SDR SDRAM Controller White Paper

SDR SDRAM Controller Description

The Single Data Rate(SDR) Synchronous Dynamic Random Access Memory(SDRAM) Controller provides a simplified interface to industry standard SDR SDRAM memory. A top level system diagram of the SDR SDRAM Controller is shown in Figure 1. The SDRAM controller is available in either Verilog or VHDL that is optimized for the Altera APEX architecture. The SDR SDRAM controller supports the following features:

- SDRAM burst lengths of 1,2,4, or 8.
- CAS latency of 2 or 3.
- 16 bit programmable refresh counter used for automatic refresh.
- Support for 2 chip selects for SDRAM devices.
- Supports the following commands: NOP, READA, WRITEA, AUTO_REFRESH, PRECHARGE, ACTIVATE, BURST_STOP, and LOAD_MR.
- Supports full-page mode operation.
- Data Mask lines are supported for write operations.
- Utilizes a PLL to increase system performance.
- Supports data path widths of 16, 32, and 64 bits.

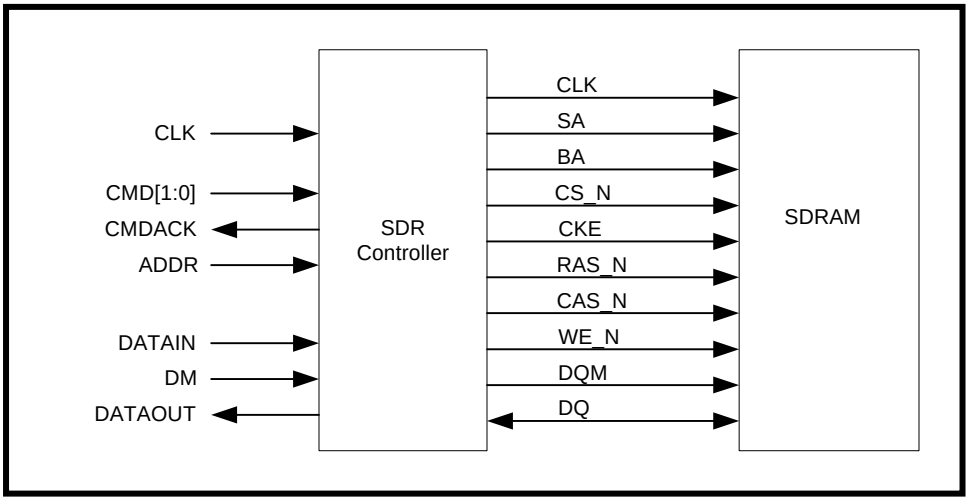

Figure 1 SDR SDRAM Controller System Level Diagram

SDRAM Overview

SDRAM is high-speed Dynamic Random Access Memory (DRAM) with a synchronous interface. The synchronous interface and fully pipelined internal architecture of SDRAM allows extremely fast data rates if used efficiently. Internally SDRAM devices are organized in banks of memory addressed by row and column. The number of row and column address bits and number of banks depends on the size of the memory.

SDRAM is controlled by bus commands that are formed using combinations of the RASN, CASN, and WEN 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.

Command	RASN	CASN	WEN
No Operation (NOP)	H	H	H
Active (ACT)	L	H	H
Read (RD)	H	L	H
Write (WR)	H	L	L
Burst Terminate (BT)	H	H	L
Precharge (PCH)	L	H	L
Autorefresh (ARF)	L	L	H
Load Mode Register (LMR)	L	L	L

Table 1 SDRAM Bus Commands

SDRAM 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 (ACT) command. When a bank is accessed for a read or a write it may be necessary to close the bank and re-open it if the row to be accessed is different than the row that is currently opened. Closing a bank is done with the Precharge (PCH) command.

The primary commands used to access SDRAM are Read (RD) and Write (WR). When the WR command is issued, the initial column address and data word is registered. When a RD command is issued, the initial address is registered. The initial data appears on the data bus 1-3 clock cycles later. This is known as CAS latency and is due to the time required to physically read the internal DRAM core and register the data on the bus. The CAS latency depends on the speed 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 RD or WR command, sequential read and writes will continue until the burst length is reached or a Burst Terminate (BT) command is issued. SDRAM memory devices support burst lengths of 1,2,4, or 8 data cycles. The Auto Refresh command (ARF) is issued periodically to insure data retention. This is function is performed by the SDRAM Controller and is transparent to the user.

The Load Mode Register command (LMR) is used to configure the SDRAM mode register. This register stores the CAS latency, burst length, burst type, and write burst mode. Consult the SDRAM specification for additional details.

SDRAM comes in Dual In-line Memory Modules (DIMM), Small Outline DIMMs (SO-DIMM) and as chips. To reduce pin count SDRAM row and column addresses are multiplexed on the same pins. SDRAM often includes more than one bank of memory internally and DIMMS may require multiple chip selects.

SDRAM Controller Description

SDRAM Controller Interface Signals

The SDRAM Controller interface signals are shown in Table 2. All signals are synchronous to the system clock and outputs are registered at the output of the SDRAM Controller Core.

Signal	Name	Active	I/O	Description
CLK	Clock	NA	Input	System Clock.
RESET_N	Reset	LOW	Input	System Reset.
ADDR[ASIZE-1:0]	Memory Address	NA	Input	Memory address for read/write requests. Width is set by ASIZE.
CMD[2:0]	Command	NA	Input	Command request.
CMDACK	Command Acknowledge	HIGH	Output	Acknowledgment of the requested command.
DATAIN[DSIZE-1:0]	Input Data	NA	Input	Input Data Bus. Width is set by DSIZE.
DATAOUT[DSIZE-1:0]	Output Data	NA	Output	Output Data Bus. Width is set by DSIZE.
DM[(DSIZE/8)-1:0]	Data Mask	HIGH	Input	Masks individual bytes during data write
SA[11:0]	Address Bus	NA	Output	SA[11:0] are sampled during the ACT command to latch the row address. SA[n:0] are sampled during the RD/WR command to latch the column address where n depends on the size of SDRAM used. SA[10] is sampled during the PCH command to determine if all banks are to be precharged or the bank selected by BA[1:0]. The address outputs also provide the op-code during the LMR command.
BA[1:0]	Bank Address	NA	Output	These signals determine to which bank the ACT, RD, WR, or PCH command is being applied.
CS_N[1:0]	Chip Selects	LOW	Output	SDRAM chip selects.
CKE	Clock Enable	HIGH	Output	SDRAM CKE input.
RAS_N	Row Address Strobe	LOW	Output	SDRAM Command input.

CAS_N	Column Address Strobe	LOW	Output	SDRAM Command input.
WE_N	Write Enable	LOW	Output	SDRAM Command input.
DQ[DSIZE-1:0]	Data Bus	NA	I/O	SDRAM Data Bus.
DQM[(DSIZE/8)-1:0]	Data Mask	HIGH	Output	SDRAM Data Masks, mask individual bytes during data write.

Table 2 Interface Signals

SDRAM Controller Command Interface

The SDR SDRAM Controller provides a synchronous command interface to the SDRAM along with several control registers. The commands are listed in Table 3 with detailed descriptions in following sections. Commands are issued to the controller as follows:

- A command, other than NOP, is driven, by the user, onto CMD[2:0] with ADDR and DATAIN set appropriately for the requested command. The controller registers the command on the next rising clock edge.
- To acknowledge the command the controller will assert CMDACK for one clock period.
- If the command was READA or WRITEA, the user will need to start receiving or writing data on the DATAOUT and DATAIN ports.
- The user must drive NOP onto CMD[2:0] by the next rising edge of CLK after CMDACK has been asserted.

