PLB Synchronous DRAM (SDRAM) Controller

DS427 (1.12.1) September 18, 2003

Product Overview

Introduction

The Xilinx PLB SDRAM controller provides a SDRAM controller that connects to the PLB bus and provides the control interface for SDRAMs. It is assumed that the reader is familiar with SDRAMs and the IBM PowerPC™.

Features

The Xilinx SDRAM Controller is a soft IP core designed for Xilinx FPGAs and contains the following features:

- PLB interface
- Performs device initialization sequence upon power-up and reset conditions
- Performs auto-refresh cycles
- Supports single-beat and burst transactions
- Supports target-word first cache-line transactions
- Supports cacheline latencies of 2 or 3 set by a design parameter
- Supports various SDRAM data widths set by a design parameter

SDRAM Controller Design Parameters

To allow the user to obtain a SDRAM Controller that is uniquely tailored for their system, certain features are parameterizable in the Xilinx SDRAM Controller design. This allows the user to have a design that only utilizes the resources required by their system and runs at the best possible performance. The features that are parameterizable in the Xilinx SDRAM Controller are shown in Table 1.

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex-II Pro™, Virtex™-II, Virtex™, Virtex™-E, Spartan™-II	
Version of Core	SDRAM	v1.00c
Resources Used. See Table 11		
	Min	Max
Slices	375	620
LUTs	338	671
FFs	416	659
Block RAMs	0	0
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	5.1i or later	
Verification	N/A	
Simulation	ModelSim SE/EE 5.6e or later	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

Table 1: **SDRAM Controller Design Parameters**

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
SDRAM Controller Features	G1	Include support for PLB bursts and cacheline transfers	C_INCLUDE_BURST_CACHELN_SUPPORT	0 = don't include logic to support PLBbursts and cacheline transfers 1 = include logic to support PLBbursts and cacheline transfers	0	integer
	G2	Use positive edge output registers for SDRAM interface signals	C_USE_POSEDGE_OUTREGS	0 = don't use positive edge output registers (SDRAM interface signals clocked on negative edge) 1 = use positive edge output registers (SDRAM interface signals clocked on positive edge)	0	integer
	G3	Include pipeline stage to increase operating frequency (increases latency by 1 clock)[(1)]	C_INCLUDE_HIGHSPEED_PIPE	0 = don't include pipeline stage 1 = include pipeline stage	1	integer

Table 1: **SDRAM Controller Design Parameters *(Continued)***

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
	G4	Target FPGA family	C_FAMILY	spartan2 spartan2e virtex virtexe virtex2 virtex2p	virtex2p	string
SDRAM Device Features	G5	Load Mode Register command cycle time (Tck)	C_SDRAM_TMRD		2	integer
	G6	Write Recovery Time[1] (ps)	C_SDRAM_TWR		15000	integer
	G7	Read/Write command to Read/Write command (Tck)	C_SDRAM_TCCD		1	integer
	G8	Delay after ACTIVE command before PRECHARGE command (ps)	C_SDRAM_TRAS		40000	integer
	G9	Delay after ACTIVE command before another ACTIVE or AUTOREFRESH command (ps)	C_SDRAM_TRC		65000	integer
	G10	Delay after AUTOREFRESH before another command(ps)	C_SDRAM_TRFC		75000	integer
	G11	Delay after ACTIVE command before READ/WRITE command(ps)	C_SDRAM_TRCD		20000	integer
	G12	Delay after ACTIVE command for a row before an ACTIVE command for another row (ps)	C_SDRAM_TRRD		15000	integer
	G13	Delay after a PRECHARGE command (ps)	C_SDRAM_TRP		20000	integer
	G14	Refresh command interval (ms)	C_SDRAM_TREF		64	integer
	G15	Number of Rows in a Refresh Period[3]	C_SDRAM_REFRESH_NUMROWS	2048, 4096, 8192, 16384	8192	integer
	G16	CAS Latency	C_SDRAM_CAS_LAT	2,3	2	integer
	G17	Total data width of devices [4]	C_SDRAM_DWIDTH	8, 16, 32, 64	32	integer

Table 1: **SDRAM Controller Design Parameters *(Continued)***

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
	G18	SDRAM address width	C_SDRAM_AWIDTH	See note(5)	13	integer
	G19	SDRAM column address width	C_SDRAM_COL_AWIDTH	See note(5)	9	integer
	G20	SDRAM bank address width	C_SDRAM_BANK_AWIDTH	See note(5)	2	integer
Address Space	G21	Base Address	C_BASEADDR	Valid address(6)		std_logic_vector
	G22	High Address	C_HIGHADDR	Valid address(6)		std_logic_vector
PLB Bus Interface	G23	PLB Data bus width	C_PLB_DWIDTH	64	64	integer
	G24	PLB Address bus width	C_PLB_AWIDTH	32	32	integer
	G25	Number of PLB bus masters	C_PLB_NUM_MASTERS	1 - 16	4	integer
	G26	PLB clock period (ps)	C_PLB_CLK_PERIOD_PS			integer
Auto-calculated parameters (7)	G27	Average periodic refresh command interval (ps)	C_SDRAM_TREFI	C_SDRAM_TREF/ C_SDRAM_REFRESH_NUMROWS	7800000	integer

Table 1: **SDRAM Controller Design Parameters *(Continued)***

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
	G28	Number of bits required to encode the number of PLB Masters	C_PLB_MID_WIDTH	1 - log2(C_NUM_MASTERS)	2	integer
Simulation only Parameter	G29	SDRAM simulation initialization time in picoseconds	C_SIM_INIT_TIME_PS (8)		10000000	integer

