# TMS320DM36x
# Digital Media System-on-Chip (DMSoC)
# DDR2/mDDR Memory Controller

# User's Guide

Texas Instruments

Literature Number: SPRUFI2
March 2009

Texas Instruments

# Contents

# List of Figures

# List of Tables

# Read This First

This document describes the DDR2/mDDR Memory Controller on the TMS320DM36x Digital Media System-on-Chip (DMSoC).

## Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

## Related Documentation from Texas Instruments

The following documents describe the TMS320DM36x Digital Media System-on-Chip (DMSoC). Copies of these documents are available on the internet at www.ti.com.

**SPRUFG5** — ***TMS320DM365 Digital Media System-on-Chip (DMSoC) ARM Subsystem Reference Guide*** This document describes the ARM Subsystem in the TMS320DM36x 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 components of the ARM Subsystem, the peripherals, and the external memories.

**SPRUFG8** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Video Processing Front End (VPFE) Users Guide*** This document describes the Video Processing Front End (VPFE) in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFG9** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Video Processing Back End (VPBE) Users Guide*** This document describes the Video Processing Back End (VPBE) in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFH0** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) 64-bit Timer Users Guide*** This document describes the operation of the software-programmable 64-bit timers in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFH1** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Serial Peripheral Interface (SPI) Users Guide*** This document describes the serial peripheral interface (SPI) in the TMS320DM36x Digital Media System-on-Chip (DMSoC). The SPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (1 to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communication between the DMSoC and external peripherals. Typical applications include an interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, SPI EPROMs and analog-to-digital converters.

**SPRUFH2** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Universal Asynchronous Receiver/Transmitter (UART) Users Guide*** This document describes the universal asynchronous receiver/transmitter (UART) peripheral in the TMS320DM36x Digital Media System-on-Chip (DMSoC). The UART peripheral performs serial-to-parallel conversion on data received from a peripheral device, and parallel-to-serial conversion on data received from the CPU.

**SPRUFH3** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Inter-Integrated Circuit (I2C) Peripheral Users Guide*** This document describes the inter-integrated circuit (I2C) peripheral in the TMS320DM36x Digital Media System-on-Chip (DMSoC). The I2C peripheral provides an interface between the DMSoC and other devices compliant with the I2C-bus specification and connected by way of an I2C-bus.

**SPRUFH5** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Multimedia Card (MMC)/Secure Digital (SD) Card Controller Users Guide*** This document describes the multimedia card (MMC)/secure digital (SD) card controller in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFH6** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Pulse-Width Modulator (PWM) Users Guide*** This document describes the pulse-width modulator (PWM) peripheral in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFH7** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Real-Time Out (RTO) Controller Users Guide*** This document describes the Real Time Out (RTO) controller in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFH8** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) General-Purpose Input/Output (GPIO) Users Guide*** This document describes the general-purpose input/output (GPIO) peripheral in the TMS320DM36x Digital Media System-on-Chip (DMSoC). The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs.

**SPRUFH9** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Universal Serial Bus (USB) Controller Users Guide*** This document describes the universal serial bus (USB) controller in the TMS320DM36x 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.

**SPRUFI0** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Enhanced Direct Memory Access (EDMA) Controller Users Guide*** This document describes the operation of the enhanced direct memory access (EDMA3) controller in the TMS320DM36x Digital Media System-on-Chip (DMSoC). The EDMA controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the DMSoC.

**SPRUFI1** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Asynchronous External Memory Interface (EMIF) Users Guide*** This document describes the asynchronous external memory interface (EMIF) in the TMS320DM36x Digital Media System-on-Chip (DMSoC). The EMIF supports a glueless interface to a variety of external devices.

**SPRUFI2** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) DDR2/Mobile DDR (DDR2/mDDR) Memory Controller Users Guide*** This document describes the DDR2/mDDR memory controller in the TMS320DM36x Digital Media System-on-Chip (DMSoC). The DDR2/mDDR memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM and mobile DDR devices.

**SPRUFI3** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Multibuffered Serial Port Interface (McBSP) User's Guide*** This document describes the operation of the multibuffered serial host port interface in the TMS320DM36x Digital Media System-on-Chip (DMSoC). The primary audio modes that are supported by the McBSP are the AC97 and IIS modes. In addition to the primary audio modes, the McBSP supports general serial port receive and transmit operation.

**SPRUFI4** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Universal Host Port Interface (UHPI) User's Guide*** This document describes the operation of the universal host port interface in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFI5** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Ethernet Media Access Controller (EMAC) User's Guide*** This document describes the operation of the ethernet media access controllerface in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFI7** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Analog to Digital Converter (ADC) User's Guide*** This document describes the operation of the analog to digital conversion in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFI8** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Key Scan User's Guide*** This document describes the key scan peripheral in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

**SPRUFI9** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Voice Codec User's Guide*** This document describes the voice codec peripheral in the TMS320DM36x Digital Media System-on-Chip (DMSoC). This module can access ADC/DAC data with internal FIFO (Read FIFO/Write FIFO). The CPU communicates to the voice codec module using 32-bit-wide control registers accessible via the internal peripheral bus.

**SPRUFJ0** — ***TMS320DM36x Digital Media System-on-Chip (DMSoC) Power Management and Real-Time Clock Subsystem (PRTCSS) User's Guide*** This document provides a functional description of the Power Management and Real-Time Clock Subsystem (PRTCSS) in the TMS320DM36x Digital Media System-on-Chip (DMSoC) and PRTC interface (PRTCIF).

# DDR2/mDDR Memory Controller

## 1 Introduction

This document describes the DDR2/mDDR memory controller in the TMS320DM36x Digital Media System-on-Chip (DMSoC).

### *1.1 Purpose of the Peripheral*

The DDR2/mDDR memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM devices and standard Mobile DDR SDRAM devices. Memories types such as DDR1 SDRAM, SDR SDRAM, SBSRAM, and asynchronous memories are not supported. The DDR2/mDDR memory is the major memory location for program and data storage in the DM36x system.

### *1.2 Features*

The DDR2/mDDR memory controller supports the following features:

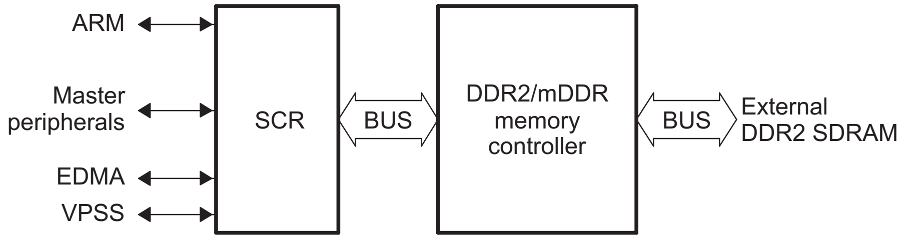
- JESD79D-2A standard compliant DDR2 SDRAM
- Standard compliant Mobile DDR
- 256 Mbyte memory space
- Data bus width of 16 bits
- CAS latencies:
  - DDR2: 2, 3, 4, and 5
  - mDDR: 2 and 3
- Internal banks:
  - DDR2: 1, 2, 4, and 8
  - mDDR: 1, 2, and 4
- Burst length: 8
- Burst type: sequential
- 1 CS signal
- Page sizes: 256, 512, 1024, and 2048
- SDRAM auto-initialization
- Self-refresh mode
- Partial array self-refresh (for mDDR)
- Power down mode
- Prioritized refresh
- Programmable refresh rate and backlog counter
- Programmable timing parameters
- Little endian

### *1.3 Functional Block Diagram*

The DDR2/mDDR memory controller is the main interface to external DDR2/mDDR memory.Figure 1 displays the general data paths to on-chip peripherals and external DDR2/mDDR SDRAM.

Master peripherals, EDMA, and the ARM processor can access the DDR2/mDDR memory controller through the switched central resource (SCR).

**Figure 1. Data Paths to DDR2/mDDR Memory Controller**




## 1.4 Supported Use Case Statement

The DDR2/mDDR memory controller supports JESD79D-2A DDR2/mDDR SDRAM memories and Mobile DDR SDRAM memories utilizing 16-bit of the DDR2/mDDR memory controller data bus. See Section 3 for more details.

## 1.5 Industry Standard(s) Compliance Statement

The DDR2/mDDR memory controller is compliant with the JESD79D-2A DDR2/mDDR SDRAM standard and Mobile DDR standard with the following exception:

- On Die Termination (ODT). The DDR2/mDDR memory controller does not include any on-die terminating resistors. Furthermore, the on-die terminating resistors of the DDR2/mDDR SDRAM device must be disabled by tying the ODT input pin of the DDR2/mDDR SDRAM to ground.

# 2 Peripheral Architecture

This section describes the architecture of the DDR2/mDDR memory controller as well as how it is structured and how it works within the context of the system-on-a-chip. The DDR2/mDDR memory controller can gluelessly interface to most standard DDR2/mDDR SDRAM devices and supports such features as self-refresh mode and prioritized refresh. In addition, it provides flexibility through programmable parameters such as the refresh rate, CAS latency, and many SDRAM timing parameters. The following sections include details on how to interface and properly configure the DDR2/mDDR memory controller to perform read and write operations to externally-connected DDR2/mDDR SDRAM devices. Also, Section 3 provides a detailed example of interfacing the DDR2/mDDR memory controller to a common DDR2/mDDR SDRAM device.

## *2.1 Clock Control*

The DDR2/mDDR memory controller receives two input clocks from internal clock sources, VCLK and X2_CLK (Figure 2). VCLK is a divided-down version of the PLL1 clock. VCLK is the configuration bus/peripheral system interfaces clock (PLLC1SYSCLK4). X2_CLK is selectable from PLL2/PLL1. For information on the selection of X2_CLK, refer to the *TMS320DM365 Digital Media System-on-Chip (DMSoC) ARM Subsystem Reference Guide* (SPRUFG5). X2_CLK should be configured to clock at the frequency of the desired data rate; it should operate at twice the frequency of the desired DDR2/mDDR memory clock. DDR_CLK and $\overline{\text{DDR\_CLK}}$ are the two output clocks of the DDR2/mDDR memory controller providing the interface clock to the DDR2/mDDR SDRAM memory. These two clocks operate at a frequency of X2_CLK/2.

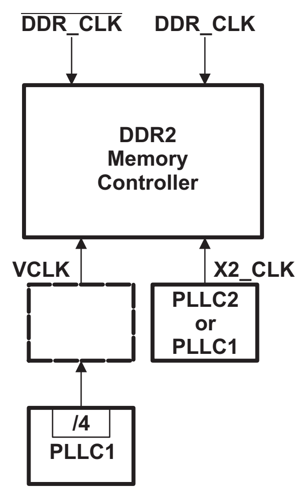
### 2.1.1 Clock Source

VCLK and X2_CLK are sourced from two independent PLLs (Figure 2). VCLK is sourced from PLL controller 1 (PLLC1) and X2_CLK is selectable from PLL Controller 1 or PLL Controller 2.

VCLK is clocked at a fixed divider ratio of PLL1. This divider is fixed at 4, that is, VCLK is clocked at a frequency of PLL1/4.

The clock from PLLC2/PLLC1 is not divided before reaching X2_CLK. PLLC2/PLLC1 should be configured to supply X2_CLK at the desired frequency. For example, if a 138-MHz DDR2/mDDR interface clock (DDR_CLK) is desired, then PLLC2/PLLC1 must be configured to generate a 276-MHz clock on X2_CLK.

**Figure 2. DDR2/mDDR Memory Controller Clock Block Diagram**



### 2.1.2 Clock Configuration

The frequency of X2_CLK is configured by selecting the appropriate PLL multiplier. The PLL multiplier is selected by programming registers within PLLC2/PLLC1. For information on programming the PLL controllers, refer to the *TMS320DM365 Digital Media System-on-Chip (DMSoC) ARM Subsystem Reference Guide* (SPRUFG5). For information on supported clock frequencies refer to the *TMS320DM365 Digital Media System-on-Chip Data Manual* (SPRS457).

**Note:** PLLC2/PLLC1 should be configured and a stable clock present on X2_CLK before releasing the DDR2/mDDR memory controller from reset.

### 2.1.3 DDR2/mDDR Memory Controller Internal Clock Domains

There are two clock domains within the DDR2/mDDR memory controller. The two clock domains are driven by VCLK and a divided-down by 2 version of X2_CLK called MCLK. The command FIFO, write FIFO, and read FIFO described in Section 2.7 are all on the VCLK domain. From this, VCLK drives the interface to the peripheral bus.

The MCLK domain consists of the DDR2/mDDR memory controller state machine and memory-mapped registers. This clock domain is clocked at the rate of the external DDR2/mDDR memory, X2_CLK/2.

To conserve power within the DDR2/mDDR memory controller, VCLK, MCLK, and X2_CLK may be stopped. See Section 2.17 for proper clock stop procedures.

## *2.2 Memory Map*

See the device-specific data manual for information describing the device memory-map.

## *2.3 Signal Descriptions*

The DDR2/mDDR memory controller signals are shown in Figure 3 and described in Table 1. The following features are included:

- The maximum data bus is 16-bits wide.
- The address bus is 14-bits wide with an additional three bank address pins.
- Two differential output clocks driven by internal clock sources.
- Command signals: Row and column address strobe, write enable strobe, data strobe, and data mask.
- One chip select signal and one clock enable signal.

**Figure 3. DDR2/mDDR Memory Controller Signals**

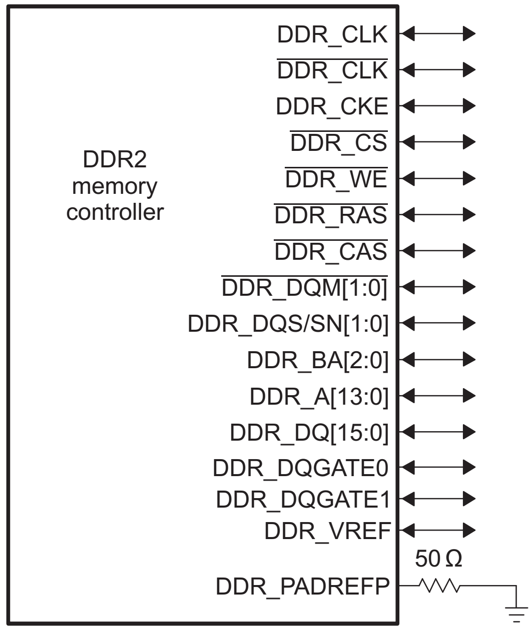


**Table 1. DDR2/mDDR Memory Controller Signal Descriptions**

| Pin | Type[(1)] | Description |
|---|---|---|
| DDR_CLK, $\overline{\text{DDR\_CLK}}$ | O/Z | **Clock:** Differential clock outputs. |
| DDR_CKE | O/Z | **Clock enable:** Active high. |
| $\overline{\text{DDR\_CS}}$ | O/Z | **Chip select:** Active low. |
| $\overline{\text{DDR\_WE}}$ | O/Z | **Write enable strobe:** Active low, command output. |
| $\overline{\text{DDR\_RAS}}$ | O/Z | **Row address strobe:** Active low, command output. |
| $\overline{\text{DDR\_CAS}}$ | O/Z | **Column address strobe:** Active low, command output. |
| DDR_DQM[1:0] | O/Z | **Data mask**: Output mask signal for write data. |
| DDR_DQS[1:0] | I/O/Z | **Data strobe:** Active high, bi-directional signals. Output with write data, input with read data. |
| DDR_BA[2:0] | O/Z | **Bank select:** Output, defining which bank a given command is applied. |
| DDR_A[13:0] | O/Z | **Address:** Address bus. |
| DDR_DQ[15:0] | I/O/Z | **Data:** Bi-directional data bus. Input for read data, output for write data. |
| DDR_DQGATE0, DDR_DQGATE1 | I/O/Z | **Loopback delay signals:** Loopback signals for timing adjustment (DQS gating). Route from DDR_DQGATE0 to DDR device and back to DDR_DQGATE1 with same constraints as used for DDR clock and data. |
| DDR_DQSN[1:0] | I/O/Z | **Complementary data strobe:** bi-directional signals. Output with write data, input with read data. |
| DDR_PADREFP | I/O/Z | **Output drive strength reference:** Reference output for drive strength calibration of N and P channel outputs. Tie to ground via 50 ohm .5% tolerance 1/16th watt resistor (49.9 ohm .5% tolerance is acceptable). |
| DDR_VREF | pwr | **Voltage reference input:** Voltage reference input for the SSTL_18 I/O buffers. Note even in the case of mDDR an external resistor divider connected to this pin is necessary. |

(1) Legend: I = input, O = Output, Z = high impedance, pwr = power, gnd = ground.

### 2.4 *Protocol Description(s)*

The DDR2/mDDR memory controller supports the DDR2/mDDR SDRAM commands listed in Table 2. Table 3 shows the signal truth table for the DDR2/mDDR SDRAM commands.

