XAPP134 (v3.1) February 1, 2000

Synthesizable High Performance SDRAM Controller

Summary

Synchronous DRAMs are available in speed grades above 100 MHz using LVTTL I/Os. The Virtex™ series of FPGAs and the Spartan™-II family of FPGAs have many features, such as SelectI/O™ resource and the Clock Delay Lock Loop, that make it easy to interface to high speed Synchronous DRAMs. This application note describes the design and implementation of a synthesizable, parameterizable, flexible, auto-placed-and-routed synchronous DRAM controller in the Virtex FPGA family. The design can also be implemented with a Spartan-II device. A 32-bit wide data interface version can run up to 125 MHz when automatically placed and routed in a Virtex -6 speed grade device. Hand placed versions of the design can run even faster.

Introduction

Synchronous DRAMs (SDRAMs) provide a significant improvement in bandwidth performance over traditional asynchronous DRAMs such as "FPM" (Fast Page Mode) and "EDO" (Extended Data Out). Synchronous DRAMs latch input address, data, and control signals on the clock rising edge, freeing the controller from having to drive address and control for the entire read or write transaction. After a preset number of clock cycles, the data is available on the output latches of the SDRAM for a READ, or can be written into its memory for a WRITE.

Synchronous DRAMs offer features to enhance overall bandwidth performance, such as multiple internal banks, burst mode access, and pipelining of operation executions. Multiple internal banks enable accessing one bank while precharging or refreshing the other banks. Normal refresh and precharge idle time can be minimized by utilizing ping-ponging among multiple banks. Burst-mode access allows the controller to write or read multiple words to the SDRAM without paying the penalty of CAS access for each word written or read. Burst mode has a similar effect as interfacing to a wider data bus memory, because the controller can access one additional word on each consecutive clock cycle. This SDRAM pipelining feature permits the controller to start accesses to other banks while data is being fetched from or written to the currently active bank, eliminating idle time during access latency.

The first section of this application note is a tutorial review of Synchronous DRAMs. The second section describes a Virtex SDRAM Controller. The third section describes the PC board layout and termination recommendations for achieving good signal integrity between Virtex or Spartan-II devices and SDRAMs.

Synchronous DRAM Review

If you are familiar with the SDRAM devices and operation you can skip this section and go straight to the SDRAM Controller Design section.

In general, SDRAMs are multi-bank DRAM arrays that operate at 3.3V and include a command-driven synchronous interface (all signals are registered on the positive edge of the clock signal, CLK). For example, the 16M device referenced in this application note is a dual 512K x 16 array. Each of the two 512K x 16-bit banks is organized as 2,048 rows by 256 columns by 16 bits.

Read and write accesses to the SDRAM are burst oriented; accesses start at a selected location and continue for a programmed number of locations in a programmed sequence. Accesses begin with the registration of the ACTIVATE ROW command, followed by a READ or

WRITE command. The address bits registered coincident with the ACTIVATE ROW command are used to select the intended bank and row. In this example of interfacing to 16M SDRAMS, BA selects the bank, while Address inputs A0-A10 select the row. The address bits coincident with the READ or WRITE command are used to select the starting column location for the burst access.

The SDRAM must be initialized according to manufacturer specifications. Initialization usually consists of a sequence of precharge-all-banks, auto-refresh, and mode-register-set commands.

Pin Descriptions

Table 1 shows the signals in a 16- Megabit, 2 x 512K x 16-bit SDRAM. Signals RAS, CAS, WE, A, and DQ have similar functions to those of a conventional DRAM. SDRAMs have a row and column address like conventional DRAMs. Row Address is presented while RAS is active Low (SDRAM ACT cycle). Like conventional DRAMS, SDRAMs offer burst-mode cycles, analogous to page-mode memory cycles, which access different column addresses. SDRAM cycles are more complex than conventional DRAM cycles. Signals CLK, CKE, DQML, DQMH, and BA are new in SDRAMs. CLK is a free-running clock from which other signals are synchronized. CKE is an enable signal that gates other control inputs. If CKE is false, the SDRAM ignores all other inputs. DQML and DQMH are byte-input enables during write cycles and output enables during read cycles. BA (Bank Address input) identifies the bank where ACT, READ, WRITE, or PRECHARGE commands are being applied. BA is also used to program the 12th bit of the Mode Register.

Table 1: **Interfacing Signals for x16, 16MB SDRAMs**

Signal Name	Type	Description
CS	Input	Chip Enable
CLK	Input	Clock
CKE	Input	Clock Enable
RAS	Input	Row Address Strobe
CAS	Input	Column Address Strobe
WE	Input	Write Enable
DQML, DQMH	Input	Data Mask for Lower, Upper Bytes
BA	Input	Bank Address
A[0:10]	Input	Address
DQ[0:15]	I/O	Data

Mode Register

The Mode Register defines the specific SDRAM mode of operation, including the selection of burst length, as shown in Table 2. The Mode Register, programmed via the LOAD MODE REGISTER command, retains stored information until it is reprogrammed or the device loses power.

Mode Register bits M[2:0] specify burst length. M3 specifies the Wrap Type of burst (sequential or interleaved), M[6:4] specify the CAS Latency, M7 and M8 are used on some SDRAMs to specify the operation mode. M9 specifies the Write Burst Mode on some SDRAMs, and M10 and M11 are reserved for future use.

The Mode Register must be loaded when all banks are idle, and the controller must wait a specified length of time (T_{MRD}) before initiating any subsequent operation. The results of violating these requirements are unpredictable.

Table 2: **Mode Register Definition for 16M SDRAM Device**

Mode Register Field	# of bits	Description
M[2:0]	3	Burst Length
M3	1	Burst Type
M[6:4]	3	CAS Latency
M[8:7]	2	Operation Mode
M9	1	Write Burst Mode
M10, M11	2	Reserved

Burst Length

Read and write accesses to the SDRAM are burst oriented and burst length is programmable. Burst length determines the maximum number of column locations that can be accessed for a given READ or WRITE command. Burst lengths of one, two, four, or eight locations are available for both sequential and interleaved burst types, and a full-page burst is available for the sequential type. The full-page burst type is used with the BURST TERMINATE command to generate arbitrary burst lengths.