Command	Value	Description
NOP	000b	No Operation.
READA	001b	SDRAM Read with Auto precharge.
WRITEA	010b	SDRAM Write with Auto precharge.
REFRESH	011b	SDRAM Auto refresh Refresh
PRECHARGE	100b	SDRAM Precharge all banks.
LOAD_MODE	101b	SDRAM Load Mode Register.
LOAD_REG1	110b	Load controller configuration register.
LOAD_REG2	111b	Load controller refresh period register.

Table** Error! No text of specified style in document. **Interface Commands

NOP Command

The NOP is a "No Operation" command to the controller. When detected by the controller, it performs a NOP in the following clock cycle. A NOP must be issued the following clock cycle after the controller has acknowledged a command. The NOP command has no effect on SDRAM accesses that are already in progress.

READA Command

The READA command instructs the controller to perform a burst read with auto-precharge to the SDRAM at the memory address specified by ADDR. The controller will issue an ACTIVATE command to the SDRAM followed by a READA command. The read burst data will first appear on DATAOUT (RC + CL + 2) after the controller has asserted command acknowledge(CMDACK). During a READA command the user must keep the DM inputs low. When the controller is configured for full-page mode, the READA command becomes READ(READ without auto-precharge). An example timing diagram for a READA command is shown in Figure 2. The general operation of a READA command is as follows:

- The user asserts READA along with ADDR and DM.
- The controller acknowledges the command by asserting CMDACK.
- Simultaneously with asserting CMDACK the controller starts issuing commands to the SDRAM devices.
- One clock after CMDACK was asserted, the user must assert NOP CMD.
- Following assertion of CMDACK by the controller, the controller presents the first read burst value on DATAOUT, with the remainder of the burst following every clock there after.

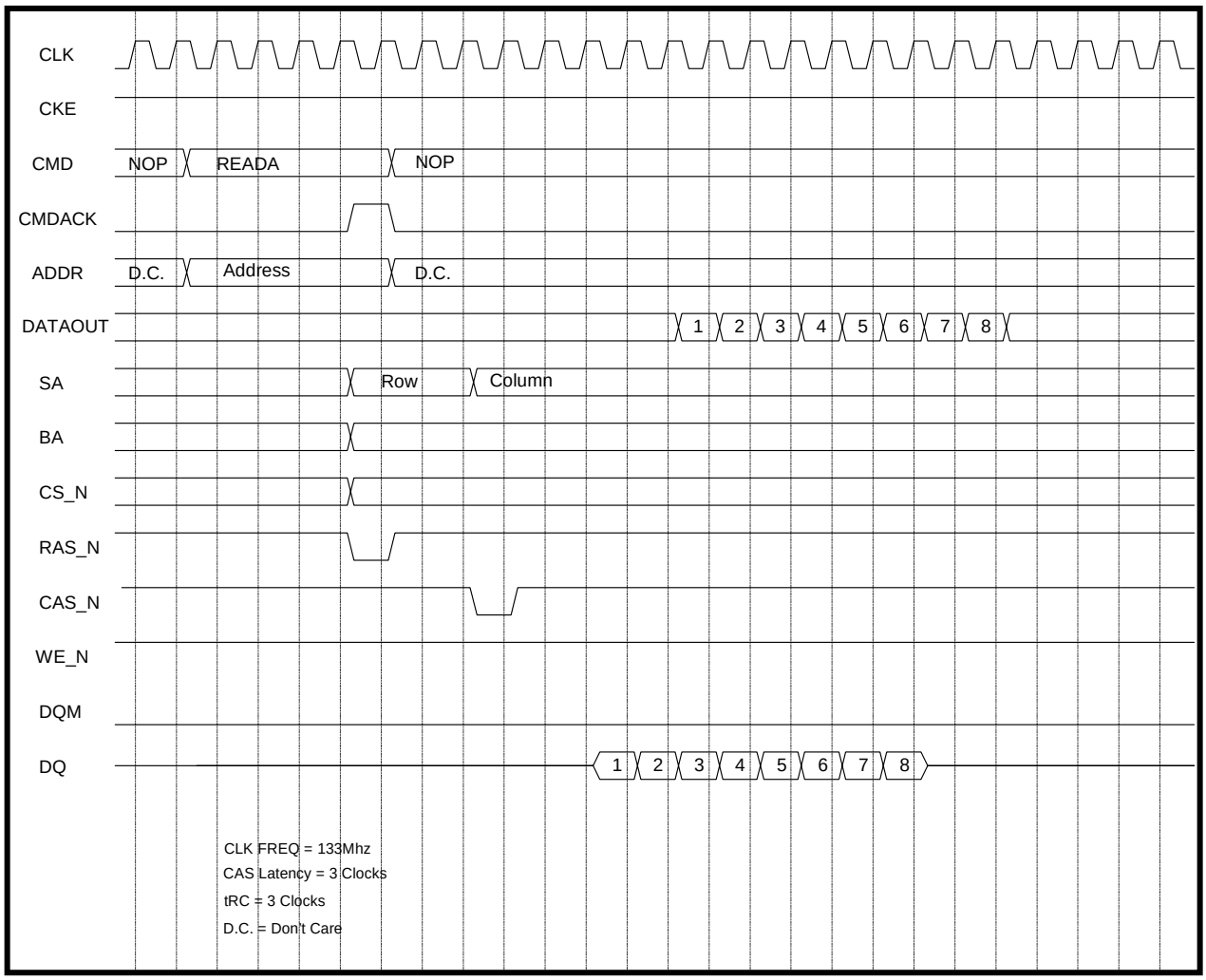

Figure 2 READA Timing Diagram

WRITEA Command

The WRITEA command instructs the controller to perform a burst write with auto-precharge to the SDRAM at the memory address specified by ADDR. The controller will issue an ACTIVATE command to the SDRAM followed by a WRITEA command. The first data value in the burst sequence must be presented with the WRITEA and ADDR address. The host must start clocking data along with the desired DM values into the controller tRC-2 clocks after the controller has acknowledged the WRITEA command. See a SDRAM data sheet for how to use the Data Mask lines DM/DQM. When the controller is in the full-page mode WRITEA becomes WRITE(write without auto-precharge). An example timing diagram for a WRITEA command is shown in Figure 3. The general operation of a WRITEA command is as follows:

- The user asserts WRITEA along with ADDR and the first write data value on DATAIN and the desired data mask value on DM.

- The controller acknowledges the command by asserting CMDACK.
- Simultaneously with asserting CMDACK the controller starts issuing commands to the SDRAM devices.
- One clock after CMDACK was asserted, the user asserts NOP on CMD.
- Following the assertion of CMDACK by the controller, the user clocks data and data mask values into the controller through DATAIN and DM.