**Table 2. DDR2/mDDR SDRAM Commands**

| Command | Function |
|---|---|
| ACTV | Activates the selected bank and row. |
| DCAB | Precharge all command. Deactivates (precharges) all banks. |
| DEAC | Precharge single command. Deactivates (precharges) a single bank. |
| DESEL | Device Deselect. |
| EMRS | Extended Mode Register set. Allows altering the contents of the mode register. |
| MRS | Mode register set. Allows altering the contents of the mode register. |
| NOP | No operation. |
| Power Down | Power down mode. |
| READ | Inputs the starting column address and begins the read operation. |
| READ with autoprecharge | Inputs the starting column address and begins the read operation. The read operation is followed by a precharge. |
| REFR | Autorefresh cycle. |
| SLFREFR | Self-refresh mode. |
| WRT | Inputs the starting column address and begins the write operation. |
| WRT with autoprecharge | Inputs the starting column address and begins the write operation. The write operation is followed by a precharge. |

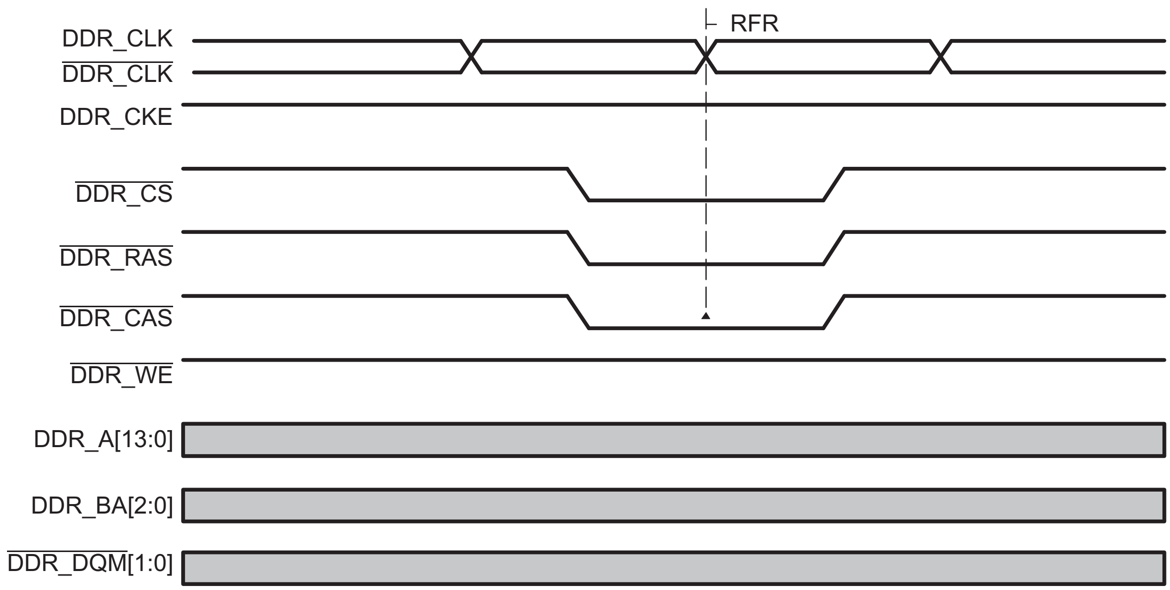
## Table 3. Truth Table for DDR2/mDDR SDRAM Commands

| DDR2/mDDR SDRAM: | CKE | | $\overline{\text{CS}}$ | $\overline{\text{RAS}}$ | $\overline{\text{CAS}}$ | $\overline{\text{WE}}$ | BA[2:0] | A[13:11, 9:0] | A10 |
|---|---|---|---|---|---|---|---|---|---|
| **DDR2/mDDR memory controller:** | **DDR_CKE** Previous Cycles | **DDR_CKE** Current Cycle | $\overline{\text{DDR\_CE}}$ | $\overline{\text{DDR\_RAS}}$ | $\overline{\text{DDR\_CAS}}$ | $\overline{\text{DDR\_WE}}$ | **DDR_BA[2:0]** | **DDR_A[13:11, 9:0]** | **DDR_A[10]** |
| ACTV | H | H | L | L | H | H | Bank | Row Address | |
| DCAB | H | H | L | L | H | L | X | X | H |
| DEAC | H | H | L | L | H | L | Bank | X | L |
| MRS | H | H | L | L | L | L | BA | OP Code | |
| EMRS | H | H | L | L | L | L | BA | OP Code | |
| READ | H | H | L | H | L | H | BA | Column Address | L |
| READ with precharge | H | H | L | H | L | H | BA | Column Address | H |
| WRT | H | H | L | H | L | L | BA | Column Address | L |
| WRT with precharge | H | H | L | H | L | L | BA | Column Address | H |
| REFR | H | H | L | L | L | H | X | X | X |
| SLFREFR entry | H | L | L | L | L | H | X | X | X |
| SLFREFR exit | L | H | H | X | X | X | X | X | X |
| | | | L | H | H | H | X | X | X |
| NOP | H | X | L | H | H | H | X | X | X |
| DESEL | H | X | H | X | X | X | X | X | X |
| Power Down entry | H | L | H | X | X | X | X | X | X |
| | | | L | H | H | H | X | X | X |
| Power Down exit | L | H | H | X | X | X | X | X | X |
| | | | L | H | H | H | X | X | X |

### 2.4.1 Refresh Mode

The DDR2/mDDR memory controller issues refresh commands to the DDR2/mDDR SDRAM memory (Figure 4). REFR is automatically preceded by a DCAB command, ensuring the deactivation of all CE spaces and banks selected. Following the DCAB command, the DDR2/mDDR memory controller begins performing refreshes at a rate defined by the refresh rate (RR) bit in the SDRAM refresh control register (SDRCR). Page information is always invalid before and after a REFR command; thus, a refresh cycle always forces a page miss. This type of refresh cycle is often called autorefresh. Autorefresh commands may not be disabled within the DDR2/mDDR memory controller. See Section 2.8 for more details on REFR command scheduling.

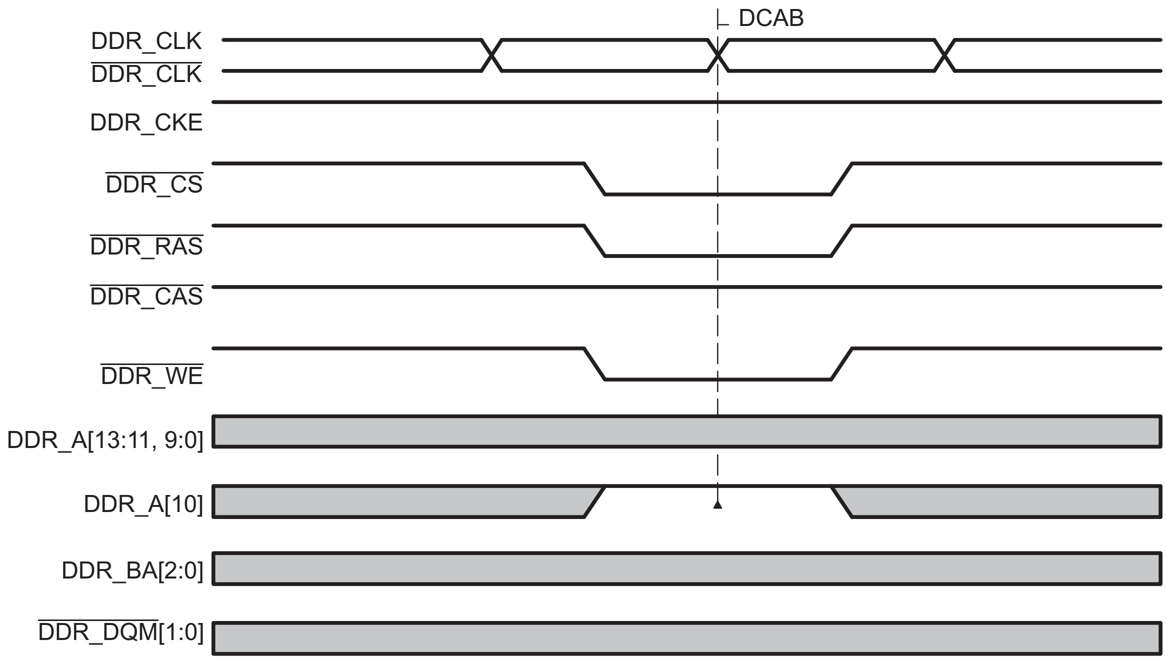
**Figure 4. Refresh Command**



### 2.4.2 Deactivation (DCAB and DEAC)

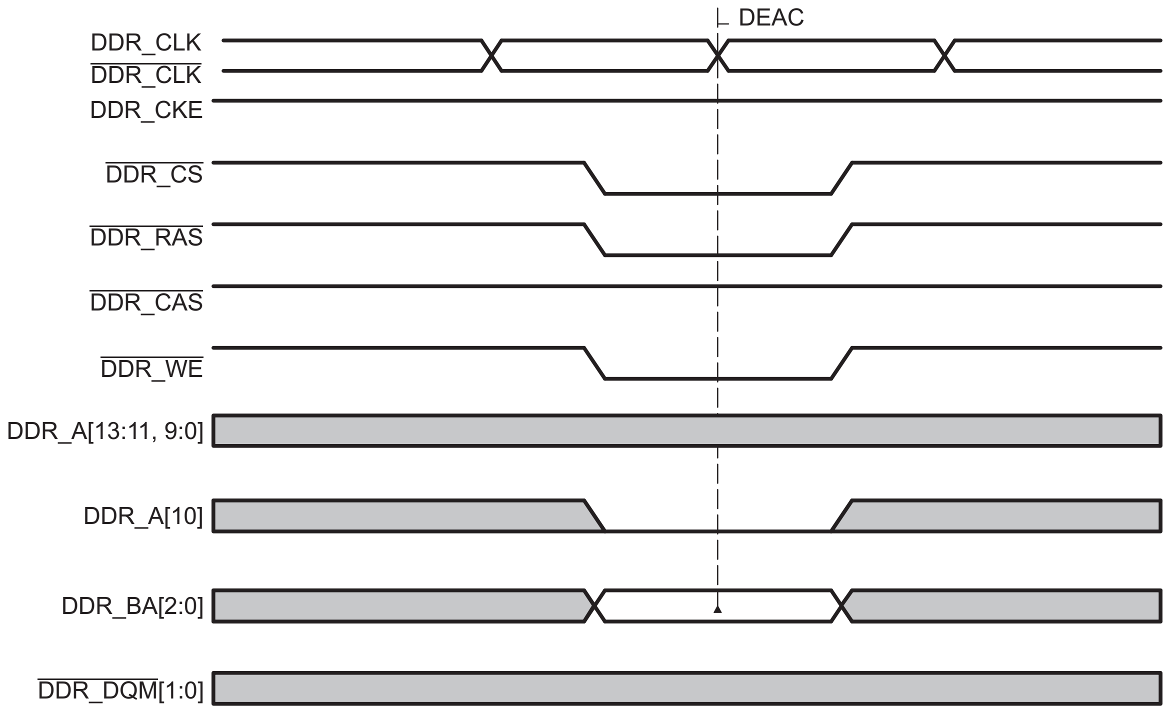
The precharge all banks command (DCAB) is performed after a reset to the DDR2/mDDR memory controller or following the initialization sequence. DDR2/mDDR SDRAMs also require this cycle prior to a refresh (REFR) and mode set register commands (MRS and EMRS). During a DCAB command, DDR_A[10] is driven high to ensure the deactivation of all banks. Figure 5 shows the timing diagram for a DCAB command.

**Figure 5. DCAB Command**



The DEAC command closes a single bank of memory specified by the bank select signals. Figure 6 shows the timings diagram for a DEAC command.

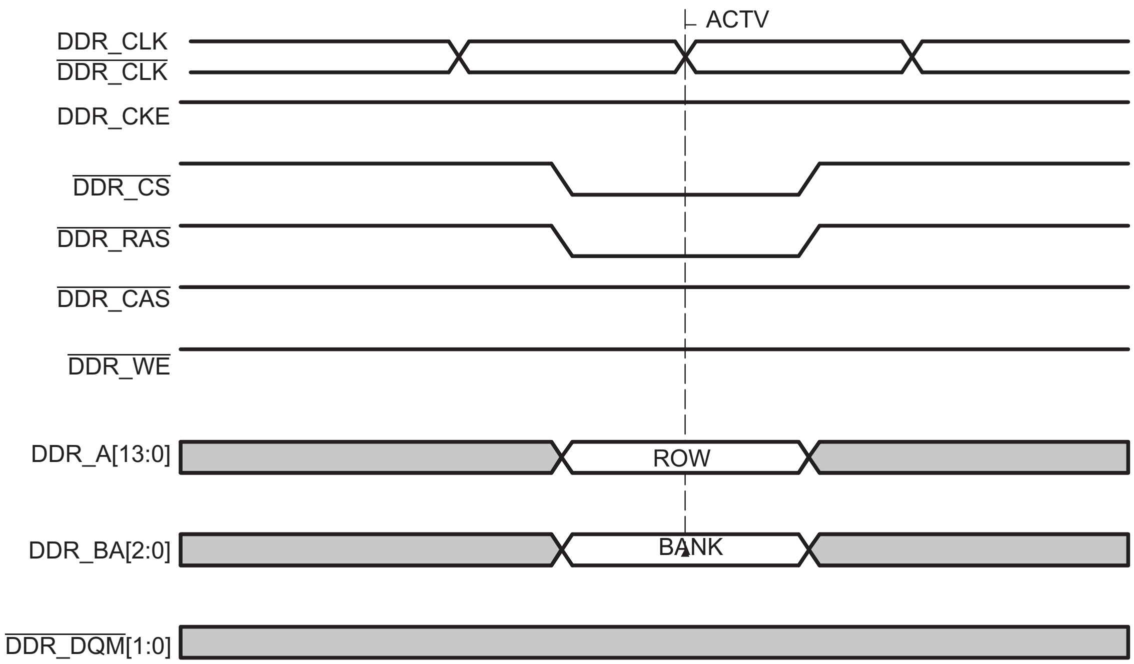
**Figure 6. DEAC Command**



### 2.4.3 Activation (ACTV)

The DDR2/mDDR memory controller automatically issues the activate (ACTV) command before a read or write to a closed row of memory. The ACTV command opens a row of memory, allowing future accesses (reads or writes) with minimum latency. The value of DDR_BA[2:0] selects the bank and the value of DDR_A[13:0] selects the row. When the DDR2/mDDR memory controller issues an ACTV command, a delay of $t_{RCD}$ is incurred before a read or write command is issued. Figure 7 shows an example of an ACTV command. Reads or writes to the currently active row and bank of memory can achieve much higher throughput than reads or writes to random areas because every time a new row is accessed, the ACTV command must be issued and a delay of $t_{RCD}$ incurred.

**Figure 7. ACTV Command**

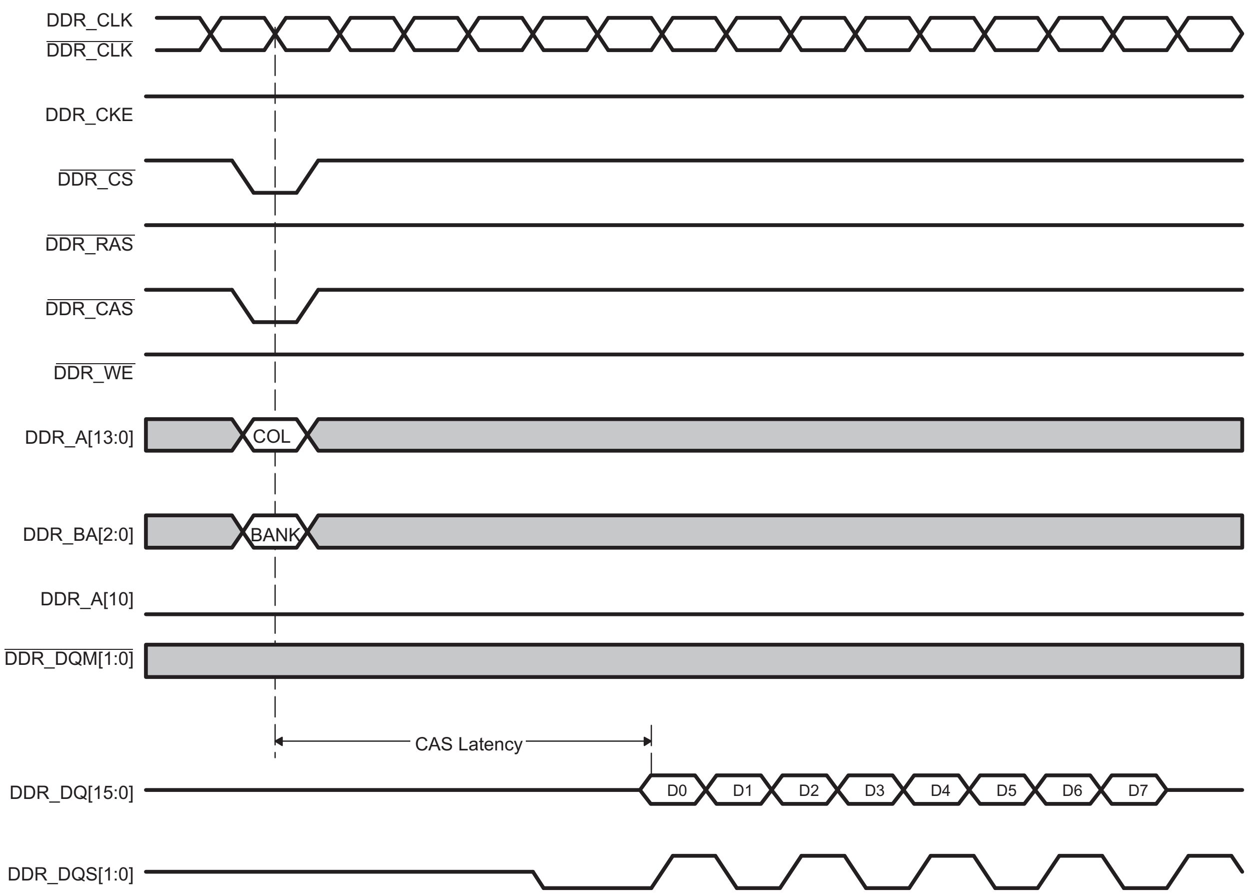


### 2.4.4 READ Command

Figure 8 shows the DDR2/mDDR memory controller performing a read burst from DDR2/mDDR SDRAM. The READ command initiates a burst read operation to an active row. During the READ command, DDR_CAS drives low, DDR_WE and DDR_RAS remain high, the column address is driven on DDR_A[13:0], and the bank address is driven on DDR_BA[2:0].