Burst Type

Accesses within a burst may be programmed to be sequential or interleaved, and bit M3 determines this burst type.

CAS Latency

CAS latency is the delay in clock cycles between registration of a READ command and availability of the first output data. Latency can be set to one, two, or three clocks, but some devices might not support CAS latency of one.

If a READ command is registered at a clock edge n, and the latency is m clocks, data becomes available by clock edge $n + m$. DQs start driving at the prior clock edge $(n + m - 1)$. If the relevant access times are met, data is valid by clock edge $(n + m)$.

Operating Mode

Normal operating mode is selected by setting M7 and M8 to zero; other values for M7 and M8 are reserved for future use or for test modes. Some SDRAM devices do not support an Operating Mode choice.

Write Burst Mode

When M9 = 0, the burst length determined by [M0:2] applies to both READ and WRITE bursts; when M9 = 1, the programmed burst length applies to READ bursts only, and write access is single-location (nonburst).

SDRAM Commands

Table 3 summarizes standard SDRAM commands.

Table 3: **SDRAM Commands Truth Table**

Function	Symbol	CS	RAS	CAS	WE	BA	A10	A[0:9]	Note
Device Deselect	DSEL	H	X	X	X	X	X	X	2
No Operation	NOP	L	H	H	H	X	X	X	2
Read	READ	L	H	L	H	V	L	V	2,3
Read w/ Auto Precharge	READAP	L	H	L	H	V	H	V	2,3
Write	WRITE	L	H	L	L	V	L	V	2,3
Write w/ Auto Precharge	WRITEAP	L	H	L	L	V	H	V	2,3
Bank Activate	ACT	L	L	H	H	V	V	V	2
Precharge Selected Bank	PRE	L	L	H	L	V	L	X	2
Precharge All Banks	PALL	L	L	H	L	X	H	X	2
Auto Refresh	CBR	L	L	L	H	X	X	X	2,4
Load Mode Register	MRS	L	L	L	L	V	V	V	2

Notes:

1. H: High level, L: Low level, X: don't care, V: Valid data input.
2. CKE is assumed to be High during all of these commands.
3. Only A[7:0 are needed to determine the Column address.
4. Self refresh uses the same command when CKE is Low.

Device Deselect and NOP

Deactivating the CS signal tells the SDRAM to ignore all control inputs. The SDRAMs are put in a No Operation mode (NOP) when CS is active and by deactivating RAS, CAS, and WE. For both Deselect and NOP modes, the current operation finishes when the command is issued.

Bank Activate

The Bank Activate command selects a row in a specified bank of the device. Read and write operations can only be initiated on this activated bank after a minimum T_{RCD} from the activate command.

Read and Read with Auto Precharge

These commands are used after the Bank activate command to initiate the burst read of data. The read command is initiated by activating CS, CAS, and de-asserting WE at the same clock sampling (rising) edge as described in the command truth table. Burst length and CAS latency time are functions of the values programmed by the MRS command. A10 determines whether READ or READAP (Read with Auto Precharge) is initiated.

Write and Write with Auto Precharge

These commands are used after the Bank activate command to initiate burst write. Write is initiated by activating CS, CAS, and WE at the same clock sampling (rising) edge as described in the command truth table. Burst is determined by the values programmed by the MRS command. The A10 state determines whether WRITE or WRITEAP (Write with Auto Precharge) is initiated.

Mode Register Set

The Mode Register Set command (MRS) is used to program the SDRAM for the desired operating mode. It should be used after power is initiated as defined in the power-up sequence before actual operation of the SDRAM. SDRAM functionality can be altered by reprogramming the mode register through the execution of Mode Register Set command if all the banks are pre-charged (i.e., in idle state) before the MRS command is given.

SDRAM Controller Design

This application note describes the design of a Virtex SDRAM controller. The design can also be implemented with a Spartan-II device. The controller has a system interface on one side and a SDRAM controller for two 16 MB SDRAMs on the other side. The design is written in High Level Design Language (Verilog and VHDL). It can be easily modified to fit different memory organizations of system speed and bandwidth requirements.

The controller design has the following features.

- Programmable burst lengths of one, two, four, and eight
- Programmable CAS latency of two and three
- Programmable 16-bit Refresh counter
- Burst length applies to both Read and Write
- Support for SDRAM commands LOAD_MR, AUTO_REFRESH, PRECHARGE, ACTROW, READA, WRITEA, BURST_STOP, and NOP
- System Interface at 62.5 MHz, double data rate (data changes at every clock edge)
- Interface with SDRAM at 125 MHz, single data rate
- Uses two DLLs to provide zero clock skew between system, FPGA and SDRAM.

Figure 1 shows the different blocks in the controller design. The top-level design has two main blocks; the system interface and the controller (sys_int.v and sdram_t.v). The controller module contains the DLLs, the state machine, and all the registers.

