Rate This Document

CoreDDR v4.0

Handbook

Actel®

POWER MATTERS

Actel Corporation, Mountain View, CA 94043

© 2008 Actel Corporation. All rights reserved.

Printed in the United States of America

Part Number: 50200109-2

Release: July 2009

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Actel.

Actel makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability or fitness for a particular purpose. Information in this document is subject to change without notice. Actel assumes no responsibility for any errors that may appear in this document.

This document contains confidential proprietary information that is not to be disclosed to any unauthorized person without prior written consent of Actel Corporation.

Trademarks

Actel, IGLOO, Actel Fusion, ProASIC, Libero, Pigeon Point and the associated logos are trademarks or registered trademarks of Actel Corporation. All other trademarks and service marks are the property of their respective owners.

Table of Contents

Introduction . . . 5
General Description . . . 5
Key Features . . . 5
Supported Device Families . . . 5
Core Versions . . . 5
CoreDDR Device Utilization and Performance . . . 5

1 Functional Block Description . . . 7
Address Mapping . . . 9
CoreDDR Operation . . . 9
Standard DDR Configurations . . . 10
Instruction Timing . . . 11

2 Tool Flows . . . 13
Licenses . . . 13
SmartDesign . . . 13
Importing into Libero IDE . . . 14
Simulation Flows . . . 14
Synthesis in Libero IDE . . . 14

3 CoreDDR . . . 15
Generics . . . 15
Controller Configuration Ports . . . 16

4 Core Interfaces . . . 19
Local Bus Signals . . . 19
DDR SDRAM Interface Signals . . . 20

5 Timing Diagrams . . . 21
SDRAM Writes . . . 21
SDRAM Reads . . . 23
Auto-Precharge . . . 25

6 Testbench Operation . . . 27
Testbench Description . . . 27
Verilog User Testbench . . . 27
VHDL User Testbench . . . 27

7 Implementation Hints . . . 29
Data Capture . . . 29

8 Ordering Information . . . 33
Ordering Codes . . . 33

A List of Document Changes 35

B Product Support 37
Customer Service 37
Actel Customer Technical Support Center 37
Actel Technical Support 37
Website 37
Contacting the Customer Technical Support Center 37

Index 39

Introduction

General Description

CoreDDR provides a high-performance interface to double data rate (DDR) synchronous dynamic random access memory (SDRAM) devices. CoreDDR accepts *read* and *write* commands using the simple local bus interface and translates these requests to the command sequences required by SDRAM devices. CoreDDR also performs all initialization and refresh functions.

CoreDDR uses bank management techniques to monitor the status of each SDRAM bank. Banks are only opened or closed when necessary, minimizing access delays. Up to four banks can be managed at one time. Access cascading is also supported, allowing read or write requests to be chained together. This results in no delay between requests, enabling up to 100% memory throughput for sequential accesses.

Key Features

CoreDDR is a highly configurable core and has the following features:

- High performance, double data rate controller for standard SDRAM chips and DIMMs
- Synchronous interface, fully pipelined internal architecture
- Supports up to 1,024 MB of memory
- Bank management logic monitors status of up to four SDRAM banks

Supported Device Families

- ProASIC3®E
- Fusion

Core Versions

This handbook applies to CoreDDR v4.0.

CoreDDR Device Utilization and Performance

Table 1 gives device utilization and performance for Actel CoreDDR.

Table 1 · CoreDDR Device Utilization and Performance

Family	**Cells or Tiles**			**Utilization**		**Performance**
	Sequential	**Combinatorial**	**Total**	**Device**	**Total**	
ProASIC3E	795	953	1,748	A3PE600-2	12%	133 MHz
Fusion	795	953	1,748	AFS600-2	12%	133 MHz

Note: All data was obtained using a typical system configuration with the local bus tied to internal user logic and the controller configuration inputs hardcoded during synthesis.

Functional Block Description

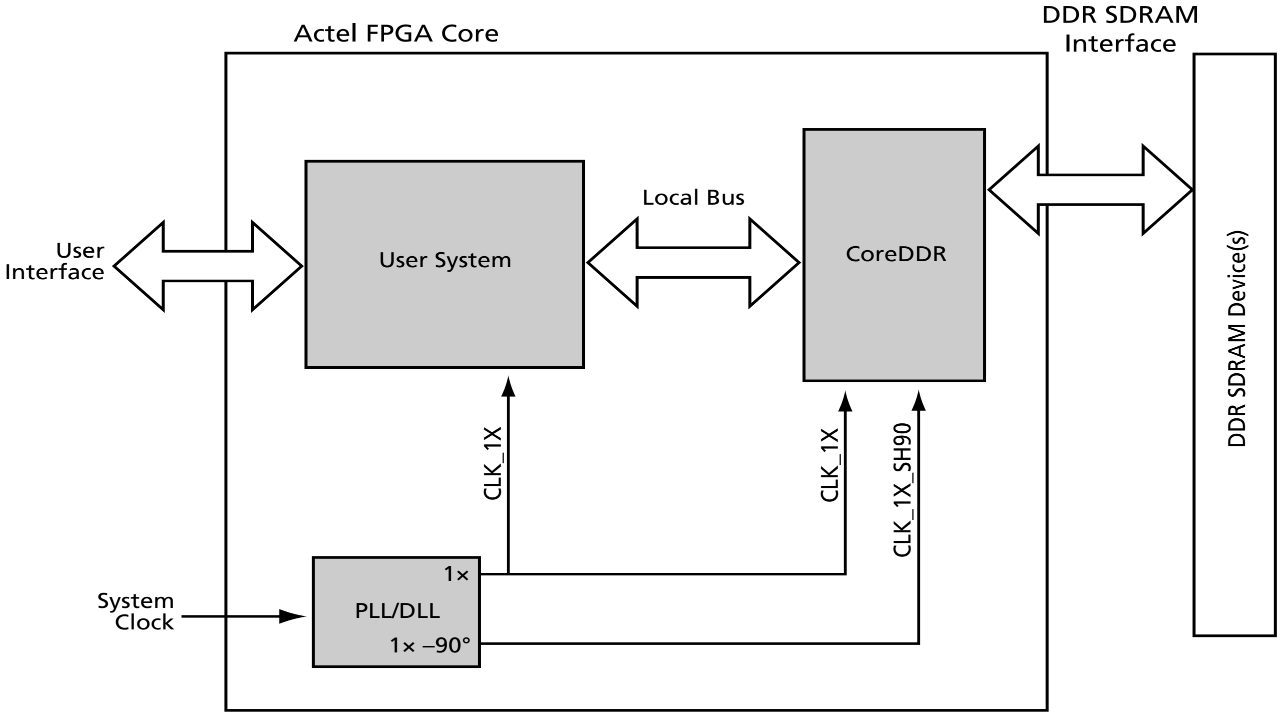


Figure 1-1 · Typical CoreDDR System[1]

CoreDDR is provided with runtime-programmable inputs for all timing parameters (CAS latency, t_{RAS}, t_{RC}, t_{RFC}, t_{RCD}, t_{RP}, t_{MRD}, t_{RRD}, t_{REFC}, t_{WR}) and memory configuration settings. This ensures compatibility with virtually any SDRAM configuration. CoreDDR is also available in netlist format with user-defined timing and memory configuration parameters for designs requiring low logic utilization or for designs requiring high-clock-rate operation.

CoreDDR consists of the following primary blocks, as shown in Figure 1-2 on page 8:

1. Control and Timing Block – Main controller logic
2. Initialization Control – Performs initialization sequence after RESET_N is deactivated or SD_INIT is pulsed.
3. Address Generation – Puts out address, bank address, and chip select signals on SDRAM interface.
4. Bank Management – Keeps track of last opened row and bank to minimize command overhead.
5. Refresh Control – Performs automatic refresh commands to maintain data integrity.
6. DDR Data Control – Handles multiplexing and demultiplexing of data to and from the DDR SDRAM devices.

For the DDR SDRAM controller, the data passes through the controller, and the controller handles all DDR-related synchronization and timing generation. The DDR SDRAM controller uses a –90° phase-shifted system clock to capture read data in the center of the data window. The core also has separate DATAIN and DATAOUT busses at the user interface. These busses are twice as wide as the data bus going to the DDR SDRAM. This multiplexing is required because the local (user) interface operates at single data rate, whereas the SDRAM data interface operates at double data rate.

---

[1] *The external tristate buffers shown in Figure 1-1 reside in the I/O. The output enables of the tristate buffers are controlled by CoreDDR.*

