMIFA (NAND), ATA/CF, and HPI. In all cases they are used as a 16 bit bi-directional data bus. For EMIFA (NAND), these are EM_D[15:0]. |
| EM_D1/ DD1/ HD1 | D3 | I/O/Z | $DV_{DD18}$ | |
| EM_D2/ DD2/ HD2 | F5 | I/O/Z | $DV_{DD18}$ | |
| EM_D3/ DD3/ HD3 | E3 | I/O/Z | $DV_{DD18}$ | |
| EM_D4/ DD4/ HD4 | E4 | I/O/Z | $DV_{DD18}$ | |
| EM_D5/ DD5/ HD5 | D2 | I/O/Z | $DV_{DD18}$ | |
| EM_D6/ DD6/ HD6 | F4 | I/O/Z | $DV_{DD18}$ | |
| EM_D7/ DD7/ HD7 | C1 | I/O/Z | $DV_{DD18}$ | |
| EM_D8/ DD8/ HD8 | F3 | I/O/Z | $DV_{DD18}$ | |
| EM_D9/ DD9/ HD9 | E2 | I/O/Z | $DV_{DD18}$ | |
| EM_D10/ DD10/ HD10 | G5 | I/O/Z | $DV_{DD18}$ | |
| EM_D11/ DD11/ HD11 | G4 | I/O/Z | $DV_{DD18}$ | |
| EM_D12/ DD12/ HD12 | D1 | I/O/Z | $DV_{DD18}$ | |
| EM_D13/ DD13/ HD13 | F2 | I/O/Z | $DV_{DD18}$ | |
| EM_D14/ DD14/ HD14 | H5 | I/O/Z | $DV_{DD18}$ | |
| EM_D15/ DD15/ HD15 | E1 | I/O/Z | $DV_{DD18}$ | |

**Table 2-9. EMIFA Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **EMIFA FUNCTIONAL PINS: NAND / SMARTMEDIA / xD** | | | | |
| EM_A[1]/ (ALE)/ HHWIL | J2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and HPI. For NAND/SmartMedia/xD, it is Address Latch Enable output (ALE). |
| EM_A[2]/ (CLE)/ HCNTL0 | J1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and HPI. For NAND/SmartMedia/xD, this pin is the Command Latch Enable output (CLE). |
| EM_WAIT/ (RDY/$\overline{BSY}$)/ IORDY/ $\overline{HRDY}$ | F1 | I/O/Z | IPU $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For NAND/SmartMedia/xD, it is ready/busy input (RDY/$\overline{BSY}$). |
| $\overline{EM\_OE}$/ ( $\overline{RE}$ )/ ( $\overline{IORD}$ )/ $\overline{DIOR}$/ $\overline{HDS1}$ | H4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For NAND/SmartMedia/xD, it is read enable output ($\overline{RE}$). |
| $\overline{EM\_WE}$ ($\overline{WE}$) ($\overline{IOWR}$)/ $\overline{DIOW}$/ $\overline{HDS2}$ | G2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For NAND/SmartMedia/xD, it is write enable output ($\overline{WE}$). |
| $\overline{EM\_CS2}$/ $\overline{HCS}$ | C2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and HPI. For EMIFA, this pin is Chip Select 2 output $\overline{EM\_CS2}$ for use with asynchronous memories (i.e. NOR flash) or NAND flash. This is the chip select for the default boot and ROM boot modes. |
| $\overline{EM\_CS3}$ | B1 | I/O/Z | $DV_{DD18}$ | For EMIFA, this pin is Chip Select 3 output $\overline{EM\_CS3}$ for use with asynchronous memories (i.e. NOR flash) or NAND flash. |
| $\overline{EM\_CS4}$/ GPIO9/ VLYNQ_SCRUN | T2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is Chip Select 4 output $\overline{EM\_CS4}$ for use with asynchronous memories (i.e., NOR flash) or NAND flash. |
| $\overline{EM\_CS5}$/ GPIO8/ VLYNQ_CLOCK | T1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For EMIFA, it is Chip Select 5 output $\overline{EM\_CS5}$ for use with asynchronous memories (i.e., NOR flash) or NAND flash. |

**Table 2-9. EMIFA Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | OTHER[2] [3] | DESCRIPTION |
|---|---|---|---|---|
| EM_D0/ DD0/ HD0 | E5 | I/O/Z | $DV_{DD18}$ | These pins are multiplexed between EMIFA (NAND), ATA/CF, and HPI. In all cases they are used as a 16 bit bi-directional data bus. For EMIFA (NAND), these are EM_D[15:0]. |
| EM_D1/ DD1/ HD1 | D3 | I/O/Z | $DV_{DD18}$ | |
| EM_D2/ DD2/ HD2 | F5 | I/O/Z | $DV_{DD18}$ | |
| EM_D3/ DD3/ HD3 | E3 | I/O/Z | $DV_{DD18}$ | |
| EM_D4/ DD4/ HD4 | E4 | I/O/Z | $DV_{DD18}$ | |
| EM_D5/ DD5/ HD5 | D2 | I/O/Z | $DV_{DD18}$ | |
| EM_D6/ DD6/ HD6 | F4 | I/O/Z | $DV_{DD18}$ | |
| EM_D7/ DD7/ HD7 | C1 | I/O/Z | $DV_{DD18}$ | |
| EM_D8/ DD8/ HD8 | F3 | I/O/Z | $DV_{DD18}$ | |
| EM_D9/ DD9/ HD9 | E2 | I/O/Z | $DV_{DD18}$ | |
| EM_D10/ DD10/ HD10 | G5 | I/O/Z | $DV_{DD18}$ | |
| EM_D11/ DD11/ HD11 | G4 | I/O/Z | $DV_{DD18}$ | |
| EM_D12/ DD12/ HD12 | D1 | I/O/Z | $DV_{DD18}$ | |
| EM_D13/ DD13/ HD13 | F2 | I/O/Z | $DV_{DD18}$ | |
| EM_D14/ DD14/ HD14 | H5 | I/O/Z | $DV_{DD18}$ | |
| EM_D15/ DD15/ HD15 | E1 | I/O/Z | $DV_{DD18}$ | |

**Table 2-10. DDR2 Memory Controller Terminal Functions**

| SIGNAL NAME | SIGNAL NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **DDR2 Memory Controller** | | | | |
| DDR_CLK0 | W7 | I/O/Z | $DV_{DDR2}$ | DDR2 Clock |
| $\overline{\text{DDR\_CLK0}}$ | W8 | I/O/Z | $DV_{DDR2}$ | DDR2 Differential clock |
| DDR_CKE | V8 | I/O/Z | $DV_{DDR2}$ | DDR2 Clock Enable |
| $\overline{\text{DDR\_CS}}$ | T9 | I/O/Z | $DV_{DDR2}$ | DDR2 Active low chip select |
| $\overline{\text{DDR\_WE}}$ | T8 | I/O/Z | $DV_{DDR2}$ | DDR2 Active low Write enable |
| DDR_DQM[3] | T16 | I/O/Z | $DV_{DDR2}$ | DDR2 Data mask outputs<br>DQM3: For upper byte data bus DDR_D[31:24]<br>DQM2: For DDR_D[23:16]<br>DQM1: For DDR_D[15:8]<br>DQM0: For lower byte DDR_D[7:0] |
| DDR_DQM[2] | T14 | I/O/Z | $DV_{DDR2}$ | |
| DDR_DQM[1] | T6 | I/O/Z | $DV_{DDR2}$ | |
| DDR_DQM[0] | T4 | I/O/Z | $DV_{DDR2}$ | |
| $\overline{\text{DDR\_RAS}}$ | U7 | I/O/Z | $DV_{DDR2}$ | DDR2 Row Access Signal output |
| $\overline{\text{DDR\_CAS}}$ | T7 | I/O/Z | $DV_{DDR2}$ | DDR2 Column Access Signal output |
| DDR_DQS[0] | U4 | I/O/Z | $DV_{DDR2}$ | Data strobe input/outputs for each byte of the 32-bit data bus. They are outputs to the DDR2 memory when writing and inputs when reading. They are used to synchronize the data transfers.<br>DQS3 : For upper byte DDR_D[31:24]<br>DQS2: For DDR_D[23:16]<br>DQS1: For DDR_D[15:8]<br>DQS0: For bottom byte DDR_D[7:0] |
| DDR_DQS[1] | U6 | I/O/Z | $DV_{DDR2}$ | |
| DDR_DQS[2] | U14 | I/O/Z | $DV_{DDR2}$ | |
| DDR_DQS[3] | U16 | I/O/Z | $DV_{DDR2}$ | |
| DDR_BS[0] | U8 | I/O/Z | $DV_{DDR2}$ | Bank select outputs (BS[2:0]). Two are required to support 1Gb DDR2 memories. |
| DDR_BS[1] | V9 | | | |
| DDR_BS[2] | U9 | | | |
| DDR_A[12] | W9 | I/O/Z | $DV_{DDR2}$ | DDR2 address bus |
| DDR_A[11] | W10 | | | |
| DDR_A[10] | U10 | | | |
| DDR_A[9] | U11 | | | |
| DDR_A[8] | V10 | | | |
| DDR_A[7] | V11 | | | |
| DDR_A[6] | W11 | | | |
| DDR_A[5] | W12 | | | |
| DDR_A[4] | V12 | | | |
| DDR_A[3] | U12 | | | |
| DDR_A[2] | V13 | | | |
| DDR_A[1] | U13 | | | |
| DDR_A[0] | W13 | | | |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal
(3) For more information, see Section 5.2, *Recommended Operating Conditions*.

**Table 2-10. DDR2 Memory Controller Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | OTHER[2] [3] | DESCRIPTION |
|---|---|---|---|---|
| DDR_D[31] | U19 | I/O/Z | $DV_{DDR2}$ | DDR2 data bus can be configured as 32 bits wide or 16 bits wide. |
| DDR_D[30] | V19 | | | |
| DDR_D[29] | W18 | | | |
| DDR_D[28] | V18 | | | |
| DDR_D[27] | W17 | | | |
| DDR_D[26] | U18 | | | |
| DDR_D[25] | U17 | | | |
| DDR_D[24] | V17 | | | |
| DDR_D[23] | T17 | | | |
| DDR_D[22] | V16 | | | |
| DDR_D[21] | W16 | | | |
| DDR_D[20] | U15 | | | |
| DDR_D[19] | V15 | | | |
| DDR_D[18] | W15 | | | |
| DDR_D[17] | V14 | | | |
| DDR_D[16] | W14 | | | |
| DDR_D[15] | V7 | | | |
| DDR_D[14] | W6 | | | |
| DDR_D[13] | V6 | | | |
| DDR_D[12] | W5 | | | |
| DDR_D[11] | V5 | | | |
| DDR_D[10] | U5 | | | |
| DDR_D[9] | W4 | | | |
| DDR_D[8] | V4 | | | |
| DDR_D[7] | W3 | | | |
| DDR_D[6] | V3 | | | |
| DDR_D[5] | U3 | | | |
| DDR_D[4] | W2 | | | |
| DDR_D[3] | V2 | | | |
| DDR_D[2] | V1 | | | |
| DDR_D[1] | U2 | | | |
| DDR_D[0] | U1 | | | |
| DDR_VREF | T15 | I | (4) | Reference voltage input for the SSTL_18 IO buffers. |
| DDR_VSSDLL | T11 | GND | (4) | Ground for the DDR2 Digital Locked Loop. |
| DDR_VDDDLL | T10 | S | (4) | Power (1.8 Volts) for the DDR2 Digital Locked Loop. |
| DDR_ZN | T12 | O/Z | (4) | Impedance control for DDR2 outputs. This must be connected via a 200 Ω resistor to $DV_{DDR2}$. |
| DDR_ZP | T13 | O/Z | (4) | Impedance control for DDR2 outputs. This must be connected via a 200 Ω resistor to $V_{SS}$. |

(4) For more information, see Section 5.2, *Recommended Operating Conditions*.

### Table 2-11. I2C Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **I2C** | | | | |
| SCL/ GPIO43 | C4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between I2C and GPIO. For I2C, it is clock output SCL. |
| SDA/ GPIO44 | B4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between I2C and GPIO. For I2C, it is bi-directional data signal SDA. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

### Table 2-12. Audio Serial Port (ASP) Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **Audio Serial Port (ASP)** | | | | |
| CLKX/ GPIO29 | B8 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between ASP and GPIO. For ASP, it is Transmit clock IO CLKX. |
| CLKR/ GPIO30 | A8 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between ASP and GPIO. For ASP, it is Receive clock IO CLKR. |
| FSX/ GPIO31 | C8 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between ASP and GPIO. For ASP, it is Transmit frame synchronization IO FSX. |
| FSR/ GPIO32 | C7 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between ASP and GPIO. For ASP, it is Receive frame synchronization IO FSR. |
| DX/ GPIO33 | B7 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between ASP and GPIO. For ASP, it is Data Transmit output DX. |
| DR/ GPIO34 | A7 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between ASP and GPIO. For ASP, it is Data Receive input DR. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

### Table 2-13. SPI Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **Serial Peripheral Interface (SPI)** | | | | |
| SPI_EN0/ GPIO37 | A4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between SPI and GPIO. When used by SPI, it is SPI slave device 0 enable output SPI_EN0. |
| SPI_EN1/ HDDIR/ GPIO42 | B2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between SPI, ATA, and GPIO. When used by SPI, it is SPI slave device 1 enable output SPI_EN1. |
| SPI_CLK/ GPIO39 | A3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between SPI and GPIO. For SPI, it is clock output SPI_CLK. |
| SPI_DI/ GPIO40 | B3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between SPI and GPIO. For SPI, it is data input SPI_DI. |
| SPI_DO/ GPIO41 | A2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between SPI and GPIO. For SPI it is data output SPI_DO. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

**Table 2-14. EMAC and MDIO Terminal Functions**

| SIGNAL NAME | SIGNAL NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **EMAC** | | | | |
| GPIOV33_0/ TXEN | B13 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Transmit Enable output TXEN. |
| GPIOV33_1/ TXCLK | A13 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Transmit Clock input TXCLK. |
| GPIOV33_2/ COL | A12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Collision Detect input COL. |
| GPIOV33_6/ TXD3 | C12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Transmit Data 3 output TXD3. |
| GPIOV33_5/ TXD2 | A11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Transmit Data 2 output TXD2. |
| GPIOV33_4/ TXD1 | D12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Transmit Data 1 output TXD1. |
| GPIOV33_3/ TXD0 | B12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Transmit Data 0 output TXD0. |
| GPIOV33_11/ RXCLK | A10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Receive Clock input RXCLK. |
| GPIOV33_12/ RXDV | D11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Receive Data Valid input RXDV. |
| GPIOV33_13/ RXER | D10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Receive Error input RXER. |
| GPIOV33_14/ CRS | C10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Carrier Sense input CRS. |
| GPIOV33_10/ RXD3 | E11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Receive Data 3 input RXD3. |
| GPIOV33_9/ RXD2 | B11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Receive Data 2 input RXD2. |
| GPIOV33_8/ RXD1 | C11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Receive data 1 input RXD1. |
| GPIOV33_7/ RXD0 | E12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Receive Data 0 input RXD0. |
| **MDIO** | | | | |
| GPIOV33_16/ MDCLK | B10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Management Data Clock output MDCLK. |
| GPIOV33_15/ MDIO | E10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In Ethernet MAC mode, it is Management Data IO MDIO. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

**Table 2-15. GPIOV33 Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **GPIOV33** | | | | |
| GPIOV33_16/ MDCLK | B10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_16. |
| GPIOV33_15/ MDIO | E10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_15. |
| GPIOV33_14/ CRS | C10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_14. |
| GPIOV33_13/ RXER | D10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_13. |
| GPIOV33_12/ RXDV | D11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_12. |
| GPIOV33_11/ RXCLK | A10 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_11. |
| GPIOV33_10/ RXD3 | E11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_10. |
| GPIOV33_9/ RXD2 | B11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_9. |
| GPIOV33_8/ RXD1 | C11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_8. |
| GPIOV33_7/ RXD0 | E12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_7. |
| GPIOV33_6/ TXD3 | C12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_6. |
| GPIOV33_5/ TXD2 | A11 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_5. |
| GPIOV33_4/ TXD1 | D12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_4. |
| GPIOV33_3/ TXD0 | B12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_3. |
| GPIOV33_2/ COL | A12 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_2. |
| GPIOV33_1/ TXCLK | A13 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, it is 3.3V GPIO GPIOV33_1. |
| GPIOV33_0/ TXEN | B13 | I/O/Z | $DV_{DD33}$ | This pin is multiplexed between GPIO and Ethernet MAC. In GPIO mode, this pin is 3.3V GPIO pin GPIOV33_0. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

**Table 2-16. Standalone GPIOV18 Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **Standalone GPIOV18** | | | | |
| GPIO7 | C3 | I/O/Z | $DV_{DD18}$ | This pin is standalone and functions as GPIO7. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

**Table 2-17. USB Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **USB 2.0** | | | | |
| M24XI | F18 | I | $DV_{DD18}$ | Crystal input for M24 oscillator (24 MHz for USB).<br>If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, this is the external oscillator clock input.<br>When the USB peripheral ***is not*** used, M24XI should be left as a No Connect. |
| M24XO | F19 | O | $DV_{DD18}$ | Crystal output for M24 oscillator.<br>If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, M24XO should be left as a No Connect.<br>When the USB peripheral ***is not*** used, M24XO should be left as a No Connect. |
| $M24V_{DD}$ | F16 | S | (3) | 1.8-V power supply for M24 oscillator.<br>If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, $M24V_{DD}$ should still be connected to the 1.8-V power supply.<br>When the USB peripheral ***is not*** used, $M24V_{DD}$ should be connected to the 1.8-V power supply. |
| $M24V_{SS}$ | F17 | GND | (3) | Ground for M24 oscillator.<br>If a crystal input ***is not*** used, but instead a physical clock-in source is supplied, $M24V_{SS}$ should still be connected to ground.<br>When the USB peripheral ***is not*** used, $M24V_{SS}$ should be connected to ground. |
| USB_VBUS | J17 | A I/O | (3) | 5-V input that signifies that VBUS is connected.<br>When the USB peripheral ***is not*** used, the USB_VBUS signal should be either pulled down or pulled up via a 10-kΩ resistor. |
| USB_ID | J16 | A I/O | | USB operating mode identification pin. For Host mode operation, pull down this pin to ground ($V_{SS}$) via an external 1.5-kΩ resistor. For Device mode operation, pull up this pin to $DV_{DD33}$ rail via an external 1.5-kΩ resistor.<br>When the USB peripheral ***is not*** used, the USB_ID signal should be either pulled down or pulled up via a 10-kΩ resistor. |
| USB_DP | G19 | A I/O | | USB bi-directional Data Differential signal pair [positive/negative]. |
| USB_DM | H19 | A I/O | | When the USB peripheral ***is not*** used, the USB_DP signal should be pulled high and the USB_DM signal should be pulled down via a 10-kΩ resistor. |
| USB_R1 | H18 | A I/O | (3) | Reference current output. This must be connected via a 10-kΩ ±1% resistor to $USB\_V_{SSREF}$.<br>When the USB peripheral ***is not*** used, the USB_R1 signal should be connected via a 10-kΩ resistor to $USB\_V_{SSREF}$. |
| $USB\_V_{SSREF}$ | G16 | GND | (3) | Ground for reference current. This must be connected via a 10-kΩ ±1% resistor to USB_R1.<br>When the USB peripheral ***is not*** used, the $USB\_V_{SSREF}$ signal should be connected to $V_{SS}$. |
| $USB\_V_{DDA3P3}$ | J19 | S | (3) | Analog 3.3 V power supply for USB phy.<br>When the USB peripheral ***is not*** used, the $USB\_V_{DDA3P3}$ signal should be connected to $DV_{DD33}$. |
| $USB\_V_{SSA3P3}$ | J18 | GND | (3) | Analog ground for USB phy. When the USB peripheral ***is not*** used, the $USB\_V_{SSA3P3}$ signal should be connected to $V_{SS}$. |
| $USB\_V_{DD1P8}$ | H17 | S | (3) | 1.8-V I/O power supply for USB phy.<br>When the USB peripheral ***is not*** used, the $USB\_V_{DD1P8}$ signal should be connected to $DV_{DD18}$. |
| $USB\_V_{SS1P8}$ | H16 | GND | (3) | I/O Ground for USB phy.<br>When the USB peripheral ***is not*** used, the $USB\_V_{SS1P8}$ signal should be connected to $V_{SS}$. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal
(3) For more information, see Section 5.2, *Recommended Operating Conditions*.

**Table 2-17. USB Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| USB_$V_{DDA1P2LDO}$ | G18 | S | (3) | Core Power supply LDO output for USB phy. This must be connected via a 1-µF capacitor to $V_{SS}$.<br>When the USB peripheral ***is not*** used, the USB_$V_{DDA1P2LDO}$ signal should still be connected via a 1-µF capacitor to $V_{SS}$. |
| USB_$V_{SSA1P2LDO}$ | G17 | GND | (3) | Core Ground for USB phy. This is the ground for the LDO and must be connected to $V_{SS}$.<br>When the USB peripheral ***is not*** used, the USB_$V_{SSA1P2LDO}$ signal should still be connected to $V_{SS}$. |

**Table 2-18. VLYNQ Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **VLYNQ** | | | | |
| $\overline{\text{EM\_CS5}}$/ GPIO8/ VLYNQ_CLOCK | T1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is the clock (VLYNQ_CLOCK). |
| $\overline{\text{EM\_CS4}}$/ GPIO9/ VLYNQ_SCRUN | T2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is the Serial Clock run request (VLYNQ_SCRUN). |
| EM_A[15]/ GPIO16/ VLYNQ_TXD3 | P3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is transmit bus bit 3 output VLYNQ_TXD3. |
| EM_A[17]/ GPIO14/ VLYNQ_TXD2 | R2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is transmit bus bit 2 output VLYNQ_TXD2. |
| EM_A[19]/ GPIO12/ VLYNQ_TXD1 | R4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is transmit bus bit 1 output VLYNQ_TXD1. |
| EM_A[21]/ GPIO10/ VLYNQ_TXD0 | T3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is bit 0 of the transmit bus (VLYNQ_TXD0). |
| EM_A[14]/ GPIO17/ VLYNQ_RXD3 | P4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is receive bus bit 3 input VLYNQ_RXD3. |
| EM_A[16]/ GPIO15/ VLYNQ_RXD2 | R5 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is receive bus bit 2 input VLYNQ_RXD2. |
| EM_A[18]/ GPIO13/ VLYNQ_RXD1 | P5 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is receive bus bit 1 input VLYNQ_RXD1. |
| EM_A[20]/ GPIO11/ VLYNQ_RXD0 | R3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, GPIO, and VLYNQ. For VLYNQ, it is receive bus bit 0 input VLYNQ_RXD0. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

**Table 2-19. VPFE Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **VIDEO/IMAGE IN (VPFE)** | | | | |
| PCLK | M19 | I | – $DV_{DD18}$ | Pixel clock input used to load image data into the CCD Controller (CCDC) on pins CI[7:0] and YI[7:0]. |
| VD | L19 | I/O/Z | – $DV_{DD18}$ | Vertical synchronization signal that can be either an input (slave mode) or an output (master mode), which signals the start of a new frame to the CCDC. |
| HD | M18 | I/O/Z | – $DV_{DD18}$ | Horizontal synchronization signal that can be either an input (slave mode) or an output (master mode), which signals the start of a new line to the CCDC. |
| CI7/ CCD15/ UART_RXD2 | N19 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between the CCDC and UART2.<br>When used by the CCDC as input CI7, it supports several modes.<br>In 16-bit CCD Analog-Front-End (AFE) mode, it is input CCD15.<br>In 16-bit YCbCr mode, it is time multiplexed between CB7 and CR7 inputs.<br>In 8-bit YCbCr mode, it is time multiplexed between Y7, CB7, and CR7 of the upper 8-bit channel. |
| CI6/ CCD14/ UART_TXD2 | N18 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between the CCDC and UART2.<br>When used by the CCDC as input CI6, it supports several modes. In 16-bit CCD AFE mode, it is input CCD14.<br>In 16-bit YCbCr mode, it is time multiplexed between CB6 and CR6 inputs.<br>In 8-bit YCbCr mode, it is time multiplexed between Y6, CB6, and CR6 of the upper 8-bit channel. |
| CI5/ CCD13/ UART_CTS2 | N17 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between the CCDC and UART2. When used by the CCDC as input CI5, it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD13.<br>In 16-bit YCbCr mode, it is time multiplexed between CB5 and CR5 inputs.<br>In 8-bit YCbCr mode, it is time multiplexed between Y5, CB5, and CR5 of the upper 8-bit channel. |
| CI4/ CCD12/ UART_RTS2 | N16 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between the CCDC and UART2. When used by the CCDC as input CI4, it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD12.<br>In 16-bit YCbCr mode, it is time multiplexed between CB4 and CR4 inputs.<br>In 8-bit YCbCr mode, it is time multiplexed between Y4, CB4, and CR4 of the upper 8-bit channel. |
| CI3/ CCD11 | N15 | I | IPD $DV_{DD18}$ | This pin is CCDC input CI3 and it supports several modes. In 16-bit CCD AFE mode, it is input CCD11.<br>In 16-bit YCbCr mode, it is time multiplexed between CB3 and CR3 inputs.<br>In 8-bit YCbCr mode, it is time multiplexed between Y3, CB3, and CR3 of the upper 8-bit channel. |
| CI2/ CCD10 | M17 | I | IPD $DV_{DD18}$ | This pin is CCDC input CI2 and it supports several modes. In 16-bit CCD AFE mode, it is input CCD10.<br>In 16-bit YCbCr mode, it is time multiplexed between CB2 and CR2 inputs.<br>In 8-bit YCbCr mode, it is time multiplexed between Y2, CB2, and CR2 of the upper 8-bit channel. |
| CI1/ CCD9 | M16 | I | IPD $DV_{DD18}$ | This pin is CCDC input CI1 and it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD9.<br>In 16-bit YCbCr mode, it is time multiplexed between CB1 and CR1 inputs.<br>In 8-bit YCbCr mode, it is time multiplexed between Y1, CB1, and CR1 of the upper 8-bit channel. |
| CI0/ CCD8 | M15 | I | IPD $DV_{DD18}$ | This pin is CCDC input CI0 and it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD8.<br>In 16-bit YCbCr mode, it is time multiplexed between CB0 and CR0 inputs.<br>In 8-bit YCbCr mode, it is time multiplexed between Y0, CB0, and CR0 of the upper 8-bit channel. |
| YI7/ CCD7 | L18 | I | IPD $DV_{DD18}$ | This pin is CCDC input YI7 and it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD7.<br>In 16-bit YCbCr mode, it is input Y7.<br>In 8-bit YCbCr mode, it is time multiplexed between Y7, CB7, and CR7 of the lower 8-bit channel. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(3) Specifies the operating I/O supply voltage for each signal

**Table 2-19. VPFE Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| YI6/ CCD6 | L17 | I | IPD $DV_{DD18}$ | This pin is CCDC input YI6 and it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD6.<br>In 16-bit YCbCr mode, it is input Y6.<br>In 8-bit YCbCr mode, it is time multiplexed between Y6, CB6, and CR6 of the lower 8-bit channel. |
| YI5/ CCD5 | L16 | I | IPD $DV_{DD18}$ | This pin is CCDC input YI5 and it supports several modes. In 16-bit CCD AFE mode, it is input CCD5.<br>In 16-bit YCbCr mode, it is input Y5.<br>In 8-bit YCbCr mode, it is time multiplexed between Y5, CB5, and CR5 of the lower 8-bit channel. |
| YI4/ CCD4 | L15 | I | IPD $DV_{DD18}$ | This pin is CCDC input YI4 and it supports several modes.<br>In 16-bit CCD Analog-Front-End (AFE) mode, it is input CCD4.<br>In 16-bit YCbCr mode, it is input Y4.<br>In 8-bit YCbCr mode, it is time multiplexed between Y4, CB4, and CR4 of the lower 8-bit channel. |
| YI3/ CCD3 | K19 | I | IPD $DV_{DD18}$ | This pin is CCDC input YI3 and it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD3.<br>In 16-bit YCbCr mode, it is input Y3.<br>In 8-bit YCbCr mode, it is time multiplexed between Y3, CB3, and CR3 of the lower 8-bit channel. |
| YI2/ CCD2 | K18 | I | IPD $DV_{DD18}$ | This pin is CCDC input YI2 and it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD2.<br>In 16-bit YCbCr mode, it is input Y2.<br>In 8-bit YCbCr mode, it is time multiplexed between Y2, CB2, and CR2 of the lower 8-bit channel. |
| YI1/ CCD1 | K17 | I | IPD $DV_{DD18}$ | This pin is CCDC input YI1 and it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD1.<br>In 16-bit YCbCr mode, it is input Y1.<br>In 8-bit YCbCr mode, it is time multiplexed between Y1, CB1, and CR1 of the lower 8-bit channel. |
| YI0/ CCD0 | K16 | I | IPD $DV_{DD18}$ | This pin is CCDC input YI0 and it supports several modes.<br>In 16-bit CCD AFE mode, it is input CCD0.<br>In 16-bit YCbCr mode, it is input Y0.<br>In 8-bit YCbCr mode, it is time multiplexed between Y0, CB0, and CR0 of the lower 8-bit channel. |
| GPIO1/ C_WE | E13 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between GPIO and the VPFE.<br>In VPFE mode, it is the CCD Controller write enable input C_WE. |
| GPIO4/ R0/ C_FIELD | B14 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between GPIO, the VPFE, and the VPBE.<br>In VPFE mode, it is CCDC field identification bidirectional signal C_FIELD. |

**Table 2-20. VPBE Terminal Functions**

| SIGNAL NAME | SIGNAL NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **VIDEO OUT (VPBE)** | | | | |
| HSYNC | C17 | I/O/Z | IPD $DV_{DD18}$ | VPBE Horizontal Sync signal that can be either an input or an output. |
| VSYNC | C18 | I/O/Z | IPD $DV_{DD18}$ | VPBE Vertical Sync signal that can be either an input or an output. |
| VCLK | D19 | I/O/Z | $DV_{DD18}$ | VPBE Clock Output |
| VPBECLK | C19 | I/O/Z | IPD $DV_{DD18}$ | VPBE Clock Input |
| COUT0/ B3/ BTSEL0 | A16 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between ARM boot mode and the VPBE. After reset, this pin is either video encoder outputs COUT0, or RGB666/888 Blue output data bits 3, B3. |
| COUT1/ B4/ BTSEL1 | B16 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between ARM boot mode and the VPBE. After reset, this pin is either video encoder outputs COUT1, or RGB666/888 Blue output data bits 4, B4. |
| COUT2/ B5/ EM_WIDTH | A17 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between EMIFA and the VPBE. After reset, it is video encoder output COUT2 or RGB666/888 Blue output data bit 5 B5. |
| COUT3/ B6/ DSP_BT | B17 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between DSP boot and the VPBE. After reset, it is video encoder output COUT3 or RGB666/888 Blue data bit 6 output B6. |
| COUT4/ B7 | A18 | O | $DV_{DD18}$ | Video encoder output COUT4 or RGB666/888 Blue data bit 7 output B7. |
| COUT5/ G2 | B18 | O | $DV_{DD18}$ | Video encoder output COUT5 or RGB666/888 Green data bit 2 output G2. |
| COUT6/ G3 | B19 | O | $DV_{DD18}$ | Video encoder output COUT6 or RGB666/888 Green data bit 3 output G3. |
| COUT7/ G4 | C16 | O | $DV_{DD18}$ | Video encoder output COUT7 or RGB666/888 Green data bit 4 output G4. |
| YOUT0/ G5/ AEAW0 | D15 | I/O/Z | IPD $DV_{DD18}$ | These pins are multiplexed between EMIFA and the VPBE. After reset, these are video encoder outputs YOUT[0:4] or RGB666/888 Red and Green data bit outputs G5, G6, G7, R3, and R4. |
| YOUT1/ G6/ AEAW1 | D16 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT2/ G7/ AEAW2 | D17 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT3/ R3/ AEAW3 | D18 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT4/ R4/ AEAW4 | E15 | I/O/Z | IPD $DV_{DD18}$ | |
| YOUT5/ R5 | E16 | O | $DV_{DD18}$ | Video encoder output YOUT5 or RGB666/888 Red data bit 5 output R5. |
| YOUT6/ R6 | E17 | O | $DV_{DD18}$ | Video encoder output YOUT6 or RGB666/888 Red data bit 6 output R6. |
| YOUT7/ R7 | E18 | O | $DV_{DD18}$ | Video encoder output YOUT7 or RGB666/888 Red data bit 7 output R7. |
| GPIO0/ LCD_OE | C13 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between GPIO and the VPBE. In VPBE mode, it is the LCD output enable LCD_OE. |
| GPIO2/ G0 | D13 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between GPIO and the VPBE. In VPBE mode, it is RGB888 Green data bit 0 output G0. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(3) Specifies the operating I/O supply voltage for each signal

**Table 2-20. VPBE Terminal Functions (continued)**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| GPIO3/ B0/ LCD_FIELD | C14 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between GPIO, and the VPBE. In VPBE mode, it is RGB888 Blue data bit 0 output B0 or LCD interlaced bidirectional LCD_FIELD. |
| GPIO4/ R0/ C_FIELD | B14 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between GPIO, the VPFE, and the VPBE. In VPBE mode, it is RGB888 Red data bit 0 output R0. |
| GPIO5/ G1 | E14 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between GPIO and the VPBE. In VPBE mode, it is RGB888 Green data bit 1 output G1. |
| GPIO6/ B1 | A14 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between GPIO and the VPBE. In VPBE mode, it is RGB888 Blue data bit 1 output B1. |
| GPIO38/ R1 | D14 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between VPBE and GPIO. In VPBE mode, it is RGB888 Red output data bit 1. |
| PWM1/ R2/ GPIO46 | B15 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between PWM1, VPBE, and GPIO. In VPBE mode, it is RGB888 Red output bit 2 (R2). |
| PWM2/ B2/ GPIO47 | A15 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between PWM2, VPBE, and GPIO. In VPBE mode, it is RGB888 Blue output bit 2 (B2). |

**Table 2-21. DAC [Part of VPBE] Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **DAC[A:D]** | | | | |
| DAC_VREF | R17 | A I | (3) | Reference voltage input (0.5 V). When the DAC is not used, the DAC_VREF signal should be connected to $V_{SS}$. |
| DAC_IOUT_A | P19 | A O | | Output of DAC A. When the DAC is not used, the DAC_IOUT_A signal should be left as a No Connect. |
| DAC_IOUT_B | P18 | A O | | Output of DAC B. When the DAC is not used, the DAC_IOUT_B signal should be left as a No Connect. |
| DAC_IOUT_C | R19 | A O | | Output of DAC C. When the DAC is not used, the DAC_IOUT_C signal should be left as a No Connect. |
| DAC_IOUT_D | T19 | A O | | Output of DAC D. When the DAC is not used, the DAC_IOUT_D signal should be left as a No Connect. |
| $V_{DDA\_1P8V}$ | R18 | S | (3) | 1.8-V analog I/O power. When the DAC is not used, the $V_{DDA\_1P8V}$ signal should be connected to $V_{SS}$. |
| $V_{SSA\_1P8V}$ | P17 | GND | (3) | Analog I/O ground. When the DAC is not used, the $V_{SSA\_1P8V}$ signal should be connected to $V_{SS}$. |
| $V_{DDA\_1P1V}$ | P16 | S | (3) | 1.20-V analog core supply voltage (A-513, -594, -810 devices). When the DAC is not used, the $V_{DDA\_1P1V}$ signal should be connected to $V_{SS}$. |
| $V_{SSA\_1P1V}$ | T18 | GND | (3) | Analog core ground. When the DAC is not used, the $V_{SSA\_1P1V}$ signal should be connected to $V_{SS}$. |
| DAC_RBIAS | R16 | A I | (3) | External resistor connection for current bias configuration. This pin must be connected via a 4-kΩ resistor to $V_{SSA\_1P8V}$. When the DAC is not used, the DAC_RBIAS signal should be connected to $V_{SS}$. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal
(3) For more information, see Section 5.2, *Recommended Operating Conditions*.

### Table 2-22. UART0, UART1, UART2 Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **UART2** | | | | |
| CI7/ CCD15/ UART_RXD2 | N19 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between the CCDC and UART2. When used by UART2 it is the receive data input UART_RXD2. |
| CI6/ CCD14/ UART_TXD2 | N18 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between the CCDC and UART2. In UART2 mode, it is the transmit data output UART_TXD2. |
| CI5/ CCD13/ UART_CTS2 | N17 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between the CCDC and UART2. In UART2 mode, it is the clear to send input UART_CTS2. |
| CI4/ CCD12/ UART_RTS2 | N16 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between the CCDC and UART2. In UART2 mode, it is the ready to send output UART_RTS2. |
| **UART1** | | | | |
| DMACK/ UART_TXD1 | H3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between ATA/CF and UART1. For UART1, it is transmit data output UART_TXD1. |
| DMARQ/ UART_RXD1 | G1 | I/O/Z | IPD $DV_{DD18}$ | This pin is multiplexed between ATA/CF and UART1. For UART1, it is receive data input UART_RXD1. |
| **UART0** | | | | |
| UART_RXD0/ GPIO35 | D5 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between UART0 and GPIO. For UART0, it is receive data input UART_RXD0. |
| UART_TXD0/ GPIO36 | C5 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between UART0 and GPIO. . For UART0, it is transmit data output UART_TXD0. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(3) Specifies the operating I/O supply voltage for each signal

### Table 2-23. PWM0, PWM1, PWM2 Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **PWM2** | | | | |
| PWM2/ B2/ GPIO47 | A15 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between PWM2, VPBE, and GPIO. For PWM2, it is output PWM2. |
| **PWM1** | | | | |
| PWM1/ R2/ GPIO46 | B15 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between PWM1, VPBE, and GPIO. For PWM1, it is output PWM1. |
| **PWM0** | | | | |
| PWM0/ GPIO45 | C15 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between PWM0 and GPIO. For PWM0, it is output PWM0. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

**Table 2-24. ATA/CF Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **ATA/CF** | | | | |
| SPI_EN1/ HDDIR/ GPIO42 | B2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between SPI, ATA, and GPIO. For ATA, it is buffer direction control output HDDIR. |
| GPIO50/ ATA_CS0 | J5 | O | $DV_{DD18}$ | This pin is multiplexed between GPIO and ATA/CF. In ATA mode, it is ATA/CF chip select output ATA_CS0. |
| GPIO51/ ATA_CS1 | H1 | O | $DV_{DD18}$ | This pin is multiplexed between GPIO and ATA/CF. In ATA mode, it is ATA/CF chip select output ATA_CS1. |
| EM_R/W/ INTRQ/ H/W | G3 | I | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, and HPI. For ATA/CF, it is interrupt request input INTRQ. |
| EM_WAIT/ (RDY/BSY)/ IORDY/ HRDY | F1 | I | IPU $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For ATA/CF, it is IO Ready input IORDY. |
| EM_OE/ ( RE )/ ( IORD )/ DIOR/ HDS1 | H4 | O | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For CF, it is read strobe output (IORD). For ATA, it is read strobe output DIOR. |
| EM_WE (WE) (IOWR)/ DIOW/ HDS2 | G2 | O | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI. For CF, it is write strobe output (IOWR). For ATA, it is write strobe output DIOW. |
| DMACK/ UART_TXD1 | H3 | O | $DV_{DD18}$ | This pin is multiplexed between ATA/CF and UART1. For ATA/CF, it is DMA acknowledge output DMACK. |
| DMARQ/ UART_RXD1 | G1 | O | IPD $DV_{DD18}$ | This pin is multiplexed between ATA/CF and UART1. For ATA/CF, it is DMA request DMARQ input. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(3) Specifies the operating I/O supply voltage for each signal

**Table 2-24. ATA/CF Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | OTHER[2] [3] | DESCRIPTION |
|---|---|---|---|---|
| EM_D15/ DD15/ HD15 | E1 | I/O/Z | $DV_{DD18}$ | These pins are multiplexed between EMIFA (NAND), ATA/CF, and HPI. In all cases they are used as a 16 bit bi-directional data bus. For ATA/CF, these are DD[15:0]. |
| EM_D14/ DD14/ HD14 | H5 | | | |
| EM_D13/ DD13/ HD13 | F2 | | | |
| EM_D12/ DD12/ HD12 | D1 | | | |
| EM_D11/ DD11/ HD11 | G4 | | | |
| EM_D10/ DD10/ HD10 | G5 | | | |
| EM_D9/ DD9/ HD9 | E2 | | | |
| EM_D8/ DD8/ HD8 | F3 | | | |
| EM_D7/ DD7/ HD7 | C1 | | | |
| EM_D6/ DD6/ HD6 | F4 | | | |
| EM_D5/ DD5/ HD5 | D2 | | | |
| EM_D4/ DD4/ HD4 | E4 | | | |
| EM_D3/ DD3/ HD3 | E3 | | | |
| EM_D2/ DD2/ HD2 | F5 | | | |
| EM_D1/ DD1/ HD1 | D3 | | | |
| EM_D0/ DD0/ HD0 | E5 | | | |
| EM_A[0]/ DA2/ HCNTL1/ GPIO53 | J4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, HPI, and GPIO. For ATA/CF, it is Device address bit 2 output DA2. |
| EM_BA[1]/ DA1/ GPIO52 | H2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, and GPIO. For ATA/CF, it is Device address bit 1 output DA1. |
| EM_BA[0]/ $\overline{DA0}$/ HINT | J3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, HPI. For ATA/CF, it is Device address bit 0 output DA0. |

### Table 2-25. MMC/SD/SDIO Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **MMC/SD/SDIO** | | | | |
| SD_CLK | A9 | O | $DV_{DD33}$ | Data clock output SD_CLK |
| SD_CMD | B9 | I/O/Z | $DV_{DD33}$ | Bi-directional command IO SD_CMD |
| SD_DATA3 | C9 | I/O/Z | $DV_{DD33}$ | These pins are the nibble-wide bi-directional data bus SD_DATA[3:0]. |
| SD_DATA2 | D9 | I/O/Z | | |
| SD_DATA1 | E9 | I/O/Z | | |
| SD_DATA0 | D8 | I/O/Z | | |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

### Table 2-26. HPI Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **Host-Port Interface (HPI)** | | | | |
| $\overline{EM\_CS3}$ | B1 | I/O/Z | $DV_{DD18}$ | For EMIFA, this pin is Chip Select 3 output.<br>In HPI mode this pin must be pulled high via an external 10-kΩ resistor. |
| EM_BA[0]/ DA0/ $\overline{HINT}$ | J3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, and HPI.<br>In HPI mode, it is the host interrupt output $\overline{HINT}$. |
| EM_A[0]/ DA2/ HCNTL1/ GPIO53 | J4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, HPI, and GPIO.<br>For HPI, it is control input HCNTL1. The state of HCNTL1 and HCNTL0 determine if address, data, or control information is being transmitted between an external host and DM644X. |
| EM_A[2]/ (CLE)/ HCNTL0 | J1 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), and HPI.<br>In HPI mode, it is control input HCNTL0. The state of HCNTL1 and HCNTL0 determine if address, data, or control information is being transmitted between an external host and DM644X. |
| EM_A[1]/ (ALE)/ HHWIL | J2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), and HPI.<br>In HPI mode, it is Half-word identification input HHWIL. |
| EM_R/$\overline{W}$/ INTRQ/ HR/$\overline{W}$ | G3 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA, ATA/CF, and HPI.<br>For HPI, it is the Host Read Write input HR/$\overline{W}$. This signal is active high for reads and low for writes. |
| $\overline{EM\_CS2}$/ $\overline{HCS}$ | C2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA and HPI.<br>In HPI mode, this pin is HPI Active Low Chip Select input $\overline{HCS}$. |
| $\overline{EM\_WE}$ ($\overline{WE}$) ($\overline{IOWR}$)/ $\overline{DIOW}$/ $\overline{HDS2}$ | G2 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI.<br>For HPI, it is data strobe 2 input $\overline{HDS2}$. |
| $\overline{EM\_OE}$/ ($\overline{RE}$)/ ($\overline{IORD}$)/ $\overline{DIOR}$/ $\overline{HDS1}$ | H4 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI.<br>For HPI, it is data strobe 1 input $\overline{HDS1}$. |
| EM_WAIT/ (RDY/$\overline{BSY}$)/ $\overline{IORDY}$/ $\overline{HRDY}$ | F1 | I/O/Z | IPU $DV_{DD18}$ | This pin is multiplexed between EMIFA (NAND/SmartMedia/xD), ATA/CF, and HPI.<br>For HPI, it is ready output $\overline{HRDY}$. |

(1) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(2) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(3) Specifies the operating I/O supply voltage for each signal

**Table 2-26. HPI Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | OTHER[2] [3] | DESCRIPTION |
|---|---|---|---|---|
| EM_D15/ DD15/ HD15 | E1 | I/O/Z | $DV_{DD18}$ | These pins are multiplexed between EMIFA (NAND), ATA/CF, and HPI. In HPI mode, these are HD[15:0] and are multiplexed internally with the HPI address lines. |
| EM_D14/ DD14/ HD14 | H5 | | | |
| EM_D13/ DD13/ HD13 | F2 | | | |
| EM_D12/ DD12/ HD12 | D1 | | | |
| EM_D11/ DD11/ HD11 | G4 | | | |
| EM_D10/ DD10/ HD10 | G5 | | | |
| EM_D9/ DD9/ HD9 | E2 | | | |
| EM_D8/ DD8/ HD8 | F3 | | | |
| EM_D7/ DD7/ HD7 | C1 | | | |
| EM_D6/ DD6/ HD6 | F4 | | | |
| EM_D5/ DD5/ HD5 | D2 | | | |
| EM_D4/ DD4/ HD4 | E4 | | | |
| EM_D3/ DD3/ HD3 | E3 | | | |
| EM_D2/ DD2/ HD2 | F5 | | | |
| EM_D1/ DD1/ HD1 | D3 | | | |
| EM_D0/ DD0/ HD0 | E5 | | | |

### Table 2-27. Timer 0, Timer 1, and Timer 2 Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) | DESCRIPTION |
|---|---|---|---|---|
| **Timer 2 and Timer 1** | | | | |
| No external pins. The Timer 2 and Timer 1 peripheral pins are not pinned out as external pins. | | | | |
| **Timer 0** | | | | |
| CLK_OUT1/ TIM_IN/ GPIO49 | E19 | I/O/Z | $DV_{DD18}$ | This pin is multiplexed between the USB clock generator, timer, and GPIO. For Timer0, it is the timer event capture input TIM_IN. |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) Specifies the operating I/O supply voltage for each signal

### Table 2-28. Reserved Terminal Functions

| SIGNAL NAME | NO. | TYPE(1) | OTHER(2) (3) | DESCRIPTION |
|---|---|---|---|---|
| **RESERVED** | | | | |
| RSV1 | A1 | | | Reserved. (Leave unconnected, **do not** connect to power or ground) |
| RSV2 | A19 | | | Reserved. (Leave unconnected, **do not** connect to power or ground) |
| RSV3 | W1 | | | Reserved. (Leave unconnected, **do not** connect to power or ground) |
| RSV4 | W19 | | | Reserved. (Leave unconnected, **do not** connect to power or ground) |
| RSV5 | D4 | I | IPD $V_{SS}$ | Reserved. This pin **must** be tied directly to $V_{SS}$ for normal device operation. |
| RSV6 | L3 | A O | | Reserved. (Leave unconnected, **do not** connect to power or ground) |
| RSV7 | R8 | A | | Reserved. (Leave unconnected, **do not** connect to power or ground) |
| RSV24 | M3 | S | | Reserved. (Leave unconnected, **do not** connect to power or ground) |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal
(2) IPD = Internal pulldown, IPU = Internal pullup. (To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
(3) Specifies the operating I/O supply voltage for each signal

**Table 2-29. Supply Terminal Functions**

| SIGNAL NAME | NO. | TYPE(1) | OTHER | DESCRIPTION |
|---|---|---|---|---|
| **SUPPLY VOLTAGE PINS** | | | | |
| $DV_{DD33}$ | F10 | S | | 3.3 V I/O supply voltage (see Section 6.3.1.2, *Power-Supply Decoupling*, of this data manual) |
| | F11 | | | |
| | F12 | | | |
| | F13 | | | |
| $DV_{DD18}$ | N5 | S | | 1.8 V I/O supply voltage (see Section 6.3.1.2, *Power-Supply Decoupling*, of this data manual) |
| | G15 | | | |
| | F14 | | | |
| | J15 | | | |
| | H14 | | | |
| | K14 | | | |
| | M14 | | | |
| | L13 | | | |
| | G9 | | | |
| | F8 | | | |
| | E7 | | | |
| | G7 | | | |
| | J7 | | | |
| | L7 | | | |
| | F6 | | | |
| | H6 | | | |
| | K6 | | | |
| | M6 | | | |
| $DV_{DDR2}$ | T5 | S | | 1.8 V DDR2 I/O supply voltage (see Section 6.3.1.2, *Power-Supply Decoupling*, of this data manual) |
| | P6 | | | |
| | N7 | | | |
| | P8 | | | |
| | N9 | | | |
| | R9 | | | |
| | P10 | | | |
| | N11 | | | |
| | R11 | | | |
| | P12 | | | |
| | N13 | | | |
| | R13 | | | |
| | P14 | | | |
| | R15 | | | |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal

**Table 2-29. Supply Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | OTHER | DESCRIPTION |
|---|---|---|---|---|
| $CV_{DD}$ | F15 | S | | 1.3 V core supply voltage (-810 devices)<br>1.20 V core supply voltage (A-513, -594, -810 devices)<br>(see Section 6.3.1.2, *Power-Supply Decoupling*, of this data manual) |
| | K12 | | | |
| | M12 | | | |
| | L11 | | | |
| | M10 | | | |
| | L10 | | | |
| | K10 | | | |
| | L9 | | | |
| | L8 | | | |
| | M8 | | | |
| $CV_{DDDSP}$ | J13 | S | | 1.3 V core supply voltage (-810 devices)<br>1.20 V DSPSS supply voltage (A-513, -594, -810 devices)<br>(see Section 6.3.1.2, *Power-Supply Decoupling*, of this data manual) |
| | H12 | | | |
| | H11 | | | |
| | J11 | | | |
| | K11 | | | |
| | J10 | | | |
| | H10 | | | |
| | J9 | | | |
| | K9 | | | |
| | K8 | | | |
| | H8 | | | |

**Table 2-30. Ground Terminal Functions**

| SIGNAL NAME | NO. | TYPE[1] | OTHER | DESCRIPTION |
|---|---|---|---|---|
| **GROUND PINS** | | | | |
| $V_{SS}$ | K5 | GND | | Ground pins |
| | M5 | | | |
| | G6 | | | |
| | J6 | | | |
| | L6 | | | |
| | N6 | | | |
| | R6 | | | |
| | F7 | | | |
| | H7 | | | |
| | K7 | | | |
| | M7 | | | |
| | P7 | | | |
| | R7 | | | |
| | E8 | | | |
| | G8 | | | |
| | J8 | | | |
| | N8 | | | |
| | F9 | | | |
| | H9 | | | |
| | M9 | | | |
| | P9 | | | |
| | G10 | | | |
| | N10 | | | |
| | R10 | | | |
| | G11 | | | |
| | M11 | | | |
| | P11 | | | |
| | G12 | | | |
| | J12 | | | |
| | N12 | | | |
| | L12 | | | |
| | R12 | | | |
| | G13 | | | |
| | H13 | | | |
| | K13 | | | |
| | M13 | | | |
| | P13 | | | |
| | G14 | | | |
| | J14 | | | |

(1) I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground, A = Analog signal

**Table 2-30. Ground Terminal Functions (continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE[1] | OTHER | DESCRIPTION |
|---|---|---|---|---|
| $V_{SS}$ | L14 | GND | | Ground pins |
| | N14 | | | |
| | R14 | | | |
| | H15 | | | |
| | K15 | | | |
| | P15 | | | |

## 2.8 Device Support

### 2.8.1 Development Support

TI offers an extensive line of development tools for the TMS320DM644x SoC 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 TMS320DM644x SoC-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 SoC application.

**Hardware Development Tools:**

Extended Development System (XDS™) Emulator

For a complete listing of development-support tools for the TMS320DM644x SoC platform, visit the Texas Instruments web site on the Worldwide Web at http://www.ti.com uniform resource locator (URL). For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

### 2.8.2 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all DSP devices and support tools. Each DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (e.g., **TMX320DM6446ZWT**). 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 against the following disclaimer:

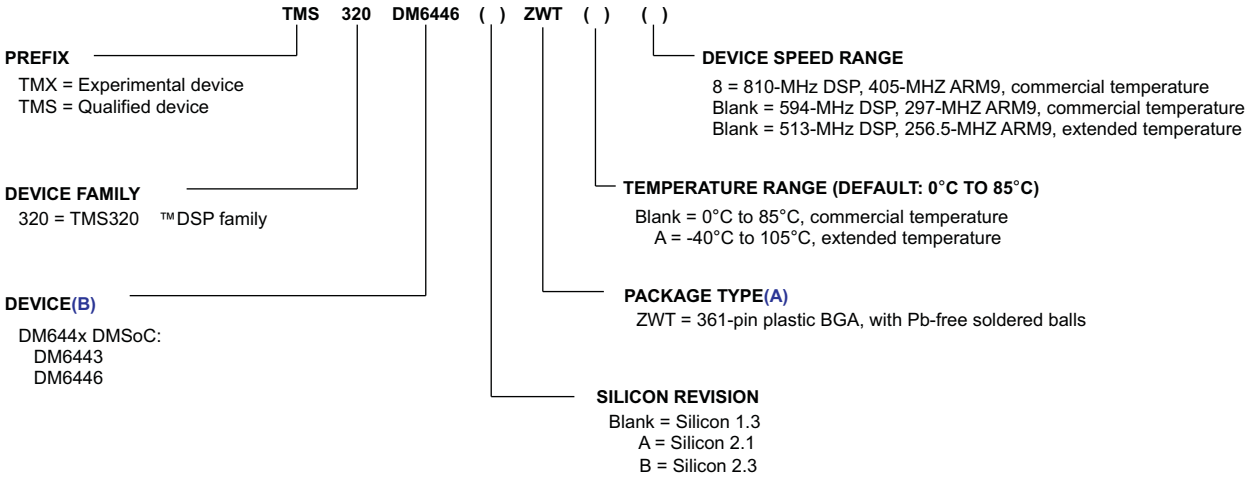
"Developmental product is intended for internal evaluation purposes."

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

Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, ZWT), the temperature range (for example, "Blank" is the commercial temperature range), and the device speed range in megahertz (for example, "Blank" is the default [594-MHz DSP, 297-MHz ARM9]).

Figure 2-6 provides a legend for reading the complete device name for any TMS320DM644x SoC platform member.




A. BGA = Ball Grid Array

B. For actual device part numbers (P/Ns) and ordering information, see the TI website (http://www.ti.com).

**Figure 2-6. Device Nomenclature**

## 2.8.3 Documentation Support

### 2.8.3.1 Related Documentation From Texas Instruments

The following documents describe the TMS320DM644x Digital Media System-on-Chip (DMSoC). 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.

The current documentation that describes the DM644x DMSoC, related peripherals, and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

**SPRU395** ***TMS320C64x Technical Overview.*** Provides an introduction to the TMS320C64x digital signal processors (DSPs) of the TMS320C6000 DSP family.

**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.

**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.

**SPRUE14** ***TMS320DM644x DMSoC ARM Subsystem Reference Guide.*** Describes the ARM subsystem in the TMS320DM644x Digital Media System-on-Chip (DMSoC). The ARM subsystem is designed to give the ARM926EJ-S (ARM9) master control of the device. In general, the ARM is responsible for configuration and control of the device; including the DSP subsystem, the video processing subsystem, and a majority of the peripherals and external memories.

**SPRUE15** ***TMS320DM644x DMSoC DSP Subsystem Reference Guide.*** Describes the digital signal processor (DSP) subsystem in the TMS320DM644x Digital Media System-on-Chip (DMSoC).

**SPRUE19** ***TMS320DM644x DMSoC Peripherals Overview Reference Guide.*** Provides an overview and briefly describes the peripherals available on the TMS320DM644x Digital Media System-on-Chip (DMSoC).

**SPRUE20** ***TMS320DM644x DMSoC Asynchronous External Memory Interface (EMIF) Reference Guide.*** Describes the asynchronous external memory interface (EMIF) in the TMS320DM644x Digital Media System-on-Chip (DMSoC). The EMIF supports a glueless interface to a variety of external devices.

**SPRUE21** ***TMS320DM644x DMSoC ATA Controller User's Guide.*** Describes the ATA controller in the TMS320DM644x Digital Media System-on-Chip (DMSoC). The ATA controller provides a glueless interface to storage media to be used by video and audio applications for video and audio data storage.

**SPRUE22** ***TMS320DM644x DMSoC DDR2 Memory Controller User's Guide.*** Describes the DDR2 memory controller in the TMS320DM644x Digital Media System-on-Chip (DMSoC). The DDR2 memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM devices.

**SPRUE23** ***TMS320DM644x DMSoC Enhanced Direct Memory Access (EDMA3) Controller User's Guide.*** Describes the operation of the enhanced direct memory access (EDMA3) controller in the TMS320DM644x Digital Media System-on-Chip (DMSoC). The EDMA3 controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the DMSoC.

**SPRUE24** ***TMS320DM644x DMSoC Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) Module User's Guide.*** Discusses the ethernet media access controller (EMAC) and physical layer (PHY) device management data input/output (MDIO) module in the TMS320DM644x Digital Media System-on-Chip (DMSoC). The EMAC controls the flow of packet data from the DMSoC to the PHY. The MDIO module controls PHY configuration and status monitoring.

**SPRUE25** ***TMS320DM644x DMSoC General-Purpose Input/Output (GPIO) User's Guide.*** Describes the general-purpose input/output (GPIO) peripheral in the TMS320DM644x Digital Media System-on-Chip (DMSoC). 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.

**SPRUE26** ***TMS320DM644x DMSoC 64-Bit Timer User's Guide.*** Describes the operation of the software-programmable 64-bit timer in the TMS320DM644x Digital Media System-on-Chip (DMSoC). Timer 0 and Timer 1 are used as general-purpose (GP) timers and can be programmed in 64-bit mode, dual 32-bit unchained mode, or dual 32-bit chained mode; Timer 2 is used only as a watchdog timer. The GP timer modes can be used to generate periodic interrupts or enhanced direct memory access (EDMA) synchronization events. The watchdog timer mode is used to provide a recovery mechanism for the device in the event of a fault condition, such as a non-exiting code loop.

**SPRUE29** ***TMS320DM644x DMSoC Audio Serial Port (ASP) User's Guide.*** Describes the operation of the audio serial port (ASP) audio interface in the TMS320DM644x Digital Media System-on-Chip (DMSoC). The primary audio modes that are supported by the ASP are the AC97 and IIS modes. In addition to the primary audio modes, the ASP supports general serial port receive and transmit operation, but is not intended to be used as a high-speed interface.

**SPRUE35** ***TMS320DM644x DMSoC Universal Serial Bus (USB) Controller User's Guide.*** Describes the universal serial bus (USB) controller in the TMS320DM644x Digital Media System-on-Chip (DMSoC). The USB controller supports data throughput rates up to 480 Mbps. It provides a mechanism for data transfer between USB devices and also supports host negotiation.

**SPRUE37** ***TMS320DM644x DMSoC Video Processing Back End (VPBE) User's Guide.*** Describes the video processing back end (VPBE) in the TMS320DM644x Digital Media System-on-Chip (DMSoC) video processing subsystem. Included in the VPBE is the video encoder, on-screen display, and digital LCD controller.

**SPRUE97** ***TMS320DM644x DMSoC Host Port Interface (HPI) User's Guide.*** Describes the features and operation of the host port interface (HPI) in the TMS320DM644x Digital Media System-on-Chip (DMSoC).

**SPRA839** ***Using IBIS Models for Timing Analysis.*** Describes how to properly use IBIS models to attain accurate timing analysis for a given system.

**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.

**SPRAAA6** ***EDMA v3.0 (EDMA3) Migration Guide for TMS320DM644x DMSoC.*** Describes migrating from the Texas Instruments TMS320C64x digital signal processor (DSP) enhanced direct memory access (EDMA2) to the TMS320DM644x Digital Media System-on-Chip (DMSoC) EDMA3. This document summarizes the key differences between the EDMA3 and the EDMA2 and provides guidance for migrating from EDMA2 to EDMA3.

**SPRAAC5** ***Implementing DDR2 PCB Layout on the TMS320DM644x DSP.*** Contains implementation instructions for the DDR2 interface contained on the TMS320DM644x digital signal processor (DSP) device.

**SPRAAD6** ***TMS320DM6446/3 Power Consumption Summary.*** This document discusses the power consumption of the Texas Instruments TMS320DM6446 and TMS320DM6443 digital media System-on-Chip (DMSoC).

# 3 Device Configurations

## 3.1 System Module Registers

The system module includes status and control registers required for configuration of the device. Brief descriptions of the various registers are shown in Table 3-1. System Module registers required for device configurations are discussed in the following sections.

**Table 3-1. System Module Register Memory Map**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C4 0000 | PINMUX0 | Pin multiplexing control 0. For details, see Section 3.5.4, *PINMUX0 Register Description*. |
| 0x01C4 0004 | PINMUX1 | Pin multiplexing control 1. For details, see Section 3.5.5, *PINMUX1 Register Description*. |
| 0x01C4 0008 | DSPBOOTADDR | Boot address of DSP. For details, see Section 3.3.1.2, *DSPBOOTADDR Register Description*. |
| 0x01C4 000C | SUSPSRC | Emulator Suspend Source. For details, see Section 3.6, *Emulation Control*. |
| 0x01C4 0010 | INTGEN | ARM/DSP Interrupt Status and Control. For details, see Section 6.7.3, *ARM/DSP Communications Interrupts*. |
| 0x01C4 0014 | BOOTCFG | Device boot configuration. For details, see Section 3.3.1.1, *BOOTCFG Register Description*. |
| 0x01C4 0018 - 0x01C4 0027 | – | Reserved. |
| 0x01C4 0028 | JTAGID | JTAGID/Device ID number. For details, see Section 6.25.1, *JTAG Peripheral Register Description(s) – JTAG ID Register*. |
| 0x01C4 002C | – | Reserved. |
| 0x01C4 0030 | HPI_CTL | HPI control. For details, see Section 3.5.6.10, *HPI and EMIFA/ATA Pin Multiplexing*. |
| 0x01C4 0034 | USBPHY_CTL | USB PHY control. For details, see Section 6.15.1, *USBPHY_CTL Register Description*. |
| 0x01C4 0038 | CHP_SHRTSW | Chip shorting switch control. For details, see Section 3.2.1, *Power Configurations at Reset*. |
| 0x01C4 003C | MSTPRI0 | Bus master priority control 0. For details, see Section 3.5.1, *Switched Central Resource (SCR) Bus Priorities*. |
| 0x01C4 0040 | MSTPRI1 | Bus master priority control 1. For details, see Section 3.5.1, *Switched Central Resource (SCR) Bus Priorities*. |
| 0x01C4 0044 | VPSS_CLKCTL | VPSS clock control. |
| 0x01C4 0048 | VDD3P3V_PWDN | VDD 3.3V I/O powerdown control. For details, see Section 3.2.2, *Power Configurations after Reset*. |
| 0x01C4 004C | DRRVTPER | Enables access to the DDR2 VTP Register. |
| 0x01C4 0050 - 0x01C4 006F | – | Reserved. |

## 3.2 Power Considerations

Global device power domains are controlled by the Power and Sleep Controller, except as shown in the following sections.

### 3.2.1 Power Configurations at Reset

As described in Section 6.3.1.3, *DM6446 Power and Clock Domains*, the DM6446 has two power domains: Always On and DSP. There is a shorting switch between the two power domains that must be opened when the DSP domain is powered off and closed when the DSP domain is powered on.

The CHP_SHRTSW register, shown in Figure 3-1, controls the shorting switch between the device always-on and DSP power domains. This switch should be enabled after powering-up the DSP domain. Setting the DSPPWRON bit to '1' closes (enables) the switch and enables the DSP power domain. The default switch value is determined by the DSP_BT configuration input. If DSP self boot is selected (DSP_BT=1), the DSP will be powered-up and DSPPWRON will be set to a value of '1'. For ARM boot operation (DSP_BT=0), DSPPWRON will be set to the disable value of '0' and must be set by the ARM before the DSP domain power is turned on.

**Note:** Once the DSP power domain is enabled (powered up), it *cannot* be disabled (powered down). Dynamic power down of the DSP is ***not*** supported on this device.

**Figure 3-1. CHP_SHRTSW Register**

| 31 | 1 | 0 |
|---|---|---|
| RESERVED | | DSPPWRON |
| R-0000 0000 0000 0000 0000 0000 0000 000 | | R/W-L |

LEGEND: R = Read, W = Write, n = value at reset, L = pin state latched at reset rising

**Table 3-2. CHP_SHRTSW Register Description**

| NAME | DESCRIPTION |
|---|---|
| DSPPWRON | DSP power domain enable.<br>0 = Shorting switch open<br>1 = Shorting switch closed |

### 3.2.2 Power Configurations after Reset

The VDD3P3V_PWDN register controls power to the 3.3V I/O buffers for MMC/SD/SDIO and GPIOV33. The 3.3V I/Os are separated into two groups for independent control as shown in Figure 3-2 and described in Table 3-3. By default, these pins are all disabled at reset.

**Figure 3-2. VDD3P3V_PWDN Register**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| RESERVED | | IOPWDN1 | IOPWDN0 |
| R-0000 0000 0000 0000 0000 0000 0000 00 | | R/W-1 | R/W-1 |

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

**Table 3-3. VDD3P3V_PWDN Register Description**

| NAME | DESCRIPTION |
|---|---|
| IOPWDN0 | GIOV33 I/O Powerdown controls GIOV33[16:0] pins.<br>0 = I/O buffers powered up<br>1 = I/O buffers powered down |
| IOPWDN1 | MMC/SD/SDIO I/O Powerdown controls SD_CLK, SD_CMD, SD_DATA[3:0] pins.<br>0 = I/O buffers powered up<br>1 = I/O buffers powered down |

## 3.3 Bootmode

The device is booted through multiple means: pin states captured at reset, primary bootloaders within internal ROM or EMIFA, and secondary user bootloaders from peripherals or external memories. Boot modes, pin configurations, and register configurations required for booting the device, are described in the following sections.

### *3.3.1 Bootmode Registers*

The BOOTCFG and DSPBOOTADDR registers are described in the following sections. At reset, the status of various pins required for proper boot are stored within these registers.

#### 3.3.1.1 BOOTCFG Register Description

The BOOTCFG register (located at address 0x01C4 0014) contains the status values of the BTSEL1, BTSEL0, DSP_BT, EM_WIDTH, and AEAW[4:0] pins captured at the rising edge of $\overline{\text{RESET}}$. The register format is shown in Figure 3-3 and bit field descriptions are shown in Table 3-4. The captured bits are software readable after reset.

**Figure 3-3. BOOTCFG Register**

| 31 | 9 | 8 | 7 | 6 | 5 | 4 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | DSP_BT | BTSEL | | EM_WIDTH | DAEAW | |
| R-0000 0000 0000 0000 0000 000 | | R-L | R-LL | | R-L | R-LLLLL | |

LEGEND: R = Read; W = Write; L = pin state latched at reset rising; -*n* = value after reset

**Table 3-4. BOOTCFG Register Description**

| NAME | DESCRIPTION |
|---|---|
| BTSEL | ARM Boot mode selection pin states (BTSEL1, BTSEL0) captured at the rising edge of $\overline{\text{RESET}}$.<br>'00' indicates ARM boots from ROM (NAND Flash/SPI Flash).<br>'01' indicates that ARM boots from EMIFA (NOR Flash).<br>'10' indicates that ARM boots from ROM (HPI).<br>'11' indicates that ARM boots from ROM (UART0). |
| DSP_BT | DSP Boot mode selection pin state captured at the rising edge of $\overline{\text{RESET}}$.<br>'0' sets ARM boot of C64x+.<br>'1' sets C64x+ self boot. |
| EM_WIDTH | EMIFA data bus width selection pin state captured at the rising edge of $\overline{\text{RESET}}$.<br>'0' sets EMIFA to 8 bit data bus width<br>'1' sets EMIFA to 16 bit data bus width. |
| DAEAW | EMIFA address bus width selection pin states (AEAW[4:0]) captured at the rising edge of $\overline{\text{RESET}}$. This configures EMIFA address pins multiplexed with GPIO. See the *GPIO and EMIFA Multiplexing* tables (Table 3-9,Table 3-10, and Table 3-11). |

#### 3.3.1.2 DSPBOOTADDR Register Description

The DSPBOOTADDR register contains the upper 22 bits of the C64x+ DSP reset vector. The register format is shown in Figure 3-4 and bit field descriptions are shown in Table 3-5. DSPBOOTADDR is readable and writable by software after reset.

**Figure 3-4. DSPBOOTADDR Register**

| 31 | 10 | 9 | 0 |
|---|---|---|---|
| BOOTADDR[21:0] | | RESERVED | |
| R- 0100 0010 0010 0000 0000 00 | | R-00 0000 0000 | |

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

**Table 3-5. DSPBOOTADDR Register Description**

| NAME | DESCRIPTION |
|---|---|
| BOOTADDR[21:0] | Upper 22 bits of the C64x+ DSP boot address. |

### *3.3.2 ARM Boot*

The DM6446 ARM can boot from EMIFA, internal ROM (NAND, SPI), UART0, or HPI, as determined by the setting of the BTSEL[1:0] pins. The BTSEL[1:0] pins are read by the ARM ROM Boot Loader (RBL) to further define the ROM boot mode. The ARM boot modes are summarized in Table 3-6.

**Table 3-6. ARM Boot Modes**

| BTSEL1 | BTSEL0 | BOOT MODE | ARM RESET VECTOR | BRIEF DESCRIPTION |
|---|---|---|---|---|
| 0 | 0 | ARM NAND, SPI RBL | 0x0000 4000 | Up to 14 K-bytes secondary boot loader through NAND with up to 2 K-bytes page sizes. |
| 0 | 1 | ARM EMIFA External Boot | 0x0200 0000 | EMIFA EM_CS2 external memory space. |
| 1 | 0 | ARM HPI RBL | 0x0000 4000 | Up to 14 K-btyes secondary boot loader through an external host. |
| 1 | 1 | ARM UART RBL | 0x0000 4000 | Up to 14 K-bytes secondary boot loader through UART0. |

When the BTSEL[1:0] pins are set to the ARM EMIFA External Boot ("01"), the ARM immediately begins executing code from the EMIFA EM_CS2 memory space (0x0200 0000). When the BTSEL[1:0] pins indicate a condition other than the ARM EMIFA External Boot (!01), the RBL begins execution.

ARM NAND/SPI Boot mode has the following features:

- Loads a secondary User Boot Loader (UBL) from NAND/SPI flash to ARM Internal RAM (AIM) and transfers control to the user software.
- Support for NAND with page sizes up to 8192 bytes.
- Support for error correction when loading UBL
- Support for up to 14KB UBL
- Optional, user selectable, support for use of DMA, I-cache, and PLL enable while loading UBL

ARM UART Boot mode has the following features:

- Loads a secondary UBL via UART0 to AIM and transfers control to the user software.
- Support for up to 14KB UBL

ARM HPI Boot Mode has the following features:

- No support for a full firmware boot. Instead, waits for external host to load a secondary UBL via HPI to AIM and transfers control to the user software.
- Support for up to 14KB UBL.

For further details on the ROM Bootloader, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

### 3.3.3 DSP Boot

For C64x+ booting, the state of the DSP_BT pin is sampled at reset. If DSP_BT is low, the ARM will be the master of C64x+ and control booting (Host Boot mode). If DSP_BT is high, the C64x+ will boot itself coming out of device reset (Self-Boot mode). Table 3-7 shows a summary of the DSP boot modes.

**Table 3-7. DSP Boot Modes**

| DSP_BT | DSP BOOT MODE | ARM BOOT MODE | DSPBOOTADDR REGISTER VALUE | BRIEF DESCRIPTION |
|---|---|---|---|---|
| 0 | Host Boot | Internal Boot | Programmable | ARM sets an internal DSP memory location in DSPBOOTADDR register where valid DSP code resides and loads code to this internal DSP memory through DMA prior to releasing DSP reset. |
| 0 | Host Boot | External Boot | Programmable | ARM sets an external DSP memory location in DSPBOOTADDR register (EMIFA or DDR2) where valid DSP code resides prior to releasing DSP reset. |
| 1 | Self Boot | Any, except HPI | 0x4220 0000 | Default EMIFA Base Address |
| 1 | Host Boot | HPI | Programmable | ARM sets a DSP memory location in the DSPBOOTADDR register. HPI loads code into the DM6446 memory map with the entry point set to the memory location specified in the DSPBOOTADDR register. Once the HPI completes loading the code, the ARM should release the DSP from reset. |

#### 3.3.3.1 Host-Boot Mode

In host boot mode, the ARM is the master and controls the reset and boot of the C64x+. The C64x+ DSP remains powered-off after device reset. The ARM is responsible for enabling power to the C64x+ and releasing it from reset (PSC MMR bits: MDCTL[39].LRST and MRSTOUT1.MRSTz[39]). Prior to releasing the C64x+ reset, the ARM must program the address from which the C64x+ will begin execution in the DSPBOOTADDR register.