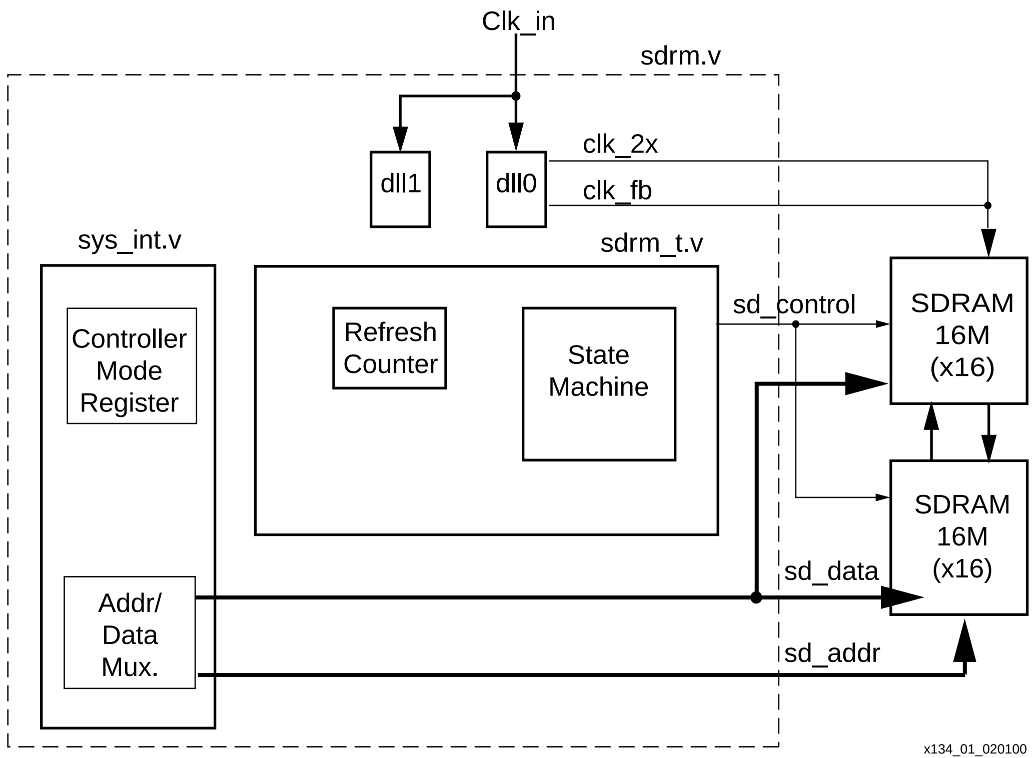

Figure 1: **Controller Block Diagram**

Pin Descriptions

The pin descriptions are outlined in Table 4, organized according to system or SDRAM interface.

Table 4: **Pin Descriptions**

Interface	Pin		Width	Notes
	Name	Direction		
System	AD	In/Out	32	Address and data bus. (double data rate)
	data_addr_n	In	1	When data_addr_n = 1, AD contains data. When data_addr_n = 0, AD contains address.
	we_rn	In	1	
	Reset	In	1	
	Clkp	In	1	62.5 MHz
	Clk_FBp	In	1	Feedback clock must connect to Clk_SDp
SDRAM	sd_data	In/Out	32	Single data rate
	sd_add	Out	11	
	sd_ras	Out	1	
	sd_cas	Out	1	
	sd_we	Out	1	
	sd_ba	Out	1	
	Clk_SDp	Out	1	125 MHz
	sd_cke	Out	1	
	sd_cs1	Out	1	
	sd_cs2	Out	1	
	sd_dqm	Out	4	

System Interface

The control signals from the system are: we_rn, and data_addr_n. These signals issue the four basic commands: addr_wr, data_wr, addr_rd, and data_rd. When data_addr_n is low, the data from AD[29:28] is used to issue additional SDRAM commands. In this design, PreCharge and Load Controller Mode Register are the same command. Table 5 shows the truth table for system commands.

Table 5: **System Commands Truth Table**

Commands		**Control Signals**		**AD[29:28]**
		data_addr_n	**we_rn**	
Addr_wr	Read/Write	0	1	00
	Auto refresh	0	1	11
	Precharge SDRAM & Load Controller Mode Register (Note 1)	0	1	10
	Load SDRAM Mode Register	0	1	01
Data_wr		1	1	Don't care
Addr_read		0	0	Same as Addr_wr
Data_read		1	0	Don't care

Notes:

1. Every time the system issues a precharge command, it is also loading the Controller Mode Register.

The following listing summarizes how the system issues different SDRAM commands to the controller.

PRECHARGE	Place Addr_wr command on the bus
	Set AD[29:28] to 10
	Place Data_wr command on the bus
	Put the values for Controller Mode register on A[27:0]
AUTO_REFRESH	Place Addr_wr command on the bus
	Set AD[29:28] to 11
LOAD_MR	Place Addr_wr command on the bus
	Set AD[29:28] to 01
	Place Mode Register values on A[20:10]
WRITEA	Place Addr_wr command on the bus
	Set AD[29:28] to 00
	Place Data_wr command on the bus
	Place data on AD on every clock edge until the end of burst
READA	Place Addr_rd command on the bus
	Set AD[29:28] to 00
	Place Data_rd command on the bus

Address Multiplexing

The controller registers the address from the AD bus when data_addr_n is Low. The address is then multiplexed into the SDRAM row and column address. In this design, AD[21] maps to ba, AD[20:10] maps to the SDRAM row address, and AD[9:2] maps to the SDRAM column address. Assuming all reads and writes are in bursts of eight, AD[1:2] are not used. Table 6 shows the system address to SDRAM address mapping.

Table 6: **System Address Mapping to x16 16M SDRAM**

SDRAM Address	Row Address	Column Address
SD_A0	A10	A2
SD_A1	A11	A3
SD_A2	A12	A4
SD_A3	A13	A5
SD_A4	A14	A6
SD_A5	A15	A7
SD_A6	A16	A8
SD_A7	A17	A9
SD_A8	A18	X
SD_A9	A19	X
SD_A10	A20	H
SD_BA	A21	A21

Data Register

During the Write cycle, the controller registers data from the AD bus with clk_2x. The controller gets data at every input clock edge. During the Read cycle, the controller registers data from the SDRAM with clk_2x and puts the data onto the AD bus.