Notes:
1. Set this parameter to 0 if C_USE_POSEDGE_OUTREGS = 1.
2. Manual precharge timing numbers should be used for this parameter if the SDRAM data sheet has different timing numbers for manual and auto precharge.
3. This parameter is used to calculate the refresh command interval and therefore should be set to the number of rows in a refresh period, which is not always the same as the number of rows in the SDRAM device. Check the data sheet carefully for this parameter.
4. Data width of SDRAM devices must be >= 8 and :
    a. = PLB data width OR
    b. = PLB data width/2 OR
    c. = PLB data width/4 OR
    d. = PLB data width/8
5. C_SDRAM_AWIDTH + C_SDRAM_COL_AWIDTH + C_SDRAM_BANK_AWIDTH + log2(C_SDRAM_DWIDTH/8) must be < C_PLB_AWDITH-1
6. The range specified by C_BASEADDR and C_HIGHADDR must comprise a complete, contiguous power of two range such that range = 2^n, and the n least significant bits of C_BASEADDR must be zero.
7. These parameters are automatically calculated by the system generation tool and are not input by the user.
8. Simulation only parameter. This parameter is used to change the SDRAM time for simulation only. Note, the SDRAM requires ~300 nS after this initialization time to complete the initialization sequence. Also note that if this parameter is modified from the default of 100 uS, simulation results will vary from hardware implementation.

Allowable Parameter Combinations

The address range specified by C_BASEADDR and C_HIGHADDR must comprise a complete, contiguous power of two range such that range = 2^n, and the n least significant bits of C_BASEADDR must be zero. The range specified by these parameters must encompass the SDRAM memory space.

SDRAM Controller I/O Signals

Table 2 provides a summary of all Xilinx SDRAM Controller input/output (I/O) signals, the interfaces under which they are grouped, and a brief description of the signal.

Table 2: **SDRAM Controller Pin Descriptions**

Grouping		Signal Name	Interface	I/O	Initial State	Description
SDRAM Signals	P1	SDRAM_Clk	SDRAM	O	0	SDRAM Clock
	P2	SDRAM_CKE	SDRAM	O	0	SDRAM Clock Enable
	P3	SDRAM_CSn	SDRAM	O	1	Active low SDRAM chip select
	P4	SDRAM_RASn	SDRAM	O	1	Active low SDRAM row address strobe

Table 2: **SDRAM Controller Pin Descriptions *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
	P5	SDRAM_CASn	SDRAM	O	1	Active low SDRAM column address strobe
	P6	SDRAM_WEn	SDRAM	O	1	Active low SDRAM write enable
	P7	SDRAM_DQM	SDRAM	O	0	SDRAM data mask
	P8	SDRAM_BankAddr	SDRAM	O	0	SDRAM bank address
	P9	SDRAM_Addr	SDRAM	O	0	SDRAM address
	P10	SDRAM_DQ_o	SDRAM	O	0	Output data to SDRAM
	P11	SDRAM_DQ_i	SDRAM	I		Input data from SDRAM
	P12	SDRAM_DQ_t	SDRAM	O	0	3-state control for SDRAM data buffers
	P13	SDRAM_Clk_in	SDRAM	I		SDRAM clock feedback. If there is no feedback from the SDRAM clock output, connect to PLB_Clk.
PLB Slave Signals (1)	P14	PLB_PAValid	PLB	I		PLB primary address valid indicator
	P15	PLB_buslock	PLB	I		PLB bus lock
	P16	PLB_masterID[0:C_PLB_NUM_MASTERS-1]	PLB	I		PLB current master indicator
	P17	PLB_RNW	PLB	I		PLB read not write
	P18	PLB_BE[0:C_PLB_DWIDTH/8-1]	PLB	I		PLB byte enables
	P19	PLB_size[0:3]	PLB	I		PLB transfer size
	P20	PLB_type[0:2]	PLB	I		PLB transfer type
	P21	PLB_MSize[0:1]	PLB	I		PLB master data bus size
	P22	PLB_compress	PLB	I		PLB compressed data transfer indicator
	P23	PLB_guarded	PLB	I		PLB guarded transfer indicator
	P24	PLB_ordered	PLB	I		PLB synchronize transfer indicator
	P25	PLB_lockErr	PLB	I		PLB lock error indicator
	P26	PLB_abort	PLB	I		PLB abort bus request indicator
	P27	PLB_ABus[0:C_PLB_AWIDTH-1]	PLB	I		PLB address bus
	P28	PLB_SAValid	PLB	I		PLB secondary address valid indicator
	P29	PLB_rdPrim	PLB	I		PLB secondary to primary read request indicator
	P30	PLB_wrPrim	PLB	I		PLB secondary to primary write request indicator
	P31	PLB_wrDBus[0:C_PLB_DWIDTH-1]	PLB	I		PLB write data bus

Table 2: **SDRAM Controller Pin Descriptions *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
	P32	PLB_wrBurst	PLB	I		PLB burst write transfer indicator
	P33	PLB_rdBurst	PLB	I		PLB burst read transfer indicator
	P34	Sl_addrAck	PLB	O	0	Slave address acknowledge
	P35	Sl_wait	PLB	O	0	Slave wait indicator
	P36	Sl_SSize[0:1]	PLB	O	0	Slave data bus size
	P37	Sl_rearbitrate	PLB	O	0	Slave rearbitrate bus indicator
	P38	Sl_MBusy[0:C_PLB_NUM_MASTERS-1]	PLB	O	0	Slave busy indicator
	P38	Sl_MErr[0:C_PLB_NUM_MASTERS-1]	PLB	O	0	Slave error indicator
	P40	Sl_wrDAck	PLB	O	0	Slave write data acknowledge
	P41	Sl_wrComp	PLB	O	0	Slave write transfer complete indicator
	P42	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
	P43	Sl_rdDBus[0:C_PLB_DWIDTH-1]	PLB	O	0	Slave read bus
	P44	Sl_rdWdAddr[0:3]	PLB	O	0	Slave read word address
	P45	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
	P46	Sl_rdComp	PLB	O	0	Slave read transfer complete indicator
	P47	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
	P48	PLB_Clk	PLB	I		PLB clock
	P49	PLB_Rst	PLB	I		PLB reset
Timer Interrupt Signal	P50	SDRAM_Init_Done		O	0	SDRAM initialization has completed

Notes:
1. Please refer to the IBM PLB Bus Architecture Specification for more detailed information on these signals.

