



Double Data Rate (DDR) SDRAM Controller (Pipelined Version)

User's Guide

Introduction

DDR (Double Data Rate) SDRAM was introduced as a replacement for SDRAM memory running at bus speeds over 75MHz. DDR SDRAM is similar in function to regular SDRAM but doubles the bandwidth of the memory by transferring data twice per cycle (on both edges of the clock signal), implementing burst mode data transfer.

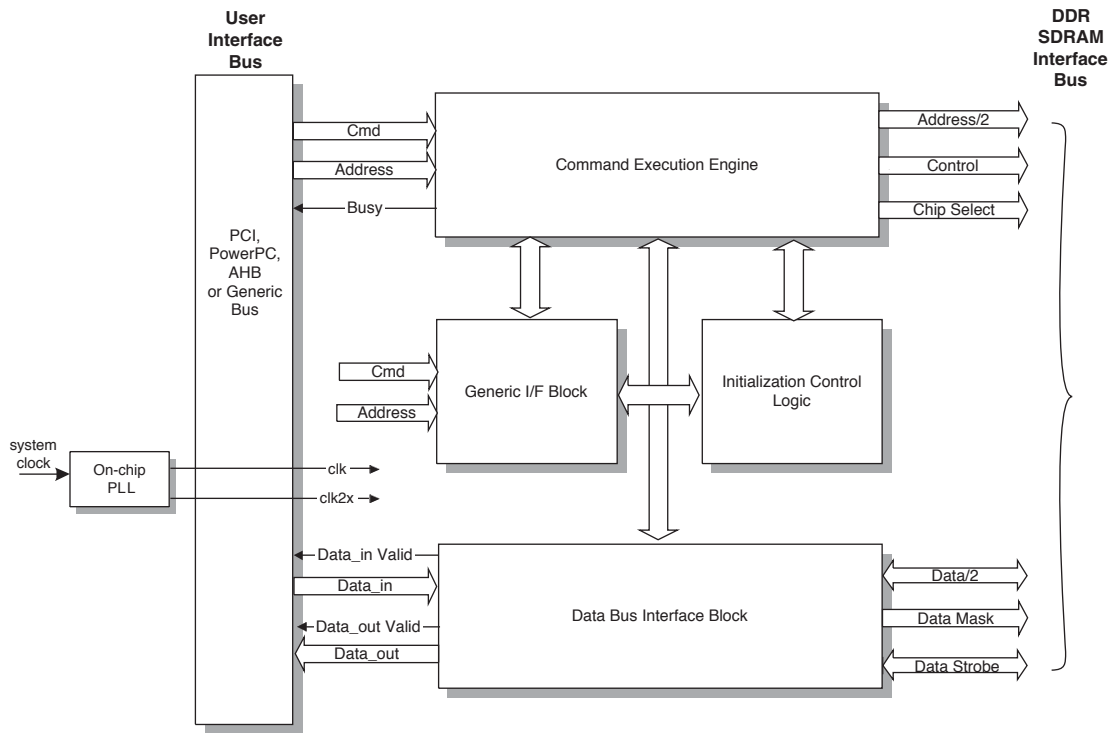
The DDR SDRAM Controller is a parameterized core. This allows the user to modify the data widths, burst transfer rates, and CAS latency settings of the design. In addition, the DDR core supports intelligent bank management. By maintaining a database of "all banks activated" and the "rows activated" in each bank, the DDR SDRAM Controller decides if an active or pre-charge command is needed. This effectively reduces the latency of read/write commands issued to the DDR SDRAM.

Since the DDR SDRAM Controller takes care of activating/pre-charging the banks, the user can simply issue simple read/write commands without regard to the bank/charge status.

Features

- Performance of Greater than 100MHz (200 DDR)
- Interfaces to JEDEC Standard DDR SDRAMs
- Supports DDR SDRAM Data Widths of 16, 32 and 64 Bits
- Supports up to 8 External Memory Banks
- Programmable Burst Lengths of 2, 4, or 8
- Programmable CAS Latency of 1.5, 2.0, 2.5 or 3.0
- Byte-level Writing Supported
- Increased Throughput Using Command Pipelining and Bank Management
- Supports Power-down and Self Refresh Modes
- Automatic Initialization
- Automatic Refresh During Normal and Power-down Modes
- Timing and Settings Parameters Implemented as Programmable Registers
- Bus Interfaces to PCI Target, PowerPC and AMBA (AHB) Buses Available
- Complete Synchronous Implementation

Figure 1. DDR Controller Block Diagram



Functional Description

The DDR SDRAM Controller block diagram, illustrated in Figure 1, consists of four functional modules: the Generic Interface block, Command Execution Engine, Data Bus Interface block and the Initialization Control Logic.

Generic Interface Block

The Generic interface block contains the configuration registers: CFG0, CFG1, CFG2, and CFG3. These registers are updated when a `Load_CFG` command is received from the user. These registers contain the programmable DDR SDRAM timing parameters and can be changed by the user to suit the DDR SDRAM memory timings being used thus giving the flexibility to use any DDR SDRAM memory.

Command Execution Engine

The command execution engine is the main component of the DDR SDRAM controller. This block accepts commands from the “User Interface Bus” and keeps a record of bank open/close status. It accepts up to two commands at any time (pipelined). Once a command is received, it decides whether to open the bank, close the bank or directly execute the READ/WRITE commands and apply the appropriate DDR SDRAM commands to the DDR SDRAM Memory. Table 1 shows the different user interface commands supported.

To maintain throughput of data this block uses two state machines to process READ/WRITE commands received from the user interface. When the commands are continuously received, one state machine works in master mode and the other state machine works in slave mode. The state machine that receives the command first becomes the master and the other becomes the slave on receiving the second command. Once the master state machine completes the command execution, the slave state machine execution is enabled.

This block also maintains an auto refresh counter, which refreshes the DDR SDRAM memory at the predetermined programmed intervals even during power down.

Table 1. DDR SDRAM Controller Generic I/F Commands

| Command Name | Cmd[2:0] | Description |
|-------------------------------|----------|---|
| NOP | 000 | No operation. |
| READ | 001 | Initiate a burst read. |
| WRITE | 010 | Initiate a burst write. |
| LOAD CONFIG REG (Load_CFG) | 011 | Load controller configuration values. The controller uses this command to load the CFG0/CFG1/CFG2/CFG3 registers. |
| LOAD MODE REG (Load_MR) | 100 | Load the Mode and Extended Mode registers. |
| POWER_DOWN | 101 | Put the DDR SDRAM into power-down or wake up from POWER DOWN. |
| SELF_REFRESH | 110 | To enter into self refresh mode or get out of self refresh mode |

Data Bus Interface

The Data Bus Interface block controls the data flow between the User Interface bus and DDR SDRAM Memory interface bus. The data received from the Memory during a read operation is converted from a double data rate to single data rate; similarly the data to be written into the memory is converted from a single data rate to a double data rate.

During a write operation, depending on the data mask signals, the data is written or masked by the DDR SDRAM-memory.

Initialization Control Logic

When the User sets the initialization bit (Bit 7 in the configuration register CFG0) using the `Load_CFG` command, this block starts initialization as specified in the DDR SDRAM specification. The DDR controller initialization can only be performed after the system power is applied and the clock is running for at least 200 μ s. An initialization is required before any read/write command is issued to the DDR SDRAM memory.

User Interface Bus

In order to connect this controller to different bus standards Lattice provides the following “Bus Interface blocks”:

1. PCI Target Interface
2. Power PC Interface
3. AMBA-AHB Interface
4. Generic Bus Interface

The main function of these “Bus Interface blocks” is to trap all transactions on the respective bus addressed to the DDR SDRAM and translate them into Generic Interface commands.

Since all the buses have burst addressing which is greater than the burst supported by the DDR SDRAM memory, all the interfaces have an address generator block which generates the appropriate address depending on the requested burst.

In all the interfaces the data going in and out of the bus is stored in a sync FIFO block, which is used as a storage buffer for read/write commands. During a read from the DDR SDRAM the data is sent to FIFO and is read by Bus Interface block. During a write the data is first written into the FIFO before actually writing this data into the DDR SDRAM.

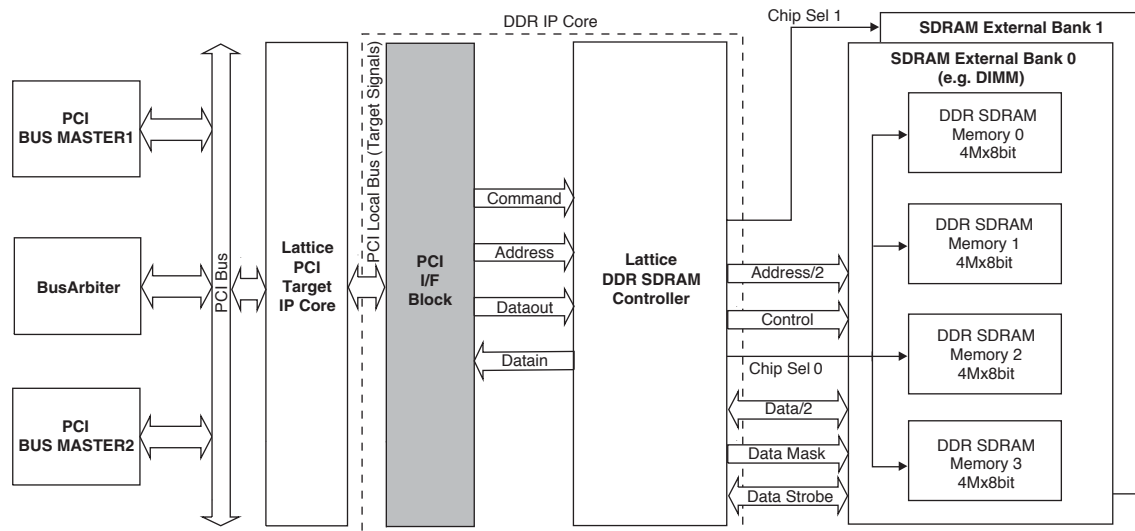
PCI Target Interface Block

The PCI Target Interface Block is used to interface the Lattice DDR Controller IP core with a Lattice PCI Target core, which simplifies the usage in a PCI Bus environment. Figure 2 shows the system with a PCI Target core.

The following are the features of the PCI Target interface block:

- Parameterized data path width of 32- or 64-bit on the PCI Local Bus and the User interface bus of DDR controller.
- Read/Write of configurable registers through PCI memory space.
- Read/Write to DDR through PCI memory space.
- Supports Power down and self refresh commands for low power applications.
- Programmable burst length.
- Programmable FIFO Depth
- Automatic wake up from power down/self refresh by a Read/Write command.

