SDR SDRAM Controller

March 2014 **Reference Design RD1174**

Introduction

The Synchronous dynamic random access memory (SDRAM) is dynamic random access memory (DRAM) that has a synchronous interface. SDR (Single Data Rate) SDRAM can accept one command and transfer one word of data per clock cycle. Typical SDR SDRAM clock rates are 66, 100, and 133 MHz (periods of 15, 10, and 7.5 ns). It is a mainstream memory of choice due to speed, burst access and pipeline features.

This SDRAM Controller reference design, located between the SDRAM and the bus master, reduces the user's effort to deal with the SDRAM command interface by providing a simple generic system interface to the bus master.

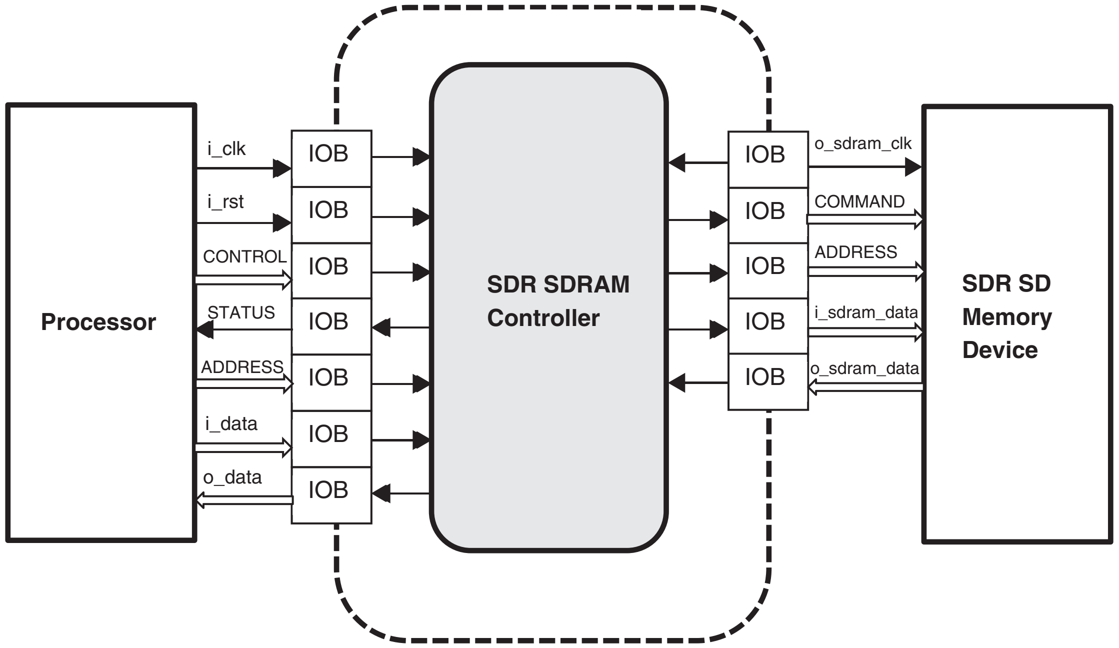
The design is implemented in Verilog language. The Lattice Diamond® for MachXO3L device and Lattice iCEcube2™ for iCE40™ device Place and Route tool integrated with the Synopsys Synplify Pro® synthesis tool are used for the implementation of the design. The design can be targeted to other iCE40 FPGA product family devices.

Features

- Supports up to 27 address space, up to 4 banks
- Compile time configurable timing parameters like (CAS latency, tRP, tRCD, tREFC, tMRD …)
- Simplifies SDRAM command interface to standard system read/write interface
- Supports Auto Refresh and Self Refresh
- Supports flexible Row and Column addressing
- Supports CAS latencies of 2 and 3
- Run time configurable Refresh rate
- Automatically generates initialization and refresh sequences
- Flexible user controlled burst length of 1, 2, 4, 8 and page as well as stop signal
- User friendly control and status signals like busy, command acknowledge, data valid and request, indications for completion of write and read data
- User controlled operation modes by configuring load mode register
- Provision for disable auto refresh
- Power down mode

