

# TMS320DM357 DMSoC DDR2 Memory Controller

## User's Guide



Literature Number: SPRUG38  
November 2008



---



---

<b>Preface</b> .....	<b>7</b>
<b>1 Introduction</b> .....	<b>10</b>
1.1 Purpose of the Peripheral .....	10
1.2 Features .....	10
1.3 Functional Block Diagram .....	10
1.4 Supported Use Case Statement .....	11
1.5 Industry Standard(s) Compliance Statement .....	11
<b>2 Peripheral Architecture</b> .....	<b>11</b>
2.1 Clock Control .....	11
2.2 Memory Map .....	13
2.3 Signal Descriptions .....	13
2.4 Protocol Description(s) .....	14
2.5 Memory Width and Byte Alignment .....	22
2.6 Address Mapping .....	23
2.7 DDR2 Memory Controller Interface .....	26
2.8 Refresh Scheduling .....	29
2.9 Self-Refresh Mode .....	29
2.10 Reset Considerations .....	30
2.11 VTP IO Buffer Calibration .....	30
2.12 Auto-Initialization Sequence .....	31
2.13 Interrupt Support .....	33
2.14 DMA Event Support .....	34
2.15 Power Management .....	34
2.16 Emulation Considerations .....	35
<b>3 Supported Use Cases</b> .....	<b>36</b>
3.1 Connecting the DDR2 Memory Controller to DDR2 Memory .....	36
3.2 Configuring Memory-Mapped Registers to Meet DDR2-400 Specification .....	36
<b>4 DDR2 Memory Controller Registers</b> .....	<b>41</b>
4.1 SDRAM Status Register (SDRSTAT) .....	41
4.2 SDRAM Bank Configuration Register (SDBCR) .....	42
4.3 SDRAM Refresh Control Register (SDRCR) .....	44
4.4 SDRAM Timing Register (SDTIMR) .....	45
4.5 SDRAM Timing Register 2 (SDTIMR2) .....	46
4.6 Peripheral Bus Burst Priority Register (PBBPR) .....	47
4.7 Interrupt Raw Register (IRR) .....	48
4.8 Interrupt Masked Register (IMR) .....	49
4.9 Interrupt Mask Set Register (IMSR) .....	50
4.10 Interrupt Mask Clear Register (IMCR) .....	51
4.11 DDR PHY Control Register (DDRPHYCR) .....	52
4.12 VTP IO Control Register (VTPIOCR) .....	53
4.13 DDR VTP Register (DDRVTTPR) .....	54

---

4.14	DDR VTP Enable Register (DDRVTPER).....	55
------	---	----

## List of Figures

1	Data Paths to DDR2 Memory Controller .....	11
2	DDR2 Memory Controller Clock Block Diagram .....	12
3	DDR2 Memory Controller Signals .....	13
4	Refresh Command .....	16
5	DCAB Command .....	17
6	DEAC Command .....	18
7	ACTV Command .....	19
8	DDR2 READ Command .....	20
9	DDR2 WRT Command .....	21
10	DDR2 MRS and EMRS Command .....	22
11	Byte Alignment.....	23
12	Logical Address-to-DDR2 SDRAM Address Map.....	25
13	DDR2 SDRAM Column, Row, and Bank Access .....	26
14	DDR2 Memory Controller FIFO Block Diagram .....	27
15	DDR2 Memory Controller Reset Block Diagram .....	30
16	DDR2 Memory Controller Power and Sleep Controller Diagram .....	34
17	Connecting DDR2 Memory Controller for 32-Bit Connection.....	37
18	Connecting DDR2 Memory Controller for 16-Bit Connection.....	37
19	SDRAM Status Register (SDRSTAT).....	41
20	SDRAM Bank Configuration Register (SDBCR) .....	42
21	SDRAM Refresh Control Register (SDRCR) .....	44
22	SDRAM Timing Register (SDTIMR) .....	45
23	SDRAM Timing Register 2 (SDTIMR2).....	46
24	Peripheral Bus Burst Priority Register (PBBPR) .....	47
25	Interrupt Raw Register (IRR) .....	48
26	Interrupt Masked Register (IMR) .....	49
27	Interrupt Mask Set Register (IMSR) .....	50
28	Interrupt Mask Clear Register (IMCR).....	51
29	DDR PHY Control Register (DDRPHYCR) .....	52
30	VTP IO Control Register (VTPIOCR) .....	53
31	DDR VTP Register (DDRVTPR) .....	54
32	DDR VTP Enable Register (DDRVTPER) .....	55

## List of Tables

1	PLL2 Configuration.....	12
2	DDR2 Memory Controller Signal Descriptions .....	14
3	DDR2 SDRAM Commands .....	14
4	Truth Table for DDR2 SDRAM Commands .....	15
5	Addressable Memory Ranges .....	22
6	Bank Configuration Register Fields for Address Mapping .....	23
7	Logical Address-to-DDR2 SDRAM Address Map for 32-Bit SDRAM .....	24
8	Logical Address-to-DDR2 SDRAM Address Map for 16-bit SDRAM.....	24
9	DDR2 Memory Controller FIFO Description .....	26
10	Refresh Urgency Levels .....	29
11	Reset Sources .....	30
12	DDR2 SDRAM Configuration by MRS Command .....	32
13	DDR2 SDRAM Configuration by EMRS(1) Command .....	32
14	SDBCR Configuration .....	38
15	DDR2 Memory Refresh Specification .....	38
16	SDRCR Configuration .....	38
17	SDTIMR Configuration.....	39
18	SDTIMR2 Configuration .....	39
19	DDRPHYCR Configuration.....	40
20	DDR2 Memory Controller Registers .....	41
21	SDRAM Status Register (SDRSTAT) Field Descriptions .....	41
22	SDRAM Bank Configuration Register (SDBCR) Field Descriptions .....	42
23	SDRAM Refresh Control Register (SDRCR) Field Descriptions .....	44
24	SDRAM Timing Register (SDTIMR) Field Descriptions .....	45
25	SDRAM Timing Register 2 (SDTIMR2) Field Descriptions .....	46
26	Peripheral Bus Burst Priority Register (PBBPR) Field Descriptions.....	47
27	Interrupt Raw Register (IRR) Field Descriptions .....	48
28	Interrupt Masked Register (IMR) Field Descriptions.....	49
29	Interrupt Mask Set Register (IMSR) Field Descriptions .....	50
30	Interrupt Mask Clear Register (IMCR) Field Descriptions.....	51
31	DDR PHY Control Register (DDRPHYCR) Field Descriptions .....	52
32	VTP IO Control Register (VTPIOCR) Field Descriptions.....	53
33	DDR VTP Register (DDRVTPR) Field Descriptions .....	54
34	DDR VTP Enable Register (DDRVTPER) Field Descriptions .....	55

## Read This First

---

---

---

### About This Manual

This document describes the DDR2 memory controller in the TMS320DM357 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 TMS320DM357 Digital Media System-on-Chip (DMSoC). Copies of these documents are available on the Internet at [www.ti.com](http://www.ti.com). *Tip:* Enter the literature number in the search box provided at [www.ti.com](http://www.ti.com).

**[SPRUG06](#)** — ***TMS320DM357 DMSoC Video Processing Back End (VPBE) User's Guide***. Describes the video processing back end (VPBE) in the TMS320DM357 Digital Media System-on-Chip (DMSoC) video processing subsystem. Included in the VPBE is the video encoder, on-screen display, and digital LCD controller.

**[SPRUG25](#)** — ***TMS320DM357 DMSoC ARM Subsystem Reference Guide***. Describes the ARM subsystem in the TMS320DM357 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 video processing subsystem, and a majority of the peripherals and external memories.

**[SPRUG26](#)** — ***TMS320DM357 DMSoC Universal Asynchronous Receiver/Transmitter (UART) User's Guide***. This document describes the universal asynchronous receiver/transmitter (UART) peripheral in the TMS320DM357 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.

**[SPRUG27](#)** — ***TMS320DM357 DMSoC Inter-Integrated Circuit (I2C) Peripheral User's Guide***. Describes the inter-integrated circuit (I2C) peripheral in the TMS320DM357 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. External components attached to this 2-wire serial bus can transmit and receive up to 8-bit wide data to and from the DMSoC through the I2C peripheral. This document assumes the reader is familiar with the I2C-bus specification.

- [SPRUG28](#)** — ***TMS320DM357 DMSoC 64-Bit Timer User's Guide***. Describes the operation of the software-programmable 64-bit timer in the TMS320DM357 Digital Media System-on-Chip (DMSoC). Timer 0 and Timer 1 are used as general-purpose (GP) timers and can be programmed in 64-bit mode, dual 32-bit unchained mode, or dual 32-bit chained mode; Timer 2 is used only as a watchdog timer. The GP timer modes can be used to generate periodic interrupts or enhanced direct memory access (EDMA) synchronization events. The watchdog timer mode is used to provide a recovery mechanism for the device in the event of a fault condition, such as a non-exiting code loop.
- [SPRUG29](#)** — ***TMS320DM357 DMSoC Serial Peripheral Interface (SPI) User's Guide***. Describes the serial peripheral interface (SPI) in the TMS320DM357 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.
- [SPRUG30](#)** — ***TMS320DM357 DMSoC Host Port Interface (HPI) Reference Guide***. This document describes the host port interface in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The HPI provides a parallel port interface through which an external host processor can directly access the TMS320DM357 DMSoC processor's resources (configuration and program/data memories).
- [SPRUG31](#)** — ***TMS320DM357 DMSoC General-Purpose Input/Output (GPIO) User's Guide***. Describes the general-purpose input/output (GPIO) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an input, you can detect the state of the input by reading the state of an internal register. When configured as an output, you can write to an internal register to control the state driven on the output pin.
- [SPRUG32](#)** — ***TMS320DM357 DMSoC Multimedia Card (MMC)/Secure Digital (SD) Card Controller User's Guide***. Describes the multimedia card (MMC)/secure digital (SD) card controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The MMC/SD card is used in a number of applications to provide removable data storage. The MMC/SD controller provides an interface to external MMC and SD cards. The communication between the MMC/SD controller and MMC/SD card(s) is performed by the MMC/SD protocol.
- [SPRUG33](#)** — ***TMS320DM357 DMSoC Asynchronous External Memory Interface (EMIF) User's Guide***. Describes the asynchronous external memory interface (EMIF) in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EMIF supports a glueless interface to a variety of external devices.
- [SPRUG34](#)** — ***TMS320DM357 DMSoC Enhanced Direct Memory Access (EDMA) Controller User's Guide***. Describes the operation of the enhanced direct memory access (EDMA3) controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EDMA3 controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the DMSoC.
- [SPRUG35](#)** — ***TMS320DM357 DMSoC Audio Serial Port (ASP) User's Guide***. Describes the operation of the audio serial port (ASP) audio interface in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The primary audio modes that are supported by the ASP are the AC97 and IIS modes. In addition to the primary audio modes, the ASP supports general serial port receive and transmit operation, but is not intended to be used as a high-speed interface.
- [SPRUG36](#)** — ***TMS320DM357 DMSoC Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) Module User's Guide***. Discusses the ethernet media access controller (EMAC) and physical layer (PHY) device management data input/output (MDIO) module in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EMAC controls the flow of packet data from the DMSoC to the PHY. The MDIO module controls PHY configuration and status monitoring.

**[SPRUG37](#) — *TMS320DM357 DMSoC Pulse-Width Modulator (PWM) Peripheral User's Guide.***

Describes the pulse-width modulator (PWM) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

**[SPRUG38](#) — *TMS320DM357 DMSoC DDR2 Memory Controller User's Guide.*** Describes the DDR2 memory controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The DDR2 memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM devices.

**[SPRUG39](#) — *TMS320DM357 DMSoC Video Processing Front End (VPFE) User's Guide.*** Describes the video processing front end (VPFE) in the TMS320DM357 Digital Media System-on-Chip (DMSoC) video processing subsystem. Included in the VPFE is the preview engine, CCD controller, resizer, histogram, and hardware 3A (H3A) statistic generator.

**[SPRUGH2](#) — *TMS320DM357 DMSoC Peripherals Overview Reference Guide.*** This document provides an overview of the peripherals in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

**[SPRUGH3](#) — *TMS320DM357 DMSoC Universal Serial Bus Controller User's Guide.*** This document describes the universal serial bus (USB) controller in the TMS320DM357 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.

## Trademarks

## **DDR2 Memory Controller**

---

---

---

### **1 Introduction**

This document describes the DDR2 memory controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

#### **1.1 Purpose of the Peripheral**

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

#### **1.2 Features**

The DDR2 memory controller supports the following features:

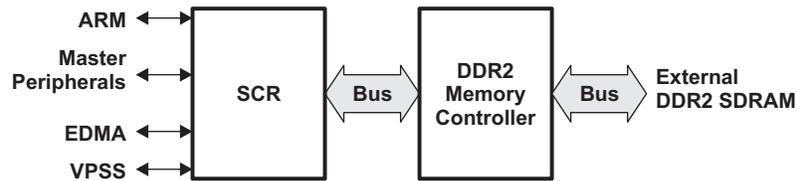
- JESD79D-2A standard compliant DDR2 SDRAM
- 256 Mbyte memory space
- Data bus width of 32 or 16 bits
- CAS latencies: 2, 3, 4, and 5
- Internal banks: 1, 2, 4, and 8
- Burst length: 8
- Burst type: sequential
- 1 CS signal
- Page sizes: 256, 512, 1024, and 2048
- SDRAM autoinitialization
- Self-refresh mode
- Prioritized refresh
- Programmable refresh rate and backlog counter
- Programmable timing parameters
- Little endian

#### **1.3 Functional Block Diagram**

The DDR2 memory controller is the main interface to external DDR2 memory. [Figure 1](#) displays the general data paths to on-chip peripherals and external DDR2 SDRAM.