The DDR2/mDDR memory controller uses a burst length of 8, and has a programmable CAS latency of 2, 3, 4, or 5. The CAS latency is three cycles in Figure 8. Read latency is equal to CAS latency plus additive latency. The DDR2/mDDR memory controller always configures the memory to have an additive latency of 0, so read latency equals CAS latency. Since the default burst size is 8, the DDR2/mDDR memory controller returns 8 pieces of data for every read command. If additional accesses are not pending to the DDR2/mDDR memory controller, the read burst completes and the unneeded data is disregarded. If additional accesses are pending, depending on the scheduling result, the DDR2/mDDR memory controller can terminate the read burst and start a new read burst. Furthermore, the DDR2/mDDR memory controller does not issue a DAB/DEAC command until page information becomes invalid.

**Figure 8. DDR2/mDDR READ Command**




(1) Single-ended DQS is used.

### 2.4.5 Write (WRT) Command

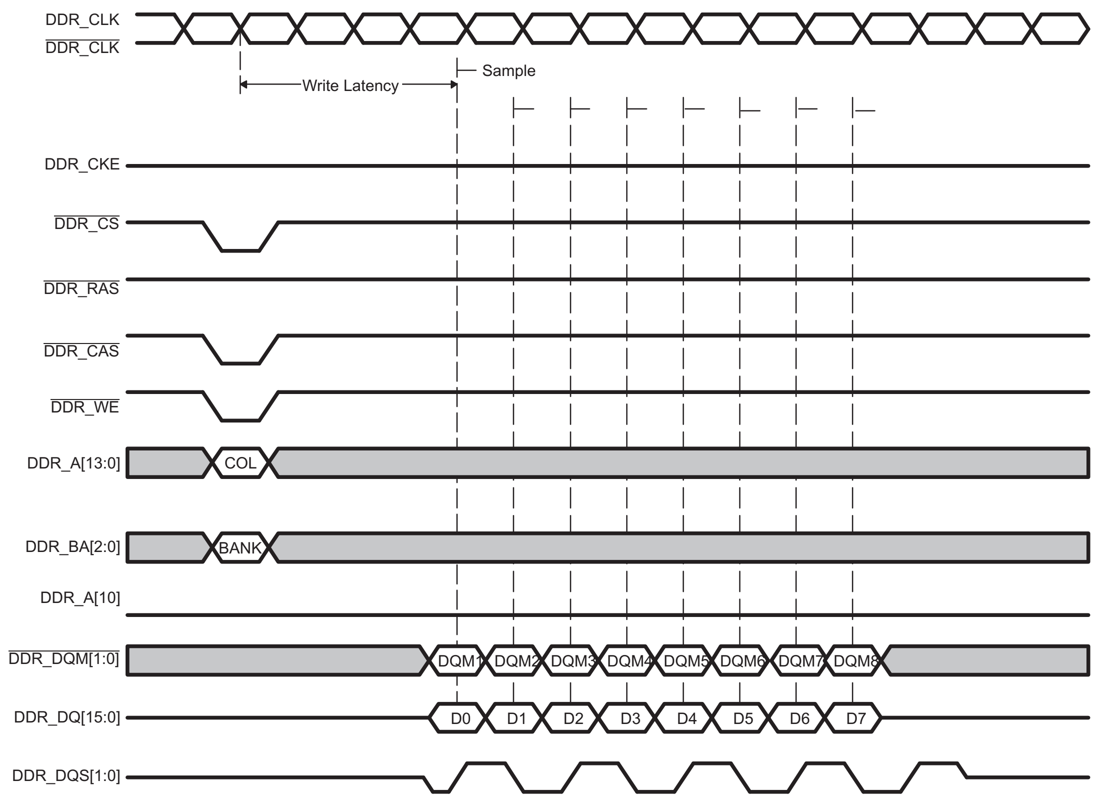
Prior to a WRT command, the desired bank and row are activated by the ACTV command. Following the WRT command, a write latency is incurred. For DDR2, write latency is equal to CAS latency minus 1 cycles. For mDDR, write latency is equal to 1 cycle, always. All writes have a burst length of 8. The use of the DDR_DQM outputs allows byte and halfword writes to be executed. Figure 9 shows the timing for a DDR2 write on the DDR2/mDDR memory controller.

If the transfer request is for less than 8 words, depending on the scheduling result and the pending commands, the DDR2/mDDR memory controller can:

- Mask out the additional data using DDR_DQM outputs
- Terminate the write burst and start a new write burst

The DDR2/mDDR memory controller does not perform the DEAC command until page information becomes invalid.

**Figure 9. DDR2/mDDR WRT Command(1)**




(1) This diagrams shows write latency for DDR2. For mDDR, write latency is always equal to 1 cycle.

(2) Single-ended DQS is used.

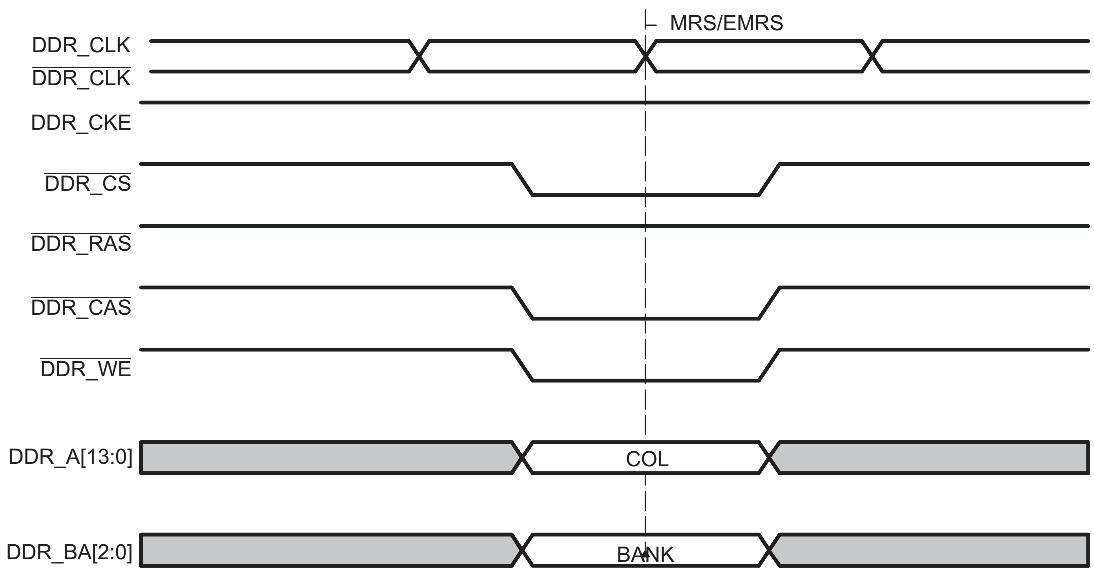
### 2.4.6 Mode Register Set (MRS and EMRS)

DDR2/mDDR SDRAM contains mode and extended mode registers that configure the DDR2/mDDR memory for operation. These registers control burst type, burst length, CAS latency, DLL enable/disable (on DDR2/mDDR device), single-ended strobe,differential strobe etc.

The DDR2/mDDR memory controller programs the mode and extended mode registers of the DDR2/mDDR memory by issuing MRS and EMRS commands. When the MRS or EMRS command is executed, the value on DDR_BA[2:0] selects the mode register to be written and the data on DDR_A[13:0] is loaded into the register. Figure 10 shows the timing for an MRS and EMRS command.

The DDR2/mDDR memory controller only issues MRS and EMRS commands during the DDR2/mDDR memory controller initialization sequence. See Section 2.14 for more information.

**Figure 10. DDR2/mDDR MRS and EMRS Command**




## 2.5 Memory Width and Byte Alignment

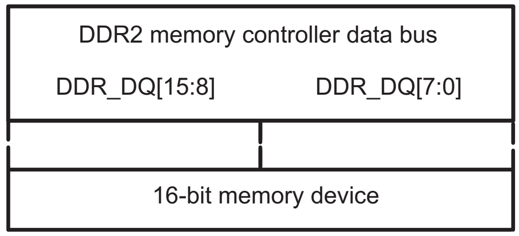
The DDR2/mDDR memory controller supports memory widths of 16 bits. Table 4 summarizes the addressable memory ranges on the DDR2/mDDR memory controller. Only little-endian format is supported.

Figure 11 shows the byte lanes used on the DDR2/mDDR memory controller. The external memory is always right aligned on the data bus.

**Table 4. Addressable Memory Ranges**

| Memory Width | Maximum addressable bytes per CS space | Description |
|---|---|---|
| ×16 | 256 Mbytes | Halfword address |

**Figure 11. Byte Alignment**




## *2.6 Address Mapping*

The memory controller views the DDR2/mDDR SDRAM device as one continuous block of memory. The memory controller receives memory access requests with a 32-bit logical address, and it uses the logical address to generate a row, column, and bank address for accessing the DDR2/mDDR SDRAM device.

The memory controller supports two address mapping schemes: normal address mapping and special address mapping. Special address mapping is typically used only with mDDR devices using partial array self-refresh.

When the internal bank position (IBANKPOS) bit in the SDRAM configuration register (SDCR) is cleared, the memory controller operates with normal address mapping. In this case, the number of column and bank address bits is determined by the IBANK and PAGESIZE fields in SDCR. The number of row address bits is determined by the number of valid address pins for the device and does not need to be set in a register.

When IBANKPOS is set to 1, the memory controller operates with special address mapping. In this case, the number of column, row, and bank address bits is determined by the PAGESIZE, ROWSIZE, and IBANK fields. The ROWSIZE field is in the SDRAM configuration 2 register (SDCR2). See Table 5 for a descriptions of these bit fields.

**Table 5. Configuration Register Fields for Address Mapping**

| Bit Field | Bit Value | Bit Description |
|---|---|---|
| IBANK | | Defines the number of internal banks in the external DDR2/mDDR memory. |
| | 0 | 1 bank |
| | 1h | 2 banks |
| | 2h | 4 banks |
| | 3h | 8 banks |
| PAGESIZE | | Defines the page size of each page in the external DDR2/mDDR memory. |
| | 0 | 256 words (requires 8 column address bits) |
| | 1h | 512 words (requires 9 column address bits) |
| | 2h | 1024 words (requires 10 column address bits) |
| | 3h | 2048 words (requires 11 column address bits) |
| ROWSIZE | | Defines the row size of each row in the external DDR2/mDDR memory |
| | 0 | 512 (requires 9 row address bits) |
| | 1h | 1024 (requires 10 row address bits) |
| | 2h | 2048 (requires 11 row address bits) |
| | 3h | 4096 (requires 12 row address bits) |
| | 4h | 8192 (requires 13 row address bits) |
| | 5h | 16384 (requires 14 row address bits) |

### 2.6.1 Normal Address Mapping (IBANKPOS=0)

As stated in Table 5, the IBANK and PAGESIZE fields of SDCR control the mapping of the logical, source address of the DDR2/mDDR memory controller to the DDR2/mDDR SDRAM row, column, and bank address bits. The DDR2/mDDR memory controller logical address always contains up to 14 row address bits, whereas the number of column and bank bits are determined by the IBANK and PAGESIZE fields. Table 6 show how the logical address bits map to the DDR2/mDDR SDRAM row, column, and bank bits for combinations of IBANK and PAGESIZE values. The same DDR2/mDDR memory controller pins provide the row and column address to the DDR2/mDDR SDRAM, thus the DDR2/mDDR memory controller appropriately shifts the address during row and column address selection.

Figure 12 shows how this address-mapping scheme organizes the DDR2/mDDR SDRAM rows, columns, and banks into the device memory map. Note that during a linear access, the DDR2/mDDR memory controller increments the column address as the logical address increments. When the DDR2/mDDR memory controller reaches a page/row boundary, it moves onto the same page/row in the next bank. This movement continues until the same page has been accessed in all banks. To the DDR2/mDDR SDRAM, this process looks as shown in Figure 13.

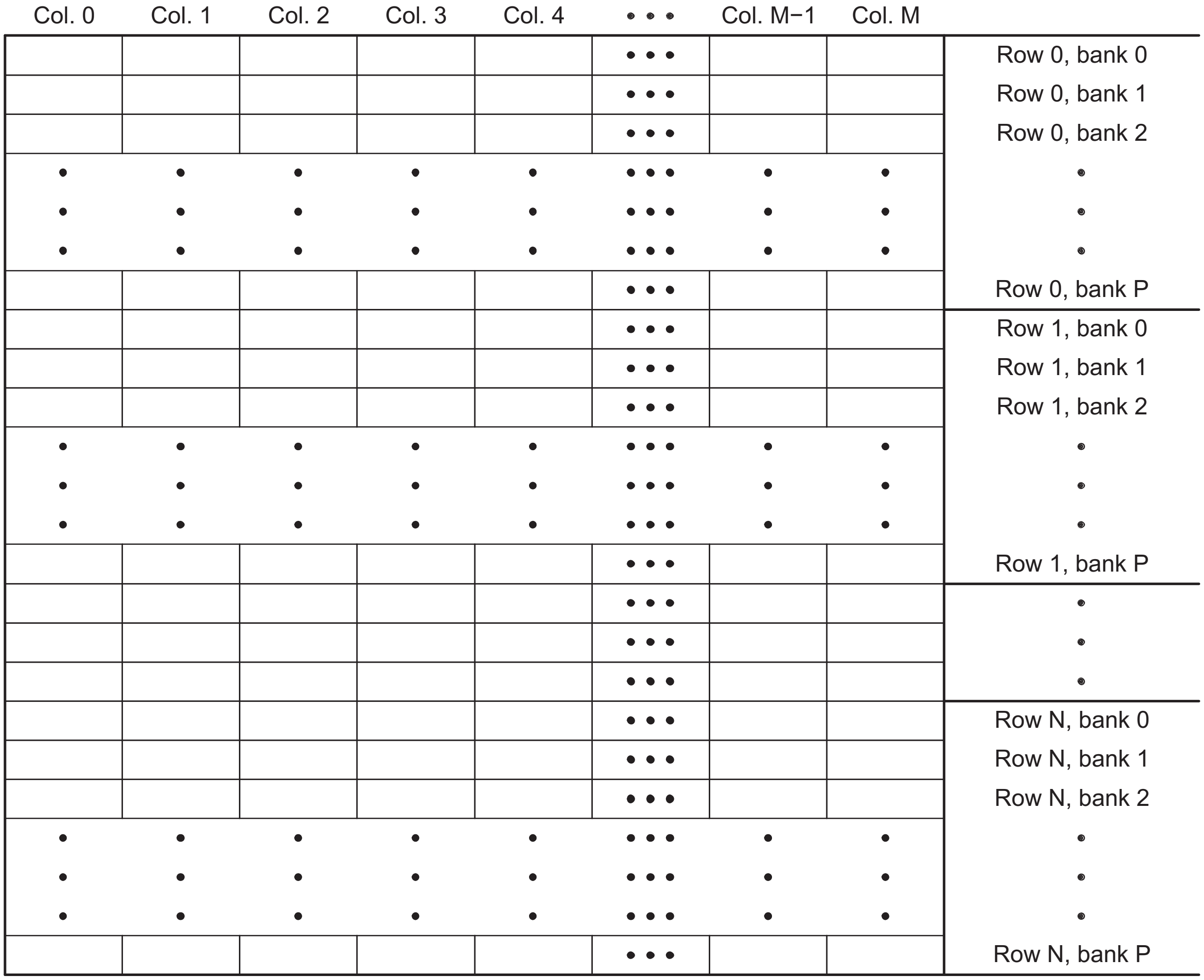
By traversing across banks while remaining on the same row/page, the DDR2/mDDR memory controller maximizes the number of activated banks for a linear access. This results in the maximum number of open pages when performing a linear access being equal to the number of banks. Note that the DDR2/mDDR memory controller never opens more than one page per bank.

Ending the current access is not a condition that forces the active DDR2/mDDR SDRAM row to be closed. The DDR2/mDDR memory controller leaves the active row open until it becomes necessary to close it. This decreases the deactivate-reactivate overhead.