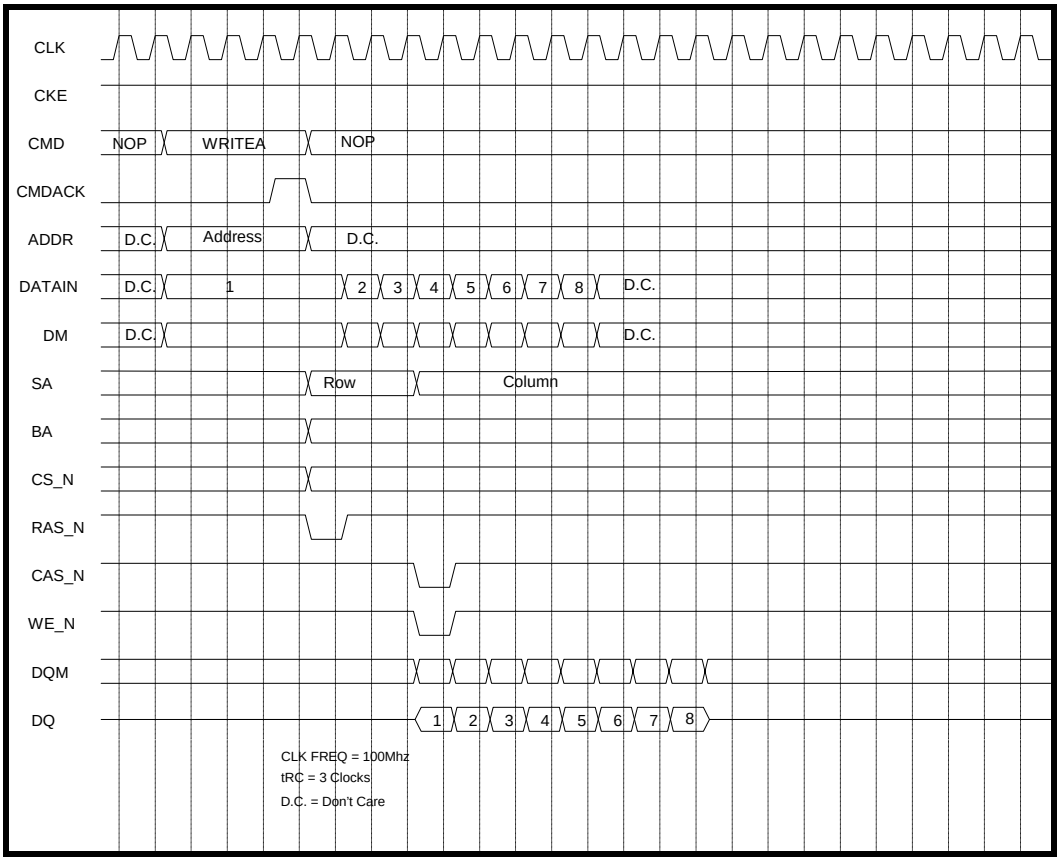

Figure 3 WRITEA Timing Diagram

REFRESH Command

The REFRESH command instructs the controller to perform a AUTO REFRESH command to the SDRAM. The controller will acknowledge the REFRESH command with CMDACK An example timing diagram of the REFRESH command is shown in Figure 4. The general operation of a REFRESH command is as follows:

- The user asserts REFRESH on CMD input.
- The controller acknowledges the command by asserting CMDACK.
- Simultaneously with asserting CMDACK the controller starts issuing commands to the SDRAM devices.
- Immediately following the assertion of CMDACK by the controller, the user asserts NOP on CMD.

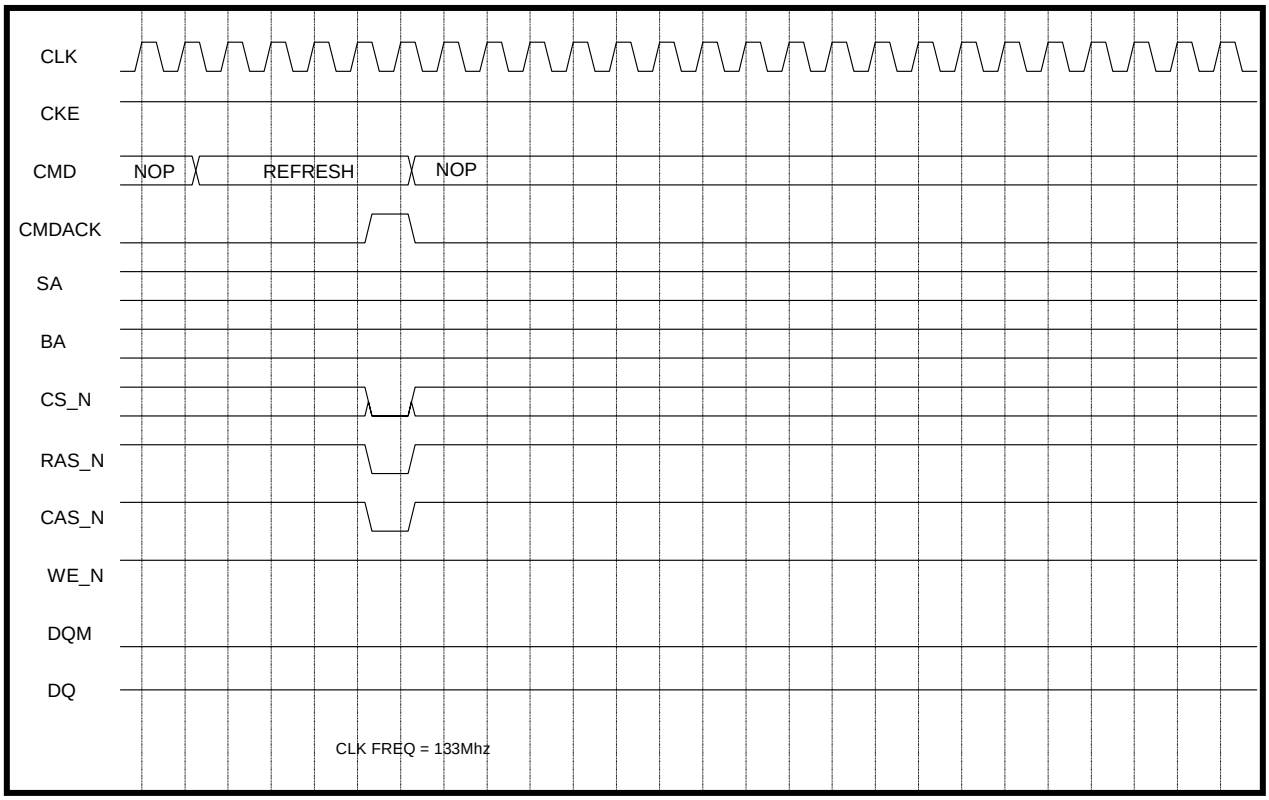

Figure 4 REFRESH Timing Diagram

PRECHARGE Command

The PRECHARGE command instructs the controller to perform a PRECHARGE command to the SDRAM. The controller will acknowledge the command with CMDACK. The PRECHARGE command is also used to generate a burst stop to the SDRAM. Using PRECHARGE to terminate a burst is only supported in the Full-Page Mode. It is important to note that the controller has adds a latency from when the host issues a command to when the SDRAM sees the PRECHARGE command of 4 clocks. This means that if a full-page read burst is to be stopped after 100 cycles, the PRECHARGE command must be asserted 4 + CL – 1 clocks before the desired end of the burst(CL – 1 requirement is imposed by the SDRAM devices). So if the CAS latency is 3, the PRECHARGE COMMAND must be issued 100 – 3 – 1 – 4 = 92 clocks into the burst. An example timing diagram of the PRECHARGE command is shown in Figure 5. The general operation of a PRECHARGE command is as follows:

- The user asserts PRECHARGE on CMD.
- The controller acknowledges the command by asserting CMDACK.
- Simultaneously with asserting CMDACK, the controller starts issuing commands to the SDRAM devices.
- Immediately after the controller asserts CMDACK, the user asserts NOP on CMD.