Parameter/Port Dependencies

There are no dependencies between the SDRAM design parameters and I/O signals. are shown in Table 3.

Table 3: **Parameter-Port Dependencies**

		Name	Affects	Depends	Relationship Description
Design Parameters	G1	C_INCLUDE_SDRAMCLK_DCM	P13, P14, P15, P16		If C_INCLUDE_SDRAMCLK_DCM = 1, SDRAM_Clk_in is used as the feedback clock to the DCM, DCM_locked indicates the status of the DCM and DCM_Rst is the input that provides the reset to the DCM. If C_INCLUDE_SDRAMCLK_DCM = 0, SDRAM_Clk_in should be connected to PLB_Clk, DCM_locked is set to '1' and has no meaning, and DCM_Rst is an unused input and should be connected to ground.

Connecting to Memory

Memory Data Types and Organization

SDRAM memory can be accessed as: byte (8 bits), halfword (2 bytes), word (4 bytes)or doubleword (8 bytes),depending on the size of the bus to which the processor is attached.From the point of view of thePLBdata is organized as big-endian. The bit and byte labeling for the big-endian data types is shown below in Figure 1.

Byte address	n	n + 1	n + 2	n + 3	n + 4	n + 5	n + 6	n + 7	Double Word
Byte label	0	1	2	3	4	5	6	7	
Byte significance	MSB							LSB	
Bit label	0							63	
Bit significance	MSBit							LSBit	

Byte address	n	n + 1	n + 2	n + 3	Word
Byte label	0	1	2	3	
Byte significance	MSByte			LSByte	
Bit label	0			31	
Bit significance	MSBit			LSByte	

Byte address	n	n + 1	Halfword
Byte label	0	1	
Byte significance	MSByte	LSByte	
Bit label	0	15	
Bit significance	MSBit	LSBit	

Byte address	n	Byte
Byte label	0	
Byte significance	MSByte	
Bit label	0 7	
Bit significance	MSBit LSBit	

DS426_01_083103

Figure 1: **Big-Endian Data Types**

Memory to FPGA Connections

The data and address signals at the memory controller are labeled with big-endian bit labeling (for example, D(0:31), D(0) is the MSB), whereas most memory devices are either endian agnostic (they can be connected either way) or little-endian D(31:0) with D(31) as the MSB.

Caution must be exercised with the connections to the external memory devices to avoid incorrect data and address connections. The following table shows the correct mapping of memory controller pins to memory device pins.

Table 4: **FPGA - Memory device pin mapping**

Description	SDRAM Signal (MSB:LSB)	Memory Device Signal (MSB:LSB)
Data Bus	SDRAM_DQ(0:C_SDRAM_DWIDTH-1)	DQ(C_SDRAM_DWIDTH-1:0)
Bank Address	SDRAM_BankAddr(0:C_SDRAM_BANK_AWIDTH-1)	BA(C_SDRAM_BANK_AWIDTH-1:0)
Address	SDRAM_Addr(0:C_SDRAM_AWIDTH-1)	A(C_SDRAM_AWIDTH-1:0)
Data Mask	SDRAM_DQM(0:C_SDRAM_DWIDTH/8-1)	DQMU, DQML

SDRAM Address Mapping

An address offset is calculated based on the width of the SDRAM data bus and the PLBdata bus. The SDRAM column address is then mapped to the PLB address bus, followed by the row address and bank address.

Since the SDRAM will always be accessed to provide data the width of the PLB bus, the column address starting bit is based on the SDRAM data width offset and the column address ending bit is based on the PLBOPB data width offset. The difference in these offsets are set to zero. This sends the proper column address to the SDRAM.

The PLB address bus bit locations for the SDRAM column, row, and bank addresses are calculated as shown in Table 5 and Table 6.

Table 5: **SDRAM Address offset calculations**

Variable	Equation
SDRAM_ADDR_OFFSET	log2(C_SDRAM_DWIDTH/8)
PLB_ADDR_OFFSET	log2(C_PLB_DWIDTH/8)
COLADDR_STARTBIT	C_PLB_AWIDTH - (C_SDRAM_COL_AWIDTH+SDRAM_ADDR_OFFSET)
COLADDR_ENDBIT	C_PLB_AWIDTH-PLB_ADDR_OFFSET-1
NUM_ZEROADDR_BITS	PLB_ADDR_OFFSET-SDRAM_ADDR_OFFSET
ROWADDR_STARTBIT	COLADDR_STARTBIT - C_SDRAM_AWIDTH
ROWADDR_ENDBIT	ROWADDR_STARTBIT + C_SDRAM_AWIDTH-1
BANKADDR_STARTBIT	ROWADDR_STARTBIT - C_SDRAM_BANK_AWIDTH
BANKADDR_ENDBIT	BANKADDR_STARTBIT + C_SDRAM_BANK_AWIDTH-1

Table 6: **SDRAM - PLBOPB Address Bus Assignments**

SDRAM Address	PLBOPB Address Bus
Column Address	PLB_ABus(COLADDR_STARTBIT to COLADDR_ENDBIT) & ZEROADDR_BITS
Row Address	PLB_ABus(ROWADDR_STARTBIT to ROWADDR_ENDBIT)
Bank Address	PLB_ABus(BANKADDR_STARTBIT to BANKADDR_ENDBIT)

Table 9 and Table 10 show an example of the mapping between the PLB address and the SDRAM address when the data width of the SDRAM is 16 and the data width of the bus is 64, the column address width is 9, the row address width is 13, and the bank address width is 2. Note that since the PLB data width is 64, its address offset is 3 where the SDRAM address offset is 1. Therefore, the column address is PLB address bus bit 22 through bit 28 with two bits set to zero.