Figure 2. Typical PCI System with Lattice DDR SDRAM Controller



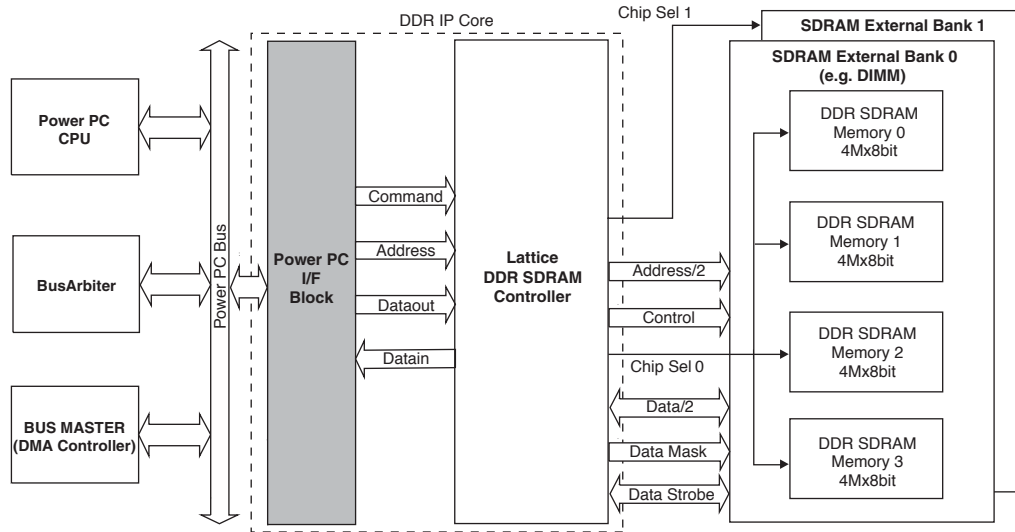
PowerPC BUS Interface Block

The Power PC Bus Interface block is used to interface with the Lattice DDR SDRAM Controller IP Core with the Power PC 60x Bus. This interface allows easy usage of the Lattice DDR SDRAM Controller in a Power PC Bus environment. Figure 3 shows the system with a Power PC Bus interface block.

The following are the features of the Power PC Bus interface block

- Supports PowerPC 601,603, 604 and processors supporting 60x bus.
- Parameterized data path widths of 32 or 64 bits.
- Supports both burst and single-beat data transfers (DDR Burst Length is 4).
- Programmable FIFO Depth
- Supports pipeline of two requests (write).
- Supports Address retry
- Supports separate address and data bus tenure.

Figure 3. Typical PowerPC System with Lattice DDR Controller



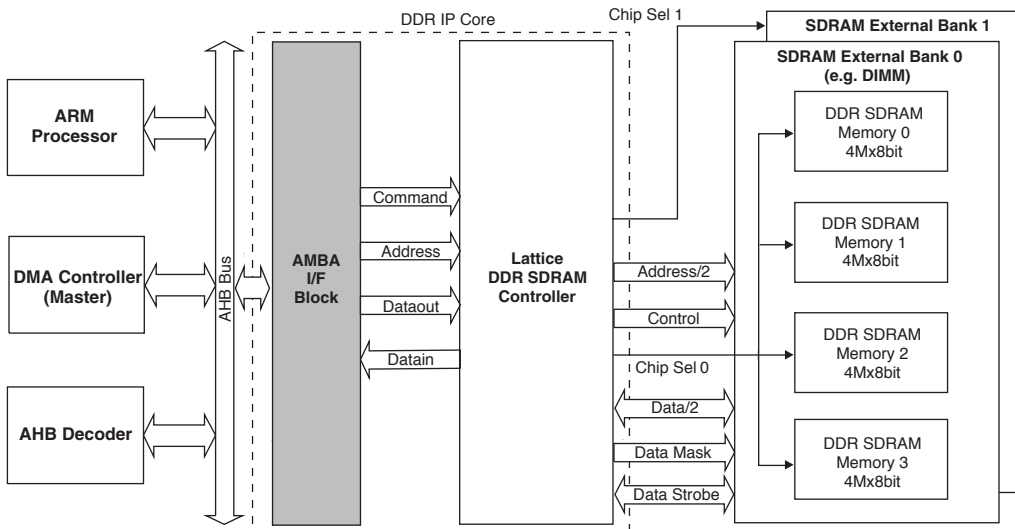
AHB Bus Interface Block

The AHB Bus Interface Block interfaces between AMBA Bus and Lattice DDR SDRAM Controller IP Core. This interface allows easy usage of the Lattice DDR SDRAM Controller on an ARM AHB bus environment. Figure 4 shows the system with an AHB Bus interface block.

The following are the features of the AHB Bus interface block

- Parameterized data path widths of 32, 64 and 128 bits on the AMBA Bus and the User Interface Bus.
- Supports INCR4/INCR8/INCR16/WRAP4/WRAP8/WRAP16/INCR Bursts (DDR side Burst Length is 4)
- Supports Byte/Half word/Word/Double Word transfers (with 64bit AHB-Bus)
- Programmable FIFO Depth

Figure 4. Typical AHB System with Lattice DDR Controller IP Cores



Parameter Descriptions

The static parameters are set before the design is synthesized to a gate-level netlist. The dynamic parameters can also be set in this way, or they can be changed after the core is programmed onto a device by writing the values into the Configuration Registers. The user should consult the DDR SDRAM specifications before choosing the dynamic parameters, which are dependent on the DDR SDRAM device being used.

Table 2. Static DDR Core Parameters

| Parameter | Description | Default Value |
|---------------------------------------|--|--------------------------|
| DSIZE | Defines the bus width for the data input/output port on the User Interface Bus side of the core. The selectable values are 32, 64, and 128 bits depending on the Interface Type. The data width on the DDR SDRAM Interface Bus (external memory side of the core) will be half of this value. For example, if a 64-bit wide bus is needed to access the DDR memory chip, the value for DSIZE needs to be set to 128. | 32 bits |
| RSIZE | Defines row address width. | 12 bits |
| CSIZE | Defines column width. | 11 bits |
| BSIZE | Defines bank width for internal chip banks. This version of the DDR SDRAM Controller is designed to work with memory chips containing four internal banks (sometimes called a quad memory array). The default value for BSIZE cannot be changed in this version of the core. | 2 bits |
| RANK_SIZE | Defines the number of external Banks (RANKS). NOTE: an external bank is a memory chip or group of chips (such as on a DIMM) that can be accessed using the same chip select signal. RANK_SIZE = 0 (External Banks = 1) RANK_SIZE = 1 (External Banks = 2) RANK_SIZE = 2 (External Banks = 4) RANK_SIZE = 3 (External Banks = 8) | 0 |
| Interface Type | The following Bus Interface Blocks are available for connecting this controller to different bus standards: PCI Target Interface, PowerPC Interface, AMBA_AHB Interface, Generic Bus Interface. | Generic |
| Interface-Dependent Parameters | | |
| ASIZE_BIM | If PCI, AHB or PowerPC Interface Type is selected, this parameter defines the BIM Address space. | 32 bits |
| Fifo Depth | If PCI, AHB or PowerPC Interface Type is selected, this parameter defines the FIFO Depth. | Varies by Interface Type |
| Fifo Add Width | If PCI, AHB or PowerPC Interface Type is selected, this parameter defines the FIFO Address Width. | Varies by Interface Type |
| Mem Base Address | If PowerPC Interface Type is selected, this parameter defines the Memory Base Address on the PowerPC Bus. | 0 |
| Mem Size | If PowerPC Interface Type is selected, this parameter defines the Memory Size on the PowerPC Bus. | 0 |
| Reg Base Address | If PowerPC Interface Type is selected, this parameter defines the Register Base Address on the PowerPC Bus. | 0 |
| Slave Select Bits | If AHB Interface Type is selected, this parameter defines the Slave Select Bits on the AHB Bus. | 2 bits |

Table 3. Dynamic DDR Core Parameters

| Parameter | Description | Default Value |
|----------------|---|-------------------------|
| INIT | Initialize DDR. Initializes the DDR SDRAM when bit is set. Set by command in register. | 0 |
| TRCD Delay | RAS to CAS Delay. This is the delay from /RAS to /CAS in number of clock cycles and is calculated using this formula $\text{INT}(t_{\text{RCD(MIN)}} / t_{\text{CK}})^*$. | 2 |
| TRRD Delay | Row ACTIVE to Row ACTIVE Delay. This delay is in clock cycles and is calculated using this formula $\text{INT}(t_{\text{RRD(MIN)}} / t_{\text{CK}})^*$. | 2 |
| TRFC Delay | AUTO REFRESH command period. This delay is in clock cycles and is calculated using this formula $\text{INT}(t_{\text{RFC(MIN)}} / t_{\text{CK}})^*$. | 9 |
| TRP Delay | PRECHARGE command period. This is calculated by the formula $\text{INT}(t_{\text{RP(MIN)}} / t_{\text{CK}})^*$. | 2 |
| TMRD Delay | LOAD MODE REGISTER command cycle time. This is calculated by the formula $\text{INT}(t_{\text{MRD(MIN)}} / t_{\text{CK}})^*$. | 2 |
| TWR Delay | Write recovery time. This is calculated by the formula $\text{INT}(t_{\text{WR(MIN)}} / t_{\text{CK}})^*$. | 2 |
| TRAS Delay | ACTIVE to PRECHARGE delay. Defines the delay between the ACTIVE and PRECHARGE commands (maximum value = 15 clock cycles). | 6 |
| TWTR Delay | WRITE to READ command delay. Defines internal write to read command delay (maximum value = 7 clock cycles). | 1 |
| TRC Delay | ACTIVE to ACTIVE /AUTOREFRESH command delay. Defines ACTIVE to ACTIVE /auto refresh command period delay (maximum value = 15 clock cycles). | 8 |
| CAS Latency | CAS Latency. Delay in clock cycles between the registration of a READ command and the first bit of output data. Valid values are 1.5, 2.0, 2.5 and 3.0. | 2 (2.0 clock cycles) |
| Burst Length | Burst Length. This number determines the maximum number of columns that can be accessed for a given READ/WRITE and is equal to Burst Length programmed in the Mode register. Valid values are 2, 4 and 8. | 2 |
| Burst Type | Burst Type. Specifies whether an interleaved or sequential burst is required. 0 represents a sequential and 1 represents an interleaved burst type. | 0 |
| DSTRENGTH | Drive Strength. Defines bit 1 in the Extended mode register. 0 represents normal drive strength, 1 represents a reduced drive strength (required by some memory devices). | 0 |
| QFCFUNC | Defines bit 2 of the extended mode register which enables or disables the QFC function (required by some memory devices). | 0 |
| Refresh Period | Refresh Period. Defines maximum time period between AUTOREFRESH commands. Calculate as follows: $\text{INT}(t_{\text{REF}} / t_{\text{CK}})^*$. | 2228 |

*Notes:

 t_{CK} = Clock cycle time $t_{\text{RCD(MIN)}}$ = ACTIVE to READ or WRITE delay $t_{\text{RRD(MIN)}}$ = ACTIVE (bank A) to ACTIVE (bank B) command period $t_{\text{RFC(MIN)}}$ = AUTOREFRESH command period (min.) $t_{\text{RP(MIN)}}$ = PRECHARGE command period $t_{\text{MRD(MIN)}}$ = LOAD_MR command cycle time $t_{\text{WR(MIN)}}$ = Write recovery time t_{REF} = AUTOREFRESH command interval (max.)