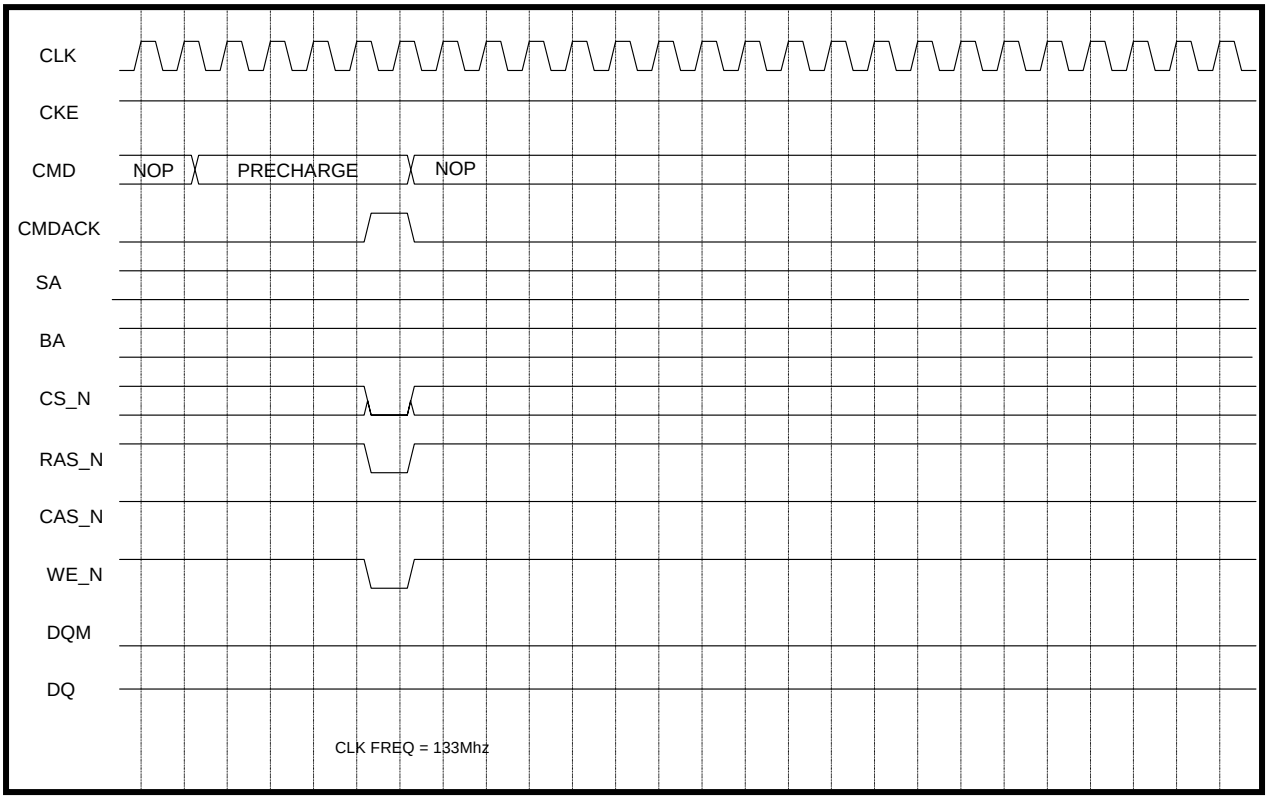

Figure 5 PRECHARGE Timing Diagram

LOAD_MODE Command

The LOAD_MODE command instructs the controller to perform a LOAD MODE REGISTER command to the SDRAM. The value that is to be written into the SDRAM mode register must be present on ADDR[11:0] with the LOAD_MODE command. The value on ADDR[11:0] is mapped directly to the SDRAM pins A11-A0 when the controller issued the LOAD MODE REGISTER to the SDRAM. An example timing diagram is shown in Figure 6. The general operation of a LOAD_MODE command is as follows:

- The users asserts LOAD_MODE on CMD.
- The controller acknowledges the command by asserting CMDACK.
- Simultaneously with asserting CMDACK the controller starts issuing commands to the SDRAM devices.
- One clock after the controller asserts CMDACK, the users asserts NOP on CMD.

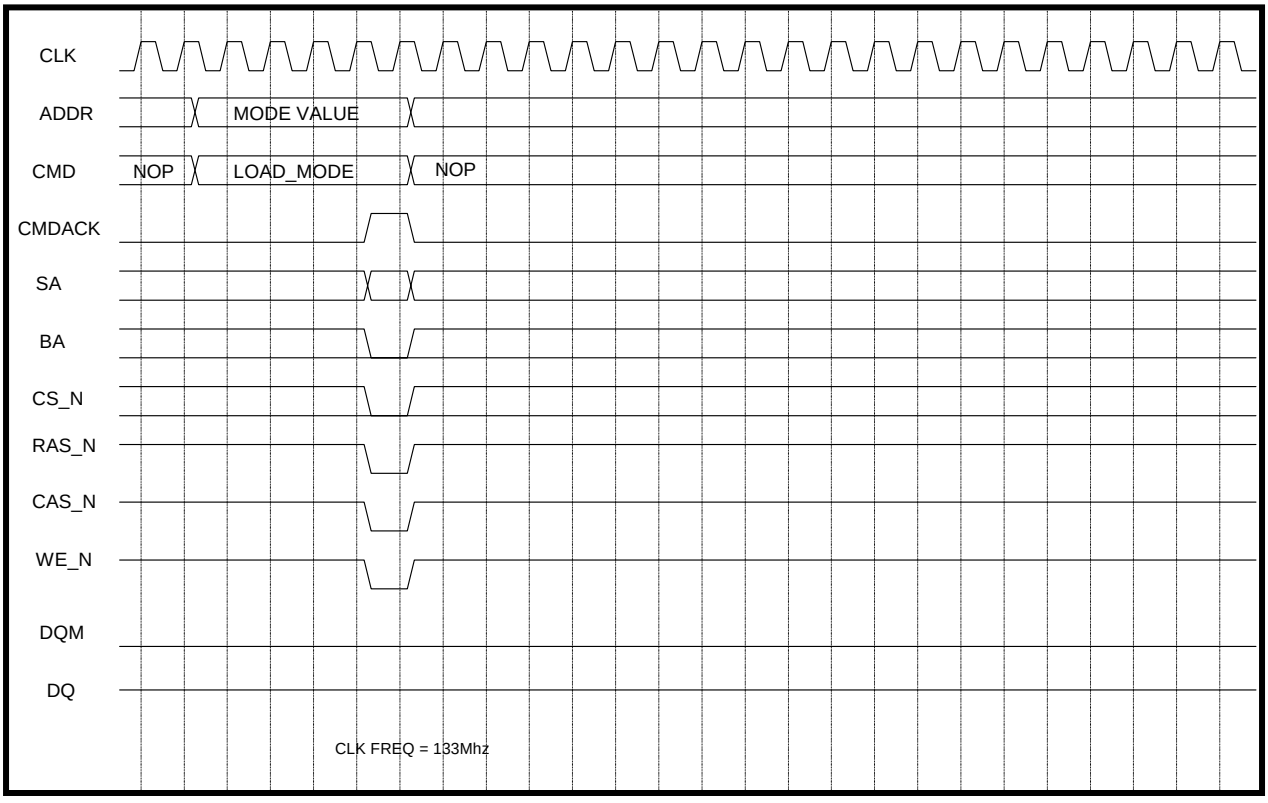

Figure 6 LOAD_MODE Timing Diagram

LOAD_REG1 Command

The LOAD_REG1 command instructs the controller to load the internal configuration register REG1. The value that is to be written into REG1 must be presented on the ADDR input simultaneously with the assertion of the command LOAD_REG1. The bit assignments for REG1 are shown in Table 4.