Controller

Clock De-skew

Clock skew and clock delay can have a substantial impact on designs running at higher than 100 MHz. This design uses DLLs to de-skew the FPGA and SDRAM clocks. The design also uses the 2x feature of the DLLs to allow the system clock to be half the SDRAM frequency. As an example, if the SDRAM interface is at 125 MHz, then the clock input to the FPGA is at 62.5 MHz

Figure 2 shows the two DLLs used in the design. DLL0 provides the clk2x signal to the SDRAM. It also receives the clock feedback from the SDRAM. DLL1 provides both the clk and the clk2x internal signal to the FPGA (Clk_j and Clk_i). DLL1s feedback is from clk2x.

It is not possible to use one DLL to provide both the FPGA and SDRAM clocks since the SDRAM clock goes through an OBUF delay creating skew between the two clocks. Using two DLLs with the same clock input and separate feedback signals achieves zero-delay between the input clock, the FPGA clock, and the SDRAM clock.

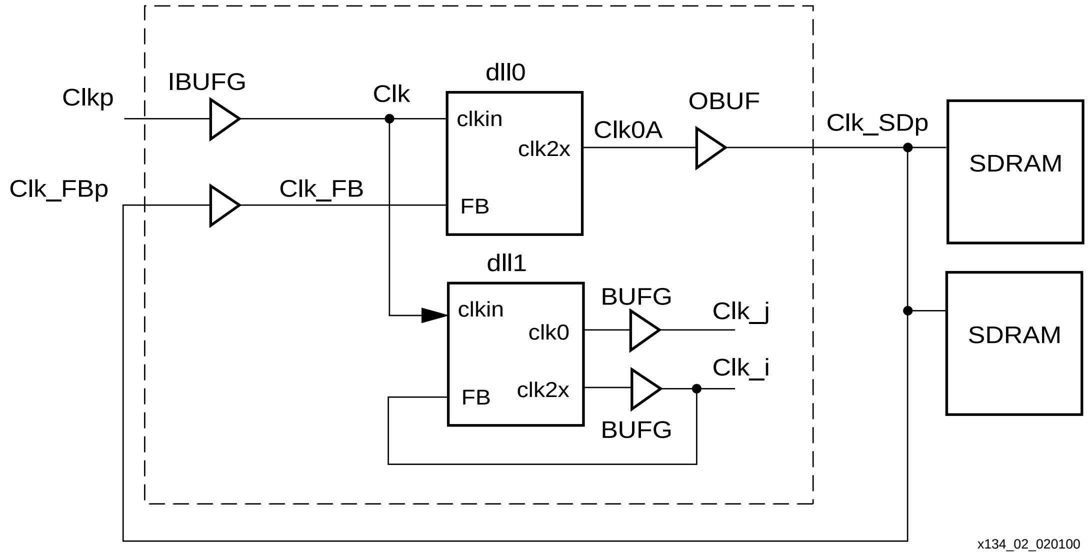

Figure 2: **DLLs in a Reference Design**

Controller Mode Register

Similar to the SDRAM, the controller has a mode register. The controller mode register should not be confused with the SDRAM mode register. The controller mode register allows burst length, CAS latency, RAS-to-CAS delay, Refresh count, and the Refresh active period to be reprogrammed without having to recompile the FPGA design, see Table 7.

- CAS Latency: CAS latency value in the controller mode register should be one less than that in the SDRAM mode register.
- RAS-to-CAS Delay: The delay should be $(T_{RCD}/T_{CK}) - 2$.
- Burst Length: The burst length value should be one less than the value in the SDRAM mode register.
- Refresh Count: The controller refresh counter periodically issues a refresh command to the SDRAM. The refresh count should be set to $T_{REF}/(T_{CK}$ x the number of rows).
- Refresh Active Period: The refresh active period determines the length of the controllers wait state before sending a new command after a refresh command. The value should be set to $(T_{RC}/T_{CK}) - 3$.

Table 7: **Controller Mode Register Definitions**

Mode Register			Mode Register Value	
Field	**# of bits**	**Description**	**SDRAM**	**Controller**
C[1:0]	2	CAS latency	2, 3	1, 2
C[3:2]	2	RAS to CAS delay		0, 1
C[6:4]	3	Burst length	1, 2, 4, 8	0, 1, 3, 7
C[23:8]	16	Refresh count		2000-4000
C[27:24]	4	Refresh active period		7

The controller mode register must be loaded before any SDRAM operations are started. Values programmed in the controller mode register should match the corresponding values programmed in the SDRAM mode register. The controller mode register is programmed when the system issues a PRECHARGE command.

State Machine

An overview of the state machine is shown in Figure 3.