Signal Descriptions

The following tables show the different interface signals for the DDR SDRAM controller. The DDR SDRAM Interface signals are the same for all core configurations.

Table 4. DDR SDRAM Interface Bus Signals

| Signal Name | Direction | Active State | Description |
|--|-----------|--------------|---|
| ddr_clk | Output | N/A | DDR SDRAM Clock, derived from the system clock. NOTE: If multiple memory banks are used (RANK_SIZE > 0), then additional clock signals will be needed at the project's top level to drive each individual memory bank. |
| ddr_clk_n | Output | N/A | Inverted DDR SDRAM Clock, derived from the system clock. NOTE: If multiple memory banks are used (RANK_SIZE > 0), then additional clock signals will be needed at the project's top level to drive each individual memory bank. |
| ddr_cke | Output | High | Clock enable. |
| ddr_cs_n [(2 ^{RANK_SIZE}) - 1:0] | Output | N/A | Active Low Chip Select. Selects and deselects the DDR SDRAM external bank. |
| ddr_we_n | Output | Low | Write Enable. Defines the part of the command being entered. |
| ddr_cas_n | Output | Low | Column Select. Defines the part of the command being entered. |
| ddr_ras_n | Output | Low | Row select. Defines the part of the command being entered. |
| ddr_ad[RSIZE-1:0] | Output | N/A | Row or column address lines depending whether the /RAS or /CAS is active. |
| ddr_ba[BFSIZE-1:0] | Output | N/A | Bank Address Select. |
| ddr_dq[DSIZE/2-1:0] | In/Out | N/A | Bi-directional Data Bus. |
| ddr_dqm[DSIZE/16-1:0] | Output | N/A | Data mask signals used to mask the byte lanes for byte level write control. |
| ddr_dqs[DSIZE/16-1:0] | In/Out | N/A | Data strobe signals used by memory to latch the write data. |

Table 5. User Generic Interface Bus Interface Signals

| Signal Name | Direction | Active State | Description |
|---------------------|-----------|--------------|--|
| clk | Input | N/A | System clock. |
| reset_n | Input | Low | System reset. |
| cmd[2:0] | Input | N/A | Command for controller. |
| datain[DSIZE-1:0] | Input | N/A | Data input. DSIZE is a programmable parameter of 32, 64, or 128. |
| addr[ASIZE-1:0] | Input | N/A | Address for read/write. ASIZE is based on size of memory, which is derived by the following formula: ASIZE = RANK_SIZE + RSIZE + BFSIZE + CSIZE. |
| dmssel[DSIZE/8-1:0] | Input | N/A | Data Mask select. |
| busy | Output | High | Busy signal indicates the controller will not accept any more commands. |
| dataout[DSIZE-1:0] | Output | N/A | Data out. |
| dataout_valid | Output | High | During a read, this signal indicates when the dataout bus from the controller contains valid data. |
| datain_valid | Output | High | This signal indicates when the user can start sending in data through datain bus during a write. |
| clk2x | Input | N/A | This is the doubled clock signal coming from the on-chip PLL. |

Table 6. Power PC Bus Interface Signals

| Signal Name | Direction | Active State | Description |
|-----------------------|-----------|--------------|--|
| clk | In | N/A | System Clock. |
| reset_n | In | LOW | System Reset. |
| ppc_aack_n | Out | LOW | Address Acknowledge. When asserted, this signal indicates the address phase of the transaction is complete. |
| ppc_addr[0:ASIZE_BIM] | In | N/A | Address Bus. Address received from a bus master after receiving a bus grant. |
| ppc_data_l[0:31] | I/O | N/A | Low Data Bus. |
| ppc_data_h[0:31] | I/O | N/A | High Data Bus. |
| ppc_artry_n | In | LOW | Address Retry. This signal indicates the current cycle is aborted and the bus master will issue a request at a later time. |
| ppc_dbb_n | In | LOW | Data Bus Busy. The bus master that has received a data bus grant issues this signal. This signal indicates the length the data bus will be used for a memory access. |
| ppc_ta_n | Out | LOW | Transfer Acknowledge. This signal indicates the data transfer on the PowerPC bus has been completed. |
| ppc_tbst_n | In | LOW | Transfer Burst. This signal indicates a burst transfer is in progress. |
| ppc_tea_n | Out | LOW | Transfer Error Acknowledge. This signal indicates an error has occurred during a data transfer. |
| ppc_ts_n | In | LOW | Transfer Start. This signal indicates the master has begun a memory bus transaction, and the address and transfer attributes are valid. |
| ppc_tsiz[0:2] | In | LOW | Transfer Size. |
| ppc_tt[0:4] | In | LOW | Transfer Type. |
| clk2x | In | N/A | This the doubled clock signal coming from the on-chip PLL. |

Table 7. PCI Local Bus Interface Signals

| Signal Name | Direction | Active | Description |
|-------------------------------|-----------|--------|---|
| clk | In | N/A | PCI System Clock. |
| reset_n | In | LOW | Asynchronous PCI Reset. |
| l_data_in[DSIZE-1:0] | Out | N/A | Local Address/Burst Length/Data Input. The address/burst length input is used in master transactions. The data input is used for a master write or for a target read. |
| l_data_out[DSIZE-1:0] | In | N/A | Local data output for a master read or a target write. |
| lt_address_out[ASIZE_BIM-1:0] | In | N/A | Local starting address output for target reads and writes. |
| lt_ben_out[DSIZE/8-1:0] | In | N/A | Local target byte enables. |
| lt_command_out[3:0] | In | N/A | Local command for target reads and writes. |
| lt_abortn | Out | LOW | Local target abort request. |
| lt_disconnectn | Out | LOW | Local target disconnect or retry. |
| l_interruptn | Out | LOW | Local side interrupt request (multi-function device may need additional IRQ signals). |
| lt_rdyn | Out | LOW | Local target ready to receive data (write) or send data (read). |
| lt_r_nw | In | HIGH | Read/Write# (read/not write). Signals whether the current transaction is a read or write. |
| cache[7:0] | In | N/A | Local target controller cache register output. |

Table 7. PCI Local Bus Interface Signals (Continued)

| Signal Name | Direction | Active | Description |
|-----------------|-----------|--------|--|
| status[5:0] | In | N/A | Local target controller status register. |
| lt_hdata_xfern | In | LOW | Memory or I/O high DWORD read or write data phase complete. The address counter can be incremented in combination with the lt_ldataxfern. |
| lt_ldata_xfern | In | LOW | Memory or I/O low DWORD read or write data phase complete. The address counter can be incremented in combination with the lt_hdataxfern. |
| exprom_hit | In | HIGH | Expansion ROM register hit. |
| bar_hit[5:0] | In | N/A | Signals that the current address is within one of the base address register ranges, and access is requested until the current cycle is done (multi-function devices will need an additional set of registers for each function). |
| lt_64bit_transn | In | LOW | Signals the local target that a 64-bit read or write transaction is underway. |
| clk2x | In | N/A | This is the doubled clock signal coming from the on-chip PLL. |

Table 8. AMBA Bus Interface Signals

| Signal Name | Direction | Active State | Description |
|----------------------|-----------|--------------|---|
| clk | In | N/A | Bus Clock. This clock times all bus transfers. All signal timings are related to the rising edge of clk. |
| reset_n | In | LOW | The bus reset signal is active LOW and is used to reset the system and the bus. This is the only active LOW signal. |
| HADDR[ASIZE_BIM-1:0] | In | N/A | The 32-bit system address bus. |
| HTRANS[1:0] | In | N/A | Transfer type. This indicates the type of the current transfer, which can be NONSEQUENTIAL, SEQUENTIAL, IDLE or BUSY. |
| HWRITE | In | HIGH | Transfer direction. When HIGH, this signal indicates a write transfer. When LOW, it indicates a read transfer. |
| HSIZE[2:0] | In | N/A | Transfer size. Indicates the size of the transfer, which is typically byte (8-bit), half word (16-bit), word (32-bit) or double word (64-bit) for a 64-bit AHB Bus. |
| HBURST[2:0] | In | N/A | Burst type. Indicates if the transfer forms part of a burst. Four, eight and sixteen beat bursts are supported. The burst may be either incremental or wrapping. |
| HWDATA[DSIZE-1:0] | In | HIGH | Write data bus. The write data bus is used to transfer data from the master to the bus slaves during write operations. |
| HSELx | In | HIGH | Slave select. This signal indicates that the current transfer is intended for the slave. This signal is a combinatorial decode of the address bus. |
| HSELregx | In | HIGH | This signal indicates the current transfer is meant for internal registers of the Bus Interface Block. This is valid only when HSELx is asserted. |
| HSELmemx | In | HIGH | This signal indicates the current transfer is meant for DDR memory data. This is valid only when HSELx is asserted. |
| HREADY | In | HIGH | Transfer done. When HIGH, the HREADY signal indicates that a transfer has finished on the bus. |
| HRDATA[DSIZE-1:0] | Out | N/A | Read data bus. The read data bus is used to transfer data from bus slaves to the bus master during read operations. |
| HREADY_out | Out | HIGH | Transfer done. When high, the HREADY_out signal indicates that a transfer has finished on the bus. This signal may be driven LOW to extend a transfer. |
| HRESP[1:0] | Out | N/A | Transfer response. The transfer response provides additional information on the status of a transfer. Four different responses are possible: OKAY, ERROR, RETRY and SPLIT. RETRY and SPLIT are not supported. |
| clk2x | In | N/A | This is the doubled clock signal coming from the on-chip PLL. |