CL is the cas latency of the SDRAM memory in clock periods and is dependent on the memory device speed grade and clock frequency. Consult the SDRAM data sheet for appropriate settings. CL must be set to the same value as CL for the SDRAM memory devices.

RC is the RAS to CAS delay in clock periods and is dependent on the SDRAM speed grade and clock frequency. The formula is RC= INT(tRC/ clock_period). Where tRC is the value from the SDRAM data sheet and clock_period is the clock period of the clock that the controller and SDRAM will be running off of.

RRD is the Refresh to RAS delay in clock periods. RRD is dependent on the SDRAM speed grade and clock frequency. The formula is RRD= INT(tRRD / clock_period). Where tRRD is the value from the SDRAM data sheet and clock_period is the clock period of the clock that the controller and SDRAM will be running off of.

PM is the page mode bit. If PM=0 then the controller operates in non-page mode. If PM=1 then the controller operates in page-mode. See Section 0 for a complete description.

BL is the burst length the SDRAM devices have been configured for.

Label	ADDR Bits	Description
CL	[1:0]	Cas Latency setting for the controller.
RC	[3:2]	RAS to CAS delay in number of clocks.
RRD	[7:4]	REFRESH command duration in clocks.
PM	[8]	Controller mode, 0 = normal 1 = page mode
BL	[12:9]	Burst Length, valid values are 1,2,4,8

Table 4 REG1 Bit Definitions

LOAD_REG2 Command

The LOAD_REG2 command instructs the controller to load the internal configuration register REG2. REG2 is a 16 bit value that represents the period between REFRESH commands that the controller issues. The value is set by the equation int(REF_PERIOD / CLK_PERIOD). For example, if a SDRAM device connected to the controller has a 64ms, 4096 Cycle refresh requirement the device must have a REFRESH command issued to it at least every 64ms/4096 = 15.625 us. If the SDRAM and controller are clocked by a 100Mhz clock then the maximum value of REG2 should be 15.625us / .01us = 1562d. The value that is to be written into REG2 must be presented on the ADDR input simultaneously with the assertion of the command LOAD_REG2.

SDRAM Device Initialization and Controller Configuration

The SDRAM devices that are connected to the SDRAM controller must be initialized before they can be accessed. This initialization process set the burst length, cas latency, burst type, and operation mode for the SDRAM devices. After the SDRAM devices are initialized the SDRAM controller's configurations registers must be set. The procedure for initializing the SDRAM devices and the Controller is as follows:

- The user asserts the PRECHARGE command (see section 0).
- The user asserts a LOAD_MODE command (see section 0).
- The user asserts a LOAD_REG2 command (see section 0).
- The user asserts a LOAD_REG1 command (see section 0).

Architecture

The SDRAM Controller Core consists of four main modules, the SDRAM controller, Control Interface, Command, and Data Path Modules. The SDRAM controller module is the top level module that instantiates the three lower modules and brings the whole design together. The Control Interface Module accepts commands and related memory addresses from the host, decoding the command and passing the request to the Command module. The Command Module accepts command and address from the Control Interface Module, generating the proper commands to the SDRAM. The Data Path Module handles the data path operations during WRITEA and READA commands. The top level module also instantiates a PLL that is used in the CLOCK_LOCK mode to improve I/O timing. This PLL is not essential to the operation of the SDRAM Controller and can be easily removed.

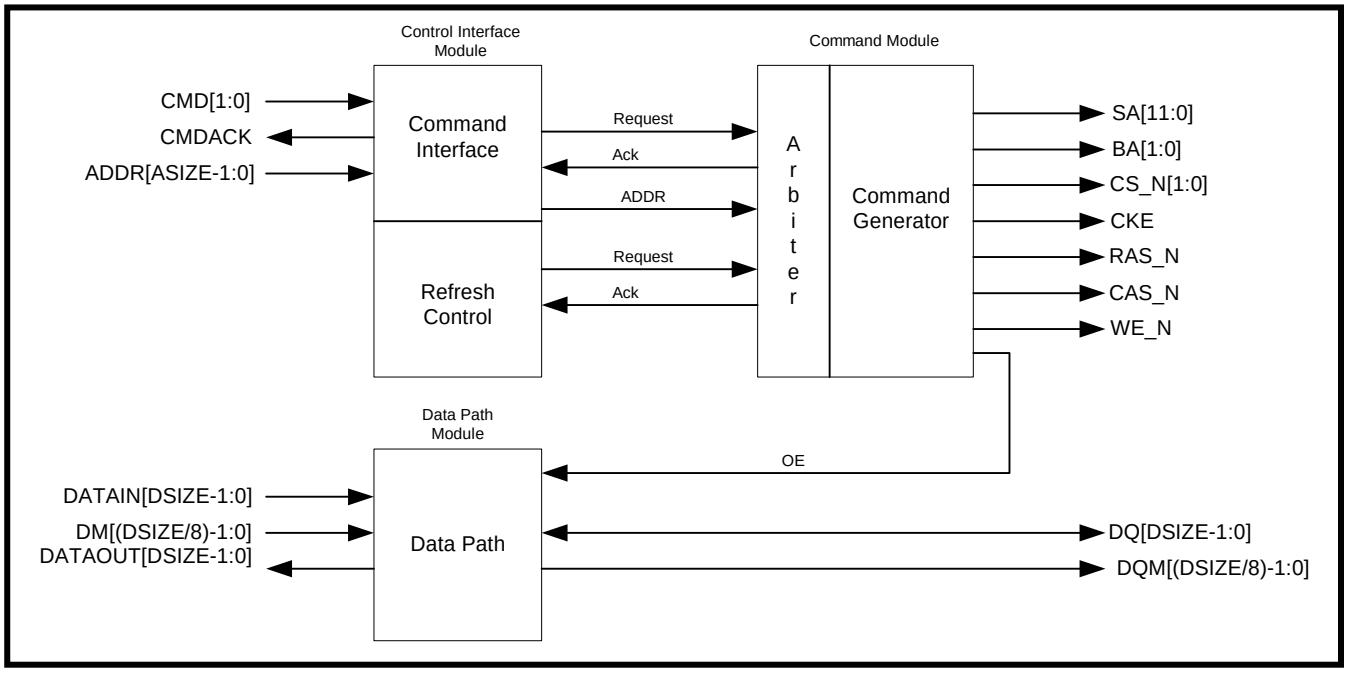

Figure 7 SDR SDRAM Controller Core Block Diagram

Control Interface Module

The control Interface Module decodes and registers commands from the host, passing the decoded NOP, WRITEA, READA, REFRESH, PRECHARGE, and LOAD_MODE commands, along with ADDR, to the Command Module. The LOAD_REG1 and LOAD_REG2 commands are decoded and used internally to load the REG1 and REG2 registers with values from ADDR. A block diagram of the Control Interface Module is shown in Figure 8.