#### 3.3.3.2 Self-Boot Mode

In self-boot mode, the C64x+ power domain is turned on and the C64x+ DSP is released from reset without ARM intervention. The C64x+ begins execution from the default EMIFA address (0x4220 0000) contained within the DSPBOOTADDR register. The C64x+ begins execution with instruction (L1P) cache enabled.

## 3.4 Configurations at Reset

The following sections give information on configuration settings for the device at reset.

### 3.4.1 Device Configuration at Device Reset

Table 3-8 shows a summary of device inputs required for booting the ARM and DSP, and configuring EMIFA data and address bus widths for proper operation of the device at the rising edge of the RESET input.

**Table 3-8. Device Configurations (Input Pins Sampled at Reset)**

| DEVICE SIGNALS SAMPLED AT RESET | DEVICE SIGNAL NAME AFTER RESET | DESCRIPTION |
|---|---|---|
| BTSEL[1:0] | COUT[1:0] | ARM Boot mode selection pins.<br>'00' indicates ARM boots from ROM (NAND/SPI Flash).<br>'01' indicates that ARM boots from EMIFA (NOR Flash).<br>'10' indicates that the ARM boots from the HPI (ROM)<br>'11' indicates that ARM boots from ROM (UART0). |
| DSP_BT | COUT3 | DSP Boot mode selection pin.<br>'0' sets ARM boot of C64x+.<br>'1' sets C64x+ self boot. |
| EM_WIDTH | COUT2 | EMIFA data bus width selection pin.<br>'0' sets EMIFA to 8-bit data bus width<br>'1' sets EMIFA to 16-bit data bus width. |
| AEAW[4:0] | YOUT[4:0] | EMIFA address bus width selection pins for EMIFA address pins multiplexed with GPIO. See the *GPIO and EMIFA Multiplexing* tables (Table 3-9, Table 3-10, and Table 3-11) for details. |

### 3.4.2 Peripheral Selection at Device Reset

As briefly mentioned in Table 3-8, the state of the AEAW[4:0] pins captured at reset configures the number of EMIFA address pins required for device boot. These values are stored in the AEAW field of the PINMUX0 register. At reset, this provides proper addressing for external boot. Unused address pins are available for use as GPIO. The register settings are software programmable after reset. Table 3-9, Table 3-10, and Table 3-11 show the AEAW[4:0] bit settings and the corresponding multiplexing for EMIFA address and GPIO pins.

The number of EMIFA address bits enabled is configurable from 0 to 23. EM_BA[1] and EM_A[21:0] pins that are not assigned to another peripheral and not enabled as address signals become GPIO pins. The enabled address pins are always contiguous from EM_BA[1] upwards and address bits cannot be skipped. The exception to this are the EM_A[2:1] pins. EM_A[2:1] are usable as the ALE and CLE signals for the NAND Flash mode of EMIFA and are always enabled as EMIFA pins. If an address width of 0 is selected, this still allows a NAND Flash to be accessed. Also, selecting an address width of 2, 3, or 4 (AEAW[4:0] = 00010, 00011, or 00100) always results in 4 address outputs. For these and other address bit enable settings, see the *GPIO and EMIFA Multiplexing* tables (Table 3-9, Table 3-10, and Table 3-11).

**Table 3-9. GPIO and EMIFA Multiplexing (Part 1)**

| Pin Mux Register AEAW[4:0] Bit Settings | | | | | | | |
|---|---|---|---|---|---|---|---|
| **00000 (default)** | **00001** | **00010** | **00011** | **00100** | **00101** | **00110** | **00111** |
| GPIO[52] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] |
| GPIO[53] | GPIO[53] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] |
| EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] |
| EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] |
| GPIO[28] | GPIO[28] | GPIO[28] | GPIO[28] | GPIO[28] | EM_A[3] | EM_A[3] | EM_A[3] |
| GPIO[27] | GPIO[27] | GPIO[27] | GPIO[27] | GPIO[27] | GPIO[27] | EM_A[4] | EM_A[4] |
| GPIO[26] | GPIO[26] | GPIO[26] | GPIO[26] | GPIO[26] | GPIO[26] | GPIO[26] | EM_A[5] |
| GPIO[25] | GPIO[25] | GPIO[25] | GPIO[25] | GPIO[25] | GPIO[25] | GPIO[25] | GPIO[25] |
| GPIO[24] | GPIO[24] | GPIO[24] | GPIO[24] | GPIO[24] | GPIO[24] | GPIO[24] | GPIO[24] |
| GPIO[23] | GPIO[23] | GPIO[23] | GPIO[23] | GPIO[23] | GPIO[23] | GPIO[23] | GPIO[23] |
| GPIO[22] | GPIO[22] | GPIO[22] | GPIO[22] | GPIO[22] | GPIO[22] | GPIO[22] | GPIO[22] |
| GPIO[21] | GPIO[21] | GPIO[21] | GPIO[21] | GPIO[21] | GPIO[21] | GPIO[21] | GPIO[21] |
| GPIO[20] | GPIO[20] | GPIO[20] | GPIO[20] | GPIO[20] | GPIO[20] | GPIO[20] | GPIO[20] |
| GPIO[19] | GPIO[19] | GPIO[19] | GPIO[19] | GPIO[19] | GPIO[19] | GPIO[19] | GPIO[19] |
| GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] |
| GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] |
| GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] |
| GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] |
| GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] |
| GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] |
| GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] |
| GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] |
| GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] |

**Table 3-10. GPIO and EMIFA Multiplexing (Part 2)**

| Pin Mux Register AEAW[4:0] Bit Settings | | | | | | | |
|---|---|---|---|---|---|---|---|
| **01000** | **01001** | **01010** | **01011** | **01100** | **01101** | **01110** | **01111** |
| EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] |
| EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] |
| EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] |
| EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] |
| EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] |
| EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] |
| EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] |
| EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] |
| GPIO[24] | EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] |
| GPIO[23] | GPIO[23] | EM_A[8] | EM_A[8] | EM_A[8] | EM_A[8] | EM_A[8] | EM_A[8] |
| GPIO[22] | GPIO[22] | GPIO[22] | EM_A[9] | EM_A[9] | EM_A[9] | EM_A[9] | EM_A[9] |
| GPIO[21] | GPIO[21] | GPIO[21] | GPIO[21] | EM_A[10] | EM_A[10] | EM_A[10] | EM_A[10] |
| GPIO[20] | GPIO[20] | GPIO[20] | GPIO[20] | GPIO[20] | EM_A[11] | EM_A[11] | EM_A[11] |
| GPIO[19] | GPIO[19] | GPIO[19] | GPIO[19] | GPIO[19] | GPIO[19] | EM_A[12] | EM_A[12] |
| GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] | GPIO[18] | EM_A[13] |
| GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] | GPIO[17] |
| GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] | GPIO[16] |
| GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] | GPIO[15] |
| GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] | GPIO[14] |
| GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] |
| GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] |
| GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] |
| GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] |

**Table 3-11. GPIO and EMIFA Multiplexing (Part 3)**

| Pin Mux Register AEAW[4:0] Bit Settings | | | | | | | |
|---|---|---|---|---|---|---|---|
| **10000** | **10001** | **10010** | **10011** | **10100** | **10101** | **10110** | **Others** |
| EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] | EM_BA[1] |
| EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] | EM_A[0] |
| EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] | EM_A[1] |
| EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] | EM_A[2] |
| EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] | EM_A[3] |
| EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] | EM_A[4] |
| EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] | EM_A[5] |
| EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] | EM_A[6] |
| EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] | EM_A[7] |
| EM_A[8] | EM_A[8] | EM_A[8] | EM_A[8] | EM_A[8] | EM_A[8] | EM_A[8] | EM_A[8] |
| EM_A[9] | EM_A[9] | EM_A[9] | EM_A[9] | EM_A[9] | EM_A[9] | EM_A[9] | EM_A[9] |
| EM_A[10] | EM_A[10] | EM_A[10] | EM_A[10] | EM_A[10] | EM_A[10] | EM_A[10] | EM_A[10] |
| EM_A[11] | EM_A[11] | EM_A[11] | EM_A[11] | EM_A[11] | EM_A[11] | EM_A[11] | EM_A[11] |
| EM_A[12] | EM_A[12] | EM_A[12] | EM_A[12] | EM_A[12] | EM_A[12] | EM_A[12] | EM_A[12] |
| EM_A[13] | EM_A[13] | EM_A[13] | EM_A[13] | EM_A[13] | EM_A[13] | EM_A[13] | EM_A[13] |
| EM_A[14] | EM_A[14] | EM_A[14] | EM_A[14] | EM_A[14] | EM_A[14] | EM_A[14] | EM_A[14] |
| GPIO[16] | EM_A[15] | EM_A[15] | EM_A[15] | EM_A[15] | EM_A[15] | EM_A[15] | EM_A[15] |
| GPIO[15] | GPIO[15] | EM_A[16] | EM_A[16] | EM_A[16] | EM_A[16] | EM_A[16] | EM_A[16] |
| GPIO[14] | GPIO[14] | GPIO[14] | EM_A[17] | EM_A[17] | EM_A[17] | EM_A[17] | EM_A[17] |
| GPIO[13] | GPIO[13] | GPIO[13] | GPIO[13] | EM_A[18] | EM_A[18] | EM_A[18] | EM_A[18] |
| GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | GPIO[12] | EM_A[19] | EM_A[19] | EM_A[19] |
| GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | GPIO[11] | EM_A[20] | EM_A[20] |
| GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | GPIO[10] | EM_A[21] |

## 3.5 Configurations After Reset

The following sections give the details on configuring the device after reset.

### 3.5.1 Switched Central Resource (SCR) Bus Priorities

Prioritization within the switched central resource (SCR) is programmable for each master. The register bit fields and default priority levels for DM6446 bus masters are shown in Table 3-12. The priority levels should be tuned to obtain the best system performance for a particular application. Lower values indicate higher priority. For most masters, their priority values are programmed at the system level by configuring the MSTPRI0 and MSTPRI1 registers. Details on the MSTPRI0/1 registers are shown in Figure 3-5 and Figure 3-6. The C64x+, VPSS, and EDMA3 masters contain registers that control their own priority values.

**Table 3-12. DM6446 Default Bus Master Priorities**

| PRIORITY BIT FIELD | BUS MASTER | DEFAULT PRIORITY LEVEL |
|---|---|---|
| VPSSP | VPSS | 0 (VPSS PCR Register, DMA_PRI bit field)<br>[For more detailed information on the DMA_PRI bit field, see the *TMS320DM644x DMSoC Video Processing Back End (VPBE) User's Guide* (literature number SPRUE37).] |
| EDMATC0P | EDMATC0 | 0 (EDMA3CC QUEPRI Register)<br>[For more detailed information on the QUEPRI register, see the *TMS320DM644x DMSoC Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (literature number SPRUE23).] |
| EDMATC1P | EDMATC1 | 0 (EDMA3CC QUEPRI Register)<br>[For more detailed information on the QUEPRI register, see the *TMS320DM644x DMSoC Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (literature number SPRUE23).] |
| ARM_DMAP | ARM (DMA) | 1 (MSTPRI0 Register) |
| ARM_CFGP | ARM (CFG) | 1 (MSTPRI0 Register) |
| C64X+_DMAP | C64X+ (DMA) | 7 (C64x+ MDMAARBE.PRI Register bit field)<br>[For more detailed information on the PRI bit field, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).] |
| C64X+_CFGP | C64X+ (CFG) | 1 (MSTPRI0 Register) |
| EMACP | EMAC | 4 (MSTPRI1 Register) |
| USBP | USB | 4 (MSTPRI1 Register) |
| ATAP | ATA/CF | 4 (MSTPRI1 Register) |
| VLYNQP | VLYNQ | 4 (MSTPRI1 Register) |
| HPIP | HPI | 4 (MSTPRI1 Register) |
| VICPP | VICP | 5 (MSTPRI0 Register) |

**Figure 3-5. MSTPRI0 Register**

| 31 | 19 | 18 | 16 |
|---|---|---|---|
| RESERVED | | VICPP(1) | |
| R-0000 0000 0000 0 | | R/W-101 | |

| 15 | 11 | 10 | 8 | 7 | 6 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | C64X+_CFGP | | RSV | ARM_CFGP | | RSV | ARM_DMAP | |
| R-0000 0 | | R/W-001 | | R-0 | R/W-001 | | R-0 | R/W-001 | |

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

(1) The VICPP bit field is configured by the third-party software. When modifying the MSTPRI0 register a read/modify/write must be performed to preserve the configuration set by the third-party software.

**Figure 3-6. MSTPRI1 Register**

| 31 | 23 | 22 | 20 | 19 | 18 | 16 |
|---|---|---|---|---|---|---|
| RESERVED | | HPIP | | RSV | VLYNQP | |
| R-0000 0000 0 | | R/W-100 | | R-0 | R/W-100 | |

| 15 | 14 | 12 | 11 | 10 | 8 | 7 | 6 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RSV | ATAP | | RSV | USBP | | RSV | RESERVED | | RSV | EMACP | |
| R-0 | R/W-100 | | R-0 | R/W-100 | | R-0 | R-100 | | R-0 | R/W-100 | |

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

### 3.5.2 Multiplexed Pin Configurations

There are numerous multiplexed pins that are shared by more than one peripheral. Some of these pins are configured by external pullup/pulldown resistors only at reset, and others are configured by software. As described in detail in Section 3.4.1 (*Device Configuration at Device Reset*) and Section 3.4.2 (*Peripheral Selection at Device Reset*), hardware configurable multiplexed pins are programmed by external pullup/pulldown resistors at reset to set the initial functionality of pins for use by a single peripheral. After reset, software configurable multiplexed pins are programmable through Memory Mapped Registers (MMR) to allow the switching of pin functionalities during run-time. See Section 3.5.3, *Peripheral Selection After Device Reset*, for more details on the register settings.

A summary of the pin multiplexing is shown in Table 3-13. The EMAC peripheral shares pins with the 3.3V GPIO pins. The VLYNQ pins overlap upper EMIFA address pins resulting in a reduced EMIFA address range as the VLYNQ width is increased. The ATA peripheral shares data lines and some control signals with EMIFA. The ATA DMA pins are multiplexed with UART1. The ASP, UART0/1/2, SPI, I2C, and PWM0/1/2 all default to GPIO pins when not enabled. The VPBE function of the VPSS requires additional pins to implement the RGB888 mode, these are multiplexed with GPIOs.

**Table 3-13. DM6446 Multiplexed Peripheral Pins and Multiplexing Controls**

| MULTIPLEXED PERIPHERALS | PRIMARY (DEFAULT) FUNCTION | SECONDARY(1) FUNCTION | TERTIARY(2) FUNCTION | SECONDARY REGISTER/PIN(3) CONTROL | TERTIARY REGISTER/PIN(3) CONTROL |
|---|---|---|---|---|---|
| EMIFA (NAND), HPI | **EMIFA (NAND):** EM_A[1] (ALE), EM_A[2] (CLE), EM_CS2, EM_CS3 | **HPI:** HHWIL, HCNTL0, HCS | | **PinMux0:**HPIEN, **Pins:**BTSEL[1:0] = 10 | |
| EMIFA, HPI, ATA (CF) | **EMIFA:** EM_D[0:15], EM_BA[0] | **ATA (CF):** DD[0:15], DA0 | **HPI:** HD[0:15], HINT | **PinMux0:**ATAEN | **PinMux0:**HPIEN, **Pins:**BTSEL[1:0] = 10 |
| EMIFA (NAND), HPI, ATA (CF) | **EMIFA (NAND):** R/W, EM_WAIT (RDY/BSY), EM_OE (RE), EM_WE (WE) | **ATA (CF):** INTRQ, IORDY, DIOR(IORD) , DIOW (IOWR) | **HPI:** HR/W, HRDY, HDS1, HDS2 | **PinMux0:**ATAEN | **PinMux0:**HPIEN, **Pins:**BTSEL[1:0] = 10 |
| VPBE LCD, GPIO | **GPIO:**GPIO[0] | **VPBE:** LCD_OE | | **PinMux0:**LOEEN | |
| VPFE CCD, GPIO | **GPIO:**GPIO[1] | **VPFE:** C_WE | | **PinMux0:**CWE | |
| VPBE RGB888, GPIO | **GPIO:**GPIO[2] | **VPBE:** RGB888 G0 | | **PinMux0:**RGB888 | |
| VPBE LCD/RGB888, GPIO | **GPIO:**GPIO[3] | **VPBE:** RGB888 B0 | **VPBE:** LCD_FIELD | **PinMux0:**RGB888 | **PinMux0:**LFLDEN |
| VPFE CCD, VPBE RGB888, GPIO | **GPIO:**GPIO[4] | **VPBE:** RGB888 R0 | **VPFE:** CCD_FIELD | **PinMux0:**RGB888 | **PinMux0:**CFLDEN |

(1) When the Secondary function is enabled, to avoid potential contention, ensure that the Primary (if not GPIO) and Tertiary functions are disabled.
(2) When the Tertiary function is enabled, to avoid potential contention, ensure that the Primary (if not GPIO), Secondary, and other Tertiary functions are disabled.
(3) Pin states are sampled at power on reset and written into the register fields.

**Table 3-13. DM6446 Multiplexed Peripheral Pins and Multiplexing Controls (continued)**

| MULTIPLEXED PERIPHERALS | PRIMARY (DEFAULT) FUNCTION | SECONDARY[1] FUNCTION | TERTIARY[2] FUNCTION | SECONDARY REGISTER/PIN[3] CONTROL | TERTIARY REGISTER/PIN[3] CONTROL |
|---|---|---|---|---|---|
| VPBE RGB888, GPIO | **GPIO:** GPIO[5:6, 38] | **VPBE:** RGB888 G1, B1, R1 | | **PinMux0:**RGB888 | |
| EMIFA, VLYNQ, GPIO | **GPIO:**GPIO[8] | **EMIFA:** $\overline{\text{EM\_CS5}}$ | **VLYNQ:** VLYNQ_CLOCK | **PinMux0:**AECS5 | **PinMux0:**VLYNQEN |
| EMIFA, VLYNQ, GPIO | **GPIO:**GPIO[9] | **EMIFA:** $\overline{\text{EM\_CS4}}$ | **VLYNQ:** VLYNQ_SCRUN | **PinMux0:**AECS4 | **PinMux0:**VLSCREN |
| EMIFA, VLYNQ, GPIO | **GPIO:** GPIO[10:17] | **EMIFA:** EM_A[21:14] | **VLYNQ:** VLYNQ_TXD[0:3], VLYNQ_RXD[0:3] | **PinMux0:**AEAW, **Pins:**DAEAW[4:0] | **PinMux0:**VLYNQEN, **PinMux0:**VLYNQWD[1:0] |
| EMIFA, GPIO | **GPIO:** GPIO[18:28] | **EMIFA:** EM_A[13:3] | | **PinMux0:**AEAW, **Pins:**DAEAW[4:0] | |
| ASP, GPIO | **GPIO:** GPIO[29:34] | **ASP:** (all pins)[4] | | **PinMux1:**ASP | |
| UART0, GPIO | **GPIO:** GPIO[35:36] | **UART0:** RXD, TXD | | **PinMux1:**UART0 | |
| SPI, GPIO | **GPIO:** GPIO[37, 39:41] | **SPI:** SPI_EN0, SPI_CLK, SPI_DI, SPI_DO | | **PinMux1:**SPI | |
| SPI, ATA, GPIO | **GPIO:**GPIO[42] | **SPI:** SPI_EN1 | **ATA:** HDDIR | **PinMux1:**SPI | **PinMux0:**HDIREN |
| I2C, GPIO | **GPIO:** GPIO[43:44] | **I2C:** SCL, SDA | | **PinMux1:**I2C | |
| PWM0, GPIO | **GPIO:**GPIO[45] | PWM0 | | **PinMux1:**PWM0 | |
| PWM1, VPBE (RGB666/RGB888), GPIO | **GPIO:**GPIO[46] | **VPBE:** RGB666/RGB888 R2 | **PWM1:** PWM1 | **PinMux0:**RGB666/ **PinMux0:**RGB888 | **PinMux1:**PWM1 |
| PWM2, VPBE (RGB666/RGB888), GPIO | **GPIO:**GPIO[47] | **VPBE:** RGB666/RGB888 B2 | **PWM2:** PWM2 | **PinMux0:**RGB666/ **PinMux0:**RGB888 | **PinMux1:**PWM2 |
| ClockOut0, GPIO | **GPIO:**GPIO[48] | CLK_OUT0 | | **PinMux1:**CLK0 | |
| ClockOut1, TIMER0, GPIO | **GPIO:**GPIO[49] | CLK_OUT1 | **TIMER0:** TIM_IN | **PinMux1:**CLK1 | **PinMux1:**TIM_IN |
| ATA, GPIO | **GPIO:** GPIO[50:51] | **ATA:** $\overline{\text{ATA\_CS0}}$, $\overline{\text{ATA\_CS1}}$ | | **PinMux0:**ATAEN | |
| EMIFA, GPIO, ATA (CF) | **GPIO:**GPIO[52] | **EMIFA:** EM_BA[1] | **ATA (CF):** DA1 | **PinMux0:**AEAW[4:0], **Pins:**DAEAW[4:0] | **PinMux0:**ATAEN |
| EMIFA, HPI, ATA (CF), GPIO | **GPIO:**GPIO[53] | **EMIFA:** EM_A[0] | **ATA (CF):** DA2/ **HPI:** HCNTL1 | **PinMux0:**AEAW[4:0], **Pins**:DAEAW[4:0] | **PinMux0:**ATAEN, **PinMux0:**HPIEN, **Pins:**BTSEL[1:0] = 10 |
| EMAC, GPIO3V | **GPIO:** GPIO3V[0:13] | **EMAC:** (all pins, except CRS)[5] | | **PinMux0:**EMACEN | |
| EMAC, MDIO, GPIO3V | **GPIO:** GPIO3V[14:16] | **EMAC:** CRS, **MDIO:** MDIO, MDCLK | | **PinMux0:**EMACEN | |
| UART1, ATA (CF) | N/A | **ATA (CF):** DMACK, DMARQ | **UART1:** TXD, RXD | **PinMux0:**ATAEN | **PinMux1:**UART1 |
| UART2, VPFE | **VPFE:** CI[7:6]/ CCD_DATA[15:14] | **UART2:** UART_RXD2, UART_TXD2 | | **PinMux1:**UART2 | |

(4) See Section 2.7, *Terminal Functions*, for pin details.
(5) See Section 2.7, *Terminal Functions*, for pin details.

**Table 3-13. DM6446 Multiplexed Peripheral Pins and Multiplexing Controls (continued)**

| MULTIPLEXED PERIPHERALS | PRIMARY (DEFAULT) FUNCTION | SECONDARY[1] FUNCTION | TERTIARY[2] FUNCTION | SECONDARY REGISTER/PIN[3] CONTROL | TERTIARY REGISTER/PIN[3] CONTROL |
|---|---|---|---|---|---|
| UART2, VPFE | **VPFE:** CI[5:4]/ CCD_DATA[13:12] | **UART2:** UART_CTS2, UART_RTS2 | | **PinMux1:**UART2, **PinMux1:**U2FLO | |

### 3.5.3 Peripheral Selection After Device Reset

After device reset, the PINMUX0 and PINMUX1 registers are software programmable to allow multiplexing of shared device pins between peripherals, as given in Section 2.7, *Terminal Functions*. Section 3.5.4 (*PINMUX0 Register Description*), Section 3.5.5 (*PINMUX1 Register Description*), and Section 3.5.6 (*Pin Multiplexing Register Field Details*) identify the register settings necessary to configure specific multiplexed functions and show the primary (default) function after reset.

### 3.5.4 PINMUX0 Register Description

The PINMUX0 pin multiplexing register controls which peripheral is given ownership over shared pins among EMAC, CCD, LCD, RGB888, RGB666, ATA, VLYNQ, EMIFA, HPI, and GPIO peripherals. The register format is shown in Figure 3-7 and bit field descriptions are given in Table 3-14. More details on the PINMUX0 pin muxing fields are given in Section 3.5.6, *Pin Multiplexing Register Field Details*. A value of '1' enables the secondary or tertiary pin function.

**Figure 3-7. PINMUX0 Register[1]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21–18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EMACEN | Rsvd | HPIEN | Rsvd | CFLDEN | CWE | LFLDEN | LOEEN | RGB888 | RGB666 | Reserved | ATAEN | HDIREN |
| R/W-0 | R/W-0 | R/W-D | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0000 | R/W-0 | R/W-0 |

| 15 | 14 | 13–12 | 11 | 10 | 9–5 | 4–0 |
|---|---|---|---|---|---|---|
| VLYNQEN | VLSCREN | VLYNQWD | AECS5 | AECS4 | Reserved | AEAW |
| R/W-0 | R/W-0 | R/W-00 | R/W-0 | R/W-0 | R-00000 | R/W-LLLLL |

LEGEND: R = Read; W = Write; L = pin state latched at reset rising edge; D = derived from pin states; -*n* = value after reset

(1) For proper DM6446 device operation, ***always*** write a value of '0' to RSV bit 30.

**Table 3-14. PINMUX0 Register Description**

| Name | Description |
|---|---|
| EMACEN | Enable EMAC and MDIO function on default GPIO3V[0:16] pins. |
| HPIEN | Enable HPI module pins. Default value is derived from BTSEL[1:0] configuration inputs. HPIEN is 1 when the BTSEL[1:0] = 10 and HPIEN is 0 (the default state) when BTSEL[1:0] is 00, 01, or 11. |
| CFLDEN | Enable CCD C_FIELD function on default GPIO[4] pin |
| CWE | Enable CCD C_WE function on default GPIO[1] pin |
| LFLDEN | Enable LCD_FIELD function on default GPIO[3] pin |
| LOEEN | Enable LCD_OE function on default GPIO[0] pin |
| RGB888 | Enable VPBE RGB888 function on default GPIO[2:6, 46:47] pins |
| RGB666 | Enable VPBE RGB666 function on default GPIO[46:47] pins |
| ATAEN | Enable ATA function on default EMIFA and GPIO[52:53] pins and shared UART1 pins |
| HDIREN | Enable HDDIR function on default GPIO[42] pin |
| VLYNQEN | Enable VLYNQ function on default GPIO[9,10:17] pins |
| VLSCREN | Enable VLYNQ SCRUN function on default GPIO[9] pin |
| VLYNQWD | VLYNQ data width selection. This expands the VLYNQ TXD[0:3] and RXD[0:3] functions on default GPIO[10:17] pins. |
| AECS5 | Enable EMIFA EM_CS5 function on GPIO[8] |

**Table 3-14. PINMUX0 Register Description (continued)**

| Name | Description |
|---|---|
| AECS4 | Enable EMIFA EM_CS4 function on GPIO[9] |
| AEAW | EMIFA address width selection. Default value is latched at reset from AEAW[4:0] configuration input pins. This enables EMIF address function on default GPIO[10:28] pins. |

### 3.5.5 PINMUX1 Register Description

The PINMUX1 pin multiplexing register controls which peripheral is given ownership over shared pins among Timer, PLL, ASP, SPI, I2C, PWM, and UART peripherals. The register format is shown in Figure 3-8 and bit field descriptions are given in Table 3-15. More details on the PINMUX1 pin muxing fields are given in Section 3.5.6, *Pin Multiplexing Register Field Details*. A value of "1" enables the secondary or tertiary pin function.

**Figure 3-8. PINMUX1 Register(1)**

| 31 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|
| Reserved | | TIMIN | CLK1 | CLK0 |
| R-0000 0000 0000 0 | | R/W-0 | R/W-0 | R/W-0 |

| 15 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | ASP | Rsvd | SPI | I2C | PWM2 | PWM1 | PWM0 | U2FLO | UART2 | UART1 | UART0 |
| R-0000 0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

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

(1) For proper DM6446 device operation, ***always*** write a value of '0' to RSV bit 9.

**Table 3-15. PINMUX1 Register Description**

| Name | Description |
|---|---|
| TIMIN | Enable TIM_IN function on default GPIO[49] pin |
| CLK1 | Enable CLK_OUT1 function on default GPIO[49] pin |
| CLK0 | Enable CLK_OUT0 function on default GPIO[48] pin |
| ASP | Enable ASP function on default GPIO[29:34] pins |
| SPI | Enable SPI function on default GPIO[37,39:42] pins |
| I2C | Enable I2C function on default GPIO[43:44] pins |
| PWM2 | Enable PWM2 function on default GPIO[47] pin |
| PWM1 | Enable PWM1 function on default GPIO[46] pin |
| PWM0 | Enable PWM0 function on default GPIO[45] pin |
| U2FLO | Enable UART2 flow control function on default VPFE CI[5:4]/CCD_DATA[13:12] pins |
| UART2 | Enable UART2 function on default VPFE CI[7:6]/CCD_DATA[15:14] pins |
| UART1 | Enable UART1 function on shared ATA (CF) DMACK, DMARQ pins |
| UART0 | Enable UART0 function on default GPIO[35:36] pins |

### 3.5.6 Pin Multiplexing Register Field Details

The bit fields for various pin multiplexing options within the PINMUX0 and PINMUX1 registers are described in the following sections.

#### 3.5.6.1 EMAC and GPIO3V Pin Multiplexing

The EMAC pin functions are selected as shown in Table 3-16. The functionality for each of the individual pins affected by the PINMUX0 field settings is given in Table 3-17.

**Table 3-16. EMAC and GPIO3V Pin Multiplexing Control**

| EMACEN | PIN FUNCTIONALITY SELECTED |
|---|---|
| 0 | GPIO3V |
| 1 | EMAC |

**Table 3-17. EMAC and GPIO3V Multiplexed Pins**

| GPIO | EMAC |
|---|---|
| GPIO3V[0] | TXEN |
| GPIO3V[1] | TXCLK |
| GPIO3V[2] | COL |
| GPIO3V[3] | TXD[0] |
| GPIO3V[4] | TXD[1] |
| GPIO3V[5] | TXD[2] |
| GPIO3V[6] | TXD[3] |
| GPIO3V[7] | RXD[0] |
| GPIO3V[8] | RXD[1] |
| GPIO3V[9] | RXD[2] |
| GPIO3V[10] | RXD[3] |
| GPIO3V[11] | RXCLK |
| GPIO3V[12] | RXDV |
| GPIO3V[13] | RXER |
| GPIO3V[14] | CRS |
| GPIO3V[15] | MDIO |
| GPIO3V[16] | MDCLK |

### 3.5.6.2 VPFE (CCD), VPBE (LCD), and GPIO Pin Multiplexing

The CCD and LCD controllers in the VPSS require multiplex control bit settings for certain modes of operation. Bits within the PinMux0 register, which select between the CCD or LCD control signal function and GPIO, are summarized in Table 3-18.

**Table 3-18. VPFE (CCD), VPBE (LCD), and GPIO Pin Multiplexing**

| PINMUX0 REGISTER FIELDS | | | | MULTIPLEXED PINS | | | |
|---|---|---|---|---|---|---|---|
| CFLDEN | LFLDEN | CWE | LOEEN | C_FIELD/ R0/ GPIO[4] | LCD_FIELD/ B0/ GPIO[3] | C_WE/ GPIO[1] | LCD_OE/ GPIO[0] |
| - | - | - | 0 | - | - | - | GPIO[0] |
| - | - | - | 1 | - | - | - | LCD_OE |
| - | - | 0 | - | - | - | GPIO[1] | - |
| - | - | 1 | - | - | - | C_WE | - |
| - | 0 | - | - | - | B0/GPIO[3] (1) | - | - |
| - | 1 | - | - | - | LCD_FIELD | - | - |
| 0 | - | - | - | R0/GPIO[4] (1) | - | - | - |
| 1 | - | - | - | C_FIELD | - | - | - |

(1) Depends on RGB888 bit setting, see Table 3-19.

### 3.5.6.3 VPBE (RGB666 and RGB888) and GPIO Pin Multiplexing

Use of the RGB666 and RGB888 modes of the VPBE requires enabling RGB pins as shown in Table 3-19 and Table 3-20. Enabling PWM2, PWM1, CCD, and LCD functionality overrides the RGB modes. RGB666 interface pin functionality requires setting the RGB666 PINMUX0 Register bit field to ‘1’ and PINMUX1 Register bit fields PWM2 and PWM1 to ‘0’. Proper RGB888 interface operation requires setting PINMUX0 Register bit field RGB888 to ‘1’ and bit fields PWM2, PWM1, CFLDEN, and LFLDEN must be set to ‘0’.

**Table 3-19. VPBE (RGB666, RGB888, and LCD), VPFE (CCD), and GPIO Pin Multiplexing**

| PINMUX0 AND PINMUX1 REGISTER BIT FIELDS | | | | | | MULTIPLEXED PINS | | | |
|---|---|---|---|---|---|---|---|---|---|
| RGB888 | RGB666 | PWM2 | PWM1 | CFLDEN | LFLDEN | PWM2/ B2/ GPIO[47] | PWM1/ R2/ GPIO[46] | C_FIELD/ R0/ GPIO[4] | LCD_FIELD/ B0/ GPIO[3] |
| 0 | 0 | 0 | 0 | 0 | 0 | GPIO[47] | GPIO[46] | GPIO[4] | GPIO[3] |
| - | - | - | - | - | 1 | - | - | - | LCD_FIELD |
| - | - | - | - | 1 | - | - | - | C_FIELD | - |
| - | - | - | 1 | - | - | - | PWM1 | - | - |
| - | - | 1 | - | - | - | PWM2 | - | - | - |
| 0 | 1 | 0 | 0 | 0 | 0 | B2 | R2 | GPIO[4] | GPIO[3] |
| 1 | - | 0 | 0 | 0 | 0 | B2 | R2 | R0 | B0 |

**Table 3-20. VPBE (RGB666, RGB888, and LCD) and GPIO Pin Multiplexing**

| PINMUX0 AND PINMUX1 REGISTER BIT FIELDS | | | | | MULTIPLEXED PINS | | | |
|---|---|---|---|---|---|---|---|---|
| RGB888 | PWM2 | PWM1 | CFLDEN | LFLDEN | R1/ GPIO[38] | B1/ GPIO[6] | G1/ GPIO[5] | G0/ GPIO[2] |
| 0 | 0 | 0 | 0 | 0 | GPIO[38] | GPIO[6] | GPIO[5] | GPIO[2] |
| 1 | 0 | 0 | 0 | 0 | R1 | B1 | G1 | G0 |

### 3.5.6.4 ATA, EMIFA, UART1, SPI, and GPIO Pin Multiplexing

The ATA peripheral shares pins with the EMIFA and UART1 as seen in Table 3-21. If ATA pin functionality is enabled by setting the ATAEN bit field, the ATA module will drive the EMIFA data and control pins. Enabling UART1 disables the use of the ATA DMARQ and DMACK signals and thus only allows the ATA module to use PIO mode. The ATA HDDIR buffer direction control bit field works in conjunction with the HDIREN enable bit field to allow the ATA pins to still be used as a GPIO or SPI_EN1 if the buffer is not being used (i.e. for Compact Flash). This multiplexing is shown in Table 3-22. When ATAEN=0 and HDIREN=1 it indicates that the ATA interface has been disabled so that the EMIFA can be used, but the ATA buffers are still present. HDDIR is driven low in this situation to ensure that the ATA buffers drive away from DM644X and don't cause bus contention with the EMIFA. Note that switching between EMIFA and ATA (clearing or setting ATAEN) must be carefully performed to prevent bus contention. Since the ATA device can be a bus master, software must ensure that all outstanding DMA requests have completed before clearing the ATAEN bit.

**Table 3-21. ATA, EMIFA, and GPIO Pin Multiplexing Control(1)**

| PINMUX0 REGISTER BIT FIELD | MULTIPLEXED PINS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ATAEN | GPIO[50]/ ATA_CS0 | GPIO[51]/ ATA_CS1 | EM_R/$\overline{W}$ INTRQ | EM_BA[0]/ ATA0 | EM_WAIT/ $\overline{IORDY}$ | $\overline{DIOR}$/ $\overline{EM\_OE}$ | $\overline{DIOW}$/ $\overline{EM\_WE}$ | EM_BA[1]/ GPIO[52]/ ATA1 | EM_A[0]/ GPIO[53]/ ATA2 | EM_D[15:0]/ DD[15:0] |
| 0 | GPIO[50] | GPIO[51] | EM_R/$\overline{W}$ | EM_BA[0] | EM_WAIT | $\overline{EM\_OE}$ | $\overline{EM\_WE}$ | EM_BA[1]/ GPIO[52](2) | EM_A[0]/ GPIO[53](2) | EM_D[15:0] |
| 1 | ATA_CS0 | ATA_CS1 | INTRQ | ATA0 | $\overline{IORDY}$ | $\overline{DIOR}$ | $\overline{DIOW}$ | ATA1 | ATA2 | DD[15:0] |

(1) This table assumes that the HPIEN bit in the PINMUX0 register is "0".
(2) This pin shares GPIO functionality set by AEAW[4:0] as shown in Table 3-9.

**Table 3-22. ATA, EMIFA, UART1, SPI, and GPIO Pin Multiplexing**

| PINMUX0 AND PINMUX1 REGISTER BIT FIELDS | | | | MULTIPLEXED PINS | | |
|---|---|---|---|---|---|---|
| ATAEN | UART1 | HDIREN | SPI | UART_TXD1/ DMACK | UART_RXD1/ DMARQ | SPI_EN1/ HDDIR/ GPIO[42] |
| 0 | 0 | 0 | 0 | DMACK | DMARQ | GPIO[42] |
| 0 | 0 | 0 | 1 | DMACK | DMARQ | SPI_EN1 |
| 0 | 0 | 1 | - | DMACK | DMARQ | Driven Low |
| 0 | 1 | 0 | 0 | UART_TXD1 | UART_RXD1 | GPIO[42] |
| 0 | 1 | 0 | 1 | UART_TXD1 | UART_RXD1 | SPI_EN1 |
| 0 | 1 | 1 | - | UART_TXD1 | UART_RXD1 | Driven Low |
| 1 | 0 | 0 | 0 | DMACK | DMARQ | GPIO[42]x |
| 1 | 0 | 0 | 1 | DMACK | DMARQ | SPI_EN1x |
| 1 | 0 | 1 | - | DMACK | DMARQ | HDDIR |
| 1 | 1 | 0 | 0 | UART_TXD1 | UART_RXD1 | GPIO[42]x |
| 1 | 1 | 0 | 1 | UART_TXD1 | UART_RXD1 | SPI_EN1x |
| 1 | 1 | 1 | - | UART_TXD1 | UART_RXD1 | HDDIR |

### 3.5.6.5 VLYNQ, EMIFA, and GPIO Pin Multiplexing

Table 3-23 and Table 3-24 show the VLYNQ pin control and multiplexing. If VLYNQ is disabled (VLYNQEN=0), the AECS5 and AECS4 bits select between the GPIO[8] / EMIFA EM_CS5 and GPIO[9] / EMIFA EM_CS4 functions, and the AEAW field determines the partitioning between GPIO and the upper EMIFA address pins. If VLYNQ is enabled (VLYNQEN=1), VLYNQ_CLOCK, VLYNQ_TXD0, and VLYNQ_RXD0 are always selected. The VLYNQ_SCRUN function is only enabled if VLYNQEN=1 and VLSCREN=1 (VLSCREN overrides AECS4). The remaining VLYNQ TX/RX pins are selected based on the VLYNQWD value. Unselected VLYNQ TX/RX pins will function as either GPIO or EMIFA address based on the AEAW value.

**Table 3-23. VLYNQ Control, EMIFA, and GPIO Pin Multiplexing**

| PINMUX0 REGISTER BIT FIELDS | | | | MULTIPLEXED PINS | |
|---|---|---|---|---|---|
| VLYNQEN | VLSCREN | AECS5 | AECS4 | EM_CS5/ GPIO[8]/ VLYNQ_CLOCK | EM_CS4/ GPIO[9]/ VLYNQ_SCRUN |
| 0 | - | 0 | 0 | GPIO[8] | GPIO[9] |
| 0 | - | 0 | 1 | GPIO[8] | EM_CS4 |
| 0 | - | 1 | 0 | EM_CS5 | GPIO[9] |
| 0 | - | 1 | 1 | EM_CS5 | EM_CS4 |
| 1 | 0 | - | 0 | VLYNQ_CLOCK | GPIO[9] |
| 1 | 0 | - | 1 | VLYNQ_CLOCK | EM_CS4 |
| 1 | 1 | - | - | VLYNQ_CLOCK | VLYNQ_SCRUN |

**Table 3-24. VLYNQ Data, EMIFA, and GPIO Pin Multiplexing**

| PINMUX0 REGISTER BIT FIELDS | | MULTIPLEXED PINS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| VLYNQEN | VLYNQWD | EM_A[21]/ GPIO[10]/ VL_TXD0 | EM_A[20]/ GPIO[11]/ VL_RXD0 | EM_A[19]/ GPIO[12]/ VL_TXD1 | EM_A[18]/ GPIO[13]/ VL_RXD1 | EM_A[17]/ GPIO[14]/ VL_TXD2 | EM_A[16]/ GPIO[15]/ VL_RXD2 | EM_A[15]/ GPIO[16]/ VL_TXD3 | EM_A[14]/ GPIO[17]/ VL_RXD3 |
| 0 | - | EM_A[21]/ GPIO[10](1) | EM_A[20]/ GPIO[11](1) | EM_A[19]/ GPIO[12](1) | EM_A[18]/ GPIO[13](1) | EM_A[17]/ GPIO[14](1) | EM_A[16]/ GPIO[15](1) | EM_A[15]/ GPIO[16](1) | EM_A[14]/ GPIO[17](1) |
| 1 | 00 | VL_TXD0 | VLRXD0 | EM_A[19]/ GPIO[12](1) | EM_A[18]/ GPIO[13](1) | EM_A[17]/ GPIO[14](1) | EM_A[16]/ GPIO[15](1) | EM_A[15]/ GPIO[16](1) | EM_A[14]/ GPIO[17](1) |
| 1 | 01 | VL_TXD0 | VLRXD0 | VL_TXD1 | VLRXD1 | EM_A[17]/ GPIO[14](1) | EM_A[16]/ GPIO[15](1) | EM_A[15]/ GPIO[16](1) | EM_A[14]/ GPIO[17](1) |
| 1 | 10 | VL_TXD0 | VLRXD0 | VL_TXD1 | VLRXD1 | VL_TXD2 | VLRXD2 | EM_A[15]/ GPIO[16](1) | EM_A[14]/ GPIO[17](1) |
| 1 | 11 | VL_TXD0 | VLRXD0 | VL_TXD1 | VLRXD1 | VL_TXD2 | VLRXD2 | VL_TXD3 | VLRXD3 |

(1) This pin shares GPIO functionality set by AEAW[4:0] as shown in Table 3-9.

### 3.5.6.6 Timer0 Input, CLK_OUT1, and GPIO Pin Multiplexing

The multiplexing of the CLK_OUT1 and Timer0 Input (Timer 0 only) functions is shown in Table 3-25.

**Table 3-25. Timer0 Input, CLK_OUT1, and GPIO Pin Multiplexing**

| PINMUX1 REGISTER BIT FIELDS | | MULTIPLEXED PINS |
|---|---|---|
| TIMIN | CLK1 | CLK_OUT1/ TIM_IN/ GPIO[49] |
| 0 | 0 | GPIO[49] |
| 0 | 1 | CLK_OUT1 |
| 1 | - | TIM_IN |

### 3.5.6.7 ASP, SPI, I2C, ATA, and GPIO Pin Multiplexing

When the ASP, SPI, or I2C serial port functions are not selected, their pins may be used as GPIOs as seen in Table 3-26, Table 3-27, and Table 3-28. The SPI_EN1 pin can also function as the HDDIR buffer control when ATAEN is selected and the HDIREN bit is set.

**Table 3-26. ASP and GPIO Pin Multiplexing**

| PINMUX1 REGISTER BIT FIELD | MULTIPLEXED PINS | | | | | |
|---|---|---|---|---|---|---|
| ASP | CLKX/ GPIO[29] | CLKR/ GPIO[30] | FSX/ GPIO[31] | FSR/ GPIO[32] | DX/ GPIO[33] | DR/ GPIO[34] |
| 0 | GPIO[29] | GPIO[30] | GPIO[31] | GPIO[32] | GPIO[33] | GPIO[34] |
| 1 | CLKX | CLKR | FSX | FSR | DX | DR |

**Table 3-27. SPI and GPIO Pin Multiplexing**

| PINMUX0 AND PINMUX1 REGISTER BIT FIELDS | | | MULTIPLEXED PINS | | | | |
|---|---|---|---|---|---|---|---|
| SPI | ATAEN | HDIREN | SP_EN1/ HDDIR/ GPIO[42] | SPI_DO/ GPIO[41] | SPI_DI/ GPIO[40] | SPI_CLK/ GPIO[39] | SPI_EN0/ GPIO[37] |
| 0 | 0 | 0 | GPIO[42] | GPIO[41] | GPIO[40] | GPIO[39] | GPIO[37] |
| 0 | 0 | 1 | Driven Low | GPIO[41] | GPIO[40] | GPIO[39] | GPIO[37] |
| 0 | 1 | 0 | GPIO[42] | GPIO[41] | GPIO[40] | GPIO[39] | GPIO[37] |
| 0 | 1 | 1 | HDDIR | GPIO[41] | GPIO[40] | GPIO[39] | GPIO[37] |
| 1 | 0 | 0 | SP_EN1 | SPI_DO | SPI_DI | SPI_CLK | SPI_EN0 |
| 1 | 0 | 1 | Driven Low | SPI_DO | SPI_DI | SPI_CLK | SPI_EN0 |
| 1 | 1 | 0 | SP_EN1 | SPI_DO | SPI_DI | SPI_CLK | SPI_EN0 |
| 1 | 1 | 1 | HDDIR | SPI_DO | SPI_DI | SPI_CLK | SPI_EN0 |

**Table 3-28. I2C and GPIO Pin Multiplexing**

| PINMUX1 REGISTER BIT FIELD | MULTIPLEXED PINS | |
|---|---|---|
| I2C | I2C_CLK/ GPIO[43] | I2C_DATA/ GPIO[44] |
| 0 | GPIO[43] | GPIO[44] |
| 1 | I2C_CLK | I2C_DATA |

### 3.5.6.8 PWM, RGB888, and GPIO Pin Multiplexing

Table 3-29 shows the PWM0/1/2 pin multiplexing. Each PWM output is independently controlled by its own enable bit. The PWM function has priority over RGB888 muxing [see Section 3.5.6.3, *VPBE (RGB666 and RGB888) and GPIO Pin Multiplexing*].

**Table 3-29. PWM0/1/2, RGB888, and GPIO Pin Multiplexing**

| PINMUX1 REGISTER BIT FIELDS | | | | MULTIPLEXED PINS | | |
|---|---|---|---|---|---|---|
| PWM2 | PWM1 | PWM0 | RGB888 | PWM2/ B2/ GPIO[47] | PWM1/ R2/ GPIO[46] | PWM0/ GPIO[45] |
| 0 | 0 | 0 | 0 | GPIO[47] | GPIO[46] | GPIO[45] |
| 0 | 0 | 0 | 1 | B2 | R2 | GPIO[45] |
| - | - | 1 | - | - | - | PWM0 |
| - | 1 | - | - | - | PWM1 | - |
| 1 | - | - | - | PWM2 | - | - |

### 3.5.6.9 UART, VPFE, ATA, and GPIO Pin Multiplexing

Each UART has independent pin multiplexing control bits in the PINMUX1 register. The UART2 peripheral may be used with or without the flow control signals. Table 3-30 shows how UART2 selection reduces the width of the VPFE interface.

Setting the UART1 bit enables UART1 transmit and receive pin functionality. Since these are shared with the ATA DMA handshake signals, enabling UART1 effectively disables the ATA DMA mode. However, ATA PIO mode is still supported with UART1 enabled. This is shown in Table 3-31. If the ATA module is not enabled, the pins are always configured for use by UART1.

**Table 3-30. UART2, VPFE, and GPIO Pin Multiplexing**

| PINMUX1 REGISTER BIT FIELDS | | MULTIPLEXED PINS | | | |
|---|---|---|---|---|---|
| UART2 | U2FLO | CCD[15]/ CI[7]/ UART_RXD2 | CCD[14]/ CI[6]/ UART_TXD2 | CCD[13]/ CI[5]/ UART_CTS2 | CCD[12]/ CI[4]/ UART_RTS2 |
| 0 | - | CCD[15]/ CI[7](1) | CCD[14]/ CI[6](1) | CCD[13]/ CI[5](1) | CCD[12]/ CI[4](1) |
| 1 | 0 | UART_RXD2 | UART_TXD2 | CCD[13]/ CI[5](1) | CCD[12]/ CI[4](1) |
| 1 | 1 | UART_RXD2 | UART_TXD2 | UART_CTS2 | UART_RTS2 |

(1) Functionality set by VPFE operating mode.

**Table 3-31. UART1 and ATA Pin Multiplexing**

| PINMUX0 AND PINMUX1 REGISTER BIT FIELDS | | MULTIPLEXED PINS | |
|---|---|---|---|
| ATAEN | UART1 | UART_TXD1/ DMACK | UART_RXD1/ DMARQ |
| 0 | - | UART_TXD1 | UART_RXD1 |
| 1 | 0 | DMACK | DMARQ |
| 1 | 1 | UART_TXD1 | UART_RXD1 |

As Table 3-32 shows, the UART0 pins are configurable for either UART0 transmit and receive data functions or for GPIO.

**Table 3-32. UART0 and GPIO Pin Multiplexing**

| PINMUX1 REGISTER BIT FIELD | MULTIPLEXED PINS | |
|---|---|---|
| UART0 | UART_TXD0/ GPIO[36] | UART_RXD0/ GPIO[35] |
| 0 | GPIO[36] | GPIO[35] |
| 1 | UART_TXD0 | UART_RXD0 |

### 3.5.6.10 HPI and EMIFA/ATA Pin Multiplexing

When the HPIEN bit is set, the HPI module is given control of most of the EMIFA/ATA control pins as well as the EMIFA/ATA data bus. Table 3-33 shows which pins the HPI controls. HPIEN is set to 1 when the state of the BTSEL[1:0] pins = 10 is latched at the rising edge of reset. Also, this bit can be manipulated after reset by software. When the ATAEN bit is set and HPIEN is 0, the ATA mode of operation for pins shared with the HPI is available. EMIFA mode functionality for the shared HPI pins is set when both HPIEN and ATAEN are '0'.

**Table 3-33. HPI and EMIFA/ATA Pin Multiplexing**

| PINMUX0 REGISTER BIT FIELDS | | MULTIPLEXED PINS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HPI EN | ATA EN | $\overline{\text{HCS}}$/ $\overline{\text{EM\_CS2}}$ | HHWIL/ EM_A[1] | HR/$\overline{\text{W}}$/ INTRQ/ EM_R/$\overline{\text{W}}$ | $\overline{\text{HRDY}}$/ EM_WAIT/ $\overline{\text{IORDY}}$ | $\overline{\text{HDS1}}$/ $\overline{\text{DIOR}}$/ $\overline{\text{EM\_OE}}$ | $\overline{\text{HDS2}}$/ $\overline{\text{DIOW}}$/ $\overline{\text{EM\_WE}}$ | HCNTLA/ EM_A[2] | HCNTLB/ ATA2/ EM_A[0] | $\overline{\text{HINT}}$/ ATA0/ EM_BA[0] | HD[15:0]/ DD[15:0]/ EM_D[15:0] |
| 0 | 0 | $\overline{\text{EM\_CS2}}$ | EM_A[1](1) | EM_R/$\overline{\text{W}}$ | EM_WAIT | $\overline{\text{EM\_OE}}$ | $\overline{\text{EM\_WE}}$ | EM_A[2](1) | EM_A[0](1) | EM_BA[0] | EM_D[15:0] |
| 0 | 1 | $\overline{\text{EM\_CS2}}$ | EM_A[1](1) | INTRQ | $\overline{\text{IORDY}}$ | $\overline{\text{DIOR}}$ | $\overline{\text{DIOW}}$ | EM_A[2](1) | EM_A[0](1) | ATA0 | DD[15:0] |
| 1 | - | $\overline{\text{HCS}}$ | HHWIL | HR/$\overline{\text{W}}$ | $\overline{\text{HRDY}}$ | $\overline{\text{HDS1}}$ | $\overline{\text{HDS2}}$ | HCNTLA | HCNTLB | $\overline{\text{HINT}}$ | HD[15:0] |

(1) This pin shares GPIO functionality and is set by AEAW[4:0] as shown in Table 3-12, Table 3-13, and Table 3-14.

## 3.6 Emulation Control

The flexibility of the DM644x architecture allows either the ARM or DSP to control the various peripherals (setup registers, service interrupts, etc.). While this assignment is purely a matter of software convention, during an emulation halt it is necessary for the device to know which peripherals are associated with the halting processor so that only those modules receive the suspend signal. This allows peripherals associated with the other (unhalted) processor to continue normal operation. The SUSPSRC register indicates the emulation suspend source for those peripherals which support emulation suspend. The SUSPSRC register format is shown in Figure 3-9. Brief details on the peripherals which correspond to the register bits is given in Table 3-34. When the associated SUSPSRC bit is '0', the peripheral's emulation suspend signal is controlled by the ARM emulator and when set to '1' it is controlled by the DSP emulator.

**Figure 3-9. Emulation Suspend Source Register (SUSPSRC)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VICP SRC | VICP EN | TIMR2 SRC | TIMR1 SRC | TIMR0 SRC | GPIO SRC | PWM2 SRC | PWM1 SRC | PWM0 SRC | SPI SRC | UART2 SRC | UART1 SRC | UART0 SRC | I2C SRC | ASP SRC | Rsvd |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |

| 15 | 13 | 12 | 11 | 10 | 9 | 8 | 6 | 5 | 4 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rsvd | | HPI SRC | Rsvd | | USB SRC | Rsvd | | EMAC SRC | Reserved | |
| R-000 | | R/W-0 | R-00 | | R/W-0 | R-000 | | R/W-0 | R-0 0000 | |

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

**Table 3-34. SUSPSRC Register Description**

| Name | Description |
|---|---|
| VICPSRC | Video Imaging Coprocessor emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| VICPEN | Video Imaging Coprocessor emulation suspend enable<br>0 = Emulation suspend ignored by VICP<br>1 = VICP emulation suspend enabled |
| TIMR2SRC | Timer2 (WD Timer) emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| TIMR1SRC | Timer1 emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| TIMR0SRC | Timer0 emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| GPIOSRC | GPIO emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| PWM2SRC | PWM2 emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| PWM1SRC | PWM1 emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| PWM0 SRC | PWM0 emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| SPISRC | SPI emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| UART2SRC | UART2 emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| UART1SRC | UART1 emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| UART0SRC | UART0 emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| I2CSRC | I2C emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| ASPSRC | ASP emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| HPISRC | HPI emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| USBSRC | USB emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |
| EMACSRC | Ethernet MAC emulation suspend source<br>0 = ARM emulation suspend<br>1 = DSP emulation suspend |

# 4 System Interconnect

On the DM6446 device, the C64x+ megamodule, the ARM subsystem, the EDMA3 transfer controllers, and the system peripherals are interconnected through a switch fabric architecture (shown in Figure 4-1). The switch fabric is composed of multiple switched central resources (SCRs) and multiple bridges. The SCRs establish low-latency connectivity between master peripherals and slave peripherals. Additionally, the SCRs provide priority-based arbitration and facilitate concurrent data movement between master and slave peripherals. Through SCR, the ARM subsystem can send data to the DDR2 Memory Controller without affecting a data transfer between the EMAC and L2 memory. Bridges are mainly used to perform bus-width conversion as well as bus operating frequency conversion. For example, in Figure 4-1, Bridge 8 performs a frequency conversion between a bus operating at DSP/6 clock rate and a bus operating at DSP/3 clock rate. Furthermore, Bridge 3 performs a bus-width conversion between a 64-bit bus and a 32-bit bus.

The C64x+ megamodule, the ARM subsystem, the EDMA3 transfer controllers, and the various system peripherals can be classified into two categories: master peripherals and slave peripherals. Master peripherals are typically capable of initiating read and write transfers in the system and do not rely on the EDMA3 or on a CPU to perform transfers to and from them. The system master peripherals include the C64x+ megamodule, the ARM subsystem, the EDMA3 transfer controllers, CF/ATA, VLYNQ, EMAC, USB, and VPSS. Not all master peripherals may connect to all slave peripherals. The supported connections are designated by an X in Table 4-1.

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

| MASTER | SLAVE | | | |
|---|---|---|---|---|
| | C64x+ | ARM | DDR2 MEMORY CONTROLLER | SCR3(1) |
| C64x+ | | X | X | X |
| ARM | X | | X | X |
| VPSS | | | X | |
| CF/ATA | X | X | X | X |
| VLYNQ | X | X | X | X |
| EMAC | X | X | X | X |
| USB | X | X | X | X |
| EDMA3TC0 | X | X | X | X |
| EDMA3TC1 | X | X | X | X |
| HPI | | X | X | X(2) |

(1) The C64x+ megamodule has access to only the following peripherals connected to SCR3: EDMA3, ASP, and Timers. All other peripherals/modules that support a connection to SCR3 have access to all peripherals/modules connected to SCR3.
(2) HPI's access to SCR3 is limited to the power and sleep controller registers, PLL1 and PLL2 registers, and HPI configuration registers.

## 4.1 System Interconnect Block Diagram

Figure 4-1 displays the DM6446 system interconnect block diagram. The following is a list that helps interpret this diagram:

- The direction of the arrows indicates either bus master or bus slave.
- The arrow originates at a bus master and terminates at a bus slave.
- The direction of the arrows does not indicate the direction of data flow. Data flow is typically bi-directional for each of the documented bus paths.
- The pattern of each arrow's line indicates the clock rate at which it is operating, either DSP/2, DSP/3, or DSP/6 clock rate.
- Some peripherals may have multiple instances shown in the diagram. A peripheral may have multiple instances shown for a variety of reasons, some of which are described below:
  - The peripheral/module has master port(s) for data transfers, as well as slave port(s) for register access, data access, and/or memory access. Examples of these peripherals are C64x+ megamodule, EDMA3, CF/ATA, USB, EMAC, VPSS, VLYNQ, and HPI.
  - The peripheral/module has a master port as well as slave memories. Examples of these are the C64x+ megamodule and the ARM subsystem.

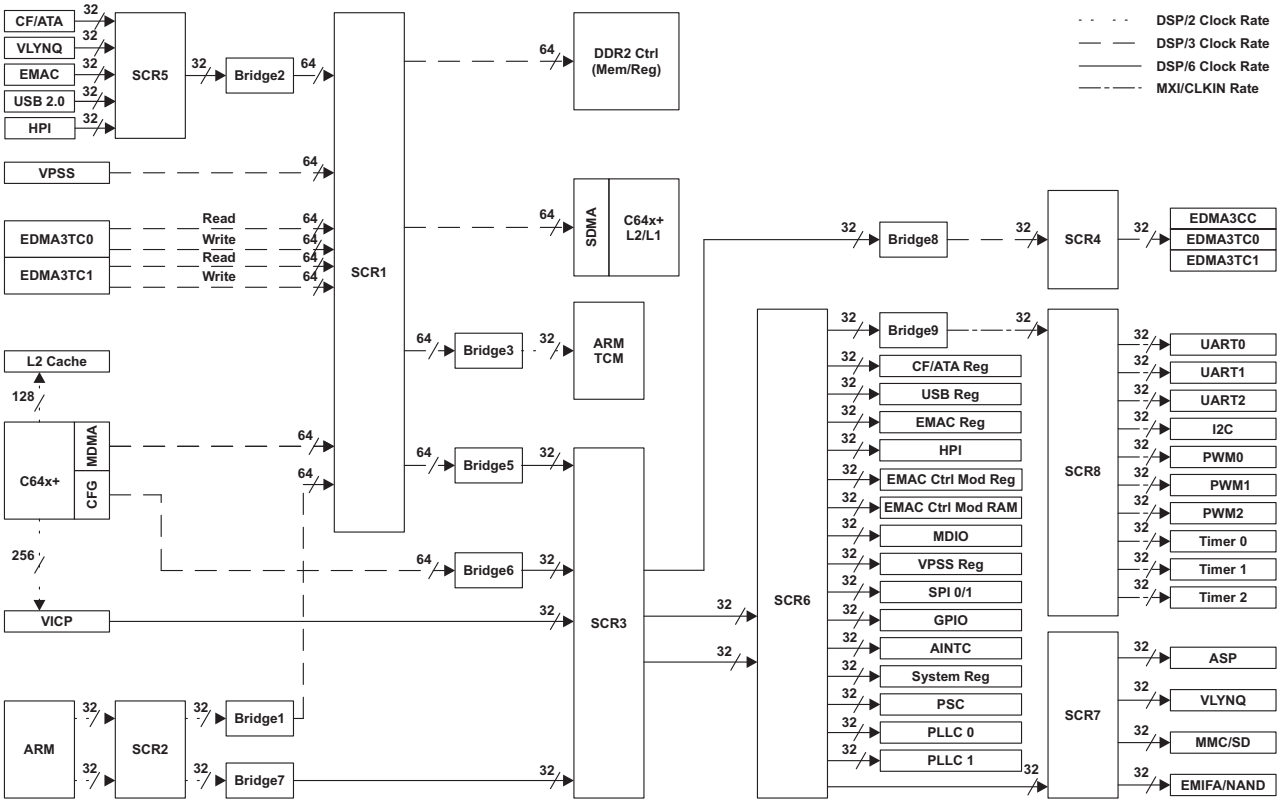



**Figure 4-1. System Interconnect Block Diagram**

# 5 Device Operating Conditions

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

| | | |
|---|---|---|
| Supply voltage ranges | Core ($CV_{DD}$, $V_{DDA1P1V}$, $USB\_V_{DDA1P2LDO}$ (2), $CV_{DDDSP}$) (3) | -0.5 V to 1.5 V |
| | I/O, 3.3V ($DV_{DD33}$, $USB\_V_{DDA3P3}$) (3) | -0.5 V to 4.2 V |
| | I/O, 1.8V ($DV_{DD18}$, $DV_{DDR2}$, $DDR\_V_{DDDLL}$, $PLLV_{DD18}$, $V_{DDA1P8V}$, $USB\_V_{DD1P8}$, $MXV_{DD}$, $M24V_{DD}$) (3) | -0.5 V to 2.5 V |
| Input voltage ranges | $V_I$ I/O, 3.3V | -0.5 V to 4.2 V |
| | $V_I$ I/O, 1.8V | -0.5 V to 2.5 V |
| Output voltage ranges | $V_O$ I/O, 3.3V | -0.5 V to 4.2 V |
| | $V_O$ I/O, 1.8V | -0.5 V to 2.5 V |
| Operating case temperature ranges, $T_C$ | (default) [-594, -810 devices] | 0°C to 85°C |
| | (A version) [A-513 device] | -40°C to 105°C |
| Storage temperature range, $T_{stg}$ | (default) | -55°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) This pin is an internal LDO output and connected via 1 µF capacitor to $USB\_V_{SSA1P2LDO}$.
(3) All voltage values are with respect to $V_{SS}$.

## 5.2 Recommended Operating Conditions

| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $CV_{DD}$ | Supply voltage, Core ($CV_{DD}$, $CV_{DDDSP}$) (1) | (-810 devices) | 1.235 | 1.3 | 1.365 | V |
| | | (A-513, -594 devices) | 1.14 | 1.2 | 1.26 | V |
| | Supply voltage, Core ($V_{DDA1P1V}$, $USB\_V_{DDA1P2LDO}$ (2)) (1) | (All devices) | 1.14 | 1.2 | 1.26 | V |
| $DV_{DD}$ | Supply voltage, I/O, 3.3V ($DV_{DD33}$, $USB\_DV_{DDA3P3}$) | | 3.15 | 3.3 | 3.45 | V |
| | Supply voltage, I/O, 1.8V ($DV_{DD18}$, $DV_{DDR2}$, $DDR\_V_{DDDLL}$, $PLLV_{DD18}$, $V_{DDA1P8V}$, $USB\_V_{DD1P8}$, $MXV_{DD}$, $M24V_{DD}$) | | 1.71 | 1.8 | 1.89 | V |
| $V_{SS}$ | Supply ground ($V_{SS}$, $V_{SSA1P8V}$, $V_{SSA1P1V}$, $DDR\_V_{SSDLL}$, $USB\_V_{SSREF}$, $USB\_V_{SS1P8}$, $USB\_V_{SSA3P3}$, $USB\_V_{SSA1P2LDO}$, $MXV_{SS}$ (3), $M24V_{SS}$ (3)) | | 0 | 0 | 0 | V |
| DDR_VREF | DDR2 reference voltage (4) | | $0.49DV_{DDR2}$ | $0.5DV_{DDR2}$ | $0.51DV_{DDR2}$ | V |
| DDR_ZP | DDR2 impedance control, connected via 200 Ω resistor to $V_{SS}$ | | | $V_{SS}$ | | V |
| DDR_ZN | DDR2 impedance control, connected via 200 Ω resistor to $DV_{DDR2}$ | | | $DV_{DDR2}$ | | V |
| DAC_VREF | DAC reference voltage input | | 0.475 | 0.5 | 0.525 | V |
| DAC_RBIAS | DAC biasing, connected via 4 kΩ resistor to $V_{SSA\_1P8V}$ | | | $V_{SSA\_1P8V}$ | | V |
| USB_VBUS | USB external charge pump input | | 4.75 | 5 | 5.25 | V |
| $V_{IH}$ | High-level input voltage, I/O, 3.3V | | 2 | | | V |
| | High-level input voltage, non-DDR I/O, 1.8V | | $0.65DV_{DD}$ | | | V |
| $V_{IL}$ | Low-level input voltage, I/O, 3.3V | | | | 0.8 | V |
| | Low-level input voltage, non-DDR I/O, 1.8V | | | | $0.35DV_{DD}$ | V |
| $T_C$ | Operating case temperature | Default (-594, -810) | 0 | | 85 | °C |
| | | A version (A-513 device) | -40 | | 105 | °C |
| $F_{SYSCLK1}$ | DSP Operating Frequency (SYSCLK1) | Default( -594) | 20 | | 600 | MHz |
| | | Default (-810) | 20 | | 810 | MHz |
| | | A version (A-513 device) | 20 | | 513 | MHz |

(1) Future variants of TI SOC devices may operate at voltages ranging from 0.9 V to 1.4 V to provide a range of system power/performance options. TI highly recommends that users design-in a supply that can handle multiple voltages within this range (i.e., 1.0 V, 1.05 V, 1.1 V, 1.14 V, 1.2 V, 1.235 V, 1.26 V, 1.3 V, 1.365 V with ±3% tolerances) by implementing simple board changes such as reference resistor values or input pin configuration modifications. Not incorporating a flexible supply may limit the system's ability to easily adapt to future versions of TI SOC devices.
(2) This pin is an internal LDO output and connected via 1 μF capacitor to $USB\_V_{SSA1P2LDO}$.
(3) Oscillator ground must be kept separate from other grounds and connected directly to the crystal load capacitor ground.
(4) DDR_VREF is expected to equal $0.5DV_{DDR2}$ of the transmitting device and to track variations in the $DV_{DDR2}$.

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

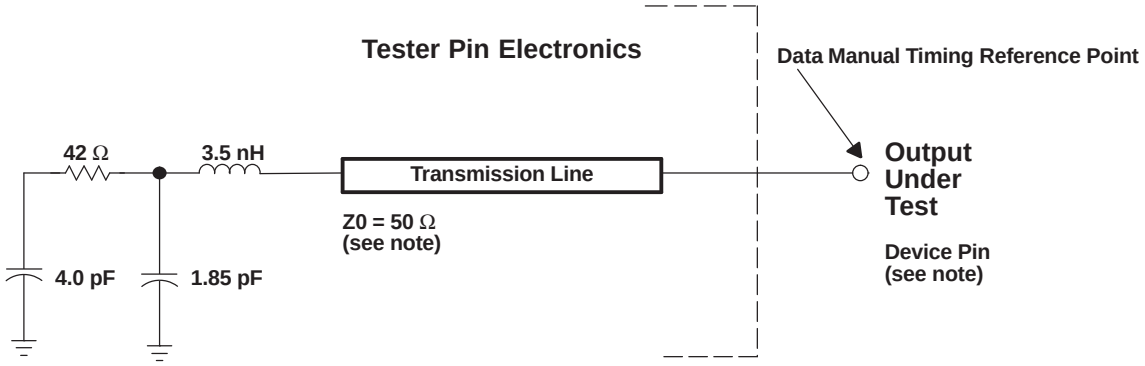
| PARAMETER | | TEST CONDITIONS (1) | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_{OH}$ | Low/full speed: USB_DN and USB_DP | | 2.8 | | $USB\_V_{DDAP3}$ | V |
| | High speed: USB_DN and USB_DP | | 360 | | 440 | mV |
| | High-level output voltage (3.3V I/O) | $DV_{DD33}$ = MIN, $I_{OH}$ = MAX | 2.4 | | | V |
| | High-level output voltage (1.8V I/O) | $DV_{DD18}$ = MIN, $I_{OH}$ = MAX | $DV_{DD}$ - 0.45 | | | V |
| $V_{OL}$ | Low/full speed: USB_DN and USB_DP | | 0.0 | | 0.3 | V |
| | High speed: USB_DN and USB_DP | | -10 | | 10 | mV |
| | Low-level output voltage (3.3V I/O) | $DV_{DD33}$ = MIN, $I_{OL}$ = MAX | | | 0.4 | V |
| | Low-level output voltage (1.8V I/O) | $DV_{DD18}$ = MIN, $I_{OL}$ = MAX | | | 0.45 | V |
| $I_I$ (2) | Input current | $V_I = V_{SS}$ to $DV_{DD}$ without opposing internal resistor | | | ±10 | µA |
| | | $V_I = V_{SS}$ to $DV_{DD}$ with opposing internal pullup resistor (3) | 50 | 100 | 250 | µA |
| | | $V_I = V_{SS}$ to $DV_{DD}$ with opposing internal pulldown resistor (3) | -250 | -100 | -50 | µA |
| $I_{OH}$ | High-level output current | All peripherals | | | -4 | mA |
| $I_{OL}$ | Low-level output current | All peripherals | | | 4 | mA |
| $I_{OZ}$ (4) | I/O Off-state output current | $V_O = DV_{DD}$ or $V_{SS}$; internal pull disabled | | | ±20 | µA |
| | | $V_O = DV_{DD}$ or $V_{SS}$; internal pull enabled | | ±100 | | µA |
| $I_{CDD}$ | Core ($CV_{DD}$, $V_{DDA1P1V}$, $USB\_V_{DDA1P2LDO}$ (5), $CV_{DDDSP}$) supply current(6) | $CV_{DD}$ = 1.2 V, DSP clock = 594 MHz | | 767 | | mA |
| $I_{DDD}$ | 3.3V I/O ($DV_{DD33}$, $USB\_V_{DDA3P3}$) supply current(6) | $DV_{DD}$ = 3.3 V, DSP clock = 594 MHz | | 6 | | mA |
| $I_{DDD}$ | 1.8V I/O ($DV_{DD18}$, $DV_{DDR2}$, $DDR\_V_{DDDLL}$, $PLLV_{DD18}$, $V_{DDA1P8V}$, $USB\_V_{DD1P8}$, MXVDD, M24VDD) supply current(6) | $DV_{DD}$ = 1.8 V, DSP clock = 594 MHz | | 102 | | mA |
| $C_I$ | Input capacitance | | | | 4 | pF |
| $C_o$ | Output capacitance | | | | 4 | pF |

(1) For test conditions shown as MIN, MAX, or NOM, use the appropriate value specified in Section 5.2, *Recommended Operating Conditions*.
(2) $I_I$ applies to input-only pins and bi-directional pins. For input-only pins, $I_I$ indicates the input leakage current. For bi-directional pins, $I_I$ indicates the input leakage current and off-state (Hi-Z) output leakage current.
(3) Applies only to pins with an internal pullup (IPU) or pulldown (IPD) resistor.
(4) $I_{OZ}$ applies to output-only pins, indicating off-state (Hi-Z) output leakage current.
(5) This pin is an internal LDO output and connected via 1 µF capacitor to $USB\_V_{SSA1P2LDO}$.
(6) Measured under the following conditions: 60% DSP CPU utilization; ARM doing typical activity (peripheral configurations, other housekeeping activities); DDR2 Memory Controller at 50% utilization (135 MHz), 50% writes, 32 bits, 50% bit switching; 2 MHz ASP at 100% utilization; Timer0 at 100% utilization. At room temperature (25°C) for typical process devices. The actual current draw varies across manufacturing processes and is highly application-dependent. For more details on core and I/O voltages, activities and typical currents for -594, A-513, and -810 devcies, as well as information relevant to board power supply design, see the *TMS320DM6446/3 Power Consumption Summary* application report (literature number SPRAAD6).