Table 7: **PLB Example SDRAM Address offset calculations**

Variable	Value
SDRAM_ADDR_OFFSET	log2(16/8) = 1
PLB_ADDR_OFFSET	log2(64/8) = 3
COLADDR_STARTBIT	32 - (9+1) = 22
COLADDR_ENDBIT	32-3-1= 28
NUM_ZEROADDR_BITS	3-1 = 2
ROWADDR_STARTBIT	22 - 13 = 9
ROWADDR_ENDBIT	9 + 13 -1 = 21
BANKADDR_STARTBIT	9 - 2 = 7
BANKADDR_ENDBIT	7 + 2 -1 = 8

Table 8: **SDRAM - PLB Address Bus Assignments**

SDRAM Address	PLB Address Bus
Column Address	PLB_ABus(22: 28) &"00"
Row Address	PLB_ABus(9:21)
Bank Address	PLB_ABus(7:8)

SDRAM Controller Design

Block Diagram

The Xilinx SDRAM controller consists of the PLB IPIF to provide the bus protocol, 3 state machines to control the SDRAM operation, an I/O module to instantiate the SDRAM I/O registers for the SDRAM data interface, and a clock generation module.

The separation of the Command State Machine and the Data State Machine allows for the application of commands to the SDRAM while data reception/transmission is in progress. Overlapping the SDRAM commands with the data transfer when accessing data in the same row of the same bank allows for more optimal SDRAM operation.

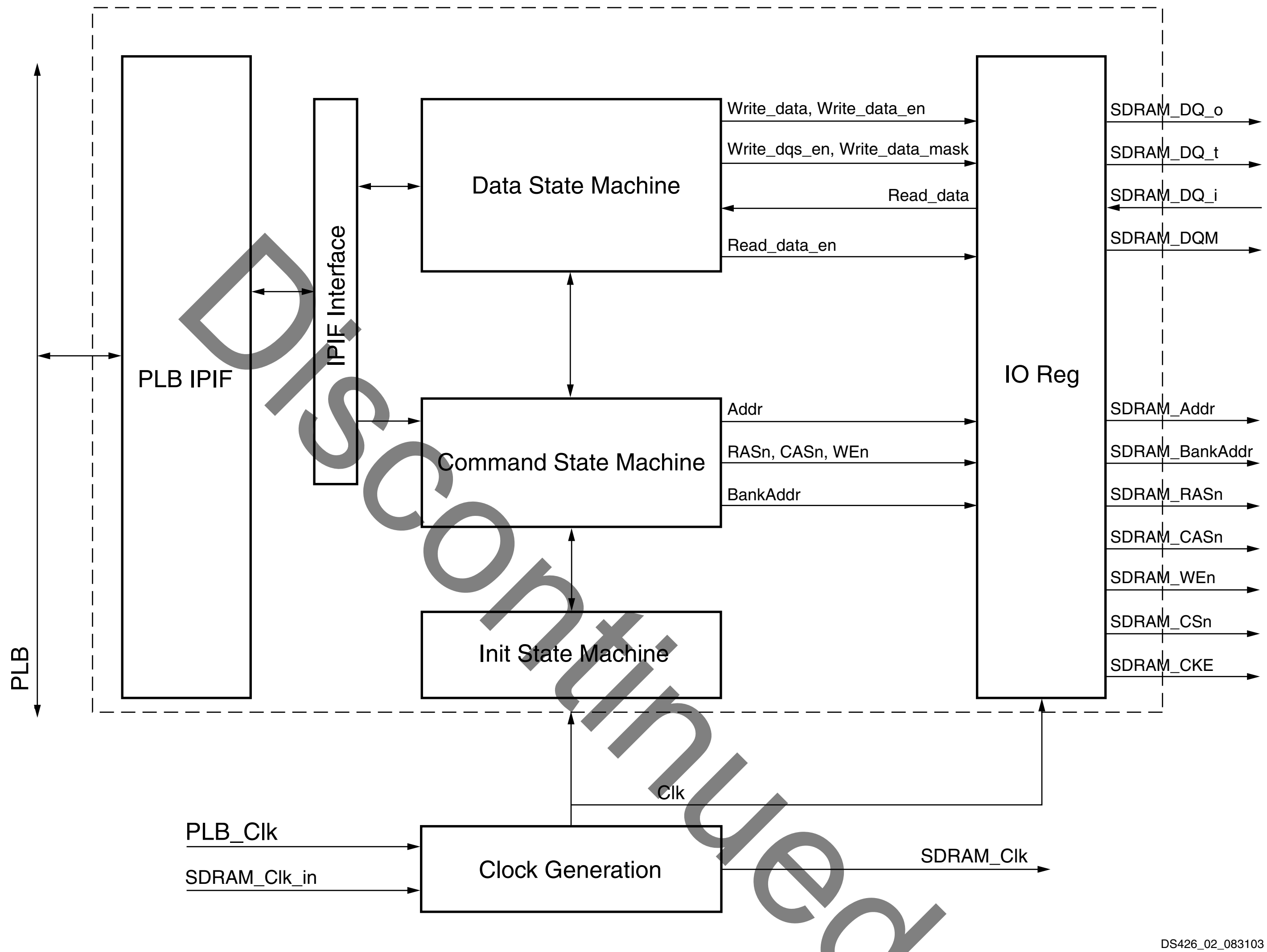

Figure 2: **SDRAM Controller Block Diagram**

Init State Machine

SDRAMs must be powered up and initialized in a predefined manner specified in the SDRAM device data sheet. Once power has been applied and the clock is stable, the SDRAM requires a 100uS delay prior to issuing any command other than a COMMAND INHIBIT or a NOP.