The Control Interface Module also contains a 16 bit down counter and control circuit that is used to generate periodic refresh commands to the Command Module. The 16 bit down counter is loaded with the value from REG2 and counts down to zero. The REFRESH_REQ output is asserted with the counter reaches zero and remains asserted until the Command Module acknowledges the request. The acknowledge from the Command Module causes the down counter to be reloaded with REG2 and the process repeats. REG2 is a 16 bit value that represents the period between REFRESH commands that the controller issues. The value is set by the equation int(REF_PERIOD / CLK_PERIOD). For example, if a SDRAM device connected to the controller has a 64ms, 4096 Cycle refresh requirement the device must have a REFRESH command issued to it at least every 64ms/4096 = 15.625 us. If the SDRAM and controller are clocked by a 100Mhz clock then the maximum value of REG2 should be 15.625us / .01us = 1562d. Figure 8 shows the timing at the SDRAM interface for a REFRESH sequence.

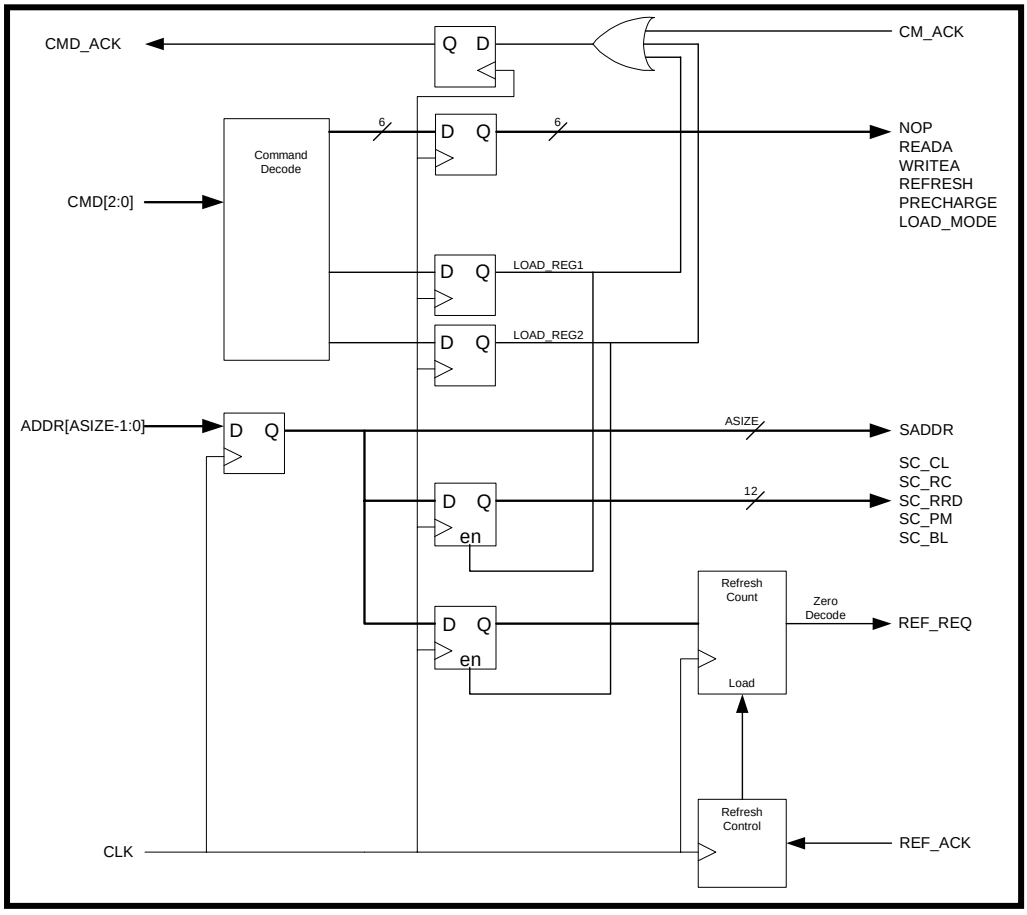

Figure 8 Control Interface Module Block Diagram

DATA Path Module

The Data Path Module provides the SDRAM data interface to the host. Host data is accepted on port DATAIN for WRITEA commands and data is provided to the host on port DATAOUT during READA commands. A block diagram of the Data Path Module is show in Figure 9. The DATAIN path consists of a 2 stage pipeline to align data properly relative to the CMDACK and the commands that are issued to the SDRAM. The DATAOUT path consists of a 2 stage pipeline that registers data from the SDRAM during a READA command. The DATAOUT path pipeline delay can be reduced to one or even zero registers, the only effect would be that the relationship of DATAOUT to CMDACK would change.

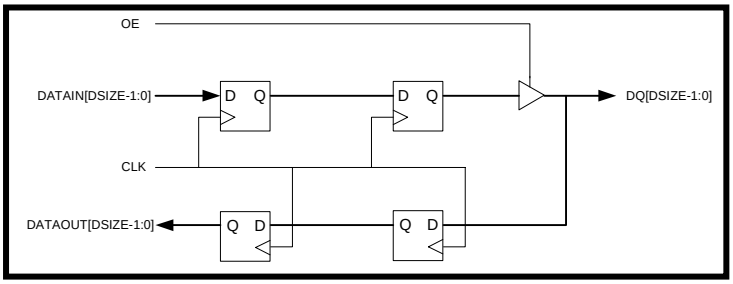

Figure 9 Data Path Module Block Diagram

Command Module

The Command Module accepts decoded commands from the Control Interface Module, along with refresh requests from the refresh control logic and generates the appropriate commands to the SDRAM. The module contains a simple arbiter that arbitrates between the commands from the host interface and the refresh requests from the refresh control logic. The refresh requests from the refresh control logic have priority over the commands from the host interface. If a command from the host arrives at the same time or during a "hidden" refresh operation, the arbiter holds off the host by not asserting CMDACK until the hidden refresh operation is complete. If a hidden refresh command is received while a host operation is in progress then the hidden refresh is held off until the host operation is complete. A block diagram of the Command Module is shown in Figure 10.

After the arbiter has accepted a command from the host, the command is passed onto the command generator portion of the Command Module. The command Module uses three shift registers to generate the appropriate timing between the commands that are issued to the SDRAM. One shift register is used to control the timing the ACTIVATE command while a second is used to control the positioning of the READA or WRITEA commands. A third is used to time command durations, in order for the arbiter to determine if the last requested operation has been completed.

The Command Module also performs the multiplexing of the address(ADDR) to the SDRAM. The row portion of the address is multiplexed out to the SDRAM outputs A[11:0] during the ACTIVATE(RAS) command. The column portion is then multiplexed out to the SDRAM address outputs during a READA(CAS) or WRITEA command.

The output signal OE is generated by the Command Module to control tristate buffers in the last stage of the DATAIN path in the Data Path Module.