# 6 Peripheral and Electrical Specifications

## 6.1 Parameter Information

### *6.1.1 Parameter Information Device-Specific Information*




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

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

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

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

#### 6.1.1.1 Signal Transition Levels

All input and output timing parameters are referenced to $V_{ref}$ for both "0" and "1" logic levels. For 3.3 V I/O, $V_{ref}$ = 1.5 V. For 1.8 V I/O, $V_{ref}$ = 0.9 V.

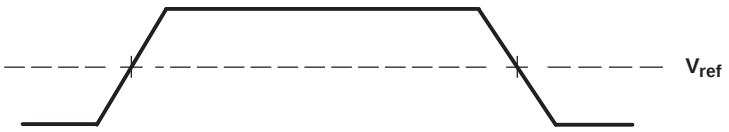



**Figure 6-2. Input and Output Voltage Reference Levels for AC Timing Measurements**

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

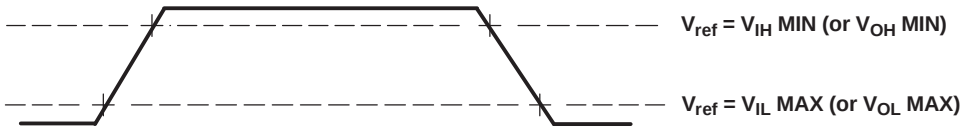



**Figure 6-3. Rise and Fall Transition Time Voltage Reference Levels**

#### 6.1.1.2 Timing Parameters and Board Routing Analysis

The timing parameter values specified in this data manual 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 the DDR2 memory controller interface, it is *not* necessary to use the IBIS models to analyze timing characteristics. TI provides a PCB routing rules solution that describes the routing rules to ensure the DDR2 memory controller interface timings are met. See the *Implementing DDR2 PCB Layout on the TMS320DM644x DSP* Application Report (literature number SPRAAC5).

## 6.2 Recommended Clock and Control Signal Transition Behavior

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

## 6.3 Power Supplies

For more information regarding TI's power management products and suggested devices to power TI DSPs, visit www.ti.com/dsppower.

### *6.3.1 Power-Supply Sequencing*

The DM6446 includes two core supplies — $CV_{DD}$ and $CV_{DDDSP}$, as well as three I/O supplies — $DV_{DD18}$, $DV_{DDR2}$, and $DV_{DD33}$. To ensure proper device operation, a specific power-up sequence *must* be followed.

The core supply power-up sequence is dependent on the DSP boot mode selected at reset. If the DSP boot mode is configured as Self-Boot mode, then both core supplies *must* be powered up at the same time.

If the DSP boot mode is configured as Host-Boot, where the ARM boots the DSP, the two core supplies may be ramped simultaneously or powered up separately. When powered up separately, the $CV_{DDDSP}$ supply *must not* be ramped prior to the $CV_{DD}$ supply. The $CV_{DDDSP}$ supply *must* be powered up before the shorting switch is closed (enabled). Prior to powering up the $CV_{DDDSP}$ supply, it should be left floating and not driven to ground. Table 6-1 and Figure 6-4 describe the power-on sequence timing requirements for DSP Host-Boot mode.

To minimize the voltage difference between these two core supplies, a single regulator source *must* be used to power the $CV_{DD}$ and $CV_{DDDSP}$ supplies.

For more information, see Section 3.2.1, *Power Configurations at Reset*.

**Table 6-1. Core Supply Power-On Timing Requirements for DSP Host-Boot Mode (see Figure 6-4)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(CVDD-CVDDDSP)}$ | Delay time, $CV_{DD}$ supply ready to $CV_{DDDSP}$ supply ramp start | 0 | (1) | ns |

(1) In Host-Boot mode, the $CV_{DDDSP}$ supply *must* be powered up prior to closing (enabling) the shorting switch between the ALWAYS ON and DSP power domains.

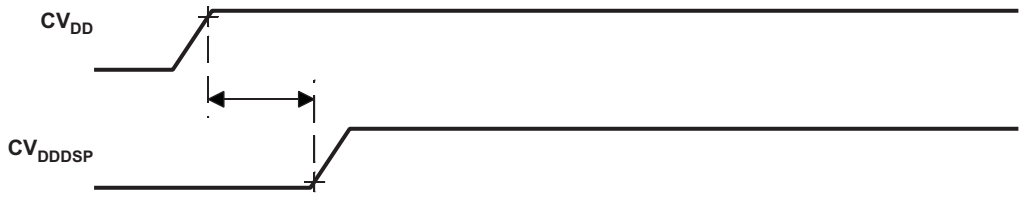
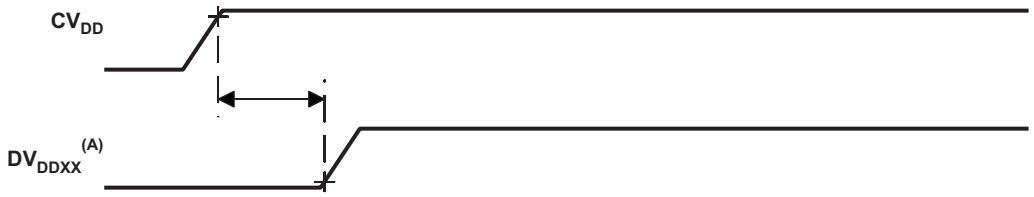



**Figure 6-4. DSP Host-Boot Mode Core Supply Timings**

Once the $CV_{DD}$ supply has been powered up, the I/O supplies may be powered up. Table 6-2 and Figure 6-5 show the power-on sequence timing requirements for the Core vs. I/O power-up. $DV_{DDXX}$ is used to denote all I/O supplies. Note: the $DV_{DDXX}$ supply power-up is specified relative to the $CV_{DD}$ supply power-up, not the $CV_{DDDSP}$ supply.

**Table 6-2. I/O Supply Power-On Timing Requirements (see Figure 6-5)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(CVDD-DVDD)}$ | Delay time, $CV_{DD}$ supply ready to $DV_{DDXX}$ supply ramp start | 0 | 100 | ms |




Note A: $DV_{DDXX}$ denotes all I/O supplies.

**Figure 6-5. I/O Supply Timings**

There is ***not*** a specific power-up sequence that must be followed with respect to the order of the power-up of the $DV_{DD18}$, $DV_{DDR2}$, and $DV_{DD33}$ supplies. Once the $CV_{DD}$ supply is powered up and the $t_{d(CVDD-DVDDXX)}$ specification is met, the $DV_{DD18}$, $DV_{DDR2}$, and $DV_{DD33}$ supplies may be powered up in any order of preference. All other supplies may also be powered up in any order of preference once the $t_{d(CVDD-DVDDXX)}$ specification has been met.

#### 6.3.1.1 Power-Supply Design Considerations

Core and I/O supply voltage regulators should be located close to the DSP (or DSP array) to minimize inductance and resistance in the power delivery path. Additionally, when designing for high-performance applications utilizing the DM6446 device, the PC board should include separate power planes for core, I/O, and ground, all bypassed with high-quality low-ESL/ESR capacitors.

#### 6.3.1.2 Power-Supply Decoupling

In order to properly decouple the supply planes from system noise, place as many capacitors (caps) as possible close to DM6446. Assuming 0603 caps, the user should be able to fit a total of 60 caps, 30 for the core supplies and 30 for the I/O supplies. These caps need to be close to the DM6446 power pins, no more than 1.25 cm maximum distance to be effective. Physically smaller caps, such as 0402, are better because of their lower parasitic inductance. Proper capacitance values are also important. Small bypass caps (near 560 pF) should be closest to the power pins. Medium bypass caps (220 nF or as large as can be obtained in a small package) should be next closest. TI recommends no less than 8 small and 8 medium caps per supply be placed immediately next to the BGA vias, using the "interior" BGA space and at least the corners of the "exterior".

Larger caps for each supply can be placed further away for bulk decoupling. Large bulk caps (on the order of 100 μF) should be furthest away, but still as close as possible. Large caps for each supply should be placed outside of the BGA footprint.

Any cap selection needs to be evaluated from a yield/manufacturing point-of-view. As with the selection of any component, verification of capacitor availability over the product's production lifetime should be considered.

#### 6.3.1.3 DM6446 Power and Clock Domains

DM6446 includes two separate power domains: "Always On" and "DSP". The "Always On" power domain is always on when the chip is on. The "Always On" domain is powered by the $V_{DD}$ pins of the DM6446. The majority of the DM6446's modules lie within the "Always On" power domain. A separate domain called the "DSP" domain houses the C64x+ and VICP. The "DSP" domain is not always on. The "DSP" power domain is powered by the $CV_{DDDSP}$ pins of the DM6446. Table 6-3 provides a listing of the DM6446 power and clock domains.

Two primary reference clocks are required for the DM6446 device. These can either be crystal input or driven by external oscillators. A 27-MHz crystal is recommended for the system PLLs, which generate the internal clocks for the ARM, DSP, coprocessors, peripherals (including imaging peripherals), and EDMA3. The recommended 27-MHz input enables the use of the video DACs to drive NTSC/PAL television signals at the proper frequencies. A 24-MHz crystal is also required if the USB peripheral is to be used. For further description of the DM6446 clock domains, see Table 6-4 (DM6446 Clock Domains) and Figure 6-6 (PLL1 and PLL2 Clock Domain Block Diagram).

**Table 6-3. DM6446 Power and Clock Domains**

| POWER DOMAIN | CLOCK DOMAIN | PERIPHERAL/MODULE |
|---|---|---|
| Always On | CLKIN | UART0 |
| Always On | CLKIN | UART1 |
| Always On | CLKIN | UART2 |
| Always On | CLKIN | I2C |
| Always On | CLKIN | Timer0 |
| Always On | CLKIN | Timer1 |
| Always On | CLKIN | Timer2 |
| Always On | CLKIN | PWM0 |
| Always On | CLKIN | PWM1 |
| Always On | CLKIN | PWM2 |
| Always On | CLKDIV2 | ARM Subsystem |
| Always On | CLKDIV3 | DDR2 |
| Always On | CLKDIV3 | VPSS |
| Always On | CLKDIV3 | EDMA3 |
| Always On | CLKDIV3 | SCR |
| Always On | CLKDIV6 | GPSC |
| Always On | CLKDIV6 | LPSCs |
| Always On | CLKDIV6 | Ice Pick |
| Always On | CLKDIV6 | EMIFA |
| Always On | CLKDIV6 | USB |
| Always On | CLKDIV6 | HPI |
| Always On | CLKDIV6 | VLYNQ |
| Always On | CLKDIV6 | EMAC |
| Always On | CLKDIV6 | ATA/CF |
| Always On | CLKDIV6 | MMC/SD/SDIO |
| Always On | CLKDIV6 | SPI |
| Always On | CLKDIV6 | ASP |
| Always On | CLKDIV6 | GPIO |
| DSP | CLKDIV1 | C64x+ CPU |

**Table 6-4. DM6446 Clock Domains(1)**

| SUBSYSTEM | FIXED RATIO vs. PLL1 | CLOCK MODES (FREQUENCY) | | | |
|---|---|---|---|---|---|
| | | PLL BYPASS | PLL ENABLED ( A-513) | PLL ENABLED ( -594) | PLL ENABLED (-810) |
| PLL1 | – | 27 MHz | 513 MHz | 594 MHz | 810 MHz |
| DSP | 1:1 | 27 MHz | 513 MHz | 594 MHz | 810 MHz |
| ARM | 1:2 | 13.5 MHz | 256.5 MHz | 297 MHz | 405 MHz |
| EDMA3/VPSS | 1:3 | 9 MHz | 171 MHz | 198 MHz | 270 MHz |
| Peripherals | 1:6 | 4.5 MHz | 85.5 MHz | 99 MHz | 135 MHz |

(1) These table values assume a MXI/CLKIN of 27 MHz and a PLL1 multiplier equal to 22 for both A-513 and -594 devices. For -810 device, these table values assume a MXI/CLKIN of 27 MHz and a PLL1 multiplier equal to 30.

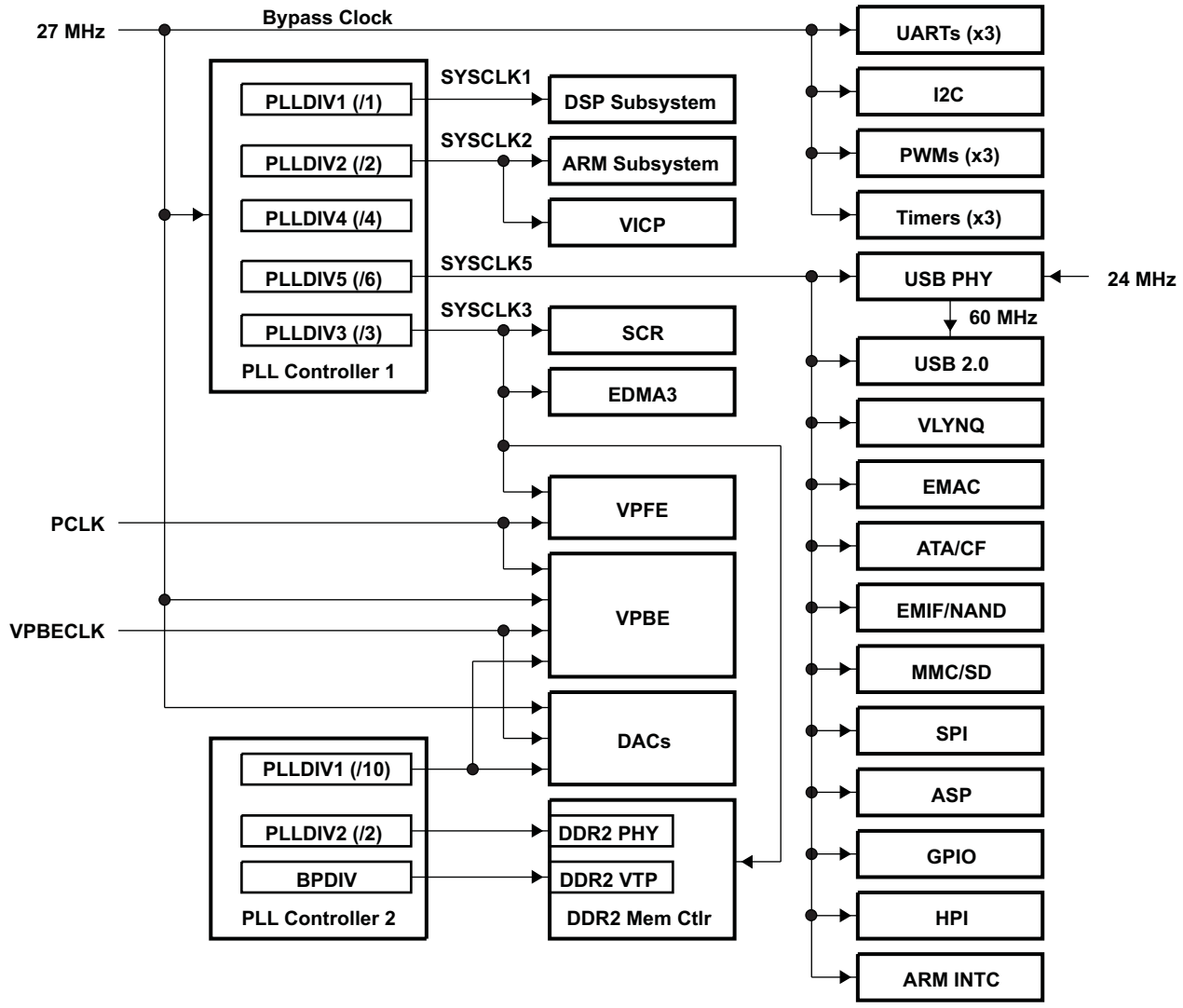



**Figure 6-6. PLL1 and PLL2 Clock Domain Block Diagram**

For further detail on PLL1 and PLL2, see the structure block diagrams Figure 6-7 and Figure 6-8, respectively.

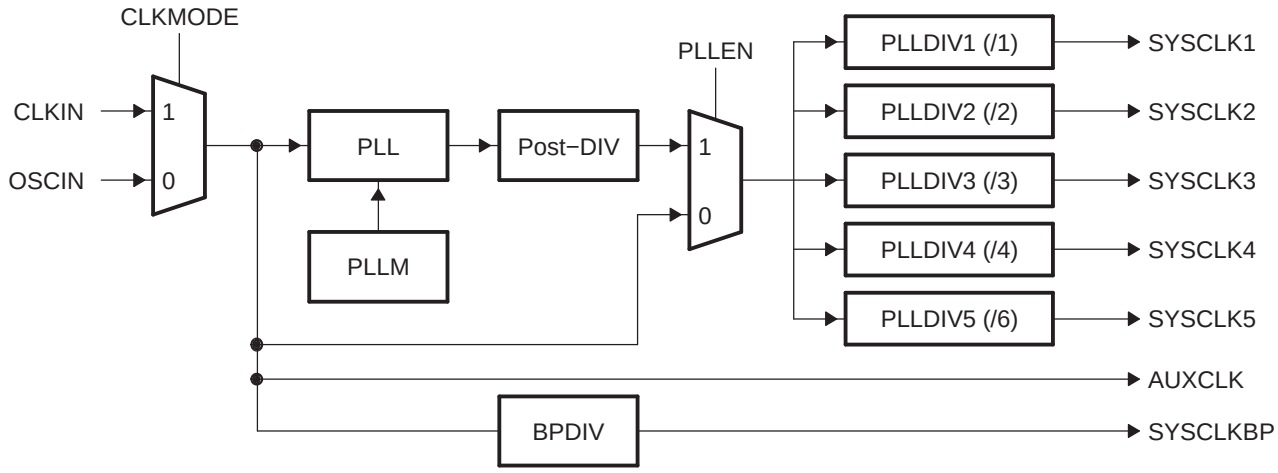



**Figure 6-7. PLL1 Structure Block Diagram**

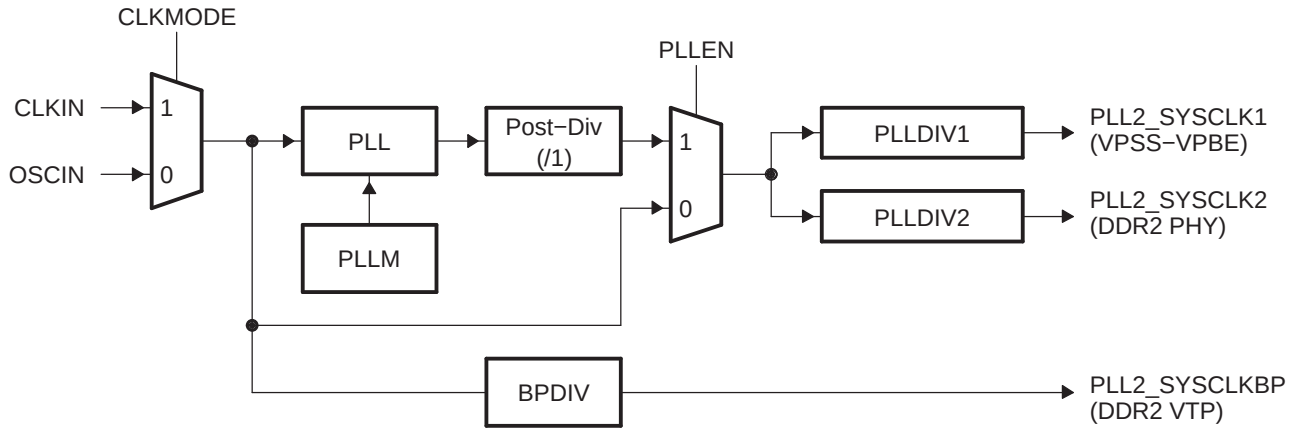



**Figure 6-8. PLL2 Structure Block Diagram**

### 6.3.1.4 Power and Sleep Controller (PSC) Module

The Power and Sleep Controller (PSC) controls DM6446 device power by turning off unused power domains or gating off clocks to individual peripherals/modules. The PSC consists of a Global PSC (GPSC) and a set of Local PSCs (LPSCs). The GPSC contains memory mapped registers, power domain control, PSC interrupt control, and a state machine for each peripheral/module. An LPSC is associated with each peripheral/module and provides clock and reset control. The GPSC controls all of DM6446's LPSCs. The ARM subsystem does not have an LPSC module. ARM sleep mode is accomplished through the wait for interrupt instruction. The LPSCs for DM6446 are shown in Table 6-5. The PSC register memory map is given in Table 6-6. For more details on the PSC, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

**Table 6-5. DM6446 LPSC Assignments**

| LPSC NUMBER | PERIPHERAL/MODULE | LPSC NUMBER | PERIPHERAL/MODULE | LPSC NUMBER | PERIPHERAL/MODULE |
|---|---|---|---|---|---|
| 0 | VPSS DMA | 14 | EMIFA | 28 | TIMER1 |
| 1 | VPSS MMR | 15 | MMC/SD/SDIO | 29 | Reserved |
| 2 | EDMA3CC | 16 | Reserved | 30 | Reserved |
| 3 | EDMA3TC0 | 17 | ASP | 31 | Reserved |
| 4 | EDMA3TC1 | 18 | I2C | 32 | Reserved |
| 5 | EMAC | 19 | UART0 | 33 | Reserved |
| 6 | EMAC Memory Controller | 20 | UART1 | 34 | Reserved |
| 7 | MDIO | 21 | UART2 | 35 | Reserved |
| 8 | Reserved | 22 | SPI | 36 | Reserved |
| 9 | USB | 23 | PWM0 | 37 | Reserved |
| 10 | ATA/CF | 24 | PWM1 | 38 | Reserved |
| 11 | VLYNQ | 25 | PWM2 | 39 | C64x+ CPU |
| 12 | HPI | 26 | GPIO | 40 | VICP |
| 13 | DDR2 Memory Controller | 27 | TIMER0 | | |

**Table 6-6. PSC Register Memory Map**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C4 1000 | PID | Peripheral Revision and Class Information Register |
| 0x01C4 1004 - 0x01C4 1014 | - | Reserved |
| 0x01C4 1018 | INTEVAL | Interrupt Evaluation Register |
| 0x01C4 101C - 0x01C4 103F | - | Reserved |
| 0x01C4 1040 | MERRPR0 | Module Error Pending 0 (mod 0 - 31) Register |
| 0x01C4 1044 | MERRPR1 | Module Error Pending 1 (mod 32- 63) Register |
| 0x01C4 1048 - 0x01C4 104F | - | Reserved |
| 0x01C4 1050 | MERRCR0 | Module Error Clear 0 (mod 0 - 31) Register |
| 0x01C4 1054 | MERRCR1 | Module Error Clear 1 (mod 32 - 63) Register |
| 0x01C4 1058 - 0x01C4 105F | - | Reserved |
| 0x01C4 1060 | PERRPR | Power Error Pending Register |
| 0x01C4 1064 - 0x01C4 1067 | - | Reserved |
| 0x01C4 1068 | PERRCR | Power Error Clear Register |
| 0x01C4 106C - 0x01C4 106F | - | Reserved |
| 0x01C4 1070 | EPCPR | External Power Error Pending Register |
| 0x01C4 1074 - 0x01C4 1077 | - | Reserved |
| 0x01C4 1078 | EPCCR | External Power Control Clear Register |
| 0x01C4 107C - 0x01C4 111F | - | Reserved |
| 0x01C4 1120 | PTCMD | Power Domain Transition Command Register |
| 0x01C4 1124 - 0x01C4 1127 | - | Reserved |
| 0x01C4 1128 | PTSTAT | Power Domain Transition Status Register |
| 0x01C4 112C - 0x01C4 11FF | - | Reserved |
| 0x01C4 1200 | PDSTAT0 | Power Domain Status 0 Register (Always On) |
| 0x01C4 1204 | PDSTAT1 | Power Domain Status 1 Register (DSP) |
| 0x01C4 1208 - 0x01C4 12FF | - | Reserved |
| 0x01C4 1300 | PDCTL0 | Power Domain Control 0 Register (Always On) |
| 0x01C4 1304 | PDCTL1 | Power Domain Control 1 Register (DSP) |
| 0x01C4 1308 - 0x01C4 17FF | - | Reserved |
| 0x01C4 1800 | MDSTAT0 | Module Status 0 Register (VPSS DMA) |
| 0x01C4 1804 | MDSTAT1 | Module Status 1 Register (VPSS MMR) |
| 0x01C4 1808 | MDSTAT2 | Module Status 2 Register (EDMA3CC) |
| 0x01C4 180C | MDSTAT3 | Module Status 3 Register (EDMA3TC0) |
| 0x01C4 1810 | MDSTAT4 | Module Status 4 Register (EDMA3TC1) |
| 0x01C4 1814 | MDSTAT5 | Module Status 5 Register (EMAC) |
| 0x01C4 1818 | MDSTAT6 | Module Status 6 Register (EMAC Memory Controller) |
| 0x01C4 181C | MDSTAT7 | Module Status 7 Register (MDIO) |
| 0x01C4 1820 | | Reserved |
| 0x01C4 1824 | MDSTAT9 | Module Status 9 Register (USB) |
| 0x01C4 1828 | MDSTAT10 | Module Status 10 Register (ATA/CF) |
| 0x01C4 182C | MDSTAT11 | Module Status 11 Register (VLYNQ) |
| 0x01C4 1830 | MDSTAT12 | Module Status 12 Register (HPI) |
| 0x01C4 1834 | MDSTAT13 | Module Status 13 Register (DDR2) |
| 0x01C4 1838 | MDSTAT14 | Module Status 14 Register (EMIFA) |
| 0x01C4 183C | MDSTAT15 | Module Status 15 Register (MMC/SD/SDIO) |
| 0x01C4 1840 | | Reserved |
| 0x01C4 1844 | MDSTAT17 | Module Status 17 Register (ASP) |
| 0x01C4 1848 | MDSTAT18 | Module Status 18 Register (I2C) |

**Table 6-6. PSC Register Memory Map (continued)**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C4 184C | MDSTAT19 | Module Status 19 Register (UART0) |
| 0x01C4 1850 | MDSTAT20 | Module Status 20 Register (UART1) |
| 0x01C4 1854 | MDSTAT21 | Module Status 21 Register (UART2) |
| 0x01C4 1858 | MDSTAT22 | Module Status 22 Register (SPI) |
| 0x01C4 185C | MDSTAT23 | Module Status 23 Register (PWM0) |
| 0x01C4 1860 | MDSTAT24 | Module Status 24 Register (PWM1) |
| 0x01C4 1864 | MDSTAT25 | Module Status 25 Register (PWM2) |
| 0x01C4 1868 | MDSTAT26 | Module Status 26 Register (GPIO) |
| 0x01C4 186C | MDSTAT27 | Module Status 27 Register (TIMER0) |
| 0x01C4 1870 | MDSTAT28 | Module Status 28 Register (TIMER1) |
| 0x01C4 1874 - 0x01C4 189B | - | Reserved |
| 0x01C4 189C | MDSTAT39 | Module Status 39 Register (C64x+ CPU) |
| 0x01C4 18A0 | MDSTAT40 | Module Status 40 Register (VICP) |
| 0x01C4 18A4 - 0x01C4 19FF | - | Reserved |
| 0x01C4 1A00 | MDCTL0 | Module Control 0 Register (VPSS DMA) |
| 0x01C4 1A04 | MDCTL1 | Module Control 1 Register (VPSS MMR) |
| 0x01C4 1A08 | MDCTL2 | Module Control 2 Register (EDMA3CC) |
| 0x01C4 1A0C | MDCTL3 | Module Control 3 Register (EDMA3TC0) |
| 0x01C4 1A10 | MDCTL4 | Module Control 4 Register (EDMA3TC1) |
| 0x01C4 1A14 | MDCTL5 | Module Control 5 Register (EMAC) |
| 0x01C4 1A18 | MDCTL6 | Module Control 6 Register (EMAC Memory Controller) |
| 0x01C4 1A1C | MDCTL7 | Module Control 7 Register (MDIO) |
| 0x01C4 1A20 | | Reserved |
| 0x01C4 1A24 | MDCTL9 | Module Control 9 Register (USB) |
| 0x01C4 1A28 | MDCTL10 | Module Control 10 Register (ATA/CF) |
| 0x01C4 1A2C | MDCTL11 | Module Control 11 Register (VLYNQ) |
| 0x01C4 1A30 | MDCTL12 | Module Control 12 Register (HPI) |
| 0x01C4 1A34 | MDCTL13 | Module Control 13 Register (DDR2) |
| 0x01C4 1A38 | MDCTL14 | Module Control 14 Register (EMIFA) |
| 0x01C4 1A3C | MDCTL15 | Module Control 15 Register (MMC/SD/SDIO) |
| 0x01C4 1A40 | | Reserved |
| 0x01C4 1A44 | MDCTL17 | Module Control 17 Register (ASP) |
| 0x01C4 1A48 | MDCTL18 | Module Control 18 Register (I2C) |
| 0x01C4 1A4C | MDCTL19 | Module Control 19 Register (UART0) |
| 0x01C4 1A50 | MDCTL20 | Module Control 20 Register (UART1) |
| 0x01C4 1A54 | MDCTL21 | Module Control 21 Register (UART2) |
| 0x01C4 1A58 | MDCTL22 | Module Control 22 Register (SPI) |
| 0x01C4 1A5C | MDCTL23 | Module Control 23 Register (PWM0) |
| 0x01C4 1A60 | MDCTL24 | Module Control 24 Register (PWM1) |
| 0x01C4 1A64 | MDCTL25 | Module Control 25 Register (PWM2) |
| 0x01C4 1A68 | MDCTL26 | Module Control 26 Register (GPIO) |
| 0x01C4 1A6C | MDCTL27 | Module Control 27 Register (TIMER0) |
| 0x01C4 1A70 | MDCTL28 | Module Control 28 Register (TIMER1) |
| 0x01C4 1A74 - 0x01C4 1A9B | - | Reserved |
| 0x01C4 1A9C | MDCTL39 | Module Control 39 Register (C64x+ CPU) |
| 0x01C4 1AA0 | MDCTL40 | Module Control 40 Register (VICP) |
| 0x01C4 1AA4 - 0x01C4 1FFF | - | Reserved |

**Table 6-6. PSC Register Memory Map (continued)**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C4 1000 | MPFAR | Memory Protection Fault Address Register |
| 0x01C4 1004 | MPFSR | Memory Protection Fault Status Register |
| 0x01C4 1008 | MPFCR | Memory Protection Fault Command Register |
| 0x01C4 100C | MPAA | Memory Protection Page Attribute Register |
| 0x01C4 1010 - 0x01C4 1FFF | - | Reserved |

## 6.4 Reset

DM6446 supports various types of resets. Power-on-reset (POR), warm reset, max reset, system reset, C64x+ local reset, and module reset are summarized in Table 6-7.

**Table 6-7. DM6446 Resets**

| Type | Initiator | Description |
|---|---|---|
| Power-on-reset (POR) | $\overline{\text{RESET}}$ pin active low while $\overline{\text{TRST}}$ is low. | Global chip reset (Cold reset). Activates the POR signal on chip, which is used to reset test and emulation logic. |
| Warm reset | $\overline{\text{RESET}}$ pin active low while $\overline{\text{TRST}}$ is high. | Resets everything except for test and emulation logic. ARM emulator stays alive during warm reset, but the C64x+ emulator does not. |
| Maximum reset | Emulator, WD Timer | Same as Warm reset, except for initiators. |
| C64x+ Local reset | Software (register bit) | MMR controls the C64x+ reset input. This is used for control of C64x+ reset by the ARM. The C64x+ Slave DMA port is still alive when in local reset. |

Power-on-reset (POR) is the global chip reset and it affects test, emulation, and other circuitry. It is invoked by driving the $\overline{\text{RESET}}$ pin active low while $\overline{\text{TRST}}$ is held low. A POR is required to place DM6446 into a known good initial state. POR can be asserted prior to ramping the core and I/O voltages or after the core and I/O voltages have reached their proper operating conditions. As a best practice, $\overline{\text{RESET}}$ should be asserted (held low) during power-up. Prior to deasserting $\overline{\text{RESET}}$ (low-to-high transition), the core and I/O voltages should be at their proper operating conditions and if an external 27 MHz oscillator is used on the MXI/CLKIN pin, the external clock should also be running at the correct frequency.

Warm reset is activated by driving the $\overline{\text{RESET}}$ pin active low, while $\overline{\text{TRST}}$ is inactive high. This does not reset test or ARM emulation logic. An ARM emulator session will stay alive during warm reset, but a C64x+ emulator session will not.

Maximum reset is initiated by the emulator or the watchdog timer and the reset effects are the same as a warm reset. The emulator initiates a maximum reset via the ICEPICK module. When the watchdog timer counter reaches zero, this will initiate a maximum reset to recover from a runaway condition. Both of the maximum reset initiators can be masked by the ARM emulator.

System reset is initiated by the emulator and is a soft reset. Memory contents are maintained. Test, emulation, clock, and power control logic are unaffected. The emulator initiates a system reset via the C64x+ emulation logic, or through ICECRUSHER. Both of these reset initiators are non-maskable resets.

The C64x+ DSP has an internal reset input that allows a host to control it. This reset is configured through a MMR bit (MDCTL[39].LRSTz) in the PSC module. When in C64x+ local reset, the slave DMA port on C64x+ will remain active and the internal memory will be accessible, including access to the VICP memory through the L2 port (UMAP port).

For details on reset control/status registers, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

For information on peripheral selection at the rising edge of $\overline{\text{RESET}}$, see Section 3, *Device Configurations*, of this data manual.

### 6.4.1 Reset Electrical Data/Timing

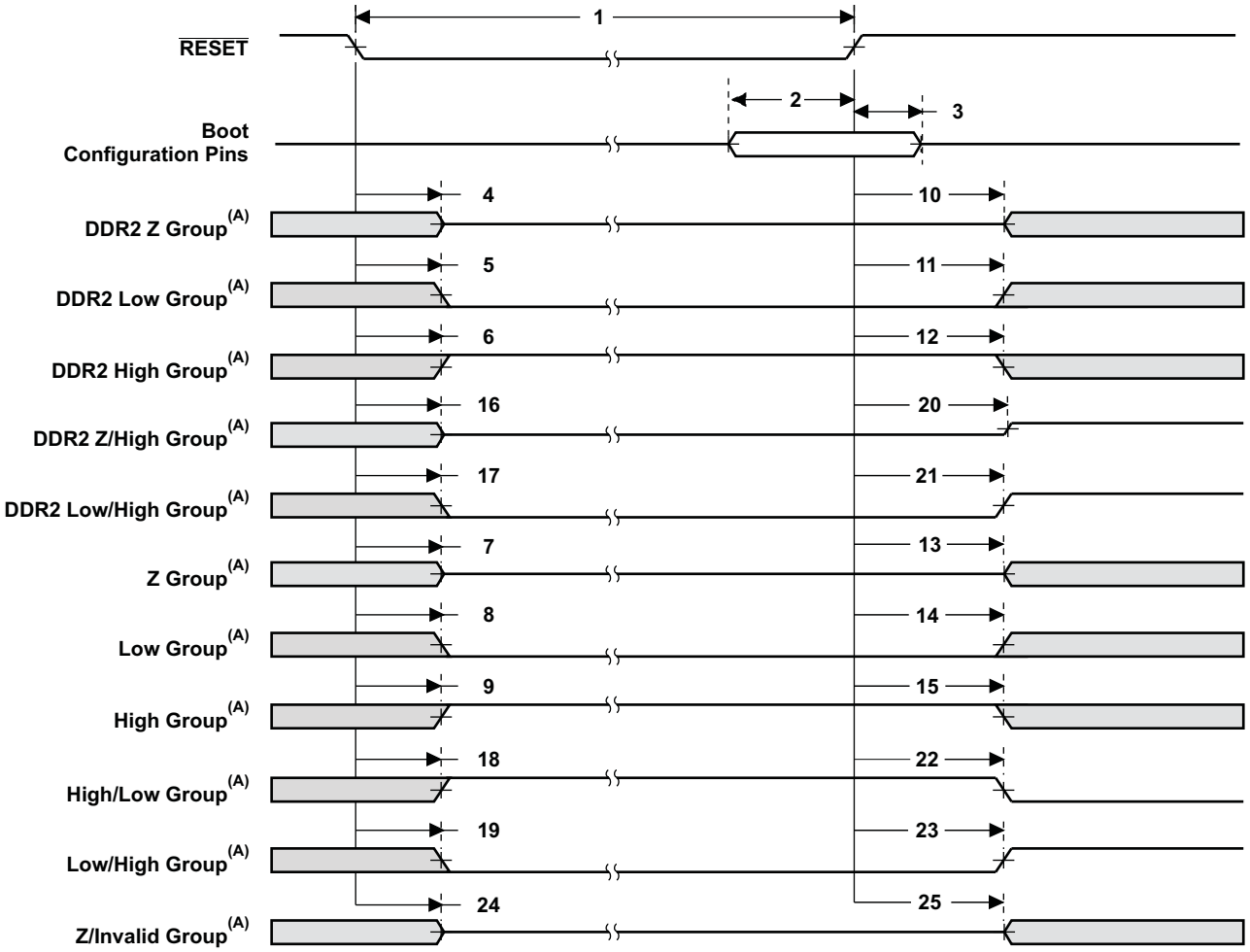
**Table 6-8. Timing Requirements for Reset (see Figure 6-9)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(RST)}$ | Width of the $\overline{\text{RESET}}$ pulse | 444 | | ns |
| 2 | $t_{su(BOOT)}$ | Setup time, boot configuration bits valid before $\overline{\text{RESET}}$ high | 444 | | ns |
| 3 | $t_{h(BOOT)}$ | Hold time, boot configuration bits valid after $\overline{\text{RESET}}$ high | 444 | | ns |

**Table 6-9. Switching Characteristics Over Recommended Operating Conditions During Reset(1) (see Figure 6-9)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 26 | $t_{d(PLL\_LOCK)}$ | Delay time, PLL1 lock time | | 2000P | ns |
| 4 | $t_{d(RSTL-DDRZZ)}$ | Delay time, $\overline{\text{RESET}}$ low to DDR2 Z Group high impedance | 0 | 2P + 20 | ns |
| 5 | $t_{d(RSTL-DDRLL)}$ | Delay time, $\overline{\text{RESET}}$ low to DDR2 Low Group low | 0 | 20 | ns |
| 6 | $t_{d(RSTL-DDRHH)}$ | Delay time, $\overline{\text{RESET}}$ low to DDR2 High Group high | 0 | 20 | ns |
| 16 | $t_{d(RSTL-DDRZHZ)}$ | Delay time, $\overline{\text{RESET}}$ low to DDR2 Z/High Group high impedance | 0 | 5P + 20 | ns |
| 17 | $t_{d(RSTL-DDRLHL)}$ | Delay time, $\overline{\text{RESET}}$ low to DDR2 Low/High Group low | 0 | 20 | ns |
| 7 | $t_{d(RSTL-ZZ)}$ | Delay time, $\overline{\text{RESET}}$ low to Z Group high impedance | 0 | 20 | ns |
| 8 | $t_{d(RSTL-LOWL)}$ | Delay time, $\overline{\text{RESET}}$ low to Low Group low | 0 | 20 | ns |
| 9 | $t_{d(RSTL-HIGHH)}$ | Delay time, $\overline{\text{RESET}}$ low to High Group high | 0 | 20 | ns |
| 18 | $t_{d(RSTL-HIGHLOWH)}$ | Delay time, $\overline{\text{RESET}}$ low to High/Low Group high | 0 | 20 | ns |
| 19 | $t_{d(RSTL-LOWHIGHL)}$ | Delay time, $\overline{\text{RESET}}$ low to Low/High Group low | 0 | 20 | ns |
| 24 | $t_{d(RSTL-ZIZ)}$ | Delay time, $\overline{\text{RESET}}$ low to Z/Invalid Group high impedance | 0 | 20 | ns |
| 10 | $t_{d(RSTH-DDRZV)}$ | Delay time, $\overline{\text{RESET}}$ high to DDR2 Z Group valid | | (2) | ns |
| 11 | $t_{d(RSTH-DDRLV)}$ | Delay time, $\overline{\text{RESET}}$ high to DDR2 Low Group valid | | (2) | ns |
| 12 | $t_{d(RSTH-DDRHV)}$ | Delay time, $\overline{\text{RESET}}$ high to DDR2 High Group valid | | (2) | ns |
| 20 | $t_{d(RSTH-DDRZHV)}$ | Delay time, $\overline{\text{RESET}}$ high to DDR2 Z/High Group valid high | | 4000P | ns |
| 21 | $t_{d(RSTH-DDRLHV)}$ | Delay time, $\overline{\text{RESET}}$ high to DDR2 Low/High Group valid high | | 4000P | ns |
| 13 | $t_{d(RSTH-ZV)}$ | Delay time, $\overline{\text{RESET}}$ high to Z Group valid | | (2) | ns |
| 14 | $t_{d(RSTH-LOWV)}$ | Delay time, $\overline{\text{RESET}}$ high to Low Group valid | | (2) | ns |
| 15 | $t_{d(RSTH-HIGHV)}$ | Delay time, $\overline{\text{RESET}}$ high to High Group valid | | (2) | ns |
| 22 | $t_{d(RSTH-HIGHLOWV)}$ | Delay time, $\overline{\text{RESET}}$ high to High/Low Group valid low | | 5100P | ns |
| 23 | $t_{d(RSTH-LOWHIGHV)}$ | Delay time, $\overline{\text{RESET}}$ high to Low/High Group valid high | | 5100P | ns |
| 25 | $t_{d(RSTH-ZIIV)}$ | Delay time, $\overline{\text{RESET}}$ high to Z/Invalid Group invalid | | 4000P | ns |

(1) P = MXI/CLKIN cycle time, in ns.
(2) Following $\overline{\text{RESET}}$ high, this signal group maintains the state the pins(s) achieved while $\overline{\text{RESET}}$ was driven low until the peripheral is enabled via the PSC. For example, the DDR2 Z Group goes high impedance following $\overline{\text{RESET}}$ low and remains in the high-impedance state following $\overline{\text{RESET}}$ high until the DDR2 controller is enabled via the PSC.




A. **DDR2 Z Group:** DDR_DQS[3:0], DDR_D[12:0]
**DDR2 Low Group:** DDR_CLK0, DDR_CKE, DDR_A[12:0]
**DDR2 High Group:** $\overline{\text{DDR\_CLK0}}$, $\overline{\text{DDR\_CS}}$, $\overline{\text{DDR\_WE}}$, $\overline{\text{DDR\_RAS}}$, $\overline{\text{DDR\_CAS}}$
**DDR2 Z/High Group:** $\overline{\text{DDR\_DQM}}$[3:0]
**DDR2 Low/High Group:** DDR_BS[2:0]
**Low Group:** DMARQ/UART_RXD1, VCLK, RTCK, TDO, VPBECLK, YOUT0/G5/AEAW0, YOUT1/G6/AEAW1, YOUT2/G7/AEAW2, YOUT3/R3/AEAW3, YOUT4/R4/AEAW4, COUT3/B6/DSP_BT, COUT2/B5/EM_WIDTH, COUT1/B4/BTSEL1, COUT0/B3/BTSEL0, $\overline{\text{TRST}}$
**High Group:** DMACK/UART_TXD1, EM_A[2]/(CLE), EM_A[1]/(ALE), $\overline{\text{EM\_CS3}}$, $\overline{\text{EM\_WE}}$/($\overline{\text{WE}}$)/($\overline{\text{IOWR}}$)/$\overline{\text{DIOW}}$
**Z Group:** All other pins not listed above, with the exception of power and ground pins.
- The following Z Group pins have an internal pullup (IPU): DMARQ/UART_RXD1, VPBECLK, HSYNC, VSYNC, $\overline{\text{TRST}}$, YI/CCD[7:0], CI[3:0]/CCD[11:8], CI4/CCD12/UART_RTS2, CI5/CCD13/UART_CTS2, CI6/CCD14/UART_TXD2, CI7/CCD15/UART_RXD2
- The following Z Group pins have an internal pulldown (IPD): EM_WAIT/$\overline{\text{IORDY}}$, TCK, TDI, TMS, EMU[1:0]

**High/Low Group:** EM_BA[0]/DA0, $\overline{\text{EM\_CS2}}$, $\overline{\text{EM\_OE}}$/($\overline{\text{RE}}$)/($\overline{\text{IORD}}$)/$\overline{\text{DIOR}}$
**Low/High Group:** EM_R/$\overline{\text{W}}$/INTRQ
**Z/Invalid Group:** EM_D[15:0]

**Figure 6-9. Reset Timing**

## 6.5 External Clock Input From MXI/CLKIN Pin

The DM6446 device has two input pins for an external clock source, MXI/CLKIN and M24XI. The MXI/CLKIN pin provides the clock source for PLL1 and PLL2 whose optimal frequency is 27 MHz. The M24XI pin provides the clock source for the USB PLL whose optimal frequency is 24 MHz.

The DM6446 device includes two options to provide an external clock input:

1. Use an on-chip oscillator with external crystal or ceramic resonator circuit (only supporting parallel-resonant mode; it does not provide overtone support). For more details, see Section 6.5.1, *Clock Input Option 1 – Crystal*.
2. Use an external 1.8-V LVCMOS-compatible clock input. For more details, see Section 6.5.2, *Clock Input Option 2 – 1.8-V LVCMOS-Compatible Clock Input*.

### *6.5.1 Clock Input Option 1 – Crystal*

#### 6.5.1.1 27-MHz Crystal for System Oscillator

In this option, a crystal is used as the external clock input to the DM6446 PLL1 and PLL2.

The 27-MHz oscillator provides the reference clock for all DM6446 subsystems and peripherals. The on-chip oscillator requires an external 27-MHz crystal connected across the MXI and MXO pins, along with two load capacitors, as shown in Figure 6-10. The external crystal load capacitors ***must*** be connected only to the 27-MHz oscillator ground pin ($MXV_{SS}$). ***Do not*** connect to board ground ($V_{SS}$). The $MXV_{DD}$ pin can be connected to the same 1.8 V power supply as $DV_{DD18}$.

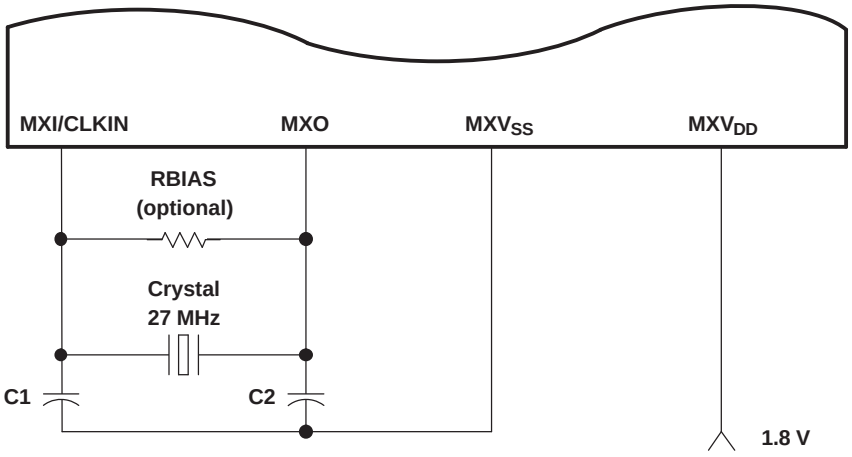



**Figure 6-10. 27-MHz System Oscillator**

The RBIAS resistor is optional. If the RBIAS resistor is used, it should equal 1 MΩ ±5%. The load capacitors, C1 and C2, should be chosen such that the equation is satisfied (typical values are C1 = C2 = 10 pF). $C_L$ in the equation is the load specified by the crystal manufacturer. All discrete components used to implement the oscillator circuit should be placed as close as possible to the associated oscillator pins (MXI and MXO) and to the $MXV_{SS}$ pin.

$$C_L = \frac{C_1 C_2}{(C_1 + C_2)}$$

**Table 6-10. Crystal Requirements for a 27-MHz System Oscillator**

| PARAMETER | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|
| Start-up time (from power up until oscillating at stable frequency of 27 MHz) | | | 4 | ms |
| Oscillation frequency | | 27 | | MHz |
| ESR | | | 60 | Ω |
| Frequency stability | | | ±50 | ppm |

### 6.5.1.2 24-MHz Crystal for USB Oscillator

In this option, a crystal is used as the external clock input to the DM6446 USB PLL.

The 24-MHz oscillator provides the reference clock for the DM6446 USB peripheral. The on-chip oscillator requires an external 24-MHz crystal connected across the M24XI and M24XO pins, along with two load capacitors, as shown in Figure 6-11.The external crystal load capacitors ***must*** be connected only to the 24-MHz oscillator ground pin ($M24V_{SS}$). ***Do not*** connect to board ground ($V_{SS}$).

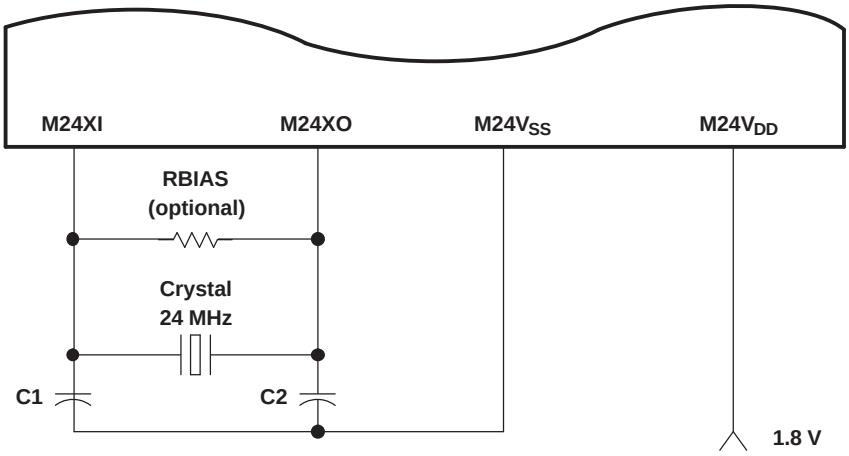



**Figure 6-11. 24-MHz USB Oscillator**

The RBIAS resistor is optional. If the RBIAS resistor is used, it should equal 1 MΩ ±5%. The load capacitors, C1 and C2, should be chosen such that the equation is satisfied (typical values are C1 = C2 = 10 pF). $C_L$ in the equation is the load specified by the crystal manufacturer. All discrete components used to implement the oscillator circuit should be placed as close as possible to the associated oscillator pins (M24XI and M24XO) and to the $M24XV_{SS}$ pin.

$$C_L = \frac{C_1 C_2}{(C_1 + C_2)}$$

**Table 6-11. Crystal Requirements for a 24-MHz USB Oscillator**

| PARAMETER | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|
| Start-up time (from power up until oscillating at stable frequency of 24 MHz) | | | 4 | ms |
| Oscillation frequency | | 24 | | MHz |
| ESR | | | 60 | Ω |
| Frequency stability | | | ±50 | ppm |

### 6.5.2 Clock Input Option 2 – 1.8-V LVCMOS-Compatible Clock Input

In this option, a 1.8-V LVCMOS-compatible clock input is used as the external clock input to the DM6446 device. The external connections are shown in Figure 6-12. The MXI/CLKIN pin is connected to the 1.8-V LVCMOS-compatible clock source. The MXO pin is left unconnected. The $MXV_{SS}$ pin is connected to board ground ($V_{SS}$). The $MXV_{DD}$ pin can be connected to the same 1.8-V power supply as $DV_{DD18}$. The clock source must meet the MXI/CLKIN timing requirements shown in Table 6-16, *Timing Requirements for MXI/CLKIN*.

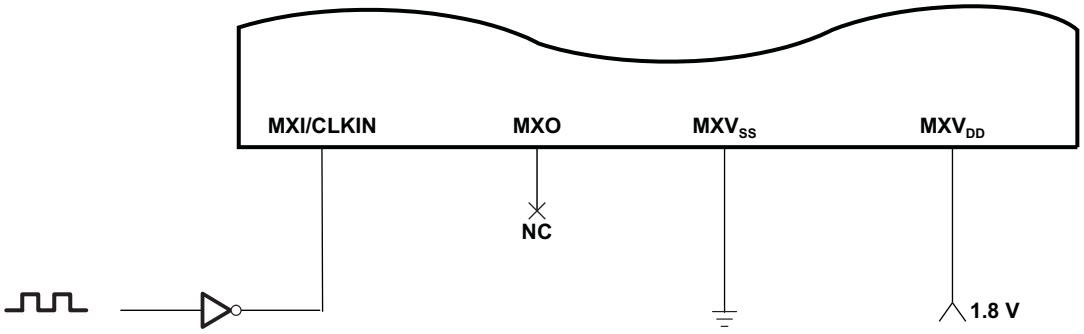



**Figure 6-12. 1.8-V LVCMOS-Compatible Clock Input**

Figure 6-12 also applies to the USB external clock input. When a 1.8-V LVCMOS-compatible clock input is used as the external clock input, the M24XI pin is connected to the 1.8-V LVCMOS-compatible clock source. The M24XO pin is left unconnected. The $M24V_{SS}$ pin is connected to board ground ($V_{SS}$). The $M24V_{DD}$ pin can be connected to the same 1.8-V power supply as $DV_{DDR2}$. The clock source must meet the MXI/CLKIN timing requirements shown in Table 6-17, *Timing Requirements for M24XI (-594) Devices*.

## 6.6 Clock PLLs

There are two independently controlled PLLs on DM6446. PLL1 generates the frequencies required for the DSP, ARM, VICP, DMA, VPFE, and other peripherals. PLL2 generates the frequencies required for the DDR2 interface and the VPBE in certain modes. The recommended reference clock for both PLLs is the 27-MHz crystal input. The USB2.0 PHY contains a third PLL embedded within it and the 24-MHz oscillator is its reference clock source. This particular PLL is only usable for USB operation, and is discussed further in the *TMS320DM644x DMSoC Universal Serial Bus (USB) Controller User's Guide* (literature number SPRUE35).

A summary of the PLL controller registers is shown in Table 6-12. For more details, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

**Table 6-12. PLL and Reset Controller Registers Memory Map**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| **PLL1 Controller Registers** | | |
| 0x01C4 0800 | PID | Peripheral Identification and Revision Information Register |
| 0x01C4 08E4 | RSTYPE | Reset Type Register |
| 0x01C4 0900 | PLLC | PLL Controller 1 Operations Control Register |
| 0x01C4 0910 | PLLM | PLL Controller 1 Multiplier Control Register |
| 0x01C4 0918 | PLLDIV1 | PLL Controller 1 Control-Divider 1 Register (SYSCLK1) |
| 0x01C4 091C | PLLDIV2 | PLL Controller 1 Control-Divider 2 Register (SYSCLK2) |
| 0x01C4 0920 | PLLDIV3 | PLL Controller 1 Control-Divider 3 Register (SYSCLK3) |
| 0x01C4 0928 | POSTDIV | PLL Controller 1 Post-Divider Control Register |
| 0x01C4 092C | BPDIV | PLL Controller 1 Bypass Control-Divider Register (SYSCLKBP) |
| 0x01C4 0938 | PLLCMD | PLL Controller 1 Command Register |
| 0x01C4 093C | PLLSTAT | PLL Controller 1 Status Register (Shows PLLCTRL Status) |

**Table 6-12. PLL and Reset Controller Registers Memory Map (continued)**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C4 0940 | ALNCTL | PLL Controller 1 Alignment Control Register (Indicates Which SYSCLKs Need to be Aligned for Proper Device Operation) |
| 0x01C4 0944 | DCHANGE | PLL Controller 1 Divider Change Register (Indicates if SYSCLK Divide Ratio has Been Modified) |
| 0x01C4 0948 | CKEN | PLL Controller 1 Clock Enable Register |
| 0x01C4 094C | CKSTAT | PLL Controller 1 Clock Status Register (For All Clocks Except SYSCLKx) |
| 0x01C4 0950 | SYSTAT | PLL Controller 1 System Clock Status 1 Register (Indicates SYSCLK on/off Status) |
| 0x01C4 0960 | PLLDIV4 | PLL Controller 1 Control-Divider 4 Register (SYSCLK4) |
| 0x01C4 0964 | PLLDIV5 | PLL Controller 1 Control-Divider 5 Register (SYSCLK5) |
| 0x01C4 0C00 | PID | Peripheral Identification and Revision Information Register |
| 0x01C4 0D00 | PLLC | PLL Controller 2 Operations Control Register |
| 0x01C4 0D10 | PLLM | PLL Controller 2 Multiplier Control Register |
| 0x01C4 0D18 | PLLDIV1 | PLL Controller 2 Control-Divider 1 Register (SYSCLK1) |
| 0x01C4 0D1C | PLLDIV2 | PLL Controller 2 Control-Divider 2 Register (SYSCLK2) |
| 0x01C4 0D20 - 0x01C4 0D2B | POSTDIV | PLL Controller 2 Post-Divider Control Register |
| 0x01C4 0D2C | BPDIV | PLL Controller 2 Bypass Control-Divider Register (SYSCLKBP) |
| 0x01C4 0D38 | PLLCMD | PLL Controller 2 Command Register |
| 0x01C4 0D3C | PLLSTAT | PLL Controller 2 Status Register (Shows PLLCTRL Status) |
| 0x01C4 0D40 | ALNCTL | PLL Controller 2 Alignment Control Register (Indicates Which SYSCLKs Need to be Aligned for Proper Device Operation) |
| 0x01C4 0D44 | DCHANGE | PLL Controller 2 Divider Change Register (Indicates if SYSCLK Divide Ratio has Been Modified) |
| 0x01C4 0D48 | CKEN | PLL Controller 2 Clock Enable Register |
| 0x01C4 0D4C | CKSTAT | PLL Controller 2 Clock Status Register (For All Clocks Except SYSCLKx) |
| 0x01C4 0D50 | SYSTAT | PLL Controller 2 System Clock Status 1 Register (Indicates SYSCLK on/off Status) |

### 6.6.1 PLL1 and PLL2

Both PLL1 and PLL2 power is supplied externally via the 1.8 V PLL power-supply pin ($PLLV_{DD18}$). It is recommended that an external EMI filter circuit be added to $PLLV_{DD18}$, as shown in Figure 6-13. The 1.8-V supply of the EMI filter must be from the same 1.8-V power plane supplying the device's 1.8-V I/O power-supply pins ($DV_{DD}$). TI recommends EMI filter manufacturer Murata, part number NFM18CC222R1C3.

All PLL external components (C1, C2, and the EMI Filter) ***should*** be placed as close to the 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 in Figure 6-13. For reduced PLL jitter, maximize the spacing between switching signals and the PLL external components (C1, C2, and the EMI Filter).

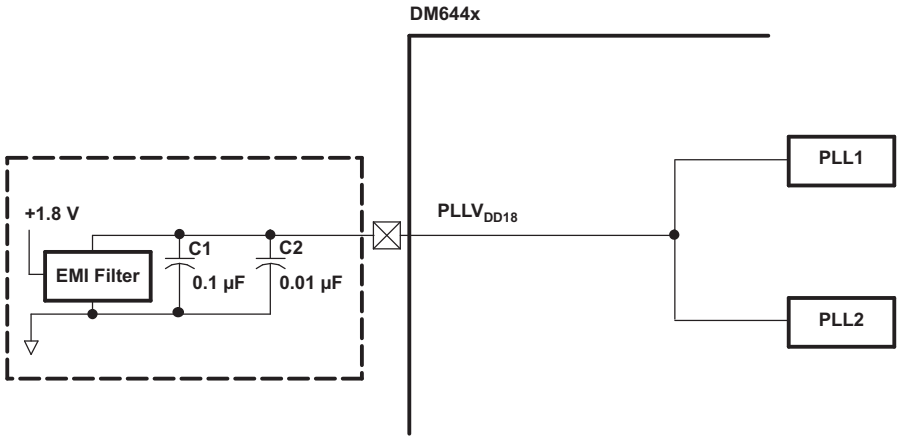



**Figure 6-13. PLL1 and PLL2 External Connection**

The minimum CLKIN rise and fall times should also be observed. For the input clock timing requirements, see Section 6.6.3, *Clock PLL Electrical Data/Timing (Input and Output Clocks)*.

There is an allowable range for PLL multiplier (PLLM). There is a minimum and maximum operating frequency for MXI/CLKIN, PLLOUT, and the device clocks (SYSCLKs). The PLL Controllers ***must*** be configured not to exceed any of these constraints documented in this section (certain combinations of external clock inputs, internal dividers, and PLL multiply ratios might not be supported).

**Table 6-13. PLLC1 Clock Frequency Ranges**

| CLOCK SIGNAL NAME | | MIN | MAX | UNIT |
|---|---|---|---|---|
| MXI/CLKIN(1) | | 20 | 30 | MHz |
| PLLOUT | At 1.2-V $CV_{DD}$ | 400 | 600 | MHz |
| | At 1.3-V $CV_{DD}$ (-810 *only*) | 400 | 810 | MHz |
| SYSCLK1 (CLKDIV1 Domain) | -594 (commercial temp) | | 600 | MHz |
| | -810 (commercial temp) | | 810 | MHz |
| | A-513 (extended temp) | | 513 | MHz |

(1) MXI/CLKIN input clock is used for both PLL Controllers (PLLC1 and PLLC2).

**Table 6-14. PLLC2 Clock Frequency Ranges**

| CLOCK SIGNAL NAME | | MIN | MAX | UNIT |
|---|---|---|---|---|
| MXI/CLKIN(1) | | 20 | 30 | MHz |
| PLLOUT | At 1.2-V $CV_{DD}$ | 400 | 900 | MHz |
| | At 1.3-V $CV_{DD}$ (-810 ***only***) | 400 | 900 | MHz |

(1) MXI/CLKIN input clock is used for both PLL Controllers (PLLC1 and PLLC2).

Both PLL1 and PLL2 have stabilization, lock, and reset timing requirements that ***must*** be followed.

The PLL stabilization time is the amount of time that ***must*** be allotted for the internal PLL regulators to become stable after the PLL is powered up (after PLLCTL.PLLPWRDN bit goes through a 1-to-0 transition). The PLL should *not* be operated until this stabilization time has expired. This stabilization step ***must*** be applied after these resets—a Power-on Reset, a Warm Reset, or a Max Reset, as the PLLCTL.PLLPWRDN bit resets to a "1". For the PLL stabliziation time value, see Table 6-15.

The PLL reset time is the amount of wait time needed for the PLL to properly reset (writing PLLRST = 0) before bringing the PLL out of reset (writing PLLRST = 1). For the PLL reset time value, see Table 6-15.

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). For the PLL lock time value, see Table 6-15.

**Table 6-15. PLL1 and PLL2 Stabilization, Lock, and Reset Times**

| PLL STABILIZATION/LOCK/RESET TIME | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|
| PLL Stabilization Time | 150 | | | µs |
| PLL Lock Time | | | 2000C(1) | ns |
| PLL Reset Time | 128C(1) | | | ns |

(1) C = CLKIN cycle time in ns. For example, when MXI/CLKIN frequency is 27 MHz, use C = 37.$\overline{037}$ ns.

For details on the PLL initialization software sequence, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

### 6.6.2 Clock PLL Considerations with External Clock Sources

If the internal oscillator is bypassed, to minimize the clock jitter a single clean power supply should power both the DM6446 device and the external clock oscillator circuit. The minimum CLKIN rise and fall times should also be observed. For the input clock timing requirements, see Section 6.6.3, *Clock PLL Electrical Data/Timing (Input and Output Clocks)*.

Rise/fall times, duty cycles (high/low pulse durations), and the load capacitance of the external clock source must meet the device requirements in this data manual (see Section 5.3, *Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Case Temperature*, and Section 6.6.3, *Clock PLL Electrical Data/Timing (Input and Output Clocks)*.

### 6.6.3 Clock PLL Electrical Data/Timing (Input and Output Clocks)

**Table 6-16. Timing Requirements for MXI/CLKIN (-594) Devices(1) (2) (3) (4) (see Figure 6-14)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(MXI)}$ | Cycle time, MXI/CLKIN | 33.$\overline{3}$ | 50 | ns |
| 2 | $t_{w(MXIH)}$ | Pulse duration, MXI/CLKIN high | 0.45C | 0.55C | ns |
| 3 | $t_{w(MXIL)}$ | Pulse duration, MXI/CLKIN low | 0.45C | 0.55C | ns |
| 4 | $t_{t(MXI)}$ | Transition time, MXI/CLKIN | | 0.05C | ns |
| 5 | $t_{J(MXI)}$ | Period jitter, MXI/CLKIN | | 0.02C | ns |

(1) The MXI/CLKIN frequency and PLL multiply factor should be chosen such that the resulting clock frequency is within the specific range for CPU operating frequency. For example, for a -594 speed device with a 27 MHz CLKIN frequency, the PLL multiply factor should be ≤ 22. For example, for a -810 speed device with a 27 MHz CLKIN frequency, the PLL multiply factor should be ≤ 30.
(2) The reference points for the rise and fall transitions are measured at $V_{IL}$ MAX and $V_{IH}$ MIN.
(3) For more details on the PLL multiplier factors, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).
(4) C = CLKIN cycle time in ns. For example, when MXI/CLKIN frequency is 27 MHz, use C = 37.$\overline{037}$ ns.

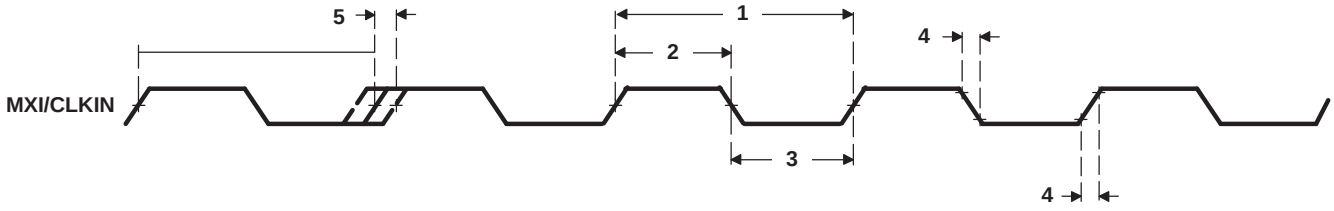



**Figure 6-14. MXI/CLKIN Timing**

**Table 6-17. Timing Requirements for M24XI (-594) Devices(1) (2) (3) (see Figure 6-15)**

| NO. | | | A-513, -594, -810 | | | UNIT |
|---|---|---|---|---|---|---|
| | | | MIN | TYP | MAX | |
| 1 | $t_{c(M24XI)}$ | Cycle time, M24XI | | 41.6̄ | | ns |
| 2 | $t_{w(M24XIH)}$ | Pulse duration, M24XI high | 0.45C | | 0.55C | ns |
| 3 | $t_{w(M24XIL)}$ | Pulse duration, M24XI low | 0.45C | | 0.55C | ns |
| 4 | $t_{t(M24XI)}$ | Transition time, M24XI | | | 0.05C | ns |
| 5 | $t_{J(M24XI)}$ | Period jitter, M24XI | | | 0.02C | ns |

(1) The reference points for the rise and fall transitions are measured at $V_{IL}$ MAX and $V_{IH}$ MIN.
(2) For more details on the PLL, see the *TMS320DM644x DMSoC Universal Serial Bus (USB) Controller User's Guide* (literature number SPRUE35).
(3) C = M24XI cycle time in ns. For example, when M24XI frequency is 24 MHz, use C = 41.6̄ ns.

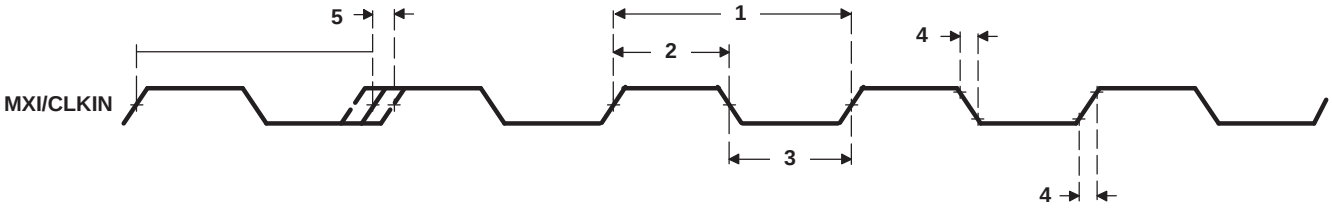



**Figure 6-15. M24XI Timing**

**Table 6-18. Switching Characteristics Over Recommended Operating Conditions for CLK_OUT0 (1) (2) (see Figure 6-16)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_C$ | Cycle time, CLK_OUT0 | 37.037 | 74.074 | ns |
| 2 | $t_{w(CLKOUT0H)}$ | Pulse duration, CLK_OUT0 high | 0.45P | 0.55P | ns |
| 3 | $t_{w(CLKOUT0L)}$ | Pulse duration, CLK_OUT0 low | 0.45P | 0.55P | ns |
| 4 | $t_{t(CLKOUT0)}$ | Transition time, CLK_OUT0 | | 0.05P | ns |
| 5 | $t_{d(CLKINH-CLKO0H)}$ | Delay time, CLKIN/MXI high to CLK_OUT0 high (divide-by-1 only) | 1 | 8 | ns |
| 6 | $t_{d(CLKINL-CLKO0L)}$ | Delay time, CLKIN/MXI low to CLK_OUT0 low (divide-by-1 only) | 1 | 8 | ns |
| 7 | $t_{d(CLKINH-CLKO0L)}$ | Delay time, CLKIN/MXI high to CLK_OUT0 low (divide-by-2 only) | 1 | 8 | ns |
| 8 | $t_{d(CLKINH-CLKO0H)}$ | Delay time, CLKIN/MXI high to CLK_OUT0 high (divide-by-2 only) | 1 | 8 | ns |

(1) The reference points for the rise and fall transitions are measured at $V_{OL}$ MAX and $V_{OH}$ MIN.
(2) P = 1/CLK_OUT0 clock frequency in nanoseconds (ns). For example, when CLK_OUT0 frequency is 27 MHz, use P = 37.04 ns.

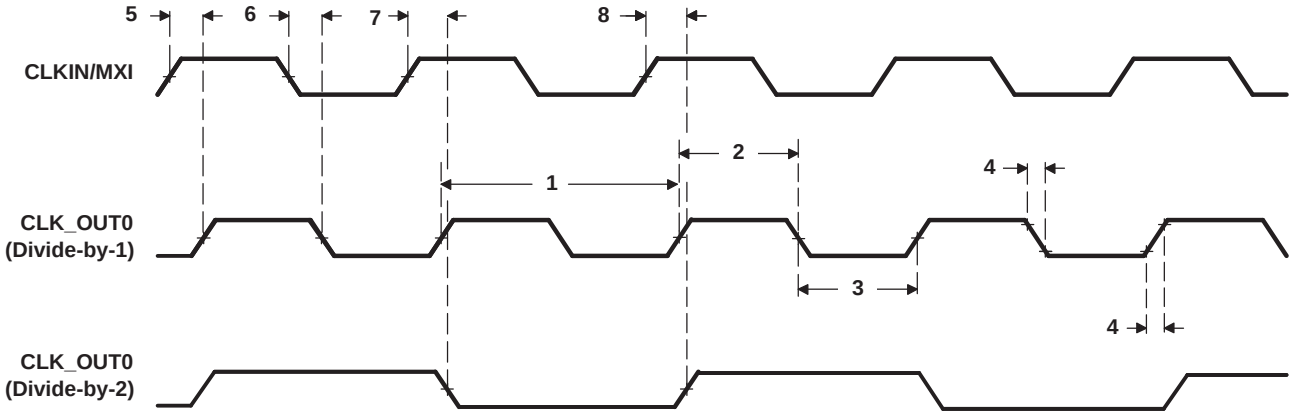



**Figure 6-16. CLK_OUT0 Timing**

**Table 6-19. Switching Characteristics Over Recommended Operating Conditions for CLK_OUT1[1] [2] (see Figure 6-17)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_c$ | Cycle time, CLK_OUT1 | 41.667 | 83.33 | ns |
| 2 | $t_{w(CLKOUT1H)}$ | Pulse duration, CLK_OUT1 high | 0.45P | 0.55P | ns |
| 3 | $t_{w(CLKOUT1L)}$ | Pulse duration, CLK_OUT1 low | 0.45P | 0.55P | ns |
| 4 | $t_{t(CLKOUT1)}$ | Transition time, CLK_OUT1 | | 0.05P | ns |
| 5 | $t_{d(CLKINH-CLKO1H)}$ | Delay time, CLKIN/MXI high to CLK_OUT1 high (divide-by-1 only) | 1 | 8 | ns |
| 6 | $t_{d(CLKINL-CLKO1L)}$ | Delay time, CLKIN/MXI low to CLK_OUT1 low (divide-by-1 only) | 1 | 8 | ns |
| 7 | $t_{d(CLKINH-CLKO1L)}$ | Delay time, CLKIN/MXI high to CLK_OUT1 low (divide-by-2 only) | 1 | 8 | ns |
| 8 | $t_{d(CLKINH-CLKO1H)}$ | Delay time, CLKIN/MXI high to CLK_OUT1 high (divide-by-2 only) | 1 | 8 | ns |

(1) The reference points for the rise and fall transitions are measured at $V_{OL}$ MAX and $V_{OH}$ MIN.
(2) P = 1/CLK_OUT1 clock frequency in nanoseconds (ns). For example, when CLK_OUT1 frequency is 24 MHz, use P = 41.$\overline{6}$ ns.