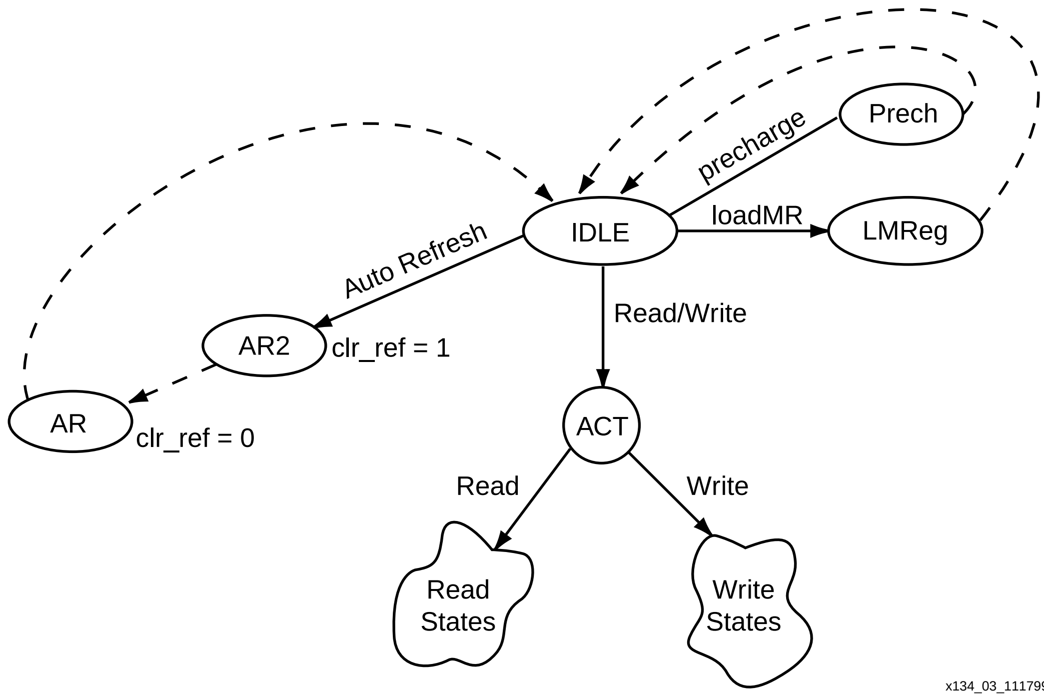

Figure 3: **State Machine Overview**

The controller awakens in the IDLE state and then changes to Precharge, Load Mode Reg, AutoRefresh, or ACT, depending on the system command. The dashed lines in the state machine diagram indicate the automatic sequence. In AutoRefresh, the state machine needs two states, one to enable clr_ref (clear refresh), and another to disable clr_ref. The Read and Write states are more complex, as shown in detail in Figure 4.

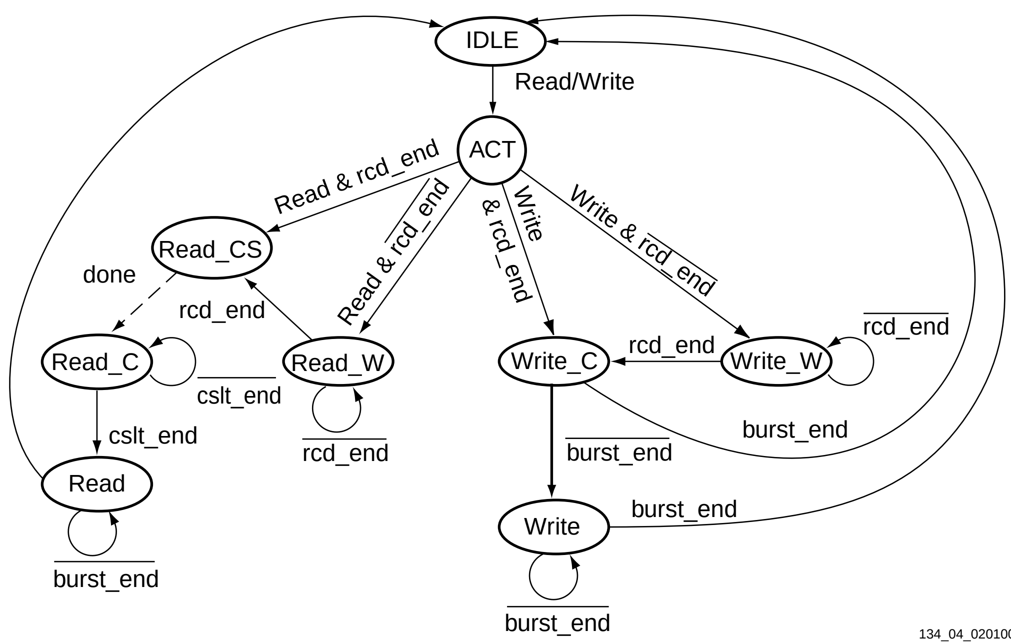

Figure 4: **Read and Write States**

For Read and Write, the controller first goes into ACT state (active row). During Write, the controller needs to check for end of RAS-to-CAS delay and burst end. Three different states are needed: WRITE_W, to wait for rcd_end, WRITE_C, to issue the first data and the WRITE command to the SDRAM, and WRITE to continue to provide data until burst end.

During Read, in addition to checking RAS-to-CAS delay and burst end, the controller also needs to check for CAS latency. The four Read states are: READ_W to wait for rcd_end, READ_CS to satisfy the RAS-to-CAS delay, READ_C to issue a READ command to the SDRAM, and READ to continue to accept data until burst end.

Timing Diagrams

The timing diagrams show the sys_clk inputs to the controller at half the speed of the controller and SDRAM clk. The 2x feature of the DLL allows the use of a slower clock at the input.

Write Cycle

Figure 5 is the timing diagram for writing a burst of eight data words to the SDRAM.

At T_1, the system places the addr_wr command on the bus and sets AD[29:28] to 00. At T_2, the system places the data_wr command on the bus and also places the first data on AD. All subsequent data is placed on AD after every SDRAM clock cycle. The SDRAM sees the ACT command and row address at T_6. Then after RAS-to-CAS delay, it receives the WRITEA command and the first data. The burst write of eight words is completed at T_{16}.