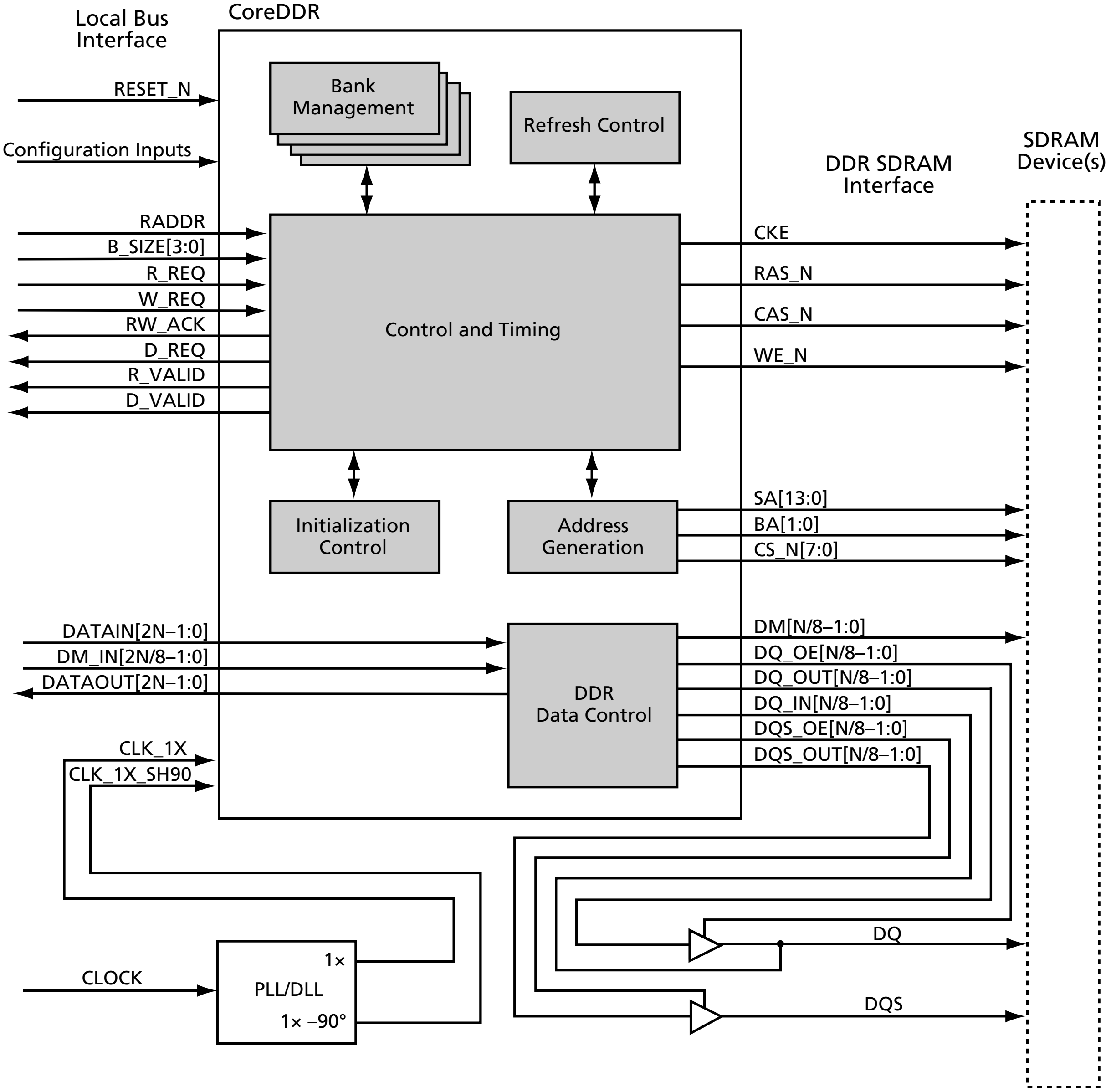

Local Bus Interface
CoreDDR
RESET_N
Configuration Inputs
Bank Management
Refresh Control
RADDR
B_SIZE[3:0]
R_REQ
W_REQ
RW_ACK
D_REQ
R_VALID
D_VALID
Control and Timing
DDR SDRAM Interface
SDRAM Device(s)
CKE
RAS_N
CAS_N
WE_N
Initialization Control
Address Generation
SA[13:0]
BA[1:0]
CS_N[7:0]
DATAIN[2N–1:0]
DM_IN[2N/8–1:0]
DATAOUT[2N–1:0]
DDR Data Control
DM[N/8–1:0]
DQ_OE[N/8–1:0]
DQ_OUT[N/8–1:0]
DQ_IN[N/8–1:0]
DQS_OE[N/8–1:0]
DQS_OUT[N/8–1:0]
CLK_1X
CLK_1X_SH90
CLOCK
PLL/DLL
1×
1× –90°
DQ
DQS

Figure 1-2 · DDR SDRAM Controller Block Diagram

Address Mapping

The mapping of the RADDR bus at the local bus interface to the chip select, row, column, and bank addresses is shown in Figure 1-3. The exact bit positions of the mapping will vary depending on the rowbits and colbits configuration port settings. The column bits, bank bits, row bits, and chip select are mapped from the least significant bits of RADDR. By mapping the bank bits from this location, long accesses to contiguous address space are more likely to take place without the need for a precharge.

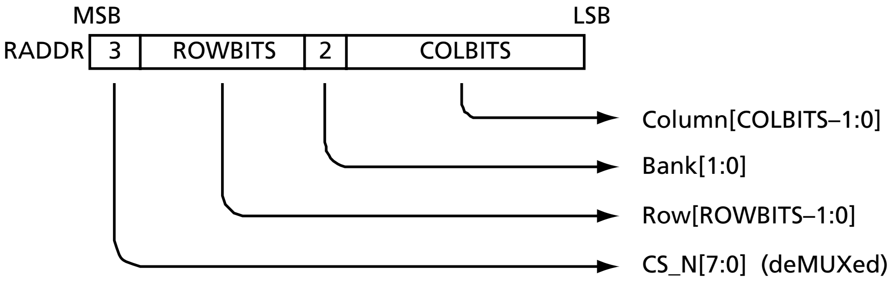


Figure 1-3 · Mapping of RADDR to SDRAM Chip Select, Row, Column, and Bank

CoreDDR Operation

The synchronous interface and fully pipelined internal architecture of SDRAM allow extremely fast data rates if used efficiently. SDRAM is organized in banks of memory addressed by row and column. The number of row and column address bits depends on the size and configuration of the memory.

SDRAM is controlled by bus commands formed using combinations of the RAS_N, CAS_N, and WE_N signals. For instance, on a clock cycle where all three signals are High, the associated command is a *no operation* (NOP). A NOP is also indicated when the chip select is not asserted. The standard SDRAM bus commands are shown in Table 1-1.

Table 1-1 · SDRAM Bus Commands

Command	RAS_N	CAS_N	WE_N
NOP	H	H	H
Active	L	H	H
Read	H	L	H
Write	H	L	L
Burst Terminate	H	H	L
Precharge	L	H	L
Auto-Refresh	L	L	H
Load Mode Register	L	L	L

SDRAM devices are typically divided into four banks. These banks must be opened before a range of addresses can be written to or read from. The row and bank to be opened are registered coincident with the *active* command. When a new row on a bank is accessed for a read or a write, it may be necessary to first close the bank and then reopen the bank to the new row. A bank is closed using the *precharge* command. Opening and closing banks costs memory bandwidth, so CoreDDR has been designed to monitor and manage the status of the four banks simultaneously. This enables the controller to intelligently open and close banks only when necessary.

When a *read* or *write* command is issued, the initial column address is presented to the SDRAM devices. In the case of DDR SDRAM, the initial data is presented one clock cycle after the *write* command. For the *read* command, the initial data appears on the data bus 1–4 clock cycles later. This delay, known as CAS latency, is due to the time required to physically read the internal DRAM and register the data on the bus. The CAS latency depends on the speed grade of the SDRAM and the frequency of the memory clock. In general, the faster the clock, the more cycles of CAS latency required.

After the initial *read* or *write* command, sequential reads and writes will continue until the burst length is reached or a *burst terminate* command is issued. SDRAM devices support a burst length of up to eight data cycles. The SDRAM controller is capable of cascading bursts to maximize SDRAM bandwidth.

SDRAM devices require periodic refresh operations to maintain the integrity of the stored data. CoreDDR automatically issues the *auto-refresh* command periodically. No user intervention is required.

The *load mode register* command is used to configure SDRAM operation. This register stores the CAS latency, burst length, burst type, and write burst mode. The SDRAM controller supports sequential burst type and programmed-length write burst mode. An additional extended mode register is used to control the integrated delay-locked loop (DLL) and the dq output drive strength. The DDR controller writes to the base mode register and extended mode register during the initialization sequence. Consult the SDRAM device specification for additional details on these registers.

Standard DDR Configurations

To reduce pin count, SDRAM row and column addresses are multiplexed on the same pins. Table 1-2 lists the number of rows, columns, banks, and chip selects required for various standard discrete DDR SDRAM devices. CoreDDR will support any of these devices.

SDRAM is typically available as dual in-line memory modules (DIMMs), small outline DIMMs (SO-DIMMs), and discrete chips. The number of row and column bits for a DIMM or SO-DIMM configuration can be found by determining the configuration of the discrete chips used on the module. This information is available in the module datasheet.

Table 1-2 · Standard DDR SDRAM Device Configurations

Chip Size	Configuration	Rows	Columns	Banks
64 MB	16M×4	12	10	4
64 MB	8M×8	12	9	4
64 MB	4M×16	12	8	4
64 MB	2M×32	11	8	4
128 MB	32M×4	12	11	4
128 MB	16M×8	12	10	4
128 MB	8M×16	12	9	4
128 MB	4M×32	12	8	4
256 MB	64M×4	13	11	4
256 MB	32M×8	13	10	4
256 MB	16M×16	13	9	4
512 MB	128M×4	13	12	4
512 MB	64M×8	13	11	4
512 MB	32M×16	13	10	4
1,024 MB	256M×4	14	12	4
1,024 MB	128M×8	14	11	4
1,024 MB	64M×16	14	10	4

Instruction Timing

Initialization

After RESET_N is deasserted or SD_INIT is pulsed, CoreDDR performs the following sequence:

1. NOP command for 200 µs (delay controlled by the delay port parameter)
2. *Precharge-all* command
3. *Extended load mode register* command to enable the DDR SDRAM DLL
4. *Load mode register* command with DLL reset bit set
5. *Precharge-all* command
6. Two *auto-refresh* commands
7. *Load mode register* command with DLL reset bit clear and proper mode settings
8. Read/write requests held off for 200 clock cycles to allow for DLL to stabilize

CoreDDR initialization timing is shown in Figure 1-4.