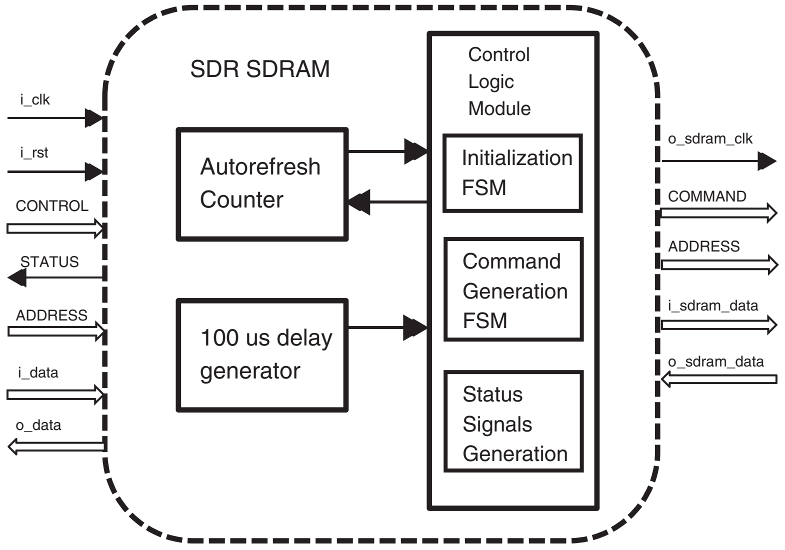
System Block Diagram

Figure 1. System Block Diagram



Functional Description

Figure 2. Functional Block Diagram

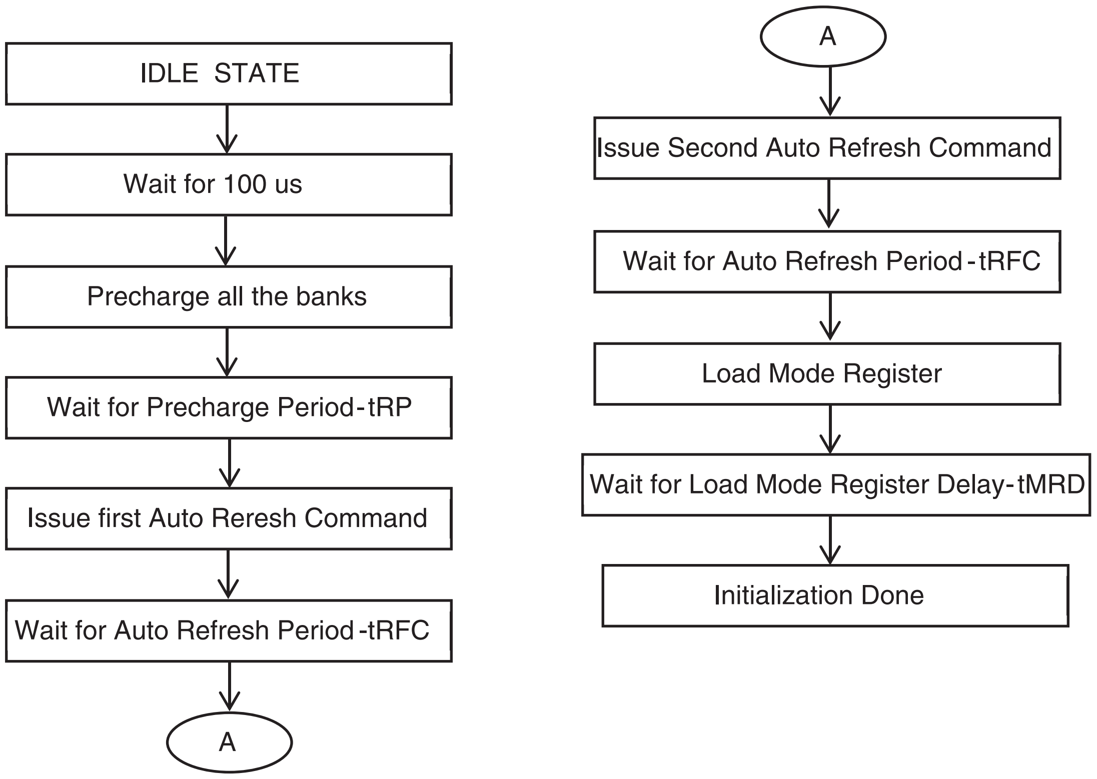


Design Details

Initialization

Prior to performing normal SDRAM memory access, memory needs to be initialized by a sequence of commands. Operational procedures other than those specified may result in undefined operation. The Initialization FSM handles this initialization.

Figure 3. SDRAM Initialization flow chart


IDLE STATE
Wait for 100 us
Precharge all the banks
Wait for Precharge Period-tRP
Issue first Auto Reresh Command
Wait for Auto Refresh Period-tRFC
A
A
Issue Second Auto Refresh Command
Wait for Auto Refresh Period-tRFC
Load Mode Register
Wait for Load Mode Register Delay-tMRD
Initialization Done