The Init State Machine provides the 100uS delay and the sequencing of the required SDRAM start-up commands. It instructs the Command State Machine to send the proper commands in the proper sequence to the SDRAM. This state machine starts execution after Reset and returns to the IDLE state when Reset is applied.

For a typical SDRAM ~300 nS is required after the 100 uS reset / power-up time to complete the initialization sequence.

During the initialization sequence, the PLB SDRAM controller will respond to accesses by acknowledging the address phase of the transaction and holding the data phase until the initialization process has completed. Once this has completed, the data transfer is serviced and acknowledged.

When the initialization sequence has been completed, the INIT_DONE signal asserts. Note that after Reset has been applied, the 100 uS delay is again implemented before any commands are issued to the SDRAM.

For simulation purposes, the 100 uS reset / power-up delay can be set by the parameter C_SIM_INIT_TIME_PS. Approximately 300 nS after this reset / power-up delay, the initialization sequence is complete.

Note: If C_SIM_INIT_TIM_PS is modified from 100000000 (100 uS), the simulation behavior will vary from the hardware implementation results during initialization. The simulation will no longer be reflecting the hardware behavior during this time.

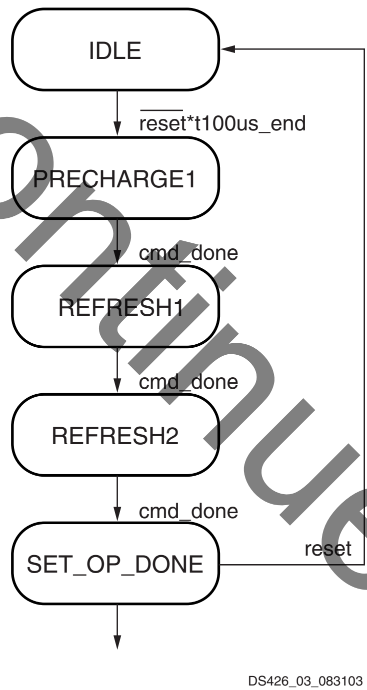

Figure 3: **SDRAM Init State Machine**

Command State Machine

The Command State Machine provides the address bus and commands signals to the SDRAM. It sends the DATA_EN signal to the Data State Machine to start the reception/transmission of data.

If a burst transaction is in progress or a secondary transaction has been received, the Command State Machine will send the next command to the SDRAM while data reception/transmission is still in progress to optimize the SDRAM operation.

A simplified version of the Command State Machine is shown in Figure 5. For readability, only the major state transitions are shown.