**Table 6. Logical Address-to-DDR2/mDDR SDRAM Address Map for 16-bit SDRAM**

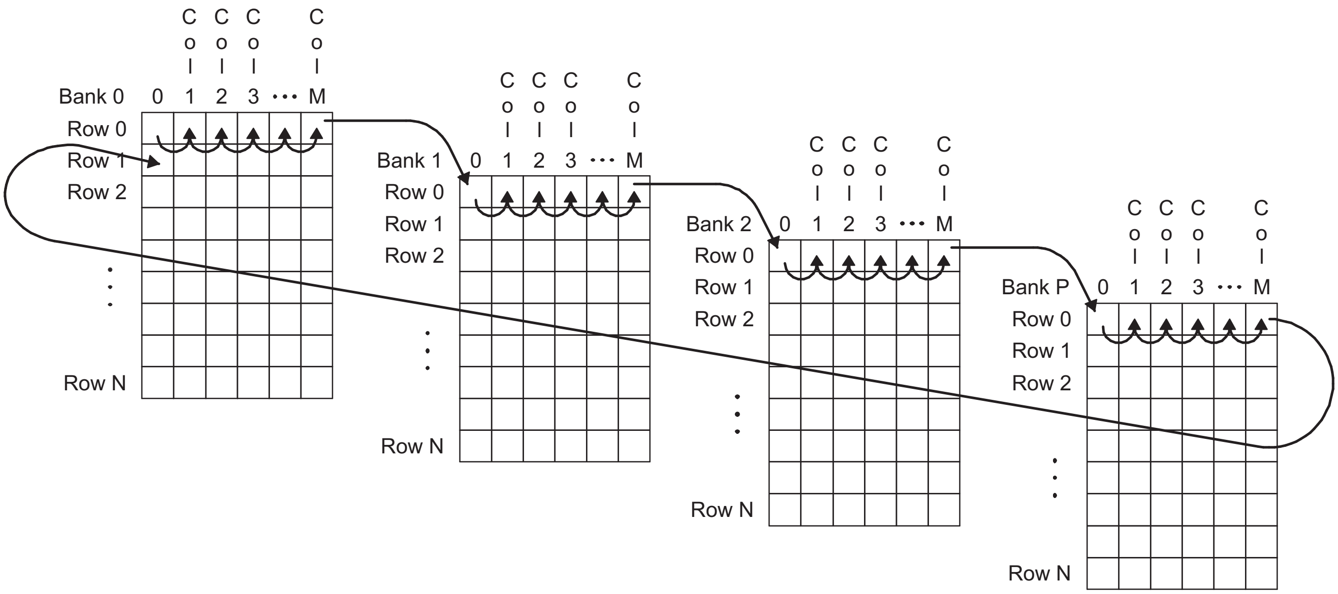
| SDCR Bit | | Logical Address | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **IBANK** | **PAGESIZE** | **31** | **30** | **29** | **28** | **27** | **26** | **25** | **24** | **23** | **22** | **21:15** | **14** | **13** | **12** | **11** | **10** | **9** | **8:1** | **0** |
| 0 | 0 | - | | | | | | | | | nrb=14 | | | | | | | | ncb=8 | |
| 1 | 0 | - | | | | | | | | nrb=14 | | | | | | | | nbb=1 | ncb=8 | |
| 2h | 0 | - | | | | | | | nrb=14 | | | | | | | | nbb=2 | | ncb=8 | |
| 3h | 0 | - | | | | | | nrb=14 | | | | | | | | nbb=3 | | | ncb=8 | |
| 0 | 1 | - | | | | | | | | nrb=14 | | | | | | | | ncb=9 | | |
| 1 | 1 | - | | | | | | | nrb=14 | | | | | | | | nbb=1 | ncb=9 | | |
| 2h | 1 | - | | | | | | nrb=14 | | | | | | | | nbb=2 | | ncb=9 | | |
| 3h | 1 | - | | | | | nrb=14 | | | | | | | | nbb=3 | | | ncb=9 | | |
| 0 | 2h | - | | | | | | | nrb=14 | | | | | | | | ncb=10 | | | |
| 1 | 2h | - | | | | | | nrb=14 | | | | | | | | nbb=1 | ncb=10 | | | |
| 2h | 2h | - | | | | | nrb=14 | | | | | | | | nbb=2 | | ncb=10 | | | |
| 3h | 2h | - | | | | nrb=14 | | | | | | | | nbb=3 | | | ncb=10 | | | |
| 0 | 3h | - | | | | | | nrb=14 | | | | | | | | ncb=11 | | | | |
| 1 | 3h | - | | | | | nrb=14 | | | | | | | | nbb=1 | ncb=11 | | | | |
| 2h | 3h | - | | | | nrb=14 | | | | | | | | nbb=2 | | ncb=11 | | | | |
| 3h | 3h | - | | | nrb=14 | | | | | | | | nbb=3 | | | ncb=11 | | | | |

**Figure 12. Logical Address-to-DDR2/mDDR SDRAM Address Map**




M is number of columns (as determined by PAGESIZE) minus 1, P is number of banks (as determined by IBANK) minus 1, and N is number of rows (as determined by both PAGESIZE and IBANK) minus 1.

**Figure 13. DDR2/mDDR SDRAM Column, Row, and Bank Access**




M is number of columns (as determined by PAGESIZE) minus 1, P is number of banks (as determined by IBANK) minus 1, and N is number of rows (as determined by both PAGESIZE and IBANK) minus 1.

### 2.6.2 Special Address Mapping (IBANKPOS=1)

When the internal bank position (IBANKPOS) bit is set to 1, the PAGESIZE, ROWSIZE, and IBANK fields control the mapping of the logical source address of the memory controller to the column, row, and bank address bits of the SDRAM device. Table 7 shows which source address bits map to the SDRAM column, row, and bank address bits for all combinations of PAGESIZE, ROWSIZE, and IBANK.

When IBANKPOS is set to 1, the effect of the address-mapping scheme is that as the source address increments across an SDRAM page boundary, the memory controller proceeds to the next page in the same bank. This movement along the same bank continues until all the pages have been accessed in the same bank. The memory controller then proceeds to the next bank in the device. This sequence is shown in Figure 14 and Figure 15.

Since, in this address mapping scheme, the memory controller can keep only one bank open, this scheme is lower in performance than the case when IBANKPOS is cleared to 0. Therefore, this case is only recommended to be used with Partial Array Self-refresh for mDDR SDRAM where performance may be traded-off for power savings.

**Table 7. Address Mapping Diagram for 16-Bit SDRAM (IBANKPOS=1)**

| 31 | Source Address | 1 |
|---|---|---|
| **Bank Address** | **Row Address** | **Column Address** |
| Number of bank bits is defined by IBANK nbb = 1, 2, or 3 | Number of row bits is defined by ROWSIZE: nrb = 9, 10, 11, 12, 13, or 14 | Number of column bits is defined by PAGESIZE: ncb = 8,9,10, or 11 |

**Figure 14. Address Mapping Diagram (IBANKPOS=1)**

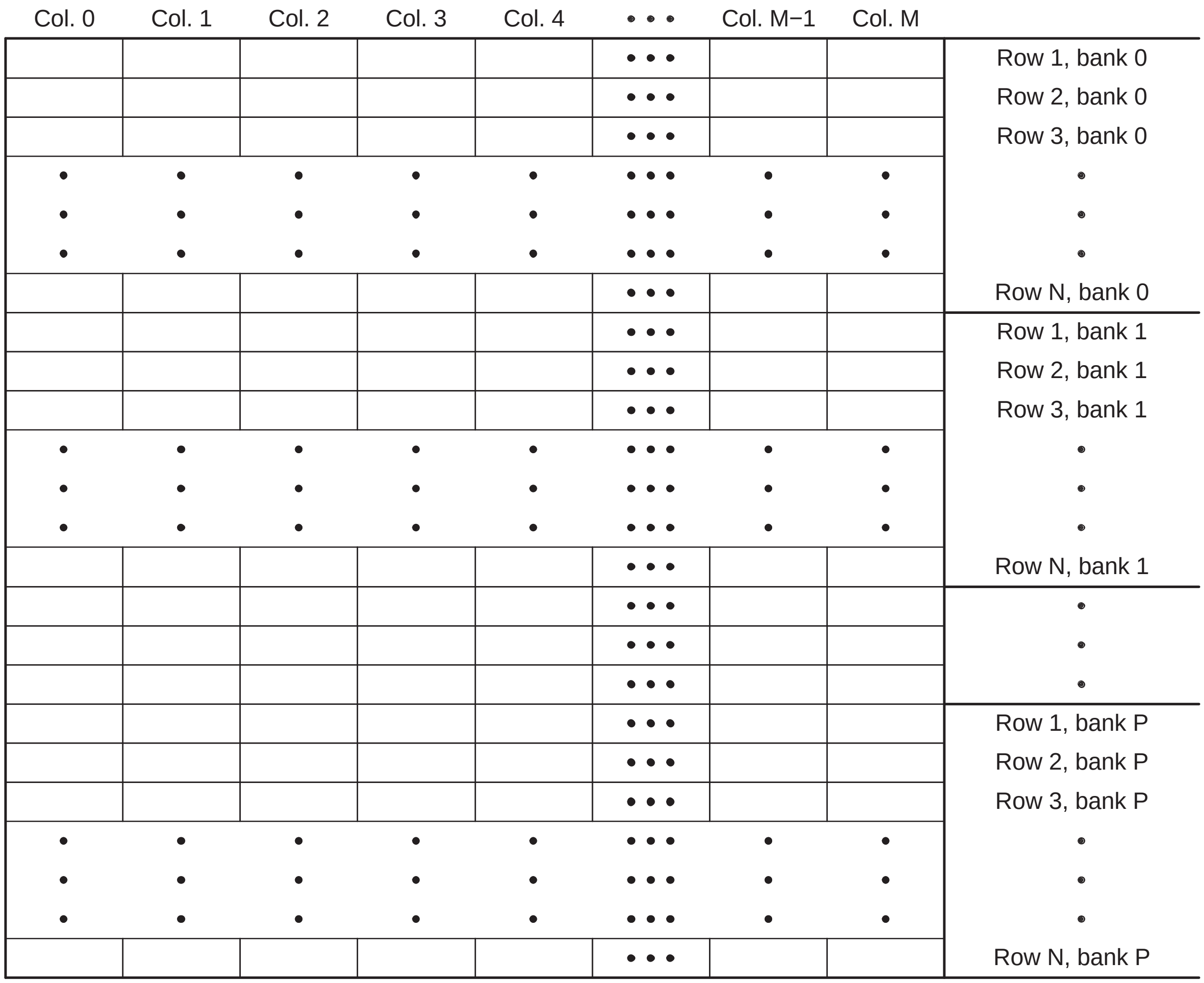
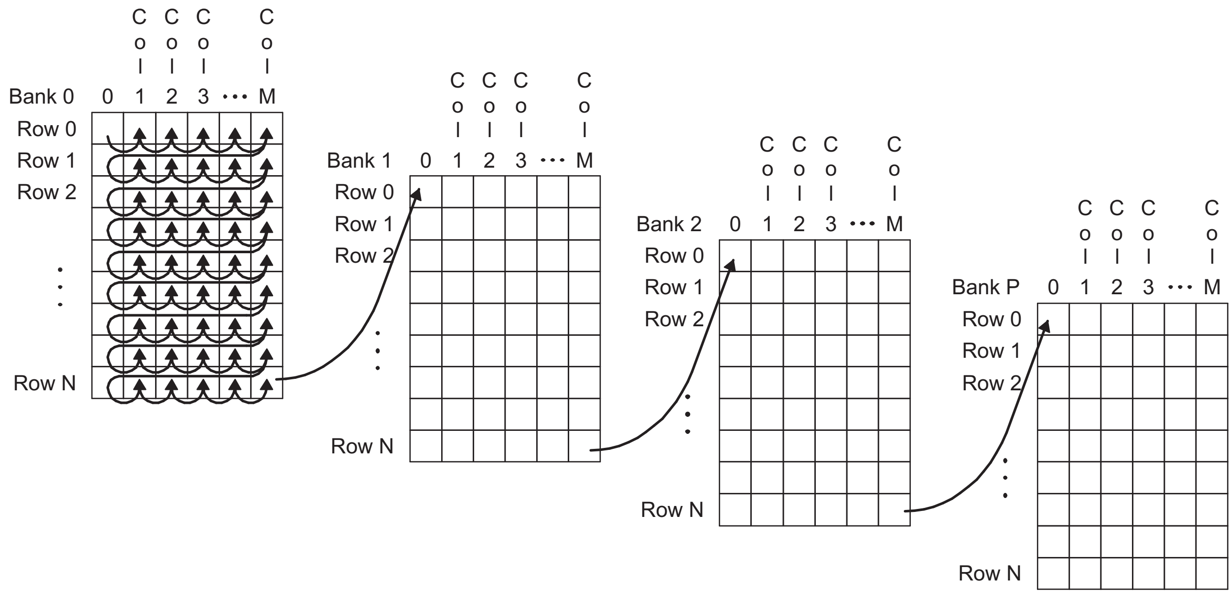


**Figure 15. SDRAM Column, Row, Bank Access (IBANKPOS=1)**




M is number of columns (as determined by PAGESIZE) minus 1, P is number of banks (as determined by IBANK) minus 1, and N is number of rows (as determined by ROWSIZE) minus 1.

## 2.7 DDR2/mDDR Memory Controller Interface

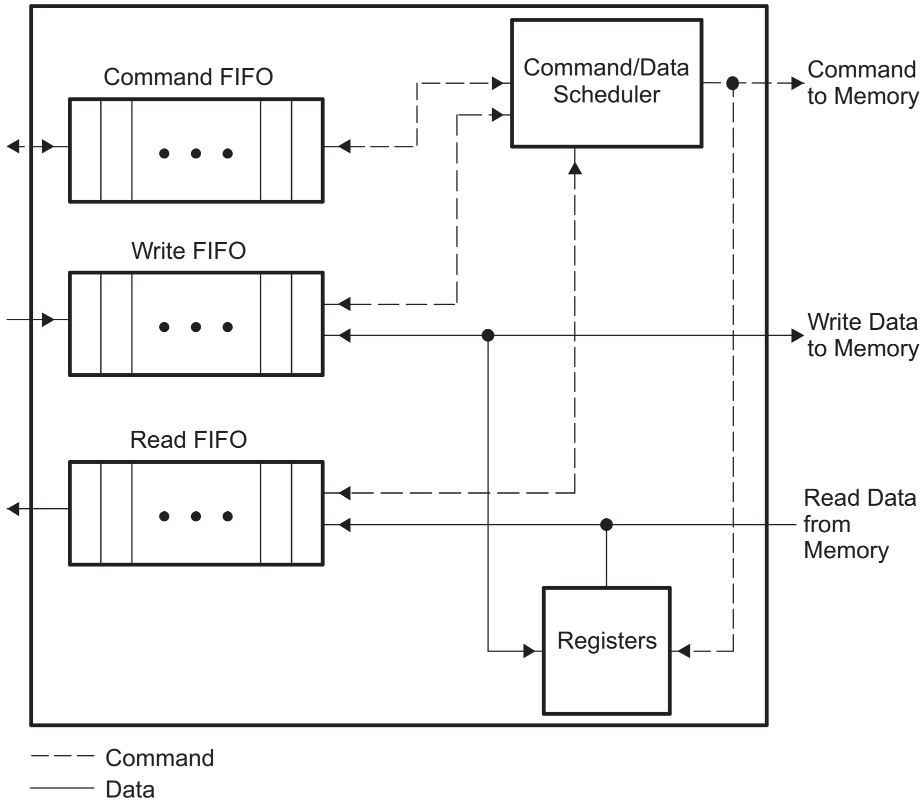
To move data efficiently from on-chip resources to external DDR2/mDDR SDRAM memory, the DDR2/mDDR memory controller makes use of a command FIFO, a write FIFO, a read FIFO, and command and data schedulers. Table 8 describes the purpose of each FIFO.

Figure 16 shows the block diagram of the DDR2/mDDR memory controller FIFOs. Commands, write data, and read data arrive at the DDR2/mDDR memory controller parallel to each other. The same peripheral bus is used to write and read data from external memory as well as internal memory-mapped registers.

**Table 8. DDR2/mDDR Memory Controller FIFO Description**

| FIFO | Description | Depth (64-bit doublewords) |
|---|---|---|
| Command | Stores all commands coming from on-chip requestors | 7 |
| Write | Stores write data coming from on-chip requestors to memory | 11 |
| Read | Stores read data coming from memory to on-chip requestors | 17 |

**Figure 16. DDR2/mDDR Memory Controller FIFO Block Diagram**




### 2.7.1 Command Ordering and Scheduling, Advanced Concept

The DDR2/mDDR memory controller performs command re-ordering and scheduling in an attempt to achieve efficient transfers with maximum throughput. The goal is to maximize the utilization of the data, address, and command buses while hiding the overhead of opening and closing DDR2/mDDR SDRAM rows. Command re-ordering takes place within the command FIFO.

Typically, a given master issues commands on a single priority. EDMA transfer controller read and write ports are different masters. The DDR2/mDDR memory controller first reorders commands from each master based on the following rules:

- Selects the oldest command (first command in the queue)
- Selects a read before a write if:
  - The read is to a different block address (2048 bytes) than the write
  - The read has greater or equal priority

The second bullet above may be viewed as an exception to the first bullet. This means that for an individual master, all of its commands will complete from oldest to newest, with the exception that a read may be advanced ahead of an older, lower or equal priority write. Following this scheduling, each master may have one command ready for execution.

Next, the DDR2/mDDR memory controller examines each of the commands selected by the individual masters and performs the following reordering:

- Among all pending reads, selects reads to rows already open. Among all pending writes, selects writes to rows already open.
- Selects the highest priority command from pending reads and writes to open rows. If multiple commands have the highest priority, then the DDR2/mDDR memory controller selects the oldest command.

The DDR2/mDDR memory controller may now have a final read and write command. If the Read FIFO is not full, then the read command will be performed before the write command, otherwise the write command will be performed first.

Besides commands received from on-chip resources, the DDR2/mDDR memory controller also issues refresh commands. The DDR2/mDDR memory controller attempts to delay refresh commands as long as possible to maximize performance while meeting the SDRAM refresh requirements. As the DDR2/mDDR memory controller issues read, write, and refresh commands to DDR2/mDDR SDRAM memory, it adheres to the following rules:

1. Refresh request resulting from the Refresh Must level of urgency being reached
2. Read request without a higher priority write (selected from above reordering algorithm)
3. Refresh request resulting from the Refresh Need level of urgency being reached
4. Write request (selected from above reordering algorithm)
5. Refresh request resulting from Refresh May level of urgency being reached
6. Request to enter self-refresh mode

The following results from the above scheduling algorithm:

- All writes from a single master will complete in order
- All reads from a single master will complete in order
- From the same master, any read to the same location (or within 2048 bytes) as a previous write will complete in order

### 2.7.2 Command Starvation

The reordering and scheduling rules listed above may lead to command starvation, which is the prevention of certain commands from being processed by the DDR2/mDDR memory controller. Command starvation results from the following conditions:

- A continuous stream of high-priority read commands can block a low-priority write command
- A continuous stream of DDR2/mDDR SDRAM commands to a row in an open bank can block commands to the closed row in the same bank.

To avoid these conditions, the DDR2/mDDR memory controller can momentarily raises the priority of the oldest command in the command FIFO after a set number of transfers have been made. The PR_OLD_COUNT bit in the peripheral bus burst priority register (PBBPR) sets the number of the transfers that must be made before the DDR2/mDDR memory controller will raise the priority of the oldest command.

### 2.7.3 Possible Race Condition

A race condition may exist when certain masters write data to the DDR2/mDDR memory controller. For example, if master A passes a software message via a buffer in DDR2/mDDR memory and does not wait for indication that the write completes, when master B attempts to read the software message it may read stale data and therefore receive an incorrect message. In order to confirm that a write from master A has landed before a read from master B is performed, master A must wait for the write completion status from the DDR2/mDDR memory controller before indicating to master B that the data is ready to be read. If master A does not wait for indication that a write is complete, it must perform the following workaround:

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

The EDMA peripheral does not need to implement the above workaround. The above workaround is required for all other peripherals. Refer to the device-specific data manual for more information.