Command Generation

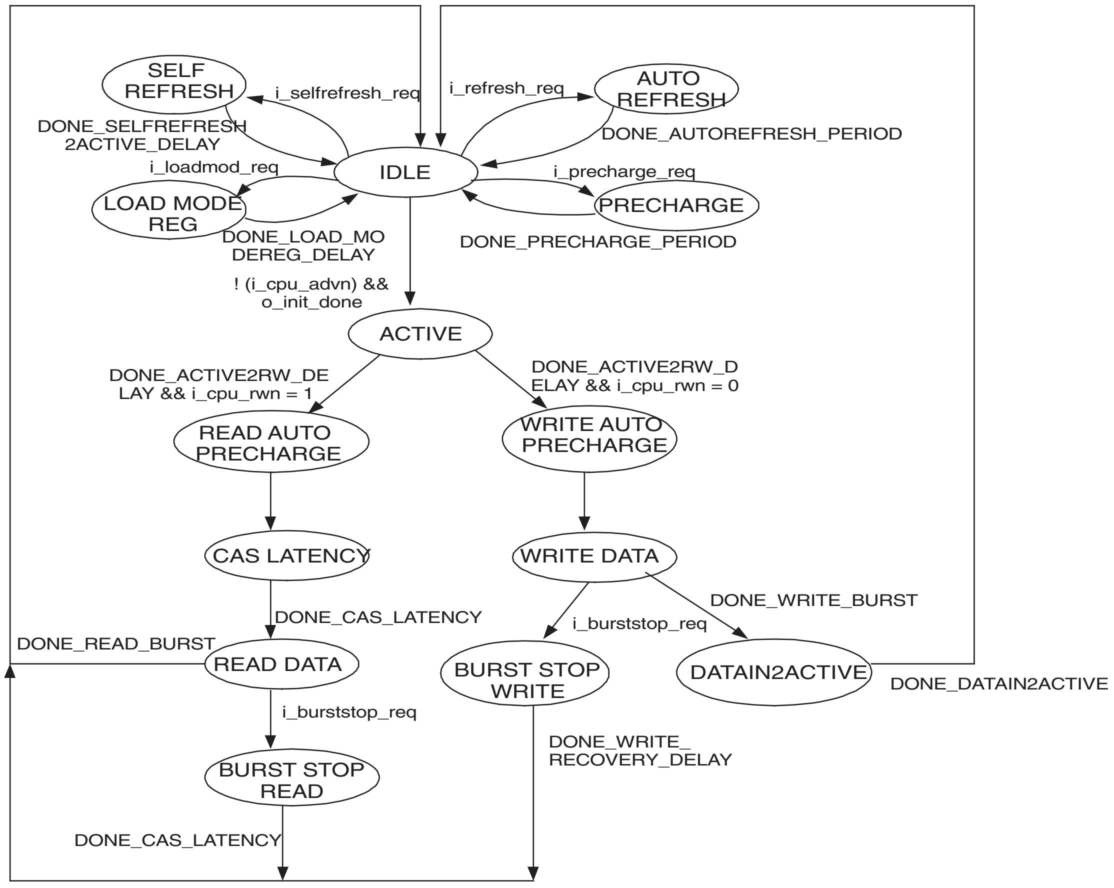
Commands are generated in command generation FSM. Table 1 shows Truth table for various commands and corresponding SDR SDRAM control signal levels.

Table 1. Truth Table for SDRAM Commands

Commands	CS	RAS	CAS	WE
No Operation (NOP)	L	H	H	H
Active	L	L	H	H
Read	L	H	L	H
Write	L	H	L	L
Burst Terminate	L	H	H	L
Precharge	L	L	H	L
Auto Refresh or Self Refresh	L	L	L	H

Command generation FSM states and its control signals are shown in Figure 4.

Figure 4. Command Generation FSM



Command Generation FSM will generate SDR SDRAM commands.

Read and Write access to the SDRAM are burst oriented, accesses starts at a specified address and continue for a programmed number of locations in a programmed sequence. Accesses begin with the registration of an ACTIVE command, which is then followed by a READ or WRITE command. The address bits coincident with the ACTIVE command are used to select the bank and row to be accessed. The address bits coincident with the READ or WRITE command are used to select the starting column location for the burst access.

The controller provides programmable READ or WRITE burst lengths 1, 2, 4 or 8 locations, or the full page, with a burst terminate option. An auto precharge function may be enabled to provide a row precharge that is initiated at the end of the burst sequence.

Signal Description

Table 2. Signal Description

Signal	Width	Type	Description
i_clk	1	Input	System Clock
i_rst	1	Input	Asynchronous Active low Reset
i_advn	1	Input	Active low Address enable for Active state
i_rwn	1	Input	R/W Enable: 1-Read,0-Write
i_addr	[ROWADDR_MSB:COLADDR_LSB]	Input	Input Address to SDRAM Controller
i_selfrefrresh_req	1	Input	Request for Self Refresh
i_loadmod_req	1	Input	Request for Loading Mode Register
i_burststop_req	1	Input	Request for Burst Stop
i_disable_active	1	Input	Disables opening a row if already opened
i_disable_precharge	1	Input	Disables precharge, keep open for next R/W
i_precharge_req	1	Input	Request for precharge
i_data	[CPU_DATA_WIDTH-1:0]	Input	Input data to the SDRAM Controller
i_power_down	1	Input	Enables power down mode if high
i_disable_autorefresh	1	Input	Disables auto refresh
o_data	[CPU_DATA_WIDTH-1:0]	Output	Output data from the SDRAM Controller
o_write_done	1	Output	When High, indicates that write to SDRAM is complete
o_read_done	1	Output	When High, indicates that read from SDRAM is complete
o_data_valid	1	Output	Output data valid, can be used for FIFO Write Enable
o_data_req	1	Output	Input data request, can be used for FIFO read Enable
o_busy	1	Output	Active low busy signal which indicates SDRAM Controller is busy
o_init_done	1	Output	Indicates Initialization of SDRAM is completed
o_ack	1	Output	Indicates controller is about to start write, read, or load mode register operation
o_sdram_addr	[SDRAM_ADDR_WIDTH-1:0]	Output	SDRAM address
o_sdram_blkaddr	[SDRAM_BLKADR_WIDTH-1:0]	Output	SDRAM Bank Address
o_sdram_casn	1	Output	SDRAM column select
o_sdram_cke	1	Output	SDRAM Clock Enable
o_sdram_csn	1	Output	SDRAM Chip Select
o_sdram_dqm	[SDRAM_DQM_WIDTH-1:0]	Output	SDRAM Data Mask
o_sdram_rasn	1	Output	SDRAM row address select
o_sdram_wen	1	Output	SDRAM write enable
o_sdram_clk	1	Output	SDRAM clock
i_sdram_dq	[SDRAM_DATA_WIDTH-1:0]	Input	SDRAM input data
o_sdram_dq	[SDRAM_DATA_WIDTH-1:0]	Output	SDRAM output data