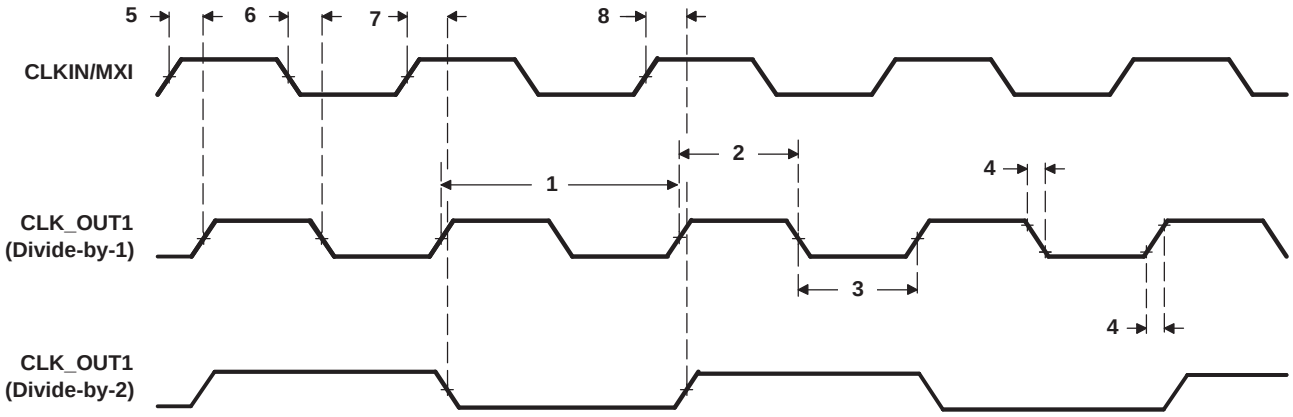



**Figure 6-17. CLK_OUT1 Timing**

## 6.7 Interrupts

The DM6446 device has a large number of interrupts to service the needs of its many peripherals and subsystems. Both the ARM and C64x+ are capable of servicing these interrupts. All of the device interrupts are routed to the ARM interrupt controller with only a limited set routed to the C64x+ interrupt controller. The interrupts can be selectively enabled or disabled in either of the controllers. In typical applications, the ARM handles most of the peripheral interrupts and grants control, to the C64x+, of interrupts that are relevant to DSP algorithms. Also, the ARM and DSP can communicate with each other through interrupts.

### *6.7.1 ARM CPU Interrupts*

The ARM9 CPU core supports 2 direct interrupts: FIQ and IRQ. The DM6446 ARM interrupt controller prioritizes up to 64 interrupt requests from various peripherals and subsystems, which are listed in Table 6-20, and interrupts the ARM CPU. Each interrupt is programmable for up to 8 levels of priority. There are 6 levels for IRQ and 2 levels for FIQ. Interrupts at the same priority level are serviced in order by the ARM Interrupt Number, with the lowest number having the highest priority. Table 6-21 shows the ARM interrupt controller registers and memory locations. For more details on ARM interrupt control, see the *TMS320DM644x DMSoC ARM Subsystem Reference Guide* (literature number SPRUE14).

**Table 6-20. DM6446 ARM Interrupts**

| ARM INTERRUPT NUMBER | ACRONYM | SOURCE | ARM INTERRUPT NUMBER | ACRONYM | SOURCE |
|---|---|---|---|---|---|
| 0 | VDINT0 | VPSS CCDC 0 | 32 | TINT0 | Timer 0 – TINT12 |
| 1 | VDINT1 | VPSS CCDC 1 | 33 | TINT1 | Timer 0 – TINT34 |
| 2 | VDINT2 | VPSS CCDC 2 | 34 | TINT2 | Timer 1 – TINT12 |
| 3 | HISTINT | VPSS Histogram | 35 | TINT3 | Timer 1 – TINT34 |
| 4 | H3AINT | VPSS AE/AWB/AF | 36 | PWMINT0 | PWM 0 |
| 5 | PRVUINT | VPSS Previewer | 37 | PWMINT1 | PWM 1 |
| 6 | RSZINT | VPSS Resizer | 38 | PWMINT2 | PWM 2 |
| 7 | - | Reserved | 39 | I2CINT | I2C |
| 8 | VENCINT | VPSS VPBE | 40 | UARTINT0 | UART 0 |
| 9 | ASQINT | VICP Sqr (ARM int) | 41 | UARTINT1 | UART 1 |
| 10 | IMXINT | VICP IMX | 42 | UARTINT2 | UART 2 |
| 11 | VLCDINT | VICP VLCD | 43 | SPINT0 | SPI |
| 12 | - | Reserved | 44 | SPINT1 | SPI |
| 13 | EMACINT | EMAC Memory Controller | 45 | - | Reserved |
| 14 | - | Reserved | 46 | DSP2ARM0 | DSP Controller to ARM 0 |
| 15 | - | Reserved | 47 | DSP2ARM1 | DSP Controller to ARM 1 |
| 16 | EDMA3CC_INT0 | EDMA3 CC Region 0 | 48 | GPIO0 | GPIO 0 |
| 17 | EDMA3CC_ERRINT | EDMA3 CC Error | 49 | GPIO1 | GPIO 1 |
| 18 | EDMA3TC_ERRINT0 | EDMA3 TC 0 Error | 50 | GPIO2 | GPIO 2 |
| 19 | EDMA3TC_ERRINT1 | EDMA3 TC 1 Error | 51 | GPIO3 | GPIO 3 |
| 20 | PSCINT | PSC ALLINT | 52 | GPIO4 | GPIO 4 |
| 21 | - | Reserved | 53 | GPIO5 | GPIO 5 |
| 22 | IDEINT | ATA / IDE | 54 | GPIO6 | GPIO 6 |
| 23 | HPINT | HPI | 55 | GPIO7 | GPIO 7 |
| 24 | ASPXINT | ASP Transmit | 56 | GPIOBNK0 | GPIO Bank 0 |
| 25 | ASPRINT | ASP Receive | 57 | GPIOBNK1 | GPIO Bank 1 |
| 26 | MMCINT | MMC | 58 | GPIOBNK2 | GPIO Bank 2 |
| 27 | SDIOINT | SD | 59 | GPIOBNK3 | GPIO Bank 3 |
| 28 | - | Reserved | 60 | GPIOBNK4 | GPIO Bank 4 |
| 29 | DDRINT | DDR2 Memory Controller | 61 | COMMTX | ARMSS |
| 30 | EMIFAINT | EMIFA | 62 | COMMRX | ARMSS |
| 31 | VLQINT | VLYNQ | 63 | EMUINT | E2ICE |

**Table 6-21. ARM Interrupt Controller Registers**

| HEX ADDRESS | ACRONYM | REGISTER DESCRIPTION |
|---|---|---|
| 0x01C4 8000 | FIQ0 | FIQ Interrupt Status 0 [Interrupt Status of INT[31:0] (If Mapped to FIQ)] |
| 0x01C4 8004 | FIQ1 | FIQ Interrupt Status 1 [Interrupt Status of INT[63:32] (If Mapped to FIQ)] |
| 0x01C4 8008 | IRQ0 | IRQ Interrupt Status 0 [Interrupt Status of INT[31:0] (If Mapped to IRQ)] |
| 0x01C4 800C | IRQ1 | IRQ Interrupt Status 1 [Interrupt Status of INT[63:32] (If Mapped to IRQ)] |
| 0x01C4 8010 | FIQENTRY | Entry Address [28:0] for Valid FIQ Interrupt |
| 0x01C4 8014 | IRQENTRY | Entry Address [28:0] for Valid IRQ Interrupt |
| 0x01C4 8018 | EINT0 | Interrupt Enable Register 0 |
| 0x01C4 801C | EINT1 | Interrupt Enable Register 1 |
| 0x01C4 8020 | INCTL | Interrupt Operation Control Register |
| 0x01C4 8024 | EABASE | Interrupt Entry Table Base Address Register |
| 0x01C4 8028 - 0x01C4 802F | - | Reserved |
| 0x01C4 8030 | INTPRI0 | Interrupt 0-7 Priority Select |
| 0x01C4 8034 | INTPRI1 | Interrupt 8-15 Priority Select |
| 0x01C4 8038 | INTPRI2 | Interrupt 16-23 Priority Select |
| 0x01C4 803C | INTPRI3 | Interrupt 24-31 Priority Select |
| 0x01C4 8040 | INTPRI4 | Interrupt 32-39 Priority Select |
| 0x01C4 8044 | INTPRI5 | Interrupt 40-47 Priority Select |
| 0x01C4 8048 | INTPRI6 | Interrupt 48-55 Priority Select |
| 0x01C4 804C | INTPRI7 | Interrupt 56-63 Priority Select |
| 0x01C4 8050 - 0x01C4 83FF | - | Reserved |

### 6.7.2 DSP Interrupts

The C64x+ DSP interrupt controller combines device events into 12 prioritized interrupts. The source for each of the 12 CPU interrupts is user programmable and is listed in Table 6-22. Also, the interrupt controller controls the generation of the CPU exception, NMI, and emulation interrupts. Table 6-23 summarizes the C64x+ interrupt controller registers and memory locations. For more details on DSP interrupt control, see the *TMS320DM644x DMSoC DSP Subsystem Reference Guide* (literature number SPRUE15).

**Table 6-22. DM6446 DSP Interrupts**

| DSP INTERRUPT NUMBER | ACRONYM | SOURCE | DSP INTERRUPT NUMBER | ACRONYM | SOURCE |
|---|---|---|---|---|---|
| 0 | EVT0 | C64x+ Int Ctl 0 | 64 | | Reserved |
| 1 | EVT1 | C64x+ Int Ctl 1 | 65 | | Reserved |
| 2 | EVT2 | C64x+ Int Ctl 2 | 66 | | Reserved |
| 3 | EVT3 | C64x+ Int Ctl 3 | 67 | | Reserved |
| 4 | TINT0 | Timer 0 – TINT12 | 68 | | Reserved |
| 5 | TINT1 | Timer 0 – TINT34 | 69 | | Reserved |
| 6 | TINT2 | Timer 1 – TINT12 | 70 | | Reserved |
| 7 | TINT3 | Timer 1 – TINT34 | 71 | | Reserved |
| 8 | | Reserved | 72 | | Reserved |
| 9 | EMU_DTDMA | C64x+ EMC | 73 | | Reserved |
| 10 | | Reserved | 74 | | Reserved |
| 11 | EMU_RTDXRX | C64x+ RTDX | 75 | | Reserved |
| 12 | EMU_RTDXTX | C64x+ RTDX | 76 | | Reserved |
| 13 | IDMAINT0 | C64x+ EMC 0 | 77 | | Reserved |
| 14 | IDMAINT1 | C64x+ EMC 1 | 78 | | Reserved |
| 15 | | Reserved | 79 | | Reserved |
| 16 | ARM2DSP0 | ARM to DSP Controller 0 | 80 | | Reserved |
| 17 | ARM2DSP1 | ARM to DSP Controller 1 | 81 | | Reserved |
| 18 | ARM2DSP2 | ARM to DSP Controller 2 | 82 | | Reserved |
| 19 | ARM2DSP3 | ARM to DSP Controller 3 | 83 | | Reserved |
| 20 | | Reserved | 84 | | Reserved |
| 21 | | Reserved | 85 | | Reserved |
| 22 | | Reserved | 86 | | Reserved |
| 23 | | Reserved | 87 | | Reserved |
| 24 | | Reserved | 88 | | Reserved |
| 25 | | Reserved | 89 | | Reserved |
| 26 | | Reserved | 90 | | Reserved |
| 27 | | Reserved | 91 | | Reserved |
| 28 | | Reserved | 92 | | Reserved |
| 29 | | Reserved | 93 | | Reserved |
| 30 | | Reserved | 94 | | Reserved |
| 31 | | Reserved | 95 | | Reserved |
| 32 | | Reserved | 96 | INTERR | C64x+ Interrupt Controller Dropped CPU Interrupt Event |
| 33 | | Reserved | 97 | EMC_IDMAERR | C64x+ EMC Invalid IDMA Parameters |
| 34 | | Reserved | 98 | | Reserved |
| 35 | | Reserved | 99 | | Reserved |

**Table 6-22. DM6446 DSP Interrupts (continued)**

| DSP INTERRUPT NUMBER | ACRONYM | SOURCE | DSP INTERRUPT NUMBER | ACRONYM | SOURCE |
|---|---|---|---|---|---|
| 36 | EDMA3CC_INT1 | EDMA3 CC Interrupt Region 1 | 100 | | Reserved |
| 37 | EDMA3CC_ERRINT | EDMA3 CC Error | 101 | | Reserved |
| 38 | EDMA3TC_ERRINT0 | EDMA3 TC0 Error | 102 | | Reserved |
| 39 | EDMA3TC_ERRINT1 | EDMA3 TC1 Error | 103 | | Reserved |
| 40 | PSCINT | PSC ALLINT | 104 | | Reserved |
| 41 | | Reserved | 105 | | Reserved |
| 42 | | Reserved | 106 | | Reserved |
| 43 | | Reserved | 107 | | Reserved |
| 44 | | Reserved | 108 | | Reserved |
| 45 | | Reserved | 109 | | Reserved |
| 46 | | Reserved | 110 | | Reserved |
| 47 | | Reserved | 111 | | Reserved |
| 48 | ASPXINT | ASP Transmit | 112 | PMC_ED | C64x+ PMC |
| 49 | ASPRINT | ASP Receive | 113 | | Reserved |
| 50 | | Reserved | 114 | | Reserved |
| 51 | | Reserved | 115 | | Reserved |
| 52 | | Reserved | 116 | UMCED1 | C64x+ UMC 1 |
| 53 | | Reserved | 117 | UMCED2 | C64x+ UMC 2 |
| 54 | | Reserved | 118 | PDCERR | C64x+ PDC |
| 55 | | Reserved | 119 | PVCINT | C64x+ PDC |
| 56 | | Reserved | 120 | PMCCMPA | C64x+ PMC |
| 57 | | Reserved | 121 | PMCDMPA | C64x+ PMC |
| 58 | | Reserved | 122 | DMCCMPA | C64x+ DMC |
| 59 | | Reserved | 123 | DMCDMPA | C64x+ DMC |
| 60 | | Reserved | 124 | UMCCMPA | C64x+ UMC |
| 61 | | Reserved | 125 | UMCDMPA | C64x+ UMC |
| 62 | | Reserved | 126 | EMCCMPA | C64x+ EMC |
| 63 | | Reserved | 127 | EMCDMPA | C64x+ EMC |

**Table 6-23. C64x+ Interrupt Controller Registers**

| HEX ADDRESS | ACRONYM | REGISTER DESCRIPTION |
|---|---|---|
| 0x0180 0000 | EVTFLAG0 | Event flag register 0 |
| 0x0180 0004 | EVTFLAG1 | Event flag register 1 |
| 0x0180 0008 | EVTFLAG2 | Event flag register 2 |
| 0x0180 000C | EVTFLAG3 | Event flag register 3 |
| 0x0180 0020 | EVTSET0 | Event set register 0 |
| 0x0180 0024 | EVTSET1 | Event set register 1 |
| 0x0180 0028 | EVTSET2 | Event set register 2 |
| 0x0180 002C | EVTSET3 | Event set register 3 |
| 0x0180 0040 | EVTCLR0 | Event clear register 0 |
| 0x0180 0044 | EVTCLR1 | Event clear register 1 |
| 0x0180 0048 | EVTCLR2 | Event clear register 2 |
| 0x0180 004C | EVTCLR3 | Event clear register 3 |
| 0x0180 0080 | EVTMASK0 | Event mask register 0 |
| 0x0180 0084 | EVTMASK1 | Event mask register 1 |
| 0x0180 0088 | EVTMASK2 | Event mask register 2 |
| 0x0180 008C | EVTMASK3 | Event mask register 3 |
| 0x0180 00A0 | MEVTFLAG0 | Masked event flag register 0 |
| 0x0180 00A4 | MEVTFLAG1 | Masked event flag register 1 |
| 0x0180 00A8 | MEVTFLAG2 | Masked event flag register 2 |
| 0x0180 00AC | MEVTFLAG3 | Masked event flag register 3 |
| 0x0180 00C0 | EXPMASK0 | Exception mask register 0 |
| 0x0180 00C4 | EXPMASK1 | Exception mask register 1 |
| 0x0180 00C8 | EXPMASK2 | Exception mask register 2 |
| 0x0180 00CC | EXPMASK3 | Exception mask register 3 |
| 0x0180 00E0 | MEXPFLAG0 | Masked exception flag register 0 |
| 0x0180 00E4 | MEXPFLAG1 | Masked exception flag register 1 |
| 0x0180 00E8 | MEXPFLAG2 | Masked exception flag register 2 |
| 0x0180 00EC | MEXPFLAG3 | Masked exception flag register 3 |
| 0x0180 0104 | INTMUX1 | Interrupt mux register 1 |
| 0x0180 0108 | INTMUX2 | Interrupt mux register 2 |
| 0x0180 010C | INTMUX3 | Interrupt mux register 3 |
| 0x0180 0140 - 0x0180 0144 | - | Reserved |
| 0x0180 0180 | INTXSTAT | Interrupt exception status |
| 0x0180 0184 | INTXCLR | Interrupt exception clear |
| 0x0180 0188 | INTDMASK | Dropped interrupt mask register |
| 0x0180 01C0 | EVTASRT | Event assert register |

### 6.7.3 ARM/DSP Communications Interrupts

The INTGEN register is used for generating interrupts between the ARM and DSP. The INTGEN register format is shown in Figure 6-18. Table 6-24 describes the register bit fields. The ARM may generate an interrupt to the DSP by setting one of the four INTDSP[3:0] bits or the INTNMI bit. The interrupt bit automatically self clears and the corresponding DSP[3:0]STAT or NMISTAT bit is automatically set to indicate that the interrupt was generated. After servicing the interrupt, the DSP clears the status bit by writing '0'. The ARM may poll the status bit to determine when the DSP has completed servicing the interrupt. The DSP may generate an interrupt to the ARM in the same manner using the INTARM[1:0] bits and monitor ARM interrupt servicing via the ARM[1:0]STAT bits.

**Figure 6-18. INTGEN Register**

| 31 | 30 | 29 | 28 | 27 | 24 | 23 | 22 | 21 | 20 | 19 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | ARM1 STAT | ARM0 STAT | Reserved | | DSP3 STAT | DSP2 STAT | DSP1 STAT | DSP0 STAT | Reserved | | NMI STAT |
| R-00 | | R/W-0 | R/W-0 | R-0000 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-000 | | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 8 | 7 | 6 | 5 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | INT ARM1 | INT ARM0 | Reserved | | INT DSP3 | INT DSP2 | INT DSP1 | INT DSP0 | Reserved | | INT NMI |
| R-00 | | R/W-0 | R/W-0 | R-0000 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-000 | | R/W-0 |

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

**Table 6-24. INTGEN Register Bit Fields Descriptions**

| Name | Description |
|---|---|
| ARM1STAT | DSP to ARM Int1 Status/Clear(1) |
| ARM0STAT | DSP to ARM Int0 Status/Clear(1) |
| DSP3STAT | ARM to DSP Int3 Status/Clear(1) |
| DSP2STAT | ARM to DSP Int2 Status/Clear(1) |
| DSP1STAT | ARM to DSP Int1 Status/Clear(1) |
| DSP0STAT | ARM to DSP Int0 Status/Clear(1) |
| NMISTAT | DSP NMI Status/Clear(1) |
| INTARM1 | DSP to ARM Int1 Set(2) |
| INTARM0 | DSP to ARM Int0 Set(2) |
| INTDSP3 | ARM to DSP Int3 Set(2) |
| INTDSP2 | ARM to DSP Int2 Set(2) |
| INTDSP1 | ARM to DSP Int1 Set(2) |
| INTDSP0 | ARM to DSP Int0 Set(2) |
| INTNMI | DSP NMI Set(2) |

(1) Write '0' to clear. Writing '1' has no effect.
(2) Write '1' to generate the interrupt. The register bit automatically clears to a value of '0'. Writing a '0' has no effect.

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

The GPIO peripheral provides general-purpose pins that can be configured as either inputs or outputs. When configured as an output, a write to an internal register can control the state driven on the output pin. When configured as an input, the state of the input is detectable by reading the state of an internal register. In addition, the GPIO peripheral can produce CPU interrupts and EDMA3 events in different interrupt/event generation modes. The GPIO peripheral provides generic connections to external devices. The GPIO pins are grouped into banks of 16 pins per bank (i.e., bank 0 consists of GPIO [0:15]).

The DM6446 GPIO peripheral supports the following:

- Up to 54 1.8v GPIO pins, GPIO[0:53]
- Up to 17 3.3v GPIO pins, GPIO3V[0:16] (GPIO[54:70])
- Interrupts:
  - Up to 8 unique GPIO[0:7] interrupts from Bank 0
  - 5 GPIO bank (aggregated) interrupt signals from each of the 5 banks of GPIOs
  - Interrupts can be triggered by rising and/or falling edge, specified for each interrupt capable GPIO signal
- DMA events:
  - Up to 8 unique GPIO DMA events from Bank 0
  - 5 GPIO bank (aggregated) DMA event signals from each of the 5 banks of GPIOs
- Set/clear functionality: Firmware writes 1 to corresponding bit position(s) to set or to clear GPIO signal(s). This allows multiple firmware processes to toggle GPIO output signals without critical section protection (disable interrupts, program GPIO, re-enable interrupts, to prevent context switching to anther process during GPIO programming).
- Separate Input/Output registers
- Output register in addition to set/clear so that, if preferred by firmware, some GPIO output signals can be toggled by direct write to the output register(s).
- Output register, when read, reflects output drive status. This, in addition to the input register reflecting pin status and open-drain I/O cell, allows wired logic be implemented.

The memory map for the GPIO registers is shown in Table 6-25. For more detailed information on GPIOs, see the *TMS320DM644x DMSoC General-Purpose Input/Output (GPIO) User's Guide* (literature number SPRUE25).

### 6.8.1 GPIO Peripheral Register Description(s)

**Table 6-25. GPIO Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C6 7000 | PID | Peripheral Identification Register |
| 0x01C6 7004 | - | Reserved |
| 0x01C6 7008 | BINTEN | GPIO interrupt per-bank enable |
| **GPIO Banks 0 and 1** | | |
| 0x01C6 700C | - | Reserved |
| 0x01C6 7010 | DIR01 | GPIO Banks 0 and 1 Direction Register (GPIO[0:31]) |
| 0x01C6 7014 | OUT_DATA01 | GPIO Banks 0 and 1 Output Data Register (GPIO[0:31]) |
| 0x01C6 7018 | SET_DATA01 | GPIO Banks 0 and 1 Set Data Register (GPIO[0:31]) |
| 0x01C6 701C | CLR_DATA01 | GPIO Banks 0 and 1 Clear data for banks 0 and 1 (GPIO[0:31]) |
| 0x01C6 7020 | IN_DATA01 | GPIO Banks 0 and 1 Input Data Register (GPIO[0:31]) |
| 0x01C6 7024 | SET_RIS_TRIG01 | GPIO Banks 0 and 1 Set Rising Edge Interrupt Register (GPIO[0:31]) |
| 0x01C6 7028 | CLR_RIS_TRIG01 | GPIO Banks 0 and 1 Clear Rising Edge Interrupt Register (GPIO[0:31]) |
| 0x01C6 702C | SET_FAL_TRIG01 | GPIO Banks 0 and 1 Set Falling Edge Interrupt Register (GPIO[0:31]) |
| 0x01C6 7030 | CLR_FAL_TRIG01 | GPIO Banks 0 and 1 Clear Falling Edge Interrupt Register (GPIO[0:31]) |
| 0x01C6 7034 | INSTAT01 | GPIO Banks 0 and 1 Interrupt Status Register (GPIO[0:31]) |
| **GPIO Banks 2 and 3** | | |
| 0x01C6 7038 | DIR23 | GPIO Banks 2 and 3 Direction Register (GPIO[32:63]) |
| 0x01C6 703C | OUT_DATA23 | GPIO Banks 2 and 3 Output Data Register (GPIO[32:63]) |
| 0x01C6 7040 | SET_DATA23 | GPIO Banks 2 and 3 Set Data Register (GPIO[32:63]) |
| 0x01C6 7044 | CLR_DATA23 | GPIO Banks 2 and 3 Clear Data Register (GPIO[32:63]) |
| 0x01C6 7048 | IN_DATA23 | GPIO Banks 2 and 3 Input Data Register (GPIO[32:63]) |
| 0x01C6 704C | SET_RIS_TRIG23 | GPIO Banks 2 and 3 Set Rising Edge Interrupt Register (GPIO[32:63]) |
| 0x01C6 7050 | CLR_RIS_TRIG23 | GPIO Banks 2 and 3 Clear Rising Edge Interrupt Register (GPIO[32:63]) |
| 0x01C6 7054 | SET_FAL_TRIG23 | GPIO Banks 2 and 3 Set Falling Edge Interrupt Register (GPIO[32:63]) |
| 0x01C6 7058 | CLR_FAL_TRIG23 | GPIO Banks 2 and 3 Clear Falling Edge Interrupt Register (GPIO[32:63]) |
| 0x01C6 705C | INSTAT23 | GPIO Banks 2 and 3 Interrupt Status Register (GPIO[32:63]) |
| **GPIO Bank 4** | | |
| 0x01C6 7060 | DIR4 | GPIO Bank 4 Direction Register (GPIO[64:70]) |
| 0x01C6 7064 | OUT_DATA4 | GPIO Bank 4 Output Data Register (GPIO[64:70]) |
| 0x01C6 7068 | SET_DATA4 | GPIO Bank 4 Set Data Register (GPIO[64:70]) |
| 0x01C6 706C | CLR_DATA4 | GPIO Bank 4 Clear Data Register (GPIO[64:70]) |
| 0x01C6 7070 | IN_DATA4 | GPIO Bank 4 Input Data Register (GPIO[64:70]) |
| 0x01C6 7074 | SET_RIS_TRIG4 | GPIO Bank 4 Set Rising Edge Interrupt Register (GPIO[64:70]) |
| 0x01C6 7078 | CLR_RIS_TRIG4 | GPIO Bank 4 Clear Rising Edge Interrupt Register (GPIO[64:70]) |
| 0x01C6 707C | SET_FAL_TRIG4 | GPIO Bank 4 Set Falling Edge Interrupt Register (GPIO[64:70]) |
| 0x01C6 7080 | CLR_FAL_TRIG4 | GPIO Bank 4 Clear Falling Edge Interrupt Register (GPIO[64:70]) |
| 0x01C6 7084 | INSTAT4 | GPIO Bank 4 Interrupt Status Register (GPIO[64:70]) |
| 0x01C6 7088 - 0x01C6 7FFF | - | Reserved |

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

**Table 6-26. Timing Requirements for GPIO Inputs[1] (see Figure 6-19)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(GPIH)}$ | Pulse duration, GPIx high | 52 | | ns |
| 2 | $t_{w(GPIL)}$ | Pulse duration, GPIx low | 52 | | ns |

(1) The pulse width given is sufficient to generate a CPU interrupt or an EDMA3 event. However, if a user wants to have DM6446 recognize the GPIx changes through software polling of the GPIO register, the GPIx duration must be extended to allow DM6446 enough time to access the GPIO register through the internal bus.

**Table 6-27. Switching Characteristics Over Recommended Operating Conditions for GPIO Outputs (see Figure 6-19)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{w(GPOH)}$ | Pulse duration, GPOx high | 26[1] | | ns |
| 4 | $t_{w(GPOL)}$ | Pulse duration, GPOx low | 26[1] | | ns |

(1) This parameter value should not be used as a maximum performance specification. Actual performance of back-to-back accesses of the GPIO is dependent upon internal bus activity.

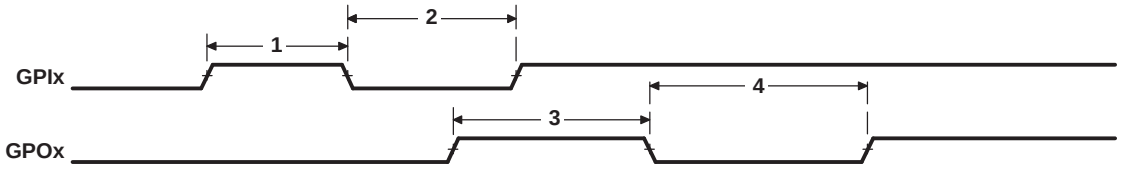



**Figure 6-19. GPIO Port Timing**

### 6.8.3 GPIO Peripheral External Interrupts Electrical Data/Timing

**Table 6-28. Timing Requirements for External Interrupts[1] (see Figure 6-20)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(ILOW)}$ | Width of the external interrupt pulse low | 52 | | ns |
| 2 | $t_{w(IHIGH)}$ | Width of the external interrupt pulse high | 52 | | ns |

(1) The pulse width given is sufficient to generate an interrupt or an EDMA3 event. However, if a user wants to have DM6446recognize the GPIO changes through software polling of the GPIO register, the GPIO duration must be extended to allow DM6446 enough time to access the GPIO register through the internal bus.

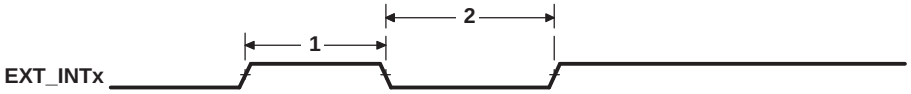



**Figure 6-20. GPIO External Interrupt Timing**

## 6.9 Enhanced Direct Memory Access (EDMA3) Controller

The EDMA3 controller handles all data transfers between memories and the device slave peripherals on the DM6446 device. These data transfers include cache servicing, non-cacheable memory accesses, user-programmed data transfers, and host accesses. These are summarized as follows:

- Transfer to/from on-chip memories
  - Coprocessor shared memory
  - DSP L1D memory
  - DSP L2 memory
  - ARM program/data RAM
- Transfer to/from external storage
  - DDR2 SDRAM
  - NAND flash
  - Asynchronous EMIF
  - Smart Media, SD, MMC, xD media storage
  - ATA/CF
- Transfer to/from peripherals/hosts
  - VLYNQ
  - ASP
  - SPI
  - PWM
  - UART

The EDMA3 controller supports two addressing modes: constant addressing mode and increment addressing mode. On the DM6446 device, constant addressing mode is not supported by any peripheral or internal memory. For more information on these two addressing modes, see the *TMS320DM644x DMSoC Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (literature number SPRUE23).

### 6.9.1 EDMA3 Channel Synchronization Events

The EDMA3 supports up to 64 EDMA3 channels which service peripheral devices and external memory. Table 6-29 lists the source of EDMA3 synchronization events associated with each of the programmable EDMA3 channels. For the DM6446 device, the association of an event to a channel is fixed; each of the EDMA3 channels has one specific event associated with it. These specific events are captured in the EDMA3 event registers (ER, ERH) even if the events are disabled by the EDMA3 event enable registers (EER, EERH). For more detailed information on the EDMA3 module and how EDMA3 events are enabled, captured, processed, linked, chained, and cleared, etc., see the *TMS320DM644x DMSoC Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (literature number SPRUE23).

**Table 6-29. DM6446 EDMA3 Channel Synchronization Events(1)**

| EDMA3 CHANNEL | EVENT NAME | EVENT DESCRIPTION |
|---|---|---|
| 0-1 | | Reserved |
| 2 | XEVT | ASP Transmit Event |
| 3 | REVT | ASP Receive Event |
| 4 | HISTEVT | VPSS Histogram Event |
| 5 | H3AEVT | VPSS H3A Event |
| 6 | PRVUEVT | VPSS Previewer Event |
| 7 | RSZEVT | VPSS Resizer Event |

(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 EDMA3 event-transfer chaining, see the *TMS320DM644x DMSoC Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (literature number SPRUE23).

**Table 6-29. DM6446 EDMA3 Channel Synchronization Events (continued)**

| EDMA3 CHANNEL | EVENT NAME | EVENT DESCRIPTION |
|---|---|---|
| 8 | IMXINT | VICP Interrupt |
| 9 | VLCDINT | VICP VLCD Interrupt |
| 10 | ASQINT | VICP ASQ Interrupt |
| 11 | DSQINT | VICP DSQ Interrupt |
| 12-15 | | Reserved |
| 16 | SPIXEVT | SPI Transmit Event |
| 17 | SPIREVT | SPI Receive Event |
| 18 | URXEVT0 | UART 0 Receive Event |
| 19 | UTXEVT0 | UART 0 Transmit Event |
| 20 | URXEVT1 | UART 1 Receive Event |
| 21 | UTXEVT1 | UART 1 Transmit Event |
| 22 | URXEVT2 | UART 2 Receive Event |
| 23 | UTXEVT2 | UART 2 Transmit Event |
| 24 | | Reserved |
| 25 | | Reserved |
| 26 | MMCRXEVT | MMC Receive Event |
| 27 | MMCTXEVT | MMC Transmit Event |
| 28 | I2CREVT | I2C Receive Event |
| 29 | I2CXEVT | I2C Transmit Event |
| 30-31 | | Reserved |
| 32 | GPINT0 | GPIO 0 Interrupt |
| 33 | GPINT1 | GPIO 1 Interrupt |
| 34 | GPINT2 | GPIO 2 Interrupt |
| 35 | GPINT3 | GPIO 3 Interrupt |
| 36 | GPINT4 | GPIO 4 Interrupt |
| 37 | GPINT5 | GPIO 5 Interrupt |
| 38 | GPINT6 | GPIO 6 Interrupt |
| 39 | GPINT7 | GPIO 7 Interrupt |
| 40 | GPBNKINT0 | GPIO Bank 0 Interrupt |
| 41 | GPBNKINT1 | GPIO Bank 1 Interrupt |
| 42 | GPBNKINT2 | GPIO Bank 2 Interrupt |
| 43 | GPBNKINT3 | GPIO Bank 3 Interrupt |
| 44 | GPBNKINT4 | GPIO Bank 4 Interrupt |
| 45-47 | | Reserved |
| 48 | TINT0 | Timer 0 Interrupt |
| 49 | TINT1 | Timer 1 Interrupt |
| 50 | TINT2 | Timer 2 Interrupt |
| 51 | TINT3 | Timer 3 Interrupt |
| 52 | PWM0 | PWM 0 Event |
| 53 | PWM1 | PWM 1 Event |
| 54 | PWM2 | PWM 2 Event |
| 55-63 | | Reserved |

### 6.9.2 EDMA3 Peripheral Register Descriptions

Table 6-30 lists the EDMA3 registers, their corresponding acronyms, and DM6446 device memory locations.

**Table 6-30. DM6446 EDMA3 Registers**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| **Channel Controller Registers** | | |
| 0x01c0 0000 - 0x01c0 0003 | | Reserved |
| 0x01c0 0004 | CCCFG | EDMA3CC Configuration Register |
| 0x01c0 0008 - 0x01c0 01FF | | Reserved |
| **Global Registers** | | |
| 0x01c0 0200 | QCHMAP0 | QDMA Channel 0 Mapping to PaRAM Register |
| 0x01c0 0204 | QCHMAP1 | QDMA Channel 1 Mapping to PaRAM Register |
| 0x01c0 0208 | QCHMAP2 | QDMA Channel 2 Mapping to PaRAM Register |
| 0x01c0 020C | QCHMAP3 | QDMA Channel 3 Mapping to PaRAM Register |
| 0x01c0 0210 | QCHMAP4 | QDMA Channel 4 Mapping to PaRAM Register |
| 0x01c0 0214 | QCHMAP5 | QDMA Channel 5 Mapping to PaRAM Register |
| 0x01c0 0218 | QCHMAP6 | QDMA Channel 6 Mapping to PaRAM Register |
| 0x01c0 021C | QCHMAP7 | QDMA Channel 7 Mapping to PaRAM Register |
| 0x01c0 0240 | DMAQNUM0 | DMA Queue Number Register 0 (Channels 00 to 07) |
| 0x01c0 0244 | DMAQNUM1 | DMA Queue Number Register 1 (Channels 08 to 15) |
| 0x01c0 0248 | DMAQNUM2 | DMA Queue Number Register 2 (Channels 16 to 23) |
| 0x01c0 024C | DMAQNUM3 | DMA Queue Number Register 3 (Channels 24 to 31) |
| 0x01c0 0250 | DMAQNUM4 | DMA Queue Number Register 4 (Channels 32 to 39) |
| 0x01c0 0254 | DMAQNUM5 | DMA Queue Number Register 5 (Channels 40 to 47) |
| 0x01c0 0258 | DMAQNUM6 | DMA Queue Number Register 6 (Channels 48 to 55) |
| 0x01c0 025C | DMAQNUM7 | DMA Queue Number Register 7 (Channels 56 to 63) |
| 0x01c0 0260 | QDMAQNUM | CC QDMA Queue Number |
| 0x01c0 0264 - 0x01c0 0283 | – | Reserved |
| 0x01c0 0284 | QUEPRI | Queue Priority Register |
| 0x01c0 0288 - 0x01c0 02FF | – | Reserved |
| 0x01c0 0300 | EMR | Event Missed Register |
| 0x01c0 0304 | EMRH | Event Missed Register High |
| 0x01c0 0308 | EMCR | Event Missed Clear Register |
| 0x01c0 030C | EMCRH | Event Missed Clear Register High |
| 0x01c0 0310 | QEMR | QDMA Event Missed Register |
| 0x01c0 0314 | QEMCR | QDMA Event Missed Clear Register |
| 0x01c0 0318 | CCERR | EDMA3CC Error Register |
| 0x01c0 031C | CCERRCLR | EDMA3CC Error Clear Register |
| 0x01c0 0320 | EEVAL | Error Evaluate Register |
| 0x01c0 0340 | DRAE0 | DMA Region Access Enable Register for Region 0 |
| 0x01c0 0344 | DRAEH0 | DMA Region Access Enable Register High for Region 0 |
| 0x01c0 0348 | DRAE1 | DMA Region Access Enable Register for Region 1 |
| 0x01c0 034C | DRAEH1 | DMA Region Access Enable Register High for Region 1 |
| 0x01c0 0350 | DRAE2 | DMA Region Access Enable Register for Region 2 |
| 0x01c0 0354 | DRAEH2 | DMA Region Access Enable Register High for Region 2 |
| 0x01c0 0358 | DRAE3 | DMA Region Access Enable Register for Region 3 |
| 0x01c0 035C | DRAEH3 | DMA Region Access Enable Register High for Region 3 |
| 0x01c0 0360 - 0x01c0 037C | – | Reserved |

**Table 6-30. DM6446 EDMA3 Registers (continued)**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01c0 0380 | QRAE0 | QDMA Region Access Enable Register for Region 0 |
| 0x01c0 0384 | QRAE1 | QDMA Region Access Enable Register for Region 1 |
| 0x01c0 0388 | QRAE2 | QDMA Region Access Enable Register for Region 2 |
| 0x01c0 038C | QRAE3 | QDMA Region Access Enable Register for Region 3 |
| 0x01c0 0390 - 0x01c0 039C | – | Reserved |
| 0x01c0 0400 | Q0E0 | Event Q0 Entry 0 Register |
| 0x01c0 0404 | Q0E1 | Event Q0 Entry 1 Register |
| 0x01c0 0408 | Q0E2 | Event Q0 Entry 2 Register |
| 0x01c0 040C | Q0E3 | Event Q0 Entry 3 Register |
| 0x01c0 0410 | Q0E4 | Event Q0 Entry 4 Register |
| 0x01c0 0414 | Q0E5 | Event Q0 Entry 5 Register |
| 0x01c0 0418 | Q0E6 | Event Q0 Entry 6 Register |
| 0x01c0 041C | Q0E7 | Event Q0 Entry 7 Register |
| 0x01c0 0420 | Q0E8 | Event Q0 Entry 8 Register |
| 0x01c0 0424 | Q0E9 | Event Q0 Entry 9 Register |
| 0x01c0 0428 | Q0E10 | Event Q0 Entry 10 Register |
| 0x01c0 042C | Q0E11 | Event Q0 Entry 11 Register |
| 0x01c0 0430 | Q0E12 | Event Q0 Entry 12 Register |
| 0x01c0 0434 | Q0E13 | Event Q0 Entry 13 Register |
| 0x01c0 0438 | Q0E14 | Event Q0 Entry 14 Register |
| 0x01c0 043C | Q0E15 | Event Q0 Entry 15 Register |
| 0x01c0 0440 | Q1E0 | Event Q1 Entry 0 Register |
| 0x01c0 0444 | Q1E1 | Event Q1 Entry 1 Register |
| 0x01c0 0448 | Q1E2 | Event Q1 Entry 2 Register |
| 0x01c0 044C | Q1E3 | Event Q1 Entry 3 Register |
| 0x01c0 0450 | Q1E4 | Event Q1 Entry 4 Register |
| 0x01c0 0454 | Q1E5 | Event Q1 Entry 5 Register |
| 0x01c0 0458 | Q1E6 | Event Q1 Entry 6 Register |
| 0x01c0 045C | Q1E7 | Event Q1 Entry 7 Register |
| 0x01c0 0460 | Q1E8 | Event Q1 Entry 8 Register |
| 0x01c0 0464 | Q1E9 | Event Q1 Entry 9 Register |
| 0x01c0 0468 | Q1E10 | Event Q1 Entry 10 Register |
| 0x01c0 046C | Q1E11 | Event Q1 Entry 11 Register |
| 0x01c0 0470 | Q1E12 | Event Q1 Entry 12 Register |
| 0x01c0 0474 | Q1E13 | Event Q1 Entry 13 Register |
| 0x01c0 0478 | Q1E14 | Event Q1 Entry 14 Register |
| 0x01c0 047C | Q1E15 | Event Q1 Entry 15 Register |
| 0x01c0 0480 - 0x01c0 05FF | | Reserved |
| 0x01c0 0600 | QSTAT0 | Queue 0 Status Register |
| 0x01c0 0604 | QSTAT1 | Queue 1 Status Register |
| 0x01c0 0608 - 0x01c0 061F | | Reserved |
| 0x01c0 0620 | QWMTHRA | Queue Watermark Threshold A Register for Q[3:0] |
| 0x01c0 0624 | – | Reserved |
| 0x01c0 0640 | CCSTAT | EDMA3CC Status Register |
| 0x01c0 0644 - 0x01c0 0FFF | | Reserved |

**Table 6-30. DM6446 EDMA3 Registers (continued)**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| **Global Channel Registers** | | |
| 0x01c0 1000 | ER | Event Register |
| 0x01c0 1004 | ERH | Event Register High |
| 0x01c0 1008 | ECR | Event Clear Register |
| 0x01c0 100C | ECRH | Event Clear Register High |
| 0x01c0 1010 | ESR | Event Set Register |
| 0x01c0 1014 | ESRH | Event Set Register High |
| 0x01c0 1018 | CER | Chained Event Register |
| 0x01c0 101C | CERH | Chained Event Register High |
| 0x01c0 1020 | EER | Event Enable Register |
| 0x01c0 1024 | EERH | Event Enable Register High |
| 0x01c0 1028 | EECR | Event Enable Clear Register |
| 0x01c0 102C | EECRH | Event Enable Clear Register High |
| 0x01c0 1030 | EESR | Event Enable Set Register |
| 0x01c0 1034 | EESRH | Event Enable Set Register High |
| 0x01c0 1038 | SER | Secondary Event Register |
| 0x01c0 103C | SERH | Secondary Event Register High |
| 0x01c0 1040 | SECR | Secondary Event Clear Register |
| 0x01c0 1044 | SECRH | Secondary Event Clear Register High |
| 0x01c0 1048 - 0x01c0 104F | | Reserved |
| 0x01c0 1050 | IER | Interrupt Enable Register |
| 0x01c0 1054 | IERH | Interrupt Enable Register High |
| 0x01c0 1058 | IECR | Interrupt Enable Clear Register |
| 0x01c0 105C | IECRH | Interrupt Enable Clear Register High |
| 0x01c0 1060 | IESR | Interrupt Enable Set Register |
| 0x01c0 1064 | IESRH | Interrupt Enable Set Register High |
| 0x01c0 1068 | IPR | Interrupt Pending Register |
| 0x01c0 106C | IPRH | Interrupt Pending Register High |
| 0x01c0 1070 | ICR | Interrupt Clear Register |
| 0x01c0 1074 | ICRH | Interrupt Clear Register High |
| 0x01c0 1078 | IEVAL | Interrupt Evaluate Register |
| 0x01c0 1080 | QER | QDMA Event Register |
| 0x01c0 1084 | QEER | QDMA Event Enable Register |
| 0x01c0 1088 | QEECR | QDMA Event Enable Clear Register |
| 0x01c0 108C | QEESR | QDMA Event Enable Set Register |
| 0x01c0 1090 | QSER | QDMA Secondary Event Register |
| 0x01c0 1094 | QSECR | QDMA Secondary Event Clear Register |
| 0x01c0 1098 - 0x01c0 1FFF | | Reserved |
| **Shadow Region 0 Channel Registers** | | |
| 0x01c0 2000 | ER | Event Register |
| 0x01c0 2004 | ERH | Event Register High |
| 0x01c0 2008 | ECR | Event Clear Register |
| 0x01c0 200C | ECRH | Event Clear Register High |
| 0x01c0 2010 | ESR | Event Set Register |
| 0x01c0 2014 | ESRH | Event Set Register High |
| 0x01c0 2018 | CER | Chained Event Register |
| 0x01c0 201C | CERH | Chained Event Register High |

**Table 6-30. DM6446 EDMA3 Registers (continued)**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01c0 2020 | EER | Event Enable Register |
| 0x01c0 2024 | EERH | Event Enable Register High |
| 0x01c0 2028 | EECR | Event Enable Clear Register |
| 0x01c0 202C | EECRH | Event Enable Clear Register High |
| 0x01c0 2030 | EESR | Event Enable Set Register |
| 0x01c0 2034 | EESRH | Event Enable Set Register High |
| 0x01c0 2038 | SER | Secondary Event Register |
| 0x01c0 203C | SERH | Secondary Event Register High |
| 0x01c0 2040 | SECR | Secondary Event Clear Register |
| 0x01c0 2044 | SECRH | Secondary Event Clear Register High |
| 0x01c0 2048 - 0x01c0 204C | - | Reserved |
| 0x01c0 2050 | IER | Interrupt Enable Register |
| 0x01c0 2054 | IERH | Interrupt Enable Register High |
| 0x01c0 2058 | IECR | Interrupt Enable Clear Register |
| 0x01c0 205C | IECRH | Interrupt Enable Clear Register High |
| 0x01c0 2060 | IESR | Interrupt Enable Set Register |
| 0x01c0 2064 | IESRH | Interrupt Enable Set Register High |
| 0x01c0 2068 | IPR | Interrupt Pending Register |
| 0x01c0 206C | IPRH | Interrupt Pending Register High |
| 0x01c0 2070 | ICR | Interrupt Clear Register |
| 0x01c0 2074 | ICRH | Interrupt Clear Register High |
| 0x01c0 2078 | IEVAL | Interrupt Evaluate Register |
| 0x01c0 207C | - | Reserved |
| 0x01c0 2080 | QER | QDMA Event Register |
| 0x01c0 2084 | QEER | QDMA Event Enable Register |
| 0x01c0 2088 | QEECR | QDMA Event Enable Clear Register |
| 0x01c0 208C | QEESR | QDMA Event Enable Set Register |
| 0x01c0 2090 | QSER | QDMA Secondary Event Register |
| 0x01c0 2094 | QSECR | QDMA Secondary Event Clear Register |
| 0x01c0 2098 - 0x01c0 21FC | - | Reserved |
| **Shadow Region 1 Channel Registers** | | |
| 0x01c0 2200 | ER | Event Register |
| 0x01c0 2204 | ERH | Event Register High |
| 0x01c0 2208 | ECR | Event Clear Register |
| 0x01c0 220C | ECRH | Event Clear Register High |
| 0x01c0 2210 | ESR | Event Set Register |
| 0x01c0 2214 | ESRH | Event Set Register High |
| 0x01c0 2218 | CER | Chained Event Register |
| 0x01c0 221C | CERH | Chained Event Register High |
| 0x01c0 2220 | EER | Event Enable Register |
| 0x01c0 2224 | EERH | Event Enable Register High |
| 0x01c0 2228 | EECR | Event Enable Clear Register |
| 0x01c0 222C | EECRH | Event Enable Clear Register High |
| 0x01c0 2230 | EESR | Event Enable Set Register |
| 0x01c0 2234 | EESRH | Event Enable Set Register High |
| 0x01c0 2238 | SER | Secondary Event Register |
| 0x01c0 223C | SERH | Secondary Event Register High |

**Table 6-30. DM6446 EDMA3 Registers (continued)**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01c0 2240 | SECR | Secondary Event Clear Register |
| 0x01c0 2244 | SECRH | Secondary Event Clear Register High |
| 0x01c0 2248 - 0x01c0 224C | - | Reserved |
| 0x01c0 2250 | IER | Interrupt Enable Register |
| 0x01c0 2254 | IERH | Interrupt Enable Register High |
| 0x01c0 2258 | IECR | Interrupt Enable Clear Register |
| 0x01c0 225C | IECRH | Interrupt Enable Clear Register High |
| 0x01c0 2260 | IESR | Interrupt Enable Set Register |
| 0x01c0 2264 | IESRH | Interrupt Enable Set Register High |
| 0x01c0 2268 | IPR | Interrupt Pending Register |
| 0x01c0 226C | IPRH | Interrupt Pending Register High |
| 0x01c0 2270 | ICR | Interrupt Clear Register |
| 0x01c0 2274 | ICRH | Interrupt Clear Register High |
| 0x01c0 2278 | IEVAL | Interrupt Evaluate Register |
| 0x01c0 227C | - | Reserved |
| 0x01c0 2280 | QER | QDMA Event Register |
| 0x01c0 2284 | QEER | QDMA Event Enable Register |
| 0x01c0 2288 | QEECR | QDMA Event Enable Clear Register |
| 0x01c0 228C | QEESR | QDMA Event Enable Set Register |
| 0x01c0 2290 | QSER | QDMA Secondary Event Register |
| 0x01c0 2294 | QSECR | QDMA Secondary Event Clear Register |
| 0x01c0 2298 - 0x01c0 23FC | - | Reserved |
| **Shadow Region 2 Channel Registers** | | |
| 0x01c0 2400 | ER | Event Register |
| 0x01c0 2404 | ERH | Event Register High |
| 0x01c0 2408 | ECR | Event Clear Register |
| 0x01c0 240C | ECRH | Event Clear Register High |
| 0x01c0 2410 | ESR | Event Set Register |
| 0x01c0 2414 | ESRH | Event Set Register High |
| 0x01c0 2418 | CER | Chained Event Register |
| 0x01c0 241C | CERH | Chained Event Register High |
| 0x01c0 2420 | EER | Event Enable Register |
| 0x01c0 2424 | EERH | Event Enable Register High |
| 0x01c0 2428 | EECR | Event Enable Clear Register |
| 0x01c0 242C | EECRH | Event Enable Clear Register High |
| 0x01c0 2430 | EESR | Event Enable Set Register |
| 0x01c0 2434 | EESRH | Event Enable Set Register High |
| 0x01c0 2438 | SER | Secondary Event Register |
| 0x01c0 243C | SERH | Secondary Event Register High |
| 0x01c0 2440 | SECR | Secondary Event Clear Register |
| 0x01c0 2444 | SECRH | Secondary Event Clear Register High |
| 0x01c0 2448 - 0x01c0 244C | - | Reserved |
| 0x01c0 2450 | IER | Interrupt Enable Register |
| 0x01c0 2454 | IERH | Interrupt Enable Register High |
| 0x01c0 2458 | IECR | Interrupt Enable Clear Register |
| 0x01c0 245C | IECRH | Interrupt Enable Clear Register High |
| 0x01c0 2460 | IESR | Interrupt Enable Set Register |

**Table 6-30. DM6446 EDMA3 Registers (continued)**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01c0 2464 | IESRH | Interrupt Enable Set Register High |
| 0x01c0 2468 | IPR | Interrupt Pending Register |
| 0x01c0 246C | IPRH | Interrupt Pending Register High |
| 0x01c0 2470 | ICR | Interrupt Clear Register |
| 0x01c0 2474 | ICRH | Interrupt Clear Register High |
| 0x01c0 2478 | IEVAL | Interrupt Evaluate Register |
| 0x01c0 247C | - | Reserved |
| 0x01c0 2480 | QER | QDMA Event Register |
| 0x01c0 2484 | QEER | QDMA Event Enable Register |
| 0x01c0 2488 | QEECR | QDMA Event Enable Clear Register |
| 0x01c0 248C | QEESR | QDMA Event Enable Set Register |
| 0x01c0 2490 | QSER | QDMA Secondary Event Register |
| 0x01c0 2494 | QSECR | QDMA Secondary Event Clear Register |
| 0x01c0 2498 - 0x01c0 25FC | - | Reserved |
| **Shadow Region 3 Channel Registers** | | |
| 0x01c0 2600 | ER | Event Register |
| 0x01c0 2604 | ERH | Event Register High |
| 0x01c0 2608 | ECR | Event Clear Register |
| 0x01c0 260C | ECRH | Event Clear Register High |
| 0x01c0 2610 | ESR | Event Set Register |
| 0x01c0 2614 | ESRH | Event Set Register High |
| 0x01c0 2618 | CER | Chained Event Register |
| 0x01c0 261C | CERH | Chained Event Register High |
| 0x01c0 2620 | EER | Event Enable Register |
| 0x01c0 2624 | EERH | Event Enable Register High |
| 0x01c0 2628 | EECR | Event Enable Clear Register |
| 0x01c0 262C | EECRH | Event Enable Clear Register High |
| 0x01c0 2630 | EESR | Event Enable Set Register |
| 0x01c0 2634 | EESRH | Event Enable Set Register High |
| 0x01c0 2638 | SER | Secondary Event Register |
| 0x01c0 263C | SERH | Secondary Event Register High |
| 0x01c0 2640 | SECR | Secondary Event Clear Register |
| 0x01c0 2644 | SECRH | Secondary Event Clear Register High |
| 0x01c0 2648 - 0x01c0 264C | - | Reserved |
| 0x01c0 2650 | IER | Interrupt Enable Register |
| 0x01c0 2654 | IERH | Interrupt Enable Register High |
| 0x01c0 2658 | IECR | Interrupt Enable Clear Register |
| 0x01c0 265C | IECRH | Interrupt Enable Clear Register High |
| 0x01c0 2660 | IESR | Interrupt Enable Set Register |
| 0x01c0 2664 | IESRH | Interrupt Enable Set Register High |
| 0x01c0 2668 | IPR | Interrupt Pending Register |
| 0x01c0 266C | IPRH | Interrupt Pending Register High |
| 0x01c0 2670 | ICR | Interrupt Clear Register |
| 0x01c0 2674 | ICRH | Interrupt Clear Register High |
| 0x01c0 2678 | IEVAL | Interrupt Evaluate Register |
| 0x01c0 267C | - | Reserved |
| 0x01c0 2680 | QER | QDMA Event Register |

**Table 6-30. DM6446 EDMA3 Registers (continued)**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01c0 2684 | QEER | QDMA Event Enable Register |
| 0x01c0 2688 | QEECR | QDMA Event Enable Clear Register |
| 0x01c0 268C | QEESR | QDMA Event Enable Set Register |
| 0x01c0 2690 | QSER | QDMA Secondary Event Register |
| 0x01c0 2694 | QSECR | QDMA Secondary Event Clear Register |
| 0x01c0 2698 - 0x01c0 27FC | - | Reserved |
| 0x01c0 2800 - 0x01c0 29FC | - | Reserved |
| 0x01c0 2A00 - 0x01c0 2BFC | - | Reserved |
| 0x01c0 2C00 - 0x01c0 2DFC | - | Reserved |
| 0x01c0 2E00 - 0x01c0 2FFC | - | Reserved |
| 0x01c0 2FFD - 0x01c0 3FFF | - | Reserved |
| 0x01c0 4000 - 0x01c0 4FFF | - | Parameter Set RAM (see Table 6-31) |
| 0x01c0 5000 - 0x01c0 7FFF | - | Reserved |
| 0x01c0 8000 - 0x01c0 FFFF | - | Reserved |
| **Transfer Controller 0 Registers** | | |
| 0x01c1 0000 | - | Reserved |
| 0x01c1 0004 | TCCFG | EDMA3 TC0 Configuration Register |
| 0x01c1 0008 - 0x01c1 00FF | - | Reserved |
| 0x01c1 0100 | TCSTAT | EDMA3 TC0 Channel Status Register |
| 0x01c1 0104 - 0x01c1 0110 | - | Reserved |
| 0x01c1 0114 - 0x01c1 011F | - | Reserved |
| 0x01c1 0120 | ERRSTAT | EDMA3 TC0 Error Status Register |
| 0x01c1 0124 | ERREN | EDMA3 TC0 Error Enable Register |
| 0x01c1 0128 | ERRCLR | EDMA3 TC0 Error Clear Register |
| 0x01c1 012C | ERRDET | EDMA3 TC0 Error Details Register |
| 0x01c1 0130 | ERRCMD | EDMA3 TC0 Error Interrupt Command Register |
| 0x01c1 0134 - 0x01c1 013F | - | Reserved |
| 0x01c1 0140 | RDRATE | EDMA3 TC0 Read Rate Register |
| 0x01c1 0144 - 0x01c1 01FF | - | Reserved |
| 0x01c1 0200 - 0x01c1 023F | - | Reserved |
| 0x01c1 0240 | SAOPT | EDMA3 TC0 Source Active Options Register |
| 0x01c1 0244 | SASRC | EDMA3 TC0 Source Active Source Address Register |
| 0x01c1 0248 | SACNT | EDMA3 TC0 Source Active Count Register |
| 0x01c1 024C | SADST | EDMA3 TC0 Source Active Destination Address Register |
| 0x01c1 0250 | SABIDX | EDMA3 TC0 Source Active Source B-Index Register |
| 0x01c1 0254 | SAMPPRXY | EDMA3 TC0 Source Active Memory Protection Proxy Register |
| 0x01c1 0258 | SACNTRLD | EDMA3 TC0 Source Active Count Reload Register |
| 0x01c1 025C | SASRCBREF | EDMA3 TC0 Source Active Source Address B-Reference Register |
| 0x01c1 0260 | SADSTBREF | EDMA3 TC0 Source Active Destination Address B-Reference Register |
| 0x01c1 0264 - 0x01c1 027F | - | Reserved |
| 0x01c1 0280 | DFCNTRLD | EDMA3 TC0 Destination FIFO Set Count Reload Register |
| 0x01c1 0284 | DFSRCBREF | EDMA3 TC0 Destination FIFO Set Source Address B-Reference Register |
| 0x01c1 0288 | DFDSTBREF | EDMA3 TC0 Destination FIFO Set Destination Address B-Reference Register |
| 0x01c1 028C - 0x01c1 02FF | - | Reserved |
| 0x01c1 0300 | DFOPT0 | EDMA3 TC0 Destination FIFO Options Register 0 |
| 0x01c1 0304 | DFSRC0 | EDMA3 TC0 Destination FIFO Source Address Register 0 |
| 0x01c1 0308 | DFCNT0 | EDMA3 TC0 Destination FIFO Count Register 0 |

**Table 6-30. DM6446 EDMA3 Registers (continued)**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01c1 030C | DFDST0 | EDMA3 TC0 Destination FIFO Destination Address Register 0 |
| 0x01c1 0310 | DFBIDX0 | EDMA3 TC0 Destination FIFO BIDX Register 0 |
| 0x01c1 0314 | DFMPPRXY0 | EDMA3 TC0 Destination FIFO Memory Protection Proxy Register 0 |
| 0x01c1 0318 - 0x01c1 033F | - | Reserved |
| 0x01c1 0340 | DFOPT1 | EDMA3 TC0 Destination FIFO Options Register 1 |
| 0x01c1 0344 | DFSRC1 | EDMA3 TC0 Destination FIFO Source Address Register 1 |
| 0x01c1 0348 | DFCNT1 | EDMA3 TC0 Destination FIFO Count Register 1 |
| 0x01c1 034C | DFDST1 | EDMA3 TC0 Destination FIFO Destination Address Register 1 |
| 0x01c1 0350 | DFBIDX1 | EDMA3 TC0 Destination FIFO BIDX Register 1 |
| 0x01c1 0354 | DFMPPRXY1 | EDMA3 TC0 Destination FIFO Memory Protection Proxy Register 1 |
| 0x01c1 0358 - 0x01c1 037F | - | Reserved |
| 0x01c1 0380 | DFOPT2 | EDMA3 TC0 Destination FIFO Options Register 2 |
| 0x01c1 0384 | DFSRC2 | EDMA3 TC0 Destination FIFO Source Address Register 2 |
| 0x01c1 0388 | DFCNT2 | EDMA3 TC0 Destination FIFO Count Register 2 |
| 0x01c1 038C | DFDST2 | EDMA3 TC0 Destination FIFO Destination Address Register 2 |
| 0x01c1 0390 | DFBIDX2 | EDMA3 TC0 Destination FIFO BIDX Register 2 |
| 0x01c1 0394 | DFMPPRXY2 | EDMA3 TC0 Destination FIFO Memory Protection Proxy Register 2 |
| 0x01c1 0398 - 0x01c1 03BF | - | Reserved |
| 0x01c1 03C0 | DFOPT3 | EDMA3 TC0 Destination FIFO Options Register 3 |
| 0x01c1 03C4 | DFSRC3 | EDMA3 TC0 Destination FIFO Source Address Register 3 |
| 0x01c1 03C8 | DFCNT3 | EDMA3 TC0 Destination FIFO Count Register 3 |
| 0x01c1 03CC | DFDST3 | EDMA3 TC0 Destination FIFO Destination Address Register 3 |
| 0x01c1 03D0 | DFBIDX3 | EDMA3 TC0 Destination FIFO BIDX Register 3 |
| 0x01c1 03D4 | DFMPPRXY3 | EDMA3 TC0 Destination FIFO Memory Protection Proxy Register 3 |
| 0x01c1 03D8 - 0x01c1 03FF | - | Reserved |
| **Transfer Controller 1 Registers** | | |
| 0x01c1 0400 | - | Reserved |
| 0x01c1 0404 | TCCFG | EDMA3 TC1 Configuration Register |
| 0x01c1 0408 - 0x01c1 04FF | - | Reserved |
| 0x01c1 0500 | TCSTAT | EDMA3 TC1 Channel Status Register |
| 0x01c1 0504 - 0x01c1 0510 | - | Reserved |
| 0x01c1 0514 - 0x01c1 051F | - | Reserved |
| 0x01c1 0520 | ERRSTAT | EDMA3 TC1 Error Status Register |
| 0x01c1 0524 | ERREN | EDMA3 TC1 Error Enable Register |
| 0x01c1 0528 | ERRCLR | EDMA3 TC1 Error Clear Register |
| 0x01c1 052C | ERRDET | EDMA3 TC1 Error Details Register |
| 0x01c1 0530 | ERRCMD | EDMA3 TC1 Error Interrupt Command Register |
| 0x01c1 0534 - 0x01c1 053F | - | Reserved |
| 0x01c1 0540 | RDRATE | EDMA3 TC1 Read Rate Register |
| 0x01c1 0544 - 0x01c1 05FF | - | Reserved |
| 0x01c1 0600 - 0x01c1 063F | - | Reserved |
| 0x01c1 0640 | SAOPT | EDMA3 TC1 Source Active Options Register |
| 0x01c1 0644 | SASRC | EDMA3 TC1 Source Active Source Address Register |
| 0x01c1 0648 | SACNT | EDMA3 TC1 Source Active Count Register |
| 0x01c1 064C | SADST | EDMA3 TC1 Source Active Destination Address Register |
| 0x01c1 0650 | SABIDX | EDMA3 TC1 Source Active Source B-Index Register |
| 0x01c1 0654 | SAMPPRXY | EDMA3 TC1 Source Active Memory Protection Proxy Register |

**Table 6-30. DM6446 EDMA3 Registers (continued)**

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01c1 0658 | SACNTRLD | EDMA3 TC1 Source Active Count Reload Register |
| 0x01c1 065C | SASRCBREF | EDMA3 TC1 Source Active Source Address B-Reference Register |
| 0x01c1 0660 | SADSTBREF | EDMA3 TC1 Source Active Destination Address B-Reference Register |
| 0x01c1 0664 - 0x01c1 067F | - | Reserved |
| 0x01c1 0680 | DFCNTRLD | EDMA3 TC1 Destination FIFO Set Count Reload Register |
| 0x01c1 0684 | DFSRCBREF | EDMA3 TC1 Destination FIFO Set Source Address B-Reference Register |
| 0x01c1 0688 | DFDSTBREF | EDMA3 TC1 Destination FIFO Set Destination Address B-Reference Register |
| 0x01c1 068C - 0x01c1 06FF | - | Reserved |
| 0x01c1 0700 | DFOPT0 | EDMA3 TC1 Destination FIFO Options Register 0 |
| 0x01c1 0704 | DFSRC0 | EDMA3 TC1 Destination FIFO Source Address Register 0 |
| 0x01c1 0708 | DFCNT0 | EDMA3 TC1 Destination FIFO Count Register 0 |
| 0x01c1 070C | DFDST0 | EDMA3 TC1 Destination FIFO Destination Address Register 0 |
| 0x01c1 0710 | DFBIDX0 | EDMA3 TC1 Destination FIFO BIDX Register 0 |
| 0x01c1 0714 | DFMPPRXY0 | EDMA3 TC1 Destination FIFO Memory Protection Proxy Register 0 |
| 0x01c1 0718 - 0x01c1 073F | - | Reserved |
| 0x01c1 0740 | DFOPT1 | EDMA3 TC1 Destination FIFO Options Register 1 |
| 0x01c1 0744 | DFSRC1 | EDMA3 TC1 Destination FIFO Source Address Register 1 |
| 0x01c1 0748 | DFCNT1 | EDMA3 TC1 Destination FIFO Count Register 1 |
| 0x01c1 074C | DFDST1 | EDMA3 TC1 Destination FIFO Destination Address Register 1 |
| 0x01c1 0750 | DFBIDX1 | EDMA3 TC1 Destination FIFO BIDX Register 1 |
| 0x01c1 0754 | DFMPPRXY1 | EDMA3 TC1 Destination FIFO Memory Protection Proxy Register 1 |
| 0x01c1 0758 - 0x01c1 077F | - | Reserved |
| 0x01c1 0780 | DFOPT2 | EDMA3 TC1 Destination FIFO Options Register 2 |
| 0x01c1 0784 | DFSRC2 | EDMA3 TC1 Destination FIFO Source Address Register 2 |
| 0x01c1 0788 | DFCNT2 | EDMA3 TC1 Destination FIFO Count Register 2 |
| 0x01c1 078C | DFDST2 | EDMA3 TC1 Destination FIFO Destination Address Register 2 |
| 0x01c1 0790 | DFBIDX2 | EDMA3 TC1 Destination FIFO BIDX Register 2 |
| 0x01c1 0794 | DFMPPRXY2 | EDMA3 TC1 Destination FIFO Memory Protection Proxy Register 2 |
| 0x01c1 0798 - 0x01c1 07BF | - | Reserved |
| 0x01c1 07C0 | DFOPT3 | EDMA3 TC1 Destination FIFO Options Register 3 |
| 0x01c1 07C4 | DFSRC3 | EDMA3 TC1 Destination FIFO Source Address Register 3 |
| 0x01c1 07C8 | DFCNT3 | EDMA3 TC1 Destination FIFO Count Register 3 |
| 0x01c1 07CC | DFDST3 | EDMA3 TC1 Destination FIFO Destination Address Register 3 |
| 0x01c1 07D0 | DFBIDX3 | EDMA3 TC1 Destination FIFO BIDX Register 3 |
| 0x01c1 07D4 | DFMPPRXY3 | EDMA3 TC1 Destination FIFO Memory Protection Proxy Register 3 |
| 0x01c1 07D8 - 0x01c1 07FF | - | Reserved |

Table 6-31 shows an abbreviation of the set of registers which make up the parameter set for each of 128 EDMA3 events. Each of the parameter register sets consist of 8 32-bit word entries. Table 6-32 shows the parameter set entry registers with relative memory address locations within each of the parameter sets.

**Table 6-31. EDMA3 Parameter Set RAM**

| HEX ADDRESS RANGE | DESCRIPTION |
|---|---|
| 0x01c0 4000 - 0x01c0 401F | Parameters Set 0 (8 32-bit words) |
| 0x01c0 4020 - 0x01c0 403F | Parameters Set 1 (8 32-bit words) |
| 0x01c0 4040 - 0x01c0 405F | Parameters Set 2 (8 32-bit words) |
| 0x01c0 4060 - 0x01c0 407F | Parameters Set 3 (8 32-bit words) |
| 0x01c0 4080 - 0x01c0 409F | Parameters Set 4 (8 32-bit words) |
| 0x01c0 40A0 - 0x01c0 40BF | Parameters Set 5 (8 32-bit words) |
| ... | ... |
| 0x01c0 4FC0 - 0x01c0 4FDF | Parameters Set 126 (8 32-bit words) |
| 0x01c0 4FE0 - 0x01c0 4FFF | Parameters Set 127 (8 32-bit words) |

**Table 6-32. Parameter Set Entries**

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

## 6.10 External Memory Interface (EMIF)

DM6446 supports several memory and external device interfaces, including:

- Asynchronous EMIF (EMIFA) for interfacing to NOR Flash, SRAM, etc.
- NAND Flash
- ATA/CF

### *6.10.1 Asynchronous EMIF (EMIFA)*

The DM6446 Asynchronous EMIF (EMIFA) provides an 8-bit or 16-bit data bus, an address bus width up to 24-bits, and 4 dedicated chip selects, along with memory control signals. These signals are multiplexed between three peripherals:

- EMIFA and NAND interfaces
- ATA/CF
- Host Port Interface

#### 6.10.1.1 NAND (NAND, SmartMedia, xD)

The EMIFA interface provides both the asynchronous EMIF and NAND interfaces. Four chip selects are provided and each are individually configurable to provide either EMIFA or NAND support. The NAND features supported are as follows.

- NAND flash on up to 4 asynchronous chip selects.
- 8 and 16-bit data bus widths.
- Programmable cycle timings.
- Performs ECC calculation.
- NAND Mode also supports SmartMedia/SSFDC (Solid State Floppy Disk Controller) and xD memory cards
- ARM ROM supports booting of the DM6446 ARM processor from NAND flash located at CS2

The memory map for EMIFA and NAND registers is shown in Table 6-33. For more details on the EMIFA and NAND interfaces, see the *TMS320DM644x DMSoC Peripherals Overview Reference Guide* (literature number SPRUE19) and the *TMS320DM644x DMSoC Asynchronous External Memory Interface (EMIF) Reference Guide* (literature number SPRUE20).

**Table 6-33. EMIFA/NAND Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01E0 0000 - 0x01E0 0003 | | Reserved |
| 0x01E0 0004 | AWCCR | Asynchronous Wait Cycle Configuration Register |
| 0x01E0 0008 - 0x01E0 000F | | Reserved |
| 0x01E0 0010 | A1CR | Asynchronous 1 Configuration Register (CS2 Space) |
| 0x01E0 0014 | A2CR | Asynchronous 2 Configuration Register (CS3 Space) |
| 0x01E0 0018 | A3CR | Asynchronous 3 Configuration Register (CS4 Space) |
| 0x01E0 001C | A4CR | Asynchronous 4 Configuration Register (CS5 Space) |
| 0x01E0 0020 - 0x01E0 003F | - | Reserved |
| 0x01E0 0040 | EIRR | EMIF Interrupt Raw Register |
| 0x01E0 0044 | EIMR | EMIF Interrupt Mask Register |
| 0x01E0 0048 | EIMSR | EMIF Interrupt Mask Set Register |
| 0x01E0 004C | EIMCR | EMIF Interrupt Mask Clear Register |
| 0x01E0 0050 - 0x01E0 005F | - | Reserved |
| 0x01E0 0060 | NANDFCR | NAND Flash Control Register |
| 0x01E0 0064 | NANDFSR | NAND Flash Status Register |
| 0x01E0 0070 | NANDF1ECC | NAND Flash 1 ECC Register (CS2 Space) |
| 0x01E0 0074 | NANDF2ECC | NAND Flash 2 ECC Register (CS3 Space) |
| 0x01E0 0078 | NANDF3ECC | NAND Flash 3 ECC Register (CS4 Space) |
| 0x01E0 007C | NANDF4ECC | NAND Flash 4 ECC Register (CS5 Space) |
| 0x01E0 0080 - 0x01E0 0FFF | - | Reserved |

### 6.10.1.2 EMIFA Electrical Data/Timing

**Table 6-34. Timing Requirements for Asynchronous Memory Cycles for EMIFA Module(1) (see Figure 6-21 and Figure 6-22)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| **READS and WRITES** | | | | | |
| 2 | $t_{w(EM\_WAIT)}$ | Pulse duration, EM_WAIT assertion and deassertion | 2E | | ns |
| **READS** | | | | | |
| 12 | $t_{su(EMDV-EMOEH)}$ | Setup time, EM_D[15:0] valid before $\overline{EM\_OE}$ high | 10.5 | | ns |
| 13 | $t_{h(EMOEH-EMDIV)}$ | Hold time, EM_D[15:0] valid after $\overline{EM\_OE}$ high | 0 | | ns |
| 14 | $t_{su\ (EMWAIT-EMOEH)}$ | Setup time, EM_WAIT asserted before $\overline{EM\_OE}$ high(2) | 4E + 10.4 | | ns |
| **WRITES** | | | | | |
| 28 | $t_{\ su(EMWAIT-EMWEH)}$ | Setup time, EM_WAIT asserted before $\overline{EM\_WE}$ high(2) | 4E + 10.4 | | ns |

(1) E = SYSCLK5 period in ns for EMIFA. For example, when running the DSP CPU at 594 MHz, use E = 10.1 ns. For example, when running the DSP CPU at 810 MHz, use E = 7.4 ns.