Timing Specifications

Figure 5. Generic I/F Timing Diagram

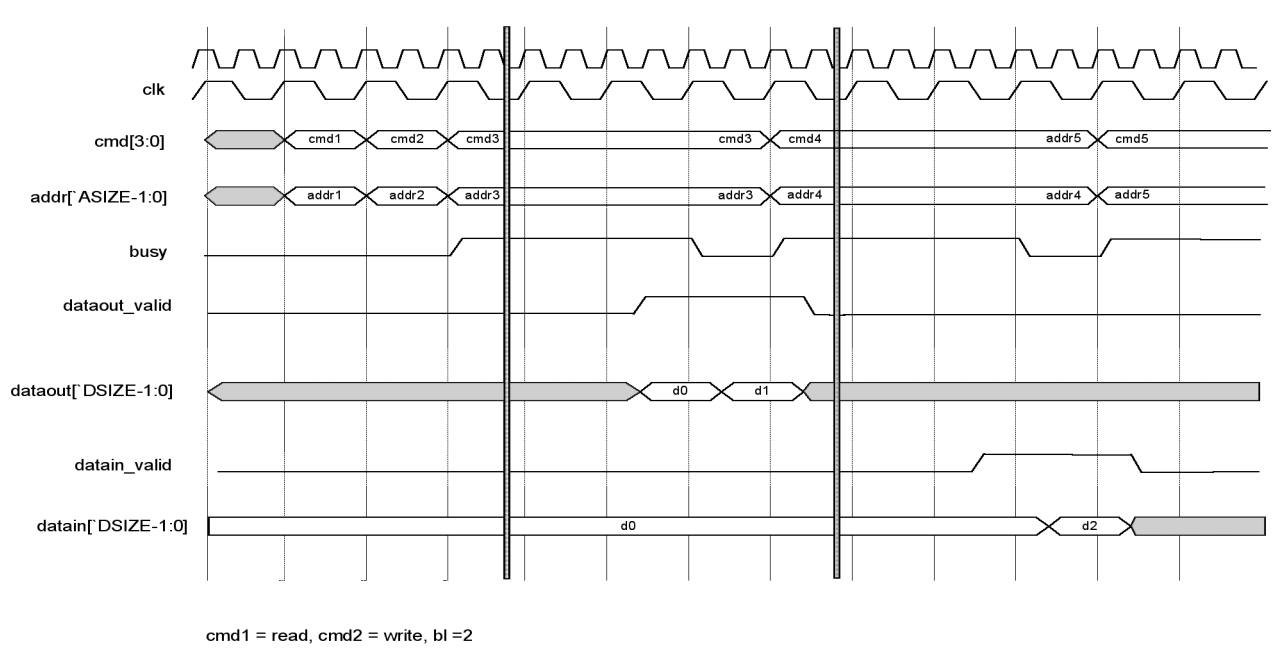


Figure 6. Read Followed By Read (Same Bank Same Row) With BL = 2

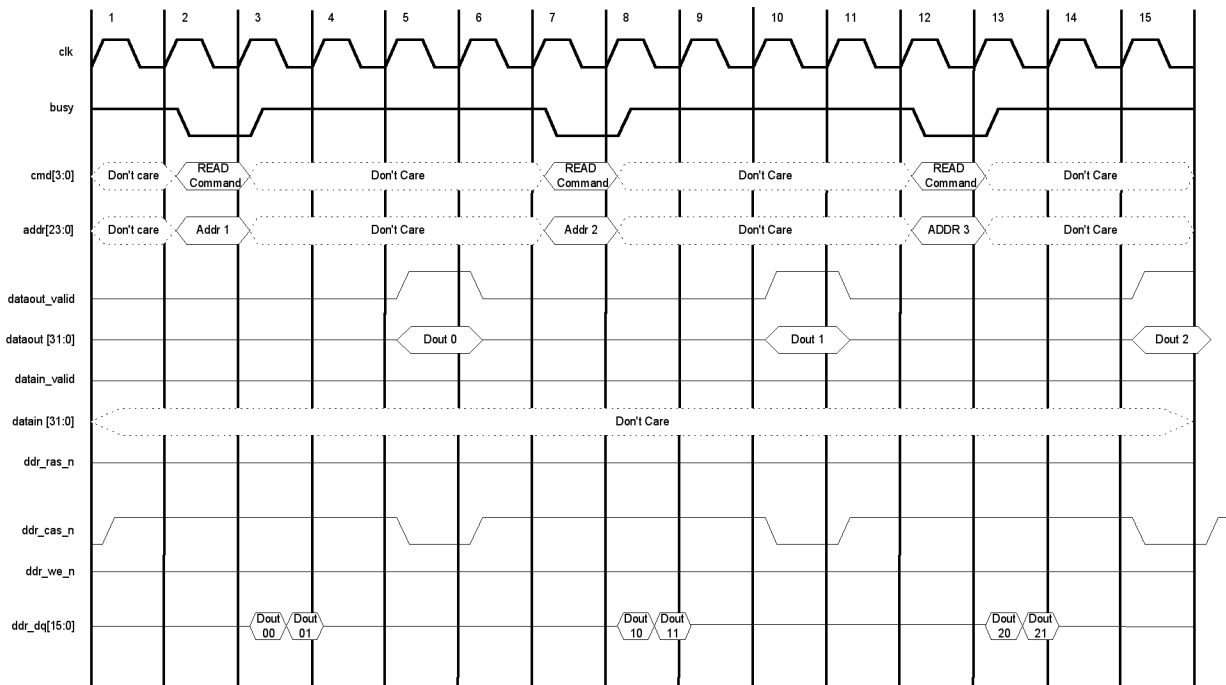


Figure 7. Read Followed by Read (Same Bank Different Row) With BL = 2

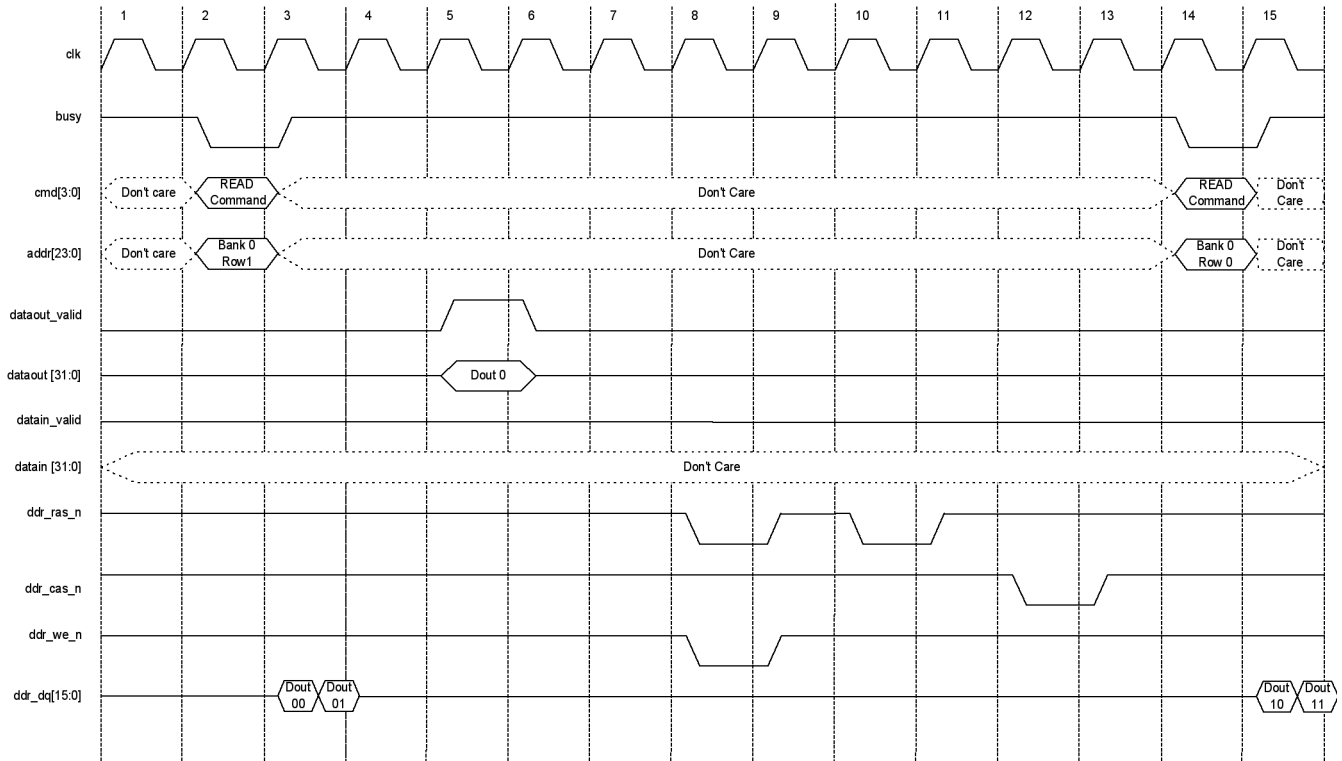


Figure 8. Read Followed by Read (Different Bank Row Open BL = 2)