## 2.8 Refresh Scheduling

The DDR2/mDDR memory controller issues autorefresh (REFR) commands to DDR2/mDDR SDRAM devices at a rate defined in the refresh rate (RR) bit field in the SDRAM refresh control register (SDRCR). A refresh interval counter is loaded with the value of the RR bit field and decrements by 1 each cycle until it reaches zero. Once the interval counter reaches zero, it reloads with the value of the RR bit. Each time the interval counter expires, a refresh backlog counter increments by 1. Conversely, each time the DDR2/mDDR memory controller performs a REFR command, the backlog counter decrements by 1. This means the refresh backlog counter records the number of REFR commands the DDR2/mDDR memory controller currently has outstanding.

The DDR2/mDDR memory controller issues REFR commands based on the level of urgency. The level of urgency is defined in Table 9. Whenever the refresh must level of urgency is reached, the DDR2/mDDR memory controller issues a REFR command before servicing any new memory access requests. Following a REFR command, the DDR2/mDDR memory controller waits T_RFC cycles, defined in the SDRAM timing register (SDTIMR), before rechecking the refresh urgency level.

In addition to the refresh counter previously mentioned, a separate backlog counter ensures the interval between two REFR commands does not exceed 8× the refresh rate. This backlog counter increments by 1 each time the interval counter expires and resets to zero when the DDR2/mDDR memory controller issues a REFR command. When this backlog counter is greater than 7, the DDR2/mDDR memory controller issues four REFR commands before servicing any new memory requests.

The refresh counters do not operate when the DDR2/mDDR memory is in self-refresh mode.

**Table 9. Refresh Urgency Levels**

| Urgency Level | Description |
|---|---|
| Refresh May | Backlog count is greater than 0. Indicates there is a backlog of REFR commands, when the DDR2/mDDR memory controller is not busy it will issue the REFR command. |
| Refresh Release | Backlog count is greater than 3. Indicates the level at which enough REFR commands have been performed and the DDR2/mDDR memory controller may service new memory access requests. |
| Refresh Need | Backlog count is greater than 7. Indicates the DDR2/mDDR memory controller should raise the priority level of a REFR command above servicing a new memory access. |
| Refresh Must | Backlog count is greater than 11. Indicates the level at which the DDR2/mDDR memory controller should perform a REFR command before servicing new memory access requests. |

## 2.9 Self-Refresh Mode

Clearing the self refresh/low power (SR_PD) bit to 0 and then setting the low power mode enable (LPMODEN) bit to 1 in the SDRAM refresh control register (SDRCR) , forces the DDR2/mDDR memory controller to place the external DDR2/mDDR SDRAM in a low-power mode (self refresh), in which the DDR2/mDDR SDRAM maintains valid data while consuming a minimal amount of power. When the LPMODEN bit is set to 1, the DDR2/mDDR memory controller continues normal operation until all outstanding memory access requests have been serviced and the refresh backlog has been cleared. At this point, all open pages of DDR2/mDDR SDRAM are closed and a self-refresh (SLFRFR) command (an autorefresh command with self refresh/low power) is issued.

The memory controller exits the self-refresh state when a memory access is received, when the LPMODEN bit in SDRCR is cleared to 0, or when the SR_PD bit in SDRCR changed to 1. While in the self-refresh state, if a request for a memory access is received, the DDR2/mDDR memory controller services the memory access request, returning to the self-refresh state upon completion. The DDR2/mDDR memory controller will not wake up from the self-refresh state (whether from a memory access request, from clearing the LPMODEN bit, or from clearing the SR_PD bit) until T_CKE + 1 cycles have expired since the self-refresh command was issued. The value of T_CKE is defined in the SDRAM timing 2 register (SDTIMR2).

In the case of DDR2, after exiting from the self-refresh state, the memory controller will not immediately start executing commands. Instead, it will wait T_SXNR+1 clock cycles before issuing non-read/write commands and T_SXRD+1 clock cycles before issuing read or write commands. The SDRAM timing 2 register (SDTIM2) programs the values of T_SXNR and T_SXRD.

In the case of mDDR, after exiting from the self-refresh state, the memory controller will not immediately start executing commands. Instead, it will wait T_SXNR+1 clock cycles and then execute auto-refresh command before issuing any other commands. The SDRAM timing 2 register (SDTIM2) programs the value of T_SXNR.

Once in self-refresh mode, the DDR2/mDDR memory controller input clocks (VCLK and X2_CLK) may be gated off or changed in frequency. Stable clocks must be present before exiting self-refresh mode. See Section 2.17 for more information describing the proper procedure to follow when shutting down DDR2/mDDR memory controller input clocks.

See Section 2.17.1 for a description of the self-refresh programming sequence.

## 2.10 Partial Array Self Refresh for Mobile DDR

For additional power savings during self-refresh, the partial array self-refresh (PASR) feature of mDDR allows you to select the amount of memory that will be refreshed during self-refresh. Use the partial array self-refresh (PASR) bit field in the SDRAM configuration 2 (SDCR2) register to select the amount of memory to refresh during self-refresh. As shown in Table 10 you may select either 4, 2, 1, 1/2, or 1/4 bank(s). The PASR bits are loaded into the extended mode register of the mDDR device, during autoinitialization (see Section 2.14).

The EMIF performs bank interleaving when the internal bank position (IBANKPOS) bit in SDRAM configuration register (SDCR) is cleared to 0. Since the SDRAM banks are only partially refreshed during partial array self-refresh, it is recommended that you set IBANKPOS to 1 to avoid bank interleaving. When IBANKPOS is cleared to 0, it is the responsibility of software to move critical data into the banks that are to be refreshed during partial array self-refresh. Refer to Section 2.6.2 for more information on IBANKPOS and addressing mapping in general.

**Table 10. Configuration Bit Field for Partial Array Self-refresh**

| Bit Field | Bit Value | Bit Description |
|---|---|---|
| PASR | 0 | Refresh banks 0, 1, 2, and 3 |
| | 1h | Refresh banks 0 and 1 |
| | 2h | Refresh bank 0 |
| | 3h | Reserved |
| | 4h | Reserved |
| | 5h | Refresh 1/2 of bank 0 |
| | 6h | Refresh 1/4 of bank 0 |
| | 7h | Reserved |

## 2.11 Power Down Mode

Setting the self-refresh / low power (SR_PD) bit and the low power mode enable (LPMODEN) bit in the SDRAM refresh control register (SDRCR) to 1, forces the DDR2/mDDR memory controller to place the external DDR2 SDRAM in the power down mode. When the LPMODEN bit is asserted, the DDR2/mDDR memory controller continues normal operation until all outstanding memory access requests have been serviced and the refresh backlog has been cleared. At this point, all open pages of DDR2 SDRAM are closed and a Power Down command (same as NOP command but driving CKE low on the same cycle) is issued.

The DDR2/mDDR memory controller exits the power down state when a memory access is received, when a Refresh Must level is reached, when the LPMODEN bit in SDRCR is cleared to 0, or when the SR_PD bit in SDRCR changed to 0. While in the power down state, if a request for a memory access is received, the DDR2/mDDR memory controller services the memory access request, returning to the power down state upon completion. The DDR2/mDDR memory controller will not wake up from the power down state (whether from a memory access request, from reaching a Refresh Must level, from clearing the LPMODEN bit, or from clearing the SR_PD bit) until T_CKE + 1 cycles have expired since the power down command was issued. The value of T_CKE is defined in the SDRAM timing 2 register (SDTIMR2).

After exiting from the power down state, the DDR2/mDDR memory controller will drive CKE high and then not immediately start executing commands. Instead, it will wait T_XP+1 clock cycles before issuing commands. The SDRAM timing 2 register (SDTIM2) programs the values of T_XP.

See Section 2.17.1 for a description of the power down mode programming sequence.

**Note:** Power down mode is best suited as a power savings mode when SDRAM is being used intermittently and the system requires power savings as well as a short recovery time. You may use self-refresh mode if you desire additional power savings from disabling clocks .

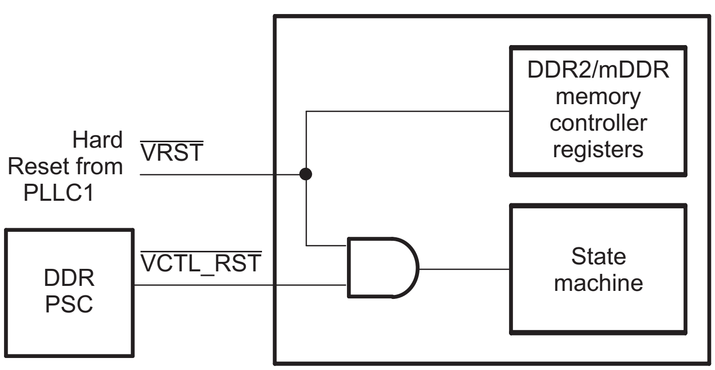
## 2.12 Reset Considerations

The DDR2/mDDR memory controller has two reset signals, $\overline{\text{VRST}}$ and $\overline{\text{VCTL\_RST}}$. The $\overline{\text{VRST}}$ is a module-level reset that resets both the state machine as well as the DDR2/mDDR memory controller memory-mapped registers. The $\overline{\text{VCTL\_RST}}$ resets the state machine only; it does not reset the controller's registers, which allows soft reset (from PSC or WDT) to reset the module without resetting the configuration registers and reduces the programming overhead for setting up access to the DDR2/mDDR device. If the DDR2/mDDR memory controller is reset independently of other peripherals, the user's software should not perform memory, as well as register accesses, while $\overline{\text{VRST}}$ or $\overline{\text{VCTL\_RST}}$ are asserted. If memory or register accesses are performed while the DDR2/mDDR memory controller is in the reset state, other masters may hang. Following the rising edge of $\overline{\text{VRST}}$ or $\overline{\text{VCTL\_RST}}$, the DDR2/mDDR memory controller immediately begins its initialization sequence. Command and data stored in the DDR2/mDDR memory controller FIFOs are lost. Table 11 describes the different methods for asserting each reset signal. The Power and Sleep Controller (PSC) acts as a master controller for power management for all of the peripherals on the device. For detailed information on power management procedures using the PSC, refer to the *TMS320DM365 Digital Media System-on-Chip (DMSoC) ARM Subsystem Reference Guide* (SPRUFG5). Figure 17 shows the DDR2/mDDR memory controller reset diagram.

**Table 11. Reset Sources**

| Reset Signal | Reset Source |
|---|---|
| $\overline{\text{VRST}}$ | Hardware/device reset |
| $\overline{\text{VCTL\_RST}}$ | Power and sleep controller |

**Figure 17. DDR2/mDDR Memory Controller Reset Block Diagram**



### 2.13 VTP IO Buffer Calibration

The DDR2/mDDR memory controller is able to control the impedance of the output IO. This feature allows the DDR2/mDDR memory controller to tune the output impedance of the IO to match that of the PCB board. Control of the output impedance of the IO is an important feature because impedance matching reduces reflections, creating a cleaner board design. Calibrating the output impedance of the IO will also reduce the power consumption of the DDR2/mDDR memory controller. The calibration is performed with respect to voltage, temperature, and process (VTP). The VTP information obtained from the calibration is used to control the output impedance of the IO.

The impedance of the output IO is selected by the value of a reference resistor connected to pin DDR_PADREFP. The DDR2/mDDR reference design requires the reference resistor to be 50 ohm .5% tolerance 1/16th watt resistor (49.9 ohm .5% tolerance is acceptable).

The VTP IO Control Register (VTPIOCR) is written to begin the calibration process. The VTP calibration process is described in the DDR2/mDDR initialization sequence in Section 2.14.2.

**Note:** VTP IO calibration must be performed following device power up and device reset. If the DDR2/mDDR memory controller is reset via the Power and Sleep Controller (PSC) and the VTP input clock is disabled, accesses to the DDR2/mDDR memory controller will not complete. To re-enable accesses to the DDR2/mDDR memory controller, enable the VTP input clock and then perform the VTP calibration sequence again.

### 2.14 Auto-Initialization Sequence

The DDR2/mDDR SDRAM contains mode and extended mode registers that configure the DDR2/mDDR memory for operation. These registers control burst type, burst length, CAS latency, DLL enable/disable (on the DDR2/mDDR device), single-ended strobedifferential strobe, etc. The DDR2/mDDR memory controller programs the mode and extended mode registers of the DDR2/mDDR memory by issuing MRS and EMRS commands during the initialization sequence. The bits SDRAMEN, MSDRAMEN, DDREN, and DDR2EN in the SDRAM Configuration register (SDCR) register determine if the DDR2/mDDR memory controller will perform a DDR2 or mobile DDR initialization sequence. Set these bits as follows for DDR2: SDRAMEN=1, MSDRAMEN=0, DDREN=1, DDR2EN=1. Set these bits as follow for mDDR: SDRAMEN=1, MSDRAMEN=1, DDREN=1, DDR2EN=0. The DDR2 initialization sequence performed by the DDR2/mDDR memory controller is compliant with the JESDEC79-2A specification and the mDDR initialization sequence is compliant with mDDR. The DDR2/mDDR memory controller performs an initialization sequence under the following conditions:

- Following reset (rising edge of $\overline{\text{VRST}}$ or $\overline{\text{VCTL\_RST}}$)
- Following a write to the DDRDRIVE, CL, IBANK, or PAGESIZE bit fields SDRAM configuration register (SDCR)

During the initialization sequence, the memory controller issues MRS and EMRS commands that configure the DDR2/Mobile DDR SDRAM mode register and extended mode register 1. The register values for DDR2 are described in Table 12 and Table 13, and the register values for mobile DDR are described in Table 14 and Table 15. The extended mode registers 2 and 3 are configured with a value of 0h. At the end of the initialization sequence, the memory controller performs an autorefresh cycle, leaving the memory controller in an idle state with all banks deactivated.

When a reset occurs, the DDR2/mDDR memory controller immediately begins the initialization sequence. Under this condition, commands and data stored in the DDR2/mDDR memory controller FIFOs will be lost. However, when the initialization sequence is initiated by a write to the two least-significant bytes in SDCR, data and commands stored in the DDR2/mDDR memory controller FIFOs will not be lost and the DDR2/mDDR memory controller will ensure read and write commands are completed before starting the initialization sequence.

**Table 12. DDR2 SDRAM Configuration by MRS Command**

| DDR2/mDDR Memory Controller Address Bus | Value | DDR2/mDDR SDRAM Register Bit | DDR2/mDDR SDRAM Field | Function Selection |
|---|---|---|---|---|
| DDR_A[12] | 0 | 12 | Power Down Exit | Fast exit |
| DDR_A[11:9] | t_WR | 11:9 | Write Recovery | Write recovery from autoprecharge. Value of 2, 3, 4, 5, or 6 is programmed based on value of the T_WR bit in the SDRAM timing register (SDTIMR). |
| DDR_A[8] | 0 | 8 | DLL Reset | Out of reset |
| DDR_A[7] | 0 | 7 | Mode: Test or Normal | Normal mode |
| DDR_A[6:4] | CL bit | 6:4 | CAS Latency | Value of 2, 3, 4, or 5 is programmed based on value of the CL bit in the SDRAM configuration register (SDCR). |
| DDR_A[3] | 0 | 3 | Burst Type | Sequential |
| DDR_A[2:0] | 3h | 2:0 | Burst Length | Value of 8 |

**Table 13. DDR2 SDRAM Configuration by EMRS(1) Command**

| DDR2/mDDR Memory Controller Address Bus | Value | DDR2/mDDR SDRAM Register Bit | DDR2/mDDR SDRAM Field | Function Selection |
|---|---|---|---|---|
| DDR_A[12] | 0 | 12 | Output Buffer Enable | Output buffer enable |
| DDR_A[11] | 0 | 11 | RDQS Enable | RDQS disable |
| DDR_A[10] | 1 | 10 | $\overline{\text{DQS enable}}$ | Disables differential DQS signaling. |
| DDR_A[9:7] | 0 | 9:7 | OCD Calibration Program | Exit OCD calibration |
| DDR_A[6] | 0 | 6 | ODT Value (Rtt) | Cleared to 0 to select 75 ohms. This feature is not supported because the DDR_ODT signal is not pinned out. |
| DDR_A[5:3] | 0 | 5:3 | Additive Latency | 0 cycles of additive latency |
| DDR_A[2] | 1 | 2 | ODT Value (Rtt) | Set to 1 to select 75 ohms. This feature is not supported because the DDR_ODT signal is not pinned out. |
| DDR_A[1] | DDRDRIVE0 bit | 1 | Output Driver Impedance | Value of 0 or 1 is programmed based on value of DDRDRIVE0 bit in SDRAM configuration register |
| DDR_A[0] | 0 | 0 | DLL enable | DLL enable |

**Table 14. Mobile DDR SDRAM Configuration by MRS Command**

| Memory Controller Address Bus | Value | Mobil DDR SDRAM Register Bit | Mobile DDR SDRAM Field | Function Selection |
|---|---|---|---|---|
| DDR_A[11:7] | 0 | 11:7 | Operating mode | Normal operating mode |
| DDR_A[6:4] | CL bit | 6:4 | CAS Latency | Value of 2 or 3 is programmed based on value of CL bit in SDRAM configuration register |
| DDR_A[3] | 0 | 3 | Burst Type | Sequential |
| DDR_A[2:0] | 3h | 2:0 | Burst Length | Value of 8 |

**Table 15. Mobile DDR SDRAM Configuration by EMRS(1) Command**

| Memory Controller Address Bus | Value | Mobil DDR SDRAM Register Bit | Mobile DDR SDRAM Field | Function Selection |
|---|---|---|---|---|
| DDR_A[11:7] | 0 | 11:7 | Operating Mode | Normal operating mode |
| DDR_A[6:5] | DDRDRIVE[1:0] bits | 6:5 | Output Driver Impedance | Value of 0, 1, 2, or 3 is programmed based on value of DDRDRIVE[1:0] bits in SDRAM configuration register |
| DDR_A[4:3] | 0 | 4:3 | Temperature Compensated Self Refresh | Value of 0 |
| DDR_A[2:0] | PASR bits | 2:0 | Partial Array Self Refresh | Value of 0, 1, 2, 5, or 6 is programmed based on value of PASR bits in SDRAM configuration 2 register |

### 2.14.1 Initializing Configuration Registers

Perform the following steps when configuring the DDR2/mDDR memory controller memory-mapped registers:

1. Program the SDRAM configuration register (SDCR) with the BOOTUNLOCK bit set to 1 (unlocked).
2. Program the SDRAM configuration register (SDCR) to the desired value with the BOOTUNLOCK set to 0 and the TIMUNLOCK bit set to 1 (unlocked).
3. For mDDR only, program the SDRAM configuration register 2 (SDCR2) to the desired value.
4. Program the SDRAM timing register (SDTIMR) and SDRAM timing register 2 (SDTIMR2) to the desired values to meet the DDR2/mDDR SDRAM memory data sheet specification.
5. Program SDCR to the desired value with the TIMUNLOCK bit cleared to 0 (locked).
6. Program the RR bit in the SDRAM refresh control register (SDRCR) to the desired value to meet the refresh requirements of the DDR2/mDDR SDRAM memory.