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

Figure 1. Data Paths to DDR2 Memory Controller



## 1.4 Supported Use Case Statement

The DDR2 memory controller supports JESD79D-2A DDR2-400 SDRAM memories utilizing either 32-bit or 16-bit of the DDR2 memory controller data bus. See [Section 3](#) for more details.

## 1.5 Industry Standard(s) Compliance Statement

The DDR2 memory controller is compliant with the JESD79D-2A DDR2 SDRAM standard with the exception of the following feature list:

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

## 2 Peripheral Architecture

This section describes the architecture of the DDR2 memory controller as well as how it is structured and how it works within the context of the system-on-a-chip. The DDR2 memory controller can gluelessly interface to most standard DDR2 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 memory controller to perform read and write operations to externally-connected DDR2 SDRAM devices. Also, [Section 3](#) provides a detailed example of interfacing the DDR2 memory controller to a common DDR2 SDRAM device.

### 2.1 Clock Control

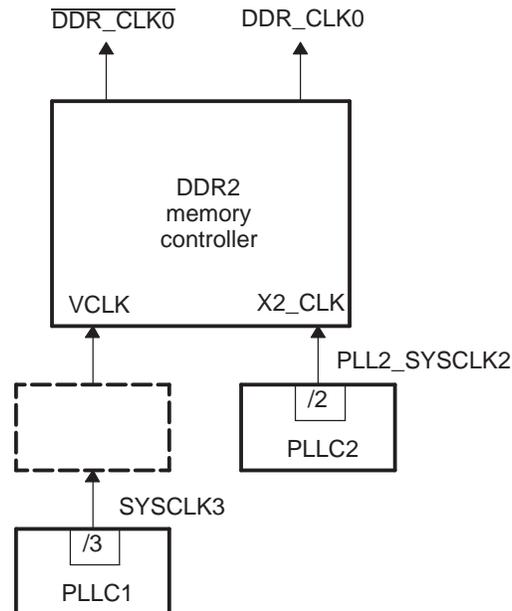
The DDR2 memory controller receives two input clocks from internal clock sources, SYSCLK3 and PLL2\_SYSCLK2 ([Figure 2](#)). PLL2\_SYSCLK2 should be configured to clock at the frequency of the desired data rate, or stated similarly, it should operate at twice the frequency of the desired DDR2 memory clock. DDR\_CLK0 and  $\overline{\text{DDR\_CLK0}}$  are the two output clocks of the DDR2 memory controller providing the interface clock to the DDR2 SDRAM memory. These two clocks operate at a frequency of  $\text{PLL2\_SYSCLK2}/2$ .

#### 2.1.1 Clock Source

SYSCLK3 and PLL2\_SYSCLK2 are sourced from two independent PLLs ([Figure 2](#)). SYSCLK3 is sourced from PLL controller 1 (PLLC1) and PLL2\_SYSCLK2 is sourced from PLL controller 2 (PLLC2).

SYSCLK3 is clocked at a fixed divider ratio of PLL1. This divider is fixed at 3, meaning SYSCLK3 is clocked at a frequency of  $\text{PLL1}/3$ . Once inside the DDR2 memory controller, this signal is called VCLK.

PLLC2 has a programmable divider that is used to divide-down the output clock of PLL2. This divider should be configured such that PLLC2 supplies the PLL2\_SYSCLK2 at the desired frequency. For example, if a 150-MHZ DDR2 interface clock (DDR\_CLK0) is desired, then PLLC2 must be configured to generate a 300-MHZ clock on PLL2\_SYSCLK2. Once inside the DDR2 memory controller, PLL2\_SYSCLK2 is called X2\_CLK.

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


### 2.1.2 Clock Configuration

The frequency of **PLL2\_SYSCLK2** is configured by selecting the appropriate PLL multiplier and divider ratio. The PLL multiplier and divider ratio are selected by programming registers within **PLL2**. [Table 1](#) shows a list of PLL multiplier and divider settings to achieve certain DDR2 frequencies. The data in [Table 1](#) is derived by assuming a 27-MHz reference clock. See the *TMS320DMxxxx DMSoC ARM Subsystem Reference Guide* (spru856) for information on the PLL controller.

---

**Note:** **PLL2** should be configured and a stable clock present on **PLL2\_SYSCLK2** before releasing the DDR2 memory controller from reset.

---

**Table 1. PLL2 Configuration**

PLL Multiplier	PLL Frequency (MHZ)	Divider Ratio	PLL2_SYSCLK2 Frequency (MHZ)	DDR2 Clock Frequency (MHZ)
28	756	3	252	126
19	513	2	256.6	128.3
29	783	3	261	130.5
20	540	2	270	135
31	837	3	279	139.5
21	567	2	283.5	141.8
32	864	3	288	144
22	594	2	297	148.5
23	621	2	310	155.3
24	648	2	324	162

### 2.1.3 DDR2 Memory Controller Internal Clock Domains

There are two clock domains within the DDR2 memory controller. The two clock domains are driven by VCLK and a divided-down by 2 version of PLL2\_SYSCLK2 called MCLK. The command FIFO, write FIFO, and read FIFO described in [Section 2.7](#) are all on the VCLK domain. From this, you can see that VCLK drives the interface to the peripheral bus.

The MCLK domain consists of the DDR2 memory controller state machine and memory-mapped registers. This clock domain is clocked at the rate of the external DDR2 memory, PLL2\_SYSCLK2/2.

To conserve power within the DDR2 memory controller, VCLK, MCLK, and PLL2\_SYSCLK2 may be stopped. See [Section 2.15](#) for proper clock stop procedures.

## 2.2 Memory Map

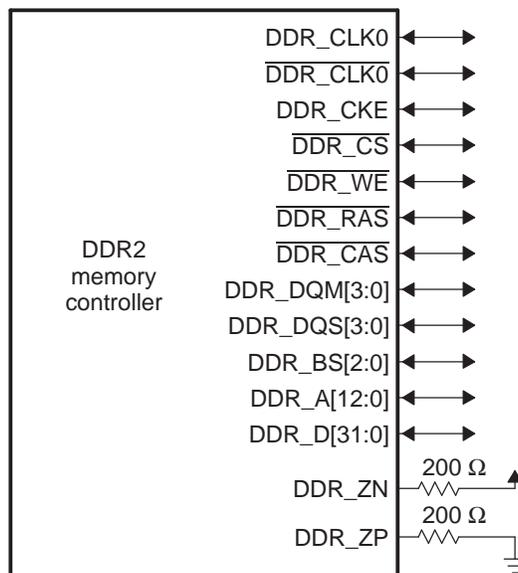
See the device-specific data manual for information describing the device memory-map.

## 2.3 Signal Descriptions

The DDR2 memory controller signals are shown in [Figure 3](#) and described in [Table 2](#). The following features are included:

- The maximum data bus is 32-bits wide.
- The address bus is 13-bits wide with an additional 3 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 Memory Controller Signals**



**Table 2. DDR2 Memory Controller Signal Descriptions**

Pin	Type	Description
DDR_CLK0, DDR_CLK0	O/Z	<b>Clock:</b> Differential clock outputs.
DDR_CKE	O/Z	<b>Clock enable:</b> Active high.
$\overline{\text{DDR\_CS}}$	O/Z	<b>Chip select:</b> Active low.
$\overline{\text{DDR\_WE}}$	O/Z	<b>Write enable strobe:</b> Active low, command output.
$\overline{\text{DDR\_RAS}}$	O/Z	<b>Row address strobe:</b> Active low, command output.
$\overline{\text{DDR\_CAS}}$	O/Z	<b>Column address strobe:</b> Active low, command output.
DDR_DQM[3:0]	O/Z	<b>Data mask:</b> Output mask signal for write data.
DDR_DQS[3:0]	I/O/Z	<b>Data strobe:</b> Active high, bi-directional signals. Output with write data, input with read data.
DDR_BS[2:0]	O/Z	<b>Bank select:</b> Output, defining which bank a given command is applied.
DDR_A[12:0]	O/Z	<b>Address:</b> Address bus.
DDR_D[31:0]	I/O/Z	<b>Data:</b> Bi-directional data bus. Input for read data, output for write data.
DDR_ZN, DDR_ZP	O	<b>Output impedance control:</b> Required to set the DDR2 output impedance. Connected by way of a 200-ohm resistor to power and ground (see <a href="#">Figure 3</a> ). The resistor should be chosen to be 4 times the desired impedance of the output buffer. By changing the size of the resistor, the DDR2 outputs can be tuned to match the board load, if necessary.

## 2.4 Protocol Description(s)

The DDR2 memory controller supports the DDR2 SDRAM commands listed in [Table 3](#). [Table 4](#) shows the signal truth table for the DDR2 SDRAM commands.

**Table 3. DDR2 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.

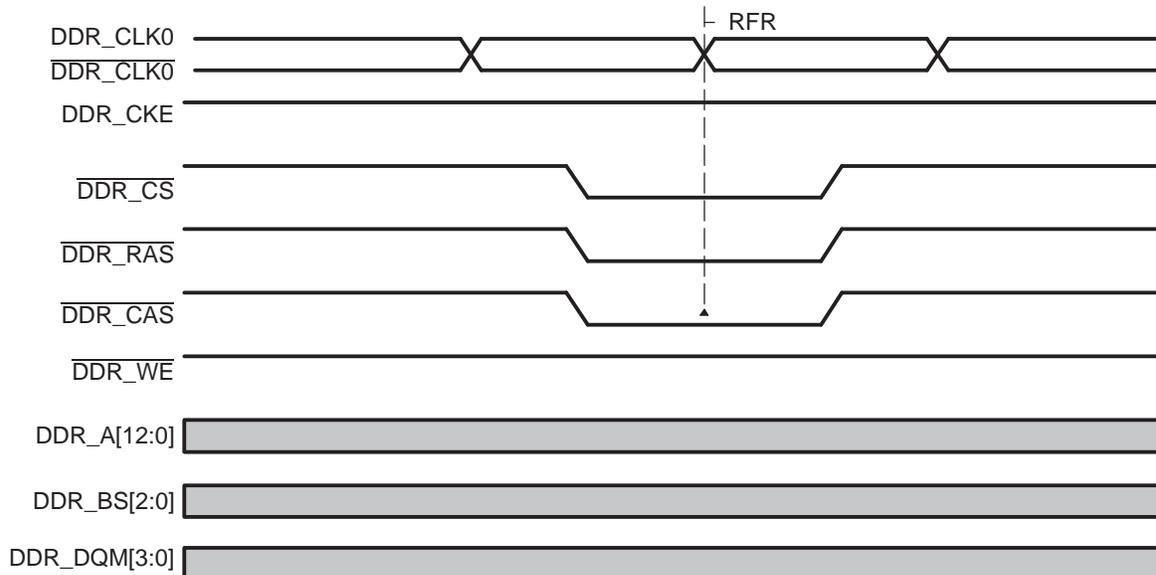
**Table 4. Truth Table for DDR2 SDRAM Commands**

DDR2 SDRAM:	CKE		$\overline{\text{CS}}$	$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	$\overline{\text{WE}}$	BA[2:0]	A[12:11, 9:0]	A10
DDR2 memory controller:	DDR_CKE		$\overline{\text{DDR\_CS}}$	$\overline{\text{DDR\_RAS}}$	$\overline{\text{DDR\_CAS}}$	$\overline{\text{DDR\_WE}}$	DDR_BS[2:0]	DDR_A[12:11, 9:0]	DDR_A[10]
	Previous Cycles	Current Cycle							
ACTV	H	H	L	L	H	H	Bank	Row Address	
DCAB	H	H	L	L	H	L	X	X	L
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	L
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
			L	H	H	H	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 memory controller issues refresh commands to the DDR2 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 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 memory controller. See Section 2.8 for more details on REFR command scheduling.