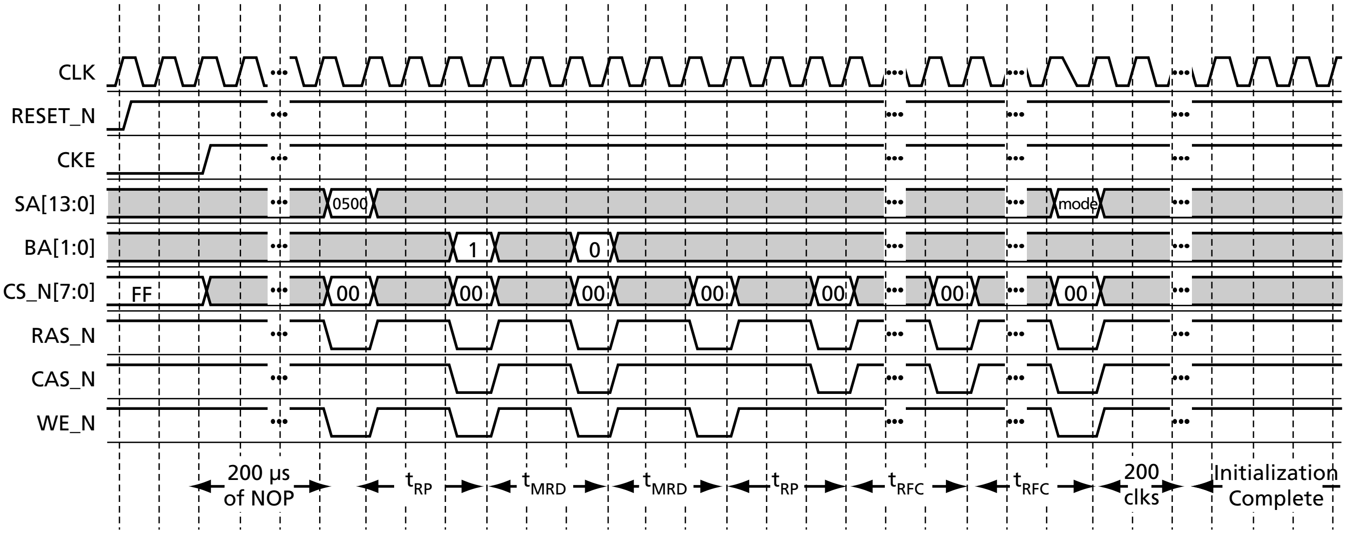

CLK
RESET_N
CKE
SA[13:0]
BA[1:0]
CS_N[7:0]
RAS_N
CAS_N
WE_N
0500
mode
1
0
FF
00
00
00
00
00
00
00
200 µs of NOP
t_{RP}
t_{MRD}
t_{MRD}
t_{RP}
t_{RFC}
t_{RFC}
200 clks
Initialization Complete

Figure 1-4 · DDR Initialization Sequence

Auto-Refresh

SDRAM devices require periodic *auto-refresh* commands to maintain data integrity. CoreDDR will automatically issue periodic *auto-refresh* commands to the SDRAM device(s) without user intervention.

The refresh period configuration port (REF) specifies the period between refreshes, in clock cycles. Figure 1-5 shows an example of two refresh commands. The first refresh sequence occurs when one or more banks have been left open as a result of a *read without precharge* or *write without precharge* operation. All open banks are closed using the *precharge-all* command (RAS_N, WE_N asserted with SA[10] and SA[8]) prior to the refresh command. In Figure 1-5, a refresh occurs again after the refresh period has elapsed, as determined using the REF configuration port. The refresh will never interrupt a read or write in the middle of a data burst. However, if the controller determines that the refresh period has elapsed at a point concurrent with or prior to a read or write request, the request may be held off (RW_ACK will not get asserted) until after the refresh has been performed.

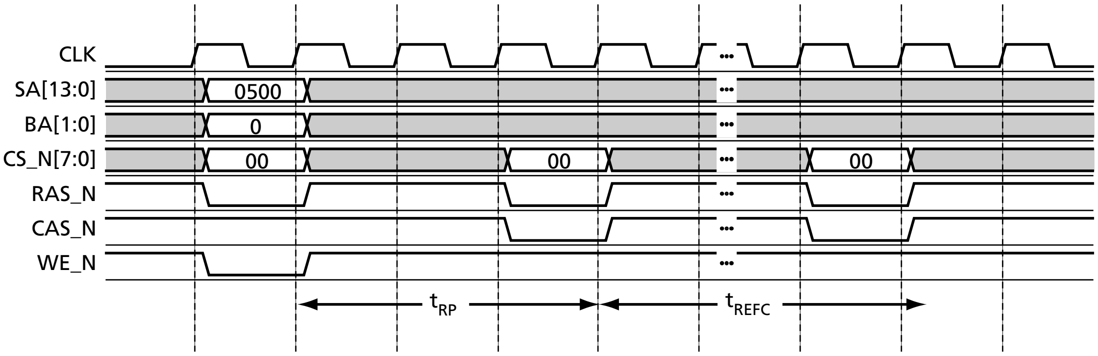


Figure 1-5 · Refresh Timing

Bank Management

CoreDDR incorporates bank management techniques to minimize command overhead. For each bank, the controller records the last opened row and whether the bank has been closed. When a local bus interface *read* or *write* request occurs, CoreDDR determines if the requested bank has already been opened and whether the request is for the same row as the one the bank is already opened to. If the bank is already opened to the requested row, CoreDDR performs the function immediately. If the bank was previously opened with a different row and was not closed, the controller closes the previously opened bank (using the *precharge* command) and reopens the bank (using the *active* command) to the requested row. If the bank is already closed, the controller opens the bank to the requested row (using the *active* command).

Requests to the controller can be issued as *read with auto-precharge*, *write with auto-precharge*, *read without auto-precharge*, and *write without auto-precharge*. Commands are issued with auto-precharge if the AUTO_PCH signal is set concurrent with the read request (R_REQ) or write request (W_REQ) signal. After a *read with auto-precharge* or *write with auto-precharge*, the accessed bank is automatically closed internally by the SDRAM devices. After a *read without auto-precharge* or *write without auto-precharge*, the accessed bank is left open until closing is required. Closing occurs whenever a request is issued to a row other than the one the bank is already open to, or during the next refresh sequence. The refresh sequence will close all the banks (using the *precharge-all* command) if all banks are not already closed.

The default configuration of the controller tracks the status of four banks at a time. This means that accesses to *row a* on *bank a* on *chip select a* is considered to be on a different row from *row a* on *bank a* on *chip select b*. Therefore, a close-and-open sequence is performed when switching between these rows.

2

Tool Flows

Licenses

CoreDDR is licensed in two ways. Depending on your license, tool flow functionality may be limited.

Obfuscated

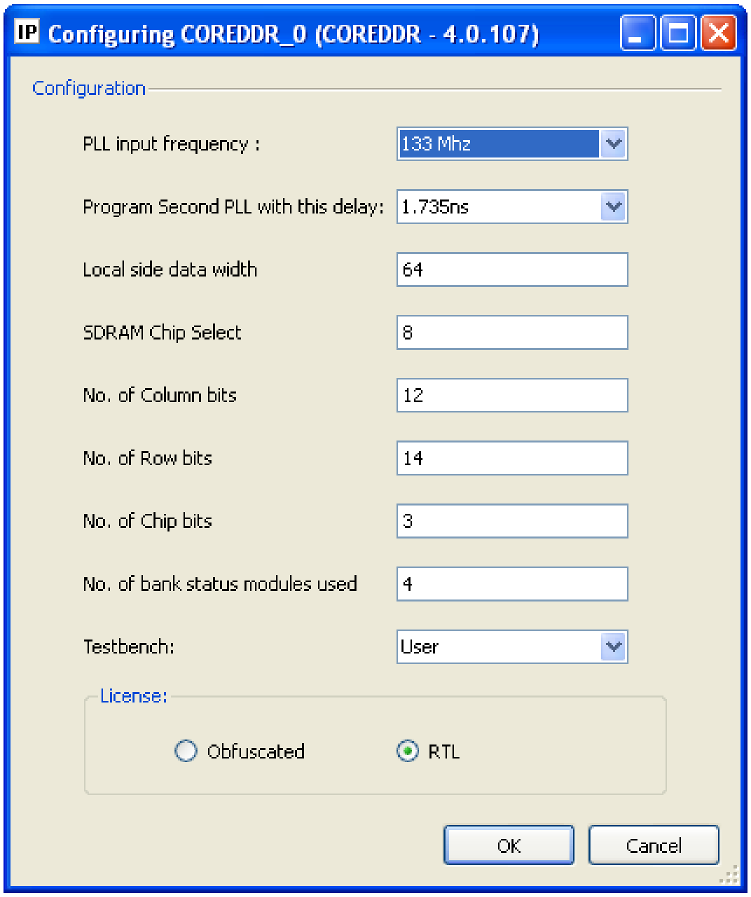
Complete RTL code is provided for the core, enabling the core to be instantiated with SmartDesign. Simulation, synthesis, and layout can be performed with Actel Libero® Integrated Design Environment (IDE). The RTL code for the core is obfuscated, and some of the testbench source files are not provided. They are pre-compiled into the compiled simulation library instead.

RTL

Complete RTL source code is provided for the core and testbenches.

SmartDesign

The core can be configured using the configuration GUI within SmartDesign, as shown in Figure 2-1.



PLL input frequency : 133 Mhz

Program Second PLL with this delay: 1.735ns

Local side data width 64

SDRAM Chip Select 8

No. of Column bits 12

No. of Row bits 14

No. of Chip bits 3

No. of bank status modules used 4

Testbench: User

License: ○ Obfuscated ● RTL

Figure 2-1 · CoreDDR Configuration Within SmartDesign

Importing into Libero IDE

Libero IDE is available for download to the SmartDesign IP Catalog, via the Libero IDE web repository. For information on using SmartDesign to instantiate, configure, connect, and generate cores, refer to the Libero IDE online help.

Simulation Flows

To run simulations, select the user testbench within the SmartDesign CoreDDR configuration GUI, right-click, and select Generate Design.