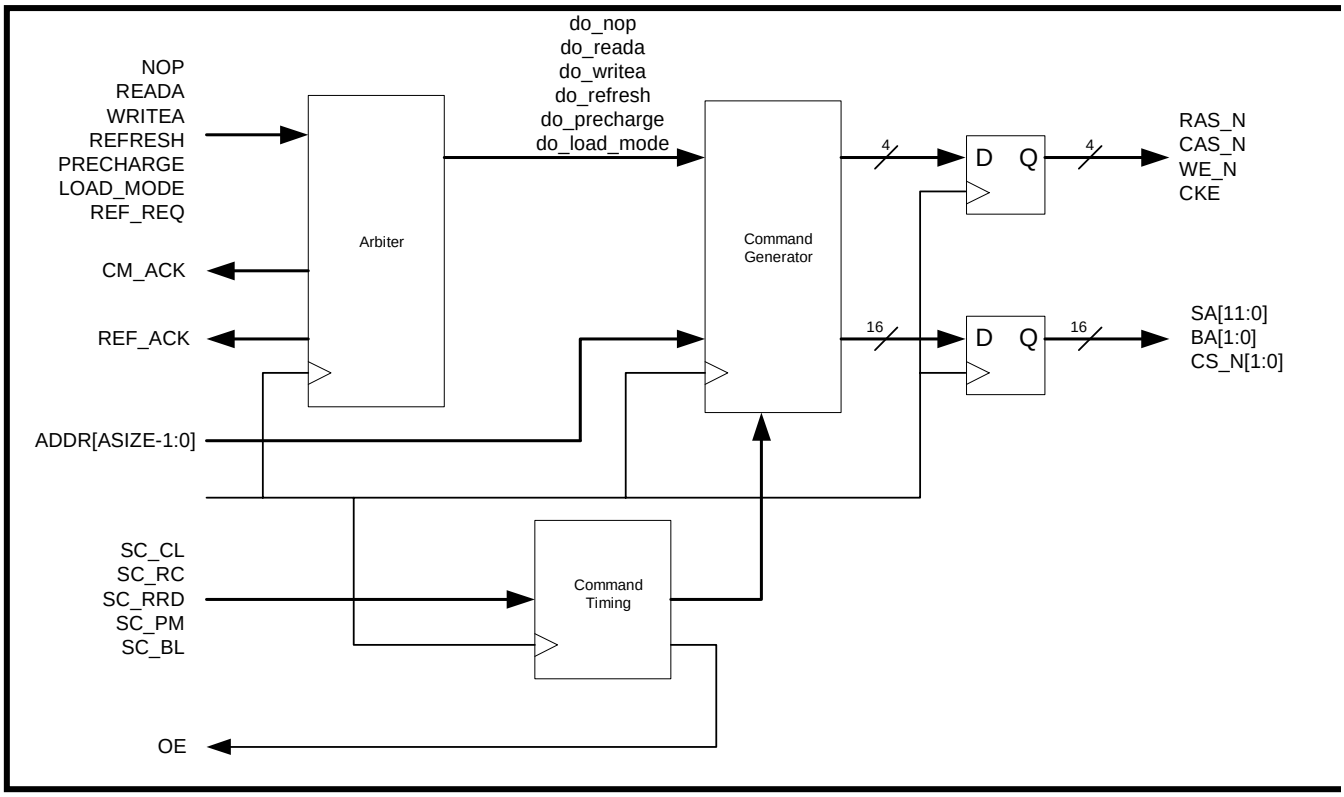

Figure 10 Command Module Block Diagram

Full-Page Mode Operation

The SDR SRAM Controller supports Full-Page Mode operation of the SDRAM devices that it controls. The controller is configured for Full-Page operation by setting bit 8 of REG 1(see section 0 for more information on setting REG1). When the SDRAM Controller is in Full-Page Mode, the Auto-Hidden Refresh function is disabled and READA and WRITEA commands become READ/WRITE(without autoprecharge). The user interface must handle refreshing the SDRAM devices via the REFRESH command. The user interface must also handle issuing appropriate PRECHARGE commands to the SDRAM devices to close a bank before starting a READ operation to a different row. A timing diagram for a Full-Page READ burst with a PRECHARGE to terminate the burst is shown in Figure 11. The Full-Page READ operation is as follows:

- The users asserts the READ Command on CMD along with ADDR.

- The controller acknowledges the command by asserting CMDACK.
- Simultaneously with asserting CMDACK the controller starts issuing commands to the SDRAM devices.
- One clock after the controller asserts CMDACK, the users asserts NOP on CMD.
- RC + CL + 2 clocks after the controller asserted CMDACK, the controller starts clocking the read data out on DATAOUT.
- CL – 1 + 7 clocks before the last READ data value is to appear on DATAOUT, the users asserts PRECHARGE on CMD.
- 4 clocks later, the controller acknowledges the PRECHARGE command by asserting CMDACK.
- Simultaneously with asserting CMDACK, the controller issues a PRECHARGE command to the SDRAM devices.
- CL – 1 + 2 clocks later, the READ burst terminates on the DATAOUT port.

A timing diagram for a Full-Page WRITE burst with a PRECHARGE to terminate the burst is shown Figure 11. The Full-Page WRITE operation is as follows:

- The users asserts WRITE on CMD along with ADDR.
- The controller acknowledges the command by asserting CMDACK.
- Simultaneously with asserting CMDACK the controller starts issuing commands to the SDRAM devices.
- One clock after the controller asserts CMDACK, the users asserts NOP on CMD.
- RC - 2 clocks after the controller asserts CMDACK, the user starts clocking data into the controller on DATAIN.
- 3 clocks before the last WRITE data value is to appear on DATAOUT, the users asserts PRECHARGE on CMD. This delay is based on 5 clk command delay – tWR(which is 2 in this example).
- 4 clocks later, the controller acknowledges the PRECHARGE command by asserting CMDACK.
- Simultaneously with asserting CMDACK, the controller issues a PRECHARGE command to the SDRAM devices.
- The Full-Page WRITE burst is then terminated.