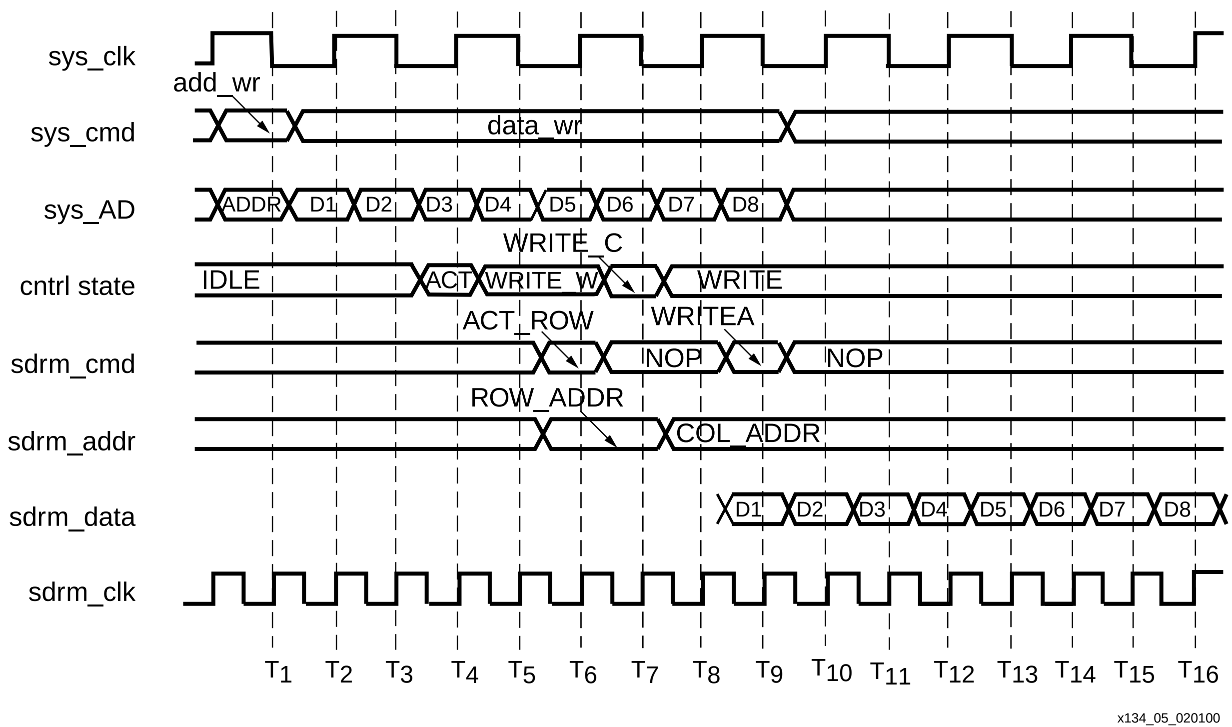

Figure 5: **Write Timing Diagram**

Read Cycle

Figure 6 is the timing diagram for reading a burst of eight data words from the SDRAM.

At T_1, the system places add_rd command on the bus, AD[29:28] is set to 00, and the SDRAM address is set on AD[21:2]. At T_2, the system places the data_rd command on the bus and AD is disabled. The SDRAM sees the ACT command and row address at T_6. After RAS-to-CAS delay, the SDRAM receives the READA command and the column address T_9. After CAS latency delay, the SDRAM places the first data on the sdrm_data bus. The system gets all eight words of data at T_{21}.

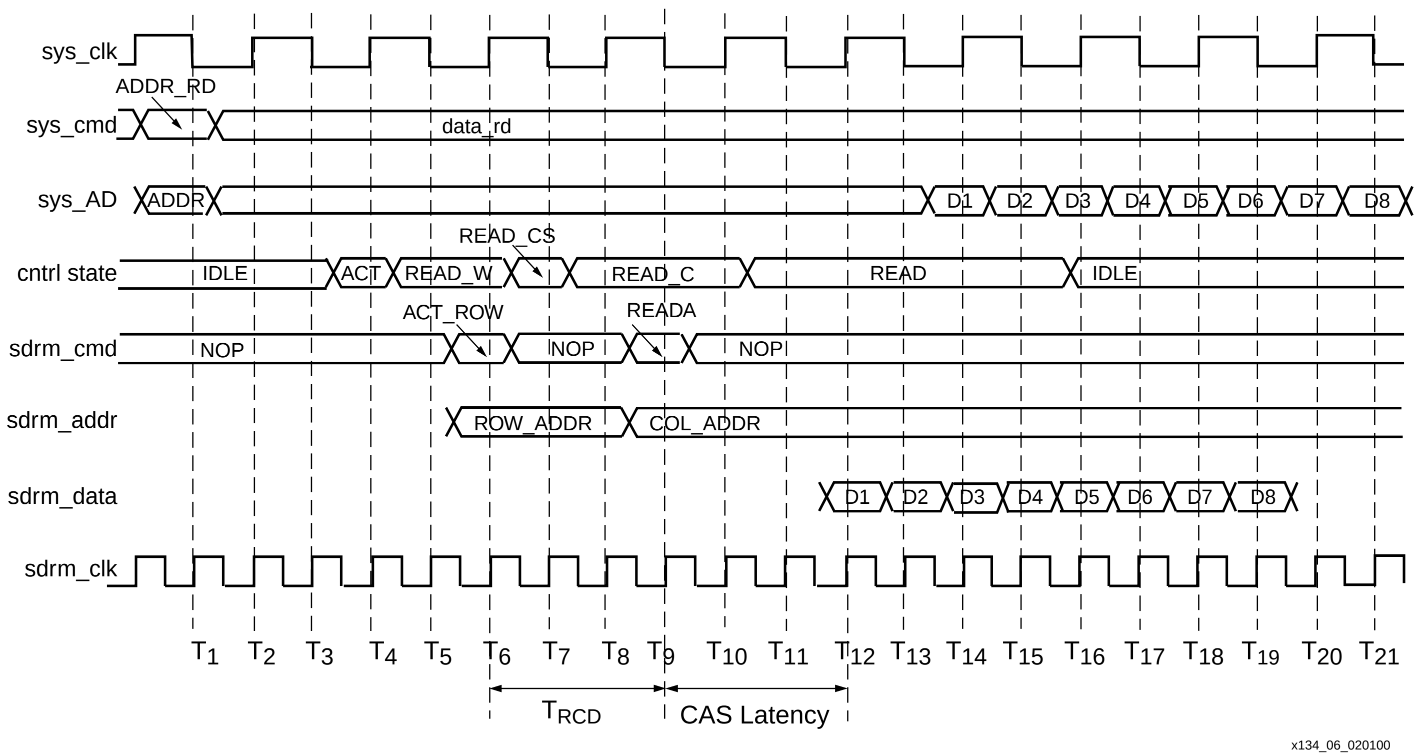

Figure 6: **Read Timing Diagram**

Implementation

The main challenge for this reference design is to meet the timing for all signals interfacing to the SDRAM. All SDRAM signals are registered in the IOBs and use fast input and output buffers. The Virtex and SDRAM I/O times are listed in Table 8.