(2) Setup before end of STROBE phase (if no extended wait states are inserted) by which EM_WAIT must be asserted to add extended wait states. Figure 6-23 and Figure 6-24 describe EMIF transactions that include extended wait states inserted during the STROBE phase. However, cycles inserted as part of this extended wait period should not be counted; the 4E requirement is to the start of where the HOLD phase would begin if there were no extended wait cycles.

**Table 6-35. Switching Characteristics Over Recommended Operating Conditions for Asynchronous Memory Cycles for EMIFA Module(1) (2) (see Figure 6-21 and Figure 6-22)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| **READS and WRITES** | | | | | |
| 1 | $t_{d(TURNAROUND)}$ | Turn-around time | (TA + 1) * E - 2 | (TA + 1) * E + 2 | ns |
| **READS** | | | | | |
| 3 | $t_{c(EMRCYCLE)}$ | EMIF read cycle time (EW = 0) | (RS + RST + RH + TA + 4) * E - 0.5 | (RS + RST + RH + TA + 4) * E + 0.5 | ns |
| | | EMIF read cycle time (EW = 1) | (RS + RST + RH + TA + 4) * E - 0.5 | 4184 * E + 0.5 | ns |
| 4 | $t_{su(EMCSL-EMOEL)}$ | Output setup time, $\overline{EM\_CS[5:2]}$ low to $\overline{EM\_OE}$ low (SS = 0) | (RS + 1) * E - 1 | (RS + 1) * E + 1.4 | ns |
| | | Output setup time, $\overline{EM\_CS[5:2]}$ low to $\overline{EM\_OE}$ low (SS = 1) | -1 | | ns |
| 5 | $t_{h(EMOEH-EMCSH)}$ | Output hold time, $\overline{EM\_OE}$ high to $\overline{EM\_CS[5:2]}$ high (SS = 0) | (RH + 1) * E - 2.1 | (RH + 1) * E + 1.4 | ns |
| | | Output hold time, $\overline{EM\_OE}$ high to $\overline{EM\_CS[5:2]}$ high (SS = 1) | -2.2 | | ns |
| 6 | $t_{su(EMBAV-EMOEL)}$ | Output setup time, $\overline{EM\_BA[1:0]}$ valid to $\overline{EM\_OE}$ low | (RS + 1) * E - 1.8 | (RS + 1) * E + 1.3 | ns |
| 7 | $t_{h(EMOEH-EMBAIV)}$ | Output hold time, $\overline{EM\_OE}$ high to $\overline{EM\_BA[1:0]}$ invalid | (RH + 1) * E - 2.3 | (RH + 1) * E + 1.1 | ns |
| 8 | $t_{su(EMAV-EMOEL)}$ | Output setup time, $\overline{EM\_A[21:0]}$ valid to $\overline{EM\_OE}$ low | (RS + 1) * E - 1.9 | (RS + 1) * E + 1.5 | ns |
| 9 | $t_{h(EMOEH-EMAIV)}$ | Output hold time, $\overline{EM\_OE}$ high to $\overline{EM\_A[21:0]}$ invalid | (RH + 1) * E - 2.6 | (RH + 1) * E + 1.2 | ns |
| 10 | $t_{w(EMOEL)}$ | $\overline{EM\_OE}$ active low width (EW = 0) | (RST + 1) * E - 2 | (RST + 1) * E + 2 | ns |
| | | $\overline{EM\_OE}$ active low width (EW = 1) | (RST + 1) * E - 2 | (RST + 4097) * E + 2 | ns |
| 11 | $t_{d(EMWAITH-EMOEH)}$ | Delay time from EM_WAIT deasserted to $\overline{EM\_OE}$ high | | 4E + 10.4 | ns |
| **WRITES** | | | | | |
| 15 | $t_{c(EMWCYCLE)}$ | EMIF write cycle time (EW = 0) | (WS + WST + WH + TA + 4) * E - 0.5 | (WS + WST + WH + TA + 4) * E + 0.5 | ns |
| | | EMIF write cycle time (EW = 1) | (WS + WST + WH + TA + 4) * E -0.5 | 4184 * E + 0.5 | ns |
| 16 | $t_{su(EMCSL-EMWEL)}$ | Output setup time, $\overline{EM\_CS[5:2]}$ low to $\overline{EM\_WE}$ low (SS = 0) | (WS + 1) * E - 0.9 | (WS + 1) * E + 1.4 | ns |
| | | Output setup time, $\overline{EM\_CS[5:2]}$ low to $\overline{EM\_WE}$ low (SS = 1) | -1 | | ns |
| 17 | $t_{h(EMWEH-EMCSH)}$ | Output hold time, $\overline{EM\_WE}$ high to $\overline{EM\_CS[5:2]}$ high (SS = 0) | (WH + 1) * E - 2.1 | (WH + 1) * E + 1.1 | ns |
| | | Output hold time, $\overline{EM\_WE}$ high to $\overline{EM\_CS[5:2]}$ high (SS = 1) | -2.1 | | ns |
| 18 | $t_{su(EMRNW-EMWEL)}$ | Output setup time, EM_R/$\overline{W}$ valid to $\overline{EM\_WE}$ low | (WS + 1) * E - 0.7 | (WS + 1) * E + 0.9 | ns |
| 19 | $t_{h(EMWEH-EMRNW)}$ | Output hold time, $\overline{EM\_WE}$ high to EM_R/$\overline{W}$ invalid | (WH + 1) * E - 0.9 | (WH + 1) * E + 0.9 | ns |
| 20 | $t_{su(EMBAV-EMWEL)}$ | Output setup time, $\overline{EM\_BA[1:0]}$ valid to $\overline{EM\_WE}$ low | (WS + 1) * E - 1.7 | (WS + 1) * E + 1.5 | ns |
| 21 | $t_{h(EMWEH-EMBAIV)}$ | Output hold time, $\overline{EM\_WE}$ high to $\overline{EM\_BA[1:0]}$ invalid | (WH + 1) * E - 2.3 | (WH + 1) * E + 0.9 | ns |
| 22 | $t_{su(EMAV-EMWEL)}$ | Output setup time, $\overline{EM\_A[21:0]}$ valid to $\overline{EM\_WE}$ low | (WS + 1) * E - 1.8 | (WS + 1) * E + 1.7 | ns |
| 23 | $t_{h(EMWEH-EMAIV)}$ | Output hold time, $\overline{EM\_WE}$ high to $\overline{EM\_A[21:0]}$ invalid | (WH + 1) * E - 2.6 | (WH + 1) * E + 1 | ns |
| 24 | $t_{w(EMWEL)}$ | $\overline{EM\_WE}$ active low width (EW = 0) | (WST + 1) * E - 2 | (WST + 1) * E + 2 | ns |
| | | $\overline{EM\_WE}$ active low width (EW = 1) | (WST + 1) * E - 2 | (WST + 4097) * E + 2 | ns |

(1) RS = Read setup, RST = Read STrobe, RH = Read Hold, WS = Write Setup, WST = Write STrobe, WH = Write Hold, TA = Turn Around, EW = Extend Wait mode, SS = Select Strobe mode. These parameters are programmed via the Asynchronous Bank and Asynchronous Wait Cycle Configuration Registers and support the following range of values: TA[3:0], RS[15:0], RST[63:0], RH[7:0], WS[15:0], WST[63:0], WH[7:0], and EW[255:0]. For more information, see the *TMS320DM644x DMSoC Asynchronous External Memory Interface (EMIF) Reference Guide* (literature number SPRUE20).

(2) E = SYSCLK5 period in ns for EMIFA. For example, when running the DSP CPU at 594 MHz, use E = 10.1 ns. For example, when running the DSP CPU at 810 MHz, use E = 7.4 ns.

**Table 6-35. Switching Characteristics Over Recommended Operating Conditions for Asynchronous Memory Cycles for EMIFA Module (see Figure 6-21 and Figure 6-22) (continued)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 25 | $t_{d(EMWAITH-EMWEH)}$ | Delay time from EM_WAIT deasserted to $\overline{EM\_WE}$ high | | 4E + 10.4 | ns |
| 26 | $t_{su(EMDV-EMWEL)}$ | Output setup time, $\overline{EM\_D[15:0]}$ valid to $\overline{EM\_WE}$ low | (WS + 1) * E - 2.2 | (WS + 1) * E + 1.4 | ns |
| 27 | $t_{h(EMWEH-EMDIV)}$ | Output hold time, $\overline{EM\_WE}$ high to $\overline{EM\_D[15:0]}$ invalid | (WH + 1) * E - 2.2 | (WH + 1) * E + 1.4 | ns |

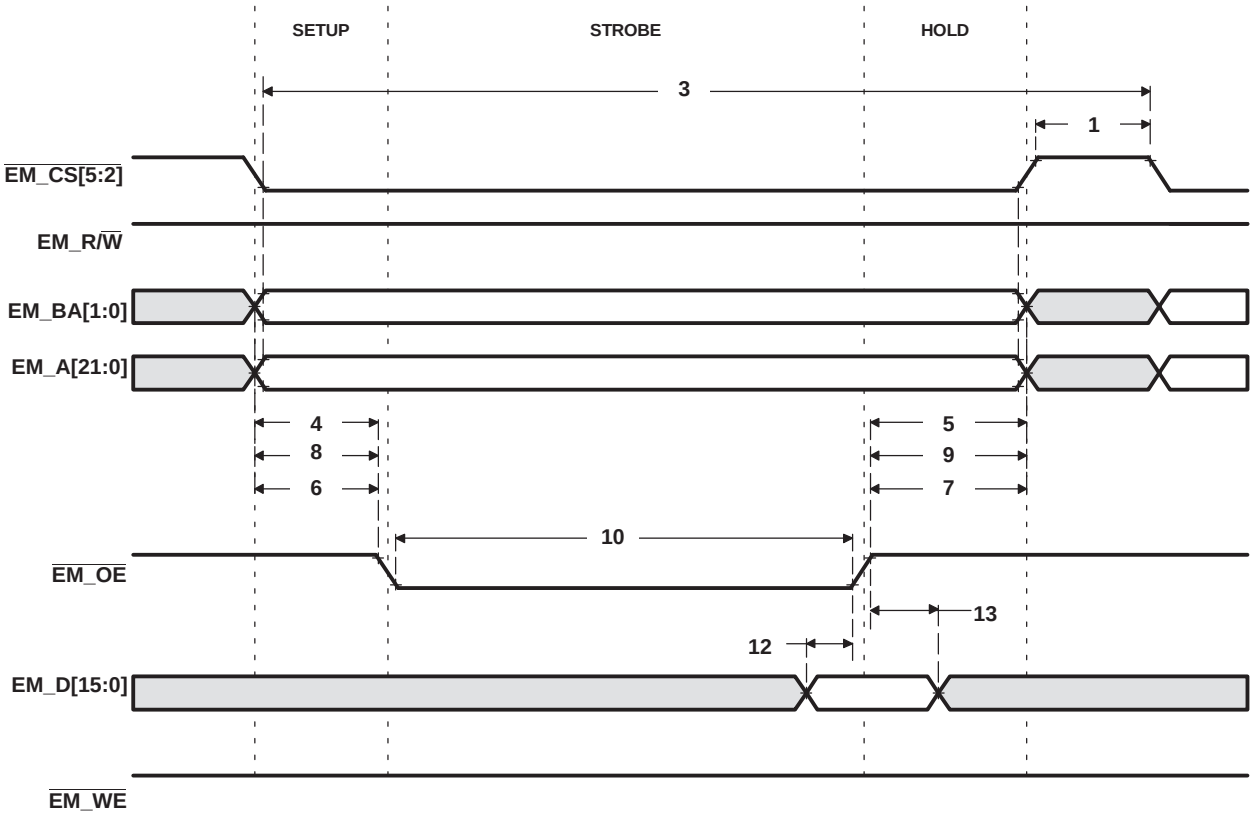



**Figure 6-21. Asynchronous Memory Read Timing for EMIF**

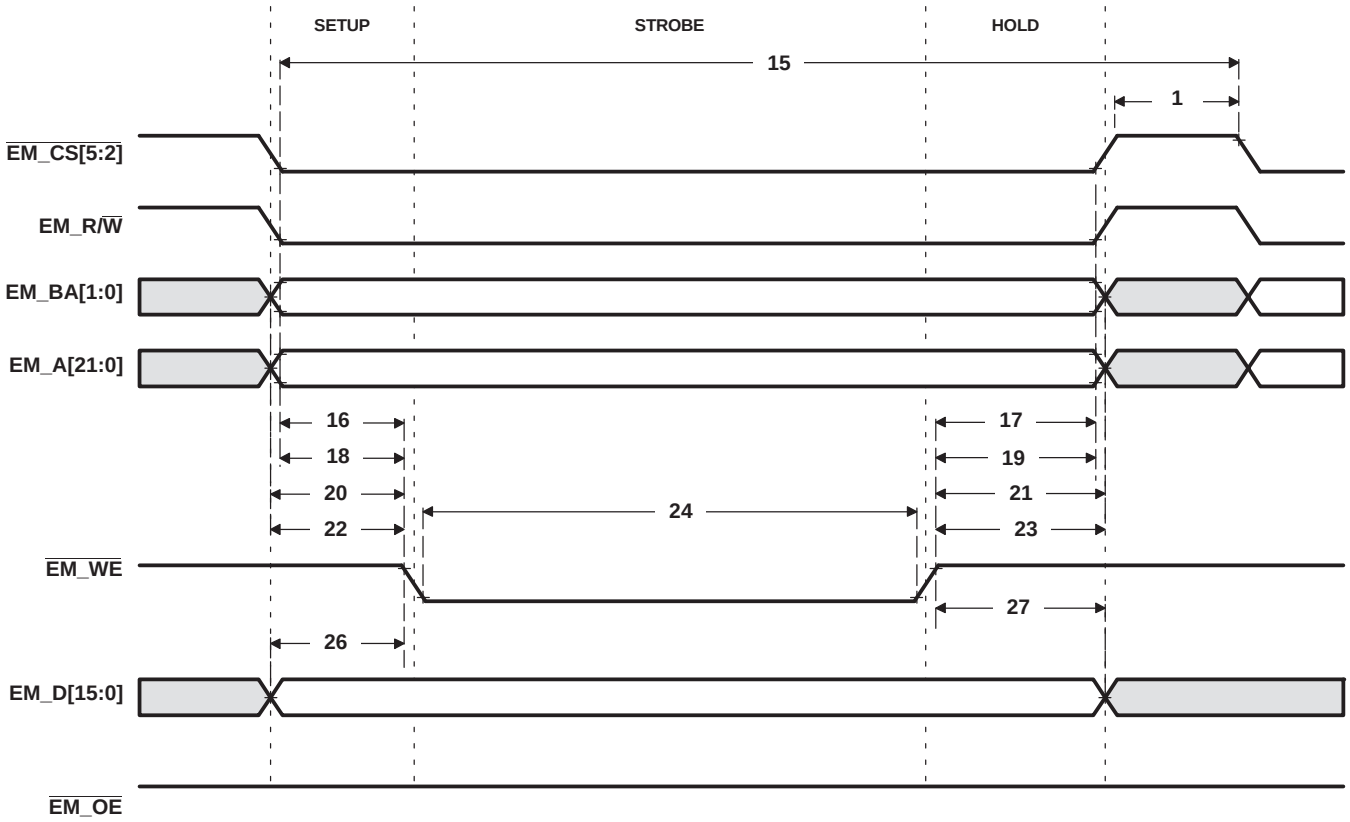



**Figure 6-22. Asynchronous Memory Write Timing for EMIF**

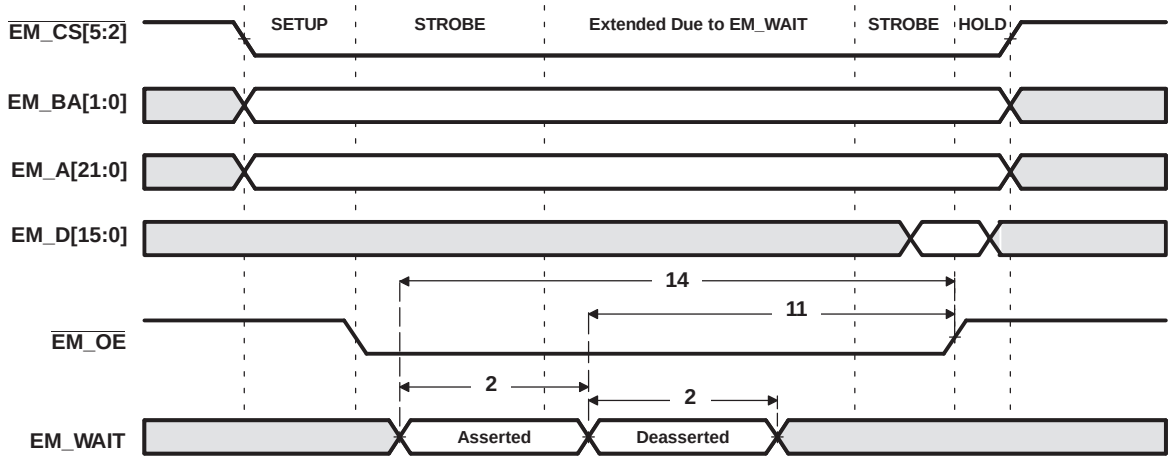



**Figure 6-23. EM_WAIT Read Timing Requirements**

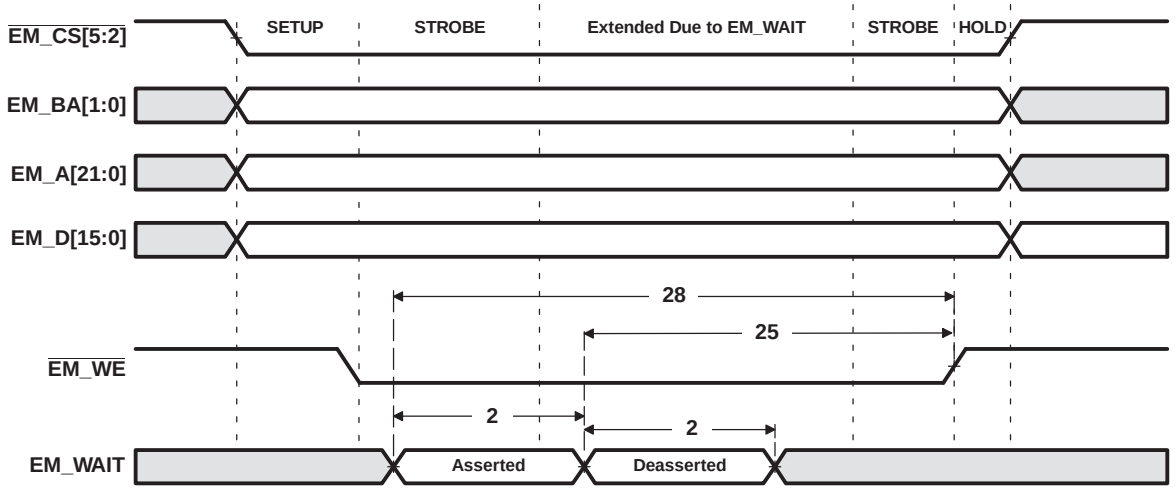



**Figure 6-24. EM_WAIT Write Timing Requirements**

### 6.10.2 DDR2 Memory Controller

The DDR2 Memory Controller is a dedicated interface to DDR2 SDRAM. It supports JESD79D-2A standard compliant DDR2 SDRAM Devices and can interface to either 16-bit or 32-bit DDR2 SDRAM devices. For details on the DDR2 Memory Controller, see the *TMS320DM644x DMSoC Peripherals Overview Reference Guide* (literature number SPRUE19) and the *TMS320DM644x DMSoC DDR2 Memory Controller User's Guide* (literature number SPRUE22).

DDR2 SDRAM plays a key role in a DaVinci-based system. Such a system is expected to require a significant amount of high-speed external memory for:

- Buffering of input image data from sensors or video sources
- Intermediate buffering for processing/resizing of image data in the VPFE
- Numerous OSD display buffers
- Intermediate buffering for large raw Bayer data image files while performing image processing functions
- Buffering for intermediate data while performing video encode and decode functions
- Storage of executable code for both the ARM and DSP

A memory map of the DDR2 Memory Controller registers is shown in Table 6-36.

**Table 6-36. DDR2 Memory Controller Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C4 004C | DDRVTPER | DDR2 VTP Enable Register |
| 0x01C4 2030 | DDRVTPR | DDR2 VTP Register |
| 0x2000 0000 - 0x2000 0003 | - | Reserved |
| 0x2000 0004 | SDRSTAT | SDRAM Status Register |
| 0x2000 0008 | SDBCR | SDRAM Bank Configuration Register |
| 0x2000 000C | SDRCR | SDRAM Refresh Control Register |
| 0x2000 0010 | SDTIMR | SDRAM Timing Register |
| 0x2000 0014 | SDTIMR2 | SDRAM Timing Register 2 |
| 0x2000 0020 | PBBPR | Peripheral Bus Burst Priority Register |
| 0x2000 0024 - 0x2000 00BF | - | Reserved |
| 0x2000 00C0 | IRR | Interrupt Raw Register |
| 0x2000 00C4 | IMR | Interrupt Masked Register |
| 0x2000 00C8 | IMSR | Interrupt Mask Set Register |

**Table 6-36. DDR2 Memory Controller Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x2000 00CC | IMCR | Interrupt Mask Clear Register |
| 0x2000 00D0 - 0x2000 00E3 | - | Reserved |
| 0x2000 00E4 | DDRPHYCR | DDR PHY Control Register |
| 0x2000 00E8 - 0x2000 00EF | - | Reserved |
| 0x2000 00F0 | VTPIOCR | VTP IO Control Register |
| 0x2000 00F4 - 0x2000 7FFF | - | Reserved |

### 6.10.2.1 DDR2 Memory Controller Electrical Data/Timing

The *Implementing DDR2 PCB Layout on the TMS320DM644x DSP* application report (literature number SPRAAC5) specifies a complete DDR2 interface solution for the DM6446 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.

TI only supports board designs that follow the guidelines outlined in the *Implementing DDR2 PCB Layout on the TMS320DM644x DSP* application report (literature number SPRAAC5).

**Table 6-37. Switching Characteristics Over Recommended Operating Conditions for DDR2 Memory Controller(1) (2)(see Figure 6-25)**

| NO. | PARAMETER | | -810 | | A-513, -594 | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(DDR\_CLK0)}$ | Cycle time, DDR_CLK0 | 5.29 | 8 | 6 | 8 | ns |
| 2 | $f_{(DDR\_CLK0)}$ | Frequency, DDR_CLK0 | 125 | 189 | 125 | 167 | MHz |

(1) DDR_CLK0 cycle time = 2 x PLL2 - SYSCLK2 cycle time.
(2) The PLL2 Controller ***must*** be programmed such that the resulting DDR_CLK0 clock frequency is within the specified range.

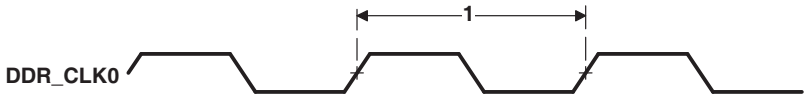



**Figure 6-25. DDR2 Memory Controller Clock Timing**

## 6.11 ATA/CF

The ATA/CF peripheral supports the following features:

- PIO, multiword DMA, and Ultra ATA 33/66
- Up to mode 4 timings on PIO mode
- Up to mode 2 timings on multiword DMA
- Up to mode 4 timings on Ultra ATA
- Programmable timing parameters
- Supports TrueIDE mode for Compact Flash

In addition, the Host IDE Controller supports multiword DMA transfers between external IDE/ATAPI devices and a system memory bus interface.

### *6.11.1 ATA/CF Peripheral Register Description(s)*

The ATA registers are shown in Table 6-38.

**Table 6-38. ATA Register Memory Map**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| **ATA Bus Master Interface DMA Engine Registers** | | |
| 0x01C6 6000 | BMICP | Primary IDE Channel DMA Control Register |
| 0x01C6 6002 | BMISP | Primary IDE Channel DMA Status Register |
| 0x01C6 6004 | BMIDTP | Primary IDE Channel DMA Descriptor Table Pointer Register |
| 0x01C6 6008 | - | Reserved |
| 0x01C6 600A | - | |
| 0x01C6 600C | - | |
| **ATA Configuration Registers** | | |
| 0x01C6 6040 | IDETIMP | Primary IDE Channel Timing Register |
| 0x01C6 6042 | - | Reserved |
| 0x01C6 6044 | - | |
| 0x01C6 6045 | - | |
| 0x01C6 6047 | IDESTAT | IDE Controller Status Register |
| 0x01C6 6048 | UDMACTL | Ultra-DMA Control Register |
| 0x01C6 604A | - | Reserved |
| 0x01C6 6050 | MISCCTL | Miscellaneous Control Register |
| 0x01C6 6054 | REGSTB | Task File Register Strobe Timing Register |
| 0x01C6 6058 | REGRCVR | Task File Register Recovery Timing Register |
| 0x01C6 605C | DATSTB | Data Register Access PIO Strobe Timing Register |
| 0x01C6 6060 | DATRCVR | Data Register Access PIO Recovery Timing Register |
| 0x01C6 6064 | DMASTB | Multiword DMA Strobe Timing Register |
| 0x01C6 6068 | DMARCVR | Multiword DMA Recovery Timing Register |
| 0x01C6 606C | UDMASTB | Ultra-DMA Strobe Timing Register |
| 0x01C6 6070 | UDMATRP | Ultra-DMA Ready-to-Pause Timing Register |
| 0x01C6 6074 | UDMATENV | Ultra-DMA Timing Envelope Register |
| 0x01C6 6078 | IORDYTMP | Primary IO Ready Timer Configuration Register |
| 0x01C6 607C - 0x01C6 67FF | - | Reserved |

### 6.11.2 ATA/CF Electrical Data/Timing

All ATA/CF AC timing data described in Section 6.11.2.1 – Section 6.11.2.3 is provided at the DM6446 device pins. For more details, see Section 6.1, *Parameter Information*.

The AC timing specifications described in Section 6.11.2.1 – Section 6.11.2.3 assume correct configuration of the ATA/CF memory-mapped control registers for the selected ATA/CF frequency of operation.

#### 6.11.2.1 ATA/CF PIO Data Transfer AC Timing

**Table 6-39. Timings for ATA/CF Module — PIO Data Transfer(1) (2) (see Figure 6-26)**

| NO. | | | MODE | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 1 | $t_0$ | Cycle time | 0-4(3) | (DATSTB + DATRCVR + 2)P - 0.5 | | ns |
| 2 | $t_1$ | Address valid to $\overline{\text{DIOW}}$/ $\overline{\text{DIOR}}$ setup | 0-4(3) | 12P - 1.6 | | ns |
| 3 | $t_2$ | $\overline{\text{DIOW}}$/ $\overline{\text{DIOR}}$ pulse duration low | 0-4(3) | (DATSTB + 1)P - 1 | | ns |
| 4 | $t_{2i}$ | $\overline{\text{DIOW}}$/$\overline{\text{DIOR}}$ recovery time, pulse duration high | 0-2 | – | | ns |
| | | | 3-4(3) | (DATRCVR + 1)P - 1 | | ns |
| 5 | $t_3$ | $\overline{\text{DIOW}}$ data setup time, DD[15:0] valid before $\overline{\text{DIOW}}$ rising edge | 0-4(3) | (DATSTB + 1)P | | ns |
| 6 | $t_4$ | $\overline{\text{DIOW}}$ data hold time, DD[15:0] valid after $\overline{\text{DIOW}}$ rising edge | 0-4(3) | (HWNHLD + 1)P + 1 | | ns |
| 7 | $t_5$ | $\overline{\text{DIOR}}$ data setup time, DD[15:0] valid before $\overline{\text{DIOR}}$ rising edge | 0 | 50 | | ns |
| | | | 1 | 35 | | ns |
| | | | 2-4(3) | 20 | | ns |
| 8 | $t_6$ | $\overline{\text{DIOR}}$ data hold time, DD[15:0] valid after $\overline{\text{DIOR}}$ rising edge | 0-4(3) | 5 | | ns |
| 9 | $t_{6Z}$ | Output data 3-state, DD[15:0] 3-state after $\overline{\text{DIOR}}$ rising edge | 0-4(3) | | 30 | ns |
| 10 | $t_9$ | $\overline{\text{DIOW}}$/$\overline{\text{DIOR}}$ to address valid hold | 0-4(3) | (HWNHLD + 1)P - 2.1 | | ns |
| 11 | $t_{RD}$ | Read data setup time, DD[15:0] valid before IORDY active | 0-4(3) | 0 | | ns |
| 12 | $t_A$ | IORDY setup | 0-4(3) (4) | | 35 | ns |
| 13 | $t_B$ | IORDY pulse width | 0-4(3) | | 1250 | ns |
| 14 | $t_C$ | IORDY assertion to release | 0-4(3) | | 5 | ns |

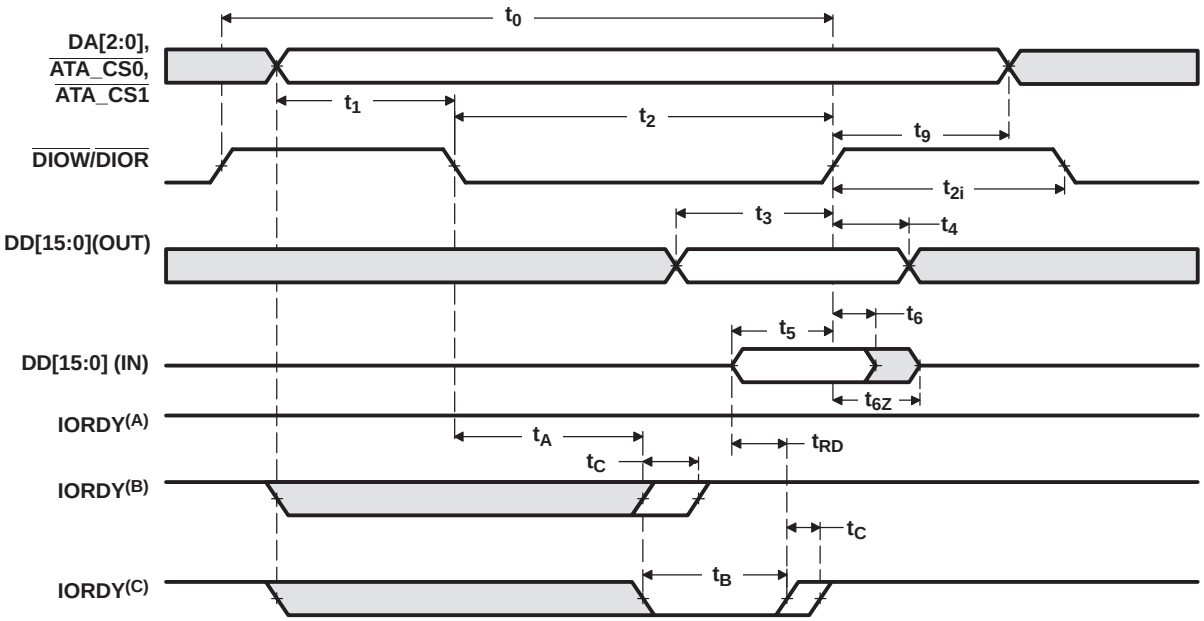
(1) P = SYSCLK5 period, in ns, for ATA. For example, when running the DSP CPU at 594 MHz, use P = 10.1 ns. For example, when running the DSP CPU at 810 MHz, use P = 7.4 ns.

(2) DATSTB equals the value programmed in the DATSTBxP bit field in the DATSTB register. DATRCVR equals the value programmed in the DATRCVRxP bit field in the DATRCVR register. HWNHLD equals the value programmed in the HWNHLDxP bit field in the MISCCTL register. For more detailed information, see the *TMS320DM644x DMSoC ATA Controller User's Guide* (literature number SPRUE21).

(3) The sustained throughput for PIO modes 3 and 4 is limited to the throughput equivalent of PIO mode 2. For more detailed information, see the *TMS320DM644x DMSoC ATA Controller User's Guide* (literature number SPRUE21).

(4) The $t_A$ parameter must be met only when the IORDY timer is enabled to allow a device to insert wait states during a transaction. In order to meet the $t_A$ parameter, a minimum frequency for SYSCLK5 is specified for each PIO as follows:

- PIO mode 0, MIN frequency = 15 MHz
- PIO mode 1, MIN frequency = 22 MHz
- PIO mode 2, MIN frequency = 31 MHz
- PIO mode 3, MIN frequency = 45 MHz
- PIO mode 4, MIN frequency = 57 MHz




A. IORDY is not negated for transfer (no wait generated)
B. IORDY is negative but is re-asserted before $t_A$ (no wait is generated)
C. IORDY is negative before $t_A$ and remains asserted until $t_B$; data is driven valid at $t_{RD}$ (wait is generated)

**Figure 6-26. ATA/CF PIO Data Transfer Timing**

### 6.11.2.2 ATA/CF Multiword DMA Timing

**Table 6-40. Timings for ATA/CF Module — Multiword DMA AC Timing(1) (2) (see Figure 6-27)**

| NO. | | | MODE | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 1 | $t_0$ | Cycle time | 0-2 | (DMASTB + DMARCVR + 2)P - 0.5 | | ns |
| 2 | $t_D$ | $\overline{DIOW}$/$\overline{DIOR}$ active low pulse duration | 0-2 | (DMASTB + 1)P - 1 | | ns |
| 3 | $t_E$ | $\overline{DIOR}$ data access, $\overline{DIOR}$ falling edge to DD[15:0] valid | 0 | | 150 | ns |
| | | | 1 | | 60 | ns |
| | | | 2 | | 50 | ns |
| 4 | $t_F$ | $\overline{DIOR}$ data hold time, DD[15:0] valid after $\overline{DIOR}$ rising edge | 0-2 | 5 | | ns |
| 5 | $t_G$ | $\overline{DIOW}$/$\overline{DIOR}$ data setup time, DD[15:0] (**OUT**) valid before $\overline{DIOW}$/$\overline{DIOR}$ rising edge | 0-2 | (DMASTB)P | | ns |
| | | $\overline{DIOW}$/$\overline{DIOR}$ data setup time, DD[15:0] (**IN**) valid before $\overline{DIOW}$/$\overline{DIOR}$ rising edge | 0 | 100 | | ns |
| | | | 1 | 30 | | ns |
| | | | 2 | 20 | | ns |
| 6 | $t_H$ | $\overline{DIOW}$ data hold time, DD[15:0] valid after $\overline{DIOW}$ rising edge | 0-2 | (HWNHLD + 1)P + 1 | | ns |
| 7 | $t_I$ | $\overline{DMACK}$ to $\overline{DIOW}$/$\overline{DIOR}$ setup | 0-2 | (DMARCVR + 1)P - 1.7 | | ns |
| 8 | $t_J$ | $\overline{DIOW}$/$\overline{DIOR}$ to $\overline{DMACK}$ hold | 0-2 | 5P - 5.9 | | ns |
| 9 | $t_{KR}$ | $\overline{DIOR}$ negated pulse width | 0-2 | (DMARCVR + 1)P - 1 | | ns |
| 10 | $t_{KW}$ | $\overline{DIOW}$ negated pulse width | 0-2 | (DMARCVR + 1)P - 1 | | ns |
| 11 | $t_{LR}$ | $\overline{DIOR}$ to DMARQ delay | 0 | | 120 | ns |
| | | | 1 | | 45 | ns |
| | | | 2 | | 35 | ns |
| 12 | $t_{LW}$ | $\overline{DIOW}$ to DMARQ delay | 0-1 | | 40 | ns |
| | | | 2 | | 35 | ns |
| 13 | $t_M$ | $\overline{ATA\_CSx}$ valid to $\overline{DIOW}$/$\overline{DIOR}$ setup | 0-2 | (DATRCVR)P - 1.7 | | ns |
| 14 | $t_N$ | $\overline{ATA\_CSx}$ valid after $\overline{DIOW}$/$\overline{DIOR}$ rising edge hold | 0-2 | 5P - 1.7 | | ns |
| 15 | $t_Z$ | $\overline{DMACK}$ to read data (DD[15:0]) released | 0 | | 20 | ns |
| | | | 1-2 | | 25 | ns |

(1) P = SYSCLK5 period, in ns, for ATA. For example, when running the DSP CPU at 594 MHz, use P = 10.1 ns. For example, when running the DSP CPU at 810 MHz, use P = 7.4 ns.

(2) DMASTB equals the value programmed in the DMASTBxP bit field in the DMASTB register. DMARCVR equals the value programmed in the DMARCVRxP bit field in the DMARCVR register. HWNHLD equals the value programmed in the HWNHLDxP bit field in the MISCCTL register. For more detailed information, see the *TMS320DM644x DMSoC ATA Controller User's Guide* (literature number SPRUE21).

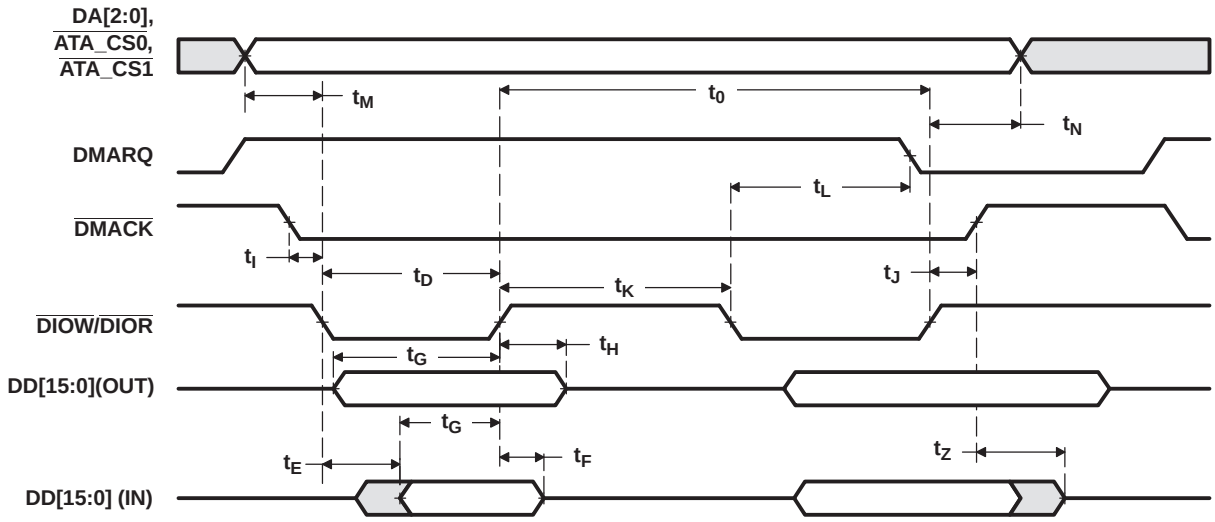



**Figure 6-27. ATA/CF Multiword DMA Timing**

### 6.11.2.3 ATA/CF Ultra DMA Timing

**Table 6-41. Timings for ATA/CF Module — Ultra DMA AC Timing(1) (2) (see Figure 6-28 through Figure 6-37)**

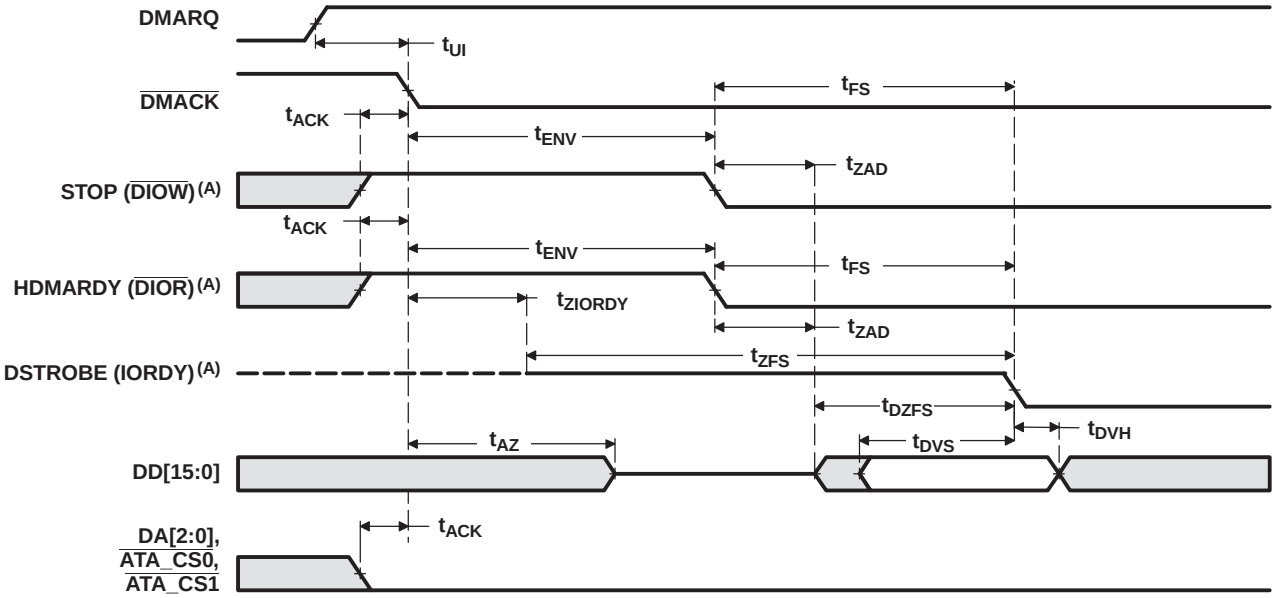
| NO. | | | MODE | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 28 | $f_{(SYSCLK5)}$ | Operating frequency, SYSCLK5 | 0-4 | 25 | | MHz |
| 1 | $t_{2CYCTYP}$ | Typical sustained average two cycle time | 0 | 240 | | ns |
| | | | 1 | 160 | | ns |
| | | | 2 | 120 | | ns |
| | | | 3 | 90 | | ns |
| | | | 4 | 60 | | ns |
| 2 | $t_{CYC}$ | Cycle time, Strobe edge to Strobe edge | 0-4 | (UDMASTB + 1)P | | ns |
| 3 | $t_{2CYC}$ | Two cycle time, rising to rising edge or falling to falling edge | 0-4 | 2(UDMASTB + 1)P | | ns |
| 4 | $t_{DS}$ | Data setup, data valid before STROBE edge | 0 | 15 | | ns |
| | | | 1 | 10 | | ns |
| | | | 2-3 | 7 | | ns |
| | | | 4 | 5 | | ns |
| 5 | $t_{DH}$ | Data hold, data valid after STROBE edge | 0-4 | 5 | | ns |
| 6 | $t_{DVS}$ | Data valid **INPUT** setup time, data valid before STROBE | 0 | 70 | | ns |
| | | | 1 | 48 | | ns |
| | | | 2 | 31 | | ns |
| | | | 3 | 20 | | ns |
| | | | 4 | 6.7 | | ns |
| | | Data valid **OUTPUT** setup time, data valid before STROBE | 0-4 | (UDMASTB)P - 3.1 | | ns |
| 7 | $t_{DVH}$ | Data valid **INPUT** hold time, data valid after STROBE | 0-4 | 6.2 | | ns |
| | | Data valid **OUTPUT** hold time, data valid after STROBE | 0-4 | 1P - 2 | | ns |
| 10 | $t_{CVS}$ | CRC word valid setup time at host, CRC valid before $\overline{\text{DMACK}}$ negation | 0-4 | (UDMASTB)P | | ns |
| 11 | $t_{CVH}$ | CRC word valid hold time at sender, CRC valid after $\overline{\text{DMACK}}$ negation | 0-4 | 2P | | ns |
| 12 | $t_{ZFS}$ | Time from STROBE output released-to-driving until the first transition of critical timing | 0-4 | 0 | | ns |
| 13 | $t_{DZFS}$ | Time from data output released-to-driving until the first transition of critical timing | 0 | 70 | | ns |
| | | | 1 | 48 | | ns |
| | | | 2 | 31 | | ns |
| | | | 3 | 20 | | ns |
| | | | 4 | 6.7 | | ns |

(1) P = SYSCLK5 period, in ns, for ATA. For example, when running the DSP CPU at 594 MHz, use P = 10.1 ns. For example, when running the DSP CPU at 810 MHz, use P = 7.4 ns.

(2) UDMASTB equals the value programmed in the UDMSTBxP bit field in the UDMASTB register. UDMATRP equals the value programmed in the UDMTRPxP bit field in the UDMATRP register. TENV equals the value programmed in the UDMATNVxP bit field in the UDMATENV register. For more detailed information, see the *TMS320DM644x DMSoC ATA Controller User's Guide* (literature number SPRUE21).

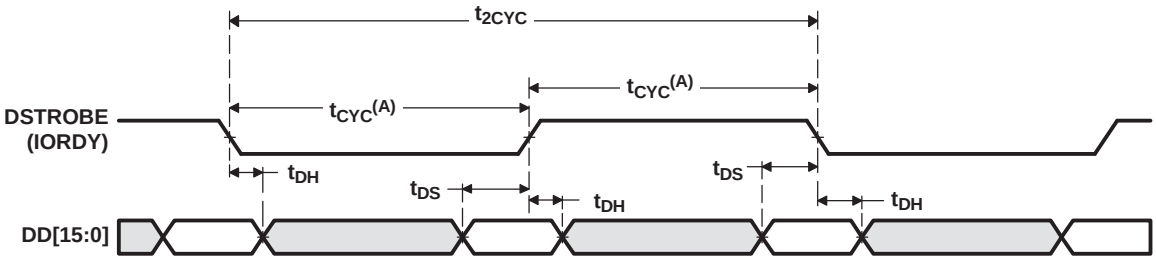
**Table 6-41. Timings for ATA/CF Module — Ultra DMA AC Timing (see Figure 6-28 through Figure 6-37) (continued)**

| NO. | | | MODE | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 14 | $t_{FS}$ | First STROBE time | 0 | | 230 | ns |
| | | | 1 | | 200 | ns |
| | | | 2 | | 170 | ns |
| | | | 3 | | 130 | ns |
| | | | 4 | | 120 | ns |
| 15 | $t_{LI}$ | Limited interlock time | 0-4 | 0 | 150 | ns |
| 16 | $t_{MLI}$ | Interlock time with minimum | 0-4 | 20 | | ns |
| 17 | $t_{UI}$ | Unlimited interlock time | 0-4 | 0 | | ns |
| 18 | $t_{AZ}$ | Maximum time allowed for output drivers to release | 0-4 | | 10 | ns |
| 19 | $t_{ZAH}$ | Minimum delay time required for output | 0-4 | 20 | | ns |
| 20 | $t_{ZAD}$ | Minimum delay time for driver to assert or negate (from released) | 0-4 | 0 | | ns |
| 21 | $t_{ENV}$ | Envelope time, $\overline{\text{DMACK}}$ to STOP and $\overline{\text{DMACK}}$ to HDMARDY during in-burst initiation and from $\overline{\text{DMACK}}$ to STOP during data out burst initiation | 0-4 | (TENV + 1)P - 0.5 | (TENV + 1)P + 1.4 | ns |
| 22 | $t_{RFS}$ | Ready-to-final-STROBE time | 0 | | 75 | ns |
| | | | 1 | | 70 | ns |
| | | | 2-4 | | 60 | ns |
| 23 | $t_{RP}$ | Ready to pause time, **(HDMARDY ($\overline{\text{DIOR}}$) to STOP ($\overline{\text{DIOW}}$))** | 0-4 | (UDMATRP + 1)P - 0.8 | | ns |
| | | Ready to pause time, **(DDMARDY (IORDY) to DMARQ)** | 0 | 160 | | ns |
| | | | 1 | 125 | | ns |
| | | | 2-4 | 100 | | ns |
| 24 | $t_{IORDYZ}$ | Maximum time before releasing IORDY | 0-4 | | 20 | ns |
| 25 | $t_{ZIORDY}$ | Minimum time before driving IORDY | 0-4 | 0 | | ns |
| 26 | $t_{ACK}$ | Setup and hold time for $\overline{\text{DMACK}}$ (before assertion or negation) | 0-4 | 20 | | ns |
| 27 | $t_{SS}$ | STROBE edge to negation of DMARQ or assertion of STOP (when sender terminates a burst) | 0-4 | 50 | | ns |




A. The definitions for the DIOW:STOP, DIOR:HDMARDY, and IORDY:DSTROBE signal lines are not in effect until DMARQ and DMACK are asserted.

**Figure 6-28. ATA/CF Initiating an Ultra DMA Data-In Burst Timing**




A. While DSTROBE (IORDY) timing is $t_{CYC}$ at the device, it may be different at the host due to propagation delay differences on the cable.

**Figure 6-29. ATA/CF Sustained Ultra DMA Data-In Data Transfer Timing**

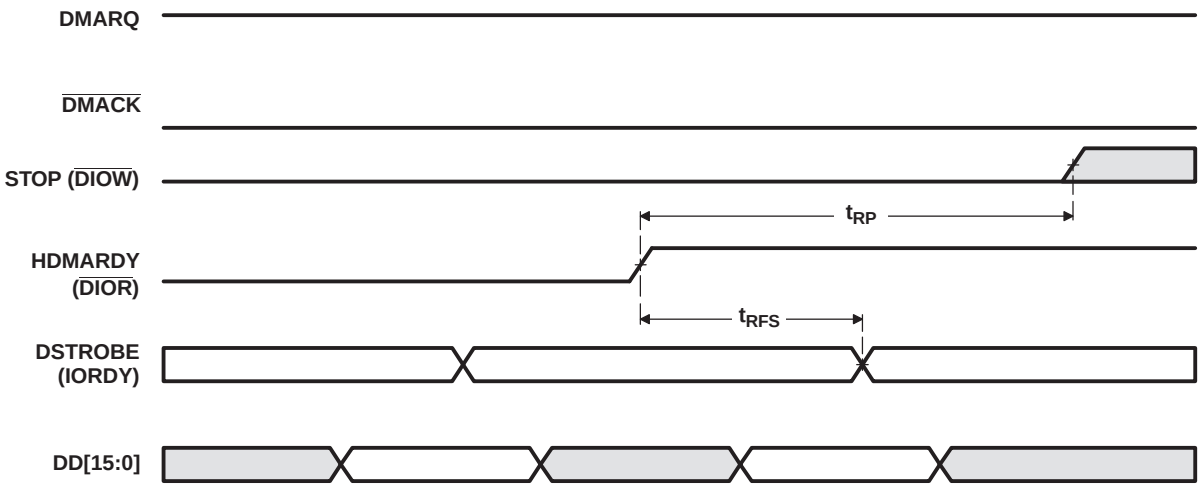



**Figure 6-30. ATA/CF Host Pausing an Ultra DMA Data-In Burst Timing**

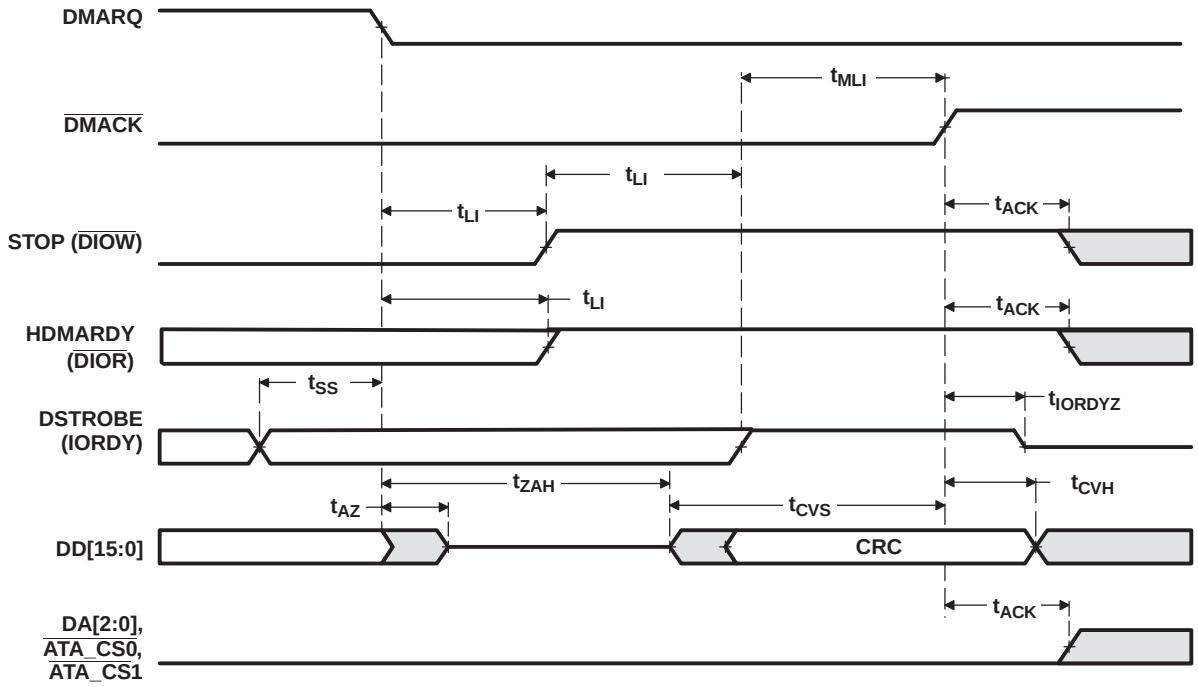



**Figure 6-31. ATA/CF Device Terminating an Ultra DMA Data-In Burst Timing**

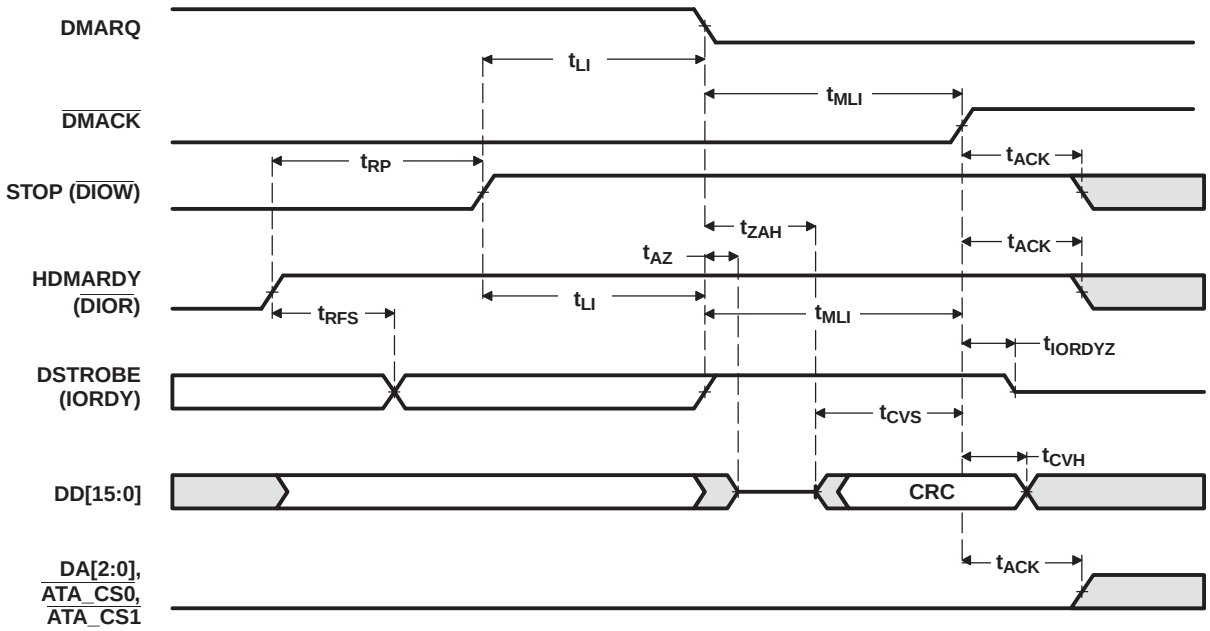
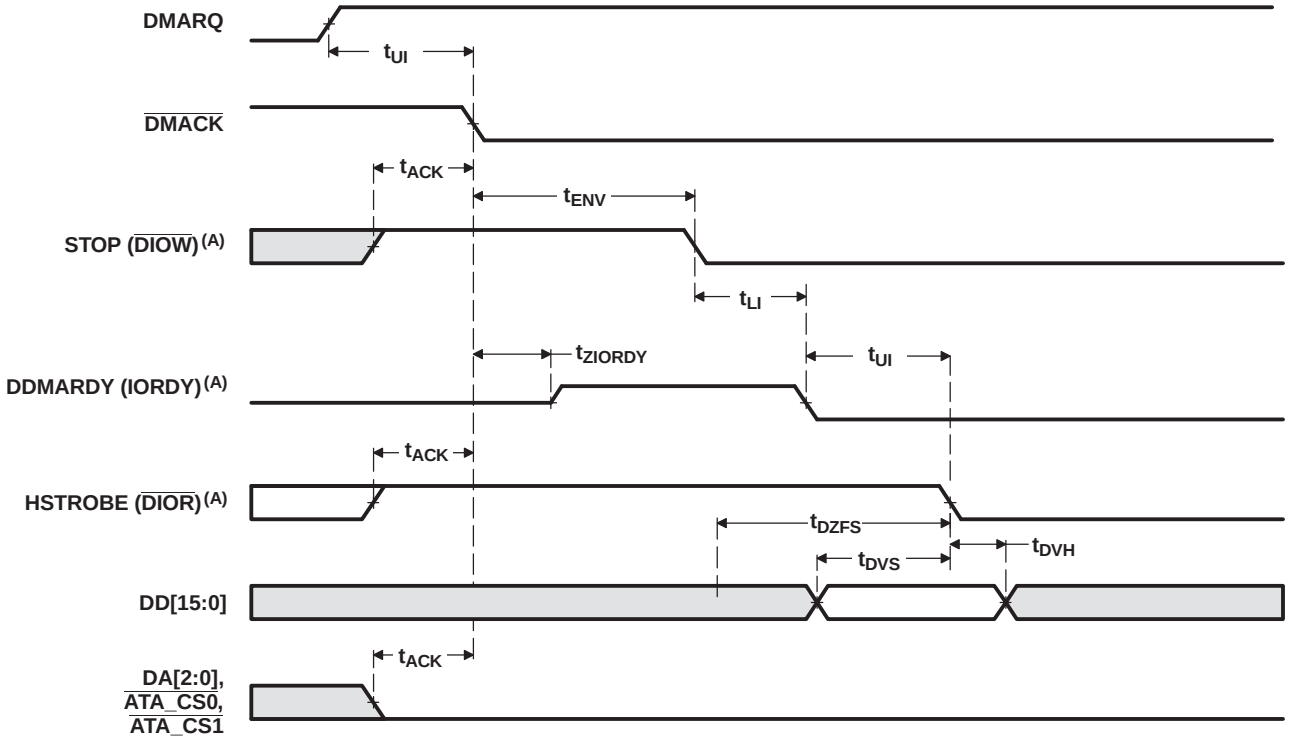
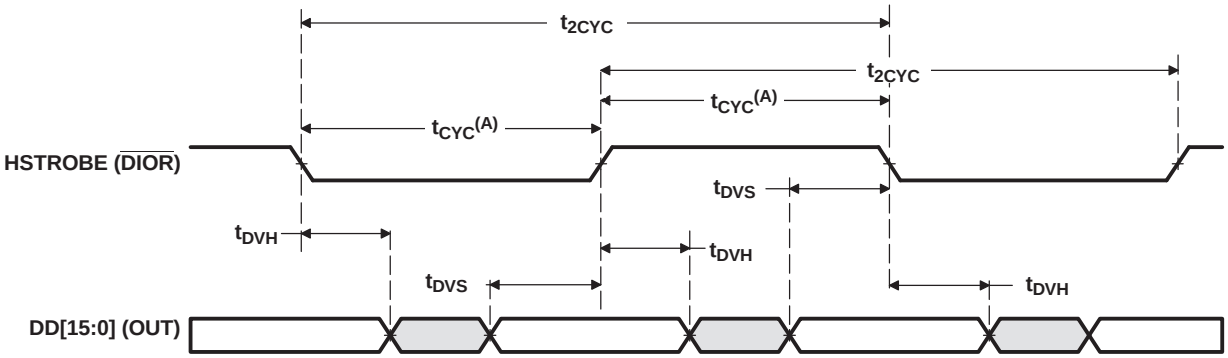



**Figure 6-32. ATA/CF Host Terminating an Ultra DMA Data-In Burst Timing**




A. The definitions for the DIOW:STOP, IORDY:DDMARDY, and DIOR:HSTROBE signal lines are not in effect until DMARQ and DMACK are asserted.

**Figure 6-33. ATA/CF Initiating an Ultra DMA Data-Out Burst Timing**




A. While HSTROBE (DIOR) timing is $t_{CYC}$ at the host, it may be different at the device due to propagation delay differences on the cable.

**Figure 6-34. ATA/CF Sustained Ultra DMA Data-Out Transfer Timing**

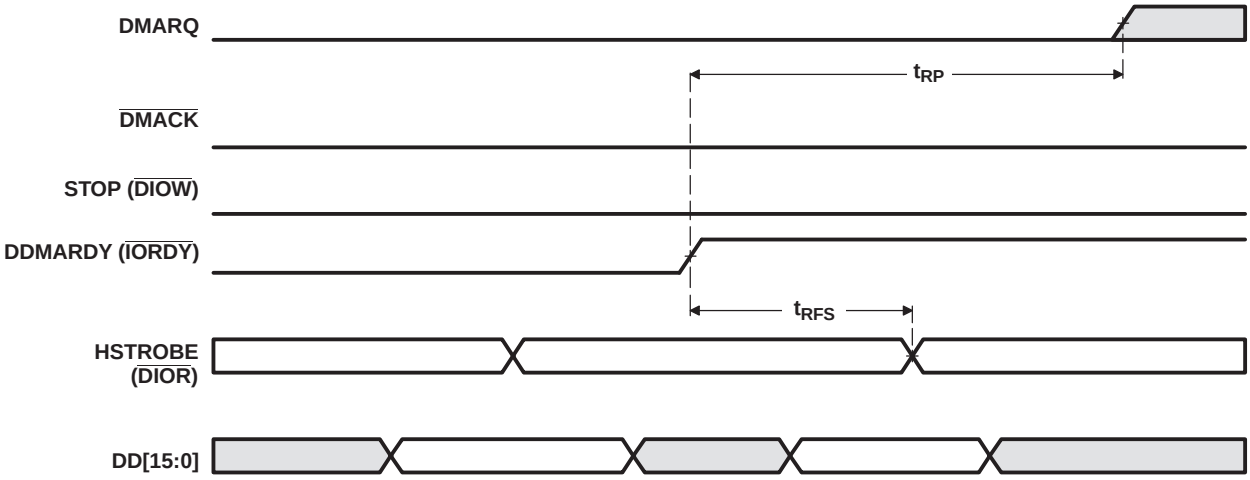



**Figure 6-35. ATA/CF Device Pausing an Ultra DMA Data-Out Burst Timing**

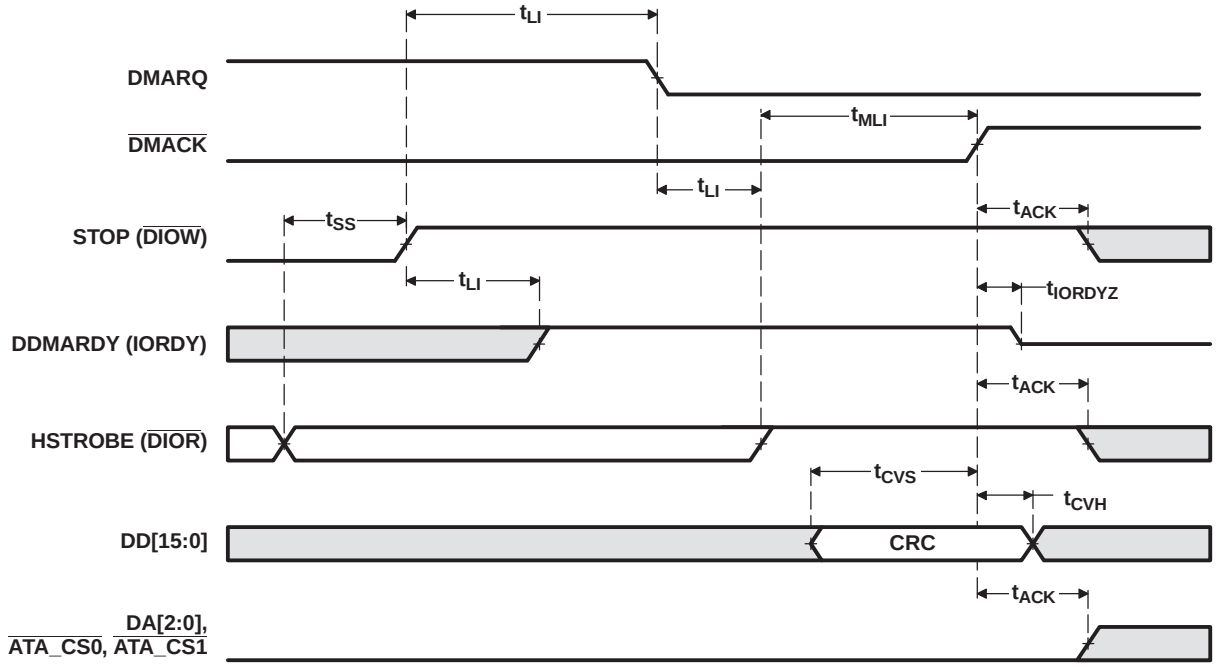



**Figure 6-36. ATA/CF Host Terminating an Ultra DMA Data-Out Burst Timing**

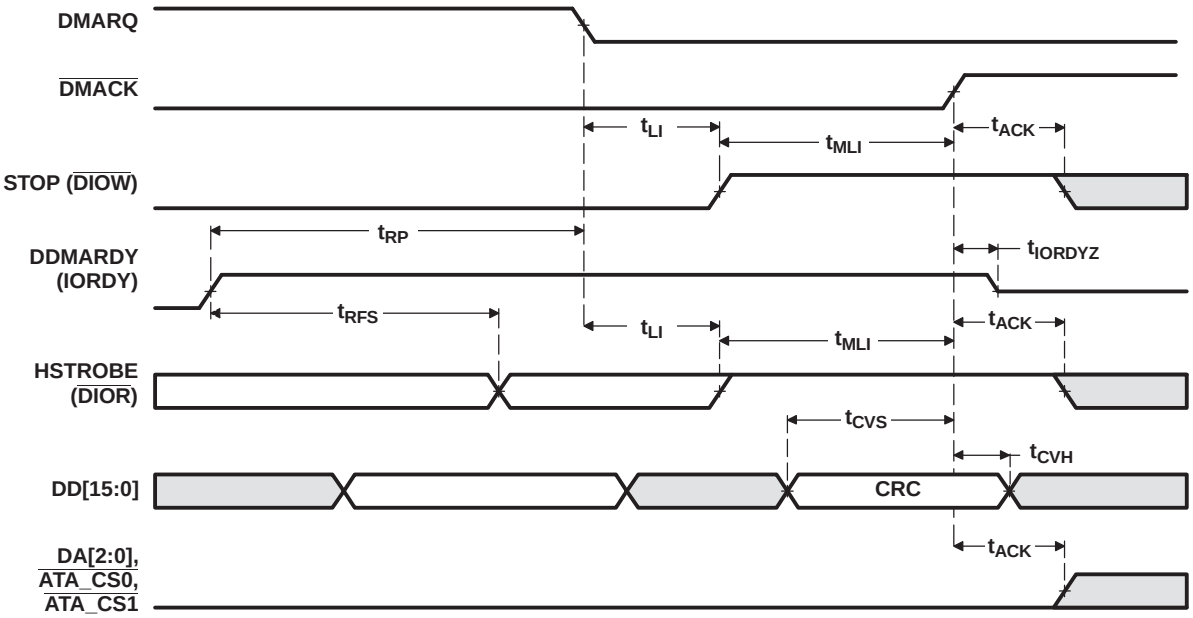



**Figure 6-37. ATA/CF Device Terminating an Ultra DMA Data-Out Burst Timing**

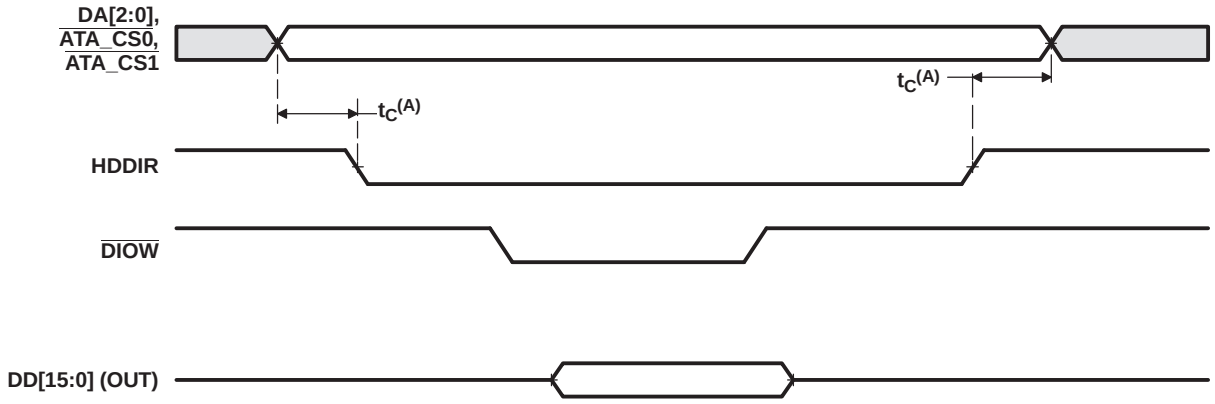
### 6.11.2.4 ATA/CF HDDIR Timing

Figure 6-38 through Figure 6-41 show the behavior of HDDIR for the different types of transfers.