**Note:** Before accessing the DDR2/mDDR memory controller registers, you must complete VTP initialization. Accessing the DDR2/mDDR memory controller registers prior to VTP initialization will result in a bus lock-up condition. See Section 2.14.2 for the overall initialization sequence.

### 2.14.2 Initializing Following Device Power Up and Device RESET

Following device power up, the DDR2/mDDR memory controller is held in reset with the internal clocks to the module gated off. Before releasing the DDR2/mDDR memory controller from reset, the clocks to the module must be turned on. Perform the following steps when turning the clocks on and initializing the module:

1. Program PLLC1 / PLLC2 registers to start X2_CLK. For information on programming PLLC1/PLLC2, refer to the *TMS320DM365 Digital Media System-on-Chip (DMSoC) ARM Subsystem Reference Guide* (SPRUFG5).
2. Program the power and sleep controller (PSC) to enable the DDR2/mDDR memory controller VCLK.
3. Perform VTP IO calibration
   a. Clear CLRZ, LOCK, IOPWRDN, and PWRDN bits in the VTP IO control register (VTPIOCR) and wait at least 1 reference clock cycle. You must wait at least 1 reference clock cycle for the CLRZ to take effect. The reference clock is the clock at MXI/MXO.
   b. Set bit CLRZ in VTPIOCR
   c. Poll READY bit in VTPIOCR until it changes to logic-high
   d. Set LOCK bit in VTPIOCR
   e. Set IOPWRDN and PWRDN bits in VTPIOCR. VTP is locked and no dynamic calibration will occur.
4. Configure the DDR PHY control register 1 (DDRPHYCR1). All of the following steps may be performed in a single register write to DDRPHYCR1.
   a. Set bit CONFIG_EXT_ STRBEN to select external DQS strobe gating
   b. Set bit CONFIG_PWRDNEN.
   c. Program the Read Latency to the required value.
5. Configure the peripheral bus burst priority register (PBBPR). You must change its default value. See Section 4.7.
6. Follow the register initialization procedure described in Section 2.14.1 to complete the DDR2/mDDR memory controller configuration.
7. Program the power and sleep controller (PSC) to reset (synchReset) the DDR2/mDDR memory controller and re-enable the clocks.

**Note:** If the DDR2/mDDR memory controller is reset via the power and sleep controller (PSC) and the VTP input clock is disabled, accesses to the DDR2/mDDR memory controller will not complete. To re-enable accesses to the DDR2/mDDR memory controller, enable the VTP input clock and then perform the VTP calibration sequence again.

## *2.15 Interrupt Support*

The DDR2/mDDR memory controller supports two addressing modes, linear incrementing and cache line wrap. Upon receipt of an access request for an unsupported addressing mode, the DDR2/mDDR memory controller generates an interrupt by setting the LT bit in the interrupt raw register (IRR). The DDR2/mDDR memory controller will then treat the request as a linear incrementing request.

This interrupt is called the line trap interrupt and is the only interrupt the DDR2/mDDR memory controller supports. It is an active-high interrupt and is enabled by the LTMSET bit in the interrupt mask set register (IMSR). This interrupt is mapped to the ARM and is muxed with RTCINT.

### 2.16 DMA Event Support

The DDR2/mDDR memory controller is a DMA slave peripheral and therefore does not generate DMA events. Data read and write requests may be made directly by masters and by the DMA.

### 2.17 Power Management

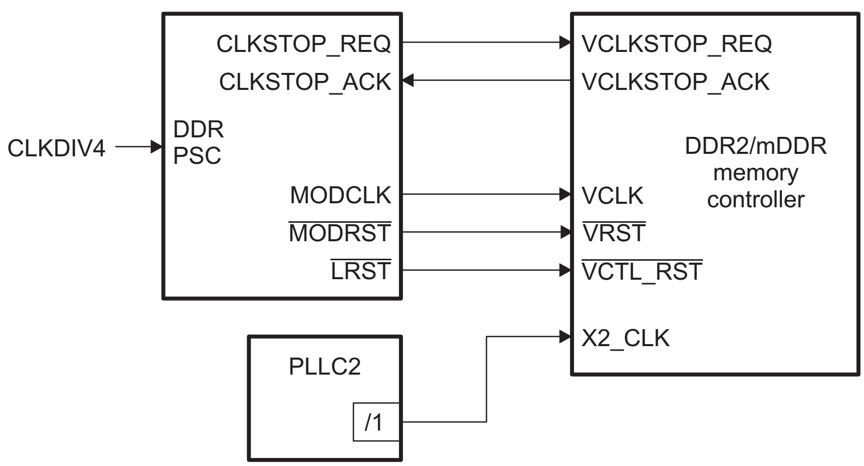
Power dissipation from the DDR2/mDDR memory controller may be managed by three methods:

- Self-refresh mode (see Section 2.9)
- Power down mode (see Section 2.11)
- Gating input clocks to the module off

Gating input clocks off to the DDR2/mDDR memory controller achieves higher power savings when compared to the power savings of self-refresh mode and power down mode. The input clocks are turned off outside of the DDR2/mDDR memory controller through the use of the Power and Sleep Controller (PSC) and the PLL controller 2 (PLLC2). Figure 18 shows the connections between the DDR2/mDDR memory controller, PSC, and PLLC2. For detailed information on power management procedures using the PSC, refer to the *TMS320DM365 Digital Media System-on-Chip (DMSoC) ARM Subsystem Reference Guide* (SPRUFG5).

Before gating clocks off, the DDR2/mDDR memory controller must place the DDR2/mDDR SDRAM memory in self-refresh mode. If the external memory requires a continuous clock, the DDR2/mDDR memory controller clock provided by PLLC2 must not be turned off because this may result in data corruption. See the following subsections for the proper procedures to follow when stopping the DDR2/mDDR memory controller clocks. Once the clocks are stopped, to re-enable the clocks follow the clock stop procedure in each respective subsection in reverse order.

**Figure 18. DDR2/mDDR Memory Controller Power Sleep Controller Diagram**



### 2.17.1 DDR2/mDDR Memory Controller Clock Stop Procedure

**Note:** If a data access occurs to the DDR2/mDDR memory after completing steps 1-5, the DLL will wake up and lock, then the MCLK will turn on and the access will be performed. Following steps 6 and 7, in which the clocks are disabled , all DDR2 accesses are not possible until the clocks are reenabled.

**Note:** In power down mode, the DDR2/mDDR memory controller input clocks (VCLK and X2_CLK) may not be gated off. This is a limitation of the DDR2/mDDR controller. For this reason, power down mode is best suited as a power savings mode when SDRAM is being used intermittently and the system requires power savings as well as a short recovery time. You may use self-refresh mode if you desire additional power savings from disabling clocks.

To achieve maximum power savings VCLK, MCLK, X2_CLK, DDR_CLK, and $\overline{\text{DDR\_CLK}}$ should be gated off. The procedure for clock gating is described in the following steps.

1. Allow software to complete the desired DDR transfers.
2. Change the SR_PD bit to 0 and set the LPMODEN bit in the DDR2 SDRAM refresh control register (SDRCR) to enable self-refresh mode. The DDR2/mDDR memory controller will complete any outstanding accesses and backlogged refresh cycles and then place the external DDR2/mDDR memory in self-refresh mode.
3. Set the MCLKSTOPEN bit in SDRCR. This enables the DDR2/mDDR memory controller to shut off the MCLK.
4. Poll the PHYRDY bit in the SDRAM status register (SDRSTAT) to be a logic-low indicating that the MCLK has been stopped.
5. Program the PSC to disable the DDR2/mDDR memory controller VCLK. You must not disable VCLK in power down mode; use only for self-refresh mode (see notes in this section).
6. Program PLLC2 registers to stop X2_CLK to DDR2/mDDR memory controller, as well as DDR_CLK and $\overline{\text{DDR\_CLK}}$. You must not disable X2_CLK in power down mode; use only for self-refresh mode (see notes in this section). For information on programming PLLC2, refer to the *TMS320DM365 Digital Media System-on-Chip (DMSoC) ARM Subsystem Reference Guide* (SPRUFG5).

To turn clocks back on:

1. Program PLLC2 registers to start X2_CLK to the DDR2/mDDR memory controller.
2. Once X2_CLK is stable, program the PSC to enable VCLK.
3. Clear the MCLKSTOPEN bit in SDRCR to 0.
4. Clear the LPMODEN bit in the DDR2 SDRAM refresh control register (SDRCR) to 0.

## *2.18 Emulation Considerations*

The DDR2/mDDR memory controller will remain fully functional during emulation halts to allow emulation access to external memory.

**Note:** VTP IO calibration must be performed before emulation tools attempt to access the register or data space of the DDR2/mDDR memory controller. A bus lock-up condition will occur if the emulation tool attempts to access the register or data space of the DDR2/mDDR memory controller before completing VTP IO calibration. See Section 2.13 for information on VTP IO calibration.

# 3 Supported Use Cases

The DDR2/mDDR memory controller allows a high degree of programmability for shaping DDR2/mDDR accesses. The programmability inherent to the DDR2/mDDR memory controller provides the DDR2/mDDR memory controller with the flexibility to interface with a variety of DDR2/mDDR devices. By programming the SDRAM configuration register (SDCR), SDRAM refresh control register (SDRCR), SDRAM timing register (SDTIMR), and SDRAM timing register 2 (SDTIMR2), the DDR2/mDDR memory controller can be configured to meet the data sheet specification for JESD79D-2A compliant DDR2 SDRAM as well as mDDR memory devices.

This section presents an example describing how to interface the DDR2 memory controller to a JESD79D DDR2/mDDR-400 1-Gb device. The DDR2/mDDR memory controller is assumed to be operating at 133 MHz. A similar procedure can be followed when interfacing to a mDDR memory device.

## 3.1 Connecting the DDR2/mDDR Memory Controller to DDR2/mDDR Memory

Figure 19 shows how to connect the DDR2/mDDR memory controller to a DDR2 device. displays a 16-bit interface; you can see that all signals are point-to-point connection.

## 3.2 Configuring Memory-Mapped Registers to Meet DDR2 Specification

As previously stated, four memory-mapped registers must be programmed to configure the DDR2/mDDR memory controller to meet the data sheet specification of the attached DDR2/mDDR device. The registers are:

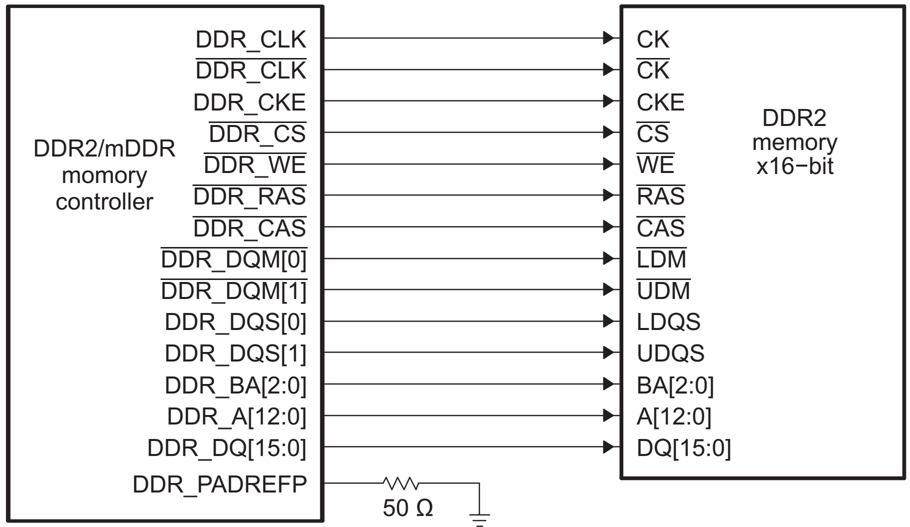
- SDRAM configuration register (SDCR)
- SDRAM refresh control register (SDRCR)
- SDRAM timing register (SDTIMR)
- SDRAM timing register 2 (SDTIMR2)

In addition to these registers, the DDR PHY control register (DDRPHYCR1) must also be programmed. The configuration of DDRPHYCR1 is not dependent on the DDR2 device specification but rather on the board layout.

The following sections describe how to configure each of these registers. See Section 4 for more information on the DDR2/mDDR memory controller registers.

**Note:** When interfacing the DDR2/mDDR memory controller to a mDDR device, the SDRAM configuration register 2 (SDCR2) must be programmed in addition to the registers mentioned above.

**Figure 19. Connecting DDR2/mDDR Memory Controller to a 16-Bit DDR2 Memory**




(1) Single-ended DQS is used.

(2) If differential DQS is used, connect DDR_DQSN0 to /LDQSN and DDR_DQSN1 to /UDQSN.

### 3.2.1 Configuring SDRAM Configuration Register (SDCR)

The SDRAM configuration register (SDCR) contains register fields that configure the DDR2/mDDR memory controller to match the data bus width, CAS latency, number of banks, and page size of the attached memory. In this example, we assume the following DDR2 configuration:

- Data bus width = 16 bits
- CAS latency = 4
- Number of banks = 8
- Page size = 1024 words

Table 16 shows the resulting SDCR configuration. Note that the value of the TIMING_UNLOCK field is dependent on whether or not it is desirable to unlock SDTIMR and SDTIMR2. The TIMING_UNLOCK bit should only be set to 1 when the SDTIMR and SDTIMR2 needs to be updated.

**Table 16. SDCR Configuration**

| Field | Value | Function Selection |
|---|---|---|
| TIMING_UNLOCK | x | Set to 1 to unlock the SDRAM timing and timing 2 registers. Cleared to 0 to lock the SDRAM timing and timing 2 registers. |
| NM | 1h | To configure the DDR2/mDDR memory controller for a 16-bit data bus width. |
| CL | 4h | To select a CAS latency of 4. |
| IBANK | 3h | To select 8 internal DDR2 banks. |
| PAGESIZE | 2h | To select 1024-word page size. |

### 3.2.2 Configuring SDRAM Refresh Control Register (SDRCR)

The SDRAM refresh control register (SDRCR) configures the DDR2/mDDR memory controller to meet the refresh requirements of the attached memory device. SDRCR also allows the DDR2/mDDR memory controller to enter and exit self refresh and enable and disable the MCLK stopping. In this example, we assume that the DDR2/mDDR memory controller is not is in self-refresh mode or power down mode and that MCLK stopping is disabled.

The RR field in SDRCR is defined as the rate at which the attached memory device is refreshed in DDR2/mDDR cycles. The value of this field may be calculated using the following equation:

*RR = DDR2/mDDR clock frequency × DDR2/mDDR memory refresh period*

Table 17 displays the DDR2-400 refresh rate specification.

**Table 17. DDR2 Memory Refresh Specification**

| Symbol | Description | Value |
|---|---|---|
| $t_{REF}$ | Average Periodic Refresh Interval | 7.8 µs |

Therefore, the following results assuming 133 MHz DDR2/mDDR clock frequency.

RR = 133 MHz × 7.8 µs = 1037.4

Therefore, RR = 1038 = 40Eh.

Table 18 shows the resulting SDRCR configuration.

**Table 18. SDRCR Configuration**

| Field | Value | Function Selection |
|---|---|---|
| LPMODEN | 0 | DDR2/mDDR memory controller is not in power down mode. |
| MCLKSTOP_EN | 0 | MCLK stopping is disabled. |
| SR_PD | 0 | Leave a default value. |
| RR | 40Eh | Set to 40Eh DDR2 clock cycles to meet the DDR2/mDDR memory refresh rate requirement. |

### 3.2.3 Configuring SDRAM Timing Registers (SDTIMR and SDTIMR2)

The SDRAM timing register (SDTIMR) and SDRAM timing register 2 (SDTIMR2) configure the DDR2/mDDR memory controller to meet the data sheet timing parameters of the attached memory device. Each field in SDTIMR and SDTIMR2 corresponds to a timing parameter in the DDR2/mDDR data sheet specification. Table 19 and Table 20 display the register field name and corresponding DDR2 data sheet parameter name along with the data sheet value. These tables also provide a formula to calculate the register field value and displays the resulting calculation. Each of the equations include a minus 1 because the register fields are defined in terms of DDR2/mDDR clock cycles minus 1. See Section 4.4 and Section 4.5 for more information.