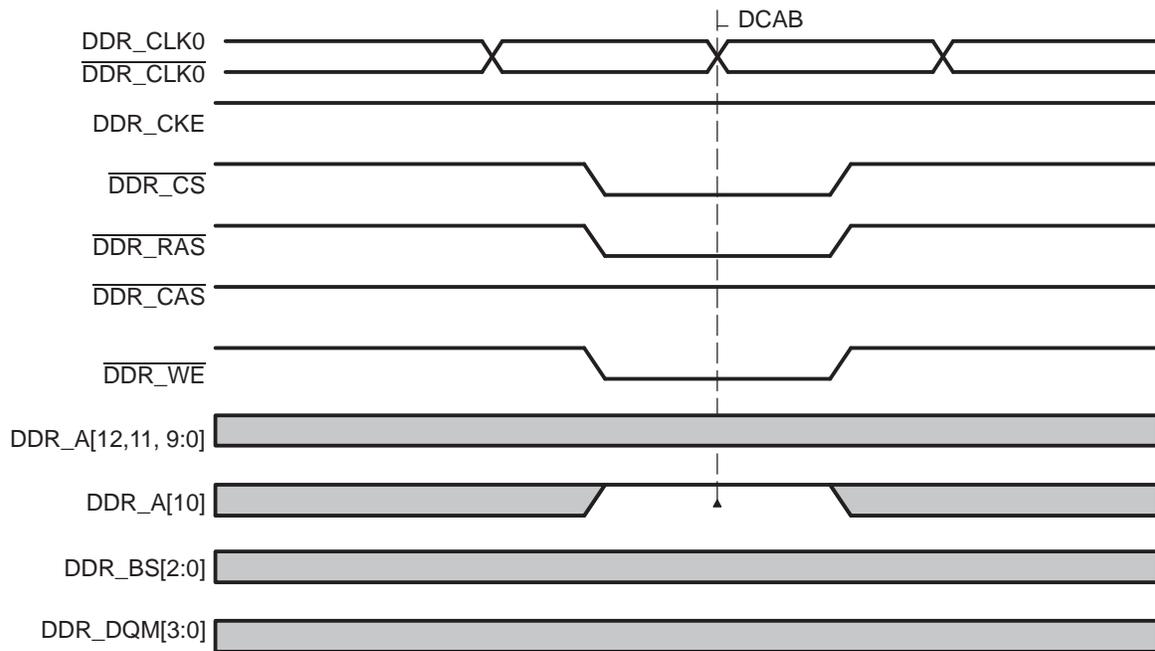
**Figure 4. Refresh Command**



### 2.4.2 Deactivation (DCAB and DEAC)

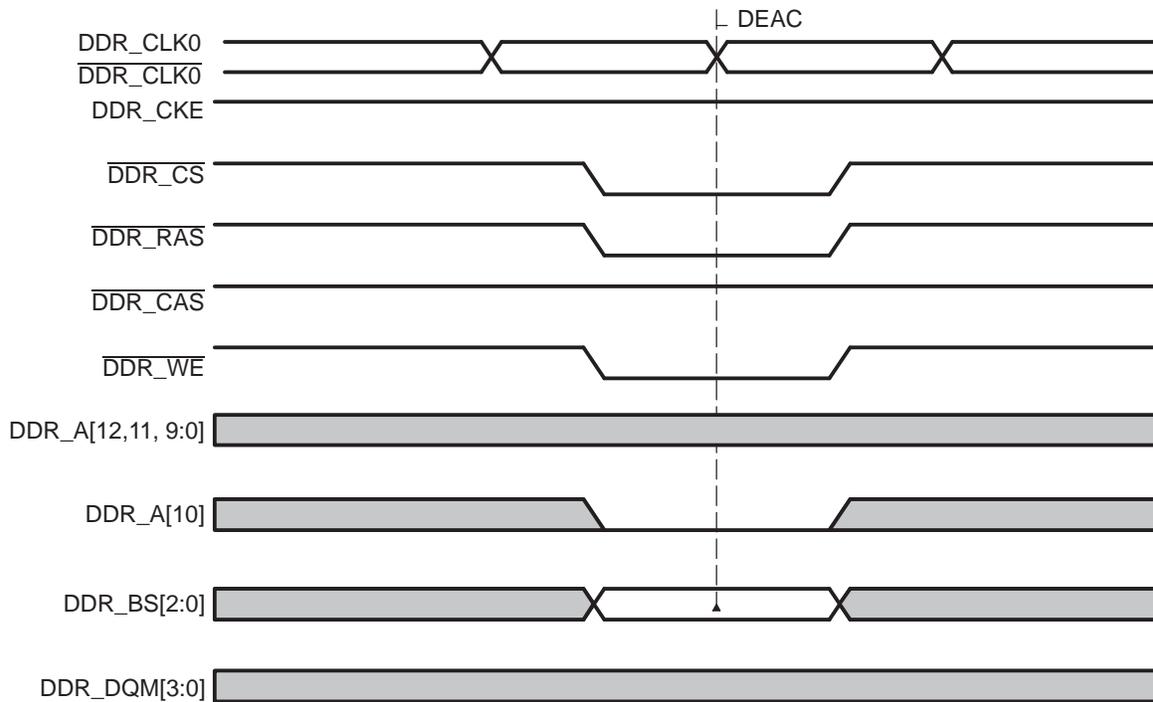
The precharge all banks command (DCAB) is performed after a reset to the DDR2 memory controller or following the initialization sequence. DDR2 SDRAMs also require this cycle prior to a refresh (REFR) and mode set register commands (MRS and EMRS). During a DCAB command, DDR\_A[10] is driven high to ensure 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.

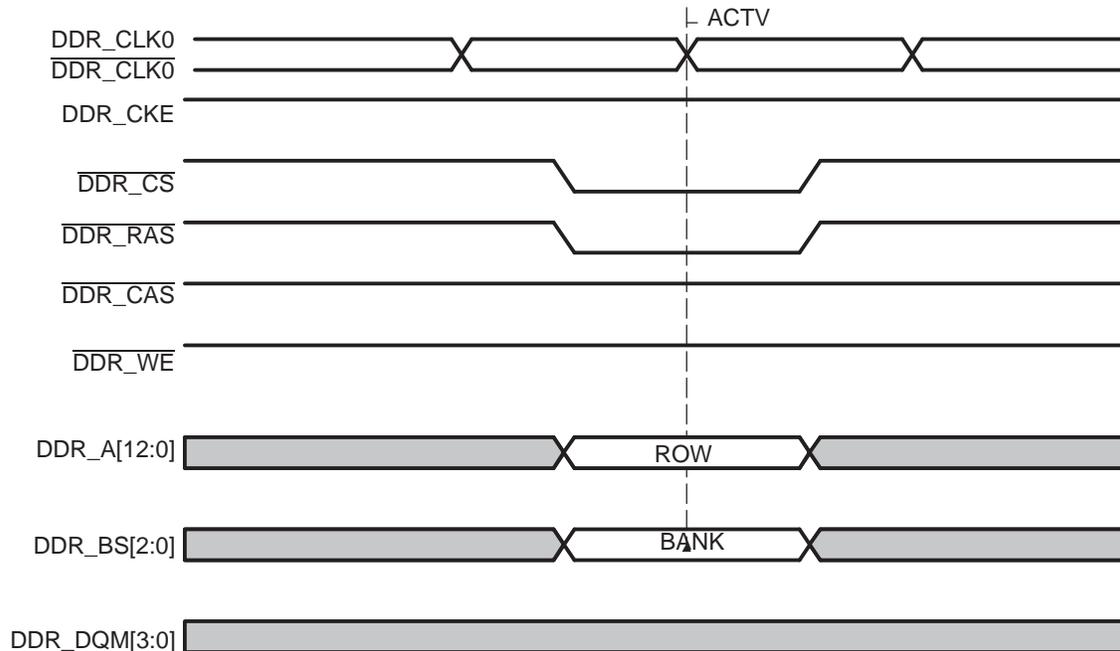
**Figure 6. DEAC Command**



### 2.4.3 Activation (ACTV)

The DDR2 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\_BS[2:0] selects the bank and the value of A[12:0] selects the row. When the DDR2 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

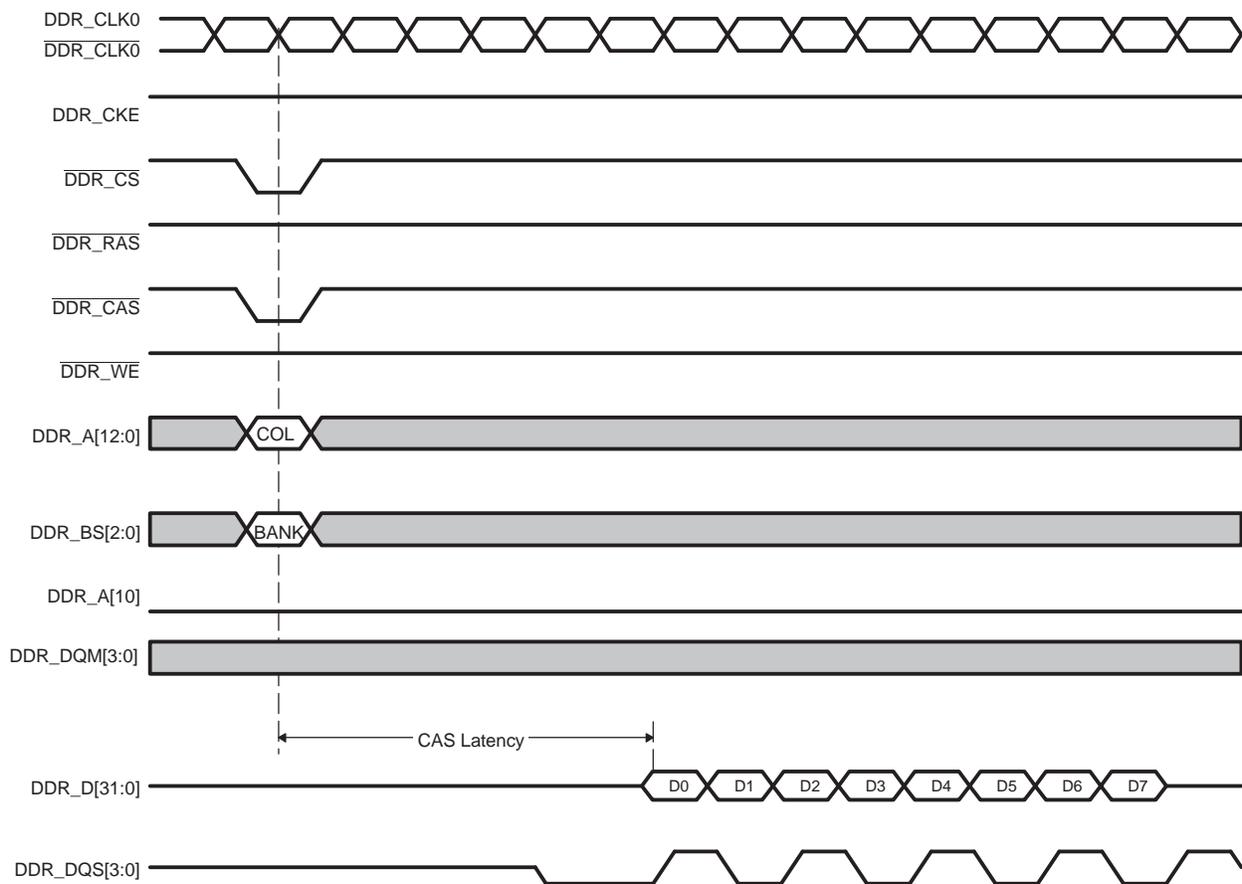


### 2.4.4 READ Command

Figure 8 shows the DDR2 memory controller performing a read burst from DDR2 SDRAM. The READ command initiates a burst read operation to an active row. During the READ command,  $\overline{\text{DDR\_CAS}}$  drives low,  $\overline{\text{DDR\_WE}}$  and  $\overline{\text{DDR\_RAS}}$  remain high, the column address is driven on  $\text{DDR\_A}[12:0]$ , and the bank address is driven on  $\text{DDR\_BS}[2:0]$ .

The DDR2 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 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 memory controller returns 8 pieces of data for every read command. If additional accesses are not pending to the DDR2 memory controller, the read burst completes and the unneeded data is disregarded. If additional accesses are pending, depending on the scheduling result, the DDR2 memory controller can terminate the read burst and start a new read burst. Furthermore, the DDR2 memory controller does not issue a DAB/DEAC command until page information becomes invalid.

**Figure 8. DDR2 READ Command**



### 2.4.5 Write (WRT) Command

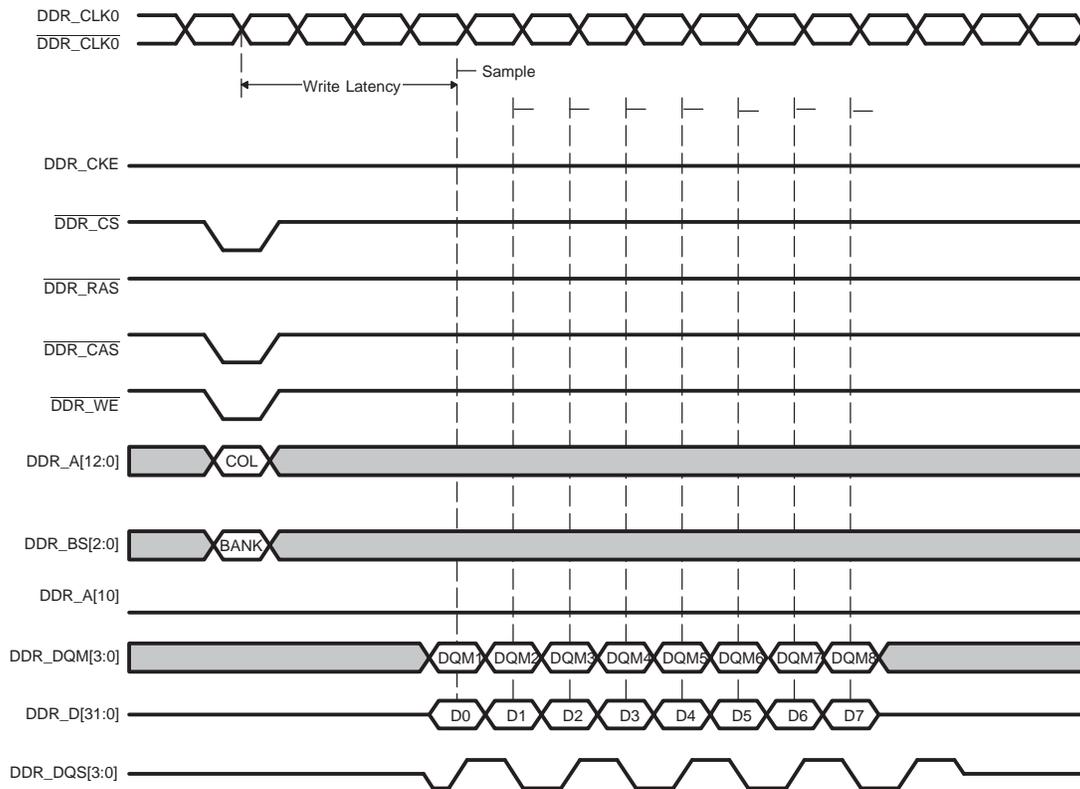
Prior to a WRT command, the desired bank and row are activated by the ACTV command. Following the WRT command, a write latency is incurred. Write latency is equal to CAS latency minus 1. All writes have a burst length of 8. The use of the DDR\_DQM outputs allows byte and halfword writes to be executed. [Figure 9](#) shows the timing for a write on the DDR2 memory controller.