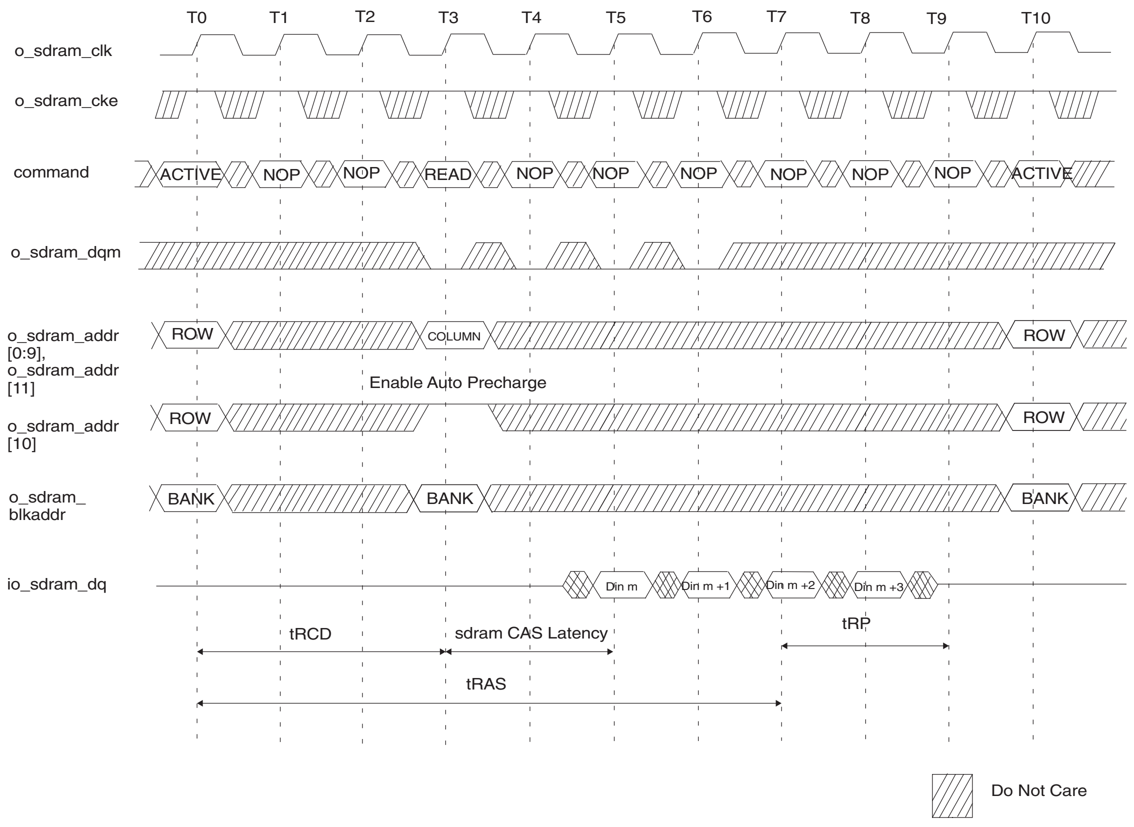
Initialization Conditions

An asynchronous active low reset signal assertion is necessary to initialize the SDR SDAM Controller to proper operating state.