When SmartDesign generates the design files, it will install the appropriate testbench files. To run the simulation, set the design root to the CoreDDR instantiation in the Libero IDE design hierarchy pane, and click the Simulation icon in the Libero IDE Design Flow window. This will invoke Model*Sim*® and automatically run the simulation.

Synthesis in Libero IDE

Set the design root appropriately and click the Synthesis icon in the Libero IDE. The Synthesis window appears, displaying the Synplicity® project. Set Synplicity to use the Verilog 2001 standard if Verilog is being used. To perform synthesis, click the **Run** icon.

Place-and-Route in Libero IDE

Having set the design root appropriately and run Synthesis, click the **Layout** icon in Actel Libero IDE to invoke Designer. CoreDDR requires no special place-and-route settings.

CoreDDR

Generics

Customers can define the generics listed in Table 3-1 as required in the source code.

Table 3-1 · CoreDDR Generics

Generic	Default Setting	Description
FPGA family	16	Must be set to match the supported FPGA family. 16: ProASIC3E 17: Fusion
PLL input frequency	133 MHz	The input frequency can be set to the following values: 33 MHz, 66 MHz, and 133 MHz.
Program second PLL with this delay	1.335 ns	This delay is needed to capture the data properly. The data capture is explained in detail in "Implementation Hints" on page 29. The following delay values are supported: 0: Disable 1: 0.735 ns 2: 0.935 ns 3: 1.135 ns 4: 1.335 ns 5: 1.535 ns 6: 1.735 ns 7: 1.935 ns 8: 2.135 ns 9: 2.335 ns 10: 2.535 ns 11: 2.735 ns 12: 2.935 ns 13: 3.135 ns 14: 3.335 ns 15: 3.535 ns
SDRAM_DSIZE	64	Local side data width
SDRAM_CHIPS	8	Number of chip selects
SDRAM_COLBITS	12	Maximum number of SDRAM column bits
SDRAM_ROWBITS	14	Maximum number of SDRAM row bits
SDRAM_CHIPBITS	3	Number of encoded chip select bits
SDRAM_BANKSTATMODULES	4	Number of bank status modules used (in multiples of four)
SDRAM_NIBBLE_DQS	0	For the DDR SDRAM controller, the typical configuration implements one dm and dqs per byte of dq bits. Some memories are arranged with one dm and one dqs per nibble of dq bits. CoreDDR can be configured to support one dm and one dqs per nibble of dq bits.

Controller Configuration Ports

CoreDDR is configured using runtime-programmable configuration ports. These ports may be tied off by the user to fixed values or programmed at runtime. These values should not change after RESET_N is deasserted. Table 3-2 lists the configuration ports.

Table 3-2 · CoreDDR Controller Parameters

Parameter	Port Bits	Valid Values	Description
RAS	4	1–10	SDRAM active to precharge (t_{RAS}), specified in clock cycles
RCD	3	2–5	SDRAM active to read or write delay (t_{RCD}), specified in clock cycles
RRD	2	2–3	SDRAM active bank a to active bank b (t_{RRD}), specified in clock cycles
RP	3	1–4	SDRAM precharge command period (t_{RP}), specified in clock cycles
RC	4	3–12	SDRAM active to active/auto-refresh command period (t_{RC}), specified in clock cycles
RFC	4	2–14	Auto-refresh to active/auto-refresh command period (t_{RFC}) specified in clock cycles
MRD	3	1–7	SDRAM load mode register command to active or refresh command (t_{MRD}), specified in clock cycles
CL	3	1–4	SDRAM CAS latency, specified in clock cycles
CL_HALF	1	0–1	DDR SDRAM half clock latency. Specifies half clock of latency in addition to CL.
BL	2	0–3	SDRAM maximum burst length (encoded). This value refers to the number of local side transfers per burst (one local side transfer results in two transfers to the DDR device). Values are decoded as follows: 0 – 1 transfer/burst 1 – 2 transfers/burst 2 – 4 transfers/burst 3 – 8 transfers/burst (valid for SDR SDRAM devices only)
DS	2	0, 1, 3	DDR drive strength setting programmed into Extended Mode Register (EMR) bits 6 and 1. Values mapped to EMR as follows (refer to DDR SDRAM device datasheet for description of drive strength settings): 0 – EMR[6] = 0, EMR[1] = 0 1 – EMR[6] = 0, EMR[1] = 1 3 – EMR[6] = 1, EMR[1] = 1
WR	2	1–3	SDRAM write recovery time (t_{WR})
DELAY	16	10–65,535	Controller's delay after a reset event before initializing the SDRAM, specified in clock cycles. Per JEDEC standards, DDR devices require this delay to be a minimum of 200 µs.
REF	16	10–65,535	Period between auto-refresh commands issued by the controller, specified in clock cycles REF = auto-refresh interval/t_{CK}
COLBITS	3	3–7	Number of bits in the column address (encoded). Values are decoded as follows: 3 – 8 column bits 4 – 9 column bits 5 – 10 column bits 6 – 11 column bits 7 – 12 column bits

Table 3-2 · CoreDDR Controller Parameters (continued)

Parameter	Port Bits	Valid Values	Description
ROWBITS	2	0–3	Number of bits in the row address (encoded). Values are decoded as follows: 0 – 11 row bits 1 – 12 row bits 2 – 13 row bits 3 – 14 row bits
REGDIMM	1	0–1	Set when using registered/buffered DIMM. Causes adjustment in local bus interface timing to synchronize with SDRAM command timing, delayed by register/buffer on DIMM.

For example:

If RCD is 20 ns in the specification and the clock is 133 MHz, then the number of clocks is equal to 20 ÷ 7.5 = 3. You must hardcode the RCD ports to the binary value “011” in the top level, since the value is "3". Example settings for the timing-related parameters are shown in Table 3-3. These settings are based on the speed grade of the SDRAM devices and the desired operating frequency. Consult the datasheet for the SDRAM device you are using for the specific timing values of that device.

Table 3-3 · Example Controller Parameter Values for CoreDDR

Parameter	133 MHz (7.5 ns period)[1]		167 MHz (6 ns period)[2]		200 MHz (5 ns period)[3]	
	Specification	Value	Specification	Value	Specification	Value
RAS	40.0 ns	6	42.0 ns	7	40.0 ns	8
RCD	20.0 ns	3	18.0 ns	3	15.0 ns	3
RRD	15.0 ns	2	12.0 ns	2	10.0 ns	2
RP	20.0 ns	3	18.0 ns	3	15.0 ns	3
RC	65.0 ns	9	60.0 ns	10	55.0 ns	11
RFC	75.0 ns	10	72.0 ns	12	70.0 ns	14
MRD	15.0 ns	2	12.0 ns	2	10.0 ns	2
CL	–	2	–	2	–	3
CL_HALF	–	1	–	1	–	0
WR	15.0 ns	2	15.0 ns	3	15.0 ns	2
DELAY	200 µs	26,667	200 µs	33,334	200 µs	40,000
REF	7.8125 µs	1,041	7.8125 µs	1,302	7.8125 µs	1,562

Notes:

1. *Values based on Micron MT46V64M8-75*
2. *Values based on Micron MT46V64m8-6T*
3. *Values based on Micron MT46V64M8-5B*

4

Core Interfaces

The port signals for CoreDDR are defined in Table 4-1, Table 4-2 on page 20, and Table 3-2 on page 16, and illustrated in Figure 1-3 on page 9. All signals are designated either Input (input only) or Output (output only).

Local Bus Signals

The user interface to CoreDDR is referred to as the local bus interface. The local bus signals are shown in Table 4-1.

Table 4-1 · Local Bus Signals

Signal	Name	I/O	Description
CLK_IN	Clock	Input	System input clock. All local interface signals are synchronous to this clock.
CLK_1X	Clock	Output	1× system clock.
CLK_1X_SH90	Clock	Output	A 90° phase-shifted version of system clock
RESET_N	Reset	Input	System reset
RADDR[30:0]	Memory Address	Input	Local interface address
B_SIZE[3:0]	Burst Size	Input	Local interface burst length. Valid values are 1 through BL, where BL is the programmed burst length. (Refer to Table 3-2 on page 16 for a discussion of the BL parameter.)
R_REQ	Read Request	Input	Local interface read request
W_REQ	Write Request	Input	Local interface write request
AUTO_PCH	Auto-Precharge Request	Input	When asserted in conjunction with R_REQ or W_REQ, causes command to be issued as *read with auto-precharge* or *write with auto-precharge*, respectively.
RW_ACK	Read/Write Acknowledge	Output	Acknowledgment of read or write request
D_REQ	Data Request	Output	Requests data on the local interface write data bus (datain) during a write transaction. Asserts one clock cycle prior to when data is required.
W_VALID	Write Data Valid	Output	Frames the active data being written to SDRAM. Mimics D_REQ, except it is delayed by one clock cycle. This signal is typically not used and is retained for legacy compatibility.
R_VALID	Read Data Valid	Output	Indicates that the data on the local interface read data bus (dataout) is valid during a read cycle.
SD_INIT	Initialization Strobe	Input	Causes the SDRAM controller to reissue the initialization sequence to SDRAM devices. The SDRAM controller will always issue the initialization sequence (including the startup delay) after reset, regardless of the SD_INIT state. This signal can be tied Low if runtime reinitialization is not required.
DATAIN[2N–1:0]	Input Data	Input	Input data bus. This data bus is twice the width of the DDR SDRAM data bus.
DATAOUT[2N–1:0]	Output Data	Output	Output data bus. This data bus is twice the width of the DDR SDRAM data bus.
DM_IN[2N/16-1:0]	Data Mask	Input	Masks individual bytes during data write.

Notes:

1. *The* N *value in vector indexing refers to data width to SDRAM interface.*
2. *All control signals are active high except RESET_N.*
3. *All local interface signals are synchronous to CLK_1X.*

DDR SDRAM Interface Signals

The external interface to SDRAM devices is referred to as the SDRAM interface. The SDRAM interface signals are shown in Table 4-2.