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

- Mask out the additional data using DDR\_DQM outputs
- Terminate the write burst and start a new write burst

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

**Figure 9. DDR2 WRT Command**



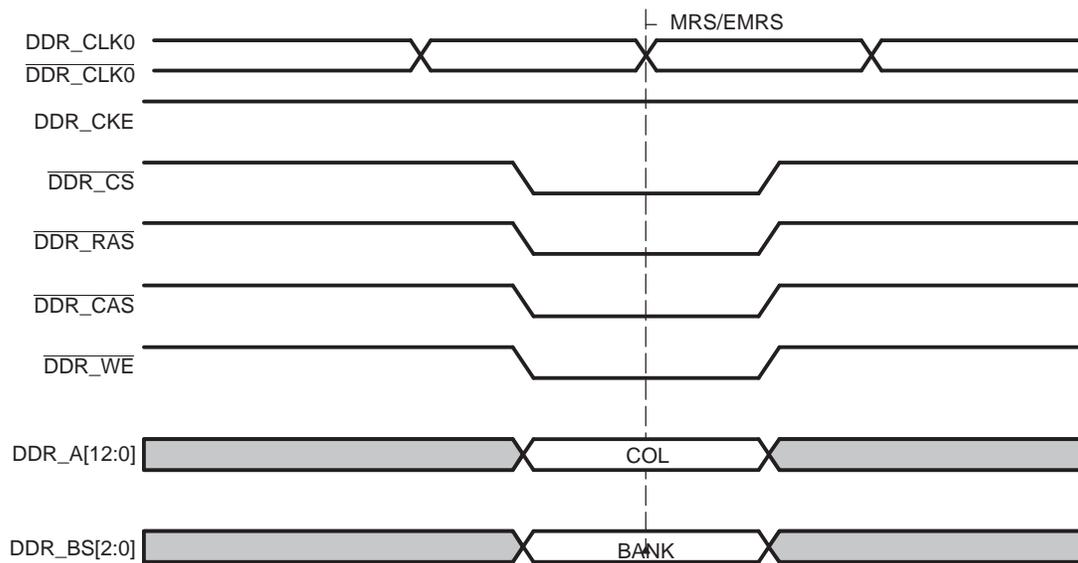
### 2.4.6 Mode Register Set (MRS and EMRS)

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

The DDR2 memory controller programs the mode and extended mode registers of the DDR2 memory by issuing MRS and EMRS commands. When the MRS or EMRS command is executed, the value on DDR\_BS[1:0] selects the mode register to be written and the data on DDR\_A[12:0] is loaded into the register. [Figure 10](#) shows the timing for an MRS and EMRS command.

The DDR2 memory controller only issues MRS and EMRS commands during the DDR2 memory controller initialization sequence. See [Section 2.12](#) for more information.

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



### 2.5 Memory Width and Byte Alignment

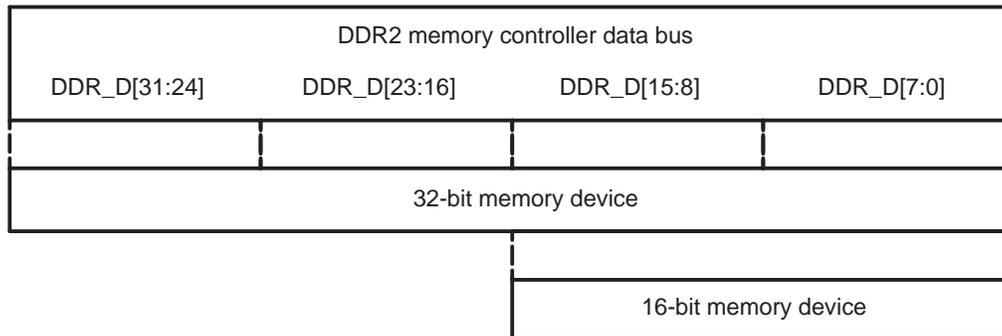
The DDR2 memory controller supports memory widths of 16 bits and 32 bits. [Table 5](#) summarizes the addressable memory ranges on the DDR2 memory controller. Only little-endian format is supported.

[Figure 11](#) shows the byte lanes used on the DDR2 memory controller. The external memory is always right aligned on the data bus.

**Table 5. Addressable Memory Ranges**

Memory Width	Maximum addressable bytes per CS space	Description
×16	128 Mbytes	Halfword address
×32	256 Mbytes	Word address

**Figure 11. Byte Alignment**



## 2.6 Address Mapping

The DDR2 memory controller views external DDR2 SDRAM as one continuous block of memory. This statement is true regardless of the number of external physical devices mapped to a given chip select space. The DDR2 memory controller receives DDR2 memory access requests along with a 32-bit logical address from the rest of the system. In turn, the DDR2 memory controller uses the logical address to generate a row/page, column, and bank address for the DDR2 SDRAM. The number of column and bank address bits used is determined by the IBANK and PAGESIZE fields in the SDRAM bank configuration register (SDBCR) (see [Table 6](#)).

**Table 6. Bank Configuration Register Fields for Address Mapping**

Bit Field	Bit Value	Bit Description
IBANK	0	1 bank
	1h	2 banks
	2h	4 banks
	3h	8 banks
PAGESIZE	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)

As stated in [Table 6](#), the IBANK and PAGESIZE fields of SDBCR control the mapping of the logical, source address of the DDR2 memory controller to the DDR2 SDRAM row, column, and bank address bits. The DDR2 memory controller logical address always contains 13 row address bits, whereas the number of column and bank bits are determined by the IBANK and PAGESIZE fields. [Table 7](#) and [Table 8](#) show how the logical address bits map to the DDR2 SDRAM row, column, and bank bits for combinations of IBANK and PAGESIZE values. The same DDR2 memory controller pins provide the row and column address to the DDR2 SDRAM, thus the DDR2 memory controller appropriately shifts the address during row and column address selection.

[Figure 12](#) shows how this address-mapping scheme organizes the DDR2 SDRAM rows, columns, and banks into the device memory map. Note that during a linear access, the DDR2 memory controller increments the column address as the logical address increments. When the DDR2 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 SDRAM, this process looks as shown in [Figure 13](#).

By traversing across banks while remaining on the same row/page, the DDR2 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 memory controller never opens more than one page per bank.

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

**Table 7. Logical Address-to-DDR2 SDRAM Address Map for 32-Bit SDRAM**

SBCR Bit		Logical Address																		
IBANK	PAGESIZE	31	30	29	28	27	26	25	24	23	22:16	15	14	13	12	11	10	9:2	1:0	
0	0	-										nrb=13						ncb=8		
1	0	-										nrb=13						nbb=1	ncb=8	
2h	0	-										nrb=13						nbb=2	ncb=8	
3h	0	-										nrb=13						nbb=3	ncb=8	
0	1	-										nrb=13						ncb=9		
1	1	-										nrb=13						nbb=1	ncb=9	
2h	1	-										nrb=13						nbb=2	ncb=9	
3h	1	-										nrb=13						nbb=3	ncb=9	
0	2h	-										nrb=13						ncb=10		
1	2h	-										nrb=13						nbb=1	ncb=10	
2h	2h	-										nrb=13						nbb=2	ncb=10	
3h	2h	-										nrb=13						nbb=3	ncb=10	
0	3h	-										nrb=13						ncb=11		
1	3h	-										nrb=13						nbb=1	ncb=11	
2h	3h	-										nrb=13						nbb=2	ncb=11	
3h	3h	-										nrb=13						nbb=3	ncb=11	

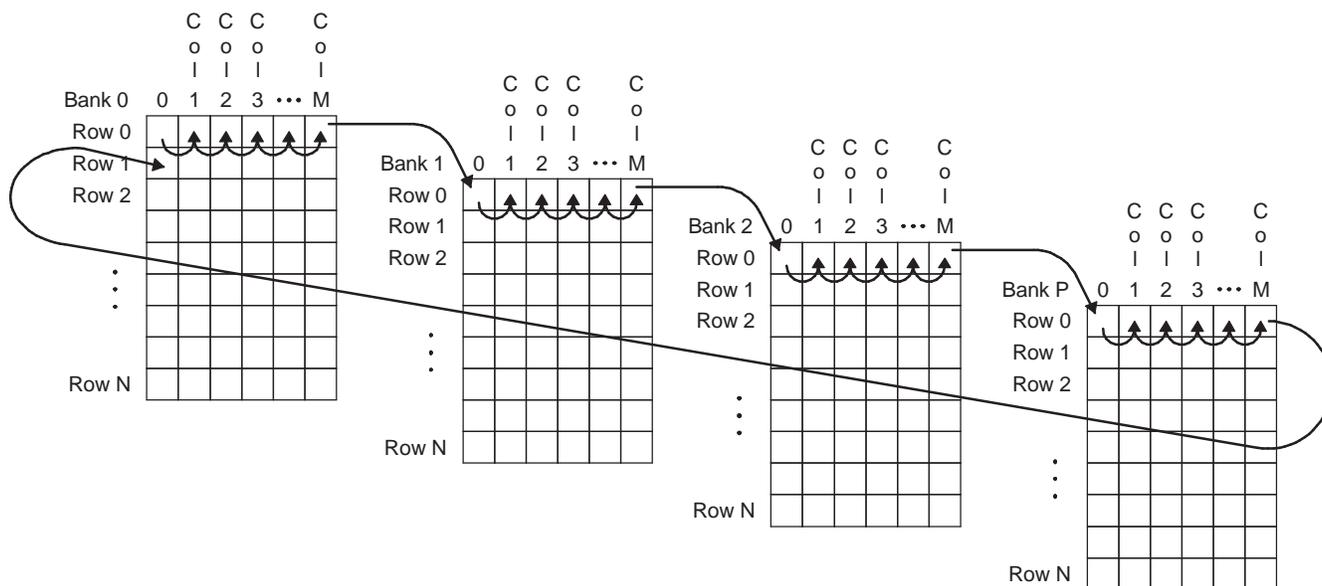
**Table 8. Logical Address-to-DDR2 SDRAM Address Map for 16-bit SDRAM**

SBCR 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=13						ncb=8		
1	0	-										nrb=13						nbb=1	ncb=8	
2h	0	-										nrb=13						nbb=2	ncb=8	
3h	0	-										nrb=13						nbb=3	ncb=8	
0	1	-										nrb=13						ncb=9		
1	1	-										nrb=13						nbb=1	ncb=9	
2h	1	-										nrb=13						nbb=2	ncb=9	
3h	1	-										nrb=13						nbb=3	ncb=9	
0	2h	-										nrb=13						ncb=10		
1	2h	-										nrb=13						nbb=1	ncb=10	
2h	2h	-										nrb=13						nbb=2	ncb=10	
3h	2h	-										nrb=13						nbb=3	ncb=10	
0	3h	-										nrb=13						ncb=11		
1	3h	-										nrb=13						nbb=1	ncb=11	
2h	3h	-										nrb=13						nbb=2	ncb=11	
3h	3h	-										nrb=13						nbb=3	ncb=11	

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

Col. 0	Col. 1	Col. 2	Col. 3	Col. 4	...	Col. M-1	Col. M	
					...			Row 0, bank 0
					...			Row 0, bank 1
					...			Row 0, bank 2
•	•	•	•	•	...	•	•	•
•	•	•	•	•	...	•	•	•
•	•	•	•	•	...	•	•	•
					...			Row 0, bank P
					...			Row 1, bank 0
					...			Row 1, bank 1
					...			Row 1, bank 2
•	•	•	•	•	...	•	•	•
•	•	•	•	•	...	•	•	•
•	•	•	•	•	...	•	•	•
					...			Row 1, bank P
					...			•
					...			•
					...			•
					...			Row N, bank 0
					...			Row N, bank 1
					...			Row N, bank 2
•	•	•	•	•	...	•	•	•
•	•	•	•	•	...	•	•	•
•	•	•	•	•	...	•	•	•
					...			Row N, bank P

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 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.

## 2.7 DDR2 Memory Controller Interface

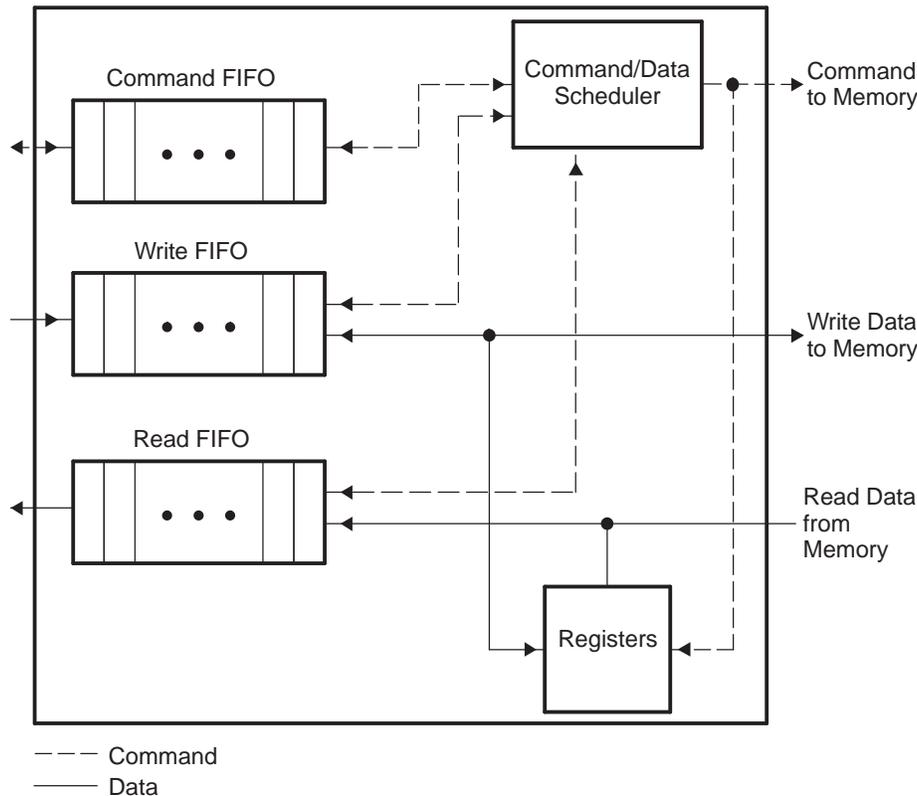
To move data efficiently from on-chip resources to external DDR2 SDRAM memory, the DDR2 memory controller makes use of a command FIFO, a write FIFO, a read FIFO, and command and data schedulers. [Table 9](#) describes the purpose of each FIFO.