Timing Diagram

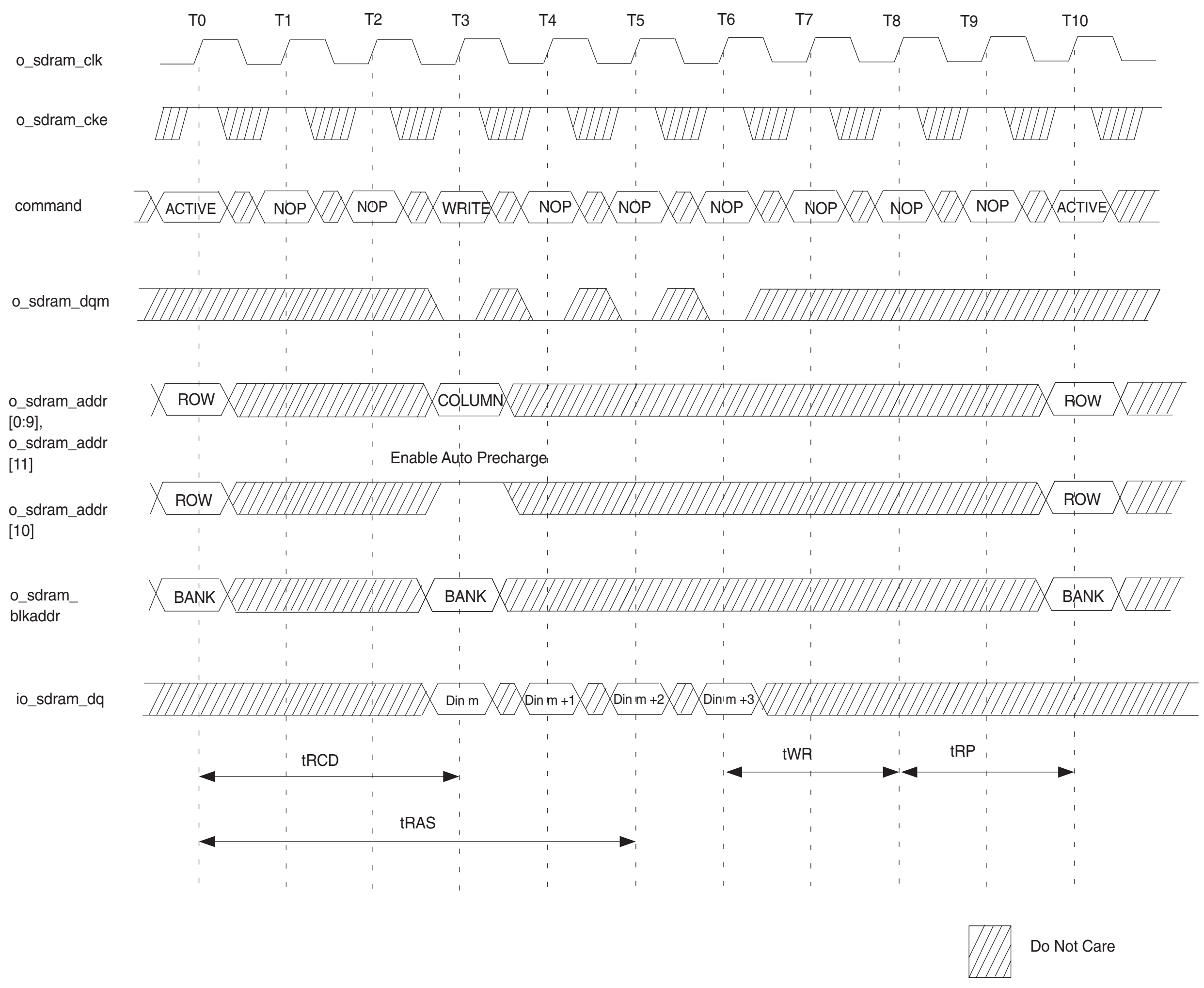
Read with auto precharge for Burst Length = 4 and CAS Latency = 2

Figure 5. Read with Auto Precharge



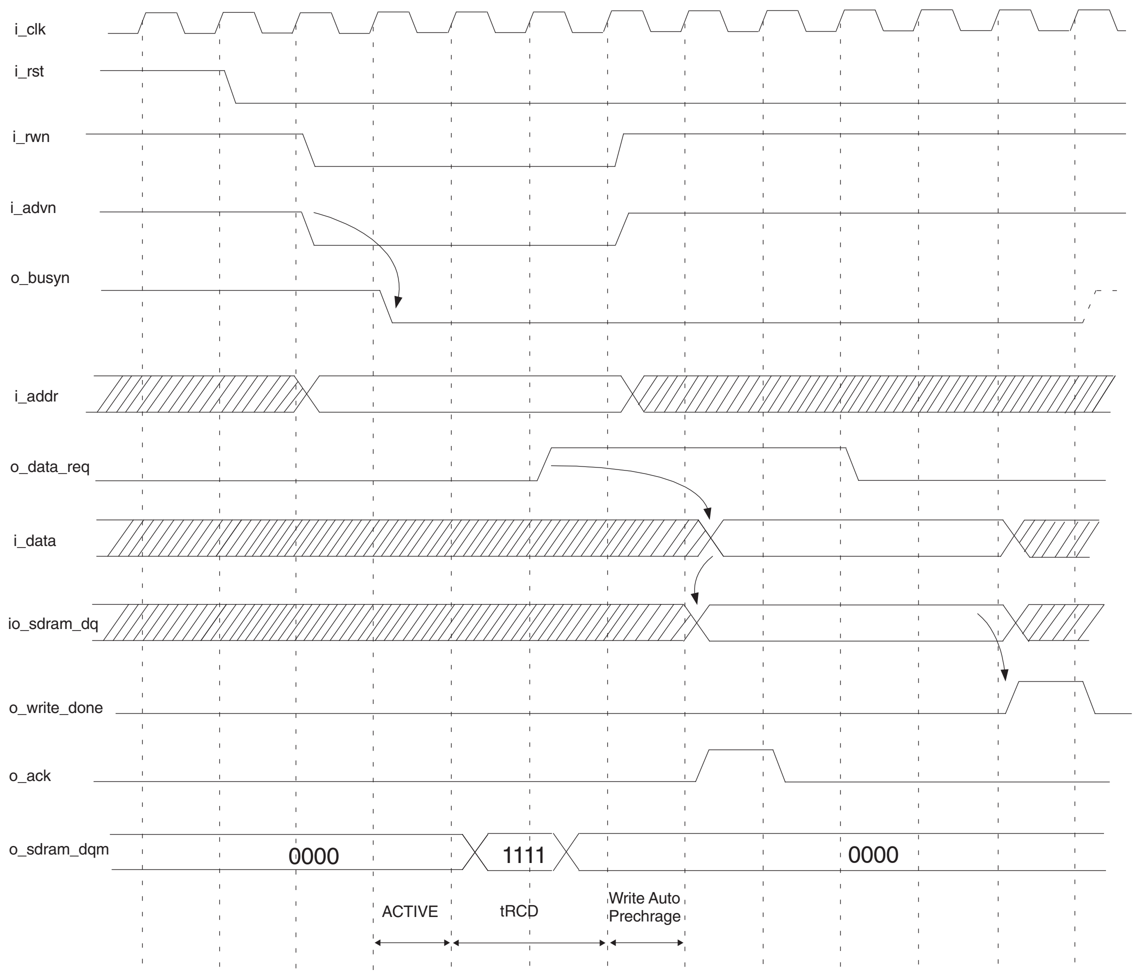
Write with auto precharge for Burst Length = 4