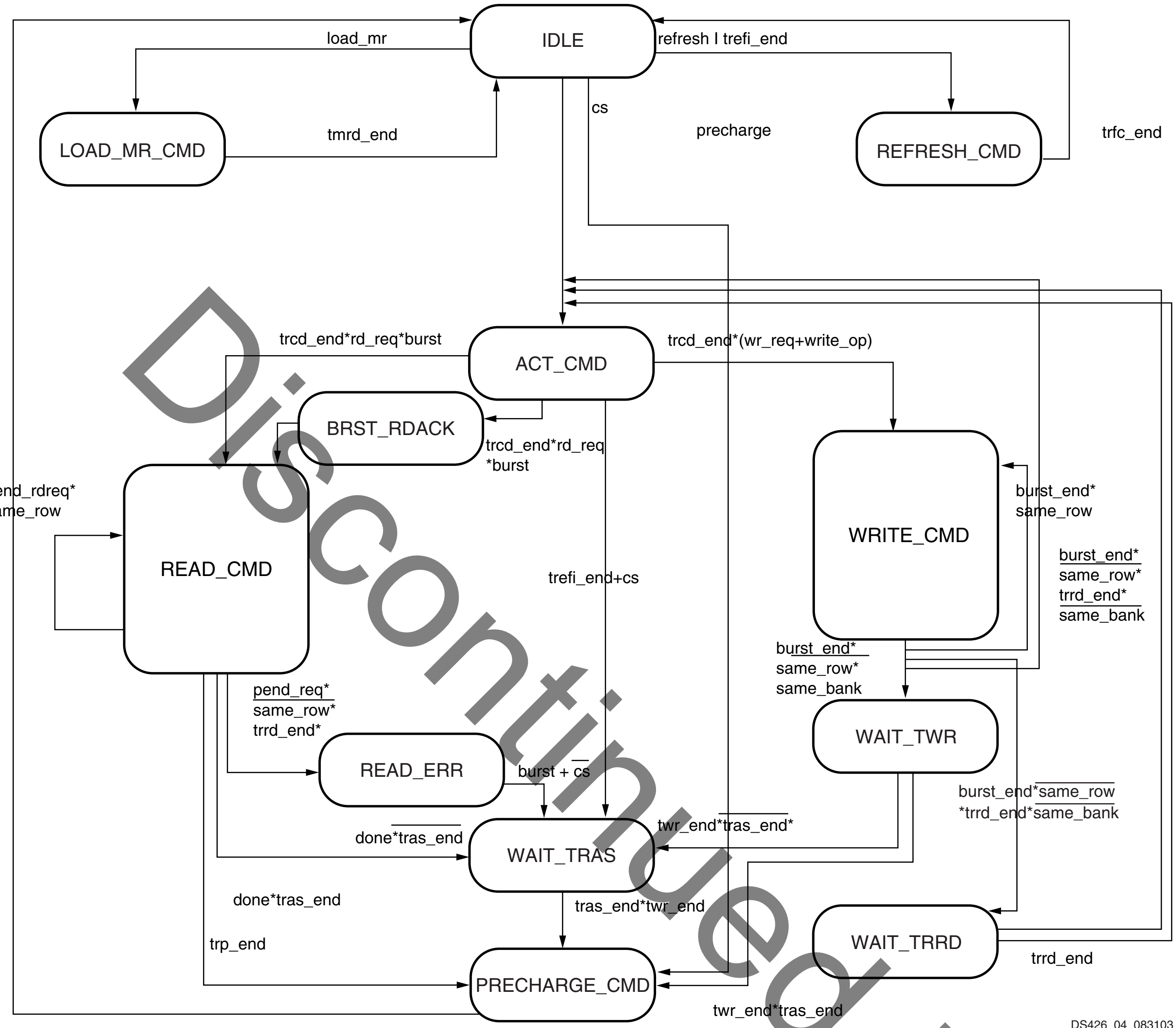

Figure 4: **SDRAM Command State Machine**

Data State Machine

The Data State Machine transfers the data to/from the SDRAM and determines when the specified SDRAM burst is complete. It monitors the PEND_READ and PEND_WRITE signals from the Command State Machine to know if more data transmissions are required. It waits for CAS_LATENCY during read operations and signals when the SDRAM has completed the data transfer for both read and write operations.

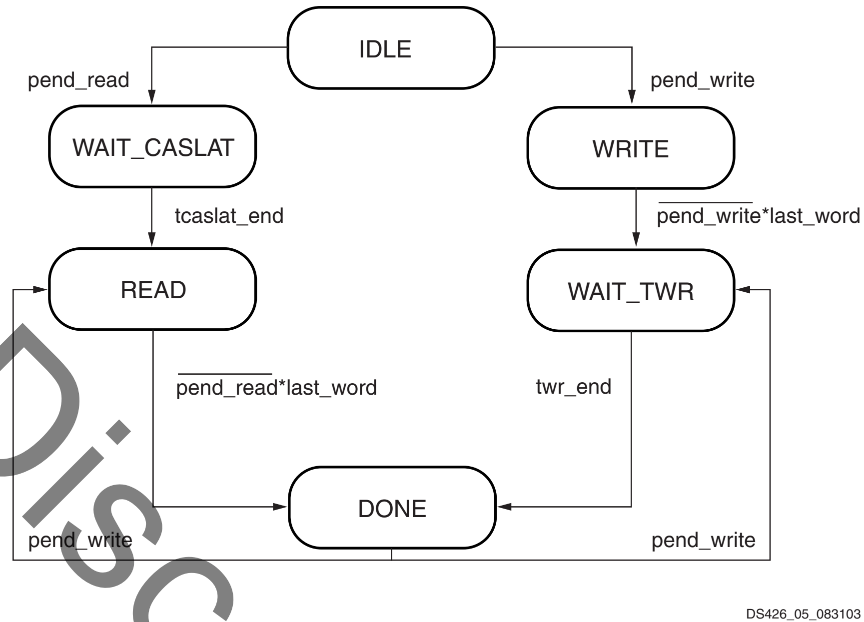

Figure 5: **SDRAM Data State Machine**

Clock Generation

The SDRAM controller will always output a clock to the SDRAM.

To synchronize the SDRAM clock to the internal FPGA clock, the FPGA system design should include a DCM external to the SDRAM core that uses the SDRAM clock input as the feedback clock as shown in Figure 6. This means that the SDRAM clock output from the FPGA must be routed back to the FPGA on a clock pin with a connection to a DCM clock feedback input. The output from the DCM in the FPGA should be connected to the SDRAM_Clk_in input to the SDRAM controller core.