**Table 19. SDTIMR Configuration**

| Register Field Name | DDR2 Data Manual Parameter Name | Description | Data Manual Value (nS) | Formula (Register field must be ≥) | Register Value |
|---|---|---|---|---|---|
| T_RFC | $t_{RFC}$ | Refresh cycle time | 127.5 | $(t_{RFC} \times f_{DDR2/mDDR\_CLK}) - 1$ | 16 |
| T_RP | $t_{RP}$ | Precharge command to refresh or activate command | 20 | $(t_{RP} \times f_{DDR2/mDDR\_CLK}) - 1$ | 2 |
| T_RCD | $t_{RCD}$ | Activate command to read/write command | 20 | $(t_{RCD} \times f_{DDR2/mDDR\_CLK}) - 1$ | 2 |
| T_WR | $t_{WR}$ | Write recovery time | 15 | $(t_{WR} \times f_{DDR2/mDDR\_CLK}) - 1$ | 1 |
| T_RAS | $t_{RAS}$ | Active to precharge command | 45 | $(t_{RAC} \times f_{DDR2/mDDR\_CLK}) - 1$ | 5 |
| T_RC | $t_{RC}$ | Activate to Activate command in the same bank | 65 | $(t_{RC} \times f_{DDR2/mDDR\_CLK}) - 1$ | 8 |

**Table 19. SDTIMR Configuration (continued)**

| Register Field Name | DDR2 Data Manual Parameter Name | Description | Data Manual Value (nS) | Formula (Register field must be ≥) | Register Value |
|---|---|---|---|---|---|
| T_RRD | $t_{RRD}$ | Activate to Activate command in a different bank | 10 | $((4 \times t_{RRD}) + (2 \times t_{CK}))/(4 \times t_{CK}) - 1$ | 1 |
| T_WTR | $t_{WTR}$ | Write to read command delay | 10 | $(t_{WTR} \times f_{DDR2/mDDR\_CLK}) - 1$ | 1 |

**Note:** The equation given above for the T_RRD field applies only for 8 bank DDR2/mDDR memories. When interfacing to DDR2/mDDR memories with less than 8 banks the T_RRD field should be calculated using the following equation $(t_{RRD} \times f_{DDR2/mDDR\_CLK})-1$.

**Table 20. SDTIMR2 Configuration**

| Register Field Name | DDR2 Data Manual Parameter Name | Description | Data Manual Value | Formula (Register field must be ≥) | Register Value |
|---|---|---|---|---|---|
| T_RASMAX | $t_{RAS}$(MAX) | Active to precharge command | 70 µs | $t_{RAS}(MAX)/_{DDR\ refresh\ rate} - 1$ | 8 |
| T_XP | $t_{XP}$ | Exit power down to a non-read command | 2($t_{CK}$ cycles) | If $t_{XP} > t_{CKE}$, then T_XP = $t_{XP}$ - 1, else T_XP = $t_{CKE}$ - 1 | 2 |
| T_XSNR | $t_{XSNR}$ | Exit self refresh to a non-read command | 137.5 nS | $(t_{XSNR} \times f_{DDR2/mDDR\_CLK}) - 1$ | 18 |
| T_XSRD | $t_{XSRD}$ | Exit self refresh to a read command | 200 ($t_{CK}$ cycles) | $t_{XSRD}$ - 1 | 199 |
| T_RTP | $t_{RTP}$ | Read to precharge command delay | 7.5 nS | $(t_{RTP} \times f_{DDR2/mDDR\_CLK}) - 1$ | 1 |
| T_CKE | $t_{CKE}$ | CKE minimum pulse width | 3 ($t_{CK}$ cycles) | $t_{CKE}$ - 1 | 2 |

### 3.2.4 Configuring DDR PHY Control Register (DDRPHYCR1)

The DDR PHY control register (DDRPHYCR1) contains a read latency (READ_LATENCY) field that helps the DDR2/mDDR memory controller determine when to sample read data. The (READ_LATENCY) field should be programmed to a value equal to CAS latency plus round trip board delay minus 1. The minimum (READ_LATENCY) value is CAS latency plus 1 and the maximum (READ_LATENCY) value is CAS latency plus 2 (again, the (READ_LATENCY) field would be programmed to these values minus 1).

When calculating round trip board delay the signals of primary concern are the differential clock signals (DDR_CLK and $\overline{\text{DDR\_CLK}}$) and data strobe signals (DDR_DQS). For these signals, calculate the round trip board delay from the DDR memory controller to the memory and then choose the maximum delay to determine the (READ_LATENCY) value. In this example we will assume the round trip board delay is one DDR_CLK cycle, therefore (READ_LATENCY) can be calculated as follows:

(READ_LATENCY) = CAS latency + round trip board delay – 1 = 4 + 1 – 1 = 4

**Table 21. DDRPHYCR1 Configuration**

| Register Field Name | Description | Register Value |
|---|---|---|
| CONFIG_EXT_STRBEN | Programs to select external strobe gating | 1 |
| READ_LATENCY | Read latency is equal to CAS latency plus round trip board delay for data minus 1 | 4 |

**Table 21. DDRPHYCR1 Configuration (continued)**

| Register Field Name | Description | Register Value |
|---|---|---|
| CONFIG_PWRDNEN | Programmed to power up the DDR2/mDDR memory controller receivers | 0 |

## 4 DDR2/mDDR Memory Controller Registers

Table 22 lists the memory-mapped registers for the DDR2/mDDR memory controller. Note that the VTP IO control register (VTPIOCR) resides in the System Control Module register set.

**Table 22. DDR2/mDDR Memory Controller Registers**

| Address | Acronym | Register Description | Section |
|---|---|---|---|
| 2000 0004h | SDRSTAT | SDRAM Status Register | Section 4.1 |
| 2000 0008h | SDCR | SDRAM Configuration Register | Section 4.2 |
| 2000 000ch | SDRCR | SDRAM Refresh Control Register | Section 4.3 |
| 2000 0010h | SDTIMR | SDRAM Timing Register | Section 4.4 |
| 2000 0014h | SDTIMR2 | SDRAM Timing Register 2 | Section 4.5 |
| 2000 001Ch | SDCR2 | SDRAM Configuration Register 2 | Section 4.6 |
| 2000 0020h | PBBPR | Peripheral Bus Burst Priority Register | Section 4.7 |
| 2000 00C0h | IRR | Interrupt Raw Register | Section 4.8 |
| 2000 00C4h | IMR | Interrupt Masked Register | Section 4.9 |
| 2000 00C8h | IMSR | Interrupt Mask Set Register | Section 4.10 |
| 2000 00CCh | IMCR | Interrupt Mask Clear Register | Section 4.11 |
| 2000 00E4h | DDRPHYCR1 | DDR PHY Control Register 1 | Section 4.12 |
| 01C40074 [(1)] | VTPIOCR | VTP IO Control Register | Section 4.13 |

(1) VTPIOCR resides in the register space of the System Control module. It is listed in the register space of the DDR2/mDDR controller because it is applicable to the DDR2/mDDR controller.

### 4.1 *SDRAM Status Register (SDRSTAT)*

The SDRAM status register (SDRSTAT) is shown in Figure 20 and described in Table 23.

**Figure 20. SDRAM Status (SDRSTAT) Register**

| 31–16 |
|---|
| Reserved |
| R-4000h |

| 15–3 | 2 | 1–0 |
|---|---|---|
| Reserved | PHYRDY | Reserved |
| R-0 | R-0 | R-0 |

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

**Table 23. SDRAM Status (SDRSTAT) Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |

**Table 23. SDRAM Status (SDRSTAT) Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 2 | PHYRDY | | DDR2/mDDR memory controller DLL ready. Specifies whether the DDR2/mDDR memory controller DLL is powered up and locked. |
| | | 0 | DLL is not ready, either powered down, in reset, or not locked. |
| | | 1 | DLL is powered up, locked, and ready for operation. |
| 1-0 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |

### 4.2 SDRAM Configuration Register (SDCR)

The SDRAM configuration register (SDCR) contains fields that program the DDR2/mDDR memory controller to meet the specification of the attached DDR2/mDDR memory. These fields configure the DDR2/mDDR memory controller to match the data bus width, CAS latency, number of internal banks, and page size of the attached DDR2/mDDR memory. The SDCR is shown in Figure 21 and described in Table 24. Writing to the DDRDRIVE[1:0], CL, IBANK, and PAGESIZE bit fields will cause the DDR2/mDDR memory controller to start the DDR2/mDDR SDRAM initialization sequence.

**Figure 21. SDRAM Configuration (SDCR) Register**

| 31-27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | IBANKPOS | MSDRAMEN | DDRDRIVE1 | BOOTUNLOCK | DDR_DDQS | Reserved | DDR2EN | Reserved | DDRDRIVE0 | DDREN | SDRAMEN |
| R-0 | RW-0 | RW-0 | RW-0 | RW-0 | R/W-0 | R-0 | RW-1 | N-0 | RW-1 | RW-1 | RW-1 |

| 15 | 14 | 13-12 | 11-9 | 8-7 | 6-4 | 3 | 2-0 |
|---|---|---|---|---|---|---|---|
| TIMUNLOCK | NM | Reserved | CL | Reserved | IBANK | Reserved | PAGESIZE |
| RW- | RW-0 | N-0 | RW-5 | N-0 | RW-2 | N-0 | RW-0 |

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

**Table 24. SDRAM Configuration (SDCR) Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | Reserved | | Any writes to these bit(s) must always have a value of 0. |
| 26 | IBANKPOS | | Internal Bank position |
| | | 0 | Normal addressing |
| | | 1 | Special addressing. Typically used with mobile DDR partial array self_refresh. |
| 25 | MSDRAMEN | | Mobile SDRAM enable. Use this bit in conjunction with DDR2EN, DDREN, and SDRAMEN to enable/disable mobile SDRAM. To change the MSDRAMEN bit use the following sequence: 1. Write a 1 to the BOOTUNLOCK bit. Write a 0 to the BOOTUNLOCK bit along with the desired value of the MSDRAMEN bit. |
| | | 0 | Disable mobile SDRAM |
| | | 1 | Enable mobile SDRAM |
| 24 | DDRDRIVE1 | | SDRAM drive strength. See bit field description for bit DDRDRIVE0. |
| 23 | BOOTUNLOCK | | Boot Unlock. Control the write permission settings for the DDRDRIVE[1:0], MSDRAMEN, DDR2EN, DDREN and SDRAMEN bits. To change these bits use the following sequence: 1. Write a 1 to the BOOTUNLOCK bit. Write a 0 to the BOOTUNLOCK bit along with the desired value of these bits. |
| | | 0 | DDRDRIVE bit may not be changed |
| | | 1 | DDRDRIVE bit may be changed |
| 22 | DDR_DDQS | | DDR2 SDRAM differential DQS enable. This bit is writeable only when the boot_unlock bit is unlocked. |
| | | 0 | Single-ended DQS |
| | | 1 | Differential DQS |
| 21 | Reserved | | .Any writes to these bit(s) must always have a value of 0. |
| 20 | DDR2EN | | DDR2 enable. Use this bit in conduction with DDR2EN, DDREN, and SDRAMEN to enable/disable DDR2. To change the DDR2EN bit use the following sequence: 1. Write a 1 to the BOOTUNLOCK bit. Write a 0 to the BOOTUNLOCK bit along with the desired value of the DDR2EN bit. |
| | | 0 | Disable DDR2 |
| | | 1 | Enable DDR2 |
| 19 | Reserved | | Any writes to these bit(s) must always have a value of 0. |

**Table 24. SDRAM Configuration (SDCR) Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18 | DDRDRIVE0 | | SDRAM drive strength. Bits DDRDRIVE[1:0] configure the output driver impedance control value of the SDRAM memory. For DDR2, set DDRDRIVE[1:0] to: 0h for normal drive strength 1h for weak drive strength For mobile DDR, set DDRDRIVE[1:0] to: 0 for full drive strength 1 for 1/2 drive strength 2 for 1/4 drive strength 3 for 1/8 drive strength. To change the DDRDRIVE[1:0] bit field use the following sequence: 1. Write a 1 to the BOOTUNLOCK bit. Write a 0 to the BOOTUNLOCK bit along with the desired value of the DDRDRIVE[1:0] bit field. |
| 17 | DDREN | | DDR enable. Use this bit in conjunction with DDR2EN, DDREN, and SDRAMEN to enable/disable DDR. To change the DDREN bit use the following sequence: 1. Write a 1 to the BOOTUNLOCK bit. Write a 0 to the BOOTUNLOCK bit along with the desired value of the DDREN bit. |
| | | 0 | Disable DDR |
| | | 1 | Enable DDR |
| 16 | SDRAMEN | | SDRAM enable. Use this bit in conjunction with SDRAMEN, SDRAMEN, and SDRAMEN to enable/disable SDRAM. To change the DRAMEN bit use the following sequence: 1. Write a 1 to the BOOTUNLOCK bit. Write a 0 to the BOOTUNLOCK bit along with the desired value of the DRAMEN bit. |
| | | 0 | Disable SDRAM |
| | | 1 | Enable SDRAM |
| 15 | TIMUNLOCK | | Timing unlock. Controls the write permission settings for the SDRAM Timing Register and SDRAM Timing Register 2. |
| | | 0 | Register fields in the SDRAM timing register (SDTIMR) and the SDRAM timing register 2 (SDTIMR2) may not be changed. |
| | | 1 | Register fields in the SDRAM timing register (SDTIMR) and the SDRAM timing register 2 (SDTIMR2) may be changed. |
| 14 | NM | | SDRAM data bus width. |
| | | 0 | This bit should always be set to 1. The 32-bit bus width is reserved on the DM36x. |
| | | 1 | 16_bit bus width. |
| 13-12 | Reserved | | Any writes to these bit(s) must always have a value of 0. |
| 11-9 | CL | | SDRAM CAS latency. 0_1h Reserved |
| | | 2 | CAS Latency = 2 |
| | | 3 | CAS Latency = 3 |
| | | 4 | CAS Latency = 4 |
| | | 5 | CAS Latency = 5 |
| 8-7 | Reserved | | Any writes to these bit(s) must always have a value of 0. |
| 6-4 | IBANK | | Internal SDRAM bank setup. Defines the number of internal banks on the external SDRAM device. 4h_7h Reserved |
| | | 0 | One bank |
| | | 1 | Two banks |
| | | 2 | Four banks |
| | | 3 | Eight banks |
| 3 | Reserved | | Any writes to these bit(s) must always have a value of 0. |
| 2-0 | PAGESIZE | | Page Size. Defines the page size of the SDRAM device. 4h_7h Reserved |
| | | 0 | 256_word page requiring 8 column address bits |
| | | 1 | 512_word page requiring 9 column address bits |
| | | 2 | 1024_word page requiring 10 column address bits |
| | | 3 | 2048_word page requiring 11 column address bits |

### 4.3 SDRAM Refresh Control Register (SDRCR)

The SDRAM refresh control register (SDRCR) is used to configure the DDR2/mDDR memory controller to:

- Enter and Exit the self-refresh and power down states.
- Enable and disable MCLK, stopping when in the self-refresh state.
- Meet the refresh requirement of the attached DDR2/mDDR device by programming the rate at which the DDR2/mDDR memory controller issues autorefresh commands.

The SDRCR is shown in Table 25 and described in Figure 22.

**Figure 22. SDRAM Refresh Control (SDRCR) Register**

| 31 | 30 | 29–24 | 23 | 22–16 |
|---|---|---|---|---|
| LPMODEN | MCLKSTOPEN | Reserved | SR_PD | Reserved |
| RW-0 | RW-0 | N-0 | RW-0 | N-0 |

| 15–0 |
|---|
| RR |
| RW0xD84 |

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

**Table 25. SDRAM Refresh Control (SDRCR) Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | LPMODEN | | Low power mode enable. |
| | | 0 | Disable low power mode. |
| | | 1 | Enable low power mode. The state of bit SR_PD selects either self refresh or power down mode. |
| 30 | MCLKSTOPEN | | MCLK stop enable. |
| | | 0 | Disables MCLK stopping, MCLK may not be stopped. |
| | | 1 | Enables MCLK stopping, MCLK may be stopped. The LPMODEN bit must be set to 1 before setting the MCLKSTOPEN bit to 1. |
| 29-24 | Reserved | | Any writes to these bit(s) must always have a value of 0. |
| 23 | SR_PD | | Self Refresh or Power Down select. Use this bit to select the low power mode to be active when bit LPMODEN is set to 1. |
| | | 0 | Select Self Refresh |
| | | 1 | Select Power Down |
| 22-16 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 15-0 | RR | 0-FFFFh | Refresh rate. Defines the rate at which the attached SDRAM devices will be refreshed. The value of this field may be calculated with the following equation: refresh rate = SDRAM frequency/SDRAM refresh rate where SDRAM refresh rate is derived from the SDRAM data sheet. |

### 4.4 SDRAM Timing Register (SDTIMR)

The SDRAM timing register (SDTIMR) configures the DDR2/mDDR memory controller to meet many of the AC timing specification of the DDR2/mDDR memory. The SDTIMR register is programmable only when the TIMUNLOCK bit is set to 1 in the SDCR. Note that DDR_CLK is equal to the period of the DDR_CLK signal. See the DDR2/mDDR memory data sheet for information on the appropriate values to program each field. The SDTIMR is shown in Figure 23 and described in Table 26.