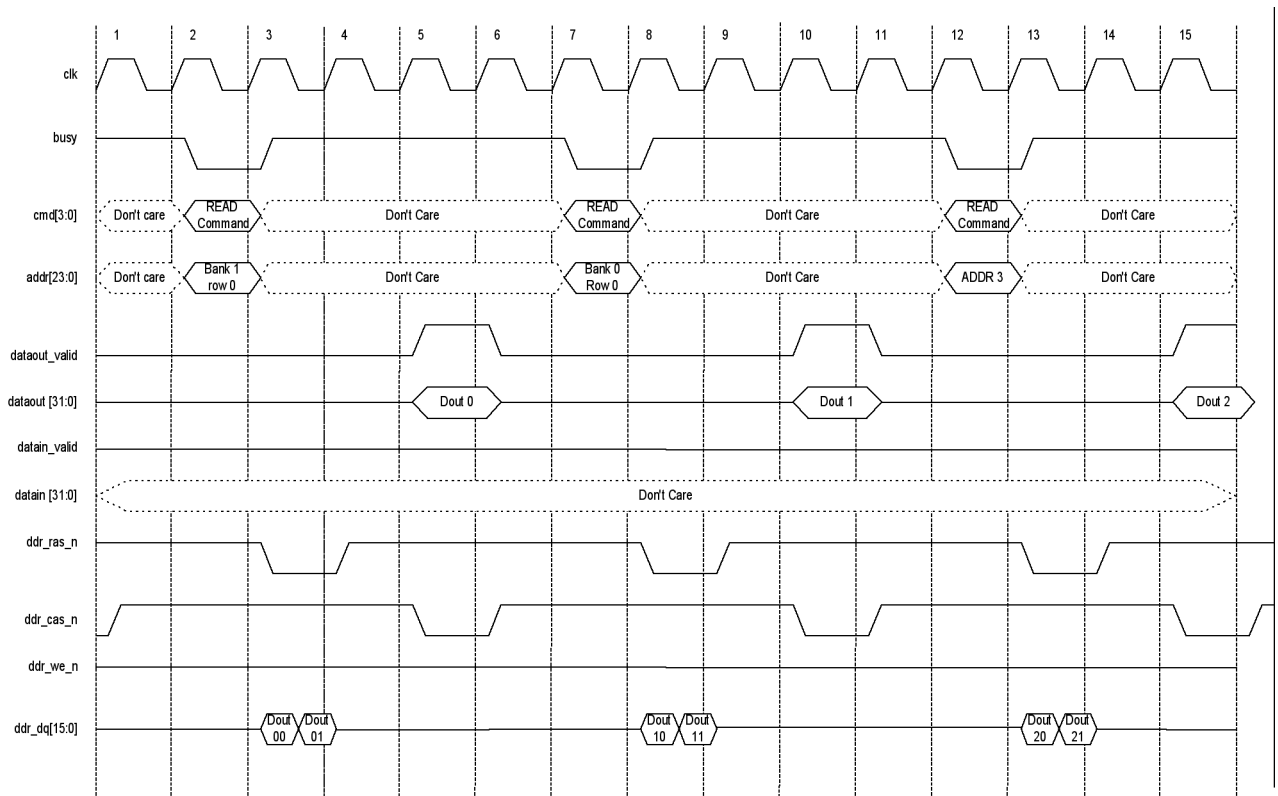


Figure 9. Read followed by Read (Different Bank Row Open Row Close BL = 2)

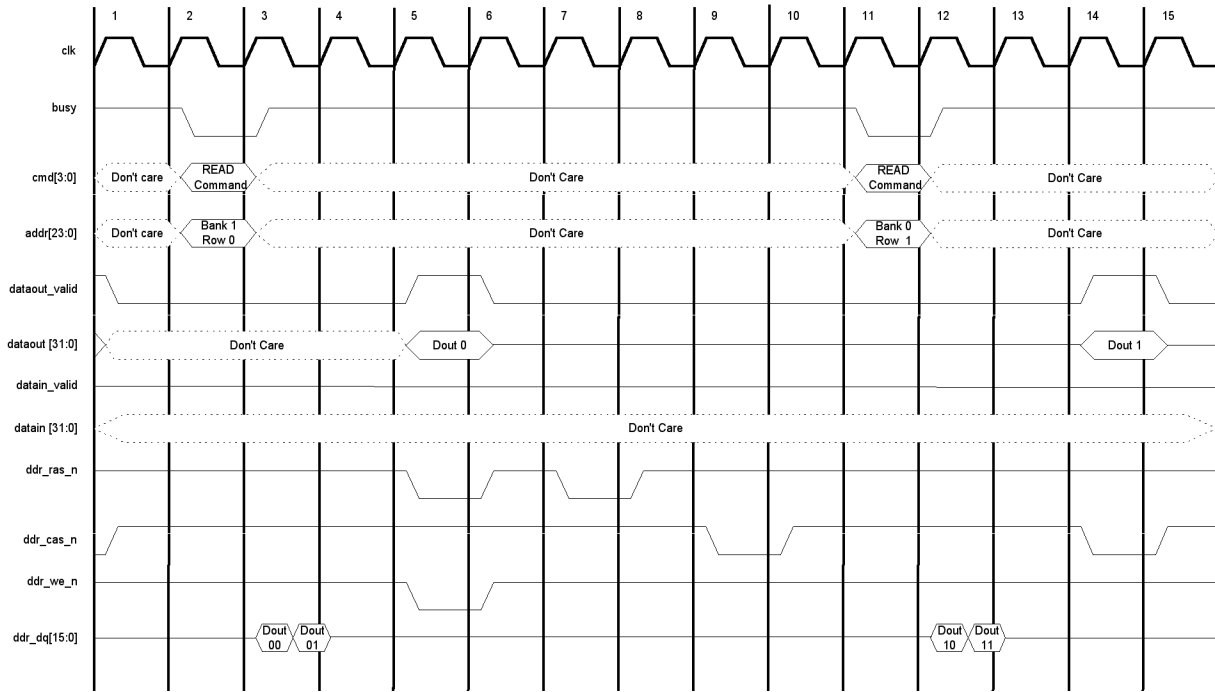


Figure 10. Read Followed by Read (Different Bank Row Closed BL = 2)

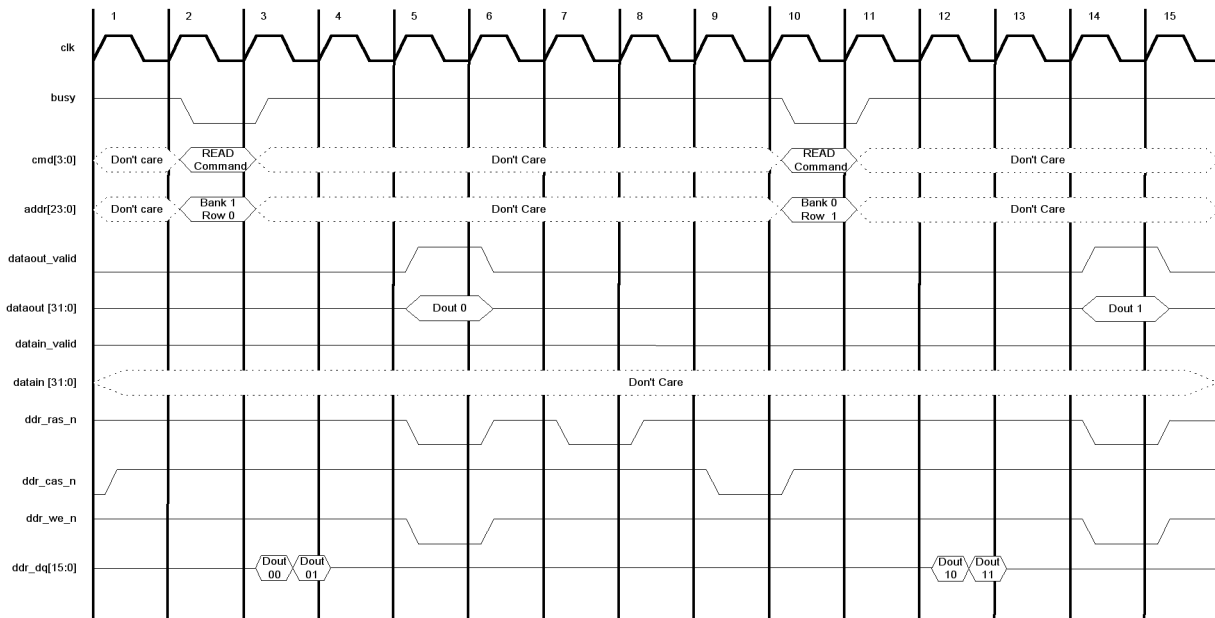


Figure 11. Write Followed by Write (Same Bank Same Row BL = 2)

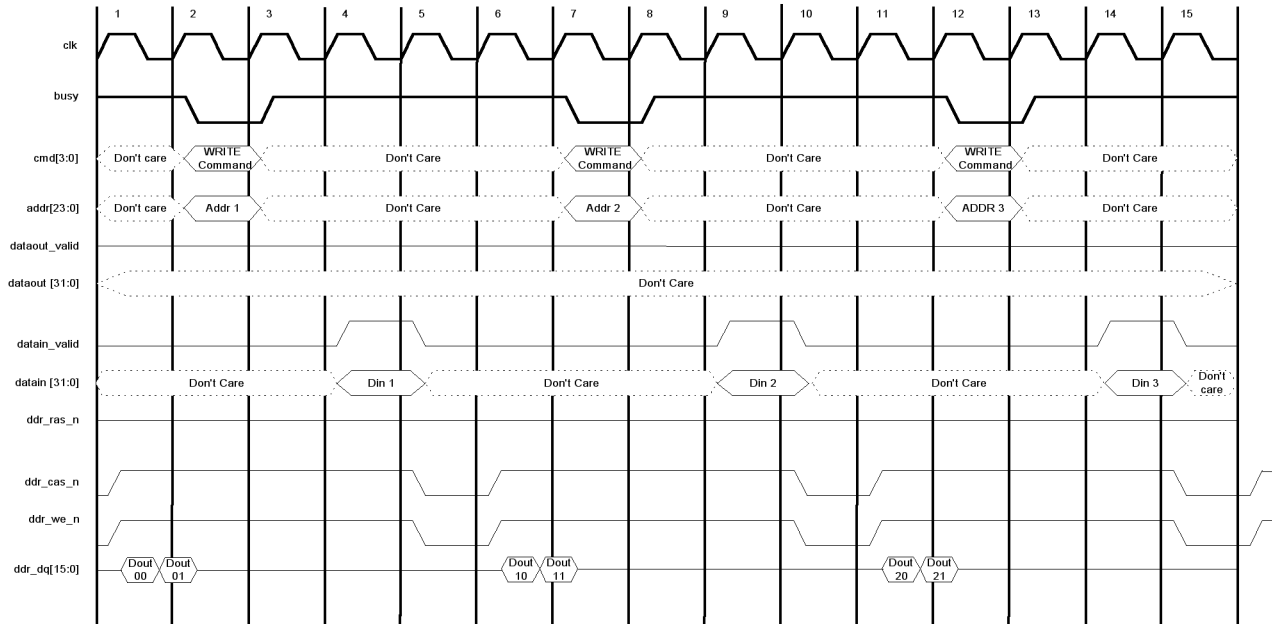


Figure 12. Write Followed by Write (Same Bank Different Row BL = 2)