Table 4-2 · DDR SDRAM Interface Signals

Signal	Name	I/O	Description
SA[13:0]	Address Bus	Output	Sampled during the *active*, *precharge*, *read*, and *write* commands. This bus also provides the mode register value during the *load mode register* command.
BA[1:0]	Bank Address	Output	Sampled during *active*, *precharge*, *read*, and *write* commands to determine which bank command is to be applied to.
CS_N[7:0]	Chip Selects	Output	SDRAM chip selects
CKE	Clock Enable	Output	SDRAM clock enable. Held Low during reset to ensure SDRAM DQ and DQS outputs are in the high-impedance state.
RAS_N	Row Address Strobe	Output	Command input
CAS_N	Column Address Strobe	Output	Command input
WE_N	Write Enable	Output	Command input
DQ_OUT[N–1:0]	Data Bus Output	Output	SDRAM data bus output to external tristate buffers. This data bus is half the width of the DATAIN and DATAOUT busses.
DQIN[N–1:0]	Data Bus Input	Input	SDRAM data bus input. This data bus is half the width of the DATAIN and DATAOUT busses.
DQ_OE[N/8–1:0]	Data Bus Output Enable	Output	SDRAM data bus output enable. Used to control external tristate buffers to DDR SDRAM.
DM[N/8–1:0]	Data Mask	Output	Masks individual bytes during data write. Multiplexed version of the DM_IN input to the controller.
DQS_OUT[N/8–1:0]	Data Strobe Output	Output	Strobes data into the SDRAM devices during writes.
DQS_OE[N/8–1:0]	Data Strobe Output Enable	Output	Used to control external tristate buffers for data strobe.

Notes:

1. *The* n *value in vector indexing refers to data width to SDRAM interface.*
2. *All control signals are active high except RESET_N.*
3. *All local interface signals are synchronous to CLK_1X.*

5

Timing Diagrams

SDRAM Writes

The user requests writes to the local bus interface by asserting the W_REQ signal and driving the starting address and burst size on RADDR and B_SIZE, respectively. The AUTO_PCH signal can also be asserted with W_REQ to cause the write to be issued as a *write with auto-precharge*.

The rules for write requests to the local bus interface are as follows:

1. Once W_REQ is asserted, it must remain asserted until RW_ACK is asserted by CoreDDR. After RW_ACK is asserted by CoreDDR, W_REQ may remain asserted to request a follow-on write transaction. W_REQ may remain asserted over any number of RW_ACK pulses to generate any number of cascaded write bursts. The only time W_REQ may be deasserted is during the clock cycle immediately following the RW_ACK pulse from CoreDDR.
2. RADDR, B_SIZE, and AUTO_PCH must maintain static values from the point when W_REQ becomes asserted until CoreDDR asserts RW_ACK. RADDR, B_SIZE, and AUTO_PCH may only change values in the clock cycle immediately following the RW_ACK pulse from CoreDDR, or when W_REQ is deasserted.
3. The W_REQ signal may not be asserted while the read request (R_REQ) signal is asserted.
4. The data request (D_REQ) signal will assert one clock cycle prior to when the user must present data on the DATAIN bus.
5. The timing relationship between an initial W_REQ and RW_ACK assertion, or between RW_ACK pulses as a result of multiple cascaded writes, will vary depending on the status of the banks being accessed, configuration port settings, refresh status, and initialization status. The user logic should not rely on any fixed timing relationship between W_REQ and RW_ACK.

Example CoreDDR Write Sequence

Figure 5-1 on page 22 shows an example DDR SDRAM controller write sequence. In this sequence, three writes are requested, all to the same row. The first and second write requests are for a burst size of four, and the third is for a burst size of two. The write request (W_REQ) signal is first asserted along with the starting address (RADDR) and the burst size (B_SIZE). As a result of this request, the DDR SDRAM controller asserts the row address (SA), bank address (BA), and chip select (CS_N) with the *active* command to open the bank to the requested row. Next, the DDR SDRAM controller requests data from the user through the local interface using the D_REQ signal, and acknowledges the write request by asserting RW_ACK. The DDR SDRAM controller begins writing the data to the SDRAM devices by issuing the *write* command and presenting the DDR data and the DQS clock. For this request, four data transfers occur to the local interface, which translates to eight data transfers to the SDRAM interface.

The local interface write request (W_REQ) remains asserted after RW_ACK asserted by the DDR SDRAM controller to request an additional write burst. After the rw_ack pulse, the local interface changes the address (RADDR) but keeps the burst size (B_SIZE) at four. The DDR SDRAM controller issues the next *write* command, and data transfers to the SDRAM interface immediately following the first burst.

After the second write request, the DDR SDRAM controller continues to assert W_REQ to cause a third burst. For this write, the burst size (B_SIZE) is reduced to two. Since the burst size of the second write request is less than the programmed DDR SDRAM burst length, the DDR SDRAM controller must prevent the last data transfers from being written to memory. DDR SDRAM devices do not allow write bursts to be terminated, so the DDR SDRAM controller handles this by masking the last four data transfers (using dm) to prevent these from being written to the SDRAM.

In the case of this write burst sequence, all the writes are to the same bank and row. If the second or third writes are to different banks or rows, the SDRAM controller issues *precharge* and/or *active* commands prior to the *write* commands.

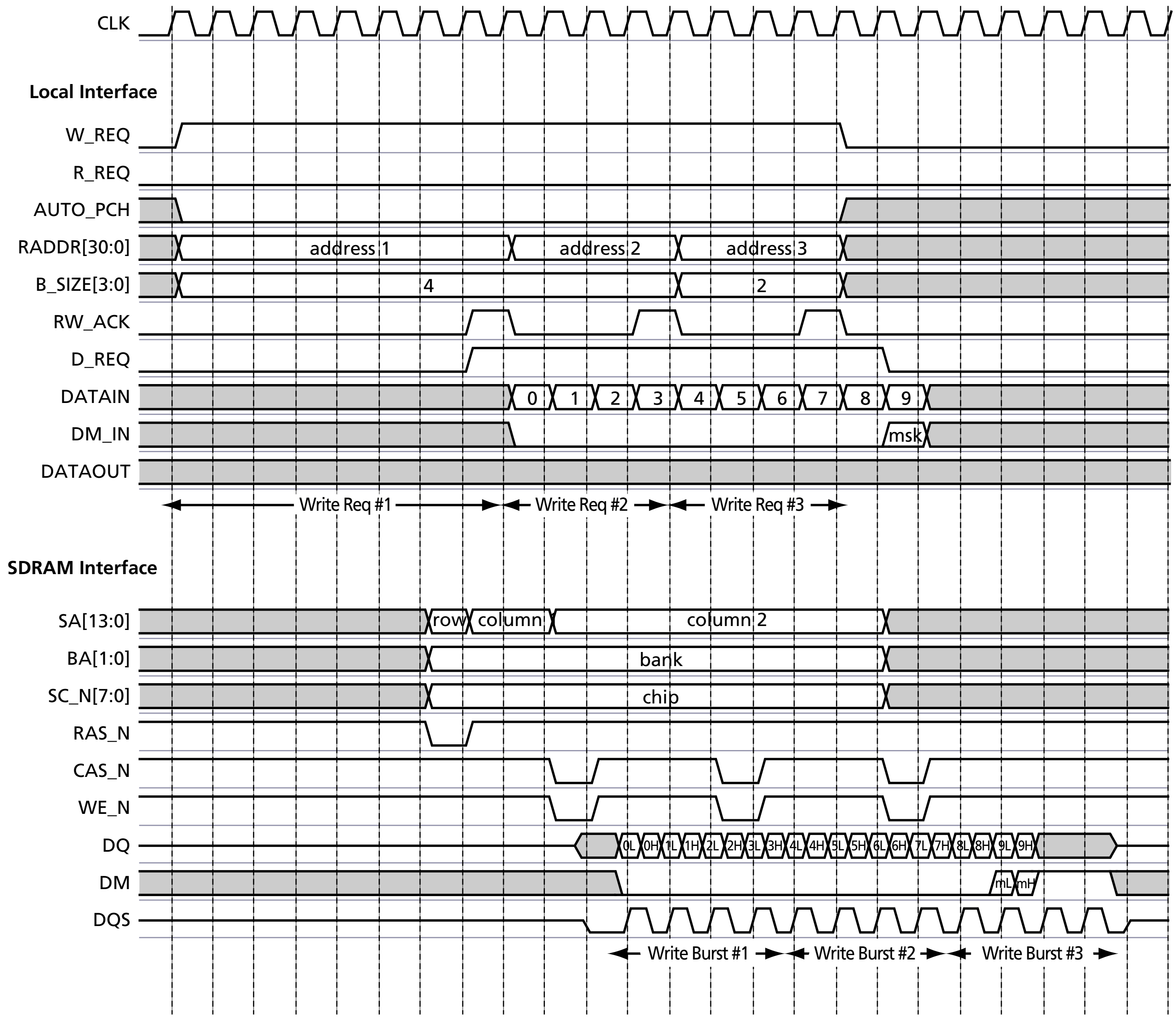

CLK
Local Interface
W_REQ
R_REQ
AUTO_PCH
RADDR[30:0]
address 1
address 2
address 3
B_SIZE[3:0]
4
2
RW_ACK
D_REQ
DATAIN
0
1
2
3
4
5
6
7
8
9
DM_IN
msk
DATAOUT
Write Req #1
Write Req #2
Write Req #3
SDRAM Interface
SA[13:0]
row
column
column 2
BA[1:0]
bank
SC_N[7:0]
chip
RAS_N
CAS_N
WE_N
DQ
0L
0H
1L
1H
2L
2H
3L
3H
4L
4H
5L
5H
6L
6H
7L
7H
8L
8H
9L
9H
DM
mL
mH
DQS
Write Burst #1
Write Burst #2
Write Burst #3

Figure 5-1 · CoreDDR Burst Write

SDRAM Reads

The user requests reads from the local bus interface by asserting the R_REQ signal and driving the starting address and burst size on RADDR and B_SIZE, respectively. The AUTO_PCH signal can also be asserted with R_REQ to cause the read to be issued as a *read with auto-precharge*.