[Figure 14](#) shows the block diagram of the DDR2 memory controller FIFOs. Commands, write data, and read data arrive at the DDR2 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 9. DDR2 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 14. DDR2 Memory Controller FIFO Block Diagram



### 2.7.1 Command Ordering and Scheduling, Advanced Concept

The DDR2 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 SDRAM rows. Command re-ordering takes place within the command FIFO.

Typically, a given master issues commands on a single priority. EDAM transfer controller read and write ports are different masters. The DDR2 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 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 memory controller selects the oldest command.

The DDR2 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 memory controller also issues refresh commands. The DDR2 memory controller attempts to delay refresh commands as long as possible to maximize performance while meeting the SDRAM refresh requirements. As the DDR2 memory controller issues read, write, and refresh commands to DDR2 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 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 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 memory controller can momentarily raise the priority of the oldest command in the command FIFO after a set number of transfers have been made. The PR\_OLD\_COUNT bit field in the peripheral bus burst priority register (PBBPR) sets the number of the transfers that must be made before the DDR2 memory controller will raise the priority of the oldest command.

---

**Note:** Leaving the PR\_OLD\_COUNT bits at their default value (FFh) disables this feature of the EMIF. This means commands can stay in the command FIFO indefinitely. Therefore, these bits should be set to FEh immediately following reset to enable this feature with the highest level of allowable memory transfers. It is suggested that system level prioritization be set to avoid placing high-bandwidth masters on the highest priority levels. These bits can be left as FEh unless advanced bandwidth/prioritization control is required.

---

### 2.7.3 Possible Race Condition

A race condition may exist when certain masters write data to the DDR2 memory controller. For example, if master A passes a software message via a buffer in DDR2 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 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 memory controller SDRAM Status register.
3. Perform a dummy read to the DDR2 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 and ATA peripherals do not need to implement the above workaround. If a peripheral is not listed here, then the above workaround is required. Refer to the device-specific data manual for more information.

## 2.8 Refresh Scheduling

The DDR2 memory controller issues autorefresh (REFR) commands to DDR2 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 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 memory controller currently has outstanding.

The DDR2 memory controller issues REFR commands based on the level of urgency. The level of urgency is defined in [Table 10](#). Whenever the refresh level of urgency is reached, the DDR2 memory controller issues a REFR command before servicing any new memory access requests. Following a REFR command, the DDR2 memory controller waits T<sub>RFC</sub> 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 memory controller issues a REFR command. When this backlog counter is greater than 7, the DDR2 memory controller issues four REFR commands before servicing any new memory requests.

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

**Table 10. 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 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 memory controller may service new memory access requests.
Refresh Need	Backlog count is greater than 7. Indicates the DDR2 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 memory controller should perform a REFR command before servicing new memory access requests.

## 2.9 Self-Refresh Mode

Setting the self refresh (SR) bit in the SDRAM refresh control register (SDRCR) to 1 forces the DDR2 memory controller to place the external DDR2 SDRAM in a low-power mode (self refresh), in which the DDR2 SDRAM maintains valid data while consuming a minimal amount of power. When the SR bit is asserted, the DDR2 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 self-refresh (SLFRFR) command (an autorefresh command with DDR\_CKE low) is issued.

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

After exiting from the self-refresh state, the DDR2 memory controller will not immediately start executing commands. Instead, it will wait T<sub>SXNR</sub>+1 clock cycles before issuing non-read commands and T<sub>SXRD</sub>+1 clock cycles before issuing read commands. The SDRAM timing 2 register (SDTIM2) programs the values of T<sub>SXNR</sub> and T<sub>SXRD</sub>.

Once in self-refresh mode, the DDR2 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 2.15](#) for more information describing the proper procedure to follow when shutting down DDR2 memory controller input clocks.

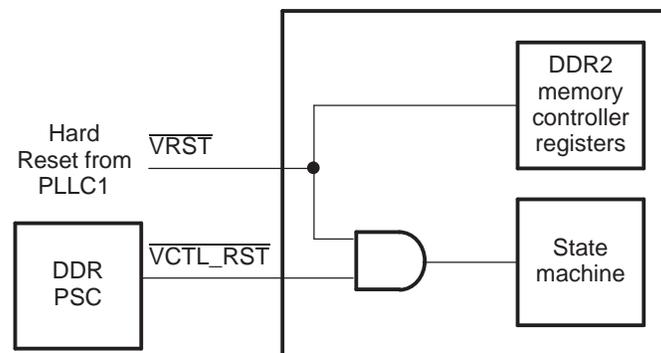
## 2.10 Reset Considerations

The DDR2 memory controller has two reset signals,  $\overline{VRST}$  and  $\overline{VCTL\_RST}$ . The  $\overline{VRST}$  is a module-level reset that resets both the state machine as well as the DDR2 memory controller memory-mapped registers. The  $\overline{VCTL\_RST}$  resets the state machine only. If the DDR2 memory controller is reset independently of other peripherals, the user's software should not perform memory, as well as register accesses, while  $\overline{VRST}$  or  $\overline{VCTL\_RST}$  are asserted. If memory or register accesses are performed while the DDR2 memory controller is in the reset state, other masters may hang. Following the rising edge of  $\overline{VRST}$  or  $\overline{VCTL\_RST}$ , the DDR2 memory controller immediately begins its initialization sequence. Command and data stored in the DDR2 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, see the *TMS320DMxxx DMSoC ARM Subsystem Reference Guide* (spru856). [Figure 15](#) shows the DDR2 memory controller reset diagram.

**Table 11. Reset Sources**

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

**Figure 15. DDR2 Memory Controller Reset Block Diagram**



## 2.11 VTP IO Buffer Calibration

The DDR2 memory controller is able to control the impedance of the output IO. This feature allows the DDR2 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 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 resistors connected to the DDR\_ZN and DDR\_ZP pins. The resistor should be chosen to be 4 times the desired impedance of the output IO. The DDR2 reference design requires the resistor values to be 200 ohms. This means that both the DDR\_ZN and DDR\_ZP pins must have a 200 ohm resistor connected to them. [Figure 3](#) describes proper connection of the DDR\_ZN and DDR\_ZP pins.

To set the output impedance of the IO, calibration must be initiated by writing to the following memory mapped registers:

- VTP IO Control Register
- DDR VTP Enable Register
- DDR VTP Register

The VTP IO Control Register is written to begin the calibration. Once the calibration is complete, the VTP information is stored in the DDR VTP Register. The DDR VTP Register should then be read, retrieving the VTP information, and the VTP information written to the VTP IO Control Register. The DDR VTP Enable Register is written to enable/disable access to the DDR VTP Register. Steps 8-15 of the initialization procedure described in [Section 2.12.2](#) shows the procedure that must be followed to perform VTP IO calibration.

---

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

---

## 2.12 Auto-Initialization Sequence

The DDR2 SDRAM contains mode and extended mode registers that configure the DDR2 memory for operation. These registers control burst type, burst length, CAS latency, DLL enable/disable (on the DDR2 device), single-ended strobe, etc. The DDR2 memory controller programs the mode and extended mode registers of the DDR2 memory by issuing MRS and EMRS commands during the initialization sequence. The initialization sequence performed by the DDR2 memory controller is compliant with the JESDEC79-2A specification. The DDR2 memory controller performs an initialization sequence under the following conditions:

- Following reset (rising edge of  $\overline{VRST}$  or  $\overline{VCTL\_RST}$ )
- Following a write to the DDRDRIVE bit field or the two least-significant bytes in the SDRAM bank configuration register (SDBCR)

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

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

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

DDR2 Memory Controller Address Bus	Value	DDR2 SDRAM Register Bit	DDR2 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 bank configuration register (SDBCR).
DDR_A[3]	0	3	Burst Type	Sequential
DDR_A[2:0]	3h	2:0	Burst Length	8

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

DDR2 Memory Controller Address Bus	Value	DDR2 SDRAM Register Bit	DDR2 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]	1	1	Output Driver Impedance	DDR2 drive strength programmed to weak (60%).
DDR_A[0]	0	0	DLL enable	DLL enable

### 2.12.1 Initializing Configuration Registers

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

1. Program the DDR PHY control register (DDRPHYCR) by setting the read latency (READLAT) bits to the desired value as well as clearing the DLLPWRDN bit to 0.
2. Program the SDRAM bank configuration register (SDBCR) to the desired value with the TIMUNLOCK bit set to 1 (unlocked).
3. Program the SDRAM timing register (SDTIMR) and SDRAM timing register 2 (SDTIMR2) to the desired values to meet the DDR2 SDRAM memory data sheet specification.
4. Program SDBCR to the desired value with the TIMUNLOCK bit cleared to 0 (locked).
5. Program the RR bit in the SDRAM refresh control register (SDRCR) to the desired value to meet the refresh requirements of the DDR2 SDRAM memory.

## 2.12.2 Initializing Following Device Power Up and Device RESET

### CAUTION

**The following power up sequence is preliminary and is documented to reflect the intended-use case. This power-up sequence may change at a future date.**

Following device power up, the DDR2 memory controller is held in reset with the internal clocks to the module gated off. Before releasing the DDR2 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 PLLC2 registers to provide a stable clock on PLL2\_SYSCLK2 at the desired frequency.
2. Program the DDR2 memory controller Power and Sleep Controller (PSC) to enable VCLK.
3. Follow the register initialization procedure described in [Section 2.12.1](#) to complete the DDR2 memory controller configuration.
4. Perform a dummy read of DDR2 memory to verify initialization sequence has completed.
5. Perform a soft reset to the DDR2 memory controller via the PSC. For details on programming the PSC, see the *TMS320DMxxx DMSoC ARM Subsystem Reference Guide* (spru856).
6. Enable VTP manual calibration by writing to the VTP IO control register (VTPIOCR). See [Section 4.12](#) for details on VTPIOCR.
  - a. With a single write, set the EN bit field (bit 13) to 1 and the RECAL bit field (bit 15) to 0 by writing a value of 0000 201Fh.
  - b. Set the RECAL bit field (bit 15) to 1, making sure the value written to the EN field is still 1 by writing a value of 0000 A01Fh. This begins the calibration sequence.
7. Wait for a minimum of 33 VTP clk cycles for calibration to complete. The VTP clock operates at 13.5 MHz.
8. Enable access to the DDR VTP register by writing a 1 to the DDR VTP enable register.
9. Read the DDR VTP register to get the P/N channel VTP value. See [Section 4.13](#) for details on the DDR VTP register.
10. Write the VTP information to the PCH and NCH fields in the VTPIOCR. Make sure the RECAL and EN fields remain set to 1.
11. Write 0 to EN bit in the VTPIOCR to disable VTP calibration.
12. Disable access to the DDR VTP register by writing a 0 to the DDR VTP enable register.
13. Disable VTP input clock by disabling the bypass clock of PLL2.

---

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

---

## 2.13 Interrupt Support

The DDR2 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 memory controller generates an interrupt by setting the LT bit in the interrupt raw register (IRR). The DDR2 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 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 not muxed with other interrupts.

## 2.14 DMA Event Support

The DDR2 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.

## 2.15 Power Management

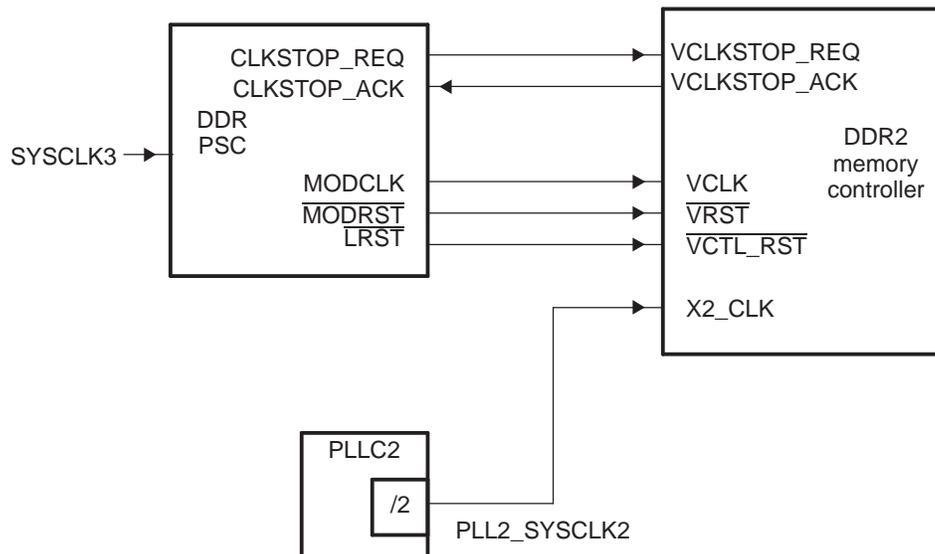
Power dissipation from the DDR2 memory controller may be managed by two methods:

- Self-refresh mode (see [Section 2.9](#))
- Gating input clocks to the module off

Gating input clocks off to the DDR2 memory controller achieves higher power savings when compared to the power savings of self-refresh mode. The input clocks are turned off outside of the DDR2 memory controller through the use of the Power and Sleep Controller (PSC) and the PLL controller 2 (PLL2). [Figure 16](#) shows the connections between the DDR2 memory controller, PSC, and PLL2. For detailed information on power management procedures using the PSC, see the *TMS320DMxxx DMSoC ARM Subsystem Reference Guide* (spru856).

Before gating clocks off, the DDR2 memory controller must place the DDR2 SDRAM memory in self-refresh mode by setting the SR bit in the SDRAM refresh control register (SDRCR) to 1. If the external memory requires a continuous clock, the DDR2 memory controller clock provided by PLL2 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 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 16. DDR2 Memory Controller Power and Sleep Controller Diagram**



### 2.15.1 DDR2 Memory Controller Clock Stop Procedure

---

**Note:** If an access occurs to the DDR2 memory controller 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 step 6, all DDR2 accesses are disabled until the DDR2 memory controller is enabled again through the LPSC.