**Table 6-42. Timing Requirements for HDDIR(1)**

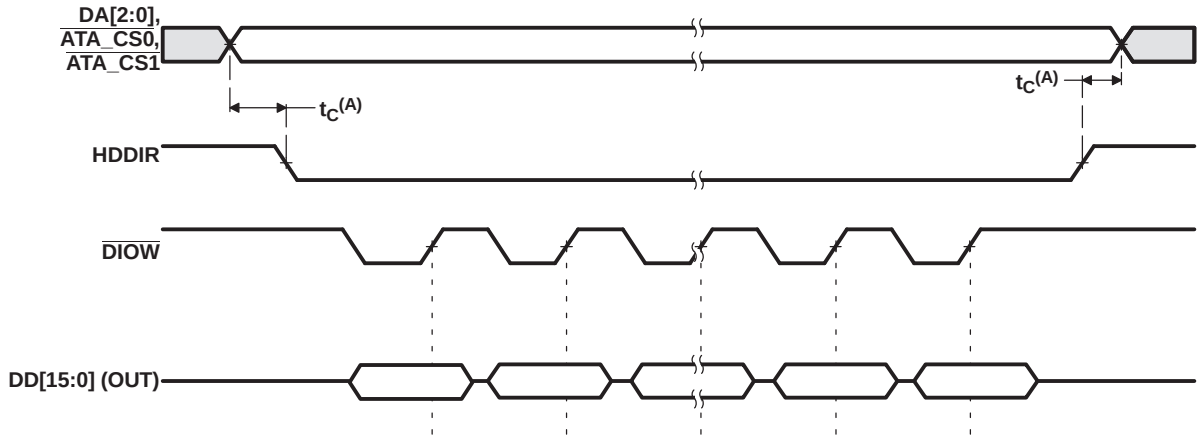
| NO. | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| 1 | $t_c$ Cycle time, $\overline{ATA\_CS[1:0]}$ to HDDIR low | E - 3.1 | 2.1 | ns |

(1) E = ATA clock cycle



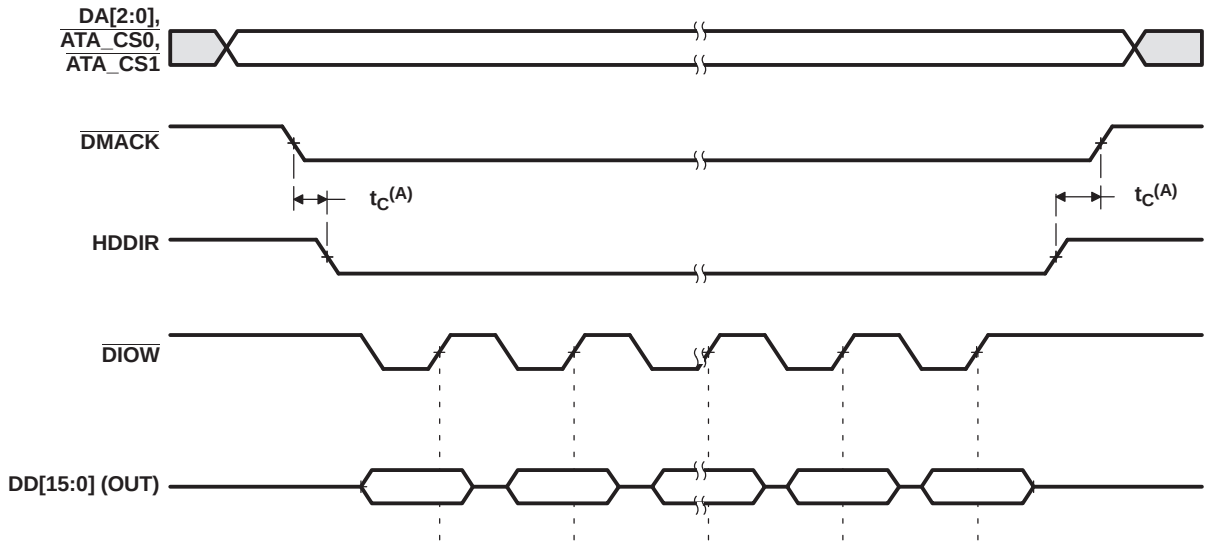

A. $t_C \geq$ one cycle

**Figure 6-38. ATA/CF HDDIR Taskfile Write/Single PIO Write Timing**



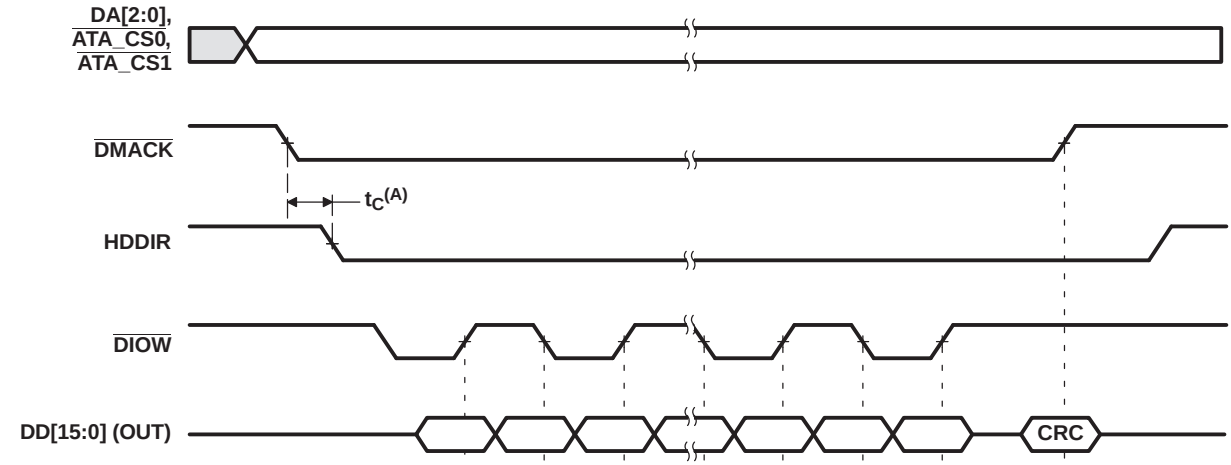

A. $t_C \geq$ one cycle

**Figure 6-39. ATA/CF HDDIR PIO Postwrite Start Timing**




A. $t_C \geq$ one cycle

**Figure 6-40. ATA/CF HDDIR Multiword DMA Write Transfer Timing**




A. $t_C \geq$ one cycle

**Figure 6-41. ATA/CF HDDIR Ultra DMA Write Transfer Timing**

## 6.12 MMC/SD/SDIO

The DM6446 MMC/SD/SDIO Controller has following features:

- MultiMediaCard (MMC).
- Secure Digital (SD) memory card with Secure Data I/O (SDIO).
- MMC/SD/SDIO protocol support.
- Programmable clock frequency.
- 256 bit Read/Write FIFO to lower system overhead.
- Slave DMA transfer capability.

SDIO is only supported for WLAN operation through TI third parties. For more information about third-party WLAN products, go to http://www.ti.com.davinciwlan.

The MMC/SD/SDIO register memory mapping is shown in Table 6-43.

### *6.12.1 MMC/SD/SDIO Peripheral Description(s)*

**Table 6-43. MMC/SD/SDIO Register Descriptions**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01E1 0000 | MMCCTL | MMC Control Register |
| 0x01E1 0004 | MMCCLK | MMC Memory Clock Control Register |
| 0x01E1 0008 | MMCST0 | MMC Status Register 0 |
| 0x01E1 000C | MMCST1 | MMC Status Register 1 |
| 0x01E1 0010 | MMCIM | MMC Interrupt Mask Register |
| 0x01E1 0014 | MMCTOR | MMC Response Time-Out Register |
| 0x01E1 0018 | MMCTOD | MMC Data Read Time-Out Register |
| 0x01E1 001C | MMCBLEN | MMC Block Length Register |
| 0x01E1 0020 | MMCNBLK | MMC Number of Blocks Register |
| 0x01E1 0024 | MMCNBLC | MMC Number of Blocks Counter Register |
| 0x01E1 0028 | MMCDRR | MMC Data Receive Register |
| 0x01E1 002C | MMCDXR | MMC Data Transmit Register |
| 0x01E1 0030 | MMCCMD | MMC Command Register |
| 0x01E1 0034 | MMCARGHL | MMC Argument Register |
| 0x01E1 0038 | MMCRSP01 | MMC Response Register 0 and 1 |
| 0x01E1 003C | MMCRSP23 | MMC Response Register 2 and 3 |
| 0x01E1 0040 | MMCRSP45 | MMC Response Register 4 and 5 |
| 0x01E1 0044 | MMCRSP67 | MMC Response Register 6 and 7 |
| 0x01E1 0048 | MMCDRSP | MMC Data Response Register |
| 0x01E1 004C - 0x01E1 004F | - | Reserved |
| 0x01E1 0050 | MMCCIDX | MMC Command Index Register |
| 0x01E1 0054 - 0x01E1 0063 | - | Reserved |
| 0x01E1 0064 | SDIOCTL | SDIO Control Register |
| 0x01E1 0068 | SDIOST0 | SDIO Status Register 0 |
| 0x01E1 006C | SDIOIEN | SDIO Interrupt Enable Register |
| 0x01E1 0070 | SDIOIST | SDIO Interrupt Status Register |
| 0x01E1 0074 | MMCFIFOCTL | MMC FIFO Control Register |
| 0x01E1 0078 - 0x01E1 FFFF | - | Reserved |

### 6.12.2 MMC/SD/SDIO Electrical Data/Timing

**Table 6-44. Timing Requirements for MMC/SD/SDIO Module (see Figure 6-43 and Figure 6-45)**

| NO. | | | A-513, -594, -810 STANDARD MODE | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(CMDV-CLKH)}$ | Setup time, SD_CMD valid before SD_CLK high | 5 | | ns |
| 2 | $t_{h(CLKH-CMDV)}$ | Hold time, SD_CMD valid after SD_CLK high | 5 | | ns |
| 3 | $t_{su(DATV-CLKH)}$ | Setup time, SD_DATx valid before SD_CLK high | 5 | | ns |
| 4 | $t_{h(CLKH-DATV)}$ | Hold time, SD_DATx valid after SD_CLK high | 5 | | ns |

**Table 6-45. Switching Characteristics Over Recommended Operating Conditions for MMC/SD/SDIO Module(1) (see Figure 6-42 through Figure 6-45)**

| NO. | PARAMETER | | A-513, -594, -810 STANDARD MODE | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 7 | $f_{(CLK)}$ | Operating frequency, SD_CLK | 0 | 25 | MHz |
| 8 | $f_{(CLK\_ID)}$ | Identification mode frequency, SD_CLK | 0 | 400 | KHz |
| 9 | $t_{W(CLKL)}$ | Pulse width, SD_CLK low | 10 | | ns |
| 10 | $t_{W(CLKH)}$ | Pulse width, SD_CLK high | 10 | | ns |
| 11 | $t_{r(CLK)}$ | Rise time, SD_CLK | | 10 | ns |
| 12 | $t_{f(CLK)}$ | Fall time, SD_CLK | | 10 | ns |
| 13 | $t_{d(CLKLL-CMD)}$ | Delay time, SD_CLK low to SD_CMD transition | -7.5 | 13 | ns |
| 14 | $t_{dis(CLKL-DAT)}$ | Disable time, SD_CLK low to SD_DATx transition | -7.5 | 13 | ns |

(1) P = Period of SD_CLK (SYSCLK5), in nanoseconds (ns).

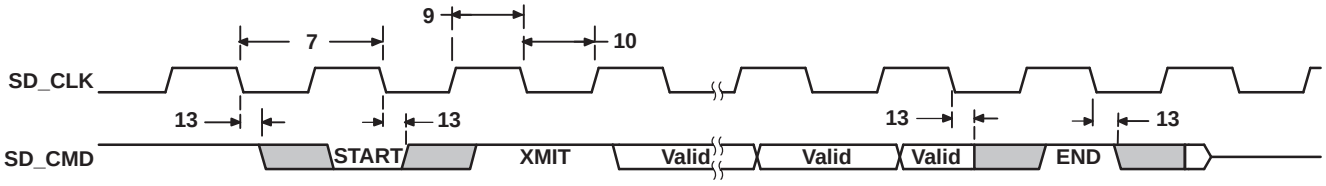



**Figure 6-42. MMC/SD/SDIO Host Command Timing**

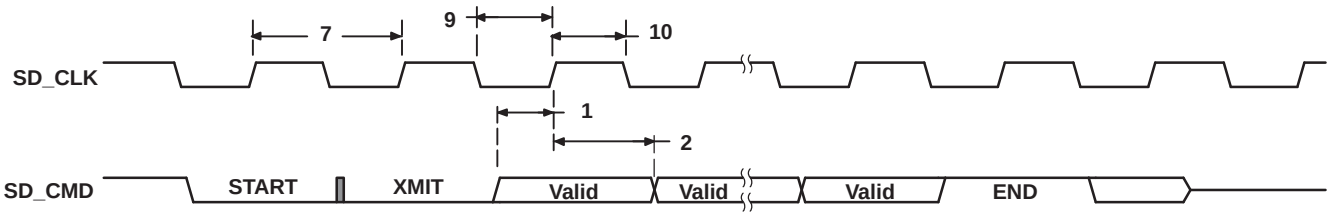



**Figure 6-43. MMC/SD/SDIO Card Response Timing**

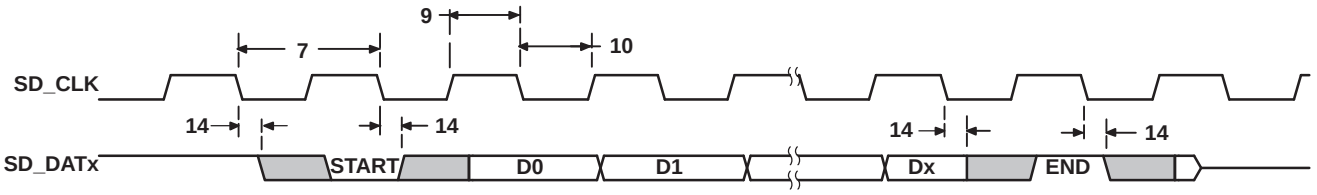



**Figure 6-44. MMC/SD/SDIO Host Write Timing**

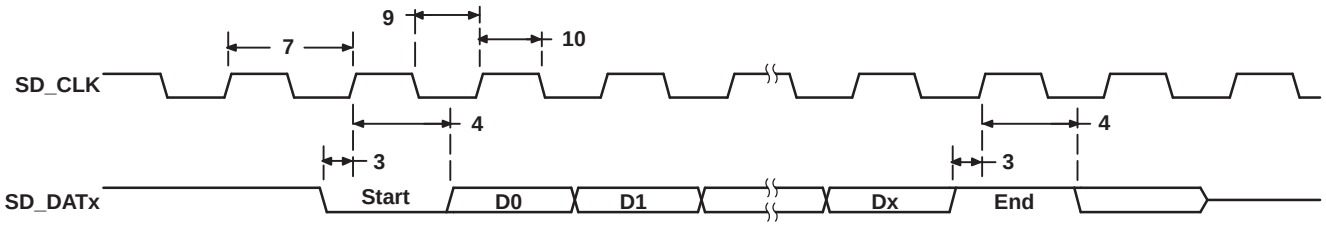



**Figure 6-45. MMC/SD/SDIO Host Read and Card CRC Status Timing**

## 6.13 Video Processing Sub-System (VPSS) Overview

The DM6446 Video Processing Sub-System (VPSS) provides a Video Processing Front End (VPFE) input interface for external imaging peripherals (i.e., image sensors, video decoders, etc.) and a Video Processing Back End (VPBE) output interface for display devices, such as analog SDTV displays, digital LCD panels, HDTV video encoders, etc.

**Note:** The VPSS module is supported with Linux Application Peripheral Interfaces (APIs) commonly used by video application developers. Video for Linux 2 or V4L2 uses APIs commonly used for video capture. The typical use cases of the VPSS Video Front-End (VPFE) have been ported to this Linux API structure. V4L2 supports standard video interfaces such as: BT.656 and Y/C mode. Other modules within the VPSS VPFE for example, the Preview Engine, H3A, and Histogram are *not* currently supported within the software APIs. The VPSS Back-End (VPBE) uses FBDev/DirectFB as the APIs. Certain functionalities within the VPBE have not been implemented in the FBDev/DirectFB APIs. For modes/functions not implemented in software, it is the user's responsibility to modify the software drivers/APIs.

The VPSS register memory mapping is shown in Table 6-46.

**Table 6-46. VPSS Register Descriptions**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C7 3400 | PID | Peripheral Revision and Class Information |
| 0x01C7 3404 | PCR | VPSS Control Register |
| 0x01C7 3408 | - | Reserved |
| 0x01C7 3508 | SDR_REG_EXP | SDRAM Non Real-Time Read Request Expand |
| 0x01C7 350C - 0x01C7 3FFF | - | Reserved |

To ensure NTSC- and PAL-compliant output video, the stability of the input clock source is very important. TI recommends a 27-MHz, 50-ppm crystal. Ceramic oscillators are not recommended. The NTSC/PAL color sub-carrier frequency is derived from the 27-MHz clock. Therefore, if the 27-MHz clock drifts, then the color sub-carrier frequency will drift as well. Assuming no 27-MHz frequency drift, the color sub-carrier frequency is generated as follows:

$$f_{sc-ntsc} = 27\ \text{MHz}\left(\frac{35}{264}\right) = 3.5795454545\ \text{MHz}$$

$$f_{sc-pal} = 27\ \text{MHz}\left(\frac{167}{1017}\right) = 4.4332628318\ \text{MHz}$$

To ensure the color sub-carrier frequency will not drift out of spec, the user must follow the crystal requirements discussed in Section 6.5.1, *Clock Input Option 1 – Crystal*. Alternatively, if the VPBE input clock is sourced from the VPBECLK or VPFE clock inputs, these clocks must have a frequency stability of ±50 ppm to ensure the NTSC and PAL compliant output video.

### 6.13.1 Video Processing Front-End (VPFE)

The Video Processing Front-End (VPFE) consists of the CCD Controller (CCDC), Preview Engine, Resizer, Hardware 3A (H3A) Statistic Generator, and Histogram blocks. Together, these modules provide DM6446 with a powerful and flexible front-end interface. These modules are briefly described below:

- The CCDC provides an interface to image sensors and digital video sources.
- The Preview Engine is a parameterized hardwired image processing block which is used for converting RAW color data from a Bayer pattern to YUV 4:2:2.
- The Resizer module re-sizes the input image data to the desired display or video encoding resolution
- The H3A module provides control loops for Auto Focus (AF), Auto White Balance (AWB) and Auto Exposure (AE).
- The Histogram module bins input color pixels, depending on the amplitude, and provides statistics required to implement various 3A (AE/AF/AWB) algorithms and tune the final image/video output.

The VPFE register memory mapping is shown in Table 6-47.

**Table 6-47. VPFE Register Address Range Descriptions**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C7 0400 – 0x01C7 07FF | CCDC | VPFE – CCD Controller |
| 0x01C7 0800 – 0x01C7 0BFF | PREV | VPFE – Preview Engine/Image Signal Processor |
| 0x01C7 0C00 – 0x01C7 09FF | RESZ | VPFE – Resizer |
| 0x01C7 1000 – 0x01C7 13FF | HIST | VPFE – Histogram |
| 0x01C7 1400 – 0x01C7 17FF | H3A | VPFE – Hardware 3A (Auto-Focus/WB/Exposure) |
| 0x01C7 3400 – 0x01C7 3FFF | VPSS | VPSS Shared Buffer Logic Registers |

#### 6.13.1.1 CCD Controller (CCDC)

The CCDC receives raw image/video data from sensors (CMOS or CCD) or YUV video data in numerous formats from video decoder devices. The following features are supported by the CCDC module.

- Conventional Bayer pattern formats.
- Generates HD/VD timing signals and field ID to an external timing generator or can synchronize to an external timing generator.
- Interface to progressive and interlaced sensors.
- REC656/CCIR-656 standard (YCbCr 4:2:2 format, either 8- or 16-bit).
- YCbCr 4:2:2 format, either 8- or 16-bit with discrete H and VSYNC signals.
- Up to 16-bit input.
- Optical black clamping signal generation.
- Shutter signal control.
- Digital clamping and black level compensation.
- 10-bit to 8-bit A-law compression.
- Low-pass filter prior to writing to SDRAM. If this filter is enabled, 2 pixels each in the left and right edges of each line are cropped from the output.
- Output range from 16-bits to 8-bits wide (8-bits wide allows for 50% saving in storage area).
- Downsampling via programmable culling patterns.
- Control output to the DDR2 via an external write enable signal.
- Up to 16K pixels (image size) in both the horizontal and vertical direction.

The CCDC register memory mapping is shown in Table 6-48.

**Table 6-48. CCDC Register Descriptions**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C7 0400 | PID | Peripheral Revision and Class Information |
| 0x01C7 0404 | PCR | Peripheral Control Register |
| 0x01C7 0408 | SYN_MODE | SYNC and Mode Set Register |
| 0x01C7 040C | HD_VD_WID | HD and VD Signal Width |
| 0x01C7 0410 | PIX_LINES | Number of Pixels in a Horizontal Line and Number of Lines in a Frame |
| 0x01C7 0414 | HORZ_INFO | Horizontal Pixel Information |
| 0x01C7 0418 | VERT_START | Vertical Line - Settings for the Starting Pixel |
| 0x01C7 041C | VERT_LINES | Number of Vertical Lines |
| 0x01C7 0420 | CULLING | Culling Information in Horizontal and Vertical Directions |
| 0x01C7 0424 | HSIZE_OFF | Horizontal Size |
| 0x01C7 0428 | SDOFST | SDRAM/DDRAM Line Offset |
| 0x01C7 042C | SDR_ADDR | SDRAM Address |
| 0x01C7 0430 | CLAMP | Optical Black Clamping Settings |
| 0x01C7 0434 | DCSUB | DC Clamp |
| 0x01C7 0438 | COLPTN | CCD Color Pattern |
| 0x01C7 043C | BLKCMP | Black Compensation |
| 0x01C7 0440 | - | Reserved |
| 0x01C7 0444 | - | Reserved |
| 0x01C7 0448 | VDINT | VD Interrupt Timing |
| 0x01C7 044C | ALAW | A-Law Setting |
| 0x01C7 0450 | REC656IF | REC656 Interface |
| 0x01C7 0454 | CCDCFG | CCD Configuration |
| 0x01C7 0458 | FMTCFG | Data Reformatter/Video Port Configuration |
| 0x01C7 045C | FMT_HORZ | Data Reformatter/Video Input Interface Horizontal Information |
| 0x01C7 0460 | FMT_VERT | Data Reformatter/Video Input Interface Vertical Information |
| 0x01C7 0464 | FMT_ADDR0 | Address Pointer 0 Setup |
| 0x01C7 0468 | FMT_ADDR1 | Address Pointer 1 Setup |
| 0x01C7 046C | FMT_ADDR2 | Address Pointer 2 Setup |
| 0x01C7 0470 | FMT_ADDR3 | Address Pointer 3 Setup |
| 0x01C7 0474 | FMT_ADDR4 | Address Pointer 4 Setup |
| 0x01C7 0478 | FMT_ADDR5 | Address Pointer 5 Setup |
| 0x01C7 047C | FMT_ADDR6 | Address Pointer 6 Setup |
| 0x01C7 0480 | FMT_ADDR7 | Address Pointer 7 Setup |
| 0x01C7 0484 | PRGEVEN_0 | Program Entries 0-7 for Even Line |
| 0x01C7 0488 | RRGEVEN_1 | Program Entries 8-15 for Even Line |
| 0x01C7 048C | PRGGODD_0 | Program Entries 0-7 for Odd Line |
| 0x01C7 0490 | PRGGODD_1 | Program Entries 8-15 for Odd Line |
| 0x01C7 0494 | VP_OUT | Video Port Output Settings |

#### 6.13.1.2 Preview Engine

The preview engine transforms raw unprocessed image/video data from a sensor (CMOS or CCD) into YCbCr 4:2:2 data. The output of the preview engine is used for both video compression and external display devices such as a NTSC/PAL analog encoder or a digital LCD. The following features are supported by the preview engine.

- Accepts conventional Bayer pattern formats.
- Input image/video data from either the CCD/CMOS controller or the DDR2 memory.
- Output width up to 1280 pixels wide.
- Automatic/mandatory cropping of pixels/lines when edge processing is performed. If all the corresponding modules are enabled, a total of 14 pixels per line (7 left most and 7 right most) and 8 lines (4 top most and 4 bottom most) will not be output.
- Simple horizontal averaging (by factors of 2, 4, or 8) to handle input widths that are greater than 1280 (plus the cropped number) pixels wide.
- Dark frame capture to DDR2.
- Dark frame subtraction for every input raw data frame, fetched from DDR2, pixel-by-pixel to improve video quality.
- Lens shading compensation. Each input pixel is multiplied with a corresponding 8-bit gain value and the result is right shifted by a programmable parameter (0-7 bits).
- A-law decompression to transform non-linear 8-bit data to 10-bit linear data. This feature allows data in DDR2 to be 8-bits, which saves 50% of the area if the input to the preview engine is from the DDR2.
- Horizontal median filter for reducing temperature induced noise in pixels.
- Programmable noise filter that operates on a 3x3 grid of the same color (effectively, this is a five line storage requirement).
- Digital gain and white balance (color separate gain for white balance).
- Programmable CFA interpolation that operates on a 5x5 grid.
- Conventional Bayer pattern RGB and complementary color sensors.
- Support for an image that is downsampled by 2x in the horizontal direction (with and without phase correction). In this case, the image is 2/3 populated instead of the conventional 1/3 colors.
- Support for an image that is downsampled by 2x in both the horizontal and vertical direction. In this case, the image is fully populated instead of the conventional 1/3 colors.
- Programmable RGB-to-RGB blending matrix (9 coefficients for the 3x3 matrix).
- Fully programmable gamma correction (1024 entries for each color held in an on-chip RAM).
- Programmable color conversion (RGB to YUV) coefficients (9 coefficients for the 3x3 matrix).
- Luminance enhancement (non-linear) and chrominance suppression & offset.

The Preview Engine register memory mapping is shown in Table 6-49.

**Table 6-49. Preview Engine Register Descriptions**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C7 0800 | PID | Peripheral Revision and Class Information |
| 0x01C7 0804 | PCR | Peripheral Control Register |
| 0x01C7 0808 | HORZ_INFO | Horizontal Information/Setup |
| 0x01C7 080C | VERT_INFO | Vertical Information/Setup |
| 0x01C7 0810 | RSDR_ADDR | Read Address From SDRAM |
| 0x01C7 0814 | RADR_OFFSET | Line Offset for the Read Data |
| 0x01C7 0818 | DSDR_ADDR | Dark Frame Address From SDRAM |
| 0x01C7 081C | DRKF_OFFSET | Line Offset for the Dark Frame Data |
| 0x01C7 0820 | WSDR_ADDR | Write Address to the SDRAM |
| 0x01C7 0824 | WADD_OFFSET | Line Offset for the Write Data |
| 0x01C7 0828 | AVE | Input Formatter/Averager |
| 0x01C7 082C | HMED | Horizontal Median Filter |
| 0x01C7 0830 | NF | Noise Filter |
| 0x01C7 0834 | WB_DGAIN | White Balance Digital Gain |
| 0x01C7 0838 | WBGAIN | White Balance Coefficients |
| 0x01C7 083C | WBSEL | White Balance Coefficients Selection |
| 0x01C7 0840 | CFA | CFA Register |
| 0x01C7 0844 | BLKADJOFF | Black Adjustment Offset |
| 0x01C7 0848 | RGB_MAT1 | RGB2RGB Blending Matrix Coefficients |
| 0x01C7 084C | RGB_MAT2 | RGB2RGB Blending Matrix Coefficients |
| 0x01C7 0850 | RGB_MAT3 | RGB2RGB Blending Matrix Coefficients |
| 0x01C7 0854 | RGB_MAT4 | RGB2RGB Blending Matrix Coefficients |
| 0x01C7 0858 | RGB_MAT5 | RGB2RGB Blending Matrix Coefficients |
| 0x01C7 085C | RGB_OFF1 | RGB2RGB Blending Matrix Offsets |
| 0x01C7 0860 | RGB_OFF2 | RGB2RGB Blending Matrix Offsets |
| 0x01C7 0864 | CSC0 | Color Space Conversion Coefficients |
| 0x01C7 0868 | CSC1 | Color Space Conversion Coefficients |
| 0x01C7 086C | CSC2 | Color Space Conversion Coefficients |
| 0x01C7 0870 | CSC_OFFSET | Color Space Conversion Offsets |
| 0x01C7 0874 | CNT_BRT | Contrast and Brightness Settings |
| 0x01C7 0878 | CSUP | Chrominance Suppression Settings |
| 0x01C7 087C | SETUP_YC | Maximum/Minimum Y and C Settings |
| 0x01C7 0880 | SET_TBL_ADDRESS | Setup Table Addresses |
| 0x01C7 0884 | SET_TBL_DATA | Setup Table Data |

#### 6.13.1.3 Resizer

The resizer module can accept input image/video data from either the preview engine or DDR2. The output of the resizer module is sent to DDR2. The following features are supported by the resizer module.

- An output width up to 1280 horizontal pixels.
- Input from external DDR2.
- Up to 4x upsampling (digital zoom).
- Bi-cubic interpolation (4-tap horizontal, 4-tap vertical) can be implemented with the programmable filter coefficients.
- 8 phases of filter coefficients.
- Optional bi-linear interpolation for the chrominance components.
- Up to 1/4x downsampling
- 4-tap horizontal and 4-tap vertical filter coefficients (with 8-phases) for 1x to 1/2x downsampling
- 1/2x to 1/4x downsampling, for 7-tap mode with 4-phases.
- Resizing either YUV 4:2:2 packed data (16-bits) or color separate data (8-bit data within DDR) that is contiguous.
- Separate/independent resizing factor for the horizontal and vertical directions.
- Upsampling and downsampling ratios that are available are: 256/N, with N ranging from 64 to 1024.
- Programmable luminance sharpening after the horizontal resizing and before the vertical resizing step.

The Resizer register memory mapping is shown in Table 6-50.

**Table 6-50. Resizer Register Descriptions**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C7 0C00 | PID | Peripheral Revision and Class Information |
| 0x01C7 0C04 | PCR | Peripheral Control Register |
| 0x01C7 0C08 | RSZ_CNT | Resizer Control Bits |
| 0x01C7 0C0C | OUT_SIZE | Output Width and Height After Resizing |
| 0x01C7 0C10 | IN_START | Input Starting Information |
| 0x01C7 0C14 | IN_SIZE | Input Width and Height Before Resizing |
| 0x01C7 0C18 | SDR_INADD | Input SDRAM Address |
| 0x01C7 0C1C | SDR_INOFF | SDRAM Offset for the Input Line |
| 0x01C7 0C20 | SDR_OUTADD | Output SDRAM Address |
| 0x01C7 0C24 | SDR_OUTOFF | SDRAM Offset for the Output Line |
| 0x01C7 0C28 | HFILT10 | Horizontal Filter Coefficients 1 and 0 |
| 0x01C7 0C2C | HFILT32 | Horizontal Filter Coefficients 3 and 2 |
| 0x01C7 0C30 | HFILT54 | Horizontal Filter Coefficients 5 and 4 |
| 0x01C7 0C34 | HFILT76 | Horizontal Filter Coefficients 7 and 6 |
| 0x01C7 0C38 | HFILT98 | Horizontal Filter Coefficients 9 and 8 |
| 0x01C7 0C3C | HFILT1110 | Horizontal Filter Coefficients 11 and 10 |
| 0x01C7 0C40 | HFILT1312 | Horizontal Filter Coefficients 13 and 12 |
| 0x01C7 0C44 | HFILT1514 | Horizontal Filter Coefficients 15 and 14 |
| 0x01C7 0C48 | HFILT1716 | Horizontal Filter Coefficients 17 and 16 |
| 0x01C7 0C4C | HFILT1918 | Horizontal Filter Coefficients 19 and 18 |
| 0x01C7 0C50 | HFILT2120 | Horizontal Filter Coefficients 21 and 20 |
| 0x01C7 0C54 | HFILT2322 | Horizontal Filter Coefficients 23 and 22 |
| 0x01C7 0C58 | HFILT2524 | Horizontal Filter Coefficients 25 and 24 |
| 0x01C7 0C5C | HFILT2726 | Horizontal Filter Coefficients 27 and 26 |
| 0x01C7 0C60 | HFILT2928 | Horizontal Filter Coefficients 29 and 28 |
| 0x01C7 0C64 | HFILT3130 | Horizontal Filter Coefficients 31 and 30 |

**Table 6-50. Resizer Register Descriptions (continued)**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C7 0C68 | VFILT10 | Vertical Filter Coefficients 1 and 0 |
| 0x01C7 0C6C | VFILT32 | Vertical Filter Coefficients 3 and 2 |
| 0x01C7 0C70 | VFILT54 | Vertical Filter Coefficients 5 and 4 |
| 0x01C7 0C74 | VFILT76 | Vertical Filter Coefficients 7 and 6 |
| 0x01C7 0C78 | VFILT98 | Vertical Filter Coefficients 9 and 8 |
| 0x01C7 0C7C | VFILT1110 | Vertical Filter Coefficients 11 and 10 |
| 0x01C7 0C80 | VFILT1312 | Vertical Filter Coefficients 13 and 12 |
| 0x01C7 0C84 | VFILT1514 | Vertical Filter Coefficients 15 and 14 |
| 0x01C7 0C88 | VFILT1716 | Vertical Filter Coefficients 17 and 16 |
| 0x01C7 0C8C | VFILT1918 | Vertical Filter Coefficients 19 and 18 |
| 0x01C7 0C90 | VFILT2120 | Vertical Filter Coefficients 21 and 20 |
| 0x01C7 0C94 | VFILT2322 | Vertical Filter Coefficients 23 and 22 |
| 0x01C7 0C98 | VFILT2524 | Vertical Filter Coefficients 25 and 24 |
| 0x01C7 0C9C | VFILT2726 | Vertical Filter Coefficients 27 and 26 |
| 0x01C7 0CA0 | VFILT2928 | Vertical Filter Coefficients 29 and 28 |
| 0x01C7 0CA4 | VFILT3130 | Vertical Filter Coefficients 31 and 30 |
| 0x01C7 0CA8 | YENH | Luminance Enhancer |

#### 6.13.1.4 Hardware 3A (H3A)

The Hardware 3A (H3A) module provides control loops for Auto Focus, Auto White Balance and Auto Exposure. There are 2 main components of the H3A module:

- Auto Focus (AF) Engine
- Auto Exposure (AE) & Auto White Balance (AWB) Engine

The AF engine extracts and filters the red, green, and blue data from the input image/video data and provides either the accumulation or peaks of the data in a specified region. The specified region is a two dimensional block of data and is referred to as a “paxel” for the case of AF.

The AE/AWB Engine accumulates the values and checks for saturated values in a sub sampling of the video data. In the case of the AE/AWB, the two-dimensional block of data is referred to as a “window”. The number, dimensions, and starting position of the AF paxels and the AE/AWB windows are separately programmable.

The H3A register memory mapping is shown in Table 6-51.

**Table 6-51. H3A Register Descriptions**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C7 1400 | PID | Peripheral Revision and Class Information |
| 0x01C7 1404 | PCR | Peripheral Control Register |
| 0x01C7 1408 | AFPAX1 | Setup for the AF Engine Paxel Configuration |
| 0x01C7 140C | AFPAX2 | Setup for the AF Engine Paxel Configuration |
| 0x01C7 1410 | AFPAXSTART | Start Position for AF Engine Paxels |
| 0x01C7 1414 | AFIIRSH | Start Position for IIRSH |
| 0x01C7 1418 | AFBUFST | SDRAM/DDRAM Start Address for AF Engine |
| 0x01C7 141C | AFCOEF010 | IIR Filter Coefficient Data for SET 0 |
| 0x01C7 1420 | AFCOEF032 | IIR Filter Coefficient Data for SET 0 |
| 0x01C7 1424 | AFCOEFF054 | IIR Filter Coefficient Data for SET 0 |
| 0x01C7 1428 | AFCOEFF076 | IIR Filter Coefficient Data for SET 0 |
| 0x01C7 142C | AFCOEFF098 | IIR Filter Coefficient Data for SET 0 |
| 0x01C7 1430 | AFCOEFF0010 | IIR Filter Coefficient Data for SET 0 |
| 0x01C7 1434 | AFCOEF110 | IIR Filter Coefficient Data for SET 1 |
| 0x01C7 1438 | AFCOEF132 | IIR Filter Coefficient Data for SET 1 |
| 0x01C7 143C | AFCOEFF154 | IIR Filter Coefficient Data for SET 1 |
| 0x01C7 1440 | AFCOEFF176 | IIR Filter Coefficient Data for SET 1 |
| 0x01C7 1444 | AFCOEFF198 | IIR Filter Coefficient Data for SET 1 |
| 0x01C7 1448 | AFCOEFF1010 | IIR Filter Coefficient Data for SET 1 |
| 0x01C7 144C | AEWWIN1 | Configuration for AE/AWB Windows |
| 0x01C7 1450 | AEWINSTART | Start Position for AE/AWB Windows |
| 0x01C7 1454 | AEWINBLK | Start Position and Height for Black Line of AE/AWB Windows |
| 0x01C7 1458 | AEWSUBWIN | Configuration for Subsample Data in AE/AWB Window |
| 0x01C7 145C | AEWBUFST | SDRAM/DDRAM Start Address for AE/AWB Engine |

#### 6.13.1.4.1 Auto Focus (AF) Engine

The following features are supported by the Auto Focus (AF) Engine.

- Peak Mode in a Paxel (a Paxel is defined as a two dimensional block of pixels).
- Accumulate the maximum Focus Value of each line in a Paxel
- Accumulation/Sum Mode (instead of Peak mode).
- Accumulate Focus Value in a Paxel.
- Up to 36 Paxels in the horizontal direction and up to 128 Paxels in the vertical direction.
- Programmable width and height for the Paxel. All paxels in the frame will be of same size.
- Programmable red, green, and blue position within a 2x2 matrix.
- Separate horizontal start for paxel and filtering.
- Programmable vertical line increments within a paxel.
- Parallel IIR filters configured in a dual-biquad configuration with individual coefficients (2 filters with 11 coefficients each). The filters are intended to compute the sharpness/peaks in the frame to focus on.

#### 6.13.1.4.2 Auto Exposure (AE) and Auto White Balance (AWB) Engine

The following features are supported by the Auto Exposure (AE) and Auto White Balance (AWB) Engine.

- Accumulate clipped pixels along with all non-saturated pixels.
- Up to 36 horizontal windows.
- Up to 128 vertical windows.
- Programmable width and height for the windows. All windows in the frame will be of same size.
- Separate vertical start coordinate and height for a black row of paxels that is different than the remaining color paxels.
- Programmable Horizontal Sampling Points in a window.
- Programmable Vertical Sampling Points in a window.

#### 6.13.1.5 Histogram

The histogram module accepts raw image/video data and bins the pixels on a value (and color separate) basis. The value of the pixel itself is not stored, but each bin contains the number of pixels that are within the appropriate set range. The source of the raw data for the histogram is typically a CCD/CMOS sensor (via the CCDC module) or optionally from DDR2. The following features are supported by the histogram module.

- Up to four regions/areas.
- Separate horizontal/vertical start and end position for each region.
- Pixels from overlapping regions are accumulated into the highest priority region. The priority is: region0 > region1 > region2 > region3.
- Interface to conventional Bayer pattern. Each region can accumulate either 3 or 4 colors.
- 32, 64, 128, or 256 bins per color per region.
- 32, 64, or 128 bins per color for 2 regions.
- 32 or 64 bins per color for 3 or 4 regions.
- Automatic clear of histogram RAM after an ARM read.
- Saturation of the pixel count if the count exceeds the maximum value (each memory location is 20-bit wide).
- Downshift ranging from 0 to 7 bits (maximum bin range 128).
- The last bin (highest range of values) will accumulate any value that is higher than the lower bound.

The Histogram register memory mapping is shown in Table 6-52.

**Table 6-52. Histogram Register Descriptions**

| HEX ADDRESS RANGE | REGISTER ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C7 1000 | PID | Peripheral Revision and Class Information Register |
| 0x01C7 1004 | PCR | Peripheral Control Register |
| 0x01C7 1008 | HIST_CNT | Histogram Control Bits Register |
| 0x01C7 100C | WB_GAIN | White/Channel Balance Settings Register |
| 0x01C7 1010 | R0_HORZ | Region 0 Horizontal Information Register |
| 0x01C7 1014 | R0_VERT | Region 0 Vertical Information Register |
| 0x01C7 1018 | R1_HORZ | Region 1 Horizontal Information Register |
| 0x01C7 101C | R1_VERT | Region 1 Vertical Information Register |
| 0x01C7 1020 | R2_HORZ | Region 2 Horizontal Information Register |
| 0x01C7 1024 | R2_VERT | Region 2 Vertical Information Register |
| 0x01C7 1028 | R3_HORZ | Region 3 Horizontal Information Register |
| 0x01C7 102C | R3_VERT | Region 3 Vertical Information Register |
| 0x01C7 1030 | HIST_ADDR | Histogram Address for Data to be Read Register |
| 0x01C7 1034 | HIST_DATA | Histogram Data That is Read From the Memory Register |
| 0x01C7 1038 | RADD | Read Address From DDR2 Memory Register |
| 0x01C7 103C | RADD_OFF | Read Address Offset for Each Line in the DDR2 Memory Register |
| 0x01C7 1040 | H_V_INFO | Horizontal/Vertical Information Register (Horizontal/Vertical Number of Pixels When Data is Read From DDR2 Memory Information Register) |

### 6.13.1.6 VPFE Electrical Data/Timing

**Table 6-53. Timing Requirements for VPFE PCLK Master/Slave Mode (see Figure 6-46)**

| NO. | | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 1 | $t_{c(PCLK)}$ | Cycle time, PCLK | (A-513 and -594 devices) | 10.204 or 13.33 (1) | | ns |
| | | | (-810 devices) | 9.25 or 13.33 (1) | | ns |
| 2 | $t_{w(PCLKH)}$ | Pulse duration, PCLK high | | 4.4 | | ns |
| 3 | $t_{w(PCLKL)}$ | Pulse duration, PCLK low | | 4.4 | | ns |
| 4 | $t_{t(PCLK)}$ | Transition time, PCLK | | | 3 | ns |

(1) When PCLK sources the clock for both the VPFE and VPBE, the minimum cycle time of 13.33 ns (specified in Table 6-60, *Timing Requirements for VPBE CLK Inputs* for VPBE) ***must*** be met. When PCLK sources the clock for only the VPFE, a minimum cycle time of 10.2 ns may be used for A-513 and -594 devices and a minimum cycle time of 9.25 ns may be used for -810 devices.

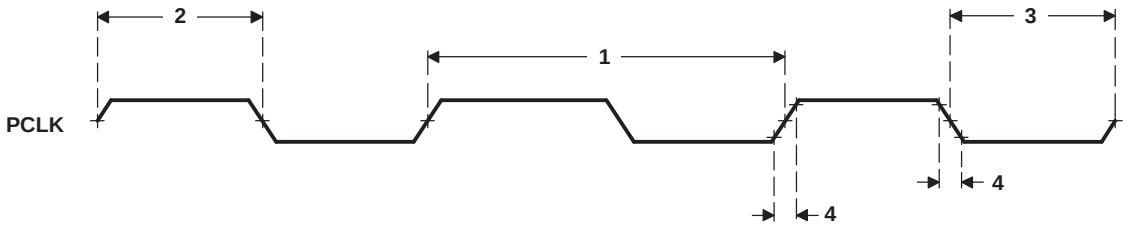



**Figure 6-46. VPFE PCLK Timing**

**Table 6-54. Timing Requirements for VPFE (CCD) Slave Mode (1) (see Figure 6-47)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 5 | $t_{su(CCDV-PCLK)}$ | Setup time, CCD valid before PCLK edge | 3 | | ns |
| 6 | $t_{h(PCLK-CCDV)}$ | Hold time, CCD valid after PCLK edge | 2 | | ns |
| 7 | $t_{su(HDV-PCLK)}$ | Setup time, HD valid before PCLK edge | 3 | | ns |
| 8 | $t_{h(PCLK-HDV)}$ | Hold time, HD valid after PCLK edge | 2 | | ns |
| 9 | $t_{su(VDV-PCLK)}$ | Setup time, VD valid before PCLK edge | 3 | | ns |
| 10 | $t_{h(PCLK-VDV)}$ | Hold time, VD valid after PCLK edge | 2 | | ns |
| 11 | $t_{su(C\_WEV-PCLK)}$ | Setup time, C_WE valid before PCLK edge | 3 | | ns |
| 12 | $t_{h(PCLK-C\_WEV)}$ | Hold time, C_WE valid after PCLK edge | 2 | | ns |
| 13 | $t_{su(C\_FIELDV-PCLK)}$ | Setup time, C_FIELD valid before PCLK edge | 3 | | ns |
| 14 | $t_{h(PCLK-C\_FIELDV)}$ | Hold time, C_FIELD valid after PCLK edge | 2 | | ns |

(1) The VPFE may be configured to operate in either positive or negative edge clocking mode. When in positive edge clocking mode the rising edge of PCLK is referenced. When in negative edge clocking mode the falling edge of PCLK is referenced.

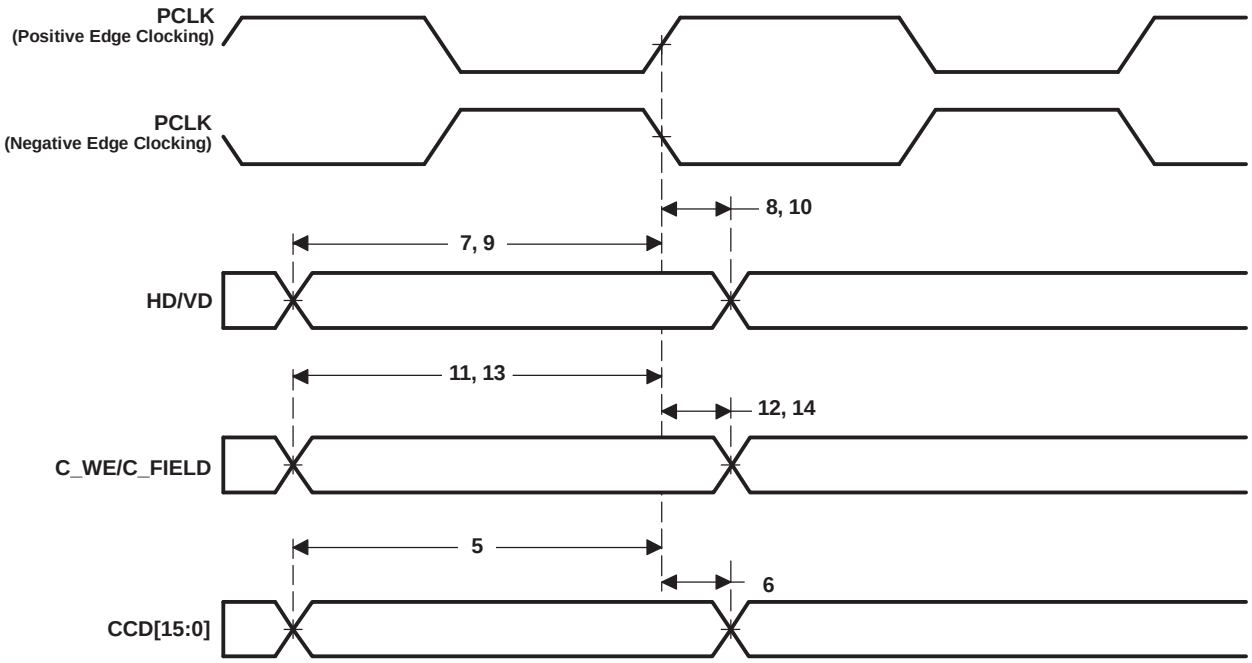



**Figure 6-47. VPFE (CCD) Slave Mode Input Data Timing**

**Table 6-55. Timing Requirements for VPFE (CCD) Master Mode(1) (see Figure 6-48)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 15 | $t_{su(CCDV-PCLK)}$ | Setup time, CCD valid before PCLK edge | 3 | | ns |
| 16 | $t_{h(PCLK-CCDV)}$ | Hold time, CCD valid after PCLK edge | 2 | | ns |
| 23 | $t_{su(CWEV-PCLK)}$ | Setup time, C_WE valid before PCLK edge | 3 | | ns |
| 24 | $t_{h(PCLK-CWEV)}$ | Hold time, C_WE valid after PCLK edge | 2 | | ns |

(1) The VPFE may be configured to operate in either positive or negative edge clocking mode. When in positive edge clocking mode the rising edge of PCLK is referenced. When in negative edge clocking mode the falling edge of PCLK is referenced.

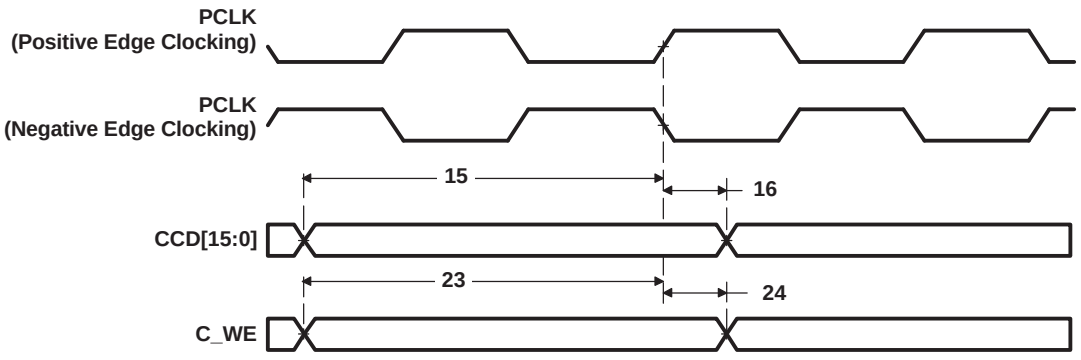



**Figure 6-48. VPFE (CCD) Master Mode Input Data Timing**

**Table 6-56. Switching Characteristics Over Recommended Operating Conditions for VPFE (CCD) Master Mode[1] (see Figure 6-49)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 18 | $t_{d(PCLK-HDV)}$ | Delay time, PCLK edge to HD valid | 0.5 | 8 | ns |
| 20 | $t_{d(PCLK-VDV)}$ | Delay time, PCLK edge to VD valid | 0.5 | 8 | ns |
| 22 | $t_{d(PCLK-C\_FIELDV)}$ | Delay time, PCLK edge to C_FIELD valid | 0.5 | 8.3 | ns |

(1) The VPFE may be configured to operate in either positive or negative edge clocking mode. When in positive edge clocking mode the rising edge of PCLK is referenced. When in negative edge clocking mode the falling edge of PCLK is referenced.

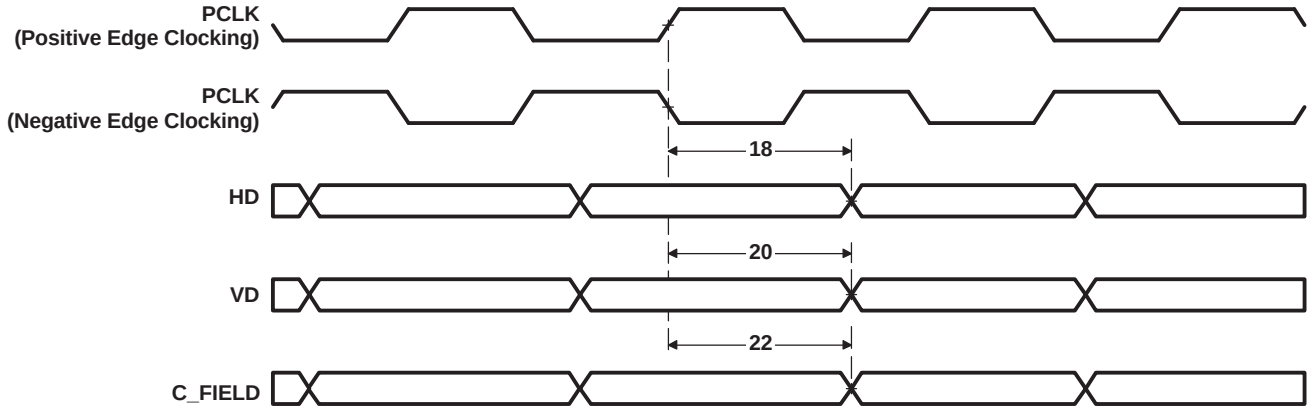



**Figure 6-49. VPFE (CCD) Master Mode Control Output Data Timing**

### *6.13.2 Video Processing Back-End (VPBE)*

The Video Processing Back-End (VPBE) consists of the On-Screen Display (OSD) module, the Video Encoder (VENC) including the Digital LCD (DLCD) and Analog (i.e., DAC) interfaces. The video encoder generates analog video output. The DLCD controller generates digital RGB/YCbCr data output and timing signals.

The VPBE register memory mapping is shown in Table 6-57.

**Table 6-57. VPBE Register Descriptions**

| Address | Register | Description |
|---|---|---|
| 0x01C7 2780 | PID | Peripheral Revision and Class Information Register |
| 0x01C7 2784 | PCR | Peripheral Control Register |

#### 6.13.2.1 On-Screen Display (OSD)

The major function of the OSD module is to gather and blend video data and display/bitmap data before feeding it to the Video Encoder (VENC) in YCbCr format. The video and display data is read from an external memory, typically DDR2. The OSD is programmed via control and parameter registers. The following are the primary features that are supported by the OSD.

- Simultaneous display of two video windows and two OSD windows (VIDWIN0/VIDWIN1 and OSDWIN0/OSDWIN1).
  - Separate enable for each window
  - Programmable width, height, and base starting coordinates for each window
  - External memory address and offset registers for each window
  - Support for x2 and x4 zoom in both the horizontal and vertical direction
  - OSDWIN1 can be used as an attribute window for OSDWIN0

  - Attribute window blinking intervals
  - Field/frame mode for the windows (interlaced/progressive)
  - Eight step blending process between the OSD and video windows
  - Transparency support for the OSD and video data (when a bitmap pixel is zero, there will be no blending for that corresponding video pixel)
  - Resize from VGA to NTSC/PAL (640x480 to 720x576) for both the OSD and video windows
  - Reads in YCbCr data in 4:2:2 format from external memory, with the capability for swapping the order of the CbCr component in the 32-bit word (this is relevant to the two video windows)
  - Support for a ping-pong buffer scheme that can be used for VIDWIN0 (allows for video data to be accessed from two different locations in DDR2)
  - Each OSD window (either one, but not both at the same time) is capable of reading in RGB data (16-bit data with six bits for the green and five bits each for the red and blue colors) instead of bitmap data in YCbCr format restricted to a maximum of 8-bits
  - The OSD bitmap data width is selectable between 1, 2, 4, or 8-bits.
  - Each OSD window supports 16 entries for the bitmap (to index into a 256 entry RAM/ROM CLUT table).
  - Indirect support for 24-bit RGB input data (which will be transformed into 16-bit YCbCr video window data) via the wrapper interface in the VPBE.
- Support for a rectangular cursor window and a programmable background color selection.
  - Programmable color palette with the ability to select between a RAM/ROM table with support for 256 colors.
  - The width, height, and color of the cursor is programmable.
  - The display priority is: Rectangular-Cursor > OSDWIN1 > OSDWIN0 > VIDWIN1 > VIDWIN0 > background color
- Support for attenuation of the YCbCr values for the REC601 standard.

The following restrictions exist in the OSD module.

- Both the OSD windows and VIDWIN1 should be fully contained inside VIDWIN0.
- When one of the OSD windows is set in RGB mode, it cannot overlap with VIDWIN1.
- The OSD cannot support more than 256 color entries in the CLUT RAM/ROM. Some applications require higher number of entries, and one workaround is to use VIDWIN1 as an overlay mimicking the OSD window. Another option is to use the RGB mode for one of the OSD windows which allows for a total of 16-bits for the R, G, and B colors (64K colors).
- The OSD can only read YCbCr in 4:2:2 interleaved format for the video windows. Other formats, either color separate storage or 4:4:4/4:2:0 interleaved data is not supported.
- If the vertical resize filter is enabled for either of the video windows, the maximum horizontal window dimension cannot be greater than 720 currently.
- It is not possible to use both of the CLUT ROMs at the same time. However, one window can use RAM while another uses ROM.
- The 24-bit RGB input mode is only valid for one of the two video windows (programmable) and does not apply to the OSD windows.

The OSD register memory mapping is shown in Table 6-58.

**Table 6-58. OSD Register Descriptions**

| Address | Register | Description |
|---|---|---|
| 0x01C7 2600 | MODE | OSD Mode Register |
| 0x01C7 2604 | VIDWINMD | Video Window Mode Setup |
| 0x01C7 2608 | OSDWIN0MD | OSD Window Mode Setup |
| 0x01C7 260C | OSDWIN1MD | OSD Window 1 Mode Setup (when used as a second OSD window) |
| 0x01C7 260C | OSDATRMD | OSD Attribute Window Mode Setup (when used as an attribute window) |
| 0x01C7 2610 | RECTCUR | Rectangular Cursor Setup |
| 0x01C7 2614 | RSV0 | Reserved |
| 0x01C7 2618 | VIDWIN0OFST | Video Window 0 Offset |
| 0x01C7 261C | VIDWIN1OFST | Video Window 1 Offset |
| 0x01C7 2620 | OSDWIN0OFST | OSD Window 0 Offset |
| 0x01C7 2624 | OSDWIN1OFST | OSD Window 1 Offset |
| 0x01C7 2628 | RSV1 | Reserved |
| 0x01C7 262C | VIDWIN0ADR | Video Window 0 Address |
| 0x01C7 2630 | VIDWIN1ADR | Video Window 1 Address |
| 0x01C7 2634 | RSV2 | Reserved |
| 0x01C7 2638 | OSDWIN0ADR | OSD Window 0 Address |
| 0x01C7 263C | OSDWIN1ADR | OSD Window 1 Address |
| 0x01C7 2640 | BASEPX | Base Pixel X |
| 0x01C7 2644 | BASEPY | Base Pixel Y |
| 0x01C7 2648 | VIDWIN0XP | Video Window 0 X-Position |
| 0x01C7 264C | VIDWIN0YP | Video Window 0 Y-Position |
| 0x01C7 2650 | VIDWIN0XL | Video Window 0 X-Size |
| 0x01C7 2654 | VIDWIN0YL | Video Window 0 Y-Size |
| 0x01C7 2658 | VIDWIN1XP | Video Window 1 X-Position |
| 0x01C7 265C | VIDWIN1YP | Video Window 1 Y-Position |
| 0x01C7 2660 | VIDWIN1XL | Video Window 1 X-Size |
| 0x01C7 2664 | VIDWIN1YL | Video Window 1 Y-Size |
| 0x01C7 2668 | OSDWIN0XP | OSD Bitmap Window 0 X-Position |
| 0x01C7 266C | OSDWIN0YP | OSD Bitmap Window 0 Y-Position |
| 0x01C7 2670 | OSDWIN0XL | OSD Bitmap Window 0 X-Size |
| 0x01C7 2674 | OSDWIN0YL | OSD Bitmap Window 0 Y-Size |
| 0x01C7 2678 | OSDWIN1XP | OSD Bitmap Window 1 X-Position |
| 0x01C7 267C | OSDWIN1YP | OSD Bitmap Window 1 Y-Position |
| 0x01C7 2680 | OSDWIN1XL | OSD Bitmap Window 1 X-Size |
| 0x01C7 2684 | OSDWIN1YL | OSD Bitmap Window 1 Y-Size |
| 0x01C7 2688 | CURXP | Rectangular Cursor Window X-Position |
| 0x01C7 268C | CURYP | Rectangular Cursor Window Y-Position |
| 0x01C7 2690 | CURXL | Rectangular Cursor Window X-Size |
| 0x01C7 2694 | CURYL | Rectangular Cursor Window Y-Size |
| 0x01C7 2698 | RSV3 | Reserved |
| 0x01C7 269C | RSV4 | Reserved |
| 0x01C7 26A0 | W0BMP01 | Window 0 Bitmap Value to Palette Map 0/1 |
| 0x01C7 26A4 | W0BMP23 | Window 0 Bitmap Value to Palette Map 2/3 |
| 0x01C7 26A8 | W0BMP45 | Window 0 Bitmap Value to Palette Map 4/5 |
| 0x01C7 26AC | W0BMP67 | Window 0 Bitmap Value to Palette Map 6/7 |
| 0x01C7 26B0 | W0BMP89 | Window 0 Bitmap Value to Palette Map 8/9 |
| 0x01C7 26B4 | W0BMPAB | Window 0 Bitmap Value to Palette Map A/B |

**Table 6-58. OSD Register Descriptions (continued)**

| | | |
|---|---|---|
| 0x01C7 26B8 | W0BMPCD | Window 0 Bitmap Value to Palette Map C/D |
| 0x01C7 26BC | W0BMPEF | Window 0 Bitmap Value to Palette Map E/F |
| 0x01C7 26C0 | W1BMP01 | Window 1 Bitmap Value to Palette Map 0/1 |
| 0x01C7 26C4 | W1BMP23 | Window 1 Bitmap Value to Palette Map 2/3 |
| 0x01C7 26C8 | W1BMP45 | Window 1 Bitmap Value to Palette Map 4/5 |
| 0x01C7 26CC | W1BMP67 | Window 1 Bitmap Value to Palette Map 6/7 |
| 0x01C7 26D0 | W1BMP89 | Window 1 Bitmap Value to Palette Map 8/9 |
| 0x01C7 26D4 | W1BMPAB | Window 1 Bitmap Value to Palette Map A/B |
| 0x01C7 26D8 | W1BMPCD | Window 1 Bitmap Value to Palette Map C/D |
| 0x01C7 26DC | W1BMPEF | Window 1 Bitmap Value to Palette Map E/F |
| 0x01C7 26E0 | - | Reserved |
| 0x01C7 26E4 | RSV5 | Reserved |
| 0x01C7 26E8 | MISCCTL | Miscellaneous Control |
| 0x01C7 26EC | CLUTRAMYCB | CLUT RAMYCB Setup |
| 0x01C7 26F0 | CLUTRAMCR | CLUT RAM Setup |
| 0x01C7 26F4 | TRANSPVAL | CLUT RAM Setup |
| 0x01C7 26F8 | RSV6 | Reserved |
| 0x01C7 26FC | PPVWIN0ADR | Ping-Pong Video Window 0 Address |

#### 6.13.2.2 Video Encoder (VENC)

Analog/DACs interface of the Video Encoder (VENC) supports the following features.

- Master Clock Input - 27MHz (x2 Upsampling)
- SDTV Support
  - Composite NTSC-M, PAL-B/D/G/H/I
  - S-Video (Y/C)
  - Component YPbPr (SMPTE/EBU N10, Betacam, MII)
  - RGB
  - Non-Interlace
  - CGMS/WSS
  - Line 21 Closed Caption Data Encoding
  - Chroma Low Pass Filter 1.5MHz/3MHz
  - Programmable SC-H phase
- HDTV Support
  - Progressive Output (525p/625p)
  - Component YPbPr
  - RGB
  - CGMS/WSS
- 4 10-bit Over-Sampling D/A Converters
- Optional 7.5% Pedestal
- 16-235/0-255 Input Amplitude Selectable
- Programmable Luma Delay
- Master/Slave Operation
- Internal Color Bar Generation (100%/75%)

The Digital LCD Controller (DLCD) of the VENC supports the following features.

- Programmable DCLK
- Various Output Formats
  - YCbCr 16bit
  - YCbCr 8bit
  - ITU-R BT. 656
  - Parallel RGB 24bit
- Low Pass Filter for Digital RGB Output
- Programmable Timing Generator
- Master/Slave Operation
- Internal Color Bar Generation (100%/75%)

The VENC register memory mapping including the Digital LCD and DACs is shown in Table 6-59.

**Table 6-59. VENC (Including Digital LCD and DACs) Register Descriptions**

| ADDRESS | REGISTER | DESCRIPTION |
|---|---|---|
| 0x01C7 2400 | VMOD | Video Mode |
| 0x01C7 2404 | VIDCTL | Video Interface I/O Control |
| 0x01C7 2408 | VDPRO | Video Data Processing |
| 0x01C7 240C | SYNCCTL | Sync Control |
| 0x01C7 2410 | HSPLS | Horizontal Sync Pulse Width |
| 0x01C7 2414 | VSPLS | Vertical Sync Pulse Width |
| 0x01C7 2418 | HINT | Horizontal Interval |
| 0x01C7 241C | HSTART | Horizontal Valid Data Start Position |
| 0x01C7 2420 | HVALID | Horizontal Data Valid Range |
| 0x01C7 2424 | VINT | Vertical Interval |
| 0x01C7 2428 | VSTART | Vertical Valid Data Start Position |
| 0x01C7 242C | VVALID | Vertical Data Valid Range |
| 0x01C7 2430 | HSDLY | Horizontal Sync Delay |
| 0x01C7 2434 | VSDLY | Vertical Sync Delay |
| 0x01C7 2438 | YCCTL | YCbCr Control |
| 0x01C7 243C | RGBCTL | RGB Control |
| 0x01C7 2440 | RGBCLP | RGB Level Clipping |
| 0x01C7 2444 | LINECTL | Line ID Control |
| 0x01C7 2448 | CULLLINE | Culling Line Control |
| 0x01C7 244C | LCDOUT | LCD Output Signal Control |
| 0x01C7 2450 | BRTS | Brightness Start Position Signal Control |
| 0x01C7 2454 | BRTW | Brightness Width Signal Control |
| 0x01C7 2458 | ACCTL | LCD_AC Signal Control |
| 0x01C7 245C | PWMP | PWM Start Position Signal Control |
| 0x01C7 2460 | PWMW | PWM Width Signal Control |
| 0x01C7 2464 | DCLKCTL | DCLK Control |
| 0x01C7 2468 | DCLKPTN0 | DCLK Pattern 0 |
| 0x01C7 246C | DCLKPTN1 | DCLK Pattern 1 |
| 0x01C7 2470 | DCLKPTN2 | DCLK Pattern 2 |
| 0x01C7 2474 | DCLKPTN3 | DCLK Pattern 3 |
| 0x01C7 2478 | DCLKPTN0A | DCLK Auxiliary Pattern 0 |
| 0x01C7 247C | DCLKPTN1A | DCLK Auxiliary Pattern 1 |
| 0x01C7 2480 | DCLKPTN2A | DCLK Auxiliary Pattern 2 |

**Table 6-59. VENC (Including Digital LCD and DACs) Register Descriptions (continued)**

| ADDRESS | REGISTER | DESCRIPTION |
|---|---|---|
| 0x01C7 2484 | DCLKPTN3A | DCLK Auxiliary Pattern 3 |
| 0x01C7 2488 | DCLKHS | Horizontal DCLK Mask Start |
| 0x01C7 248C | DCLKHSA | Horizontal Auxiliary DCLK Mask Start |
| 0x01C7 2490 | DCLKHR | Horizontal DCLK Mask Range |
| 0x01C7 2494 | DCLKVS | Vertical DCLK Mask Start |
| 0x01C7 2498 | DCLKVR | Vertical DCLK Mask Range |
| 0x01C7 249C | CAPCTL | Caption Control |
| 0x01C7 24A0 | CAPDO | Caption Data Odd Field |
| 0x01C7 24A4 | CAPDE | Caption Data Even Field |
| 0x01C7 24A8 | ATR0 | Video Attribute Data # 0 |
| 0x01C7 24AC | ATR1 | Video Attribute Data # 1 |
| 0x01C7 24B0 | ATR2 | Video Attribute Data # 2 |
| 0x01C7 24B4 | | Reserved |
| 0x01C7 24B4 | | |
| 0x01C7 24B4 | | |
| 0x01C7 24B4 | | |
| 0x01C7 24B8 | VSTAT | Video Status |
| 0x01C7 24BC | | Reserved |
| 0x01C7 24C0 | | |
| 0x01C7 24C4 | DACTST | DAC Test |
| 0x01C7 24C8 | YCOLVL | YOUT and COUT Levels |
| 0x01C7 24CC | SCPROG | Sub-Carrier Programming |
| 0x01C7 24D0 | | Reserved |
| 0x01C7 24D4 | | |
| 0x01C7 24D8 | | |
| 0x01C7 24DC | CVBS | Composite Mode |
| 0x01C7 24E0 | CMPNT | Component Mode |
| 0x01C7 24E4 | ETMG0 | CVBS Timing Control 0 |
| 0x01C7 24E8 | ETMG1 | CVBS Timing Control 1 |
| 0x01C7 24EC | ETMG2 | Component Timing Control 0 |
| 0x01C7 24F0 | ETMG3 | Component Timing Control 1 |
| 0x01C7 24F4 | DACSEL | DAC Output Select |
| 0x01C7 24F8 | | Reserved |
| 0x01C7 24FC | | |
| 0x01C7 2500 | ARGBX0 | Analog RGB Matrix 0 |
| 0x01C7 2504 | ARGBX1 | Analog RGB Matrix 1 |
| 0x01C7 2508 | ARGBX2 | Analog RGB Matrix 2 |
| 0x01C7 250C | ARGBX3 | Analog RGB Matrix 3 |
| 0x01C7 2510 | ARGBX4 | Analog RGB Matrix 4 |
| 0x01C7 2514 | DRGBX0 | Digital RGB Matrix 0 |
| 0x01C7 2518 | DRGBX1 | Digital RGB Matrix 1 |
| 0x01C7 251C | DRGBX2 | Digital RGB Matrix 2 |
| 0x01C7 2520 | DRGBX3 | Digital RGB Matrix 3 |
| 0x01C7 2524 | DRGBX4 | Digital RGB Matrix 4 |
| 0x01C7 2528 | VSTARTA | Vertical Data Valid Start Position for Even Field |
| 0x01C7 252C | OSDCLK0 | OSD Clock Control 0 |
| 0x01C7 2530 | OSDCLK1 | OSD Clock Control 1 |

**Table 6-59. VENC (Including Digital LCD and DACs) Register Descriptions (continued)**

| ADDRESS | REGISTER | DESCRIPTION |
|---|---|---|
| 0x01C7 2534 | HVLDCL0 | Horizontal Valid Culling Control 0 |
| 0x01C7 2538 | HVLDCL1 | Horizontal Valid Culling Control 1 |
| 0x01C7 253C | OSDHADV | OSD Horizontal Sync Advance |

### 6.13.2.3 VPBE Electrical Data/Timing

**Table 6-60. Timing Requirements for VPBE CLK Inputs (see Figure 6-50)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(PCLK)}$ | Cycle time, PCLK | 13.33 | 160 | ns |
| 2 | $t_{w(PCLKH)}$ | Pulse duration, PCLK high | 5.7 | | ns |
| 3 | $t_{w(PCLKL)}$ | Pulse duration, PCLK low | 5.7 | | ns |
| 4 | $t_{t(PCLK)}$ | Transition time, PCLK | | 3 | ns |
| 5 | $t_{c(VPBECLK)}$ | Cycle time, VPBECLK | 13.33 | 160 | ns |
| 6 | $t_{w(VPBECLKH)}$ | Pulse duration, VPBECLK high | 5.7 | | ns |
| 7 | $t_{w(VPBECLKL)}$ | Pulse duration, VPBECLK low | 5.7 | | ns |
| 8 | $t_{t(VPBECLK)}$ | Transition time, VPBECLK | | 3 | ns |

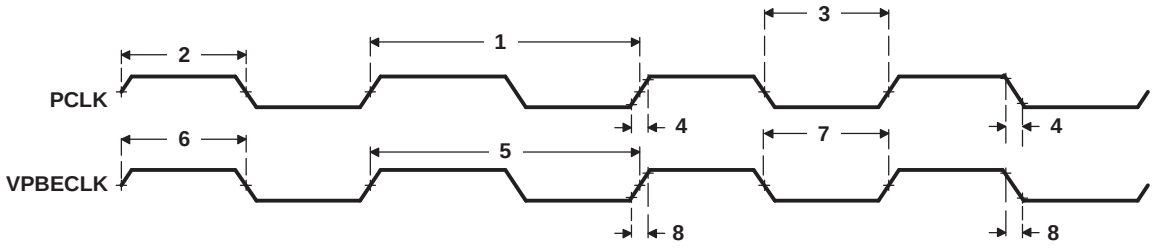
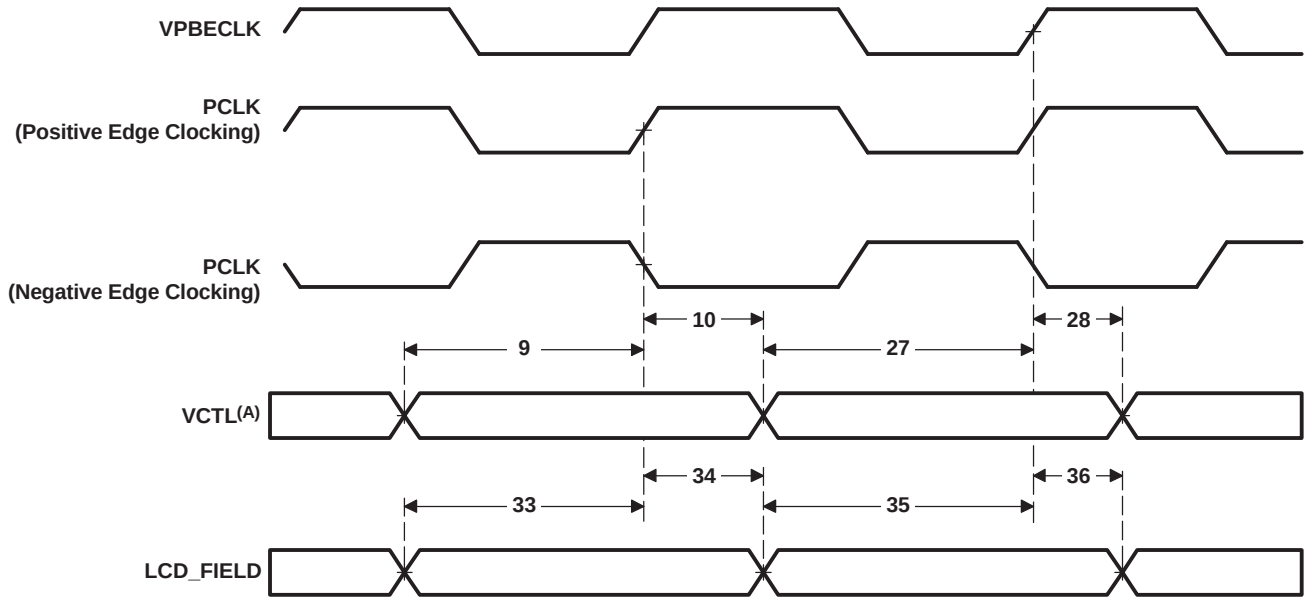



**Figure 6-50. VPBE PCLK and VPBECLK Timing**

**Table 6-61. Timing Requirements for VPBE Control Input With Respect to PCLK and VPBECLK(1) (see Figure 6-51)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 9 | $t_{su(VCTLV-PCLK)}$ | Setup time, VCTL valid before PCLK edge | 2 | | ns |
| 10 | $t_{h(PCLK-VCTLV)}$ | Hold time, VCTL valid after PCLK edge | 0.5 | | ns |
| 27 | $t_{su(VCTLV-VPBECLK)}$ | Setup time, VCTL valid before VPBECLK rising edge | 2 | | ns |
| 28 | $t_{h(VPBECLK-VCTLV)}$ | Hold time, VCTL valid after VPBECLK rising edge | 0.5 | | ns |
| 33 | $t_{su(FIELD-PCLK)}$ | Setup time, LCD_FIELD valid before PCLK edge | 5P(2) | | ns |
| 34 | $t_{h(PCLK-FIELD)}$ | Hold time, LCD_FIELD valid after PCLK edge | 5P(2) | | ns |
| 35 | $t_{su(FIELD-VPBECLK)}$ | Setup time, LCD_FIELD valid before VPBECLK edge | 5P(2) | | ns |
| 36 | $t_{h(VPBECLK-FIELD)}$ | Hold time, LCD_FIELD valid after VPBECLK edge | 5P(2) | | ns |

(1) PCLK may be configured to operate in either positive or negative edge clocking mode. When in positive edge clocking mode, the rising edge of PCLK is referenced. When in negative edge clocking mode, the falling edge of PCLK is referenced.
(2) P = 1/(VCLKIN clock frequency) in ns. VCLKIN is either PCLK or VPBECLK, whichever is used.