The rules for read requests at the local bus interface are as follows:

1. Once R-REQ is asserted, it must remain asserted until RW_ACK is asserted by CoreDDR. After RW_ACK is asserted by CoreDDR, R_REQ may remain asserted to request a follow-on read transaction. R_REQ may remain asserted over any number of RW_ACK pulses to generate any number of cascaded read bursts. The only time R_REQ may be deasserted is during the clock cycle immediately following the RW_ACK pulse from CoreDDR.
2. RADDR, B_SIZE, and AUTO_PCH must maintain static values from the point when R_REQ becomes asserted until CoreDDR asserts RW_ACK. RADDR, B_SIZE, and AUTO_PCH may only change values in the clock cycle immediately following the RW_ACK pulse from CoreDDR, or when R-REQ is deasserted.
3. The R-REQ signal may not be asserted while the write request (W_REQ) signal is asserted.
4. The read data valid (R_VALID) signal will assert when valid data is available on the DATAOUT bus.

The timing relationship between an initial R_REQ and RW_ACK assertion, or between RW_ACK pulses as a result of multiple cascaded reads, will vary depending on the status of the banks being accessed, configuration port settings, refresh status and initialization status. The user logic should not rely on any fixed timing relationship between R_REQ and RW_ACK.

Example CoreDDR Read Sequence

Figure 5-2 on page 24 shows an example DDR SDRAM controller read sequence. In this sequence, three reads are requested, all from the same row. The first and second read requests are for a burst size of four, and the third is for a burst size of two. The read request (R_REQ) signal is first asserted with the starting address (RADDR) and the burst size (B_SIZE). As a result of this request, the DDR SDRAM controller asserts the row address (SA), bank address (BA), and chip select (CS_N) with the *active* command to open the bank to the requested row. Next, the DDR SDRAM controller acknowledges the read request by asserting RW_ACK, then issues the *read* command to the SDRAM devices. Since the CAS latency is set at 2.5 in this case, read data appears at the SDRAM interface 2.5 clock cycles after the *read* command is issued. The DDR SDRAM controller demultiplexes the data at the SDRAM interface and presents it to the local interface dataout bus. The DDR SDRAM controller asserts the R_VALID signal to indicate valid data to the DATAOUT bus. For this request, eight data transfers occur at the SDRAM interface, which translates to four data transfers to the local interface.

The local interface read request (R_REQ) remains asserted after RW_ACK is asserted by the DDR SDRAM controller to request an additional read burst. After the RW_ACK pulse, the local interface changes the address (RADDR) but keeps the burst size (B_SIZE) at four. The DDR SDRAM controller issues the next *read* command, and data transfers at the SDRAM interface immediately following the first burst.

After the second read request, the DDR SDRAM controller continues to assert R_REQ to cause a third burst. For this write, the burst size (B_SIZE) is reduced to two. Since the burst size of the second read request is less than the programmed DDR SDRAM burst length, the burst must be terminated using the *burst terminate* command. The DDR SDRAM controller does this automatically, asserting CAS_N two clock cycles after the *read* command was issued. This causes the SDRAM devices to discontinue the read after the first four data transfers to the SDRAM interface. The SDRAM devices also stop driving the DQS clock signal.

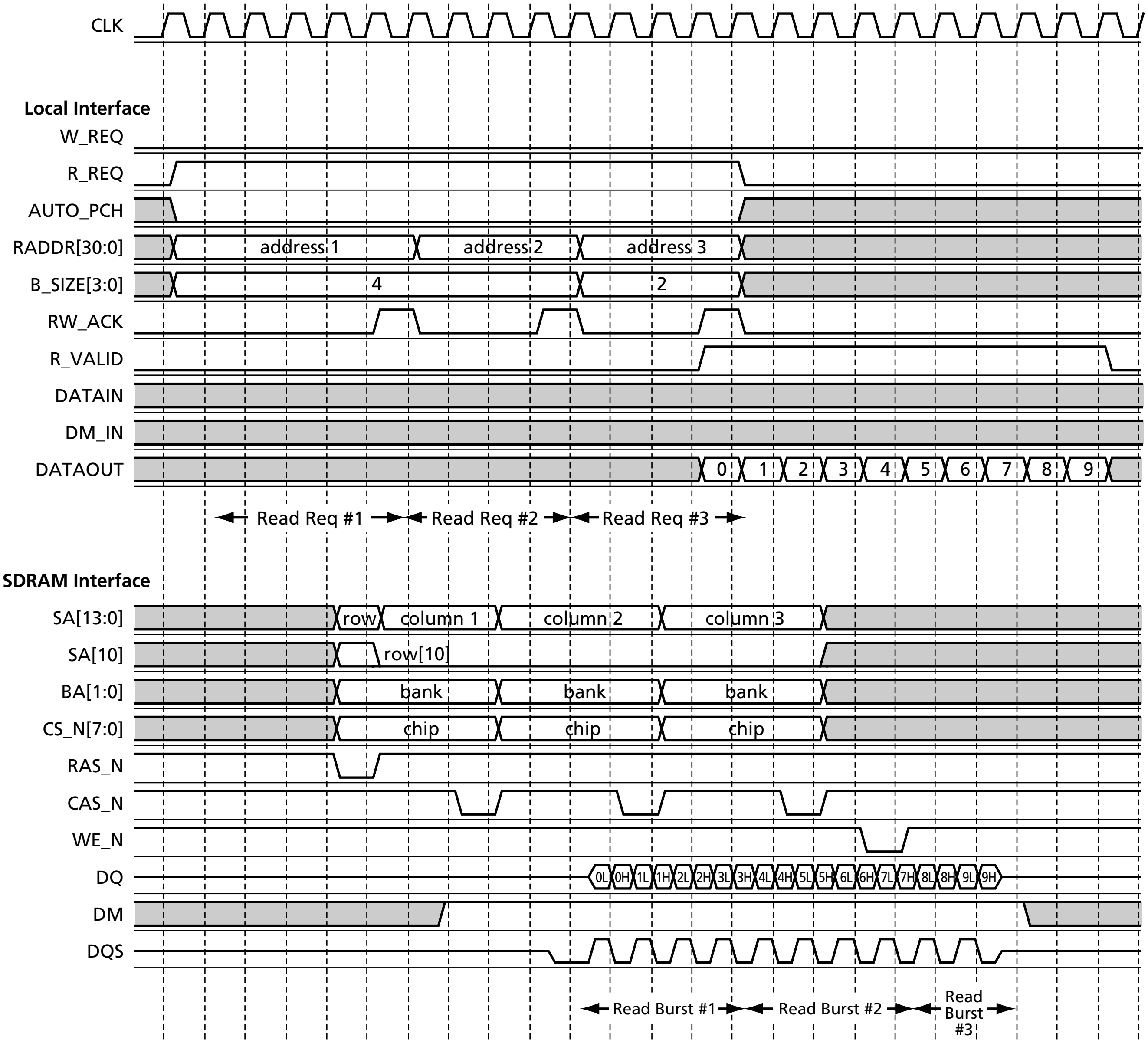

CLK
Local Interface
W_REQ
R_REQ
AUTO_PCH
RADDR[30:0]
address 1
address 2
address 3
B_SIZE[3:0]
4
2
RW_ACK
R_VALID
DATAIN
DM_IN
DATAOUT
0
1
2
3
4
5
6
7
8
9
Read Req #1
Read Req #2
Read Req #3
SDRAM Interface
SA[13:0]
row
column 1
column 2
column 3
SA[10]
row[10]
BA[1:0]
bank
bank
bank
CS_N[7:0]
chip
chip
chip
RAS_N
CAS_N
WE_N
DQ
0L
0H
1L
1H
2L
2H
3L
3H
4L
4H
5L
5H
6L
6H
7L
7H
8L
8H
9L
9H
DM
DQS
Read Burst #1
Read Burst #2
Read Burst #3

Figure 5-2 · CoreDDR Burst Read

Auto-Precharge

Read commands can be issued to the SDRAM devices as *read with auto-precharge* or *read without auto-precharge*. Likewise, *write* commands can be issued to the SDRAM devices as *write with auto-precharge* or *write without auto-precharge*. If the auto-precharge option is used, the SDRAM device will automatically close (precharge) banks being read from or written to at the end of the transaction. Any subsequent reads or writes to this bank will not require an explicit *precharge* command from CoreDDR.

The user selects whether *read* or *write* commands are issued with auto-precharge through the AUTO_PCH signal. If AUTO_PCH is asserted along with W_REQ or R_REQ, the command will be issued to the SDRAM with auto-precharge. The auto-precharge option is useful in situations where the requested read or write addresses tend to be random. With random address sequences, banks are seldom left open with the exact row required by a subsequent request. If auto-precharge is not used for the previous access to a bank, subsequent transactions to that bank first require the bank to be closed (precharged), causing a delay in the transaction. If auto-precharge is used for the previous access, the bank is already closed and ready to be opened to the desired row.

Figure 5-3 on page 26 shows example transactions using the auto-precharge feature of CoreDDR. In the example, two write requests are issued, the first using auto-precharge and the second not using auto-precharge. Both requests are to the same bank but may be to different rows. CoreDDR issues the first command as a *write with auto-precharge* by driving bit SA[10] High during the *write* command. CoreDDR issues the second command as a *write without auto-precharge* by driving bit SA[10] Low during the *write* command. Since the first and second requests are to the same bank, CoreDDR must wait before reopening the bank to meet the SDRAM t_{WR} and t_{RP} requirements. Had the first and second requests been to different banks, the second request would have followed the first request with no interruption to data flow.