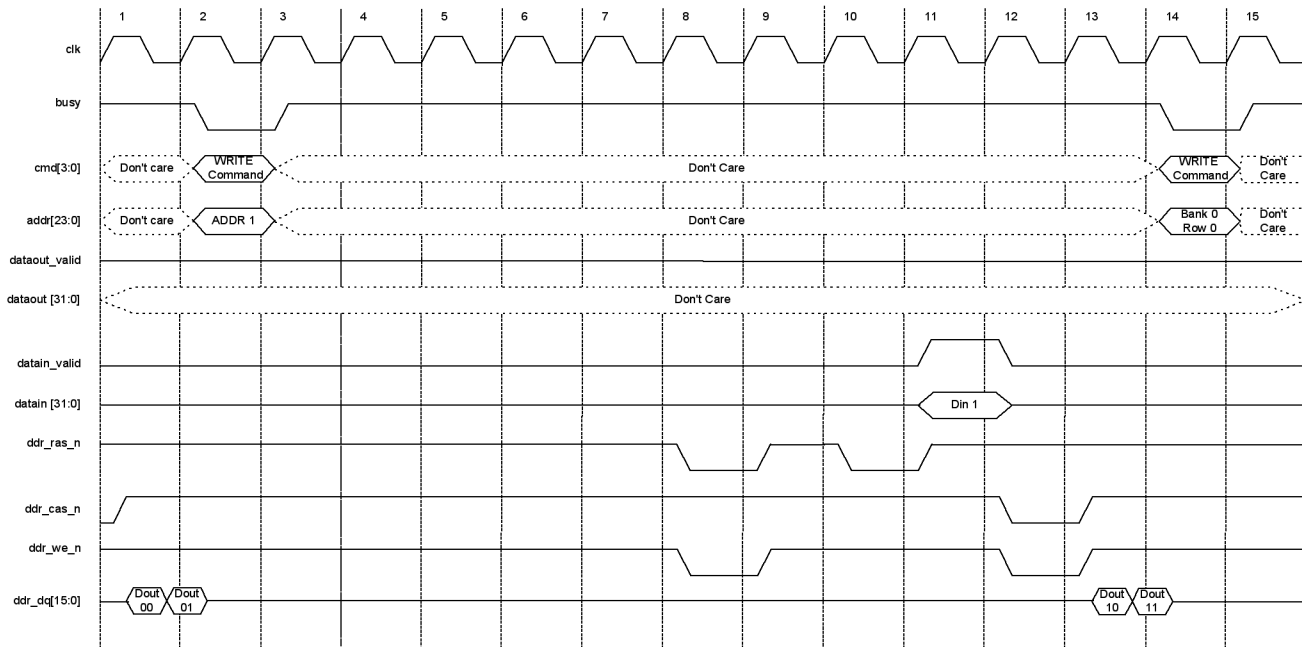


Figure 13. Write Followed by Write (Different Bank Row Open BL = 2)

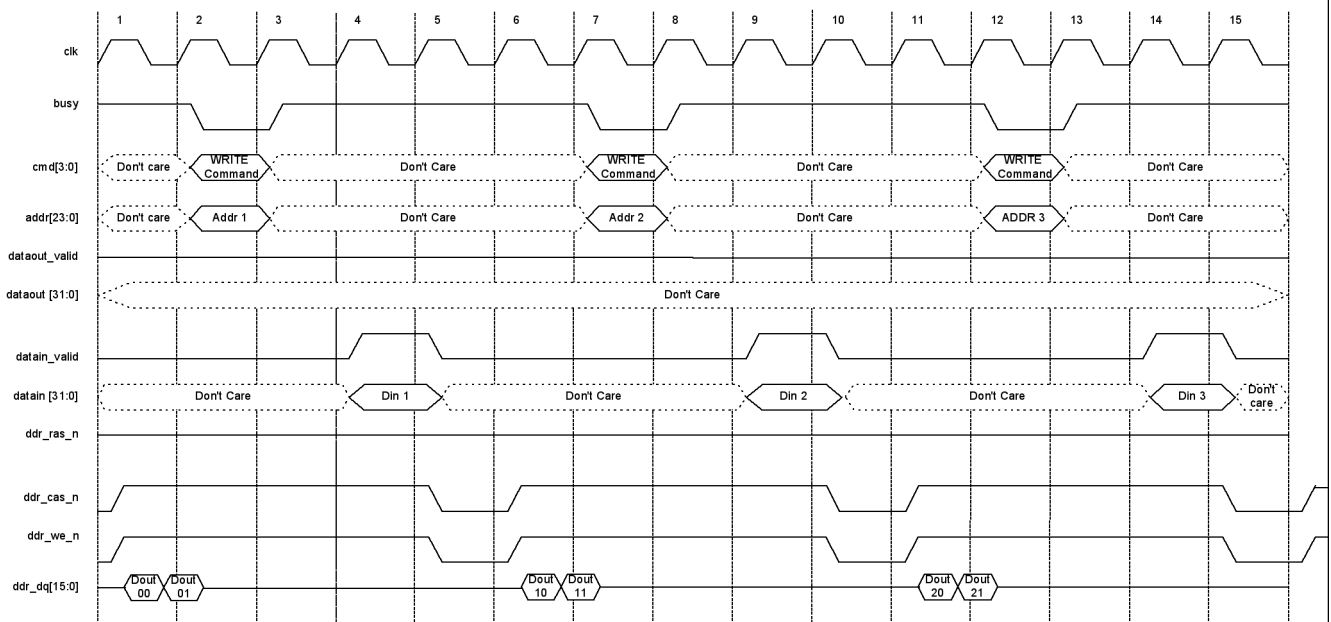


Figure 14. Write Followed by Write (Different Bank Row Open BL = 2)

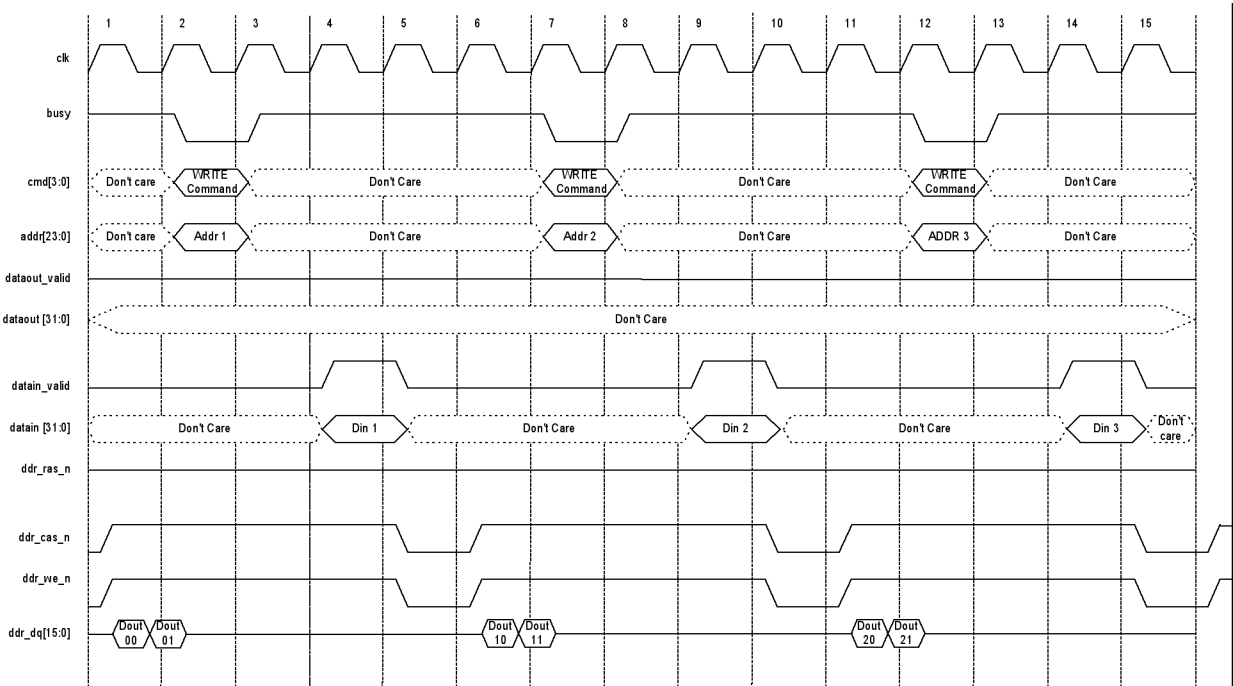


Figure 15. Write Followed by Write (Different Bank Row Open Row Close BL = 2)

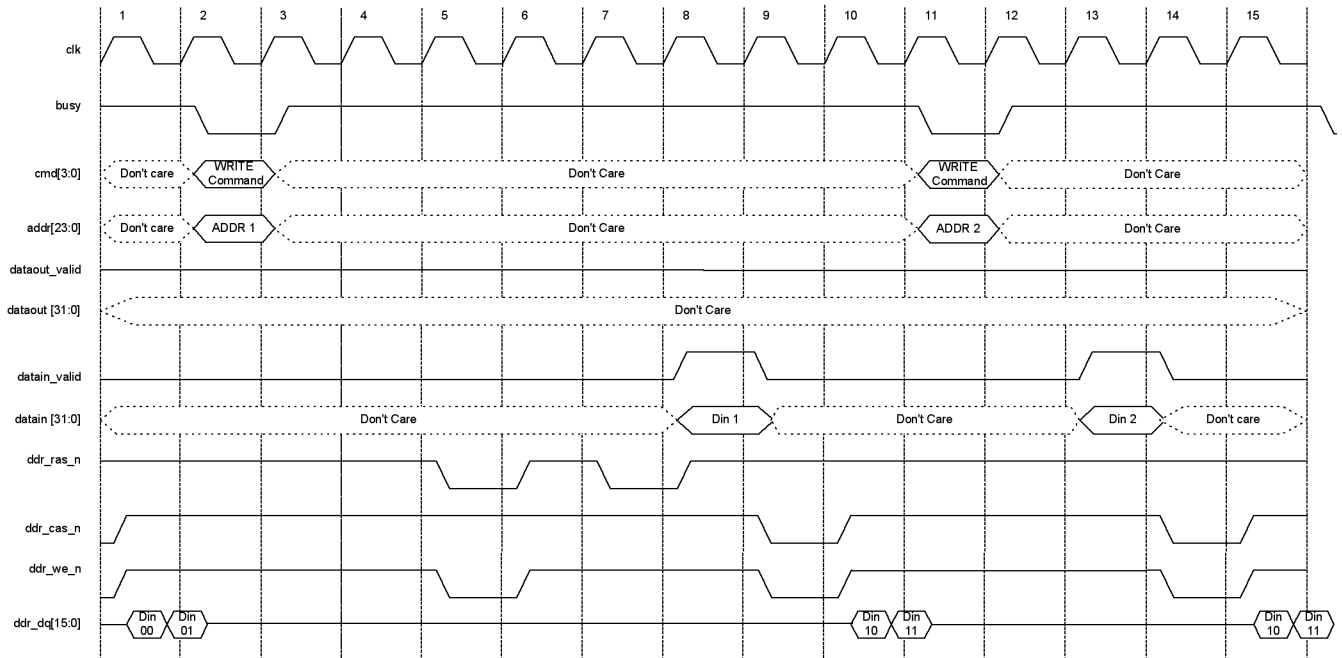


Figure 16. Write Followed by Write (Different Bank Row Open Row Close BL = 2)