---

To achieve maximum power savings VCLK, MCLK, X2\_CLK, DDR\_CLK0, and  $\overline{\text{DDR\_CLK0}}$  should be gated off, as well as the DDR2 memory controller DLL powered down. Perform the following procedure when shutting down clocks to achieve maximum power savings:

1. Allow software to complete the desired DDR transfers.
2. Set the SR bit in the DDR2 SDRAM refresh control register (SDRCR). The DDR2 memory controller will complete any outstanding accesses and backlogged refresh cycles and then place the external DDR2 memory in self-refresh mode.
3. Set the MCLKSTOPEN bit in SDCR. This enables the DDR2 memory controller to shut off the MCLK.
4. Set the DLLPWRDN bit in the DDR PHY control register (DDRPHYCR) to 1 to power down the DDR2 memory controller DLL.
5. Poll the PHYRDY bit in the SDRAM status register (SDRSTAT) to be a logic-low indicating that the MCLK has been stopped and the DLL is powered down.
6. Program DDR2 memory controller LPSC to disable VCLK.
7. Program PLLC2 registers to stop PLL2\_SYSCLK2 which disables X2\_CLK of the DDR2 memory controller, as well as DDR\_CLK0 and  $\overline{\text{DDR\_CLK0}}$ .

To turn clocks back on:

1. Program PLLC2 registers to start PLL2\_SYSCLK2 which sources X2\_CLK of the DDR2 memory controller.
2. Once PLL2\_SYSCLK2 is stable, program the DDR2 memory controller LPSC to enable VCLK.
3. Clear the SR bit in the DDR2 SDRAM refresh control register (SDRCR) to 0.
4. Clear the MCLKSTOPEN bit in SDCR to 0.
5. Clear the DLLPWRDN bit in the DDR PHY control register (DDRPHYCR) to 0 to power up the DDR2 memory controller DLL.
6. Poll the PHYRDY bit in the SDRAM status register (SDRSTAT) to be a logic-high indicating that the MCLK is running and the DLL is powered up.
7. Perform a soft reset to the DDR2 memory controller via the PSC. For details on programming the PSC, see the *TMS320DMxxxx DMSoC ARM Subsystem Reference Guide* (spru856).

### 2.16 Emulation Considerations

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

### 3 Supported Use Cases

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

This section presents an example describing how to interface the DDR2 memory controller to a JESD79D DDR2-400 1-Gb device. The DDR2 memory controller is assumed to be operating at 133 MHz.

#### 3.1 *Connecting the DDR2 Memory Controller to DDR2 Memory*

The following figures show how to connect the DDR2 memory controller to a DDR2 device. [Figure 17](#) displays a 32-bit interface; therefore, two 16-bit DDR2 devices are connected to the DDR2 memory controller. From [Figure 17](#), you can see that the data bus, data strobe, and data mask (byte enable) signals are point-to-point where as all other address, control, and clocks are not. [Figure 18](#) displays a 16-bit interface; therefore, all signals are point-to-point.

#### 3.2 *Configuring Memory-Mapped Registers to Meet DDR2-400 Specification*

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

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

In addition to these registers, the DDR PHY control register (DDRPHYCR) must also be programmed. The configuration of DDRPHYCR 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 memory controller registers.

Figure 17. Connecting DDR2 Memory Controller for 32-Bit Connection

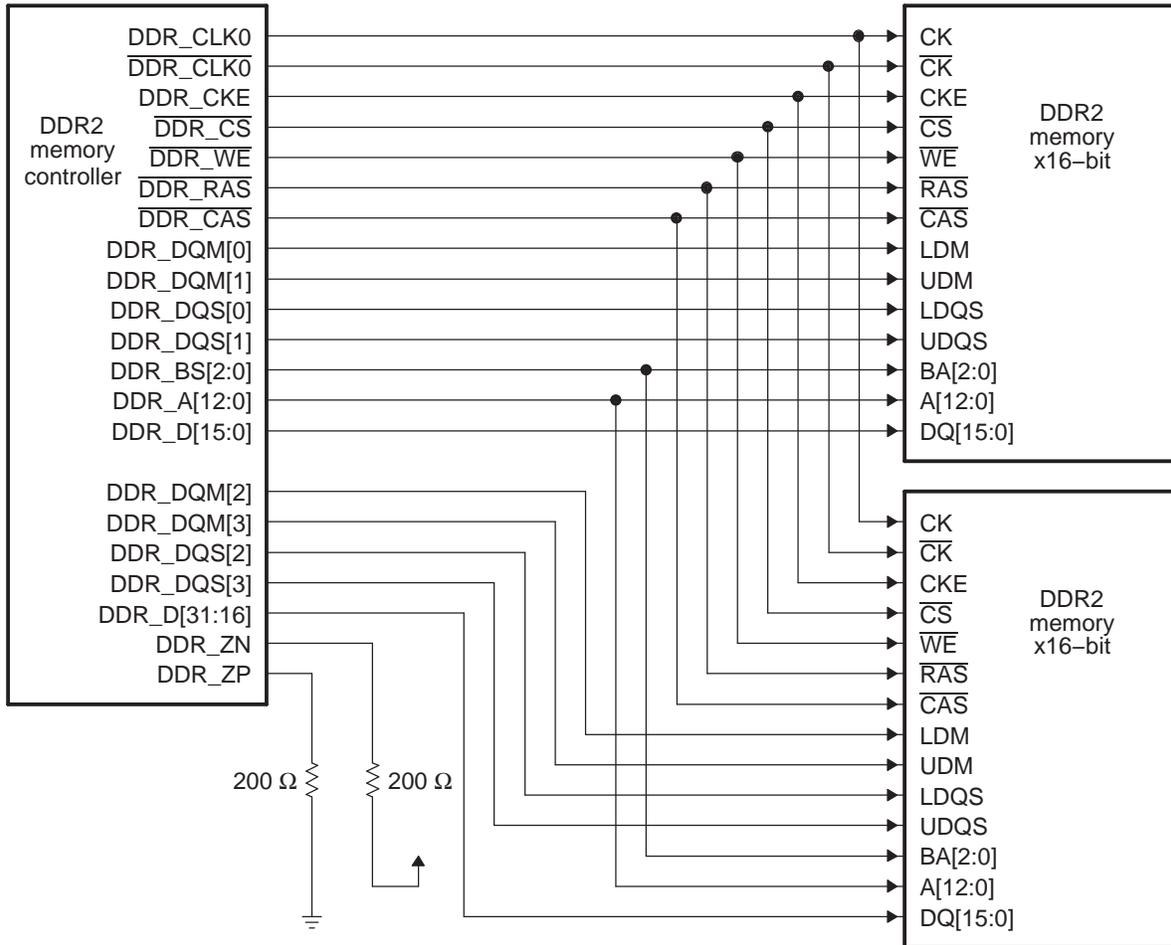
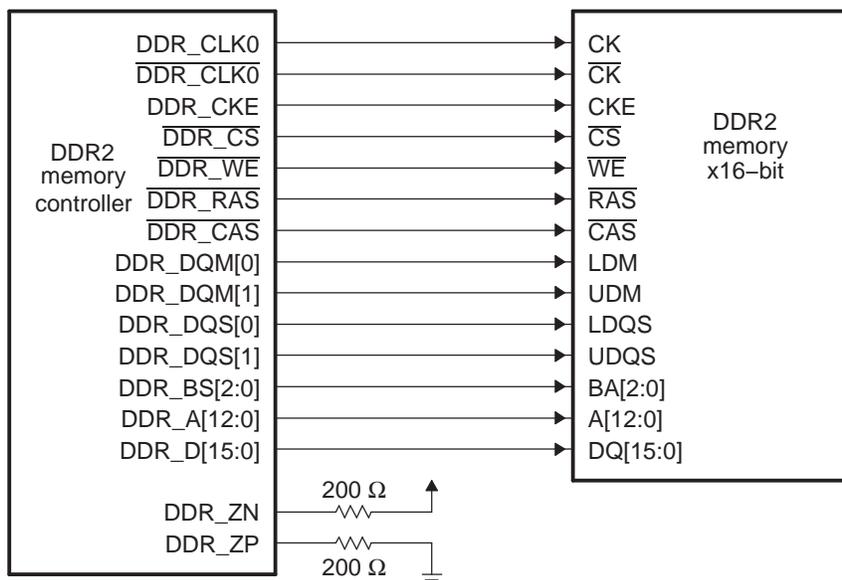


Figure 18. Connecting DDR2 Memory Controller for 16-Bit Connection



### 3.2.1 Configuring SDRAM Bank Configuration Register (SDBCR)

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

- Data bus width = 32 bits
- CAS latency = 4
- Number of banks = 8
- Page size = 1024 words

Table 14 shows the resulting SDBCR configuration. Note that the value of the TIMUNLOCK field is dependent on whether or not it is desirable to unlock SDTIMR and SDTIMR2. The TIMUNLOCK bit should only be set to 1 when the SDTIMR and SDTIMR2 need to be updated.

**Table 14. SDBCR Configuration**

Field	Value	Function Selection
TIMUNLOCK	x	Set to 1 to unlock the SDRAM timing register (SDTIMR) and the SDRAM timing register 2 (SDTIMR2). Cleared to 0 to lock SDTIMR and SDTIMR2.
NM	0h	To configure the DDR2 memory controller for a 32-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 memory controller to meet the refresh requirements of the attached DDR2 device. SDRCR also allows the DDR2 memory controller to enter and exit self refresh and enable and disable the MCLK stopping. In this example, we assume that the DDR2 memory controller is not in self-refresh mode and that MCLK stopping is disabled.

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

$$RR = \text{DDR2 clock frequency} \times \text{DDR2 refresh rate}$$

Table 15 displays the DDR2-400 refresh rate specification.

**Table 15. DDR2 Memory Refresh Specification**

Symbol	Description	Value
$t_{REF}$	Average Periodic Refresh Interval	7.8 $\mu$ s

Therefore, the following results assuming 133-MHZ DDR2 clock frequency.

$$RR = 133 \text{ MHz} \times 7.8 \mu\text{s} = 1037.4$$

Therefore,  $RR = 1038 = 40Eh$ .

Table 16 shows the resulting SDRCR configuration.

**Table 16. SDRCR Configuration**

Field	Value	Function Selection
SR	0	DDR2 memory controller is not in self-refresh mode.
MCLKSTOPEN	0	MCLK stopping is disabled.
RR	40Eh	Set to 40Eh DDR2 clock cycles to meet the DDR2 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 memory controller to meet the data sheet timing parameters of the attached DDR2 device. Each field in SDTIMR and SDTIMR2 corresponds to a timing parameter in the DDR2 data sheet specification. Table 17 and Table 18 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 clock cycles minus 1. See Section 4.4 and Section 4.5 for more information.

**Table 17. SDTIMR Configuration**

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

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

**Table 18. SDTIMR2 Configuration**

Register Field Name	DDR2 Data Manual Parameter Name	Description	Data Manual Value	Formula (Register field must be $\geq$ )	Register Value
T_XSNR	$t_{XSNR}$	Exit self refresh to a non-read command	137.5 nS	$(t_{XSNR} \times f_{DDR2\_CLK}) - 1$	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	$(t_{RTP} \times f_{DDR2\_CLK}) - 1$	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 (DDRPHYCR)

The DDR PHY control register (DDRPHYCR) contains a read latency (READLAT) field that helps the DDR2 memory controller determine when to sample read data. The READLAT field should be programmed to a value equal to CAS latency plus round trip board delay minus 1. The minimum READLAT value is CAS latency plus 1 and the maximum READLAT value is CAS latency plus 3 (again, the READLAT 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\_CLK0 and  $\overline{\text{DDR\_CLK0}}$ ) 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 READLAT value. In this example we will assume the round trip board delay is 1 DDR\_CLK0 cycle, therefore READLAT can be calculated as follows:

$$\text{READLAT} = \text{CAS latency} + \text{round trip board delay} - 1 = 4 + 1 - 1 = 4$$

**Table 19. DDRPHYCR Configuration**

Register Field Name	Description	Register Value
DLLRESET	Programmed to remove the DDR2 memory controller DLL from reset.	0
DLLPWRDN	Programmed to power up the DDR2 memory controller DLL.	0
READLAT	Read latency is equal to CAS latency plus round trip board delay for data minus 1.	4

## 4 DDR2 Memory Controller Registers

Table 20 lists the memory-mapped registers for the DDR2 memory controller. See the device-specific data manual for the memory address of these registers.

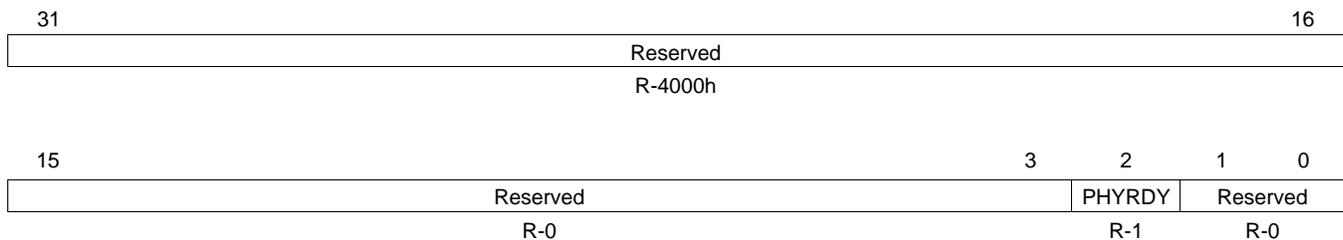
**Table 20. DDR2 Memory Controller Registers**

Acronym	Register Description	Section
SDRSTAT	SDRAM Status Register	<a href="#">Section 4.1</a>
SDBCR	SDRAM Bank Configuration Register	<a href="#">Section 4.2</a>
SDRCR	SDRAM Refresh Control Register	<a href="#">Section 4.3</a>
SDTIMR	SDRAM Timing Register	<a href="#">Section 4.4</a>
SDTIMR2	SDRAM Timing Register 2	<a href="#">Section 4.5</a>
PBBPR	Peripheral Bus Burst Priority Register	<a href="#">Section 4.6</a>
IRR	Interrupt Raw Register	<a href="#">Section 4.7</a>
IMR	Interrupt Masked Register	<a href="#">Section 4.8</a>
IMSR	Interrupt Mask Set Register	<a href="#">Section 4.9</a>
IMCR	Interrupt Mask Clear Register	<a href="#">Section 4.10</a>
DDRPHYCR	DDR PHY Control Register	<a href="#">Section 4.11</a>
VTPIOCR	VTP IO Control Register	<a href="#">Section 4.12</a>
DDRVTPR	DDR VTP Register	<a href="#">Section 4.13</a>
DDRVTPER	DDR VTP Enable Register	<a href="#">Section 4.14</a>

### 4.1 SDRAM Status Register (SDRSTAT)

The SDRAM status register (SDRSTAT) is shown in Figure 19 and described in Table 21.