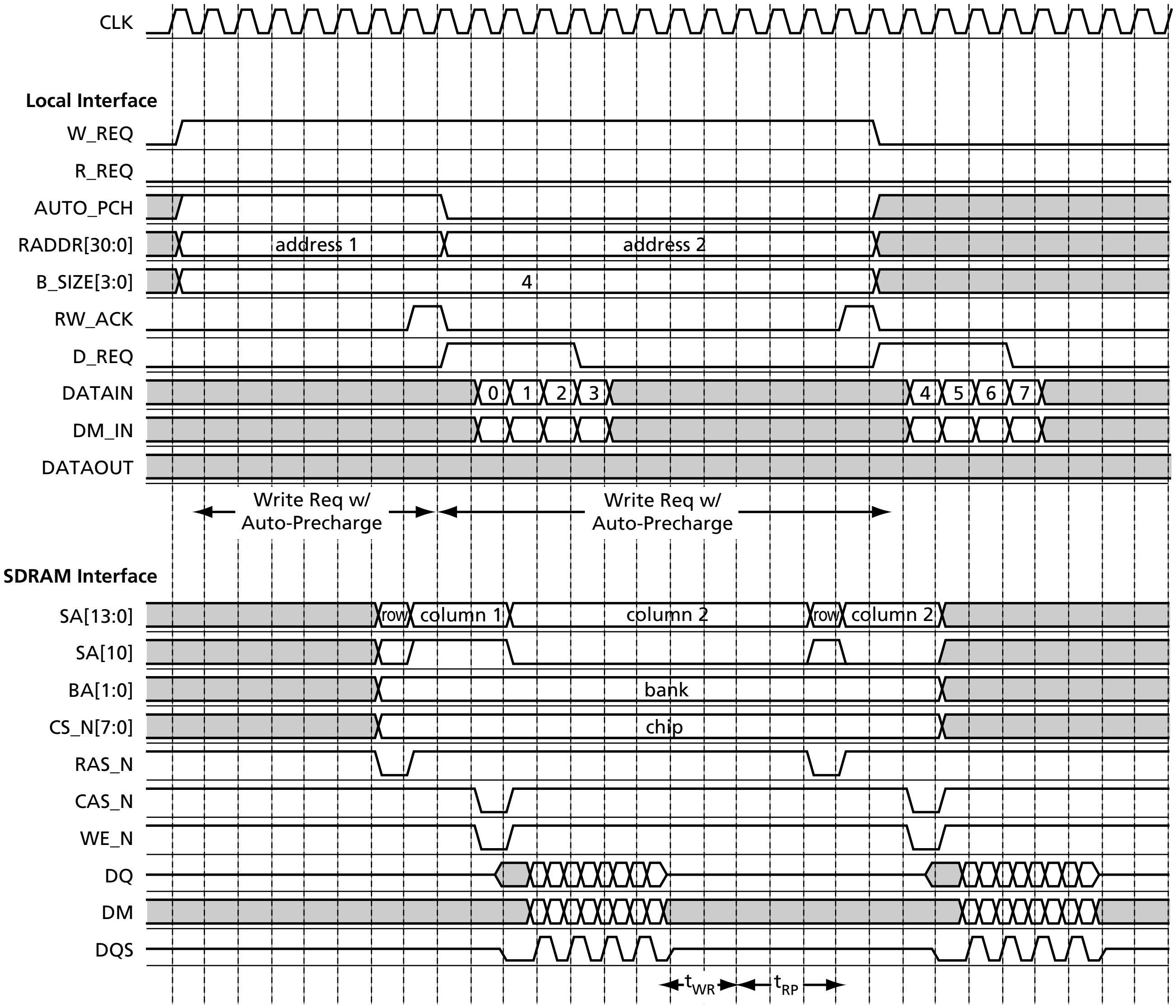

CLK
Local Interface
W_REQ
R_REQ
AUTO_PCH
RADDR[30:0]
address 1
address 2
B_SIZE[3:0]
4
RW_ACK
D_REQ
DATAIN
0
1
2
3
4
5
6
7
DM_IN
DATAOUT
Write Req w/ Auto-Precharge
Write Req w/ Auto-Precharge
SDRAM Interface
SA[13:0]
row
column 1
column 2
row
column 2
SA[10]
BA[1:0]
bank
CS_N[7:0]
chip
RAS_N
CAS_N
WE_N
DQ
DM
DQS
t_{WR}
t_{RP}

Figure 5-3 · CoreDDR Burst Writes with Auto-Precharge Option

Testbench Operation

Two testbenches are provided with CoreDDR:

- Verilog testbench
- VHDL testbench

Testbench Description

Included with the Obfuscated and RTL releases of CoreDDR is a user testbench that gives an example of CoreDDR usage. A simplified block diagram of the testbench is shown in Figure 6-1. By default, the Verilog version, tb_user.v, instantiates a Micron SDRAM model (mt46v8m16.*v*). The VHDL version, tb_user.vhd, instantiates a Micron SDRAM model (mt46v8m16.vhd). The testbench instantiates the DUT (design under test), which is the CoreDDR macro, the SDRAM model, as well as the test vector modules that provide stimuli sources for the DUT. A procedural testbench controls each module and applies the sequential stimuli to the DUT.

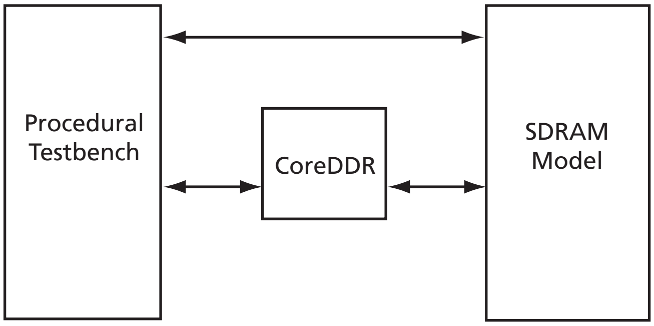


Figure 6-1 · CoreDDR Testbench

Verilog User Testbench

The Verilog user testbench is provided as a reference and can be modified to suit your needs. The source code for the Verilog user testbench is provided to ease the process of integrating the CoreDDR macro into your design and verifying its functionality.

VHDL User Testbench

The VHDL user testbench is provided as a reference and can be modified to suit your needs. The source code for the VHDL testbench is provided to ease the process of integrating the CoreDDR macro into your design and verifying its functionality.

Implementation Hints

Data Capture

There are two ways of achieving data capture with DDR: self-timed and by use of the Distributed Queueing System (DQS) strobes. Due to lack of multiple DLL elements in ProASIC3E technology and the relatively low speeds (<200 MHz), Actel recommends the self-timed approach.

Handling the Data Capture from DIMM

Self-timed mode requires the controller to generate a data sample clock that captures the incoming data (Figure 7-1). Ideally this is 90° phase-shifted from the main clock; in reality, allowance has to be made for propagation delays. The DIMM (DDRRAM) generates data on both the positive and negative clock edges. The secondary clock of the PLL is ideally a 90° phase-shifted clock and needs to sample the data correctly. When capturing data, the 90° phase-shifted clock should sample the data clock to the mid point of the data window.

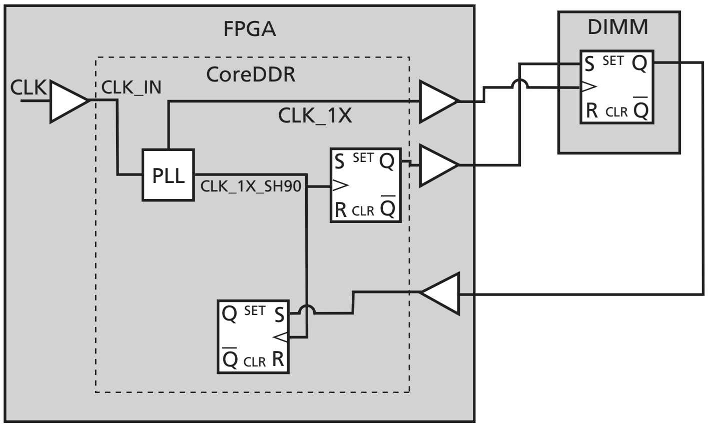


Figure 7-1 · Self-Timed Mode

The DIMM (DDRRAM) generates data on both the positive and negative clock edges. The FPGA will ideally clock on the 90° phase-shifted clock and correctly sample the data. Due to the propagation delays, however, the data is not captured correctly. The 90° phase-shifted clock is replaced with the CLKDLY cell to adjust the delay values and capture the data properly, as shown in Figure 7-3 on page 30.

The best and worst case signal propagation delays are calculated in "Delay Calculation" on page 30. In the ProASIC3E-2 part, the data round trip delay from the internal clock to the data arriving at the sample register at worst case conditions is 4.85 ns. The best case conditions is 2.5 ns (Figure 7-2).

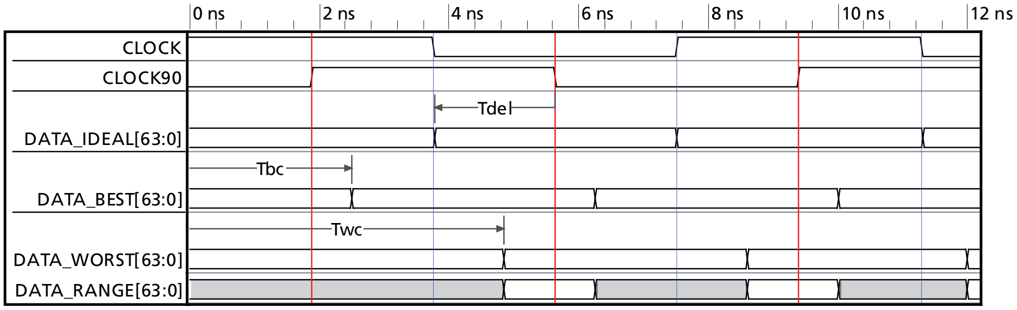


Figure 7-2 · Signal Propagation Delays

Note: In Figure 7-2 on page 29 we assume that the secondary clock, which has a phase shift of 90°, is ideal. The sample points (rising edge and falling edge) are in red. The DATA_BEST shows the fastest that the data can change (2.5 ns round trip) and DATA_WORST the slowest (4.85 ns round trip). The DATA_RANGE indicates when the data should be valid at the capture register. What we want to achieve is that the sample clock hits the midpoint of the data window. EQ 7-1 is the equation for calculating the clock delay (for the CLKDLY cell in the ProASIC3E FPGA cell library) to achieve this midpoint sample.