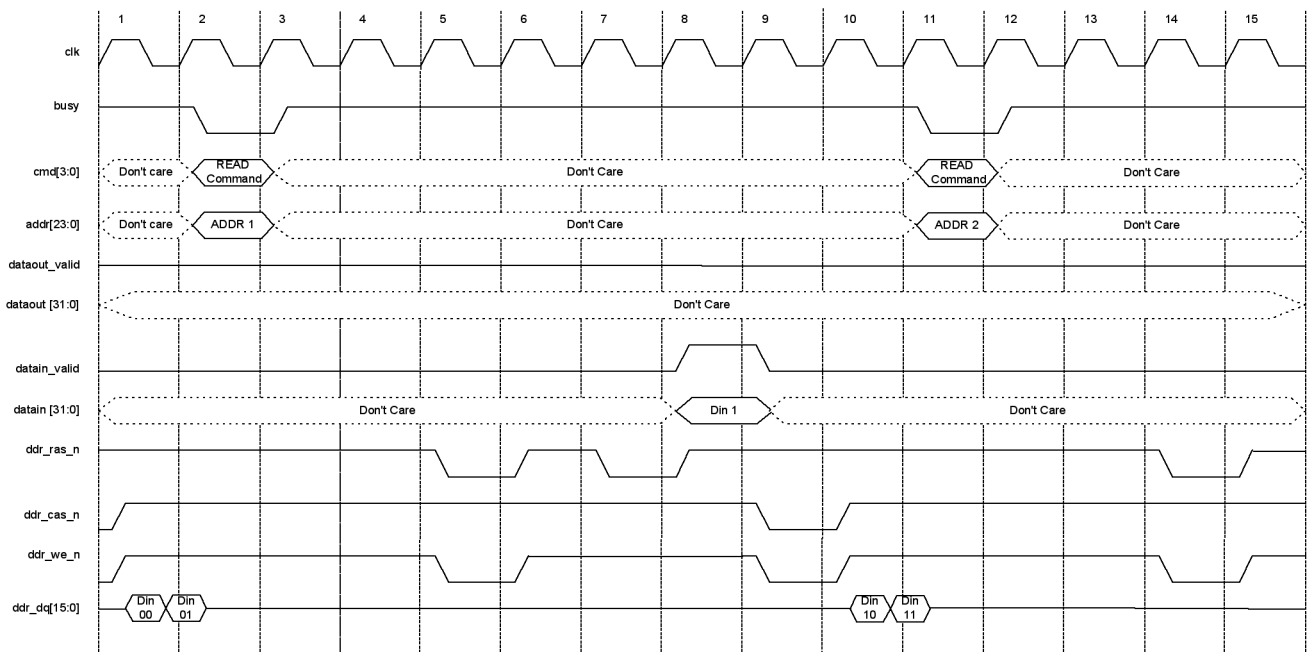


Figure 17. Power Down

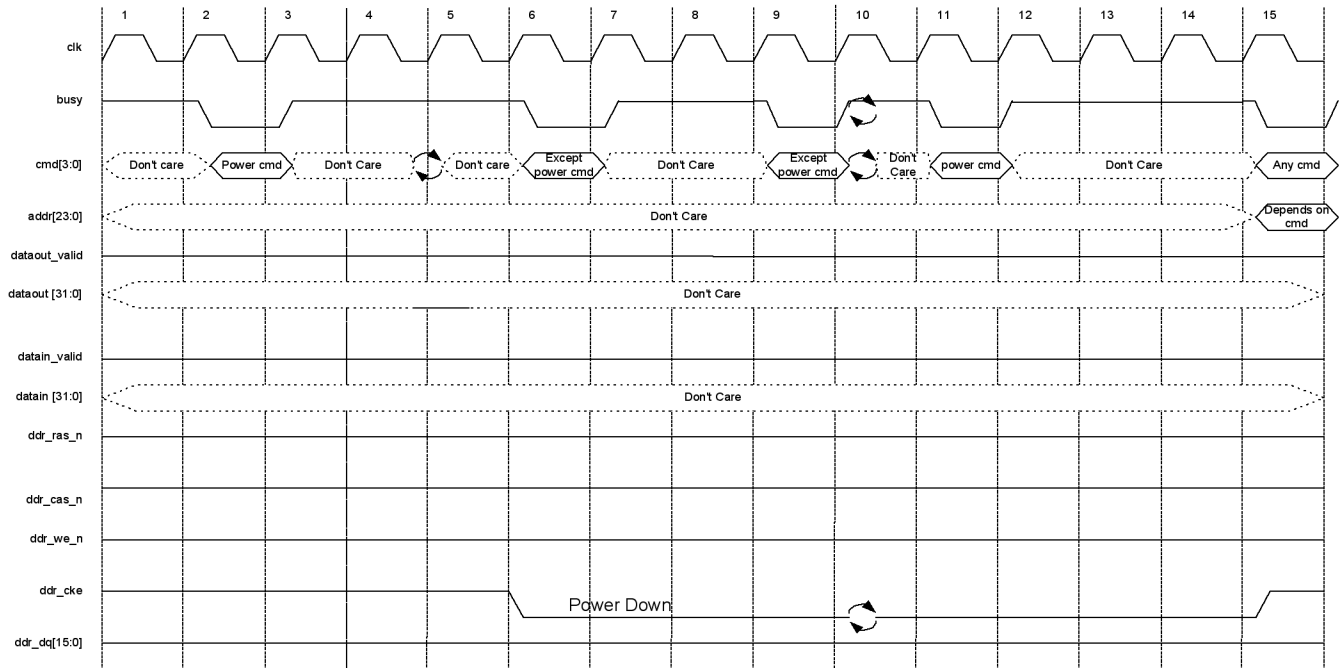
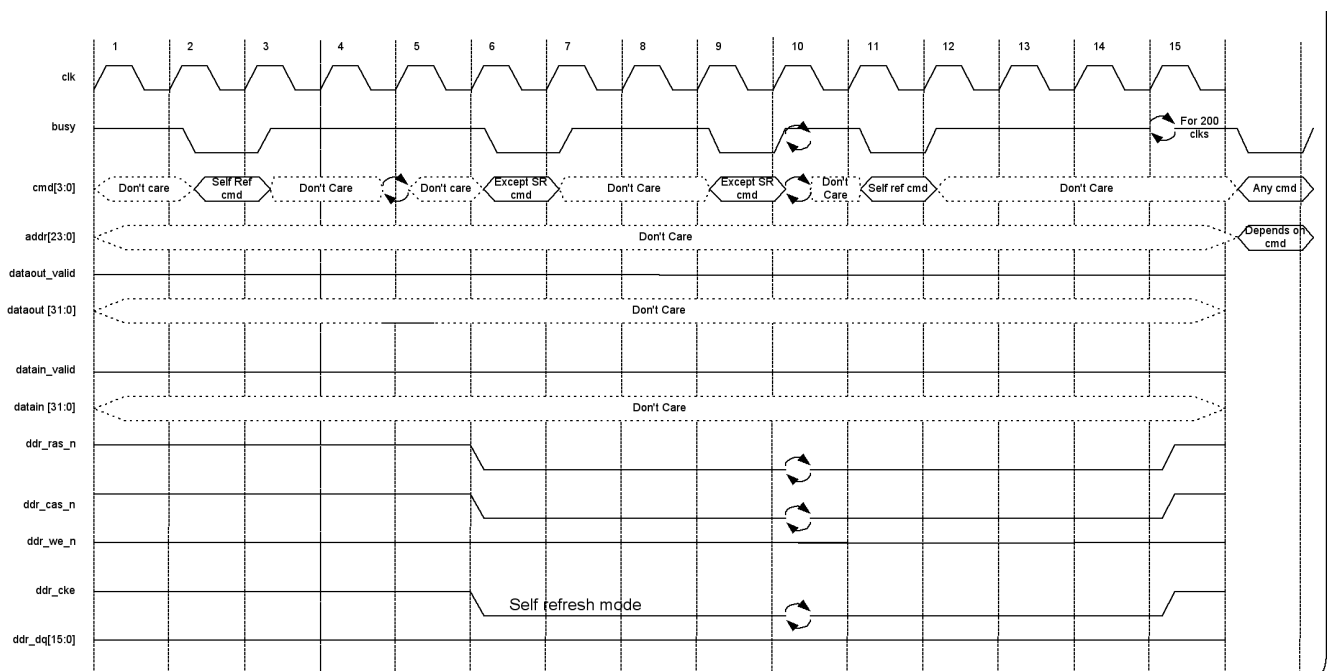


Figure 18. Self Refresh



Command Descriptions and Usage

This section describes the commands supported and their usage at the generic interface block.

NOP

This command is issued when the interface is waiting to issue commands. This command does not perform any operation on the DDR SDRAM.

| Command Name | Cmd[2:0] | Addr['ASIZE-1:0] | |
|--------------|----------|------------------|-------------------|
| | | Rank | Absolute Address |
| NOP | 000 | Rank Number | xxxx (Don't Care) |

READ

This command is issued when a READ is required from a memory location. The address is comprised of rank number and absolute address. The absolute address is formed by the row, bank and column addresses.

This read command automatically applies ACTIVATE, READ and PRECHARGE commands to the DDR SDRAM by looking at the bank and row address.

Once a READ command is issued, the dataout bus will contain valid read data when the dataout_valid signal is high.

| Command Name | Cmd[2:0] | Addr['ASIZE-1:0] | |
|--------------|----------|------------------|-------------------|
| | | Rank | Row, Bank, Column |
| READ | 001 | Rank Number | Absolute Address |

WRITE

This command is issued when a WRITE is required from a memory location. The address is composed of rank number and absolute address. The absolute address is formed by the row, bank and column addresses.

The WRITE command automatically applies ACTIVATE, READ and PRECHARGE commands to the DDR SDRAM by looking at the bank and row address.

After a WRITE command is issued, the controller accepts the data to be written from the datain bus when the datain_valid signal is high.

| Command Name | Cmd[2:0] | Addr['ASIZE-1:0] | |
|--------------|----------|------------------|-------------------|
| | | Rank | Row, Bank, Column |
| WRITE | 010 | Rank Number | Absolute Address |

SELF REFRESH

The SELF REFRESH command is issued to retain the data in the DDR SDRAM. This can be issued even if the rest of the system is powered down. In this mode, the DDR SDRAM retains data without applying an external clock signal. During this command, the address is "don't care" since all the DDR SDRAM devices enter self-refresh mode.