Figure 6. Write with Auto Precharge



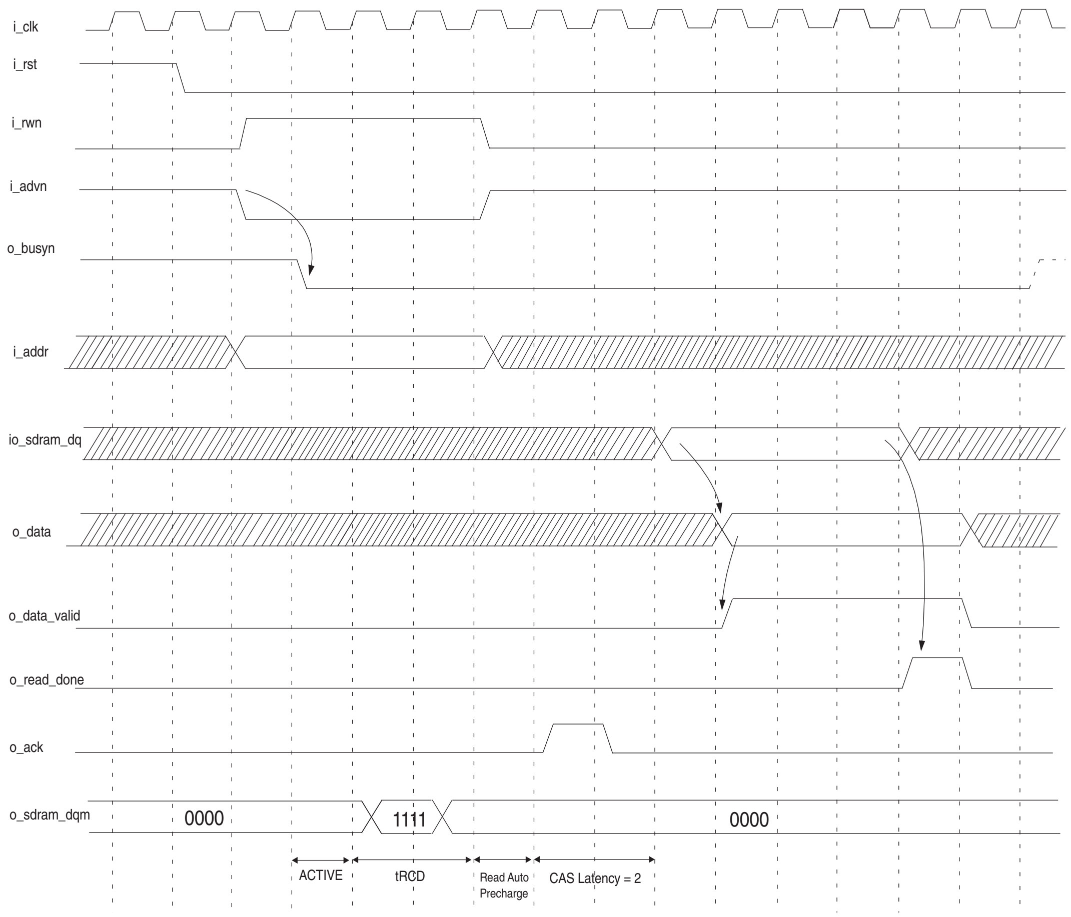
Processor Write Operation for Burst Length = 4

Figure 7. Processor Write Operation



Processor Read Operation for Burst Length =4 and CAS Latency=2

Figure 8. Processor Read Operation



Operation Sequence and Usage Examples

Usage Example#1

This usage example explains how to use the SDR SDRAM Controller for Micron's MT48LC8M16A2 for Burst length = 4 and CAS Latency = 2 cycles (for read). (These must be set in sdram_defines.v file)

1. The Table 3 below illustrates how to convert MT48LC8M16A2's timing values to SDRAM controller parameters in terms of 100 MHz clock frequency. These parameters must be set in the file sdram_defines.v.

Table 3. MT48LC8M16A2 timing values to SDRAM controller parameters

Parameter	Symbol	Value (Clock Cycles)
NUM_CLK_LOAD_MODEREG_DELAY	tMRD	2
NUM_CLK_PRECHARGE_PERIOD	tRP	2
NUM_CLK_AUTOREFRESH_PERIOD	tRFC	6
NUM_CLK_ACTIVE2RW_DELAY	tRCD	2
NUM_CLK_WRITE_RECOVERY_DELAY	tWR	2
DATAIN2ACTIVE	tDAL	5
LDMODEREG2ACTIVE	tMRD	2

Also set Mode Register specific parameters for the burst operation as shown in the table in the file sdram_defines.v.