Definitions

Tbc = Best case data round trip delay

Twc = Worst case round trip delay

Tclk = System clock period

Tdel = Delay needed to shift the capture clock

$$\text{DLL Delay} = (Tclk/2 + Tbc + Twc)/2 - Tclk/2 \tag{EQ 7-1}$$

$$= (Tclk/2 + Tbc + Twc - Tclk)/2$$

$$= (Tbc + Twc - Tclk/2)/2$$

Using typical figures obtained with an A3PE600FG484-2 part and a socketed, mounted 1 GB DIMM module we get

$$= (2.5 + 4.85 - 3.75)/2$$

$$Tdel = 1.8\text{ ns (equivalent to } 86° \text{ shift from the system clock)}$$

From EQ 7-1, the Clock Delay macro should be generated with a delay of 1.8 ns to capture the data correctly, as shown in Figure 7-3.

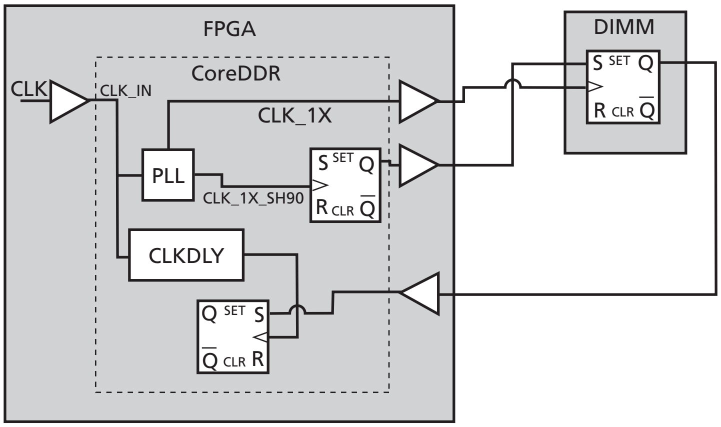


Figure 7-3 · Data Capture

Delay Calculation

Worst Case Delay Calculation

1. CLK output PAD = 1.6 ns
2. PCB Delay to DIMM = 0.61 ns
3. CLKK_OUT on DIMM = 0.75 ns
4. PCB Delay to FPGA = 0.6 1 ns

5. Input PAD Delay to DDR Reg = 0.996 ns
6. Setup at DDR Reg = 0.282 ns

Adding the numbers from 1 through 6 gives a worst case delay of 4.85 ns.

Best Case Delay Calculation

1. CLK to output PAD = 0.86 ns
2. PCB Delay to DIMM = 0.43 ns
3. CLK_OUT on DIMM = 0 ns
4. PCB Delay to FPGA = 0.43 ns
5. Input PAD Delay to DDR Reg = 0.6 ns
6. Setup at DDR Reg = 0.203 ns

Adding the numbers from 1 through 6 gives a best case delay of 2.5 ns.

Ordering Information

Ordering Codes

CoreDDR can be ordered through your local Actel sales representative. It should be ordered using the following number scheme: CoreDDR-XX, where XX is listed in Table 8-1.

Table 8-1 · Ordering Codes

XX	**Description**
OM	RTL for Obfuscated RTL – multiple-use license
RM	RTL for RTL source – multiple-use license

Note: CoreDDR-OM is included free with a Libero IDE license.

List of Document Changes

The following table lists critical changes that were made in the current version of the document.

Previous Version	Changes in Current Version ( )	Page
v2.2	The signal names were all made upper case.	N/A
	The buffers were removed from Figure 1-1 · Typical CoreDDR System. They are depicted more completely in Figure 1-2 · DDR SDRAM Controller Block Diagram.	7
	The "Address Mapping" section was revised to state that column bits, bank bits, row bits, and chip select are mapped from the least significant bits of RADDR.	9
	Figure 2-1 · CoreDDR Configuration Within SmartDesign was replaced.	13
	Table 3-1 · CoreDDR Generics was revised to remove the parameters SDRAM_BANKS, SDRAM_OE_BITS, and SDRAM_BANKBITS, and SDRAM_RASIZE. The default setting for "Program second PLL with this delay" was changed to 1.335 ns.	15
	The text preceding Table 3-3 · Example Controller Parameter Values for CoreDDR was revised.	17
	The description for CLK_1X_SH90 in Table 4-1 · Local Bus Signals was revised to change –90º to 90°.	19
	The "SDRAM Reads" section was revised to correct errors in points 1 and 3 so that "read" is used with the R_REQ signal and "write" with the W_REQ signal.	23
v2.1	The core version was changed from v3.0 to v4.0	N/A
	The "Key Features" section is new.	5
	Fusion was added to the "Supported Device Families" section.	5
	Table 1 · CoreDDR Device Utilization and Performance was expanded to include Fusion.	5
	The "Tool Flows" section was revised. The Evaluation version is no longer available. CoreConsole was replaced by SmartDesign.	13
	Table 3-1 · CoreDDR Generics was revised to add Fusion as an available setting for the FPGA Family generic.	15
	"Testbench Description" section was revised.	27
	The "Verilog Testbench" section was replaced by the "Verilog User Testbench" section. The "VHDL Testbench" section was replaced by the "VHDL User Testbench" section.	27
	The "Ordering Information" chapter is new.	33
v2.0	The "Supported Device Families" section was added.	5

Product Support

Actel backs its products with various support services including Customer Service, a Customer Technical Support Center, a web site, an FTP site, electronic mail, and worldwide sales offices. This appendix contains information about contacting Actel and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From Northeast and North Central U.S.A., call **650.318.4480**
From Southeast and Southwest U.S.A., call **650. 318.4480**
From South Central U.S.A., call **650.318.4434**
From Northwest U.S.A., call **650.318.4434**
From Canada, call **650.318.4480**
From Europe, call **650.318.4252** or **+44 (0) 1276 401 500**
From Japan, call **650.318.4743**
From the rest of the world, call **650.318.4743**
Fax, from anywhere in the world **650.318.8044**

Actel Customer Technical Support Center

Actel staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions. The Customer Technical Support Center spends a great deal of time creating application notes and answers to FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Actel Technical Support

Visit the Actel Customer Support website (www.actel.com/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the Actel web site.

Website

You can browse a variety of technical and non-technical information on Actel's home page, at www.actel.com.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. Several ways of contacting the Center follow:

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is tech@actel.com.

Phone

Our Technical Support Center answers all calls. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. The Technical Support numbers are:

650.318.4460
800.262.1060

Customers needing assistance outside the US time zones can either contact technical support via email (tech@actel.com) or contact a local sales office. Sales office listings can be found at www.actel.com/company/contact/default.aspx.

Index

A

Actel
  electronic mail 37
  telephone 38
  web-based technical support 37
  website 37
address mapping 9
auto-precharge 25
auto-refresh 10, 12

B

bank management 5, 12
block diagram 7
bus commands 9

C

CAS latency 10
contacting Actel
  customer service 37
  electronic mail 37
  telephone 38
  web-based technical support 37
controller configuration ports 16
core versions 5
customer service 37

D

DDR SDRAM interface signals 20
device utilization and performance 5
DQS 29

G

general description 5
generics 15

I

initialization sequence 11
instruction timing 11

K

key blocks 7

L

Libero IDE
  importing into 14
  place-and-route 14
  simulation 14
  synthesis 14
licenses 13
local bus signals 19

O

Obfuscated version 13
operation 9
ordering code 33
ordering information 33

P

place-and-route in Libero IDE 14
product support 37–38
  customer service 37
  electronic mail 37
  technical support 37
  telephone 38
  website 37

R

refresh
  operations 10
  timing 12
RTL 13

S

SDRAM
  device configurations 10
  reads 23
  writes 21
self-timed mode 29
signal propagation delays 29
simulation flows 14
SmartDesign 13
synthesis in Libero IDE 14
system blocks 7

T

technical support 37
testbenches
  description 27
  operation 33
  Verilog tests 27
  Verilog verification 27
  VHDL tests 27
  VHDL user 27

typical system 7

V
Verilog
  testbench tests 27
  verification testbench 27

VHDL
  testbench tests 27
  user testbench 27

W
web-based technical support 37

Actel is the leader in low-power and mixed-signal FPGAs and offers the most comprehensive portfolio of system and power management solutions. Power Matters. Learn more at www.actel.com.

Actel Corporation *• 2061 Stierlin Court • Mountain View, CA 94043 • USA*
Phone 650.318.4200 • Fax 650.318.4600 • Customer Service: 650.318.1010 • Customer Applications Center: 800.262.1060
Actel Europe Ltd. *• River Court, Meadows Business Park • Station Approach, Blackwater • Camberley Surrey GU17 9AB • United Kingdom*
Phone +44 (0) 1276 609 300 • Fax +44 (0) 1276 607 540
Actel Japan *• EXOS Ebisu Building 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan*
Phone +81.03.3445.7671 • Fax +81.03.3445.7668 • http://jp.actel.com
Actel Hong Kong *• Room 2107, China Resources Building • 26 Harbour Road • Wanchai • Hong Kong*
Phone +852 2185 6460 • Fax +852 2185 6488 • www.actel.com.cn