Table 8: **Virtex and SDRAM I/O Times**

Device	T_{OH}	T_{AC}	T_{SU}	T_{HOLD}	T_{CYC}
SDRAM-8	3.0	6.0	2.0	1.0	8 ns
XCVxxx-6	1.0	3.9	1.7	0.0	8 ns

Write Cycle Timing

In a Write cycle, all signals must meet the setup and hold of the SDRAM device.

$$\text{Virtex } T_{AC} + \text{board delay} + \text{SDRAM } T_{SU} \leq T_{CK}$$

$$3.9 + \text{board delay} + 2.0 \leq 8.0$$

$$\text{board delay} + 5.9 \leq 8.0$$

The Virtex clock-to-out and SDRAM setup is 5.9 ns. This allowing ample margin for the board delay.

The SDRAM also has a 1 ns hold time requirement. FPGA vendors have traditionally not published minimum output hold times. This parameter (T_{OH}) is expected to be in the range of 1 ns for all clocked LVTTL outputs with drive capability below 16 mA.

Read Cycle Timing

In a Read cycle, data must meet the setup and hold of the FPGA device.

$$\text{SDRAM } T_{AC} + \text{board delay} + \text{Virtex } T_{SU} \leq T_{CK}$$

$$6.0 + \text{board delay} + 1.7 \leq 8.0$$

$$\text{board delay} + 7.7 \leq 8.0$$

The Virtex setup time listed in the data sheet is 1.9 ns. However, with the DLL, the actual device setup time is 1.7 ns.

The SDRAM clock-to-out and Virtex setup is 7.7 ns, allowing only 0.3 ns for board delay. The FPGA has a zero hold time requirement.

Synthesis scripts and place and route constraints

Details of the synthesis and place and route scripts are presented in the following sections

Synthesis script

Set the period constraint for all clock signals

```
create_clock Clk_i -period 10 -waveform {0 5}
create_clock Clk_j -period 20 -waveform {0 5}
```

Place and route constraints

- All place and route constraints are in the sdrm.ucf file. The periods for each clock are specified along with the time specification for paths between clk1x and clk2x.

```
NET "Clkp" PERIOD = 16ns ;
NET "Clk_j" PERIOD = 16ns ;
NET "Clk_i" PERIOD = 8ns ;
NET Clk_i  TNM=c2x;
NET Clk_j  TNM=c1x;
TIMESPEC TS10= FROM: c2x: TO: c1x: 8ns;
TIMESPEC TS11= FROM: c1x: TO: c2x: 8ns;
```

- For signals going to the SDRAM, the OFFSET attribute is used to subtract the SDRAM clock-to-out delay/setup and board delay from the PERIOD.

```
#The min setup (TSU) of the SDRAM-8 is 2ns, plus
500ps of board delay
#we need to add this OFFSET to all outputs to SDRAM
#
```

```
NET sd_add[*] OFFSET = OUT : 2.5 : BEFORE : Clkp ;
NET sd_data[*] OFFSET = OUT : 2.5 : BEFORE : Clkp ;
NET sd_ras OFFSET = OUT : 2.5 : BEFORE : Clkp ;
NET sd_cas OFFSET = OUT : 2.5 : BEFORE : Clkp ;
NET sd_we OFFSET = OUT : 2.5 : BEFORE : Clkp ;
NET sd_ba OFFSET = OUT : 2.5 : BEFORE : Clkp ;
#
#The max clock-to-out (Tac) of the SDRAM-8 is 6ns, plus 300ps of
board delay
#we need to add this OFFSET to all inputs from SDRAM
#NET sd_data[*] OFFSET = IN : 6.3 : AFTER : Clkp;
```

- Set NODELAY mode for inputs from SDRAM.

```
#By default, the IBUF has a DELAY element to
guarantee 0 hold time
#By turning off the DELAY element, we save ~500ps
in IBUF delay
NET sd_data[0] NODELAY;
NET sd_data[1] NODELAY;
```

Timing Analysis

The trace command is used to verify whether the constraints specified in the ucf is met.

```
tre -v -u sdrm_par sdrm.pcf
```

Results

The SDRAM controller design uses 165 CLB slices, two DLLS, two global clock buffers, and 90 IOBs. The design is verified with a back-annotated simulation at 125 MHz. Table 9 shows projected performance for each of the three Virtex speed grades and the corresponding SDRAM speed grade.

Table 9: **Design Performance with Different Virtex Speed Grades**

32-bit Data Bus Width		
Projected Performance	**Recommended Virtex Speed Grade**	**Recommended SDRAM Speed Grade**
80-90 MHz	-4	-12 or faster
95-105 MHz	-5	-10 or faster
110-125 MHz	-6	-8 or faster

Printed Circuit Board Design and Trace Termination

Figure 7 shows a printed circuit board reference design for a 32-bit wide Synchronous DRAM interfacing to the Virtex FPGA XCV300 device. Two Micron MT48LC1M16A1 SDRAM devices along with the XCV300 in a BGA 432 package are used in this example. The assumptions are: the PC board trace impedance is controlled to 50Ω and the two SDRAMs are located within two inches of the FPGA.