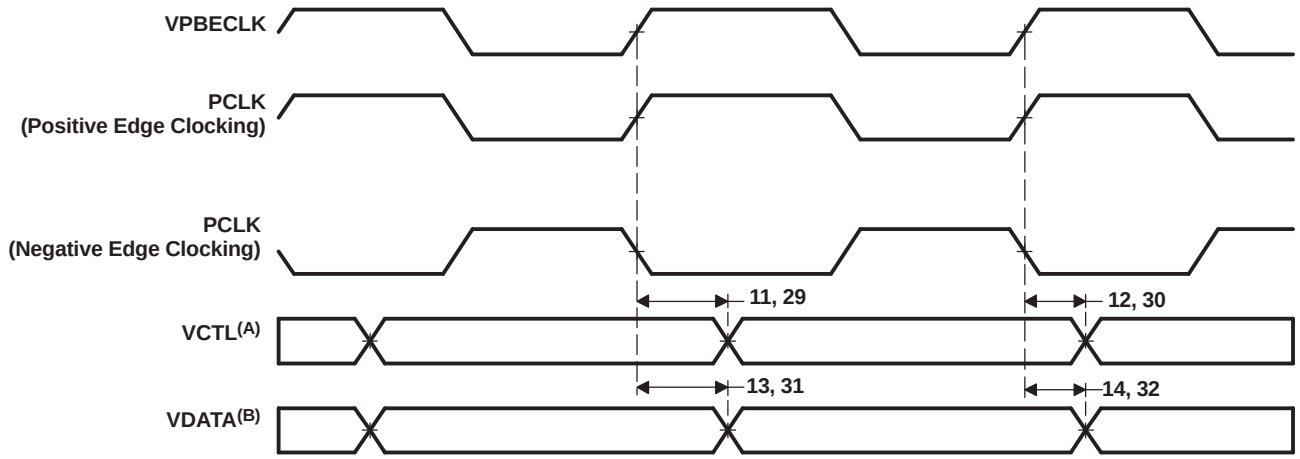

A. VCTL = HSYNC and VSYNC

**Figure 6-51. VPBE Input Timing With Respect to PCLK and VPBECLK**

**Table 6-62. Switching Characteristics Over Recommended Operating Conditions for VPBE Control and Data Output With Respect to PCLK and VPBECLK(1) (see Figure 6-52)**

| NO. | PARAMETER | | A-513 | | -594 -810 | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| 11 | $t_{d(PCLK-VCTLV)}$ | Delay time, PCLK edge to VCTL valid | | 13.3 | | 13.3 | ns |
| 12 | $t_{d(PCLK-VCTLIV)}$ | Delay time, PCLK edge to VCTL invalid | 2 | | 2 | | ns |
| 13 | $t_{d(PCLK-VDATAV)}$ | Delay time, PCLK edge to VDATA valid | | 13.6 | | 13.3 | ns |
| 14 | $t_{d(PCLK-VDATAIV)}$ | Delay time, PCLK edge to VDATA invalid | 2 | | 2 | | ns |
| 29 | $t_{d(VPBECLK-VCTLV)}$ | Delay time, VPBECLK rising edge to VCTL valid | | 13.3 | | 13.3 | ns |
| 30 | $t_{d(VPBECLK-VCTLIV)}$ | Delay time, VPBECLK rising edge to VCTL invalid | 2 | | 2 | | ns |
| 31 | $t_{d(VPBECLK-VDATAV)}$ | Delay time, VPBECLK rising edge to VDATA valid | | 13.6 | | 13.3 | ns |
| 32 | $t_{d(VPBECLK-VDATAIV)}$ | Delay time, VPBECLK rising edge to VDATA invalid | 2 | | 2 | | ns |

(1) PCLK may be configured to operate in either positive or negative edge clocking mode. When in positive edge clocking mode, the rising edge of PCLK is referenced. When in negative edge clocking mode, the falling edge of PCLK is referenced.



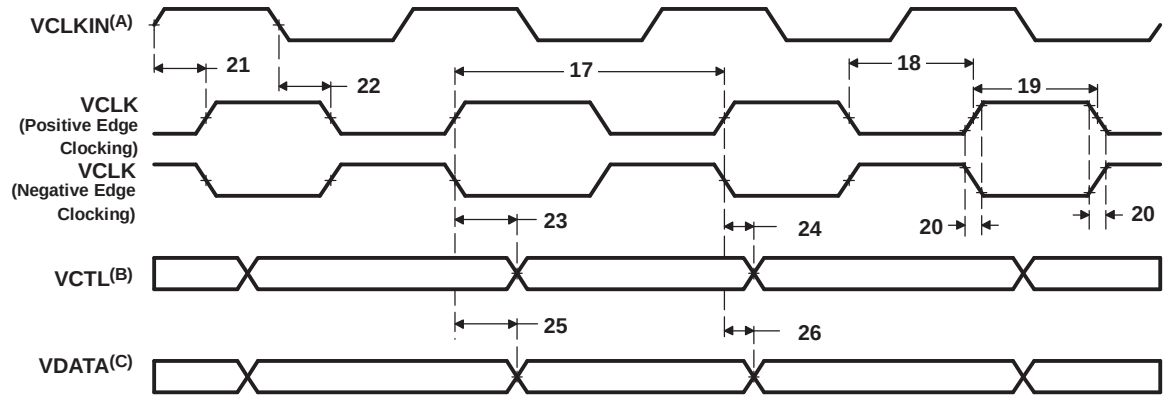

A. VCTL = HSYNC, VSYNC, LCD_FIELD, and LCD_OE
B. VDATA = COUT[7:0], YOUT[7:0], R[7:0], G[7:0], and B[7:0]

**Figure 6-52. VPBE Output Timing With Respect to PCLK and VPBECLK**

**Table 6-63. Switching Characteristics Over Recommended Operating Conditions for VPBE Control and Data Output With Respect to VCLK(1) (2) (see Figure 6-53)**

| NO. | PARAMETER | | MODE(3) | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 17 | $t_{c(VCLK)}$ | Cycle time, VCLK | | 13.33 | 160 | ns |
| 18 | $t_{w(VCLKH)}$ | Pulse duration, VCLK high (positive-edge clocking) | | H - 1.3(4) | H - 0.3(4) | ns |
| | | Pulse duration, VCLK high (negative-edge clocking) | | L - 1.3(4) | L - 0.3(4) | ns |
| 19 | $t_{w(VCLKL)}$ | Pulse duration, VCLK low (positive-edge clocking) | | L + 0.3(4) | L + 1.3(4) | ns |
| | | Pulse duration, VCLK low (negative-edge clocking) | | H + 0.3(4) | H + 1.3(4) | ns |
| 20 | $t_{t(VCLK)}$ | Transition time, VCLK | | | 3 | ns |
| 21 | $t_{d(VCLKINH-VCLKH)}$ | Delay time, VCLKIN high to VCLK high | | 2 | 12 | ns |
| 22 | $t_{d(VCLKINL-VCLKL)}$ | Delay time, VCLKIN low to VCLK low | | 2 | 12 | ns |
| 23 | $t_{d(VCLK-VCTLV)}$ | Delay time, VCLK negative edge to VCTL valid | | | 7.5 | ns |
| | | Delay time, VCLK positive edge to VCTL valid | | | 6.9 | ns |
| 24 | $t_{d(VCLKL-VCTLIV)}$ | Delay time, VCLK negative edge to VCTL invalid | | 2 | | ns |
| | | Delay time, VCLK positive edge to VCTL invalid | | 1.5 | | ns |
| 25 | $t_{d(VCLK-VDATAV)}$ | Delay time, VCLK negative edge to VDATA valid | | | 6.8 | ns |
| | | Delay time, VCLK positive edge to VDATA valid | | | 6.3 | ns |
| 26 | $t_{d(VCLKL-VDATAIV)}$ | Delay time, VCLK negative edge to VDATA invalid | RGB | 2.1 | | ns |
| | | | YCC | 2.5 | | ns |
| | | Delay time, VCLK positive edge to VDATA invalid | RGB | 1.9 | | ns |
| | | | YCC | 2.1 | | ns |

(1) The VPBE may be configured to operate in either positive or negative edge clocking mode. When in positive edge clocking mode, the rising edge of VCLK is referenced. When in negative edge clocking mode, the falling edge of VCLK is referenced.
(2) VCLKIN = PCLK or VPBECLK
(3) RGB and YCC modes utilize different data pins. RGB mode uses data pins: R[7:0], G[7:0], and B[7:0]. YCC mode uses data pins: COUT[7:0] and YOUT[7:0].
(4) H and L are the high and low pulse widths of the input clock to the VPBE, respectively. For example, if VPBECLK is used as the input clock and it has a high pulse duration of 6.67 ns, the resulting high pulse duration of VCLK, if positive-edge clocking is selected, will be a MAX of 6.37 ns and a MIN of 5.27 ns.




A. VCLKIN = PCLK or VPBECLK
B. VCTL = HSYNC, VSYNC, LCD_FIELD, and LCD_OE
C. VDATA = COUT[7:0], YOUT[7:0], R[7:0], G[7:0], and B[7:0]

**Figure 6-53. VPBE Control and Data Output Timing With Respect to VCLK**

### 6.13.2.4 DAC Electrical Data/Timing

**Table 6-64. Switching Characteristics Over Recommended Operating Conditions for DAC Static Specifications**

| NO. | PARAMETER | TEST CONDITIONS | A-513, -594, -810 | | | UNIT |
|---|---|---|---|---|---|---|
| | | | MIN | TYP | MAX | |
| | **DC Accuracy**<br>Integral Non-Linearity (INL)<br>Differential Non-Linearity (DNL) | | -1.0<br>-0.5 | | 1.0<br>0.5 | LSB<br>LSB |
| | **Analog Output**<br>Offset Error<br>Gain Error<br>Full-Scale Output Voltage | $R_{LOAD}$ = 500 Ω | | 0.5<br>5<br>500 | | LSB<br>%$F_S$<br>$mV_{PP}$ |
| | Output Capacitance | | | 200 | | pF |
| | **Reference**<br>Reference Voltage Range ($V_{REF}$)<br>Full-Scale Current Adjust Resistor (RBIAS) | | 0.475<br>3.8 | 0.5<br>4.0 | 0.525<br>4.2 | V<br>kΩ |

**Table 6-65. Switching Characteristics Over Recommended Operating Conditions for DAC Dynamic Specifications**

| NO. | PARAMETER | TEST CONDITIONS | A-513, -594, -810 | | | UNIT |
|---|---|---|---|---|---|---|
| | | | MIN | TYP | MAX | |
| | Output Update Rate ($F_{CLK}$) | | | 27 | 60 | MHz |
| | Signal Bandwidth | | | 6 | | MHz |
| | SFDR to Nyquist | $F_{CLK}$ = 27 MHz<br>$F_{OUT}$ = 2.0 MHz | | 60 | | dB |
| | | $F_{CLK}$ = 60 MHz<br>$F_{OUT}$ = 2.0 MHz | | 60 | | dB |
| | SFDR within Bandwidth | $F_{CLK}$ = 27 MHz<br>$F_{OUT}$ = 2.0 MHz | | 60 | | dB |
| | | $F_{CLK}$ = 60 MHz<br>$F_{OUT}$ = 2.0 MHz | | 60 | | dB |
| | PSRR Over Temp vs Power Supply | | 50 | | | dB |

The DM6446's analog video DAC outputs are designed to drive a 500-Ω load. Figure 6-54 describes a typical circuit that will permit connecting the analog video output from the DM6446 device to standard 75-Ω impedance video systems. Another solution is to use a Video Amplifier with an integrated filter to provide a complete solution to the typical output circuit shown in Figure 6-54.

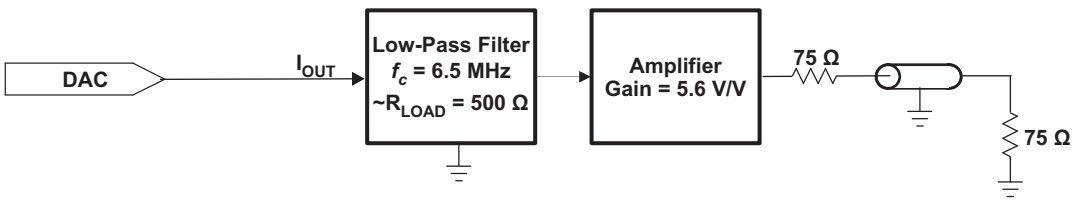



**Figure 6-54. Typical Output Circuit for NTSC/PAL Video From DACs**

## 6.14 Host-Port Interface (HPI)

The Host Port Interface (HPI) provides a parallel port through which an external host processor can access the DM6446 memory space. The host device is asynchronous to the DM6446 clocks and functions as a master to the HPI interface. The HPI enables a host device and DM6446 to exchange information via internal or external memory. Both the host and DM6446 can access the HPI control register (HPIC) and the HPI address registers (HPIAR, HPIAW). The host can access the HPI data register (HPID) and the HPIC by using the external data and interface control signals.

The HPI interface shares the DaVinci EMIFA 16-bit data bus pins for multiplexed address/data and supports the following modes:

- 16 Bit Multiplexed mode / dual half-word cycles (16 bit host data bus/32 bit memory width)
- ARM ROM supports booting of DM6446 ARM processor from an external processor

The HPI registers are summarized in Table 6-66. For more detailed information on the HPI peripheral, see the *TMS320DM644x DMSoC Host Port Interface (HPI) User's Guide* (literature number SPRUE97).

**Table 6-66. Host-Port Interface (HPI) Register Descriptions**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C4 0030 | HPI_CTL | Host-Port Interface Configuration Register |
| 0x01C6 7800 | HPI_PID | |
| 0x01C6 7804 | HPIPWREMU | HPI Power and Emulation Management Register |
| 0x01C6 7808 - 0x01C6 782F | – | Reserved |
| 0x01C6 7830 | HPIC | Host-Port Interface Control Register |
| 0x01C6 7834 | HPIAW | Host-Port Interface Write Address Register |
| 0x01C6 7838 | HPIAR | Host-Port Interface Read Address Register |
| 0x01C6 783C - 0x01C6 7FFF | – | Reserved |

The HPI_CTL register sets the owner of HPIA(R/W) and HPIC registers for HPI address and control. The details for HPI_CTL are shown in Figure 6-55 and Table 6-67.

**Figure 6-55. HPI_CTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0000000000000000 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CTL MODE | ADD MODE | TIMOUT | | | | | | | |
| R-0 | | | | | | R/W-0 | R/W-0 | R/W-10000000 | | | | | | | |

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

**Table 6-67. HPI_CTL Register Description**

| Name | Description |
|---|---|
| CTLMODE | HPIC register write access<br>0 = External Host<br>1 = DM6446 (if ADDMODE = 1) |
| ADDMODE | HPIA register write access<br>0 = External Host<br>1 = DM6446 |
| TIMOUT | Host burst write timeout value |

### 6.14.1 Host-Port Interface (HPI) Electrical Data/Timing

**Table 6-68. Timing Requirements for Host-Port Interface Cycles(1) (2) (see Figure 6-56 through Figure 6-57)**

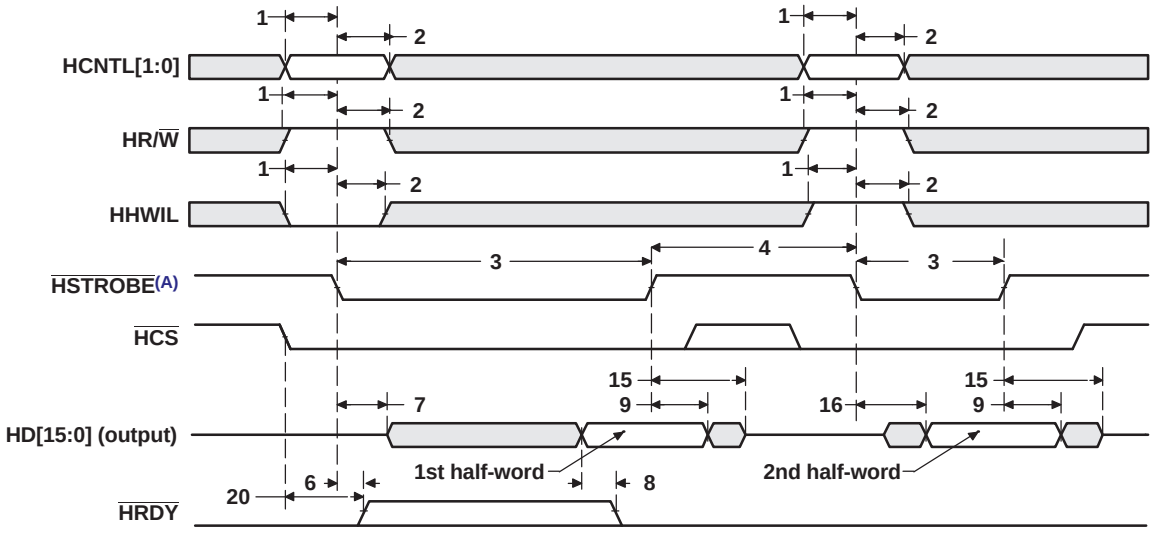
| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(SELV-HSTBL)}$ | Setup time, select signals(3) valid before $\overline{\text{HSTROBE}}$ low | 5 | | ns |
| 2 | $t_{h(HSTBL-SELV)}$ | Hold time, select signals(3) valid after $\overline{\text{HSTROBE}}$ low | 2 | | ns |
| 3 | $t_{w(HSTBL)}$ | Pulse duration, $\overline{\text{HSTROBE}}$ low | 15 | | ns |
| 4 | $t_{w(HSTBH)}$ | Pulse duration, $\overline{\text{HSTROBE}}$ high between consecutive accesses | 2P | | ns |
| 12 | $t_{su(HDV-HSTBH)}$ | Setup time, host data valid before $\overline{\text{HSTROBE}}$ high | 5 | | ns |
| 13 | $t_{h(HSTBH-HDV)}$ | Hold time, host data valid after $\overline{\text{HSTROBE}}$ high | 0 | | ns |
| 14 | $t_{h(HRDYL-HSTBH)}$ | Hold time, $\overline{\text{HSTROBE}}$ high after $\overline{\text{HRDY}}$ low. $\overline{\text{HSTROBE}}$ should not be inactivated until $\overline{\text{HRDY}}$ is active (low); otherwise, HPI writes will not complete properly. | 2 | | ns |

(1) $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.
(2) P = 1/CPU clock frequency in ns. For example, when running parts at 594 MHz, use P = 1.68 ns; when running parts at 810 MHz, use P = 1.23 ns.
(3) Select signals include: HCNTL[1:0] and HR/$\overline{\text{W}}$. For HPI16 mode only, select signals also include HHWIL.

**Table 6-69. Switching Characteristics Over Recommended Operating Conditions During Host-Port Interface Cycles(1) (see Figure 6-56 through Figure 6-57)**

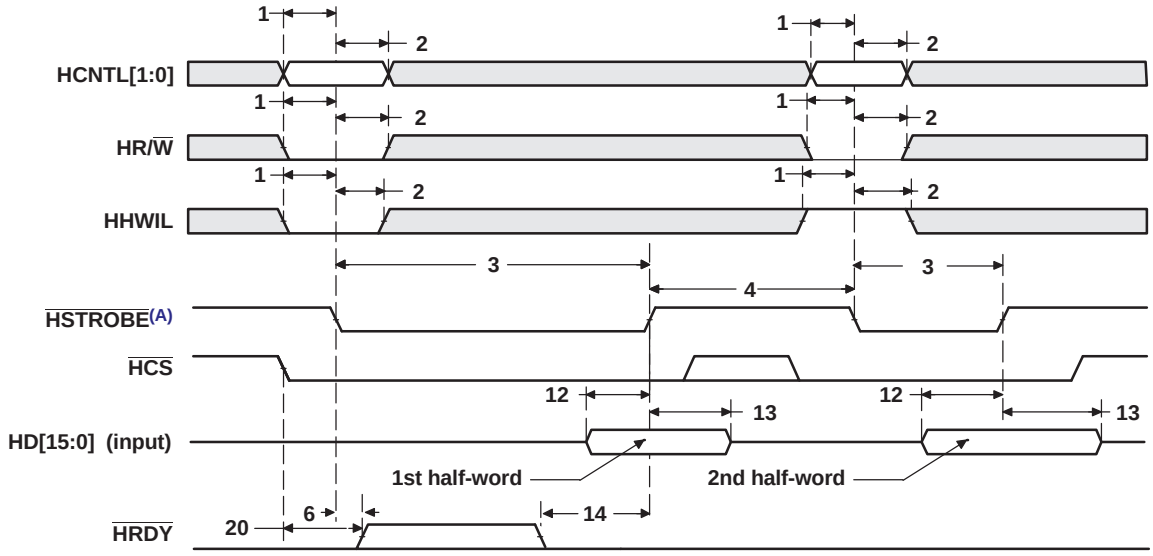
| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 6 | $t_{d(HSTBL-HRDYH)}$ | Delay time, $\overline{\text{HSTROBE}}$ low to $\overline{\text{HRDY}}$ high(2) | 0 | 12 | ns |
| 7 | $t_{d(HSTBL-HDLZ)}$ | Delay time, $\overline{\text{HSTROBE}}$ low to HD low impedance for an HPI read | 0 | | ns |
| 8 | $t_{d(HDV-HRDYL)}$ | Delay time, HD valid to $\overline{\text{HRDY}}$ low | 0 | | ns |
| 9 | $t_{oh(HSTBH-HDV)}$ | Output hold time, HD valid after $\overline{\text{HSTROBE}}$ high | 1.5 | | ns |
| 15 | $t_{d(HSTBH-HDHZ)}$ | Delay time, $\overline{\text{HSTROBE}}$ high to HD high impedance | | 7 | ns |
| 16 | $t_{d(HSTBL-HDV)}$ | Delay time, $\overline{\text{HSTROBE}}$ low to HD valid (HPI16 mode, 2nd half-word only) | | 15 | ns |
| 20 | $t_{d(HCSL-HRDYH)}$ | Delay time, $\overline{\text{HCS}}$ low to $\overline{\text{HRDY}}$ high | 0 | 12 | ns |

(1) $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.
(2) This parameter is used during HPID reads and writes. For reads, at the beginning of the first half-word transfer (HPI16) on the falling edge of $\overline{\text{HSTROBE}}$, the HPI sends the request to the EDMA3 internal address generation hardware, and $\overline{\text{HRDY}}$ remains high until the EDMA3 internal address generation hardware loads the requested data into HPID. For writes, $\overline{\text{HRDY}}$ goes high if the internal write buffer is full.




A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

**Figure 6-56. HPI16 Read Timing**




A. HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

**Figure 6-57. HPI16 Write Timing**

## 6.15 USB 2.0

The DM6446 USB2.0 peripheral supports the following features:

- USB 2.0 peripheral at speeds high speed (HS: 480 Mb/s) and full speed (FS: 12 Mb/s)
- USB 2.0 host at speeds HS, FS, and low speed (LS: 1.5 Mb/s)
- All transfer modes (control, bulk, interrupt, and isochronous)
- 4 Transmit (TX) and 4 Receive (RX) endpoints in addition to endpoint 0
- FIFO RAM
  - 4K endpoint
  - Programmable size
- Connects to a standard UTMI+ PHY with a 60 MHz, 8-bit interface
- Connects to a standard Charge Pump for VBUS 5 V generation
- RNDIS mode for accelerating RNDIS type protocols using short packet termination over USB

### *6.15.1 USBPHY_CTL Register Description*

The USB physical interface control register USBPHY_CTL is described in Figure 6-58 and Table 6-70.

**Figure 6-58. USBPHY_CTL Register**

| 31 | | | | | | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | PHYCLKGD |
| R-0000 0000 0000 0000 0000 000 | | | | | | | R-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SESNDEN | VBDTCTEN | Reserved | PHYPLLON | CLKO1SEL | OSCPDWN | Reserved | PHYPDWN |
| R/W-1 | R/W-1 | R-0 | R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-1 |

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

**Table 6-70. USBPHY_CTL Register Descriptions**

| Name | Description |
|---|---|
| PHYCLKGD | USB PHY Power and Clock Good<br>0 = Phy power not ramped or PLL not locked<br>1 = Phy power is good and PLL is locked |
| SESNDEN | Session End Comparator enable<br>0 = comparator disabled<br>1 = comparator enabled |
| VBDTCTEN | vbus comparator enable<br>0 = comparators (except session end) disabled<br>1 = comparators (except session end) enabled |
| PHYPLLON | USB PHY PLL suspend override<br>0 = Normal PLL operation<br>1 = Override PLL suspend state |
| CLKO1SEL | CLK_OUT1 frequency select<br>0 = 24 MHz<br>1 = 12 MHz |
| OSCPDWN | USB PHY oscillator power down control<br>0 = PHY oscillator powered<br>1 = PHY oscillator power off |
| PHYPDWN | USB PHY power down control<br>0 = PHY powered<br>1 = PHY power off |

### 6.15.2 USB2.0 Peripheral Register Description(s)

The USB register memory mapping is shown in Table 6-71.

**Table 6-71. USB 2.0 Register Descriptions**

| Address | Acronym | Register Description |
|---|---|---|
| 0x01C6 4000 | REVR | Revision Register |
| 0x01C6 4004 | CTRLR | Control Register |
| 0x01C6 4008 | STATR | Status Register |
| 0x01C6 4010 | RNDISR | RNDIS Register |
| 0x01C6 4014 | AUTOREQ | Auto Request Register |
| 0x01C6 4020 | INTSRCR | USB Interrupt Source Register |
| 0x01C6 4024 | INTSETR | USB Interrupt Source Set Register |
| 0x01C6 4028 | INTCLRR | USB Interrupt Source Clear Register |
| 0x01C6 402C | INTMSKR | USB Interrupt Mask Register |
| 0x01C6 4030 | INTMSKSETR | USB Interrupt Mask Set Register |
| 0x01C6 4034 | INTMSKCLRR | USB Interrupt Mask Clear Register |
| 0x01C6 4038 | INTMASKEDR | USB Interrupt Source Masked Register |
| 0x01C6 403C | EOIR | USB End of Interrupt Register |
| 0x01C6 4040 | INTVECTR | USB Interrupt Vector Register |
| 0x01C6 4080 | TCPPICR | TX CPPI Control Register |
| 0x01C6 4084 | TCPPITDR | TX CPPI Teardown Register |
| 0x01C6 4088 | TCPPIEOIR | TX CPPI DMA Controller End of Interrupt Register |
| 0x01C6 408C | TCPPIIVECTR | TX CPPI DMA Controller Interrupt Vector Register |
| 0x01C6 4090 | TCPPIMSKSR | TX CPPI Masked Status Register |
| 0x01C6 4094 | TCPPIRAWSR | TX CPPI Raw Status Register |
| 0x01C6 4098 | TCPPIIENSETR | TX CPPI Interrupt Enable Set Register |
| 0x01C6 409C | TCPPIIENCLRR | TX CPPI Interrupt Enable Clear Register |
| 0x01C6 40C0 | RCPPICR | RX CPPI Control Register |
| 0x01C6 40D0 | RCPPIMSKSR | RX CPPI Masked Status Register |
| 0x01C6 40D4 | RCPPIRAWSR | RX CPPI Raw Status Register |
| 0x01C6 40D8 | RCPPIENSETR | RX CPPI Interrupt Enable Set Register |
| 0x01C6 40DC | RCPPIIENCLRR | RX CPPI Interrupt Enable Clear Register |
| 0x01C6 40E0 | RBUFCNT0 | RX Buffer Count 0 Register |
| 0x01C6 40E4 | RBUFCNT1 | RX Buffer Count 1 Register |
| 0x01C6 40E8 | RBUFCNT2 | RX Buffer Count 2 Register |
| 0x01C6 40EC | RBUFCNT3 | RX Buffer Count 3 Register |
| **TX/RX CCPI Channel 0 State Block** | | |
| 0x01C6 4100 | TCPPIDMASTATEW0 | TX CPPI DMA State Word 0 |
| 0x01C6 4104 | TCPPIDMASTATEW1 | TX CPPI DMA State Word 1 |
| 0x01C6 4108 | TCPPIDMASTATEW2 | TX CPPI DMA State Word 2 |
| 0x01C6 410C | TCPPIDMASTATEW3 | TX CPPI DMA State Word 3 |
| 0x01C6 4110 | TCPPIDMASTATEW4 | TX CPPI DMA State Word 4 |
| 0x01C6 4114 | TCPPIDMASTATEW5 | TX CPPI DMA State Word 5 |
| 0x01C6 4118 | TCPPIDMASTATEW6 | TX CPPI DMA State Word 6 |
| 0x01C6 411C | TCPPICOMPPTR | TX CPPI Completion Pointer |
| 0x01C6 4120 | RCPPIDMASTATEW0 | RX CPPI DMA State Word 0 |
| 0x01C6 4124 | RCPPIDMASTATEW1 | RX CPPI DMA State Word 1 |
| 0x01C6 4128 | RCPPIDMASTATEW2 | RX CPPI DMA State Word 2 |
| 0x01C6 412C | RCPPIDMASTATEW3 | RX CPPI DMA State Word 3 |

**Table 6-71. USB 2.0 Register Descriptions (continued)**

| Address | Acronym | Register Description |
|---|---|---|
| 0x01C6 4130 | RCPPIDMASTATEW4 | RX CPPI DMA State Word 4 |
| 0x01C6 4134 | RCPPIDMASTATEW5 | RX CPPI DMA State Word 5 |
| 0x01C6 4138 | RCPPIDMASTATEW6 | RX CPPI DMA State Word 6 |
| 0x01C6 413C | RCPPICOMPPTR | RX CPPI Completion Pointer |
| **TX/RX CCPI Channel 1 State Block** | | |
| 0x01C6 4140 | TCPPIDMASTATEW0 | TX CPPI DMA State Word 0 |
| 0x01C6 4144 | TCPPIDMASTATEW1 | TX CPPI DMA State Word 1 |
| 0x01C6 4148 | TCPPIDMASTATEW2 | TX CPPI DMA State Word 2 |
| 0x01C6 414C | TCPPIDMASTATEW3 | TX CPPI DMA State Word 3 |
| 0x01C6 4150 | TCPPIDMASTATEW4 | TX CPPI DMA State Word 4 |
| 0x01C6 4154 | TCPPIDMASTATEW5 | TX CPPI DMA State Word 5 |
| 0x01C6 4158 | TCPPIDMASTATEW6 | TX CPPI DMA State Word 6 |
| 0x01C6 415C | TCPPICOMPPTR | TX CPPI Completion Pointer |
| 0x01C6 4160 | RCPPIDMASTATEW0 | RX CPPI DMA State Word 0 |
| 0x01C6 4164 | RCPPIDMASTATEW1 | RX CPPI DMA State Word 1 |
| 0x01C6 4168 | RCPPIDMASTATEW2 | RX CPPI DMA State Word 2 |
| 0x01C6 416C | RCPPIDMASTATEW3 | RX CPPI DMA State Word 3 |
| 0x01C6 4170 | RCPPIDMASTATEW4 | RX CPPI DMA State Word 4 |
| 0x01C6 4174 | RCPPIDMASTATEW5 | RX CPPI DMA State Word 5 |
| 0x01C6 4178 | RCPPIDMASTATEW6 | RX CPPI DMA State Word 6 |
| 0x01C6 417C | RCPPICOMPPTR | RX CPPI Completion Pointer |
| **TX/RX CCPI Channel 2 State Block** | | |
| 0x01C6 4180 | TCPPIDMASTATEW0 | TX CPPI DMA State Word 0 |
| 0x01C6 4184 | TCPPIDMASTATEW1 | TX CPPI DMA State Word 1 |
| 0x01C6 4188 | TCPPIDMASTATEW2 | TX CPPI DMA State Word 2 |
| 0x01C6 418C | TCPPIDMASTATEW3 | TX CPPI DMA State Word 3 |
| 0x01C6 4190 | TCPPIDMASTATEW4 | TX CPPI DMA State Word 4 |
| 0x01C6 4194 | TCPPIDMASTATEW5 | TX CPPI DMA State Word 5 |
| 0x01C6 4198 | TCPPIDMASTATEW6 | TX CPPI DMA State Word 6 |
| 0x01C6 419C | TCPPICOMPPTR | TX CPPI Completion Pointer |
| 0x01C6 41A0 | RCPPIDMASTATEW0 | RX CPPI DMA State Word 0 |
| 0x01C6 41A4 | RCPPIDMASTATEW1 | RX CPPI DMA State Word 1 |
| 0x01C6 41A8 | RCPPIDMASTATEW2 | RX CPPI DMA State Word 2 |
| 0x01C6 41AC | RCPPIDMASTATEW3 | RX CPPI DMA State Word 3 |
| 0x01C6 41BA | RCPPIDMASTATEW4 | RX CPPI DMA State Word 4 |
| 0x01C6 41B4 | RCPPIDMASTATEW5 | RX CPPI DMA State Word 5 |
| 0x01C6 41B8 | RCPPIDMASTATEW6 | RX CPPI DMA State Word 6 |
| 0x01C6 41BC | RCPPICOMPPTR | RX CPPI Completion Pointer |
| **TX/RX CCPI Channel 3 State Block** | | |
| 0x01C6 41C0 | TCPPIDMASTATEW0 | TX CPPI DMA State Word 0 |
| 0x01C6 41C4 | TCPPIDMASTATEW1 | TX CPPI DMA State Word 1 |
| 0x01C6 41C8 | TCPPIDMASTATEW2 | TX CPPI DMA State Word 2 |
| 0x01C6 41CC | TCPPIDMASTATEW3 | TX CPPI DMA State Word 3 |
| 0x01C6 41D0 | TCPPIDMASTATEW4 | TX CPPI DMA State Word 4 |
| 0x01C6 41D4 | TCPPIDMASTATEW5 | TX CPPI DMA State Word 5 |
| 0x01C6 41D8 | TCPPIDMASTATEW6 | TX CPPI DMA State Word 6 |
| 0x01C6 41DC | TCPPICOMPPTR | TX CPPI Completion Pointer |

**Table 6-71. USB 2.0 Register Descriptions (continued)**

| Address | Acronym | Register Description |
|---|---|---|
| 0x01C6 41E0 | RCPPIDMASTATEW0 | RX CPPI DMA State Word 0 |
| 0x01C6 41E4 | RCPPIDMASTATEW1 | RX CPPI DMA State Word 1 |
| 0x01C6 41E8 | RCPPIDMASTATEW2 | RX CPPI DMA State Word 2 |
| 0x01C6 41EC | RCPPIDMASTATEW3 | RX CPPI DMA State Word 3 |
| 0x01C6 41F0 | RCPPIDMASTATEW4 | RX CPPI DMA State Word 4 |
| 0x01C6 41F4 | RCPPIDMASTATEW5 | RX CPPI DMA State Word 5 |
| 0x01C6 41F8 | RCPPIDMASTATEW6 | RX CPPI DMA State Word 6 |
| 0x01C6 41FC | RCPPICOMPPTR | RX CPPI Completion Pointer |
| **Core Registers** | | |
| 0x01C6 4400 | FADDR | Function Address Register |
| 0x01C6 4401 | POWER | Power Management Register |
| 0x01C6 4402 | INTRTX | Interrupt Register for Endpoint 0 plus TX Endpoints 1 to 4 |
| 0x01C6 4404 | INTRRX | Interrupt Register for RX Endpoints 1 to 4 |
| 0x01C6 4406 | INTRTXE | Interrupt Enable Register for INTRTX |
| 0x01C6 4408 | INTRRXE | Interrupt Enable Register for INTRRX |
| 0x01C6 440A | INTRUSB | Interrupt Register for Common USB Interrupts |
| 0x01C6 440B | INTRUSBE | Interrupt Enable Register for INTRUSB |
| 0x01C6 440C | FRAME | Frame Number Register |
| 0x01C6 440E | INDEX | Index register for selecting the endpoint status and control registers |
| 0x01C6 440F | TESTMODE | Register to enable the USB 2.0 test modes |
| 0x01C6 4410 | TXMAXP | Maximum packet size for peripheral/host TX endpoint (Index register set to select Endpoints 1 - 4 only) |
| 0x01C6 4412 | PERI_CSR0 | Control Status register for Endpoint 0 in Peripheral mode. (Index register set to select Endpoint 0) |
| | HOST_CSR0 | Control Status register for Endpoint 0 in Host mode. (Index register set to select Endpoint 0) |
| | PERI_TXCSR | Control Status register for peripheral TX endpoint. (Index register set to select Endpoints 1 - 4) |
| | HOST_TXCSR | Control Status register for host TX endpoint. (Index register set to select Endpoints 1 - 4) |
| 0x01C6 4414 | RXMAXP | Maximum packet size for peripheral/host RX endpoint (Index register set to select Endpoints 1 - 4 only) |
| 0x01C6 4416 | PERI_RXCSR | Control Status register for peripheral RX endpoint. (Index register set to select Endpoints 1 - 4) |
| | HOST_RXCSR | Control Status register for host RX endpoint. (Index register set to select Endpoints 1 - 4) |
| 0x01C6 4418 | COUNT0 | Number of received bytes in Endpoint 0 FIFO. (Index register set to select Endpoint 0) |
| | RXCOUNT | Number of bytes in host RX endpoint FIFO. (Index register set to select Endpoints 1 - 4) |
| 0x01C6 441A | HOST_TYPE0 | Defines the speed of Endpoint 0 |
| 0x01C6 441A | HOST_TXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host TX endpoint. (Index register set to select Endpoints 1 - 4 only) |
| 0x01C6 441B | HOST_NAKLIMIT0 | Sets the NAK response timeout on Endpoint 0. (Index register set to select Endpoint 0) |
| 0x01C6 441B | HOST_TXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host TX endpoint. (Index register set to select Endpoints 1 - 4 only) |
| 0x01C6 441C | HOST_RXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host RX endpoint. (Index register set to select Endpoints 1 - 4 only) |
| 0x01C6 441D | HOST_RXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host RX endpoint. (Index register set to select Endpoints 1 - 4 only) |

**Table 6-71. USB 2.0 Register Descriptions (continued)**

| Address | Acronym | Register Description |
|---|---|---|
| 0x01C6 441F | CONFIGDATA | Returns details of core configuration (Index register set to select Endpoint 0) |
| 0x01C6 4420 | FIFO0 | TX and RX FIFO Register for Endpoint 0 |
| 0x01C6 4424 | FIFO1 | TX and RX FIFO Register for Endpoint 1 |
| 0x01C6 4428 | FIFO2 | TX and RX FIFO Register for Endpoint 2 |
| 0x01C6 442C | FIFO3 | TX and RX FIFO Register for Endpoint 3 |
| 0x01C6 4430 | FIFO4 | TX and RX FIFO Register for Endpoint 4 |
| 0x01C6 4462 | TXFIFOSZ | TX Endpoint FIFO Size (Index register set to select Endpoints 0 - 4 only) |
| 0x01C6 4463 | RXFIFOSZ | RX Endpoint FIFO Size (Index register set to select Endpoints 0 - 4 only) |
| 0x01C6 4464 | TXFIFOADDR | TX Endpoint FIFO Address (Index register set to select Endpoints 0 - 4 only) |
| 0x01C6 4466 | RXFIFOADDR | RX Endpoint FIFO Address (Index register set to select Endpoints 0 - 4 only) |
| **Target Endpoint Control Registers (Valid Only in Host Mode) - EPTRG0** | | |
| 0x01C6 4480 | TXFUNCADDR | Address of the target function that has to be accessed through the associated TX Endpoint |
| 0x01C6 4482 | TXHUBADDR | Address of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 4483 | TXHUBPORT | Port of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 4484 | RXFUNCADDR | Address of the target function that has to be accessed through the associated RX Endpoint |
| 0x01C6 4486 | RXHUBADDR | Address of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 4487 | RXHUBPORT | Port of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| **Target Endpoint Control Registers (Valid Only in Host Mode) - EPTRG1** | | |
| 0x01C6 4488 | TXFUNCADDR | Address of the target function that has to be accessed through the associated TX Endpoint |
| 0x01C6 448A | TXHUBADDR | Address of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 448B | TXHUBPORT | Port of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 448C | RXFUNCADDR | Address of the target function that has to be accessed through the associated RX Endpoint |
| 0x01C6 448E | RXHUBADDR | Address of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 448F | RXHUBPORT | Port of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| **Target Endpoint Control Registers (Valid Only in Host Mode) - EPTRG2** | | |
| 0x01C6 4490 | TXFUNCADDR | Address of the target function that has to be accessed through the associated TX Endpoint |
| 0x01C6 4492 | TXHUBADDR | Address of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 4493 | TXHUBPORT | Port of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 4494 | RXFUNCADDR | Address of the target function that has to be accessed through the associated RX Endpoint |

**Table 6-71. USB 2.0 Register Descriptions (continued)**

| Address | Acronym | Register Description |
|---|---|---|
| 0x01C6 4496 | RXHUBADDR | Address of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 4497 | RXHUBPORT | Port of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| **Target Endpoint Control Registers (Valid Only in Host Mode) - EPTRG3** | | |
| 0x01C6 4498 | TXFUNCADDR | Address of the target function that has to be accessed through the associated TX Endpoint |
| 0x01C6 449A | TXHUBADDR | Address of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 449B | TXHUBPORT | Port of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 449C | RXFUNCADDR | Address of the target function that has to be accessed through the associated RX Endpoint |
| 0x01C6 449E | RXHUBADDR | Address of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 449F | RXHUBPORT | Port of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| **Target Endpoint Control Registers (Valid Only in Host Mode) - EPTRG4** | | |
| 0x01C6 44A0 | TXFUNCADDR | Address of the target function that has to be accessed through the associated TX Endpoint |
| 0x01C6 44A2 | TXHUBADDR | Address of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 44A3 | TXHUBPORT | Port of the hub that has to be accessed through the associated TX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 44A4 | RXFUNCADDR | Address of the target function that has to be accessed through the associated RX Endpoint |
| 0x01C6 44A6 | RXHUBADDR | Address of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| 0x01C6 44A7 | RXHUBPORT | Port of the hub that has to be accessed through the associated RX Endpoint. This is used only when full speed or low speed device is connected via a USB2.0 high speed hub |
| **Control and Status Register for Endpoint 0 - EOCSR0** | | |
| 0x01C6 4502 | PERI_CSR0 | Control Status Register for Endpoint 0 in Peripheral mode |
| | HOST_CSR0 | Control Status Register for Endpoint 0 in Host mode |
| 0x01C6 4508 | COUNT0 | Number of Received Bytes in Endpoint 0 FIFO |
| 0x01C6 450A | HOST_TYPE0 | Defines the Speed of Endpoint 0 |
| 0x01C6 450B | HOST_NAKLIMIT0 | Sets the NAK response timeout on Endpoint 0. |
| 0x01C6 450F | CONFIGDATA | Returns details of core configuration |
| **Control and Status Register for Endpoint 1 - EOCSR1** | | |
| 0x01C6 4510 | TXMAXP | Maximum Packet size for Peripheral/Host TX Endpoint |
| 0x01C6 4512 | PERI_TXCSR | Control Status Register for Peripheral TX Endpoint |
| | HOST_TXCSR | Control Status Register for Host TX Endpoint |
| 0x01C6 4514 | RXMAXP | Maximum Packet Size for Peripheral/Host RX Endpoint |
| 0x01C6 4516 | PERI_RXCSR | Control Status Register for Peripheral RX Endpoint |
| | HOST_RXCSR | Control Status Register for Host RX Endpoint |
| 0x01C6 4518 | RXCOUNT | Number of Bytes in Host RX Endpoint FIFO |

**Table 6-71. USB 2.0 Register Descriptions (continued)**

| Address | Acronym | Register Description |
|---|---|---|
| 0x01C6 451A | HOST_TXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host TX endpoint. |
| 0x01C6 451B | HOST_TXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host TX endpoint. |
| 0x01C6 451C | HOST_RXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host RX endpoint. |
| 0x01C6 451D | HOST_RXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host RX endpoint. |
| **Control and Status Register for Endpoint 2 - EOCSR2** | | |
| 0x01C6 4520 | TXMAXP | Maximum Packet Size for Peripheral/Host TX Endpoint |
| 0x01C6 4522 | PERI_TXCSR | Control Status Register for Peripheral TX Endpoint |
| | HOST_TXCSR | Control Status Register for Host TX Endpoint |
| 0x01C6 4524 | RXMAXP | Maximum Packet Size for Peripheral/Host RX Endpoint |
| 0x01C6 4526 | PERI_RXCSR | Control Status Register for Peripheral RX Endpoint |
| | HOST_RXCSR | Control Status Register for Host RX Endpoint |
| 0x01C6 4528 | RXCOUNT | Number of Bytes in Host RX Endpoint FIFO |
| 0x01C6 452A | HOST_TXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host TX endpoint. |
| 0x01C6 452B | HOST_TXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host TX endpoint. |
| 0x01C6 452C | HOST_RXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host RX endpoint. |
| 0x01C6 452D | HOST_RXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host RX endpoint. |
| **Control and Status Register for Endpoint 3 - EOCSR3** | | |
| 0x01C6 4530 | TXMAXP | Maximum Packet Size for Peripheral/Host TX Endpoint |
| 0x01C6 4532 | PERI_TXCSR | Control Status Register for Peripheral TX Endpoint |
| | HOST_TXCSR | Control Status Register for Host TX Endpoint |
| 0x01C6 4534 | RXMAXP | Maximum Packet Size for Peripheral/Host RX Endpoint |
| 0x01C6 4536 | PERI_RXCSR | Control Status Register for Peripheral RX Endpoint |
| | HOST_RXCSR | Control Status Register for Host RX Endpoint |
| 0x01C6 4538 | RXCOUNT | Number of Bytes in Host RX Endpoint FIFO |
| 0x01C6 453A | HOST_TXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host TX endpoint. |
| 0x01C6 453B | HOST_TXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host TX endpoint. |
| 0x01C6 453C | HOST_RXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host RX endpoint. |
| 0x01C6 453D | HOST_RXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host RX endpoint. |
| **Control and Status Register for Endpoint 4 - EOCSR4** | | |
| 0x01C6 4540 | TXMAXP | Maximum Packet Size for Peripheral/Host TX Endpoint |
| 0x01C6 4542 | PERI_TXCSR | Control Status Register for Peripheral TX Endpoint |
| | HOST_TXCSR | Control Status Register for Host TX Endpoint |
| 0x01C6 4544 | RXMAXP | Maximum Packet Size for Peripheral/Host RX Endpoint |
| 0x01C6 4546 | PERI_RXCSR | Control Status Register for Peripheral RX Endpoint |
| | HOST_RXCSR | Control Status Register for Host RX Endpoint |
| 0x01C6 4548 | RXCOUNT | Number of Bytes in Host RX Endpoint FIFO |
| 0x01C6 454A | HOST_TXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host TX endpoint. |
| 0x01C6 454B | HOST_TXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host TX endpoint. |

**Table 6-71. USB 2.0 Register Descriptions (continued)**

| Address | Acronym | Register Description |
|---|---|---|
| 0x01C6 454C | HOST_RXTYPE | Sets the operating speed, transaction protocol and peripheral endpoint number for the host RX endpoint. |
| 0x01C6 454D | HOST_RXINTERVAL | Sets the polling interval for Interrupt/ISOC transactions or the NAK response timeout on Bulk transactions for host RX endpoint. |

### 6.15.3 USB2.0 Electrical Data/Timing

**Table 6-72. Switching Characteristics Over Recommended Operating Conditions for USB2.0 (see Figure 6-59)**

| NO. | PARAMETER | | A-513, -594, -810 | | | | | | UNIT |
|---|---|---|---|---|---|---|---|---|---|
| | | | LOW SPEED 1.5 Mbps | | FULL SPEED 12 Mbps | | HIGH SPEED 480 Mbps | | |
| | | | MIN | MAX | MIN | MAX | MIN | MAX | |
| 1 | $t_{r(D)}$ | Rise time, USB_DP and USB_DM signals(1) | 75 | 300 | 4 | 20 | 0.5 | | ns |
| 2 | $t_{f(D)}$ | Fall time, USB_DP and USB_DM signals(1) | 75 | 300 | 4 | 20 | 0.5 | | ns |
| 3 | $t_{rfM}$ | Rise/Fall time, matching(2) | 80 | 125 | 90 | 111.11 | – | – | % |
| 4 | $V_{CRS}$ | Output signal cross-over voltage(1) | 1.3 | 2 | 1.3 | 2 | – | – | V |
| 5 | $t_{jr(source)NT}$ | Source (Host) Driver jitter, next transition | | 2 | | 2 | | (3) | ns |
| | $t_{jr(FUNC)NT}$ | Function Driver jitter, next transition | | 25 | | 2 | | (3) | ns |
| 6 | $t_{jr(source)PT}$ | Source (Host) Driver jitter, paired transition(4) | | 1 | | 1 | | (3) | ns |
| | $t_{jr(FUNC)PT}$ | Function Driver jitter, paired transition | | 10 | | 1 | | (3) | ns |
| 7 | $t_{w(EOPT)}$ | Pulse duration, EOP transmitter | 1250 | 1500 | 160 | 175 | – | – | ns |
| 8 | $t_{w(EOPR)}$ | Pulse duration, EOP receiver(5) | 670 | | 82 | | – | | ns |
| 9 | $t_{(DRATE)}$ | Data Rate | | 1.5 | | 12 | | 480 | Mb/s |
| 10 | $Z_{DRV}$ | Driver Output Resistance | – | – | 28 | 49.5 | 40.5 | 49.5 | Ω |
| 11 | USB_R1 | USB reference resistor | 9.9 | 10.1 | 9.9 | 10.1 | 9.9 | 10.1 | kΩ |

(1) Low Speed: $C_L$ = 200 pF, Full Speed: $C_L$ = 50 pF, High Speed: $C_L$ = 50 pF
(2) $t_{RFM} = (t_r/t_f) \times 100$. [Excluding the first transaction from the Idle state.]
(3) For more detailed information, see the Universal Serial Bus Specification Revision 2.0, Chapter 7, Electrical.
(4) $t_{jr} = t_{px(1)} - t_{px(0)}$
(5) Must accept as valid EOP

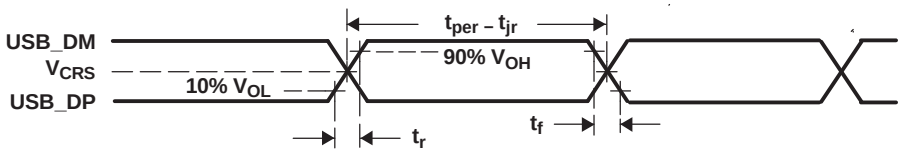
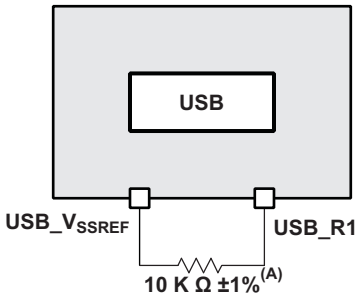



**Figure 6-59. USB2.0 Integrated Transceiver Interface Timing**




A. Place the 10 K Ω ± 1% as close to the device as possible.

**Figure 6-60. USB Reference Resistor Routing**

## 6.16 Universal Asynchronous Receiver/Transmitter (UART)

DM6446 has 3 UART peripherals. Each UART has the following features:

- 16-byte storage space for both the transmitter and receiver FIFOs
- 1, 4, 8, or 14 byte selectable receiver FIFO trigger level for autoflow control and DMA
- DMA signaling capability for both received and transmitted data
- Programmable auto-rts and auto-cts for autoflow control
- Frequency pre-scale values from 1 to 65,535 to generate appropriate baud rates
- Prioritized interrupts
- Programmable serial data formats
  - 5, 6, 7, or 8-bit characters
  - Even, odd, or no parity bit generation and detection
  - 1, 1.5, or 2 stop bit generation
- False start bit detection
- Line break generation and detection
- Internal diagnostic capabilities
  - Loopback controls for communications link fault isolation
  - Break, parity, overrun, and framing error simulation
- Modem control functions (CTS, RTS) on ***UART2 only***.

The UART0/1/2 registers are listed in Table 6-73, Table 6-74, and Table 6-75.

### *6.16.1 UART Peripheral Register Description(s)*

**Table 6-73. UART0 Register Descriptions**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C2 0000 | RBR | UART0 Receiver Buffer Register (Read Only) |
| 0x01C2 0000 | THR | UART0 Transmitter Holding Register (Write Only) |
| 0x01C2 0004 | IER | UART0 Interrupt Enable Register |
| 0x01C2 0008 | IIR | UART0 Interrupt Identification Register (Read Only) |
| 0x01C2 0008 | FCR | UART0 FIFO Control Register (Write Only) |
| 0x01C2 000C | LCR | UART0 Line Control Register |
| 0x01C2 0010 | MCR | UART0 Modem Control Register |
| 0x01C2 0014 | LSR | UART0 Line Status Register |
| 0x01C2 0018 | - | Reserved |
| 0x01C2 001C | - | Reserved |
| 0x01C2 0020 | DLL | UART0 Divisor Latch (LSB) |
| 0x01C2 0024 | DLH | UART0 Divisor Latch (MSB) |
| 0x01C2 0028 | PID1 | Peripheral Identification Register 1 |
| 0x01C2 002C | PID2 | Peripheral Identification Register 2 |
| 0x01C2 0030 | PWREMU_MGMT | UART0 Power and Emulation Management Register |
| 0x01C2 0034 - 0x01C2 03FF | - | Reserved |

**Table 6-74. UART1 Register Descriptions**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C2 0400 | RBR | UART1 Receiver Buffer Register (Read Only) |
| 0x01C2 0400 | THR | UART1 Transmitter Holding Register (Write Only) |
| 0x01C2 0404 | IER | UART1 Interrupt Enable Register |
| 0x01C2 0408 | IIR | UART1 Interrupt Identification Register (Read Only) |
| 0x01C2 0408 | FCR | UART1 FIFO Control Register (Write Only) |
| 0x01C2 040C | LCR | UART1 Line Control Register |
| 0x01C2 0410 | MCR | UART1 Modem Control Register |
| 0x01C2 0414 | LSR | UART1 Line Status Register |
| 0x01C2 0418 | - | Reserved |
| 0x01C2 041C | - | Reserved |
| 0x01C2 0420 | DLL | UART1 Divisor Latch (LSB) |
| 0x01C2 0424 | DLH | UART1 Divisor Latch (MSB) |
| 0x01C2 0428 | PID1 | Peripheral Identification Register 1 |
| 0x01C2 042C | PID2 | Peripheral Identification Register 2 |
| 0x01C2 0430 | PWREMU_MGMT | UART1 Power and Emulation Management Register |
| 0x01C2 0434 - 0x01C2 07FF | - | Reserved |

**Table 6-75. UART2 Register Descriptions**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C2 0800 | RBR | UART2 Receiver Buffer Register (Read Only) |
| 0x01C2 0800 | THR | UART2 Transmitter Holding Register (Write Only) |
| 0x01C2 0804 | IER | UART2 Interrupt Enable Register |
| 0x01C2 0808 | IIR | UART2 Interrupt Identification Register (Read Only) |
| 0x01C2 0808 | FCR | UART2 FIFO Control Register (Write Only) |
| 0x01C2 080C | LCR | UART2 Line Control Register |
| 0x01C2 0810 | MCR | UART2 Modem Control Register |
| 0x01C2 0814 | LSR | UART2 Line Status Register |
| 0x01C2 0818 | - | Reserved |
| 0x01C2 081C | - | Reserved |
| 0x01C2 0820 | DLL | UART2 Divisor Latch (LSB) |
| 0x01C2 0824 | DLH | UART2 Divisor Latch (MSB) |
| 0x01C2 0828 | PID1 | Peripheral Identification Register 1 |
| 0x01C2 082C | PID2 | Peripheral Identification Register 2 |
| 0x01C2 0830 | PWREMU_MGMT | UART2 Power and Emulation Management Register |
| 0x01C2 0834 - 0x01C2 0BFF | - | Reserved |

### 6.16.2 UART Electrical Data/Timing

**Table 6-76. Timing Requirements for UARTx Receive[1] (see Figure 6-61)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 4 | $t_{w(URXDB)}$ | Pulse duration, receive data bit (RXDn) [15/30/100 pF] | 0.96U | 1.05U | ns |
| 5 | $t_{w(URXSB)}$ | Pulse duration, receive start bit [15/30/100 pF] | 0.96U | 1.05U | ns |

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

**Table 6-77. Switching Characteristics Over Recommended Operating Conditions for UARTx Transmit[1] (see Figure 6-61)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $f_{(baud)}$ | Maximum programmable baud rate | | 128 | kHz |
| 2 | $t_{w(UTXDB)}$ | Pulse duration, transmit data bit (TXDn) [15/30/100 pF] | U - 2 | U + 2 | ns |
| 3 | $t_{w(UTXSB)}$ | Pulse duration, transmit start bit [15/30/100 pF] | U - 2 | U + 2 | ns |

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

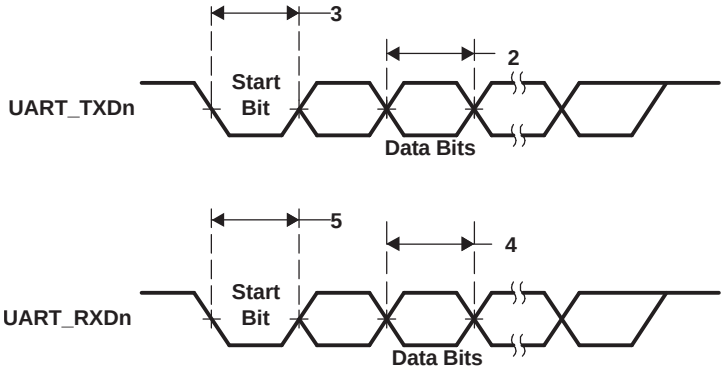



**Figure 6-61. UART Transmit/Receive Timing**

## 6.17 Serial Peripheral Interface (SPI)

The DM6446 SPI peripheral provides a programmable length shift register which allows serial communication with other SPI devices through a 3 or 4 wire interface. The SPI supports the following features:

- Master mode operation
- 2 chip selects for interfacing to multiple slave SPI devices.
- 3 or 4 wire interface

The SPI registers are shown in Table 6-78.

### *6.17.1 SPI Peripheral Register Description(s)*

**Table 6-78. SPI Register Descriptions**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C6 6800 | SPIGCR0 | SPI Global Control Register 0 |
| 0x01C6 6804 | SPIGCR1 | SPI Global Control Register 1 |
| 0x01C6 6808 | SPIINT | SPI Interrupt Register |
| 0x01C6 680C | SPILVL | SPI Interrupt Level Register |
| 0x01C6 6810 | SPIFLG | SPI Flag Status Register |
| 0x01C6 6814 | SPIPC0 | SPI Pin Control Register 0 |
| 0x01C6 6818 | – | Reserved |
| 0x01C6 681C | SPIPC2 | SPI Pin Control Register 2 |
| 0x01C6 6820 - 0x01C6 6838 | – | Reserved |
| 0x01C6 683C | SPIDAT1 | SPI Shift Register 1 |
| 0x01C6 6840 | SPIBUF | SPI Buffer Register |
| 0x01C6 6844 | SPIEMU | SPI Emulation Register |
| 0x01C6 6848 | SPIDELAY | SPI Delay Register |
| 0x01C6 684C | SPIDEF | SPI Default Chip Select Register |
| 0x01C6 6850 | SPIFMT0 | SPI Data Format Register 0 |
| 0x01C6 6854 | SPIFMT1 | SPI Data Format Register 1 |
| 0x01C6 6858 | SPIFMT2 | SPI Data Format Register 2 |
| 0x01C6 685C | SPIFMT3 | SPI Data Format Register 3 |
| 0x01C6 6860 | INTVEC0 | SPI Interrupt Vector Register 0 |
| 0x01C6 6864 | INTVEC1 | SPI Interrupt Vector Register 1 |
| 0x01C6 6868 - 0x01C6 6FFF | | Reserved |

### 6.17.2 SPI Electrical Data/Timing

**Table 6-79. Timing Requirements for SPI (All Modes)(1) (see Figure 6-62)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(CLK)}$ | Cycle time, SPI_CLK | 30.3 | 56888.89 | ns |
| 2 | $t_{w(CLKH)}$ | Pulse duration, SPI_CLK high (All Master Modes) | 0.45*T | 0.55*T | ns |
| 3 | $t_{w(CLKL)}$ | Pulse duration, SPI_CLK low (All Master Modes | 0.45*T | 0.55*T | ns |

(1) T = $t_{c(CLK)}$ [SPI_CLK period is equal to the SPI module clock divided by a configurable divider.]

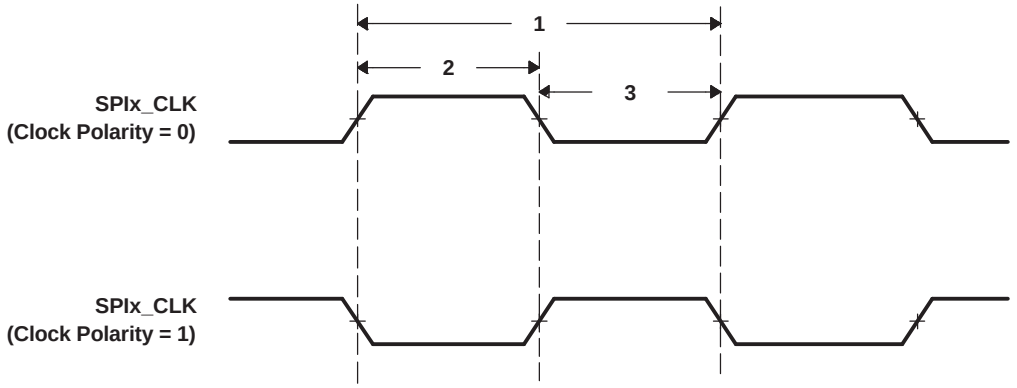



**Figure 6-62. SPI_CLK Timing**

### 6.17.2.1 SPI Master Mode Timings (Clock Phase = 0)

**Table 6-80. Timing Requirements for SPI Master Mode [Clock Phase = 0] (1)(see Figure 6-63)**

| NO. | | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 4 | $t_{su(DIV-CLKL)}$ | Setup time, SPI_DI (input) valid before SPI_CLK (output) falling edge | Clock Polarity = 0 | 0.5P + 9.4 | | ns |
| 5 | $t_{su(DIV-CLKH)}$ | Setup time, SPI_DI (in put) valid before SPI_CLK (output) rising edge | Clock Polarity = 1 | 0.5P + 9.4 | | ns |
| 6 | $t_{h(CLKL-DIV)}$ | Hold time, SPI_DI (input) valid after SPI_CLK (output) falling edge | Clock Polarity = 0 | 0.5P - 4.5 | | ns |
| 7 | $t_{h(CLKH-DIV)}$ | Hold time, SPI_DI (input) valid after SPI_CLK (output) rising edge | Clock Polarity = 1 | 0.5P - 4.5 | | ns |

(1) P = Period of the SPI module clock in nanoseconds (SYSCLK5).

**Table 6-81. Switching Characteristics Over Recommended Operating Conditions for SPI Master Mode [Clock Phase = 0] (see Figure 6-63)**

| NO. | PARAMETER | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 8 | $t_{d(CLKH-DOV)}$ | Delay time, SPI_CLK (output) rising edge to SPI_DO (output) transition | Clock Polarity = 0 | -4 | 5 | ns |
| 9 | $t_{d(CLKL-DOV)}$ | Delay time, SPI_CLK (output) falling edge to SPI_DO (output) transition | Clock Polarity = 1 | -4 | 5 | ns |
| 10 | $t_{d(ENL-CLKH/L)}$ | Delay time, SPI_EN[1:0] (output) falling edge to first SPI_CLK (output) rising or falling edge(1) (2) | | 2P - 2.3 | | ns |
| 11 | $t_{d(CLKH/L-ENH)}$ | Delay time, SPI_CLK (output) rising or falling edge to SPI_EN[1:0] (output) rising edge(1) (2) (3) | | 1P + 0.5C - 0.2 | | ns |

(1) P = Period of the SPI module clock in nanoseconds (SYSCLK5).
(2) This delay can be increased under software control by the C2TDELAY register bit field in the SPIDELAY register.
(3) C = Period of SPI_CLK signal in ns.

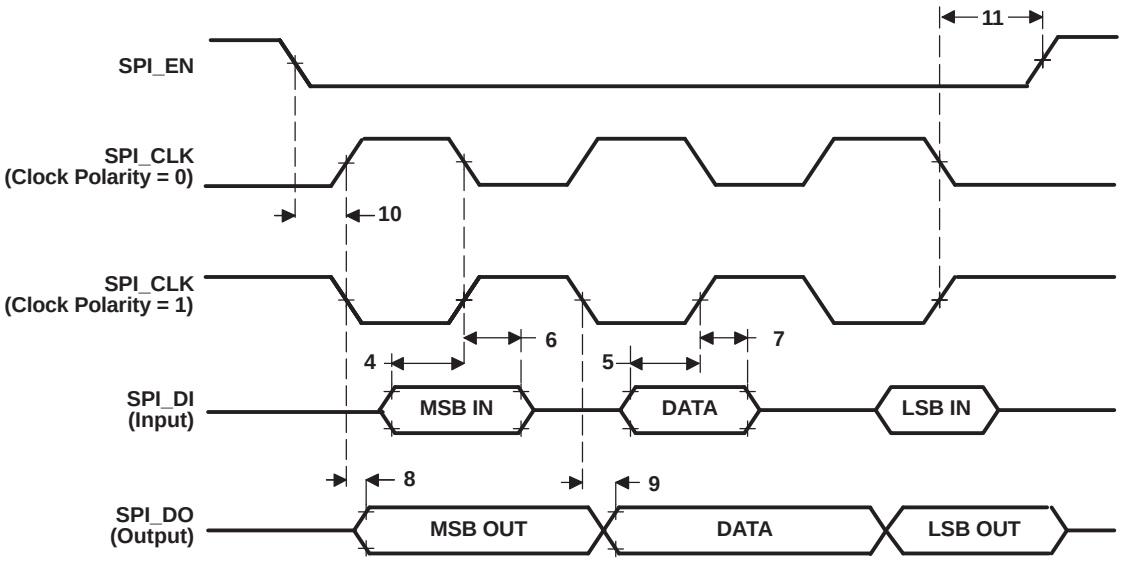



**Figure 6-63. SPI Master Mode External Timing (Clock Phase = 0)**

### 6.17.2.2 SPI Master Mode Timings (Clock Phase = 1)

**Table 6-82. Timing Requirements for SPI Master Mode [Clock Phase = 1](1) (see Figure 6-64)**

| NO. | | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 13 | $t_{su(DIV-CLKL)}$ | Setup time, SPI_DI (input) valid before SPI_CLK (output) rising edge | Clock Polarity = 0 | 0.5P + 9.4 | | ns |
| 14 | $t_{su(DIV-CLKH)}$ | Setup time, SPI_DI (in put) valid before SPI_CLK (output) falling edge | Clock Polarity = 1 | 0.5P + 9.4 | | ns |
| 15 | $t_{h(CLKL-DIV)}$ | Hold time, SPI_DI (input) valid after SPI_CLK (output) rising edge | Clock Polarity = 0 | 0.5P - 4.5 | | ns |
| 16 | $t_{h(CLKH-DIV)}$ | Hold time, SPI_DI (input) valid after SPI_CLK (output) falling edge | Clock Polarity = 1 | 0.5P - 4.5 | | ns |

(1) P = Period of the SPI module clock in nanoseconds (SYSCLK5).

**Table 6-83. Switching Characteristics Over Recommended Operating Conditions for SPI Master Mode [Clock Phase = 1] (see Figure 6-64)**

| NO. | PARAMETER | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 17 | $t_{d(CLKL-DOV)}$ | Delay time, SPI_CLK (output) falling edge to SPI_DO (output) transition | Clock Polarity = 0 | -4 | 5 | ns |
| 18 | $t_{d(CLKH-DOV)}$ | Delay time, SPI_CLK (output) rising edge to SPI_DO (output) transition | Clock Polarity = 1 | -4 | 5 | ns |
| 19 | $t_{d(ENL-CLKH/L)}$ | Delay time, SPI_EN[1:0] (output) falling edge to first SPI_CLK (output) rising or falling edge(1) (2) (3) | | 2P + 0.5C - 2.3 | | ns |
| 20 | $t_{d(CLKH/L-ENH)}$ | Delay time, SPI_CLK (output) rising or falling edge to SPI_EN[1:0] (output) rising edge(1) (2) | | 1P - 0.2 | | ns |

(1) P = Period of the SPI module clock in nanoseconds (SYSCLK5).
(2) This delay can be increased under software control by the C2TDELAY register bit field in the SPIDELAY register.
(3) C = Period of SPI_CLK signal in ns.

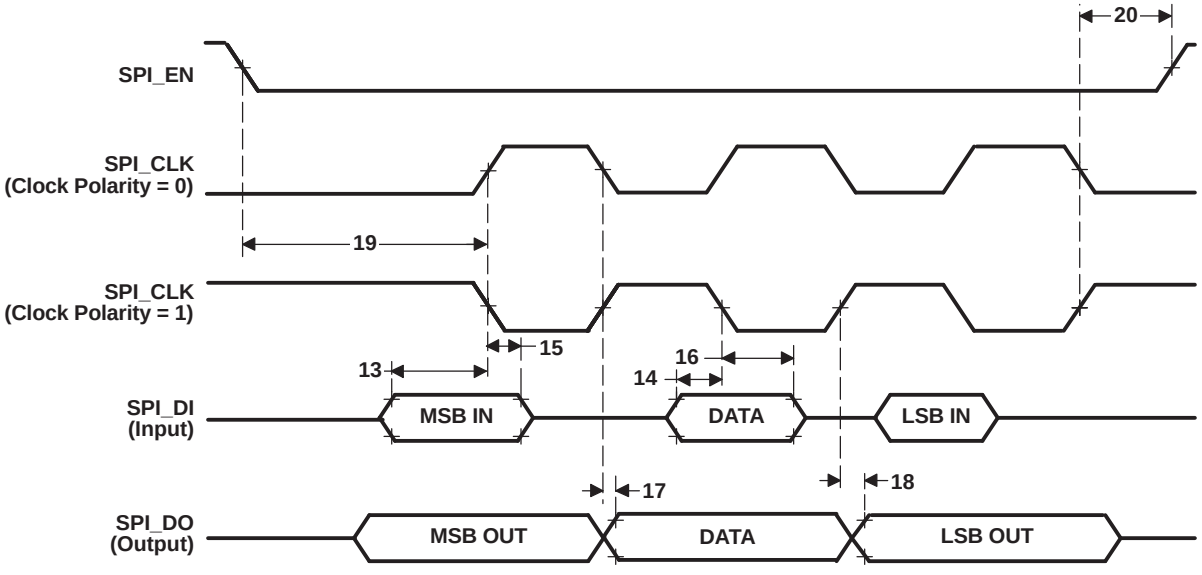



**Figure 6-64. SPI Master Mode External Timing (Clock Phase = 1)**

## 6.18 Inter-Integrated Circuit (I2C)

The inter-integrated circuit (I2C) module provides an interface between DM6446 and other devices compliant with Philips Semiconductors Inter-IC bus ($I^2C$-bus™) specification version 2.1. 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. The I2C port *does not* support CBUS compatible devices.

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

For more detailed information on the I2C peripheral, see the *TMS320DM644x DMSoC Peripherals Overview Reference Guide* (literature number SPRUE19).

**CAUTION**

The DM6446 I2C pins use a standard ±4-mA LVCMOS buffer, not the slow I/O buffer defined in the I2C specification. Series resistors may be necessary to reduce noise at the system level.

### *6.18.1 I2C Peripheral Register Description(s)*

**Table 6-84. I2C Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x1c2 1000 | ICOAR | I2C Own Address Register |
| 0x1c2 1004 | ICIMR | I2C Interrupt Mask Register |
| 0x1c2 1008 | ICSTR | I2C Interrupt Status Register |
| 0x1c2 100C | ICCLKL | I2C Clock Divider Low Register |
| 0x1c2 1010 | ICCLKH | I2C Clock Divider High Register |
| 0x1c2 1014 | ICCNT | I2C Data Count Register |
| 0x1c2 1018 | ICDRR | I2C Data Receive Register |
| 0x1c2 101C | ICSAR | I2C Slave Address Register |
| 0x1c2 1020 | ICDXR | I2C Data Transmit Register |
| 0x1c2 1024 | ICMDR | I2C Mode Register |
| 0x1c2 1028 | ICIVR | I2C Interrupt Vector Register |
| 0x1c2 102C | ICEMDR | I2C Extended Mode Register |
| 0x1c2 1030 | ICPSC | I2C Prescaler Register |
| 0x1c2 1034 | ICPID1 | I2C Peripheral Identification Register 1 |
| 0x1c2 1038 | ICPID2 | I2C Peripheral Identification Register 2 |

## 6.18.2 I2C Electrical Data/Timing

### 6.18.2.1 Inter-Integrated Circuits (I2C) Timing

**Table 6-85. Timing Requirements for I2C Timings(1) (see Figure 6-65)**

| NO. | | | A-513, -594, -810 | | | | 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 | $t_{su(SDAV-SCLH)}$ | Setup time, SDA valid before SCL high | 250 | | 100(2) | | ns |
| 7 | $t_{h(SDA-SCLL)}$ | Hold time, SDA valid after SCL low | 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 I²C-bus™ device can be used in a Standard-mode I²C-bus system, but the requirement $t_{su(SDA-SCLH)} \geq 250$ ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line $t_r$ max + $t_{su(SDA-SCLH)}$= 1000 + 250 = 1250 ns (according to the Standard-mode I²C-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.