**Figure 19. SDRAM Status Register (SDRSTAT)**



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

**Table 21. SDRAM Status Register (SDRSTAT) Field Descriptions**

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	PHYRDY	0	DDR2 memory controller DLL ready. Specifies whether the DDR2 memory controller DLL is powered up and locked.
		1	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	Reserved

## 4.2 SDRAM Bank Configuration Register (SDBCR)

The SDRAM bank configuration register (SDBCR) contains fields that program the DDR2 memory controller to meet the specification of the attached DDR2 memory. These fields configure the DDR2 memory controller to match the data bus width, CAS latency, number of internal banks, and page size of the attached DDR2 memory. The SDBCR is shown in Figure 20 and described in Table 22. Writing to the DDRDRIVE, CL, IBANK, and PAGESIZE bit fields will cause the DDR2 memory controller to start the DDR2 SDRAM initialization sequence.

**Figure 20. SDRAM Bank Configuration Register (SDBCR)**

31	Reserved				24	23	22	Reserved		19	18	17	16
				BOOTUNLOCK					DDRDRIVE				
R-0				R/W-0	R-2h				R/W-1	R-3h			
15	14	13	12	Reserved			11	CL		9	Reserved		
TIMUNLOCK	NM						CL						
R/W-0	R/W-0	R-0					R/W-5h			R-0			
7	6	Reserved			4	3	2	PAGESIZE		Reserved			
Reserved		IBANK			Reserved		PAGESIZE						
R-0		R/W-2h			R-0		R/W-0						

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

**Table 22. SDRAM Bank Configuration Register (SDBCR) Field Descriptions**

Bit	Field	Value	Description
31-24	Reserved	0	Reserved.
23	BOOTUNLOCK	0 1	Boot unlock. Controls the write permission settings for the DDRDRIVE bit. To change the DDRDRIVE bit value, use the following sequence: 1. Write a 1 to the BOOTUNLOCK bit. 2. Write a 0 to the BOOTUNLOCK bit along with the desired value of the DDRDRIVE bit. 0 DDRDRIVE bit may not be changed 1 DDRDRIVE bit may be changed
22-19	Reserved	2h	Reserved. Always write a value of 2h to these bits.
18	DDRDRIVE	0 1	DDR2 SDRAM drive strength. Configures the output driver impedance control value of the DDR2 SDRAM memory. To change the DDRDRIVE bit value, use the following sequence: 1. Write a 1 to the BOOTUNLOCK bit. 2. Write a 0 to the BOOTUNLOCK bit along with the desired value of the DDRDRIVE bit. 0 Normal drive strength. 1 Weak drive strength.
17-16	Reserved	3h	Reserved. Always write a value of 3h to these bits.
15	TIMUNLOCK	0 1	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	0 1	DDR2 data bus width. 0 32-bit bus width. 1 16-bit bus width
13-12	Reserved	0	Reserved

**Table 22. SDRAM Bank Configuration Register (SDBCR) Field Descriptions (continued)**

Bit	Field	Value	Description
11-9	CL	0-7h	CAS latency.
		0-1h	Reserved
		2h	CAS latency of 2
		3h	CAS latency of 3
		4h	CAS latency of 4
		5h	CAS latency of 5
		6h-7h	Reserved
8-7	Reserved	0	Reserved
6-4	IBANK	0-7h	Internal DDR2 bank setup. Defines the number of internal banks on the external DDR2 memory.
		0	1 bank
		1h	2 banks
		2h	4 banks
		3h	8 banks
		4h-7h	Reserved
3	Reserved	0	Reserved. Always write a 0 to this bit.
2-0	PAGESIZE	0-7h	DDR2 page size. Defines the page size of each page of the external DDR2 memory.
		0	256-word page requiring 8 column address bits.
		1h	512-word page requiring 9 column address bits.
		2h	1024-word page requiring 10 column address bits.
		3h	2048-word page requiring 11 column address bits.
		4h-7h	Reserved

### 4.3 SDRAM Refresh Control Register (SDRCR)

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

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

The SDRCR is shown in [Figure 21](#) and described in [Table 23](#).

**Figure 21. SDRAM Refresh Control Register (SDRCR)**

31	30	29	16
SR	MCLKSTOPEN	Reserved	
R/W-0	R/W-0	R-0	
			0
RR			
R/W-884h			

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

**Table 23. SDRAM Refresh Control Register (SDRCR) Field Descriptions**

Bit	Field	Value	Description
31	SR	0	DDR2 memory controller exits the self-refresh mode.
		1	DDR2 memory controller enters the self-refresh mode.
30	MCLKSTOPEN	0	Disables MCLK stopping, MCLK may not be stopped.
		1	Enables MCLK stopping, MCLK may be stopped. The SR bit must be set to 1 before setting the MCLKSTOPEN bit to 1.
29-16	Reserved	0	Reserved
15-0	RR	0-FFFFh	Refresh rate. Defines the rate at which the attached DDR2 devices will be refreshed. The value of this field may be calculated with the following equation: $RR = \text{DDR2 clock frequency (in MHz)} \times \text{DDR2 refresh rate (in } \mu\text{s)}$ where <i>DDR2 refresh rate</i> is derived from the DDR2 data sheet.

#### 4.4 SDRAM Timing Register (SDTIMR)

The SDRAM timing register (SDTIMR) configures the DDR2 memory controller to meet many of the AC timing specification of the DDR2 memory. The SDTIMR register is programmable only when the TIMUNLOCK bit is set to 1 in the SDBCR. Note that DDR\_CLK0 is equal to the period of the DDR\_CLK0 signal. See the DDR2 memory data sheet for information on the appropriate values to program each field. The SDTIMR is shown in Figure 22 and described in Table 24.

**Figure 22. SDRAM Timing Register (SDTIMR)**

31	25	24	22	21	19	18	16	
T_RFC			T_RP		T_RCD		T_WR	
R/W-1Ah			R/W-5h		R/W-5h		R/W-3h	
15	11	10	6	5	3	2	1 0	
T_RAS		T_RC			T_RRD		Rsvd	T_WTR
R/W-9h		R/W-Eh			R/W-3h		R-0	R/W-3h

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

**Table 24. SDRAM Timing Register (SDTIMR) Field Descriptions**

Bit	Field	Value	Description
31-25	T_RFC	0-7Fh	Specifies the minimum number of DDR_CLK0 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 data sheet. Calculate by: $T\_RFC = (t_{rfc}/DDR\_CLK0 \text{ period}) - 1$
24-22	T_RP	0-7h	Specifies the minimum number of DDR_CLK0 cycles from a precharge command to a refresh or activate command, minus 1. Corresponds to the $t_{rp}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_RP = (t_{rp}/DDR\_CLK0 \text{ period}) - 1$
21-19	T_RCD	0-7h	Specifies the minimum number of DDR_CLK0 cycles from an activate command to a read or write command, minus 1. Corresponds to the $t_{rcd}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_RCD = (t_{rcd}/DDR\_CLK0 \text{ period}) - 1$
18-16	T_WR	0-7h	Specifies the minimum number of DDR_CLK0 cycles from the last write transfer to a precharge command, minus 1. Corresponds to the $t_{wtr}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_WR = (t_{wtr}/DDR\_CLK0 \text{ period}) - 1$ 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_CLK0 cycles from an activate command to a precharge command, minus 1. Corresponds to the $t_{ras}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_RAS = (t_{ras}/DDR\_CLK0 \text{ period}) - 1$ $T\_RAS$ must be greater than or equal to $T\_RCD$ .
10-6	T_RC	0-1Fh	Specifies the minimum number of DDR_CLK0 cycles from an activate command to an activate command, minus 1. Corresponds to the $t_{rc}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_RC = (t_{rc}/DDR\_CLK0 \text{ period}) - 1$
5-3	T_RRD	0-7h	Specifies the minimum number of DDR_CLK0 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 data sheet. Calculate by: $T\_RRD = (t_{rrd}/DDR\_CLK0 \text{ period}) - 1$ Note: for an 8 bank DDR2 device this field must be equal to $((4 \times t_{RRD}) + (2 \times t_{CK})) / (4 \times t_{CK}) - 1$ .
2	Reserved	0	Reserved
1-0	T_WTR	0-3h	Specifies the minimum number of DDR_CLK0 cycles from the last write to a read command, minus 1. Corresponds to the $t_{wtr}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_WTR = (t_{wtr}/DDR\_CLK0 \text{ period}) - 1$

#### 4.5 SDRAM Timing Register 2 (SDTIMR2)

Like the SDRAM timing register (SDTIMR), the SDRAM timing register 2 (SDTIMR2) also configures the DDR2 memory controller to meet the AC timing specification of the DDR2 memory. The SDTIMR2 register is programmable only when the TIMUNLOCK bit is set to 1 in the SDBCR. See the DDR2 data sheet for information on the appropriate values to program each field. SDTIMR2 is shown in [Figure 23](#) and described in [Table 25](#).

**Figure 23. SDRAM Timing Register 2 (SDTIMR2)**

31	25	24	23	22	16
Reserved		Reserved		T_XSNR	
R-0		R/W-x		R/W-10h	
15	8	7	5	4	0
T_XSRD		T_RTP		T_CKE	
R/W-F1h		R/W-2h		R/W-5h	

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

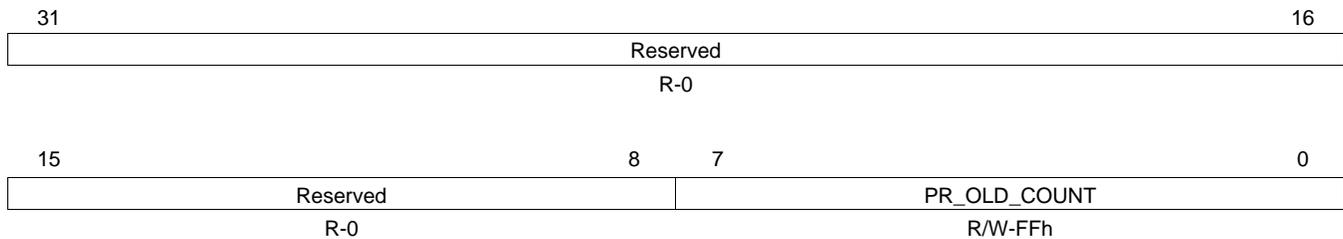
**Table 25. SDRAM Timing Register 2 (SDTIMR2) Field Descriptions**

Bit	Field	Value	Description
31-25	Reserved	0	Reserved
24-23	Reserved	x	Reserved. Reset value is indeterminate.
22-16	T_XSNR	0-7Fh	Specifies the minimum number of DDR_CLK0 cycles from a self-refresh exit to any other command except a read command, minus 1. Corresponds to the $t_{xsnr}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_XSNR = (t_{xsnr}/DDR\_CLK0 \text{ period}) - 1$
15-8	T_XSRD	0-FFh	Specifies the minimum number of DDR_CLK0 cycles from a self-refresh exit to a read command, minus 1. Corresponds to the $t_{xsrđ}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_XSRD = t_{xsrđ} - 1$
7-5	T_RTP	0-7h	Specifies the minimum number of DDR_CLK0 cycles from a last read command to a precharge command, minus 1. Corresponds to the $t_{rtp}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_RTP = (t_{rtp}/DDR\_CLK0 \text{ period}) - 1$
4-0	T_CKE	0-1Fh	Specifies the minimum number of DDR_CLK0 cycles between transitions on the DDR_CKE pin, minus 1. Corresponds to the $t_{cke}$ AC timing parameter in the DDR2 data sheet. Calculate by: $T\_CKE = t_{cke} - 1$

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

The peripheral bus burst priority register (PBBPR) helps prevent command starvation within the DDR2 memory controller. To avoid command starvation, the DDR2 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 memory controller raises the priority of the oldest command. The PBBPR is shown in [Figure 24](#) and described in [Table 26](#). See [Section 2.7.2](#) for more details on command starvation.