Table 4. Mode Register specific parameters (MT48LC8M16A2)

Parameter	Value (in binary)
MODEREG_BURST_TYPE	0
MODEREG_OPERATION_MODE	00
MODEREG_WRITE_BURST_MODE	0

2. Use the System Designer to configure these parameters SDRAM Controller IP. Generate the necessary files from System Designer, synthesize and P&R the design using Synplify Pro and iCECube2 or Lattice Diamond.
3. Apply high on i_rst for few cycles to initialize the FSM to a known state and then de-assert this signal.
4. SDRAM controller initializes the FSM to configured parameters and o_init_done goes high after successful completion of initialization.
5. Write sequence for 4 bytes:

   a. Wait till the o_busyn signal to go low
   b. Keeping i_advn and i_rwn low, feed the SDRAM write address to the controller.
   c. Based on o_data_req feed the write data to the controller. Provide the input data through i_data, 2 cycles after o_data_req goes high
   d. o_ack goes high when the controller sends write command to SDRAM
   e. o_write_done will indicate completion of write operation

6. Read sequence for 4 bytes:

   a. Wait till the o_busyn signal to go low
   b. Keeping i_advn low and i_rwn high, feed the SDRAM read address to the controller.
   c. Use o_data_valid to sample the output o_data from the specified address of SDRAM.
   d. o_ack goes high when the controller sends write command to SDRAM
   e. o_read_done will indicate completion of write operation

Additional features

- Once controller is configured for a particular Burst Length and CAS latency in the initialization process, these parameters can be reconfigured by making i_loadmod_req high.
  a. Before loading the mode register, precharge all the banks by making i_precharge_req high.
  b. Wait for o_ack to go high and then, make i_precharge_req low.
  c. Assert i_loadmod_req and feed the input address line i_addr with the required Mode register configurations.

- By making i_selfrefresh_req high, controller enters to self refresh state where Self refresh command will be sent to SDRAM

- Precharging a row can be prevented by making use of active high i_disable_precharge. Then, data can be written to the same row but to different column by disabling opening that row by active high i_disable_active. For example,
  a. Assert i_disable_precharge - Open a row, but do not precharge
  b. Wait till the o_busyn signal to go low
  c. Keeping i_advn and i_rwn low, feed the SDRAM write address to the controller
  d. Assert i_disable_active – Do not open the same row
  e. De-assert i_disable_precharge – Close that row after data write operation

f. Keeping i_advn and i_rwn low, feed the SDRAM write address (same row address but different column address), write the data

g. De-assert i_disable_active

- Auto refresh count can be configured to have different Auto refresh counts – 500, 750, 1000, 1500 or 2000 depends on the configuration of the SDRAM

Example #1

For 128Mbit SDRAM, providing a AUTOREFRESH command every 15.625us will meet the refresh requirement and ensure that each row is refreshed. For 128Mbit part, refresh interval @100MHz = 100x10^6 x 15.625x10^-6 = 1562.5, so choose auto refresh count as 1500(Minimum count).

Example #2

For 256Mbit SDRAM, providing a AUTOREFRESH command every 7.8125us will meet the refresh requirement and ensure that each row is refreshed. For 256Mbit part, refresh interval @100MHz = 100x10^6 x 7.8125x10^-6 = 781.25, so choose auto refresh count as 750(Minimum count).

Usage Example#2

This usage example explains how to use the SDR SDRAM Controller for ISSI's IS42VM32200G for Page mode and CAS Latency = 3 cycles (for read). (These can be set in sdram_defines.v file)

1. The Table 5 below illustrates how to convert S42VM32200G's timing values to SDRAM controller parameters in terms of 100 MHz clock frequency. These parameters have to be set in the file sdram_defines.v

Table 5. S42VM32200G timing values to SDRAM controller parameters

Parameter	Symbol	Value (Clock Cycles)
NUM_CLK_LOAD_MODEREG_DELAY	tMRD	2
NUM_CLK_PRECHARGE_PERIOD	tRP	2
NUM_CLK_AUTOREFRESH_PERIOD	tRFC	7
NUM_CLK_ACTIVE2RW_DELAY	tRCD	2
NUM_CLK_WRITE_RECOVERY_DELAY	tWR	2
DATAIN2ACTIVE	tDAL	4
LDMODEREG2ACTIVE	tMRD	2

Also set Mode Register specific parameters for the burst operation as shown in the table in the file sdram_defines.v.

Table 6. Mode Register specific parameters (S42VM32200G)

Parameter	Value (in binary)
MODEREG_BURST_TYPE	0
MODEREG_OPERATION_MODE	00
MODEREG_WRITE_BURST_MODE	0

2. Use the System Designer to configure these parameters SDRAM Controller IP. Generate the necessary files from System Designer, synthesize and P&R the design using SynplifyPro and iCECube2 or Lattice Diamond.

3. Apply high on i_rst for few cycles to initialize the FSM to a known state and then de-assert this signal.

4. SDRAM controller initializes the FSM to configured parameters and o_init_done goes high after successful completion of initialization.