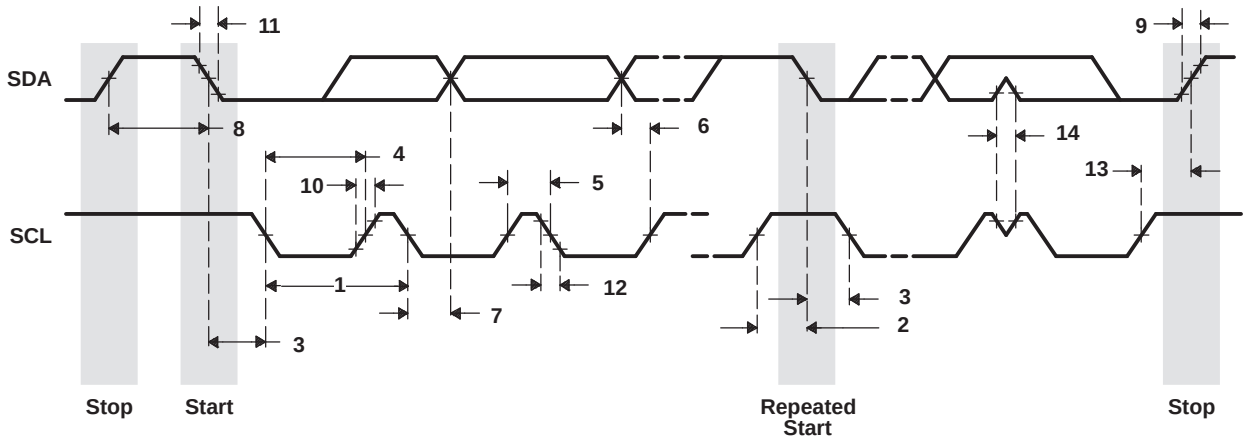



**Figure 6-65. I2C Receive Timings**

**Table 6-86. Switching Characteristics for I2C Timings (see Figure 6-66)**

| NO. | PARAMETER | | A-513, -594, -810 | | | | 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 | $t_{d(SDAV-SCLH)}$ | Delay time, SDA valid to SCL high | 250 | | 100 | | ns |
| 22 | $t_{v(SCLL-SDAV)}$ | Valid time, SDA valid after SCL low | 0 | | 0 | 0.9 | µs |
| 23 | $t_{w(SDAH)}$ | Pulse duration, SDA high between STOP and START conditions | 4.7 | | 1.3 | | µs |
| 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 |

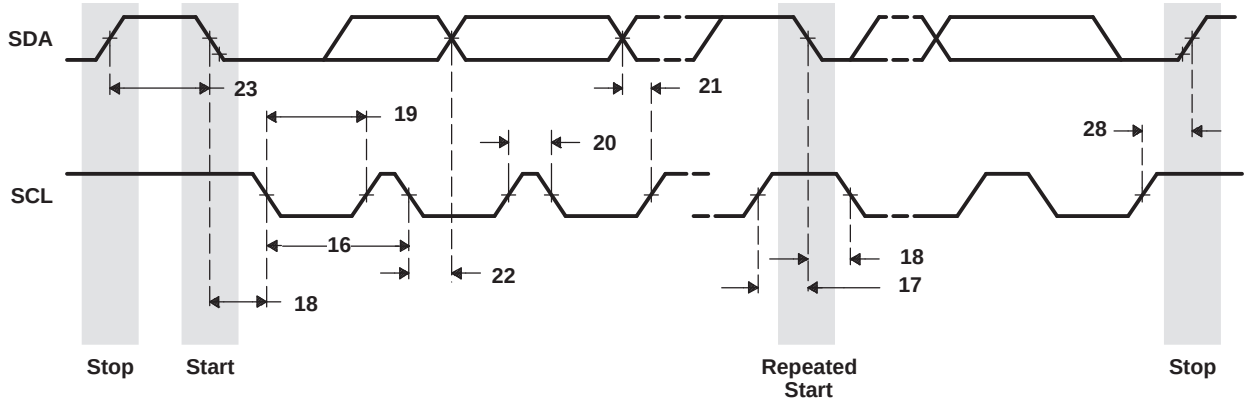



**Figure 6-66. I2C Transmit Timings**

**CAUTION**

The DM6446 I2C pins use a standard ±4-mA LVCMOS buffer, not the slow I/O buffer defined in the I2C specification. Series resistors may be necessary to reduce noise at the system level.

## 6.19 Audio Serial Port (ASP)

The ASP 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 ASP peripheral, see the *TMS320DM644x DMSoC Audio Serial Port (ASP) User's Guide* (literature number SPRUE29).

### *6.19.1 ASP Peripheral Register Description(s)*

**Table 6-87. ASP Register Descriptions**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01E0 2000 | DRR | ASP Data Receive Register |
| 0x01E0 2004 | DXR | ASP Data Transmit Register |
| 0x01E0 2008 | SPCR | ASP Serial Port Control Register |
| 0x01E0 200C | RCR | ASP Receive Control Register |
| 0x01E0 2010 | XCR | ASP Transmit Control Register |
| 0x01E0 2014 | SRGR | ASP Sample Rate Generator Register |
| 0x01E0 2024 | PCR | ASP Pin Control Register |

## 6.19.2 ASP Electrical Data/Timing

### 6.19.2.1 Audio Serial Port (ASP) Timing

**Table 6-88. Timing Requirements for ASP(1) (see Figure 6-67)**

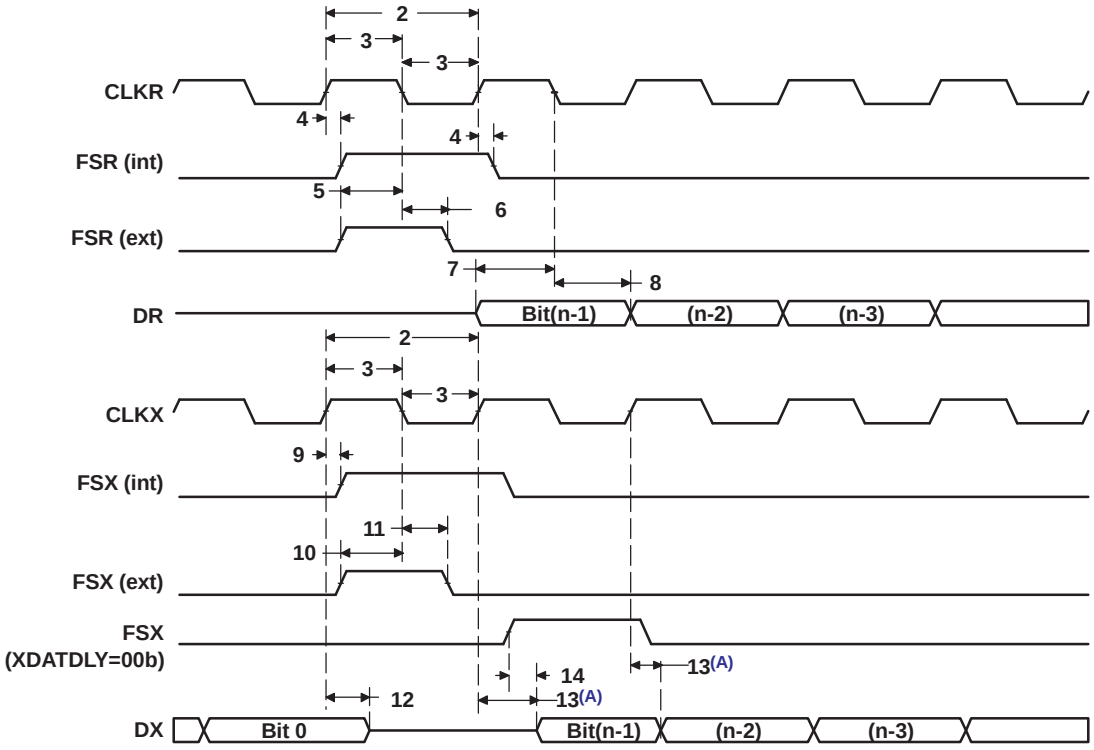
| NO. | | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X ext | 38.5 or 2P(2) (3) | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X ext | 19.25 or P(2) (3) (4) | | ns |
| 5 | $t_{su(FRH-CKRL)}$ | Setup time, external FSR high before CLKR low | CLKR int | 11.8 | | 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 | 10.7 | | 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 | 12.2 | | ns |
| | | | CLKX ext | 1.4 | | |
| 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/SYSCLK5 clock frequency in ns. For example, when running parts at DSP frequency of 594 MHz, use P = 10.1 ns; when running parts at DSP frequency of 810 MHz, use P = 7.4 ns.
(3) Use whichever value is greater.
(4) The ASP ***does not*** require a duty cycle specification, just ensure the minimum pulse duration specification is met.

**Table 6-89. Switching Characteristics Over Recommended Operating Conditions for ASP[1] [2] (see Figure 6-67)**

| NO. | PARAMETER | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X int | 38.5[3] | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X int | C - 1[4] | C + 1[4] | ns |
| 4 | $t_{d(CKRH-FRV)}$ | Delay time, CLKR high to internal FSR valid | CLKR int | -2.1 | 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 | 14.4 | |
| 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 | 13 | |
| 13 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | CLKX int | -3.9 + D1[5] | 4 + D2[5] | ns |
| | | | CLKX ext | 2.1 + D1[5] | 14.5 + D2[5] | ns |
| 14 | $t_{d(FXH-DXV)}$ | Delay time, FSX high to DX valid ONLY applies when in data delay 0 (XDATDLY = 00b) mode | FSX int | -2.3 + D1[6] | 4 + D2[6] | ns |
| | | | FSX ext | 1.9 + D1[6] | 12.1 + D2[6] | |

(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) 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 EDMA3 limitations and AC timing requirements.
(4) C = H or L
S = sample rate generator input clock = 4P if CLKSM = 1 (P = 1/CPU clock frequency [SYSCLK1])
S = sample rate generator input clock = Not Supported if CLKSM = 0 (no CLKS pin on DM6446)
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
H = (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
L = (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
CLKGDV should be set appropriately to ensure the ASP bit rate *does not* exceed the maximum limit [see footnote (3) above].
(5) 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 = 4P, D2 = 8P
(6) 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 = 4P, D2 = 8P




A. Parameter No. 13 applies to the first data bit *only* when XDATDLY ≠ 0.

**Figure 6-67. ASP Timing**

## 6.20 Ethernet Media Access Controller (EMAC)

The Ethernet Media Access Controller (EMAC) provides an efficient interface between DM6446 and the network. The DM6446 EMAC support both 10Base-T and 100Base-TX, or 10 Mbits/second (Mbps) and 100 Mbps in either half- or full-duplex mode, with hardware flow control and quality of service (QOS) support.

The EMAC controls the flow of packet data from the DM6446 device to the PHY. The MDIO module controls PHY configuration and status monitoring.

Both the EMAC and the MDIO modules interface to the DM6446 device through a custom interface that allows efficient data transmission and reception. This custom interface is referred to as the EMAC control module, and is considered integral to the EMAC/MDIO peripheral. The control module is also used to multiplex and control interrupts.

For more detailed information on the EMAC peripheral, see the *TMS320DM644x DMSoC Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) Module User's Guide* (literature number SPRUE24). For a list of supported registers and register fields, see Table 6-90 [Ethernet MAC (EMAC) Control Registers] and Table 6-91 [EMAC Statistics Registers] in this data manual.

### *6.20.1 EMAC Peripheral Register Description(s)*

**Table 6-90. Ethernet MAC (EMAC) Control Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 01C8 0000 | TXIDVER | Transmit Identification and Version Register |
| 01C8 0004 | TXCONTROL | Transmit Control Register |
| 01C8 0008 | TXTEARDOWN | Transmit Teardown Register |
| 01C8 0010 | RXIDVER | Receive Identification and Version Register |
| 01C8 0014 | RXCONTROL | Receive Control Register |
| 01C8 0018 | RXTEARDOWN | Receive Teardown Register |
| 01C8 0080 | TXINTSTATRAW | Transmit Interrupt Status (Unmasked) Register |
| 01C8 0084 | TXINTSTATMASKED | Transmit Interrupt Status (Masked) Register |
| 01C8 0088 | TXINTMASKSET | Transmit Interrupt Mask Set Register |
| 01C8 008C | TXINTMASKCLEAR | Transmit Interrupt Mask Clear Register |
| 01C8 0090 | MACINVECTOR | MAC Input Vector Register |
| 01C8 00A0 | RXINTSTATRAW | Receive Interrupt Status (Unmasked) Register |
| 01C8 00A4 | RXINTSTATMASKED | Receive Interrupt Status (Masked) Register |
| 01C8 00A8 | RXINTMASKSET | Receive Interrupt Mask Set Register |
| 01C8 00AC | RXINTMASKCLEAR | Receive Interrupt Mask Clear Register |
| 01C8 00B0 | MACINTSTATRAW | MAC Interrupt Status (Unmasked) Register |
| 01C8 00B4 | MACINTSTATMASKED | MAC Interrupt Status (Masked) Register |
| 01C8 00B8 | MACINTMASKSET | MAC Interrupt Mask Set Register |
| 01C8 00BC | MACINTMASKCLEAR | MAC Interrupt Mask Clear Register |
| 01C8 0100 | RXMBPENABLE | Receive Multicast/Broadcast/Promiscuous Channel Enable Register |
| 01C8 0104 | RXUNICASTSET | Receive Unicast Enable Set Register |
| 01C8 0108 | RXUNICASTCLEAR | Receive Unicast Clear Register |
| 01C8 010C | RXMAXLEN | Receive Maximum Length Register |
| 01C8 0110 | RXBUFFEROFFSET | Receive Buffer Offset Register |
| 01C8 0114 | RXFILTERLOWTHRESH | Receive Filter Low Priority Frame Threshold Register |
| 01C8 0120 | RX0FLOWTHRESH | Receive Channel 0 Flow Control Threshold Register |
| 01C8 0124 | RX1FLOWTHRESH | Receive Channel 1 Flow Control Threshold Register |
| 01C8 0128 | RX2FLOWTHRESH | Receive Channel 2 Flow Control Threshold Register |
| 01C8 012C | RX3FLOWTHRESH | Receive Channel 3 Flow Control Threshold Register |

**Table 6-90. Ethernet MAC (EMAC) Control Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 01C8 0130 | RX4FLOWTHRESH | Receive Channel 4 Flow Control Threshold Register |
| 01C8 0134 | RX5FLOWTHRESH | Receive Channel 5 Flow Control Threshold Register |
| 01C8 0138 | RX6FLOWTHRESH | Receive Channel 6 Flow Control Threshold Register |
| 01C8 013C | RX7FLOWTHRESH | Receive Channel 7 Flow Control Threshold Register |
| 01C8 0140 | RX0FREEBUFFER | Receive Channel 0 Free Buffer Count Register |
| 01C8 0144 | RX1FREEBUFFER | Receive Channel 1 Free Buffer Count Register |
| 01C8 0148 | RX2FREEBUFFER | Receive Channel 2 Free Buffer Count Register |
| 01C8 014C | RX3FREEBUFFER | Receive Channel 3 Free Buffer Count Register |
| 01C8 0150 | RX4FREEBUFFER | Receive Channel 4 Free Buffer Count Register |
| 01C8 0154 | RX5FREEBUFFER | Receive Channel 5 Free Buffer Count Register |
| 01C8 0158 | RX6FREEBUFFER | Receive Channel 6 Free Buffer Count Register |
| 01C8 015C | RX7FREEBUFFER | Receive Channel 7 Free Buffer Count Register |
| 01C8 0160 | MACCONTROL | MAC Control Register |
| 01C8 0164 | MACSTATUS | MAC Status Register |
| 01C8 0168 | EMCONTROL | Emulation Control Register |
| 01C8 016C | FIFOCONTROL | FIFO Control Register (Transmit and Receive) |
| 01C8 0170 | MACCONFIG | MAC Configuration Register |
| 01C8 0174 | SOFTRESET | Soft Reset Register |
| 01C8 01D0 | MACSRCADDRLO | MAC Source Address Low Bytes Register (Lower 16-bits) |
| 01C8 01D4 | MACSRCADDRHI | MAC Source Address High Bytes Register (Upper 32-bits) |
| 01C8 01D8 | MACHASH1 | MAC Hash Address Register 1 |
| 01C8 01DC | MACHASH2 | MAC Hash Address Register 2 |
| 01C8 01E0 | BOFFTEST | Back Off Test Register |
| 01C8 01E4 | TPACETEST | Transmit Pacing Algorithm Test Register |
| 01C8 01E8 | RXPAUSE | Receive Pause Timer Register |
| 01C8 01EC | TXPAUSE | Transmit Pause Timer Register |
| 01C8 0200 - 01C8 02FC | (see Table 6-91) | EMAC Statistics Registers |
| 01C8 0500 | MACADDRLO | MAC Address Low Bytes Register |
| 01C8 0504 | MACADDRHI | MAC Address High Bytes Register |
| 01C8 0508 | MACINDEX | MAC Index Register |
| 01C8 0600 | TX0HDP | Transmit Channel 0 DMA Head Descriptor Pointer Register |
| 01C8 0604 | TX1HDP | Transmit Channel 1 DMA Head Descriptor Pointer Register |
| 01C8 0608 | TX2HDP | Transmit Channel 2 DMA Head Descriptor Pointer Register |
| 01C8 060C | TX3HDP | Transmit Channel 3 DMA Head Descriptor Pointer Register |
| 01C8 0610 | TX4HDP | Transmit Channel 4 DMA Head Descriptor Pointer Register |
| 01C8 0614 | TX5HDP | Transmit Channel 5 DMA Head Descriptor Pointer Register |
| 01C8 0618 | TX6HDP | Transmit Channel 6 DMA Head Descriptor Pointer Register |
| 01C8 061C | TX7HDP | Transmit Channel 7 DMA Head Descriptor Pointer Register |
| 01C8 0620 | RX0HDP | Receive Channel 0 DMA Head Descriptor Pointer Register |
| 01C8 0624 | RX1HDP | Receive Channel 1 DMA Head Descriptor Pointer Register |
| 01C8 0628 | RX2HDP | Receive Channel 2 DMA Head Descriptor Pointer Register |
| 01C8 062C | RX3HDP | Receive Channel 3 DMA Head Descriptor Pointer Register |
| 01C8 0630 | RX4HDP | Receive Channel 4 DMA Head Descriptor Pointer Register |
| 01C8 0634 | RX5HDP | Receive Channel 5 DMA Head Descriptor Pointer Register |
| 01C8 0638 | RX6HDP | Receive Channel 6 DMA Head Descriptor Pointer Register |
| 01C8 063C | RX7HDP | Receive Channel 7 DMA Head Descriptor Pointer Register |
| 01C8 0640 | TX0CP | Transmit Channel 0 Completion Pointer (Interrupt Acknowledge) Register |

**Table 6-90. Ethernet MAC (EMAC) Control Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 01C8 0644 | TX1CP | Transmit Channel 1 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0648 | TX2CP | Transmit Channel 2 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 064C | TX3CP | Transmit Channel 3 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0650 | TX4CP | Transmit Channel 4 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0654 | TX5CP | Transmit Channel 5 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0658 | TX6CP | Transmit Channel 6 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 065C | TX7CP | Transmit Channel 7 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0660 | RX0CP | Receive Channel 0 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0664 | RX1CP | Receive Channel 1 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0668 | RX2CP | Receive Channel 2 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 066C | RX3CP | Receive Channel 3 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0670 | RX4CP | Receive Channel 4 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0674 | RX5CP | Receive Channel 5 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 0678 | RX6CP | Receive Channel 6 Completion Pointer (Interrupt Acknowledge) Register |
| 01C8 067C | RX7CP | Receive Channel 7 Completion Pointer (Interrupt Acknowledge) Register |

**Table 6-91. EMAC Statistics Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 01C8 0200 | RXGOODFRAMES | Good Receive Frames Register |
| 01C8 0204 | RXBCASTFRAMES | Broadcast Receive Frames Register (Total number of good broadcast frames received) |
| 01C8 0208 | RXMCASTFRAMES | Multicast Receive Frames Register (Total number of good multicast frames received) |
| 01C8 020C | RXPAUSEFRAMES | Pause Receive Frames Register |
| 01C8 0210 | RXCRCERRORS | Receive CRC Errors Register (Total number of frames received with CRC errors) |
| 01C8 0214 | RXALIGNCODEERRORS | Receive Alignment/Code Errors Register (Total number of frames received with alignment/code errors) |
| 01C8 0218 | RXOVERSIZED | Receive Oversized Frames Register (Total number of oversized frames received) |
| 01C8 021C | RXJABBER | Receive Jabber Frames Register (Total number of jabber frames received) |
| 01C8 0220 | RXUNDERSIZED | Receive Undersized Frames Register (Total number of undersized frames received) |
| 01C8 0224 | RXFRAGMENTS | Receive Frame Fragments Register |
| 01C8 0228 | RXFILTERED | Filtered Receive Frames Register |
| 01C8 022C | RXQOSFILTERED | Received QOS Filtered Frames Register |
| 01C8 0230 | RXOCTETS | Receive Octet Frames Register (Total number of received bytes in good frames) |
| 01C8 0234 | TXGOODFRAMES | Good Transmit Frames Register (Total number of good frames transmitted) |
| 01C8 0238 | TXBCASTFRAMES | Broadcast Transmit Frames Register |
| 01C8 023C | TXMCASTFRAMES | Multicast Transmit Frames Register |
| 01C8 0240 | TXPAUSEFRAMES | Pause Transmit Frames Register |
| 01C8 0244 | TXDEFERRED | Deferred Transmit Frames Register |
| 01C8 0248 | TXCOLLISION | Transmit Collision Frames Register |
| 01C8 024C | TXSINGLECOLL | Transmit Single Collision Frames Register |
| 01C8 0250 | TXMULTICOLL | Transmit Multiple Collision Frames Register |
| 01C8 0254 | TXEXCESSIVECOLL | Transmit Excessive Collision Frames Register |
| 01C8 0258 | TXLATECOLL | Transmit Late Collision Frames Register |
| 01C8 025C | TXUNDERRUN | Transmit Underrun Error Register |
| 01C8 0260 | TXCARRIERSENSE | Transmit Carrier Sense Errors Register |
| 01C8 0264 | TXOCTETS | Transmit Octet Frames Register |
| 01C8 0268 | FRAME64 | Transmit and Receive 64 Octet Frames Register |
| 01C8 026C | FRAME65T127 | Transmit and Receive 65 to 127 Octet Frames Register |
| 01C8 0270 | FRAME128T255 | Transmit and Receive 128 to 255 Octet Frames Register |
| 01C8 0274 | FRAME256T511 | Transmit and Receive 256 to 511 Octet Frames Register |
| 01C8 0278 | FRAME512T1023 | Transmit and Receive 512 to 1023 Octet Frames Register |
| 01C8 027C | FRAME1024TUP | Transmit and Receive 1024 to 1518 Octet Frames Register |
| 01C8 0280 | NETOCTETS | Network Octet Frames Register |
| 01C8 0284 | RXSOFOVERRUNS | Receive FIFO or DMA Start of Frame Overruns Register |
| 01C8 0288 | RXMOFOVERRUNS | Receive FIFO or DMA Middle of Frame Overruns Register |
| 01C8 028C | RXDMAOVERRUNS | Receive DMA Start of Frame and Middle of Frame Overruns Register |

**Table 6-92. EMAC Control Module Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C8 1004 | EWCTL | Interrupt control register |
| 0x01C8 1008 | EWINTTCNT | Interrupt timer count |

**Table 6-93. EMAC Control Module RAM**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C8 2000 - 0x01C8 3FFF | | EMAC Control Module Descriptor Memory |

### 6.20.2 EMAC Electrical Data/Timing

**Table 6-94. Timing Requirements for MRCLK (see Figure 6-68)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(MRCLK)}$ | Cycle time, MRCLK | 40 | | ns |
| 2 | $t_{w(MRCLKH)}$ | Pulse duration, MRCLK high | 14 | | ns |
| 3 | $t_{w(MRCLKL)}$ | Pulse duration, MRCLK low | 14 | | ns |

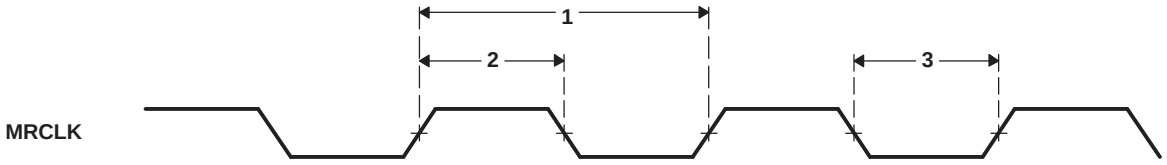



**Figure 6-68. MRCLK Timing (EMAC - Receive)**

**Table 6-95. Timing Requirements for MTCLK (see Figure 6-69)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(MTCLK)}$ | Cycle time, MTCLK | 40 | | ns |
| 2 | $t_{w(MTCLKH)}$ | Pulse duration, MTCLK high | 14 | | ns |
| 3 | $t_{w(MTCLKL)}$ | Pulse duration, MTCLK low | 14 | | ns |

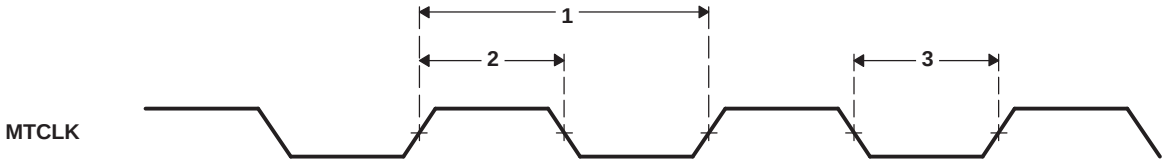



**Figure 6-69. MTCLK Timing (EMAC - Transmit)**

**Table 6-96. Timing Requirements for EMAC MII Receive 10/100 Mbit/s(1) (see Figure 6-70)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(MRXD-MRCLKH)}$ | Setup time, receive selected signals valid before MRCLK high | 8 | | ns |
| 2 | $t_{h(MRCLKH-MRXD)}$ | Hold time, receive selected signals valid after MRCLK high | 8 | | ns |

(1) Receive selected signals include: MRXD3-MRXD0, MRXDV, and MRXER.

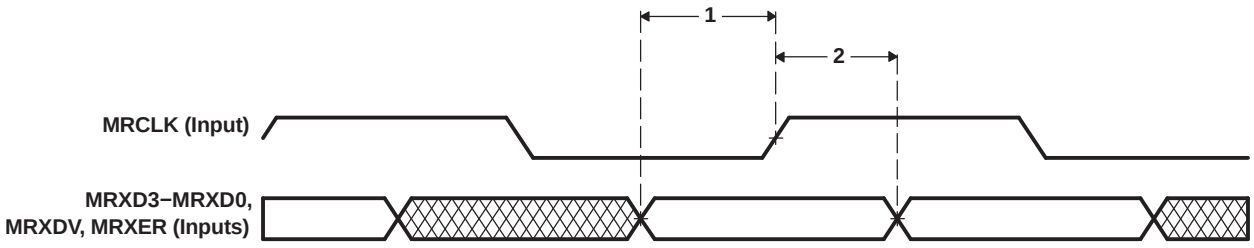



**Figure 6-70. EMAC Receive Interface Timing**

**Table 6-97. Switching Characteristics Over Recommended Operating Conditions for EMAC MII Transmit 10/100 Mbit/s(1) (see Figure 6-71)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(MTCLKH-MTXD)}$ | Delay time, MTCLK high to transmit selected signals valid | 5 | 25 | ns |

(1) Transmit selected signals include: MTXD3-MTXD0, and MTXEN.

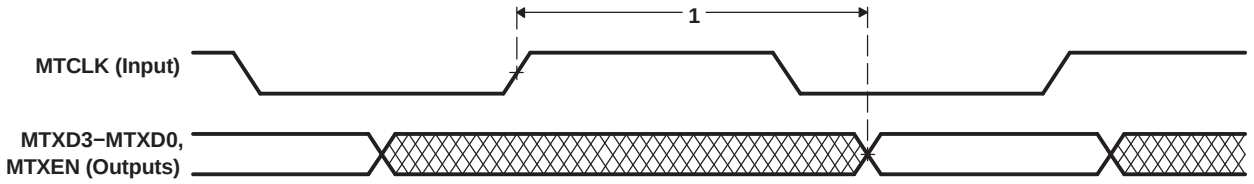



**Figure 6-71. EMAC Transmit Interface Timing**

## 6.21 Management Data Input/Output (MDIO)

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

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

For more detailed information on the MDIO peripheral, see the *TMS320DM644x DMSoC Ethernet Media Access Controller (EMAC)/ Management Data Input/Output (MDIO) Module User's Guide* (literature number SPRUE24). For a list of supported registers and register fields, see Table 6-98 [MDIO Registers] in this data manual.

### *6.21.1 Peripheral Register Description(s)*

**Table 6-98. MDIO Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C8 4000 | – | Reserved |
| 0x01C8 4004 | CONTROL | MDIO Control Register |
| 0x01C8 4008 | ALIVE | MDIO PHY Alive Status Register |
| 0x01C8 400C | LINK | MDIO PHY Link Status Register |
| 0x01C8 4010 | LINKINTRAW | MDIO Link Status Change Interrupt (Unmasked) Register |
| 0x01C8 4014 | LINKINTMASKED | MDIO Link Status Change Interrupt (Masked) Register |
| 0x01C8 4018 | – | Reserved |
| 0x01C8 4020 | USERINTRAW | MDIO User Command Complete Interrupt (Unmasked) Register |
| 0x01C8 4024 | USERINTMASKED | MDIO User Command Complete Interrupt (Masked) Register |
| 0x01C8 4028 | USERINTMASKSET | MDIO User Command Complete Interrupt Mask Set Register |
| 0x01C8 402C | USERINTMASKCLEAR | MDIO User Command Complete Interrupt Mask Clear Register |
| 0x01C8 4030 - 0x01C8 407C | – | Reserved |
| 0x01C8 4080 | USERACCESS0 | MDIO User Access Register 0 |
| 0x01C8 4084 | USERPHYSEL0 | MDIO User PHY Select Register 0 |
| 0x01C8 4088 | USERACCESS1 | MDIO User Access Register 1 |
| 0x01C8 408C | USERPHYSEL1 | MDIO User PHY Select Register 1 |
| 0x01C8 4090 - 0x01C8 47FF | – | Reserved |

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

**Table 6-99. Timing Requirements for MDIO Input (see Figure 6-72 and Figure 6-73)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(MDCLK)}$ | Cycle time, MDCLK | 400 | | ns |
| 2 | $t_{w(MDCLK)}$ | Pulse duration, MDCLK high/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 | 15 | | ns |
| 5 | $t_{h(MDCLKH-MDIO)}$ | Hold time, MDIO data input valid after MDCLK high | 0 | | ns |

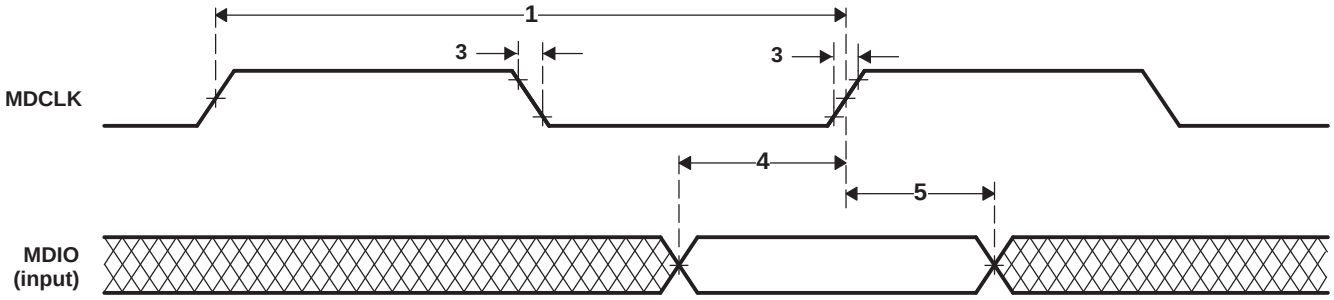



**Figure 6-72. MDIO Input Timing**

**Table 6-100. Switching Characteristics Over Recommended Operating Conditions for MDIO Output (see Figure 6-73)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 7 | $t_{d(MDCLKL-MDIO)}$ | Delay time, MDCLK low to MDIO data output valid | -0.6 | 100 | ns |

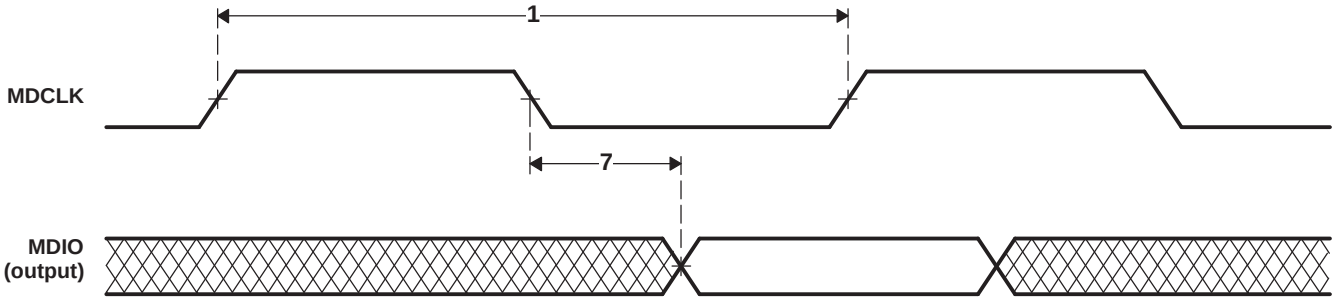



**Figure 6-73. MDIO Output Timing**

## 6.22 Timer

The DM6446 device has 3 64-bit general-purpose timers which have the following features:

- 64-bit count-up counter
- Timer modes:
  - 64-bit general-purpose timer mode
  - Dual 32-bit general-purpose timer mode (Timer 0 and 1)
  - Watchdog timer mode (Timer 2)
- 2 possible clock sources:
  - Internal clock
  - External clock input via timer input pin TIM_IN (Timer 0 only)
- 2 operation modes:
  - One-time operation (timer runs for one period then stops)
  - Continuous operation (timer automatically resets after each period)
- Generates interrupts to both the DSP and the ARM CPUs
- Generates sync event to EDMA3

For more detailed information, see the *TMS320DM644x DMSoC 64-Bit Timer User's Guide* (literature number SPRUE26).

### *6.22.1 Timer Peripheral Register Description(s)*

**Table 6-101. Timer 0 Registers**

| HEX ADDRESS RANGE | ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C2 1400 | - | Reserved |
| 0x01C2 1404 | EMUMGT_CLKSPD | Timer 0 Emulation Management/Clock Speed Register |
| 0x01C2 1410 | TIM12 | Timer 0 Counter Register 12 |
| 0x01C2 1414 | TIM34 | Timer 0 Counter Register 34 |
| 0x01C2 1418 | PRD12 | Timer 0 Period Register 12 |
| 0x01C2 141C | PRD34 | Timer 0 Period Register 34 |
| 0x01C2 1420 | TCR | Timer 0 Control Register |
| 0x01C2 1424 | TGCR | Timer 0 Global Control Register |
| 0x01C2 1428 - 0x01C2 17FF | - | Reserved |

**Table 6-102. Timer 1 Registers**

| HEX ADDRESS RANGE | ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C2 1800 | - | Reserved |
| 0x01C2 1804 | EMUMGT_CLKSPD | Timer 1 Emulation Management/Clock Speed Register |
| 0x01C2 1810 | TIM12 | Timer 1 Counter Register 12 |
| 0x01C2 1814 | TIM34 | Timer 1 Counter Register 34 |
| 0x01C2 1818 | PRD12 | Timer 1 Period Register 12 |
| 0x01C2 181C | PRD34 | Timer 1 Period Register 34 |
| 0x01C2 1820 | TCR | Timer 1 Control Register |
| 0x01C2 1824 | TGCR | Timer 1 Global Control Register |
| 0x01C2 1828 - 0x01C2 1BFF | - | Reserved |

**Table 6-103. Timer 2 (Watchdog) Registers**

| HEX ADDRESS RANGE | ACRONYM | DESCRIPTION |
|---|---|---|
| 0x01C2 1C00 | - | Reserved |
| 0x01C2 1C04 | EMUMGT_CLKSPD | Timer 2 Emulation Management/Clock Speed Register |
| 0x01C2 1C10 | TIM12 | Timer 2 Counter Register 12 |
| 0x01C2 1C14 | TIM34 | Timer 2 Counter Register 34 |
| 0x01C2 1C18 | PRD12 | Timer 2 Period Register 12 |
| 0x01C2 1C1C | PRD34 | Timer 2 Period Register 34 |
| 0x01C2 1C20 | TCR | Timer 2 Control Register |
| 0x01C2 1C24 | TGCR | Timer 2 Global Control Register |
| 0x01C2 1C28 | WDTCR | Timer 2 Watchdog Timer Control Register |
| 0x01C2 1C2C - 0x01C2 1FFF | - | Reserved |

### 6.22.2 Timer Electrical Data/Timing

**Table 6-104. Timing Requirements for Timer Input(1) (2) (see Figure 6-74)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(TIN)}$ | Cycle time, TIM_IN | 4P | | ns |
| 2 | $t_{w(TINPH)}$ | Pulse duration, TIM_IN high | 0.45C | 0.55C | ns |
| 3 | $t_{w(TINPL)}$ | Pulse duration, TIM_IN low | 0.45C | 0.55C | ns |
| 4 | $t_{t(TIN)}$ | Transition time, TIM_IN | | 0.05C | ns |

(1) P = MXI/CLKIN cycle time in ns. For example, when MXI/CLKIN frequency is 27 MHz, use P = 37.0$\overline{37}$ ns.
(2) C = TIM_IN cycle time in ns. For example, when TIM_IN frequency is 27 MHz, use C = 37.0$\overline{37}$ ns

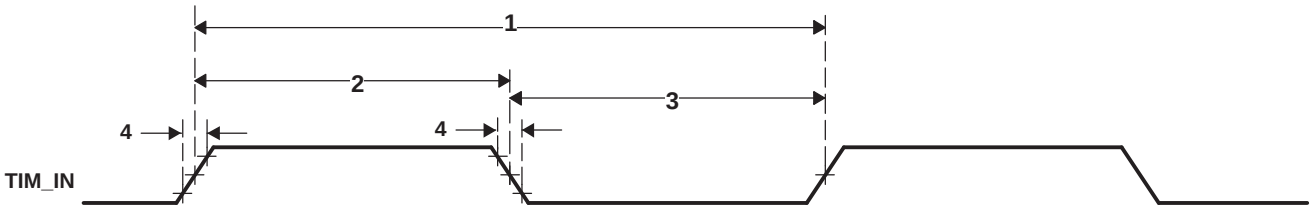



**Figure 6-74. Timer Timing**

## 6.23 Pulse Width Modulator (PWM)

The 3 DM6446 Pulse Width Modulator (PWM) peripherals support the following features:

- Period counter
- First-phase duration counter
- Repeat count for one-shot operation
- Configurable to operate in either one-shot or continuous mode
- Buffered period and first-phase duration registers
- One-shot operation triggerable by hardware events with programmable edge transitions. (low-to-high or high-to-low).
- One-shot operation generates N+1 periods of waveform, N being the repeat count register value
- Emulation support

The register memory maps for PWM0/1/2 are shown in Table 6-105, Table 6-106, and Table 6-107.

**Table 6-105. PWM0 Register Memory Map**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C2 2000 | | Reserved |
| 0x01C2 2004 | PCR | PWM0 Peripheral Control Register |
| 0x01C2 2008 | CFG | PWM0 Configuration Register |
| 0x01C2 200C | START | PWM0 Start Register |
| 0x01C2 2010 | RPT | PWM0 Repeat Count Register |
| 0x01C2 2014 | PER | PWM0 Period Register |
| 0x01C2 2018 | PH1D | PWM0 First-Phase Duration Register |
| 0x01C2 201C - 0x01C2 23FF | - | Reserved |

**Table 6-106. PWM1 Register Memory Map**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C2 2400 | | Reserved |
| 0x01C2 2404 | PCR | PWM1 Peripheral Control Register |
| 0x01C2 2408 | CFG | PWM1 Configuration Register |
| 0x01C2 240C | START | PWM1 Start Register |
| 0x01C2 2410 | RPT | PWM1 Repeat Count Register |
| 0x01C2 2414 | PER | PWM1 Period Register |
| 0x01C2 2418 | PH1D | PWM1 First-Phase Duration Register |
| 0x01C2 241C -0x01C2 27FF | - | Reserved |

**Table 6-107. PWM2 Register Memory Map**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01C2 2800 | | Reserved |
| 0x01C2 2804 | PCR | PWM2 Peripheral Control Register |
| 0x01C2 2808 | CFG | PWM2 Configuration Register |
| 0x01C2 280C | START | PWM2 Start Register |
| 0x01C2 2810 | RPT | PWM2 Repeat Count Register |
| 0x01C2 2814 | PER | PWM2 Period Register |
| 0x01C2 2818 | PH1D | PWM2 First-Phase Duration Register |
| 0x01C2 281C - 0x01C2 2BFF | - | Reserved |

### 6.23.1 PWM0/1/2 Electrical/Timing Data

**Table 6-108. Switching Characteristics Over Recommended Operating Conditions for PWM0/1/2 Outputs (see Figure 6-75 and Figure 6-76)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(PWMH)}$ | Pulse duration, PWMx high | 37 | | ns |
| 2 | $t_{w(PWML)}$ | Pulse duration, PWMx low | 37 | | ns |
| 3 | $t_{t(PWM)}$ | Transition time, PWMx | | 5 | ns |
| 4 | $t_{d(CCDC-PWMV)}$ | Delay time, CCDC(VD) trigger event to PWMx valid | 2 | 10 | ns |

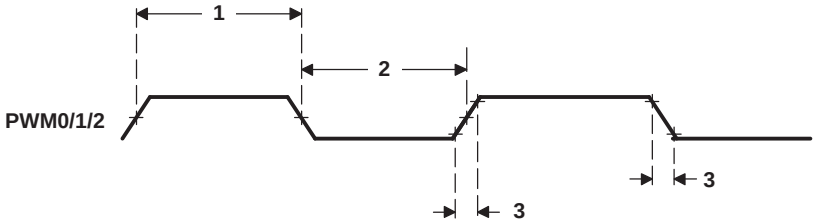



**Figure 6-75. PWM Output Timing**

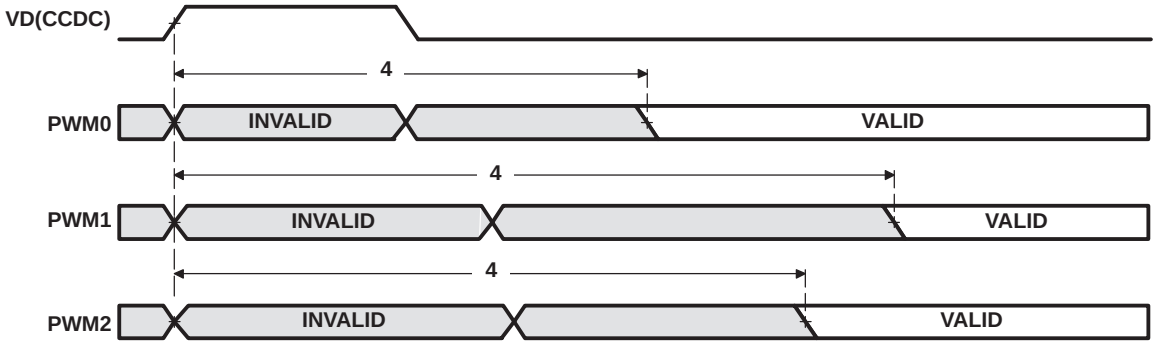



**Figure 6-76. PWM Output Delay Timing**

## 6.24 VLYNQ

The DM6446 VLYNQ peripheral provides a high speed serial communications interface with the following features.

- Low Pin Count
- Scalable Performance / Support
- Simple Packet Based Transfer Protocol for Memory Mapped Access
  - Write Request / Data Packet
  - Read Request Packet
  - Read Response Data Packet
  - Interrupt Request Packet
- Supports both Symmetric and Asymmetric Operation
  - Tx pins on first device connect to Rx pins on second device and vice versa
  - Data pin widths are automatically detected after reset
  - Request packets, response packets, and flow control information are all multiplexed and sent across the same physical pins
  - Supports both Host/Peripheral and Peer to Peer communication
- Simple Block Code Packet Formatting (8b/10b)
- In Band Flow Control
  - No extra pins needed
  - Allows receiver to momentarily throttle back transmitter when overflow is about to occur
  - Uses built in special code capability of block code to seamlessly interleave flow control information with user data
  - Allows system designer to balance cost of data buffering versus performance
- Multiple outstanding transactions
- Automatic packet formatting optimizations
- Internal loop-back mode

### 6.24.1 VLYNQ Peripheral Register Description(s)

**Table 6-109. VLYNQ Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01E0 1000 | - | Reserved |
| 0x01E0 1004 | CTRL | VLYNQ Local Control Register |
| 0x01E0 1008 | STAT | VLYNQ Local Status Register |
| 0x01E0 100C | INTPRI | VLYNQ Local Interrupt Priority Vector Status/Clear Register |
| 0x01E0 1010 | INTSTATCLR | VLYNQ Local Unmasked Interrupt Status/Clear Register |
| 0x01E0 1014 | INTPENDSET | VLYNQ Local Interrupt Pending/Set Register |
| 0x01E0 1018 | INTPTR | VLYNQ Local Interrupt Pointer Register |
| 0x01E0 101C | XAM | VLYNQ Local Transmit Address Map Register |
| 0x01E0 1020 | RAMS1 | VLYNQ Local Receive Address Map Size 1 Register |
| 0x01E0 1024 | RAMO1 | VLYNQ Local Receive Address Map Offset 1 Register |
| 0x01E0 1028 | RAMS2 | VLYNQ Local Receive Address Map Size 2 Register |
| 0x01E0 102C | RAMO2 | VLYNQ Local Receive Address Map Offset 2 Register |
| 0x01E0 1030 | RAMS3 | VLYNQ Local Receive Address Map Size 3 Register |
| 0x01E0 1034 | RAMO3 | VLYNQ Local Receive Address Map Offset 3 Register |
| 0x01E0 1038 | RAMS4 | VLYNQ Local Receive Address Map Size 4 Register |
| 0x01E0 103C | RAMO4 | VLYNQ Local Receive Address Map Offset 4 Register |
| 0x01E0 1040 | CHIPVER | VLYNQ Local Chip Version Register |

**Table 6-109. VLYNQ Registers (continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 0x01E0 1044 | AUTNGO | VLYNQ Local Auto Negotiation Register |
| 0x01E0 1048 | - | Reserved |
| 0x01E0 104C | - | Reserved |
| 0x01E0 1050 - 0x01E0 105C | - | Reserved |
| 0x01E0 1060 | - | Reserved |
| 0x01E0 1064 | - | Reserved |
| 0x01E0 1068 - 0x01E0 107C | - | Reserved *for future use* |
| 0x01E0 1080 | RREVID | VLYNQ Remote Revision Register |
| 0x01E0 1084 | RCTRL | VLYNQ Remote Control Register |
| 0x01E0 1088 | RSTAT | VLYNQ Remote Status Register |
| 0x01E0 108C | RINTPRI | VLYNQ Remote Interrupt Priority Vector Status/Clear Register |
| 0x01E0 1090 | RINTSTATCLR | VLYNQ Remote Unmasked Interrupt Status/Clear Register |
| 0x01E0 1094 | RINTPENDSET | VLYNQ Remote Interrupt Pending/Set Register |
| 0x01E0 1098 | RINTPTR | VLYNQ Remote Interrupt Pointer Register |
| 0x01E0 109C | RXAM | VLYNQ Remote Transmit Address Map Register |
| 0x01E0 10A0 | RRAMS1 | VLYNQ Remote Receive Address Map Size 1 Register |
| 0x01E0 10A4 | RRAMO1 | VLYNQ Remote Receive Address Map Offset 1 Register |
| 0x01E0 10A8 | RRAMS2 | VLYNQ Remote Receive Address Map Size 2 Register |
| 0x01E0 10AC | RRAMO2 | VLYNQ Remote Receive Address Map Offset 2 Register |
| 0x01E0 10B0 | RRAMS3 | VLYNQ Remote Receive Address Map Size 3 Register |
| 0x01E0 10B4 | RRAMO3 | VLYNQ Remote Receive Address Map Offset 3 Register |
| 0x01E0 10B8 | RRAMS4 | VLYNQ Remote Receive Address Map Size 4 Register |
| 0x01E0 10BC | RRAMO4 | VLYNQ Remote Receive Address Map Offset 4 Register |
| 0x01E0 10C0 | RCHIPVER | VLYNQ Remote Chip Version Register (values on the device_id and device_rev pins of remote VLYNQ) |
| 0x01E0 10C4 | RAUTNGO | VLYNQ Remote Auto Negotiation Register |
| 0x01E0 10C8 | RMANNGO | VLYNQ Remote Manual Negotiation Register |
| 0x01E0 10CC | RNGOSTAT | VLYNQ Remote Negotiation Status Register |
| 0x01E0 10D0 - 0x01E0 10DC | - | Reserved |
| 0x01E0 10E0 | RINTVEC0 | VLYNQ Remote Interrupt Vectors 3 - 0 (sourced from vlynq_int_i[3:0] port of remote VLYNQ) |
| 0x01E0 10E4 | RINTVEC1 | VLYNQ Remote Interrupt Vectors 7 - 4 (sourced from vlynq_int_i[7:4] port of remote VLYNQ) |
| 0x01E0 10E8 - 0x01E0 10FC | - | Reserved *for future use* |
| 0x01E0 1100 - 0x01E0 1FFF | - | Reserved |

### 6.24.2 VLYNQ Electrical Data/Timing

**Table 6-110. Timing Requirements for VLYNQ_CLK for VLYNQ (see Figure 6-77)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(VCLK)}$ | Cycle time, VLYNQ_CLK | 10 | | ns |
| 2 | $t_{w(VCLKH)}$ | Pulse duration, VLYNQ_CLK high **[CLK External]** | 3 | | ns |
| | | Pulse duration, VLYNQ_CLK high **[CLK Internal]** | 4 | | ns |
| 3 | $t_{w(VCLKL)}$ | Pulse duration, VLYNQ_CLK low **[CLK External]** | 3 | | ns |
| | | Pulse duration, VLYNQ_CLK low **[CLK Internal]** | 4 | | ns |
| 4 | $t_{t(VCLK)}$ | Transition time, VLYNQ_CLK | | 3 | ns |

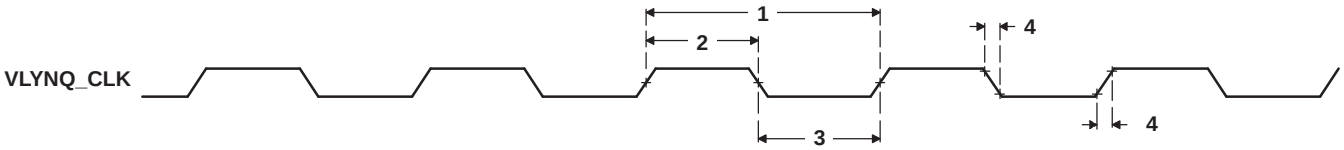



**Figure 6-77. VLYNQ_CLK Timing for VLYNQ**

**Table 6-111. Switching Characteristics Over Recommended Operating Conditions for Transmit Data for the VLYNQ Module (see Figure 6-78)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(VCLKH-TXDI)}$ | Delay time, VLYNQ_CLK high to VLYNQ_TXD[3:0] invalid [**SLOW** Mode] | 1 | | ns |
| | | Delay time, VLYNQ_CLK high to VLYNQ_ TXD[3:0] invalid [**FAST** Mode] | 0.5 | | ns |
| 2 | $t_{d(VCLKH-TXDV)}$ | Delay time, VLYNQ_CLK to VLYNQ_TXD[3:0] valid | | 9.75 | ns |

**Table 6-112. Timing Requirements for Receive Data for the VLYNQ Module (see Figure 6-78)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{su(RXDV-VCLKH)}$ | Setup time, VLYNQ_RXD[3:0] valid before VLYNQ_CLK high | RTM disabled, RTM sample = 3 | 0.8 | ns |
| | | | RTM enabled, RXD Flop = 0 | 2.2 | ns |
| | | | RTM enabled, RXD Flop = 1 | 1.9 | ns |
| | | | RTM enabled, RXD Flop = 2 | 1.4 | ns |
| | | | RTM enabled, RXD Flop = 3 | 0.8 | ns |
| | | | RTM enabled, RXD Flop = 4 | 0.4 | ns |
| | | | RTM enabled, RXD Flop = 5 | 0.1 | ns |
| | | | RTM enabled, RXD Flop = 6 | -0.2 | ns |
| | | | RTM enabled, RXD Flop = 7 | -0.4 | ns |
| 4 | $t_{h(VCLKH-RXDV)}$ | Hold time, VLYNQ_RXD[3:0] valid after VLYNQ_CLK high | RTM disabled, RTM sample = 3 | 2 | ns |
| | | | RTM enabled, RXD Flop = 0 | 0.6 | ns |
| | | | RTM enabled, RXD Flop = 1 | 1.0 | ns |
| | | | RTM enabled, RXD Flop = 2 | 1.5 | ns |
| | | | RTM enabled, RXD Flop = 3 | 2.0 | ns |
| | | | RTM enabled, RXD Flop = 4 | 2.5 | ns |
| | | | RTM enabled, RXD Flop = 5 | 3.0 | ns |
| | | | RTM enabled, RXD Flop = 6 | 3.5 | ns |
| | | | RTM enabled, RXD Flop = 7 | 4.0 | ns |

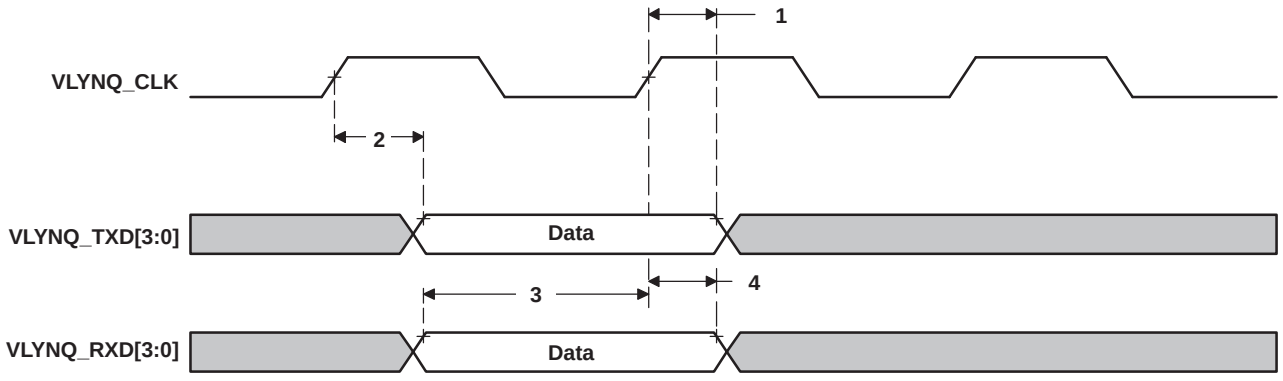



**Figure 6-78. VLYNQ Transmit/Receive Timing**

## 6.25 IEEE 1149.1 JTAG

The JTAG(1) interface is used for BSDL testing and emulation of the DM6446 device.

The DM6446 device requires that both $\overline{\text{TRST}}$ and $\overline{\text{RESET}}$ be asserted upon power up to be properly initialized. While $\overline{\text{RESET}}$ initializes the device, $\overline{\text{TRST}}$ initializes the device's emulation logic. Both resets are required for proper operation.

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

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

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

For maximum reliability, DM6446 includes an internal pulldown (IPD) on the $\overline{\text{TRST}}$ pin to ensure that $\overline{\text{TRST}}$ will always be asserted upon power up and the device's internal emulation logic will always be properly initialized.

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

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

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

### *6.25.1 JTAG Peripheral Register Description(s) – JTAG ID Register*

**Table 6-113. JTAG ID Register**

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

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

The JTAG ID register is a read-only register that identifies to the customer the JTAG/Device ID. For the DM6446 device, the JTAG ID register resides at address location 0x01C4 0028. The register hex value for DM6446 is: **0x0B70 002F for silicon revisions 1.3 and earlier, and 0x1B70 002F for silicon revisions 2.1 and later**. For the actual register bit names and their associated bit field descriptions, see Figure 6-79 and Table 6-114.

| 31-28 | 27-12 | 11-1 | 0 |
|---|---|---|---|
| VARIANT (4-Bit)(A) | PART NUMBER (16-Bit) | MANUFACTURER (11-Bit) | LSB |
| R-000x | R-1011 0111 0000 0000 | R-0000 0010 111 | R-1 |

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

(A) **For silicon revisions 1.3 and earlier, VARIANT = 0000. For silicon revisions 2.1 and later, VARIANT = 0001.**

**Figure 6-79. JTAG ID Register Description - DM6446 Register Value - 0xXB70 002F**

**Table 6-114. JTAG ID Register Selection Bit Descriptions**

| BIT | NAME | DESCRIPTION |
|---|---|---|
| 31:28 | VARIANT | Variant (4-Bit) value. DM6446 value: **0000 for silicon revisions 1.3 and earlier, and 0001 for silicon revisions 2.1 and later**. |
| 27:12 | PART NUMBER | Part Number (16-Bit) value. DM6446 value: 1011 0111 0000 0000. |
| 11-1 | MANUFACTURER | Manufacturer (11-Bit) value. DM6446 value: 0000 0010 111. |
| 0 | LSB | LSB. This bit is read as a "1" for DM6446. |

### *6.25.2 JTAG Test-Port Electrical Data/Timing*

**Table 6-115. Timing Requirements for JTAG Test Port (see Figure 6-80)**

| NO. | | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(TCK)}$ | Cycle time, TCK | 20 | | ns |
| 2 | $t_{w(TCKH)}$ | Pulse duration, TCK high | 8 | | ns |
| 3 | $t_{w(TCKL)}$ | Pulse duration, TCK low | 8 | | ns |
| 4 | $t_{c(RTCK)}$ | Cycle time, RTCK | 20 | | ns |
| 5 | $t_{w(RTCKH)}$ | Pulse duration, RTCK high | 10 | | ns |
| 6 | $t_{w(RTCKL)}$ | Pulse duration, RTCK low | 10 | | ns |
| 7 | $t_{su(TDIV-RTCKH)}$ | Setup time, TDI/TMS valid before RTCK high | 10 | | ns |
| 8 | $t_{h(RTCKH-TDIV)}$ | Hold time, TDI/TMS valid after RTCK high | 1 | | ns |

**Table 6-116. Switching Characteristics Over Recommended Operating Conditions for JTAG Test Port (see Figure 6-80)**

| NO. | PARAMETER | | A-513, -594, -810 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 9 | $t_{d(RTCKL-TDOV)}$ | Delay time, RTCK low to TDO valid | | 15 | ns |

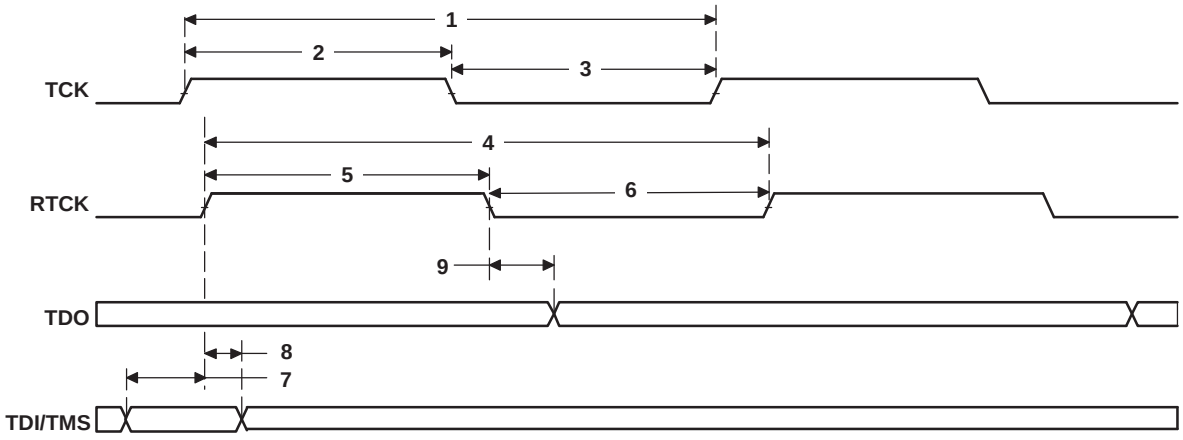



**Figure 6-80. JTAG Test-Port Timing**

# 7 Mechanical Packaging and Orderable Information

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

## 7.1 Thermal Data for ZWT

**Table 7-1. Thermal Resistance Characteristics (PBGA Package) [ZWT]**

| NO. | | | °C/W(1) | AIR FLOW (m/s)(2) |
|---|---|---|---|---|
| 1 | $R\Theta_{JC}$ | Junction-to-case | 6.54 | N/A |
| 2 | $R\Theta_{JB}$ | Junction-to-board | 15.62 | N/A |
| 3 | $R\Theta_{JA}$ | Junction-to-free air | 29.75 | 0.00 |
| 4 | | | 26.78 | 1.0 |
| 5 | | | 26.20 | 2.00 |
| 6 | | | 25.80 | 3.00 |
| 7 | $Psi_{JT}$ | Junction-to-package top | 0.11 | 0.00 |
| 8 | | | 0.15 | 1.0 |
| 9 | | | 0.16 | 2.00 |
| 10 | | | 0.16 | 3.00 |
| 11 | $Psi_{JB}$ | Junction-to-board | 14.79 | 0.00 |
| 12 | | | 14.66 | 1.0 |
| 13 | | | 14.66 | 2.00 |
| 14 | | | 14.66 | 3.00 |

(1) These measurements were conducted in a JEDEC defined 1S2P system and will change based on environment as well as application. For more information, see these EIA/JEDEC standards – EIA/JESD51-2, *Integrated Circuits Thermal Test Method Environment Conditions - Natural Convection (Still Air)* and JESD51-7, *High Effective Thermal Conductivity Test Board for Leaded Surface Mount Packages*.
(2) m/s = meters per second

## 7.2 Packaging Information

The following packaging information and addendum reflect the most current 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DM6446ANB6C2127VC | LIFEBUY | NFBGA | ZWT | 361 | 90 | RoHS & Green | Call TI | Level-3-260C-168 HR | 0 to 85 | DM6446AZWT<br>TMS320<br>DAVINCI | |
| DM6446AZWTKEDACOM | LIFEBUY | NFBGA | ZWT | 361 | 90 | RoHS & Green | Call TI | Level-3-260C-168 HR | 0 to 85 | DM6446AZWT<br>TMS320<br>DAVINCI | |
| DM6446GB6C0121MV | LIFEBUY | NFBGA | ZWT | 361 | 90 | RoHS & Green | Call TI | Level-3-260C-168 HR | 0 to 85 | DM6446AZWT<br>TMS320<br>DAVINCI | |
| TMS320DM6446AZWT | ACTIVE | NFBGA | ZWT | 361 | 90 | RoHS & Green | Call TI | Level-3-260C-168 HR | 0 to 85 | DM6446AZWT<br>TMS320<br>DAVINCI | Samples |
| TMS320DM6446AZWTA | ACTIVE | NFBGA | ZWT | 361 | 90 | RoHS & Green | SNAGCU | Level-3-260C-168 HR | -40 to 105 | DM6446AZWTA<br>TMS320<br>DAVINCI | Samples |
| TMS320DM6446BZWT | ACTIVE | NFBGA | ZWT | 361 | 90 | RoHS & Green | Call TI | Level-3-260C-168 HR | 0 to 85 | DM6446BZWT<br>TMS320<br>DAVINCI | Samples |
| TMS320DM6446BZWT8 | ACTIVE | NFBGA | ZWT | 361 | 90 | RoHS & Green | SNAGCU | Level-3-260C-168 HR | 0 to 85 | DM6446BZWT8<br>TMS320<br>189<br>DAVINCI | Samples |
| TMS320DM6446BZWTA | ACTIVE | NFBGA | ZWT | 361 | 90 | RoHS & Green | SNAGCU | Level-3-260C-168 HR | -40 to 105 | DM6446BZWTA<br>TMS320<br>DAVINCI | Samples |
| TMSDM6446AZWT-COM | LIFEBUY | NFBGA | ZWT | 361 | 90 | RoHS & Green | Call TI | Level-3-260C-168 HR | 0 to 85 | DM6446AZWT<br>TMS320<br>DAVINCI | |
| TNETV6446AINZWT | LIFEBUY | NFBGA | ZWT | 361 | 90 | RoHS & Green | Call TI | Level-3-260C-168 HR | 0 to 85 | DM6446AZWT<br>TMS320<br>DAVINCI | |
| TNETV6446AINZWT8 | LIFEBUY | NFBGA | ZWT | 361 | 90 | RoHS & Green | SNAGCU | Level-3-260C-168 HR | 0 to 85 | DM6446BZWT8<br>TMS320<br>189<br>DAVINCI | |

(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.

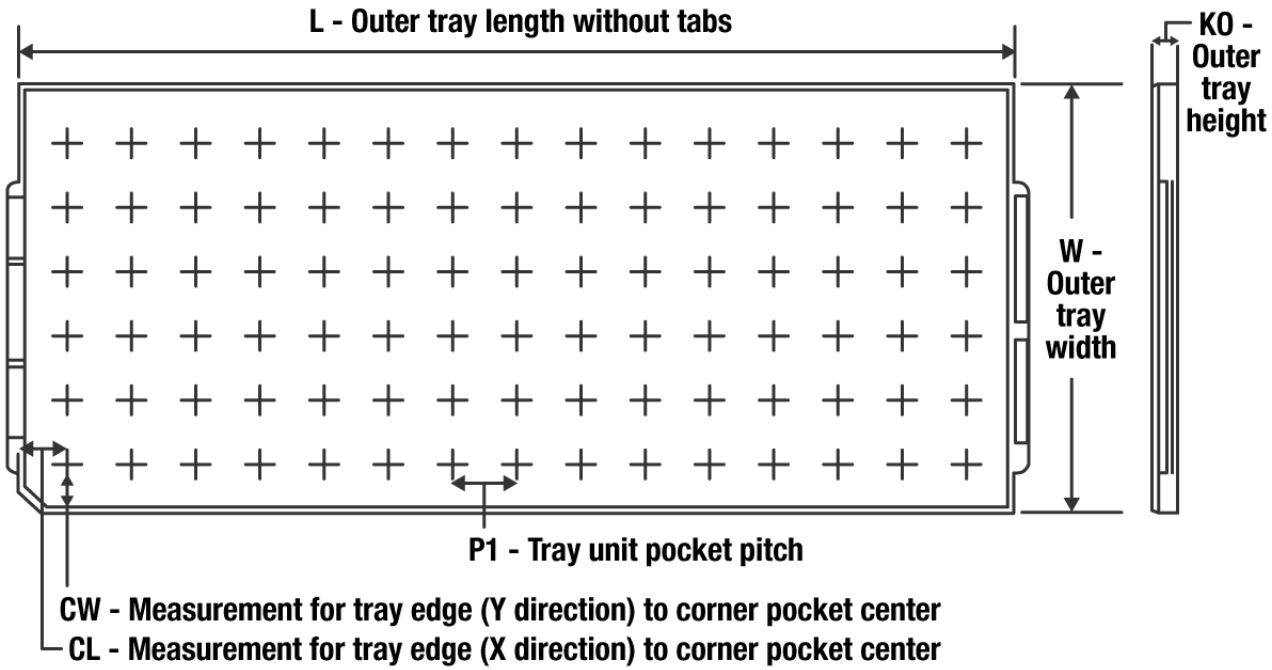
(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(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

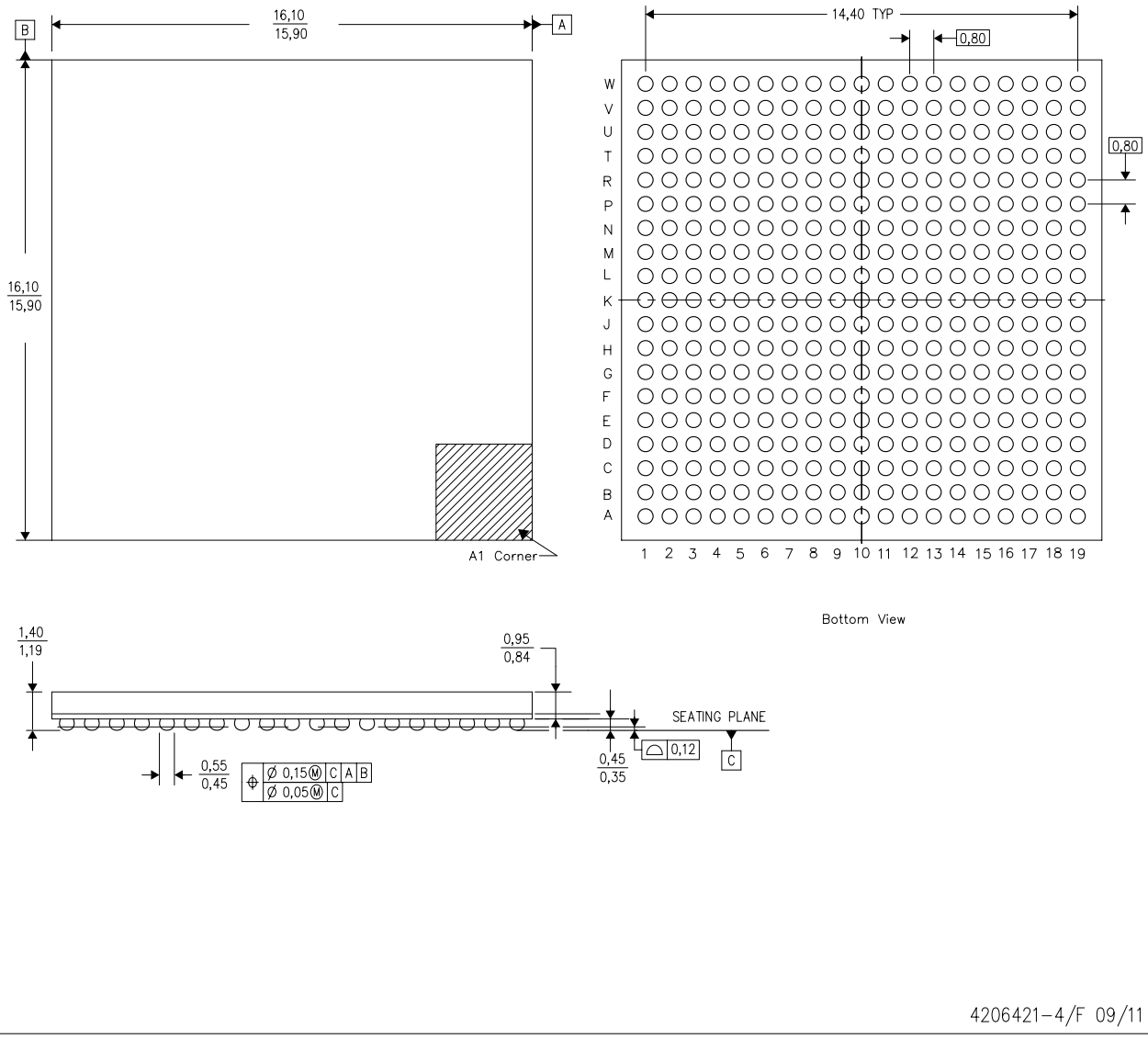



Chamfer on Tray corner indicates Pin 1 orientation of packed units.

*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) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DM6446ANB6C2127VC | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| DM6446AZWTKEDACOM | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| DM6446GB6C0121MV | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| TMS320DM6446AZWT | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| TMS320DM6446AZWTA | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| TMS320DM6446BZWT | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| TMS320DM6446BZWT8 | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| TMS320DM6446BZWTA | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| TMSDM6446AZWT-COM | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| TNETV6446AINZWT | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |
| TNETV6446AINZWT8 | ZWT | NFBGA | 361 | 90 | 6 X 15 | 150 | 315 | 135.9 | 7620 | 20 | 17.5 | 15.45 |

ZWT (S–PBGA–N361) PLASTIC BALL GRID ARRAY




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

# 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 TMS320DM6446BZWT8 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