Once the command is given, the DDR Controller enters the self-refresh mode. The DDR Controller will remain in this mode until another SELF REFRESH command is sent. All other user commands are ignored while the DDR Controller is in the self-refresh mode.

| Command Name | Cmd[2:0] | Addr['ASIZE-1:0] | |
|--------------|----------|------------------|-------------------|
| | | Rank | Row, Bank, Column |
| SELF REFRESH | 110 | xxxx | xxxx (Don't Care) |

Load_MR

The Load_MR command is issued to program either the Mode Register or Extended Mode Register depending on the BA1:BA0 bits in the address.

BA1:BA0 = 00 Addresses the Mode Register

BA1:BA0 = 01 Addresses the Extended Mode Register

Once the Mode register is programmed, the CFG0 register in the controller is also updated

| Command Name | Cmd[2:0] | Addr['ASIZE-1:0] | | | |
|--------------|----------|------------------|-------------------|-------------|------------|
| | | Rank | Addr['ASIZE-1:14] | Addr[13:12] | Addr[11:0] |
| Load_MR | 100 | xxxx | xxxx | BA1:BA0 | A11:A0 |

POWER DOWN

The POWER DOWN command is issued to make the DDR SDRAM enter power down mode. The Controller automatically wakes up the DDR SDRAM, then puts the DDR SDRAM back into power down mode.

Once the command is given, the DDR Controller enters power-down mode. The power-down mode (and the DDR SDRAM) remains in power-down mode until another POWER DOWN command is sent. All other user commands are ignored while the DDR is in power-down mode.

| Command Name | Cmd[2:0] | Addr['ASIZE-1:0] | |
|--------------|----------|------------------|-------------------|
| | | Rank | Row, Bank, Column |
| POWER DOWN | 101 | xxxx | xxxx (Don't Care) |

Load_CFG

The Load_CFG command is used to program the Controller CFG0, CFG1, CFG2 and CFG3 registers. The controller register is selected with the A1:A0 bits in the address as shown in the table below.

The Load_CFG command can be used before the initialization of the DDR Controller.

| A[1:0] | Configuration Register Selected |
|--------|---------------------------------|
| 00 | CFG0 |
| 01 | CFG1 |
| 10 | CFG2 |
| 11 | CFG3 |

Once the CFG0 is programmed, the Mode Register in the DDR SDRAM is also updated.

| Command Name | Cmd[2:0] | Addr['ASIZE-1:0] | | | |
|--------------|----------|------------------|-------------------|------------|-----------|
| | | Rank | Addr['ASIZE-1:20] | Addr[19:2] | Addr[1:0] |
| Load_CFG | 011 | xxxx | xxxx | A[19:2] | A1:A0 |

The A[19:2] maps to the register (refer to the next section on Configuration Registers).

Configuration Registers

There are four Configuration registers in the DDR Controller: CFG0, CFG1, CFG2 and CFG3. These registers are selected using the Load_CFG command (see above). The A1 and A0 address bits determine the CFG register being addressed. A2 to A19 contain the data to be loaded into the selected register. See the description of the Load_CFG command for more information.

Configuration Register 0

The Configuration Register 0 (CFG0) is used to modify the DDR SDRAM Controller timing and behavior. Table 9 describes the contents of CFG0. When any change is made to this register, the DDR Controller automatically updates the DDR SDRAM mode register. The DDR SDRAM mode register is also updated when the DDR Controller receives a LOAD MODE REG command.

The user can program the `Burst Length`, `Burst Type`, and `CAS Latency` bits or use the default values without any programming.

After the power up condition is satisfied (all power supply and reference voltages are stable at approximately 200 μ s), the user triggers initialization of the DDR SDRAM by setting the `INIT` bit using the `Load_CFG` command. Once this bit is set, the controller starts the initialization process. The `busy` signal is held high until the initialization process is completed.

The default values for this register are set in the Verilog parameters file.

Table 9. Configuration Register 0 (CFG0)

| Configuration Bits | Parameter | Description |
|--------------------|--------------|---|
| CFG0[2:0] | Burst Length | Burst Length, valid values are 2, 4 and 8. 001b = 2 010b = 4 011b = 8 All others are reserved. Default value is Burst Length parameter. |
| CFG0[3] | Burst Type | Burst Type, Sequential / Interleaved 0 = Sequential 1 = Interleaved Default value is Burst Type parameter |
| CFG0[6:4] | CAS Latency | CAS Latency in number of clock cycles. 101b = 1.5 010b = 2.0 110b = 2.5 011b = 3.0 Default value is CAS Latency parameter |
| CFG0[7] | INIT | Initialize the DDR SDRAM when this bit is set. Default value is 0. |
| CFG0[19:8] | Reserved | These bits are reserved. Default value is unknown. |

Configuration Register 1

The Configuration Register 1 (CFG1) is used to modify the DDR SDRAM Controller timing and behavior. The correct settings will depend on the DDR SDRAM being used with the controller. Table 10 shows the contents of CFG1.

Table 10. Configuration Register 1 (CFG1)

| Configuration Bits | Parameter | Description |
|--------------------|------------|--|
| CFG1[2:0] | TRCD Delay | RAS to CAS Delay in number of clocks (maximum value 7 clocks). Default value is TRCD Delay parameter |
| CFG1[5:3] | TRRD Delay | Active bank A to active bank B command (maximum value 7 clocks). Default value is TRRD Delay parameter |
| CFG1[9:6] | TRFC Delay | AUTO REFRESH command delay period in number of clocks (maximum value 15 clocks). Default value is TRFC Delay parameter |
| CFG1[12:10] | TRP Delay | PRECHARGE Command period (maximum value 7 clocks). Default value is TRP Delay parameter |
| CFG1[15:13] | TMRD Delay | LOAD MODE REGISTER Command period (maximum value 7 clocks). Default value is TMRD Delay parameter |
| CFG1[18:16] | TWR Delay | Write recovery time (maximum value 7 clocks). Default value is TWR Delay parameter |
| CFG1[19] | Reserved | This bit is reserved. Default value is unknown. |

Configuration Register 2

The Configuration register 2 (CFG2) contains the 16 bits used to store the Refresh Period (RP). The DDR Controller regularly issues AUTO REFRESH commands to the DDR SDRAM during normal operation mode. Table 11 shows the contents of CFG2.

Table 11. Configuration Register 2 (CFG2)

| Configuration Bits | Parameter | Description |
|--------------------|----------------|---|
| CFG2[15:0] | Refresh Period | Refresh Period (max. = $t_{CK} * FFFFh$). Default value is Refresh Period parameter. |
| CFG2[19:16] | Reserved | These bits are reserved. Default value is unknown. |

Configuration Register 3

The Configuration register 3 (CFG3) is used to modify the DDR SDRAM Controller timing and behavior. The correct settings will depend on the DDR SDRAM being used with the controller. Table 12 shows the contents of CFG3.

Table 12. Configuration Register 3 (CFG3)

| Configuration Bits | Parameter | Description |
|--------------------|------------|--|
| CFG3[3:0] | TRAS Delay | Active to pre-charge command (maximum value of 15 clocks). Default value is TRAS Delay parameter. |
| CFG3[6:4] | TWTR Delay | Internal write to read command delay (maximum value of 7 clocks). Default value is TWTR Delay parameter. |
| CFG3[10:7] | TRC Delay | Active to Active/AUTO REFRESH command delay period in number of clocks (maximum value of 15 clocks). Default value is TRC Delay parameter. |
| CFG3[19:11] | Reserved | These bits are reserved. Default value is unknown. |

The following procedure is required for the ORCA® Series 4 version of this IP core. For other device versions, refer to the Readme release notes included in that evaluation package.

Before going to the Place and Route, the user needs to run `edifmod.pl` provided in the package to generate a new EDIF netlist. Installation of Perl programming software is required before running the script (`edifmod.pl`) provided. To download Perl programming software, please visit www.activeperl.com for the latest version of the software. The script (`edifmod.pl`) will allow user to insert the I/O types property to the specific instances that declare in the port inside the EDIF netlist. The current LeonardoSpectrum synthesis tool only inserts I/O type property on ports and the ispLEVER software requires the I/O type property inserted on the instances. Therefore, the script (`edifmod.pl`) is provided to work around the problem.

Open DOS-shell, change directory to where the EDIF netlist located and type:

```
perl edifmod.pl <EDIF name>.edf
```

By running the `edifmod.pl` provided, it will generate a new EDIF netlist as `<EDIF name>.edf.new`. The new EDIF netlist need to be renamed to `<new EDIF name>.edf` before importing the netlist.

References

- Double Data Rate (DDR) SDRAM Data Sheet, Micron Technology, Inc., 2001.
- 128 M-bit Synchronous DRAM with Double Data Rate Data Sheet, NEC Corp., December 1998.
- DDR SDRAM Specification Version 0.3, Samsung Electronics, 2000.
- DDR SDRAM Controller Data Sheet, Lattice Semiconductor Corporation, 2003.
- ispLeverCORE™ Evaluation Tutorials, Lattice Semiconductor Corp., 2003.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-408-826-6002 (Outside North America)

e-mail: techsupport@latticesemi.com

Internet: www.latticesemi.com

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Lattice:

[DDRCT-AHB-5X-N1](#) [DDRCT-AHB-O4-N1](#) [DDRCT-AHB-XP-N1](#) [DDRCT-GEN-5X-N1](#) [DDRCT-GEN-O4-N1](#) [DDRCT-GEN-XP-N1](#) [DDRCT-NP-E2-N3](#) [DDRCT-NP-O4-N1](#) [DDRCT-PCI-5X-N1](#) [DDRCT-PCI-O4-N1](#) [DDRCT-PCI-XP-N1](#) [DDRCT-PPC-5X-N1](#) [DDRCT-PPC-O4-N1](#) [DDRCT-PPC-XP-N1](#) [DDRCT-GEN-P2-U6](#) [DDRCT-NP-XM-N3](#) [DDRCT-GEN-E2-U6](#) [DDRCT-GEN-PM-U6](#) [DDRCT-GEN-SC-U6](#) [DDRCT-GEN-XM-U6](#) [DDRCT-GEN-X2-U6](#) [DDRCT-GEN-E3-U6](#) [DDRCT-GEN-P2-UT6](#) [DDRCT-GEN-X2-UT6](#) [DDRCT-GEN-E3-UT6](#)



Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.



Как с нами связаться

Телефон: 8 (812) 309 58 32 (многоканальный)

Факс: 8 (812) 320-02-42

Электронная почта: org@eplast1.ru

Адрес: 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.