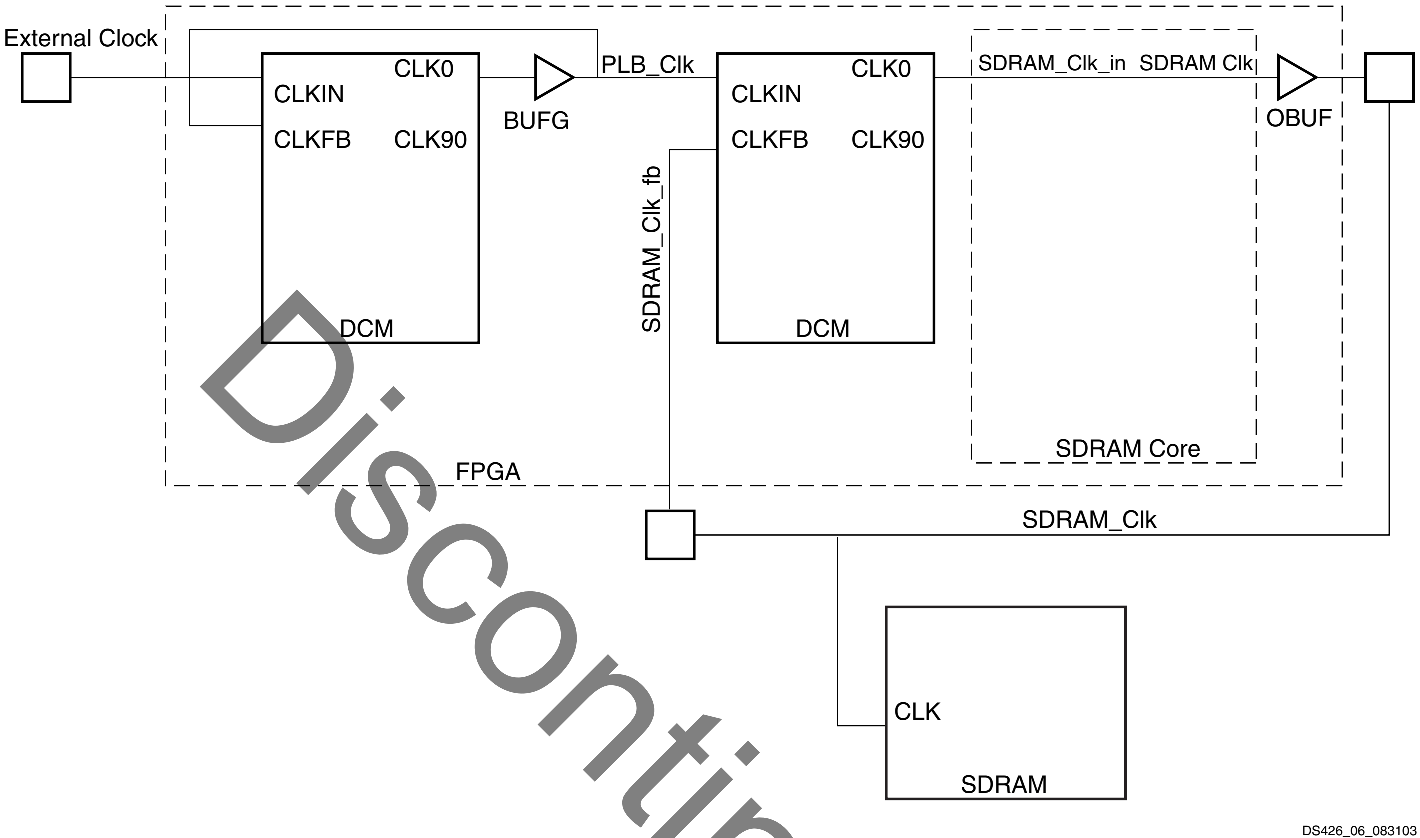

Figure 6: **SDRAM clocked by FPGA Output with feedback**

If the SDRAM is clocked by the same external clock as the FPGA, or if the SDRAM clock feedback is not available, the DCM shown in Figure 7 (or something similar) or Figure 8 should be included in the FPGA external to the SDRAM core. The SDRAM_Clk_in input to the SDRAM core should be connected to PLB_Clk.

NOTE: If DLLs are used, the designer must reference XAPP132 v2.4, "Using the Virtex Delay-Locked Loop" for the correct DLL implementation