5. Write sequence for 4 bytes:

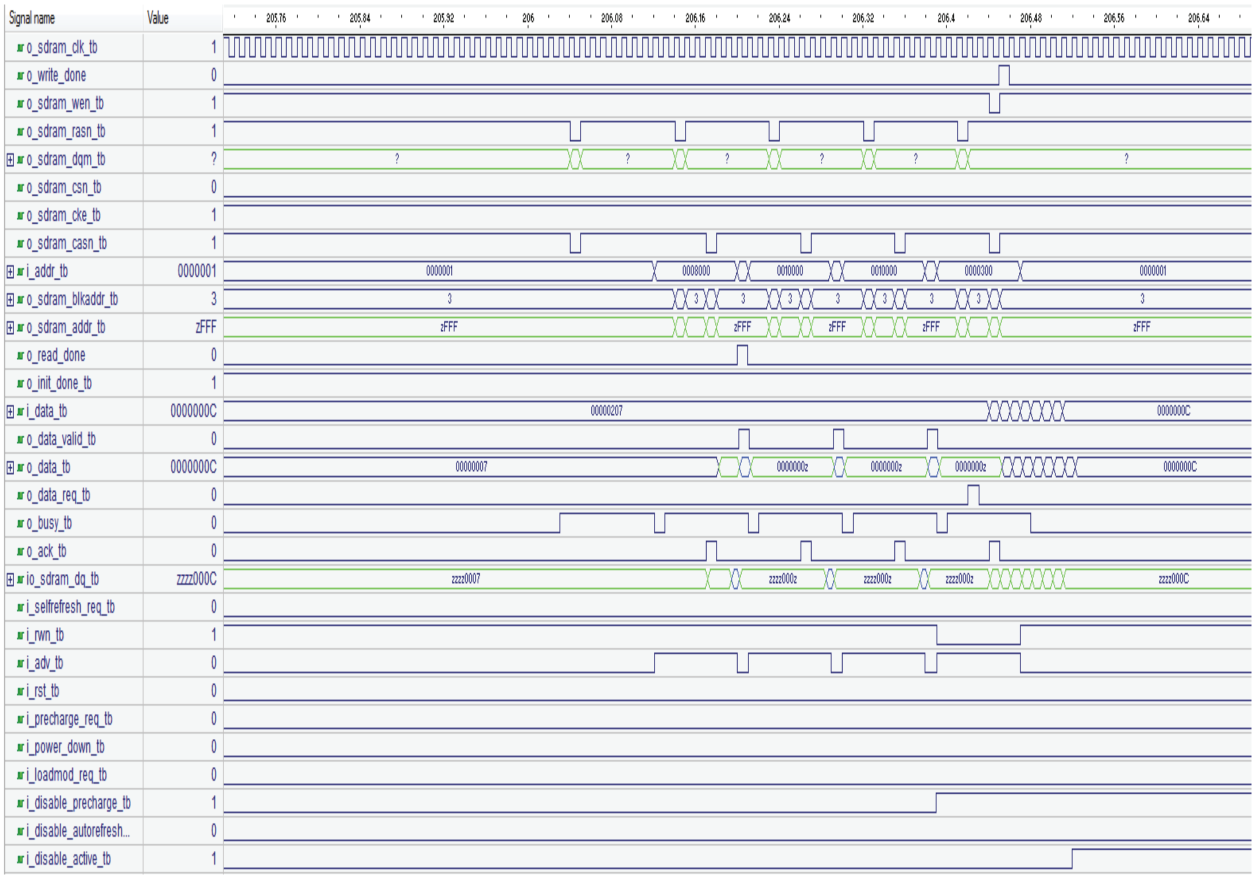
   a. Wait till the o_busyn signal to go low
   b. Keeping i_advn and i_rwn low, feed the SDRAM write address to the controller.
   c. Based on o_data_req feed the write data to the controller. Provide the input data through i_data, two cycles after o_data_req goes high
   d. o_ack goes high when the controller sends write command to SDRAM
   e. Assert i_burststop_req to stop the data write operation.
   f. o_write_done will indicate completion of write operation
   g. Then, SDRAM will be automatically precharged

6. Read sequence for 4 bytes:

   a. Wait till the o_busyn signal to go low
   b. Keeping i_advn low and i_rwn high, feed the SDRAM read address to the controller.
   c. o_ack goes high when the controller sends read command to SDRAM
   d. Use o_data_valid to sample the output o_data from the specified address of SDRAM.
   e. Assert i_burststop_req to stop the data write operation.
   f. For example If the page length = 255 then assert i_burststop_req at 253
   g. o_read_done will indicate completion of read operation
   h. Then, SDRAM will be automatically precharged

Simulation Waveforms

Figure 9. Simulation Waveforms



Implementation

This design is implemented in Verilog language. When using this design in a different device, density, speed or grade, performance and utilization may vary.

Table 7. Performance and Resource Utilization

Family	Language	Utilization (LUTs)	f_{MAX} (MHz)	I/Os	Architectural Resources
iCE40[1]	Verilog	227	>100	130	NA
MachXO3L [2]	Verilog-LSE	170	>100	130	NA
	Verilog-Syn	172	>100	130	NA

1. Performance and utilization characteristics are generated using iCE40LP8K-CM225 with iCEcube2 design software.
2. Performance and utilization characteristics are generated for Micron4m32 using LCMOX3L-4300C-5BG256C with Lattice Diamond 3.1 Design Software using Synplify and LSE (Lattice Synthesis Engine).

References

- iCE40 Family Handbook

Technical Support Assistance

e-mail: techsupport@latticesemi.com

Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2013	01.0	Initial release.
March 2014	01.1	Updated Table 7, Performance and Resource and Utilization. - Added support for MachXO3L device family. - Added support for Lattice Diamond 3.1 design software.
		Updated Technical Support Assistance information.