**Figure 23. SDRAM Timing (SDTIMR) Register**

| 31-25 | 24-22 | 21-19 | 18-16 |
|---|---|---|---|
| T_RFC | T_RP | T_RCD | T_WR |
| R/W-0x1a | R/W-5 | R/W-5 | R/W-3 |

| 15-11 | 10-6 | 5-3 | 2 | 1-0 |
|---|---|---|---|---|
| T_RAS | T_RC | T_RRD | Reserved | T_WTR |
| R/W-9 | R/W-0xe | R/W-3 | R-0 | R/W-3 |

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

**Table 26. SDRAM Timing (SDTIMR) Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-25 | T_RFC | 0-7Fh | Specifies the minimum number of DDR_CLK cycles from a refresh or load mode command to a refresh or activate command, minus 1. Corresponds to the $t_{rfc}$ AC timing parameter in the DDR2/mDDR data sheet. Calculate by:<br>T_RFC = ($t_{rfc}$/DDR_CLK) - 1 |
| 24-22 | T_RP | 0-7h | Specifies the minimum number of DDR_CLK cycles from a precharge command to a refresh or activate command, minus 1. Corresponds to the $t_{rp}$ AC timing parameter in the DDR2/mDDR data sheet. Calculate by:<br>T_RP = ($t_{rp}$/DDR_CLK) - 1 |
| 21-19 | T_RCD | 0-7h | Specifies the minimum number of DDR_CLK cycles from an activate command to a read or write command, minus 1. Corresponds to the $t_{rcd}$ AC timing parameter in the DDR2/mDDR data sheet. Calculate by:<br>T_RCD = ($t_{rcd}$/DDR_CLK) - 1 |
| 18-16 | T_WR | 0-7h | Specifies the minimum number of DDR_CLK cycles from the last write transfer to a precharge command, minus 1. Corresponds to the $t_{wr}$ AC timing parameter in the DDR2/mDDR data sheet. Calculate by:<br>T_WR = ($t_{wr}$/DDR_CLK) - 1<br>When the value of this field is changed from its previous value, the initialization sequence will begin. |
| 15-11 | T_RAS | 0-1Fh | Specifies the minimum number of DDR_CLK cycles from an activate command to a precharge command, minus 1. Corresponds to the $t_{ras}$ AC timing parameter in the DDR2/mDDR data sheet. Calculate by:<br>T_RAS = ($t_{ras}$/DDR_CLK) - 1<br>T_RAS must be greater than or equal to T_RCD. |
| 10-6 | T_RC | 0-1Fh | Specifies the minimum number of DDR_CLK cycles from an activate command to an activate command, minus 1. Corresponds to the $t_{rc}$ AC timing parameter in the DDR2/mDDR data sheet. Calculate by:<br>T_RC = ($t_{rc}$/DDR_CLK) - 1 |
| 5-3 | T_RRD | 0-7h | Specifies the minimum number of DDR_CLK cycles from an activate command to an activate command in a different bank, minus 1. Corresponds to the $t_{rrd}$ AC timing parameter in the DDR2/mDDR data sheet. Calculate by:<br>T_RRD = ($t_{rrd}$/DDR_CLK) - 1<br>Note: for an 8 bank DDR2/mDDR device this field must be equal to ((4 × $t_{RRD}$) + (2 × $t_{CK}$)) / (4 × $t_{CK}$) - 1. |
| 2 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |

**Table 26. SDRAM Timing (SDTIMR) Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1-0 | T_WTR | 0-3h | Specifies the minimum number of DDR_CLK cycles from the last write to a read command, minus 1. Corresponds to the $t_{wtr}$ AC timing parameter in the DDR2/mDDR data sheet. Calculate by: T_WTR = ($t_{wtr}$/DDR_CLK) - 1 |

## 4.5 SDRAM Timing Register 2 (SDTIMR2)

Like the SDRAM timing register (SDTIMR), the SDRAM timing register 2 (SDTIMR2) also configures the DDR2/mDDR memory controller to meet the AC timing specification of the DDR2/mDDR memory. The SDTIMR2 register is programmable only when the TIMUNLOCK bit is set to 1 in the SDCR. See the DDR2/mDDR data sheet for information on the appropriate values to program each field. SDTIMR2 is shown in Figure 24 and described in Table 27.

**Figure 24. SDRAM Timing 2 (SDTIMR2) Register**

| 31 | 30-27 | 26-25 | 24-23 | 22-16 |
|---|---|---|---|---|
| Reserved | T_RASMAX | T_XP | Reserved | T_XSNR |
| N-0 | RW-0 | RW-0 | N-x | RW-0x1d |

| 15-8 | 7-5 | 4-0 |
|---|---|---|
| T_XSRD | T_RTP | T_CKE |
| RW-0xF1 | RW-2 | RW-5 |

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

**Table 27. SDRAM Timing 2 (SDTIMR2) Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 30-27 | T_RASMAX | 0-Fh | Specifies the maximum number of refresh rate intervals from Activate to Precharge command. Corresponds to the tras AC timing parameter and the refresh rate in the DDR2/mDDR datasheet. Calculate by: T_RASMAX = (trasmax/refresh_rate) - 1. Round down to the nearest cycle. |
| 26-25 | T_XP | 0-3h | Specifies the minimum number of DDR_CLK cycles from Power Down exit to any other command except a read command, minus 1. Corresponds to the txp or tcke AC timing parameter in the DDR2/mDDR data sheet. This field must satisfy the greater of $t_{XP}$ or $t_{CKE}$. If $t_{XP} > t_{CKE}$, then calculate by T_XP = $t_{XP}$ - 1. If $t_{XP} < t_{CKE}$, then calculate by T_XP = $t_{CKE}$ - 1. |
| 24-23 | _RESV | 0-3h | Any writes to these bit(s) must always have a value of 0. |
| 22-16 | T_XSNR | 0-7Fh | Specifies the minimum number of DDR_CLK cycles from a self_refresh exit to any other command except a read command, minus 1. Corresponds to the txsnr AC timing parameter in the DDR2/mDDR data sheet. Calculate by: T_XSNR = (txsnr/DDR_CLK) - 1 |
| 15-8 | T_XSRD | 0-FFh | Specifies the minimum number of DDR_CLK cycles from a self_refresh exit to a read command, minus 1. Corresponds to the txsrd AC timing parameter in the DDR2/mDDR data sheet. Calculate by: T_XSRD = txsrd - 1 |
| 7-5 | T_RTP | 0-7h | Specifies the minimum number of DDR_CLK cycles from a last read command to a precharge command, minus 1. Corresponds to the trtp AC timing parameter in the DDR2/mDDR data sheet. Calculate by: T_RTP = (trtp/DDR_CLK) - 1 |
| 4-0 | T_CKE | 0-1Fh | Specifies the minimum number of DDR_CLK cycles between transitions on the DDR_CKE pin, minus 1. Corresponds to the tcke AC timing parameter in the DDR2/mDDR data sheet. Calculate by: T_CKE = tcke - 1 |

## 4.6 SDRAM Configuration Register 2 (SDCR2)

The SDRAM configuration register 2 (SDCR2) contains fields to configure partial array self refresh and rowsize of the mDDR. SDCR2 is shown in Figure 25 and described in Table 28. Writing to the PASR and ROWSIZE bit fields will cause the DDR2/mDDR memory controller to start the DDR2/mDDR SDRAM initialization sequence. This register is applicable only when IBANK in SDCR is set to '1' for special addressing. This register is for mDDR.

**Figure 25. SDRAM Configuration 2 (SDCR2) Register**

| 31–19 | 18–16 |
|---|---|
| Reserved | PASR |
| N-0 | RW-0 |

| 15–3 | 2–0 |
|---|---|
| Reserved | ROWSIZE |
| N-0 | RW-0 |

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

**Table 28. SDRAM Configuration 2 (SDCR2) Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-19 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 18-16 | PASR | | Partial array self refresh. 3_4h Reserved 7h Reserved |
| | | 0 | 4 banks will be refreshed |
| | | 1 | 2 banks will be refreshed |
| | | 2 | 1 bank will be refreshed |
| | | 5 | Half of one bank will be refreshed |
| | | 6 | One fourth of one bank will be refreshed |
| 15-3 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 2-0 | ROWSIZE | | Row size. Defines the number of row address bit for DDR device. |
| | | 0 | 9 row address bits |
| | | 1 | 10 row address bits |
| | | 2 | 11 row address bits |
| | | 3 | 12 row address bits |
| | | 4 | 13 row address bits |
| | | 5 | 14 row address bits |
| | | 6 | 15 row address bits |
| | | 7 | 16 row address bits |

### 4.7 Peripheral Bus Burst Priority Register (PBBPR)

The peripheral bus burst priority register (PBBPR) helps prevent command starvation within the DDR2/mDDR memory controller. To avoid command starvation, the DDR2/mDDR memory controller momentarily raises the priority of the oldest command in the command FIFO after a set number of transfers have been made. The PR_OLD_COUNT bit sets the number of transfers that must be made before the DDR2/mDDR memory controller raises the priority of the oldest command. The PBBPR is shown in Figure 26 and described in Table 29. See Section 2.7.2 for more details on command starvation.

**Note:** Due to an internal DM36x hardware issue, you must not accept the default bit field value for PR_OLD_COUNT. The default bit field value for PR_OLD_COUNT is 0xFF, you must change the value of PR_OLD_COUNT to a value other than 0xFF.

**Figure 26. Peripheral Bus Burst Priority Register (PBBPR)**

| 31–16 |
|---|
| Reserved |
| R-0 |

| 15–8 | 7–0 |
|---|---|
| Reserved | PR_OLD_COUNT |
| R-0 | R/W-FFh |

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

**Table 29. Peripheral Bus Burst Priority Register (PBBPR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 7-0 | PR_OLD_COUNT | 0-FFh | Priority raise old counter. Specifies the number of memory transfers after which the DDR2/mDDR memory controller will elevate the priority of the oldest command in the command FIFO. This bit field must not be set to 0xFF. |

## 4.8 Interrupt Raw Register (IRR)

The interrupt raw register (IRR) displays the raw status of the interrupt. If the interrupt condition occurs, the corresponding bit in IRR is set independent of whether or not the interrupt is enabled. The IRR is shown in Figure 27 and described in Table 30.

**Figure 27. Interrupt Raw Register (IRR)**

| 31–16 |
|---|
| Reserved |
| R-0 |

| 15–3 | 2 | 1–0 |
|---|---|---|
| Reserved | LT | Reserved |
| R-0 | R/W1C-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear (writing 0 has no effect); -*n* = value after reset

**Table 30. Interrupt Raw Register (IRR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 2 | LT | | Line trap. Write a 1 to clear LT and the LTM bit in the interrupt masked register (IMR); a write of 0 has no effect. |
| | | 0 | A line trap condition has not occurred. |
| | | 1 | Illegal memory access type. See Section 2.15 for more details. |
| 1-0 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |

### 4.9 Interrupt Masked Register (IMR)

The interrupt masked register (IMR) displays the status of the interrupt when it is enabled. If the interrupt condition occurs and the corresponding bit in the interrupt mask set register (IMSR) is set, then the IMR bit is set. The IMR bit is not set if the interrupt is not enabled in IMSR. The IMR is shown in Figure 28 and described in Table 31.

**Figure 28. Interrupt Masked Register (IMR)**

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

| 15-3 | 2 | 1-0 |
|---|---|---|
| Reserved | LTM | Reserved |
| R-0 | R/W1C-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear (writing 0 has no effect); -*n* = value after reset

**Table 31. Interrupt Masked Register (IMR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 2 | LTM | | Line trap masked. Write a 1 to clear LTM and the LT bit in the interrupt raw register (IRR); a write of 0 has no effect. |
| | | 0 | A line trap condition has not occurred. |
| | | 1 | Illegal memory access type (only set if the LTMSET bit in IMSR is set). See Section 2.15 for more details. |
| 1-0 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |

## 4.10 Interrupt Mask Set Register (IMSR)

The interrupt mask set register (IMSR) enables the DDR2/mDDR memory controller interrupt. The IMSR is shown in Figure 29 and described in Table 32.

**Note:** If the LTMSET bit in IMSR is set concurrently with the LTMCLR bit in the interrupt mask clear register (IMCR), the interrupt is not enabled and neither bit is set to 1.

**Figure 29. Interrupt Mask Set Register (IMSR)**

| 31–16 |
|---|
| Reserved |
| R-0 |

| 15–3 | 2 | 1–0 |
|---|---|---|
| Reserved | LTMSET | Reserved |
| R-0 | R/W-0 | R-0 |

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

**Table 32. Interrupt Mask Set Register (IMSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 2 | LTMSET | | Line trap interrupt set. Write a 1 to set LTMSET and the LTMCLR bit in the interrupt mask clear register (IMCR); a write of 0 has no effect. |
| | | 0 | Line trap interrupt is not enabled; a write of 1 to the LTMCLR bit in IMCR occurred. |
| | | 1 | Line trap interrupt is enabled. |
| 1-0 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |

### 4.11 Interrupt Mask Clear Register (IMCR)

The interrupt mask clear register (IMCR) disables the DDR2/mDDR memory controller interrupt. Once an interrupt is enabled, it may be disabled by writing a 1 to the IMCR bit. The IMCR is shown in Figure 30 and described in Table 33.

**Note:** If the LTMCLR bit in IMCR is set concurrently with the LTMSET bit in the interrupt mask set register (IMSR), the interrupt is not enabled and neither bit is set to 1.

**Figure 30. Interrupt Mask Clear Register (IMCR)**

| 31–16 |
|---|
| Reserved |
| R-0 |

| 15–3 | 2 | 1–0 |
|---|---|---|
| Reserved | LTMCLR | Reserved |
| R-0 | R/W1C-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear (writing 0 has no effect); -*n* = value after reset

**Table 33. Interrupt Mask Clear Register (IMCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 2 | LTMCLR | | Line trap interrupt clear. Write a 1 to clear LTMCLR and the LTMSET bit in the interrupt mask set register (IMSR); a write of 0 has no effect. |
| | | 0 | Line trap interrupt is not enabled. |
| | | 1 | Line trap interrupt is enabled; a write of 1 to the LTMSET bit in IMSR occurred. |
| 1-0 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |

## 4.12 DDR PHY Control Register (DDRPHYCR1)

The DDR PHY control register 1 (DDRPHYCR1) configures the DDR2/mDDR memory controller read latency. The DDRPHYCR1 is shown in Figure 31 and described in Table 34.

**Figure 31. DDR PHY Control Register 1 (DDRPHYCR1)**

| 31–17 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15–8 | 7 | 6 | 5–3 | 2–0 |
|---|---|---|---|---|
| Reserved | CONFIG_EXT_STRBEN | CONFIG_PWRDNEN | Reserved | READ_LATENCY |
| R-0 | RW-1 | RW-1 | R-0 | R/W-7 |

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

**Table 34. DDR PHY Control 1 Register (DDRPHYCR1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | | Any writes to these bit(s) must always have a value of 0. |
| 7 | CONFIG_EXT_STRBEN | | Internal/External strobe gating select |
| | | 0 | Select internal strobe gating |
| | | 1 | Select external strobe gating |
| 6 | CONFIG_PWRDNEN | | Power down receivers. |
| | | 0 | Receivers powered up |
| | | 1 | Receivers powered down |
| 5-3 | Reserved | | Any writes to these bit(s) must always have a value of 0. |
| 2-0 | READ_LATENCY | | Read latency. Read latency is equal to CAS latency plus round trip board delay for data minus 1. The maximum value of read latency that is supported is CAS latency plus 2. The minimum read latency value that is supported is CAS latency plus 1. The read latency value is defined in number of MCLK/DDR_CLK cycles. |

### 4.13 VTP IO Control Register (VTPIOCR)

The VTP IO control register (VTPIOCR) is used to control the calibration of the DDR2/mDDR memory controller IOs with respect to voltage, temperature, and process (VTP). The voltage, temperature, and process information is used to control the IO's output impedance. The VTPIOCR is shown in Figure 32 and described in Table 35. Note that the VTPIOCR resides in the register space of the System Control module.

**Figure 32. VTP IO Control Register (VTPIOCR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R+000 0000 0000 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | DLLRSTZ | CLKRSTZ | VREFEN | VREFTAP | |
| R+000 0000 0000 | | | R/W+1 | R/W+1 | R/W+0 | R/W+00 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| READY | IOPWRDN | CLRZ | FORCEDNP | FORCEDNN | FORCEUPP | FORCEUPN | PWRSAVE |
| R+0 | R/W+0 | R/W+0 | R/W+0 | R/W+0 | R/W+0 | R/W+0 | R/W+0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LOCK | PWRDN | D0 | D1 | D2 | F0 | F1 | F2 |
| R/W+0 | R/W+1 | R/W+1 | R/W+1 | R/W+0 | R/W+1 | R/W+1 | R/W+1 |

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

**Table 35. VTP IO (VTPIOCR) Control Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-21 | Reserved | 0 | Any writes to these bit(s) must always have a value of 0. |
| 20 | DLLRSTZ | 1 | Active low reset is used to reset the DLL. This is for test use and should code '1' in normal operation. |
| 19 | CLKRSTZ | 1 | Active low reset is used to reset the clock divider of DDR address/data macro. This is for test use and should code '1' in normal operation. |
| 18 | VREFEN | | Internal DDR IO Vref enable |
| | | 0 | Connected to pad, external reference |
| | | 1 | Connected to internal reference |
| 17-16 | VREFTAP | | Selection for internal reference voltage level |
| | | 00 | Vref = 50.0% of VDDS |
| | | 01 | Vref = 47.5% of VDDS |
| | | 10 | Vref = 52.5% of VDDS |
| | | 11 | Vref = 50.0% of VDDS |
| 15 | READY | | VTP Ready status |
| | | 0 | VTP not ready |
| | | 1 | VTP ready |
| 14 | IOPWRDN | | Power down enable for DDR input buffer |
| | | 0 | Disable power down control by config_pwrdnen register |
| | | 1 | Enable power down control by config_pwrdnen register |
| 13 | CLRZ | 0 | VTP clear. Write 0 to clear VTP flops |
| 12 | FORCEDNP | 0 | Force decrease PFET drive |
| 11 | FORCEDNN | 0 | Force decrease NFET drive |
| 10 | FORCEUPP | 0 | Force increase PFET drive |

**Table 35. VTP IO (VTPIOCR) Control Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 9 | FORCEUPN | 0 | Force increase PFET drive |
| 8 | PWRSAVE | | VTP power save mode |
| | | 0 | Disable power save mode |
| | | 1 | Enable power save mode |
| 7 | LOCK | | VTP impedance lock |
| | | 0 | Unlock impedance |
| | | 1 | Lock impedance |
| 6 | PWRDN | | VTP power down |
| | | 0 | Disable power down |
| | | 1 | Enable power down |
| 5 | D0 | 1 | Drive strength control bit |
| 4 | D1 | 1 | Drive strength control bit |
| 3 | D2 | 0 | Drive strength control bit |
| 2 | F0 | 1 | Digital filter control bit |
| 1 | F1 | 1 | Digital filter control bit |
| 0 | F2 | 1 | Digital filter control bit |