**Figure 24. Peripheral Bus Burst Priority Register (PBBPR)**



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

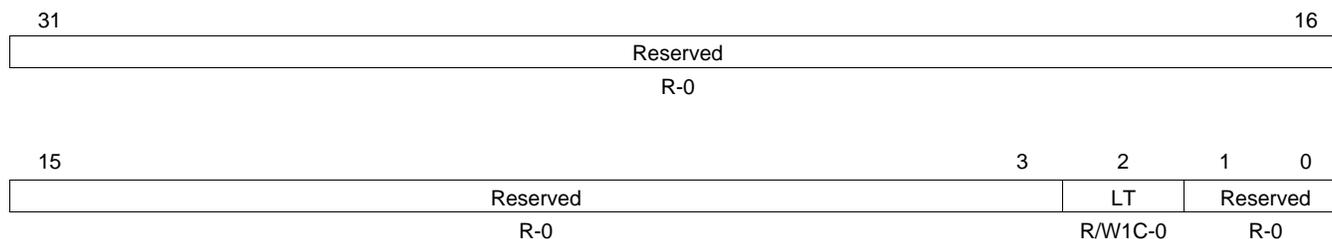
**Table 26. Peripheral Bus Burst Priority Register (PBBPR) Field Descriptions**

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	PR_OLD_COUNT	0-FFh	Priority raise old counter. Specifies the number of memory transfers after which the DDR2 memory controller will elevate the priority of the oldest command in the command FIFO. Setting this field to FFh disables this feature, thereby allowing old commands to stay in the FIFO indefinitely.
		0	1 memory transfer
		1	2 memory transfers
		2	3 memory transfers
		3-FEh	4 to 255 memory transfers
		FFh	Feature disabled, commands may stay in command FIFO indefinitely

## 4.7 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 25](#) and described in [Table 27](#).

**Figure 25. Interrupt Raw Register (IRR)**



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

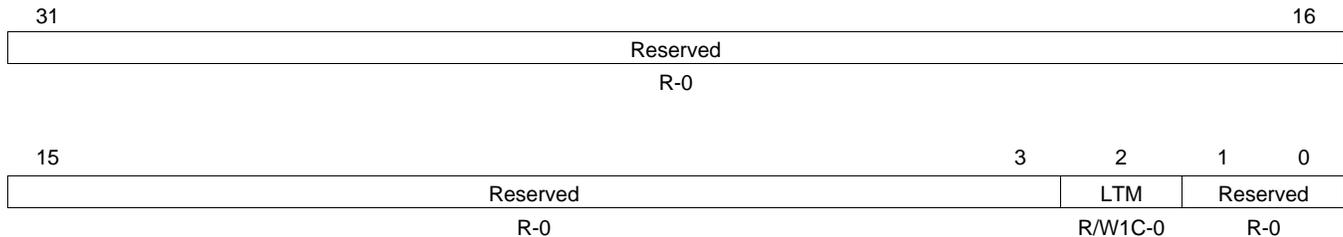
**Table 27. Interrupt Raw Register (IRR) Field Descriptions**

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	LT	0	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.
		1	A line trap condition has not occurred.
			Illegal memory access type. See <a href="#">Section 2.13</a> for more details.
1-0	Reserved	0	Reserved

#### 4.8 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 26](#) and described in [Table 28](#).

**Figure 26. Interrupt Masked Register (IMR)**



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

**Table 28. Interrupt Masked Register (IMR) Field Descriptions**

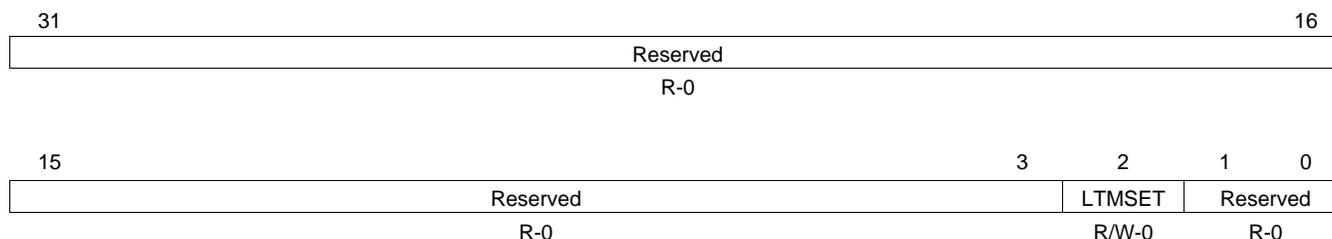
Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	LTM	0	Line trap masked. Write a 1 to clear LTM and the LT bit in the interrupt raw register (IRR); a write of 0 has no effect. A line trap condition has not occurred.
		1	Illegal memory access type (only set if the LTMSET bit in IMSR is set). See <a href="#">Section 2.13</a> for more details.
1-0	Reserved	0	Reserved

## 4.9 Interrupt Mask Set Register (IMSR)

The interrupt mask set register (IMSR) enables the DDR2 memory controller interrupt. The IMSR is shown in [Figure 27](#) and described in [Table 29](#).

**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 27. Interrupt Mask Set Register (IMSR)**



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

**Table 29. Interrupt Mask Set Register (IMSR) Field Descriptions**

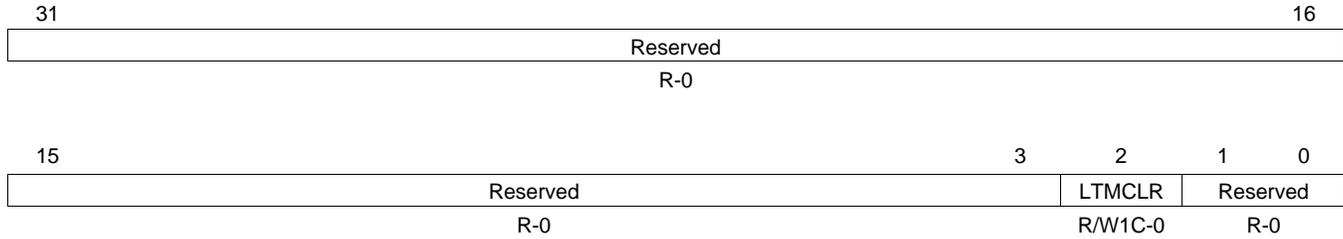
Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	LTMSET	0	Line trap interrupt set. Write a 1 to set LTMSET and the LTMCLR bit in the interrupt mask clear register (IMCR); a write of 0 has no effect.
		1	Line trap interrupt is not enabled; a write of 1 to the LTMCLR bit in IMCR occurred.
		1	Line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

#### 4.10 Interrupt Mask Clear Register (IMCR)

The interrupt mask clear register (IMCR) disables the DDR2 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 28](#) and described in [Table 30](#).

**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 28. Interrupt Mask Clear Register (IMCR)**



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

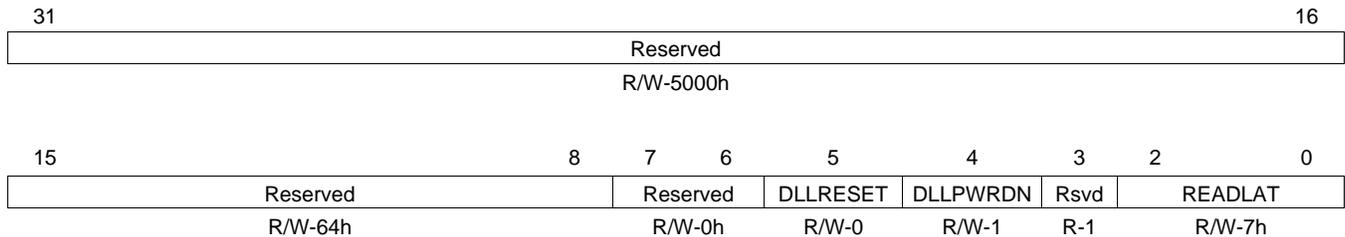
**Table 30. Interrupt Mask Clear Register (IMCR) Field Descriptions**

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	LTMCLR	0	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.
		1	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	Reserved

#### 4.11 DDR PHY Control Register (DDRPHYCR)

The DDR PHY control register (DDRPHYCR) configures the DDR2 memory controller DLL for operation and determines whether the DLL is in reset, whether it is powered up, and the read latency. The DDRPHYCR is shown in Figure 29 and described in Table 31.

**Figure 29. DDR PHY Control Register (DDRPHYCR)**



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

**Table 31. DDR PHY Control Register (DDRPHYCR) Field Descriptions**

Bit	Field	Value	Description
31-16	Reserved	5000h	Reserved. Always write 5000h to these bits.
15-8	Reserved	64h	Reserved. Always write 64h to these bits.
7-6	Reserved	0	Reserved. Always write 0 to these bits.
5	DLLRESET	0 1	Reset DLL. 0 DLL is out of reset. 1 Places the DLL in reset.
4	DLLPWRDN	0 1	Power down DLL. 0 DLL is powered up. 1 DLL is powered down, if DLLPWRDN and the SR bit and MCLKSTOPEN bit in the SDRAM refresh control register (SDRCR) are set to 1.
3	Reserved	0	Reserved
2-0	READLAT	0-7h	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 3. The minimum read latency value that is supported is CAS latency plus 1. The read latency value is defined in number of MCLK/DDR_CLK0 cycles.

#### 4.12 VTP IO Control Register (VTPIOCR)

The VTP IO control register (VTPIOCR) is used to control the calibration of the DDR2 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 30](#) and described in [Table 32](#).

**Figure 30. VTP IO Control Register (VTPIOCR)**

31	Reserved										16
R-0											
15	14	13	12	11	10	9	PCH		4	0	
RECAL	Rsvd	EN	Reserved	Rsvd					NCH		
R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0		R/W-0		R/W-1Fh		

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

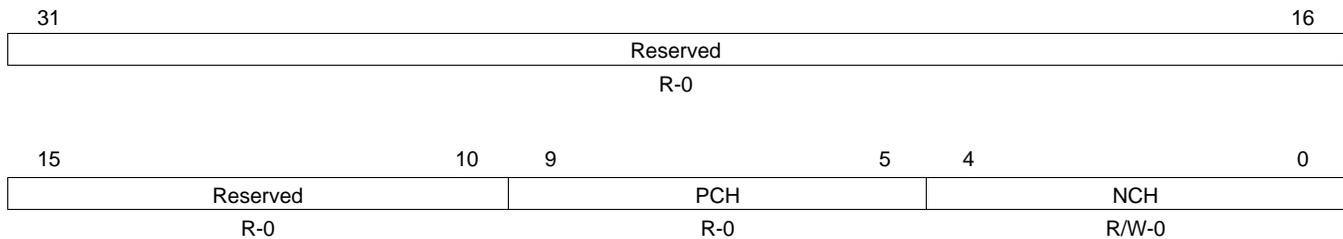
**Table 32. VTP IO Control Register (VTPIOCR) Field Descriptions**

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	RECAL	0 1	Start VTP IO calibration. Normal operation Transition from 0 to 1 starts VTP IO calibration.
14	Reserved	0	Reserved. Always write a 0 to this bit.
13	EN	0 1	VTP enable. VTP IO calibration is disabled. VTP IO calibration is enabled.
12-11	Reserved	0	Reserved. Always write a 0 to this bit.
10	Reserved	0	Reserved
9-5	PCH	0-1Fh	P channel value. This value is driven to the IO to calibrate the impedance of the IO. The value of PCH is determined by reading the DFT DDR VTP register (DDRVTTPR). See <a href="#">Section 4.13</a> for details.
4-0	NCH	0-1Fh	N channel value. This value is driven to the IO to calibrate the impedance of the IO. The value of NCH is determined by reading the DFT DDR VTP register (DDRVTTPR). See <a href="#">Section 4.13</a> for details.

### 4.13 DDR VTP Register (DDRVTPR)

The DDR VTP register (DDRVTPR) is used in conjunction with the VTP IO control register (VTPIOCR) to calibrate the output impedance of the DDR2 memory controller IOs with respect to voltage, temperature, and process. Following the calibration sequence, DDRVTPR contains the information needed to calibrate the impedance of the IO. Once the calibration sequence has completed, DDRVTPR should be read and the data written to the PCH and NCH fields in VTPIOCR. The DDRVTPR is shown in [Figure 31](#) and described in [Table 33](#).

**Figure 31. DDR VTP Register (DDRVTPR)**



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

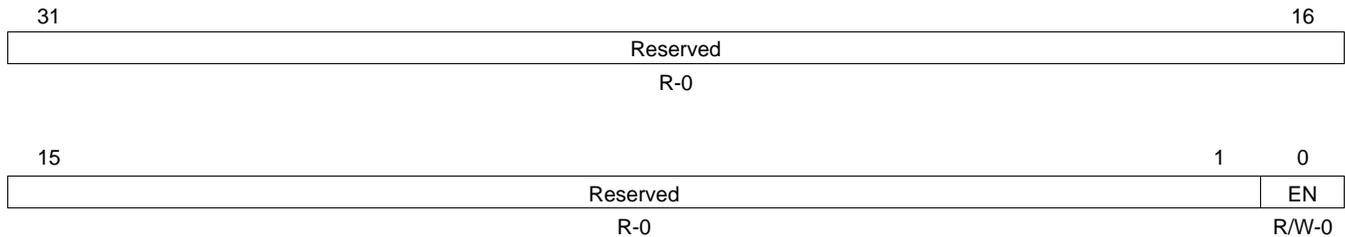
**Table 33. DDR VTP Register (DDRVTPR) Field Descriptions**

Bit	Field	Value	Description
31-10	Reserved	0	Reserved.
9-5	PCH	0-1Fh	P channel value for IO impedance calibration. Following the VTP calibration sequence, this value should be read and written to the PCH field in the VTP IO control register (VTPIOCR).
4-0	NCH	0-1Fh	N channel value for IO impedance calibration. Following the VTP calibration sequence, this value should be read and written to the NCH field in the VTP IO control register (VTPIOCR).

#### 4.14 DDR VTP Enable Register (DDRVTPER)

The DDR VTP enable register (DDRVTPER) is used to enable/disable accesses to the DDR VTP register (DDRVTPR). Writing a value of 1 to DDRVTPER enables accesses to DDRVTPR and writing a value of 0 disables accesses to DDRVTPR. The DDRVTPER is shown in [Figure 32](#) and described in [Table 34](#).

**Figure 32. DDR VTP Enable Register (DDRVTPER)**



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

**Table 34. DDR VTP Enable Register (DDRVTPER) Field Descriptions**

Bit	Field	Value	Description
31-1	Reserved	0	Reserved. Always write 0 to these bits.
0	EN	0	DDRVTPR access enable. Access to DDRVTPR is disabled.
		1	Access to DDRVTPR is enabled.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
RF/IF and ZigBee® Solutions	<a href="http://www.ti.com/lprf">www.ti.com/lprf</a>

### Applications

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2008, Texas Instruments Incorporated