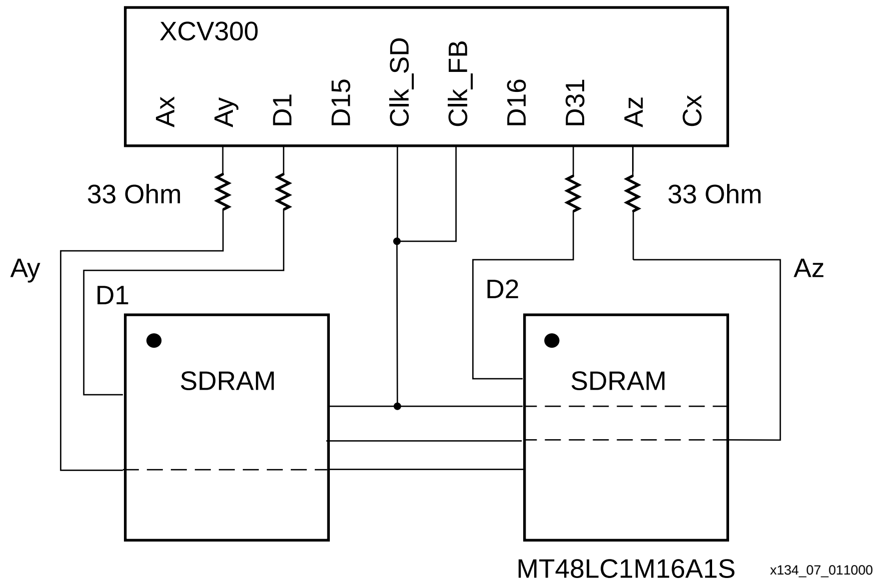

Figure 7: **Reference Printed Circuit Board Design for 32-bit SDRAM Interface.**

All address and data lines are terminated with a source terminating resistor located as close as possible to the FPGA package. The value of the resistor is 33Ω. To make traces as short as possible, all data lines are driven from the pins located in the center of the package side facing the SDRAMs. The address and control pins surround the data pins on both sides. They have two loads since both SDRAMs get the same address and control signals.

A clock output pin close to the clock input and clock feedback input pins was selected. Two traces are driven out of the clock output pin. One drives the two SDRAM clock inputs on both packages. The second trace loops back to drive the Clock Feedback Input pin. With the assumption that the two clock input loads on the SDRAMs are almost equivalent to one FPGA clock feedback input load, the two traces are made equal in length. This guarantees the delay to the SDRAM clock pin is the same as the delay to the Clock Feedback Input pin. No termination is necessary for the clock signals if the trace length is kept below two inches.

For a more complex configuration, use a commercially available PC board design tool to analyze the trace signal integrity. Xilinx provides IBIS models for all of the Virtex series and Spartan-II family I/Os for use with these tools. SDRAM output IBIS models are also available from a variety of memory vendors. Designers can insure signal integrity of all signals on the board by using these tools and the available input/output models.

Design Implementation Summary

The following is a summary of the design techniques used in the SDRAM controller to meet the desired performance:

1. Use DLLs and Global Clock Buffers to remove all clock delays and clock skew.
2. Use fast output buffers (IOBUF_F_12) for all outputs to the SDRAM. They are about 2 ns faster than the regular output buffers
3. Use an input buffer in non-DELAY mode on all inputs from the SDRAM. These non-DELAY inputs are about 500 ps faster than the default input buffers.
4. Make sure all signals interfaced to the SDRAM and the system, and the 3-state signal for the SDRAM data, are registered in the IOB. This is done by using the `- pr b` option when running map.
5. Some high fan-out signals in the designs are duplicated to reduce net delay. The sd_data 3-state signal has a fan-out of 32. This signal is duplicated to four signals, each having eight loads.

6. Specify timing constraints in the ucf file
7. Use the one_hot state machine.
8. Use SRL16 (Shift Register LUT) to delay data.

Conclusion

This application note and reference design demonstrates a Virtex FPGA interfacing to external SDRAMs at 125 MHz. The design is developed using Verilog/VHDL, it can be easily modified for a different system design requirements. The design can also be implemented with a Spartan-II device.

Restrictions and Disclaimers

Behavioral and back-annotated versions of the design have been simulated using the Verilog-XL simulator. This design has been also bench-tested using the provided test vectors.

The design is available in VHDL and Verilog for PC and UNIX formats:

ftp://ftp.xilinx.com/pub/applications/xapp/xapp134_vhdl.zip
ftp://ftp.xilinx.com/pub/applications/xapp/xapp134_verilog.zip
ftp://ftp.xilinx.com/pub/applications/xapp/xapp134_vhdl.tar.Z
ftp://ftp.xilinx.com/pub/applications/xapp/xapp134_verilog.tar.Z

References

1. High Performance Memories, New architecture DRAMs and SRAMs evolution and function: Betty Prince, 1996.
2. Micron Technology, Synchronous DRAM: MT48LC1M16A1 Data Sheet, 1997.
3. Xilinx Inc., **Virtex 2.5 V Field Programmable Gate Arrays Data Sheet**,2000.
4. Xilinx Inc., **Virtex-E 1.8V Field Programmable Gate Arrays Data Sheet**, 2000.
5. Xilinx Inc., **XAPP132**, Using Virtex Delay Locked Loop, Application Note,1999.
6. Xilinx Inc., **XAPP133**, Using the Virtex Select I/O, Application Note,1999.
7. Xilinx Inc., **Spartan-II 2.5 V., Field Programmable Gate Array Data Sheet**, 2000.
8. Xilinx Inc., **XAPP174**, Using Delay-Locked Loops in Spartan-II FPGAs, Application Note, 2000.
9. Xilinx Inc., **XAPP179**, Using SelectI/O Interfaces in Spartan-II FPGAs, Application Note, 2000.

Revision History

Date	Version #	Revision
4/28/99	1.0	Initial Xilinx release.
5/7/99	1.1	Corrected URL address for design reference on page 6.
7/21/99	2.0	Updated SDRAM Controller design section.
9/10/99	2.1	Updated OFFSET constraints.
12/15/99	2.2	Reformatted for new template, revised System Interface notes.
1/10/00	3.0	Added support for Spartan-II family FPGAs.
2/1/00	3.1	Revised Figures 5 & 6