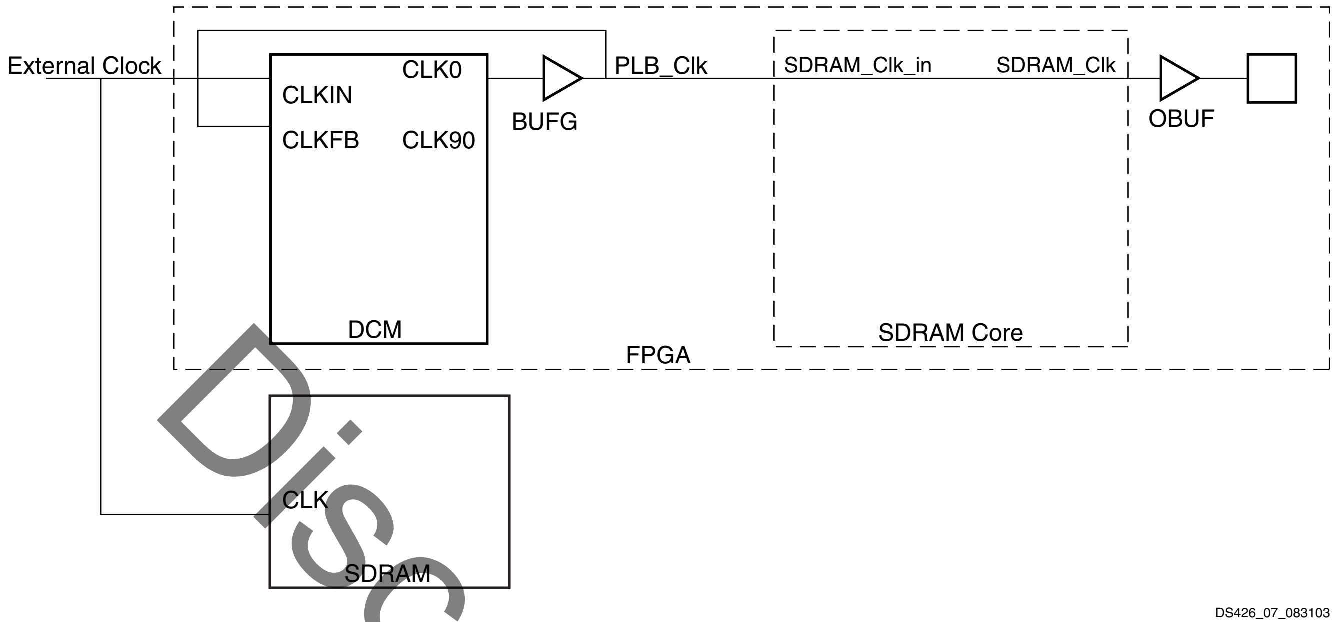

Figure 7: **SDRAM clocked by external clock**

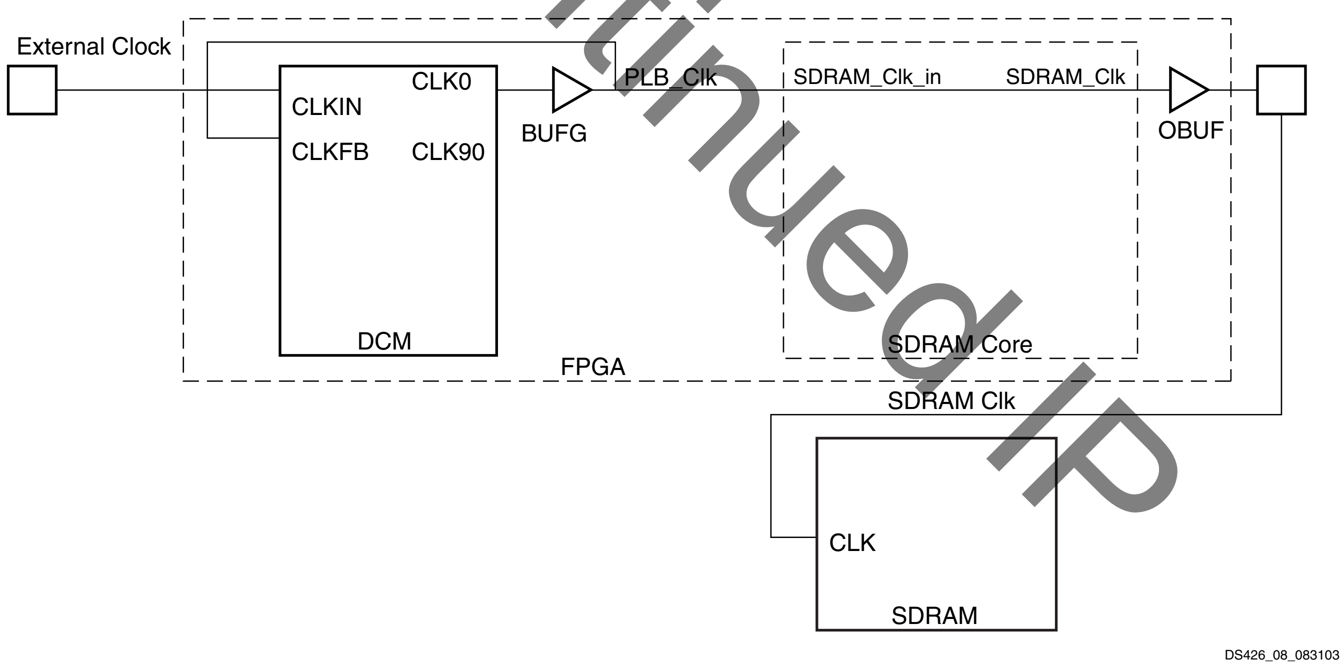

Figure 8: **SDRAM clocked by FPGA Output - no feedback available**

I/O Registers

Control Signals

All control signals and the address bus to the SDRAM are registered in the IOBs of the FPGA.

Write Data Path

The SDRAM I/O registers are used to output the write data to the SDRAM using either the rising or the falling edge of the clock as determined by the C_USE_POSEDGE_OUTREG parameter.

Read Data Path

The SDRAM I/O registers are used to input data from the SDRAM. These registers are always closed on the rising edge of the clock.

Design Implementation

Target Technology

The intended target technology is a Virtex-II Pro FPGA.

Device Utilization and Performance Benchmarks

Since the SDRAM Controller is a module that will be used with other design pieces in the FPGA, the utilization and timing numbers reported in this section are just estimates. As the SDRAM Controller is combined with other pieces of the FPGA design, the utilization of FPGA resources and timing of the SDRAM Controller design will vary from the results reported here.

The SDRAM Controller benchmarks are shown in Table 11 for a Virtex-II Pro -7 FPGA.

Table 9: **SDRAM FPGA Performance and Resource Utilization Benchmarks (Virtex-II Pro -7)**

Parameter Values				Device Resources			f_{MAX} (MHz)
C_SDRAM_DWIDTH	C_USE_POSEDGE_OUTREGS	C_INCLUDE_HIGHSPEED_PIPE	C_INCLUDE_BURST_CACHELN_SUPPORT	Slices	Slice Flip-Flops	4-input LUTs	f_{MAX}
64	0	1	0	402	503	338	95
32	0	1	0	417	532	369	109
16	0	1	0	416	516	366	102
8	0	1	0	439	510	398	125
64	0	1	1	598	637	639	101
32	0	1	1	608	659	664	101
16	0	1	1	617	643	671	102
8	0	1	1	617	640	664	106

Table 9: **SDRAM FPGA Performance and Resource Utilization Benchmarks (Virtex-II Pro -7)**

64	1	0	0	375	416	350	109
32	1	0	0	416	475	371	107
16	1	0	0	417	477	367	105
8	1	0	0	437	480	392	113
64	1	0	1	559	539	633	112
32	1	0	1	609	603	661	103
16	1	0	1	613	605	668	105
8	1	0	1	620	609	669	101

Notes:

1. These benchmark designs contain only the SDRAM Controller without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user conditions.

Reference Documents

The following documents contain reference information important to understanding the Xilinx SDRAM Controller design:

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/18/03	1.0	Initial Xilinx release.
05/20/02	1.1	Update for EDK 1.0
06/20/02	1.2	Revisions for EDK 1.0
07/10/02	1.3	Revisions for version v1_00_b of SDRAM core which added multi-family support and additional SDRAM data width. Changed Tmrd generic to be specified in Tcks instead of ps.
07/23/02	1.4	Revisions for version v1_00_c of SDRAM core which added a pipeline stage to increase operating frequency, burst support, and clarified the C_SDRAM_NUMROWS parameter.
07/29/02	1.5	Add XCO parameters for System Generator
09/10/02	1.6	Added more detail to the Clock Generation section and more information about the C_INCLUDE_SDRAMCLK_DCM parameter.
09/12/02	1.7	Modified document to no longer support C_INCLUDE_SDRAMCLK_DCM. This parameter will always be 0.
10/06/02	1.8	Removed all references (generics/ports) to including the C_INCLUDE_SDRAMCLK_DCM.
11/11/02	1.9	Added tables depicting the address bus slices for SDRAM addresses
01/17/03	1.10	Document cleanup and addition of more design details

Date	Version	Revision
01/31/03	1.11	Added note and cross-reference to the Connecting to Memory section on the first page to the SDRAM Controller Features
04/07/03	1.12	Added generics to allow choice of positive edge or negative edge output registers and to allow setting of the simulation initialization time.
09/18/03	1.12.1	Update graphics; correct trademarks