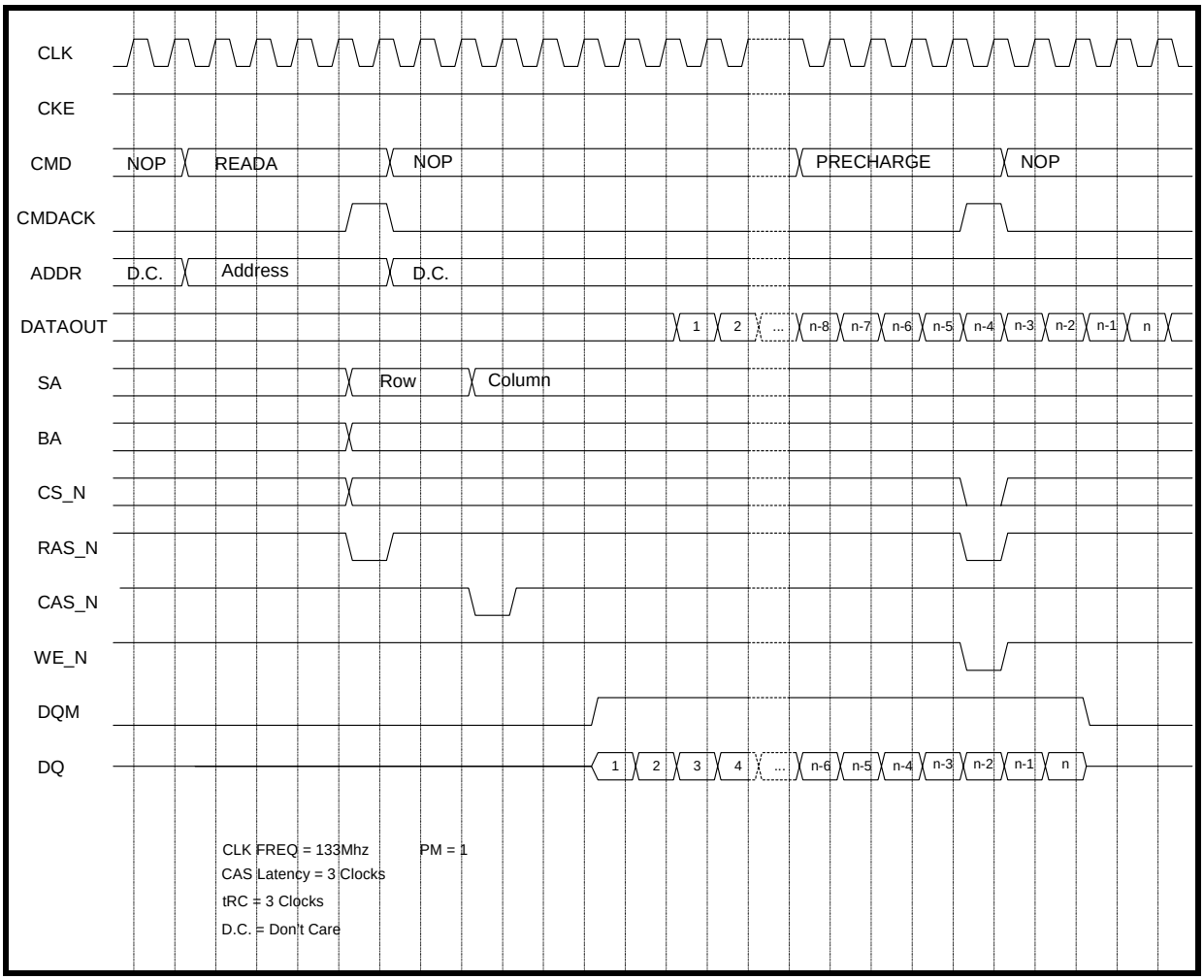

Figure 11 Full-Page READ Burst Timing Diagram

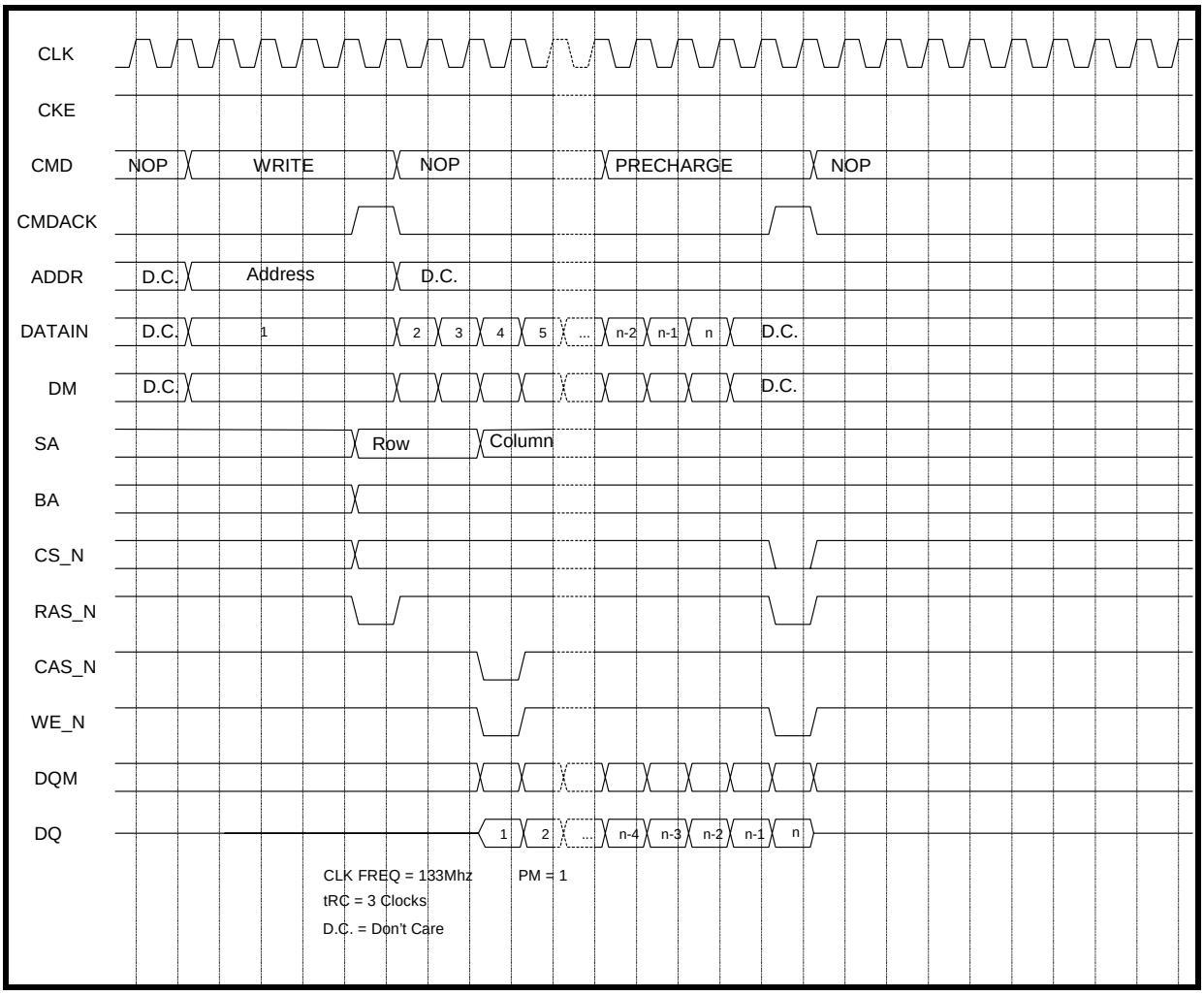

Figure 12 Full-Page WRITE Burst Timing Diagram

Simulation, Synthesis, Place and Route, and Results

The SDR SDRAM Controller includes source files, test bench, synthesis scripts, and support files for place and route using Quartus 2000.02 targeting a EP20K200EFC672-1X. The directory structure for this release is shown in Table 5. The various directories and files are discussed below.

Directory	Description
\source	Contains the source files for the SDR SDRAM controller.
\simulation	Contains the SDR SDRAM controller testbench.
\model	SDRAM memory model.
\synthesis\synplicity	Contains synplicity project file and synthesis results.
\synthesis\fpgaexpress	Contains FPGA express project file and synthesis results.
\route	Contains all files necessary to route the SDR SDRAM controller using Quartus 00.02
\doc	Documentation

Table 5 SDR SDRAM Controller release directory structure

Simulation

The SDR SDRAM Controller can be simulated by using the behavioral source file sdr_sdram_tb found in the simulation directory. The test bench instantiates the SDR Controller along with two SDRAM memory models. The test bench then configures the SDRAM devices and controller and runs a ramp pattern test using all combinations of CAS latency, burst length, and RAS-CAS delays for a given clock frequency. Higher clock frequencies(133mhz vs. 100mhz for example) limit the RAS-CAS delays and CAS latency choices.

Synthesis

The \synthesis\synplicity and \synthesis\fpgaexpress directories contain project files for synplicity and FPGA Express, respectively. These directories also contain the result files from synthesis runs with the SDR SDRAM controller. The only synthesis constraint is the setting of the global fmax to 133Mhz. The project file also sets the target device for synthesis to an EP20K200E-1X, BC672 package.

Place and Route

The place and route directory contains the files from a Quartus 2000.02 build of the SDR SDRAM Controller. The global FMAX constraint is inherited from the PLL that is instantiated by the controller and is set to 133Mhz. PC133 SDRAM devices typically have a Tco of 5.4ns and a Tsu of 1.5ns. In order to meet these timing requirements and provide for PCB wiring delays, the global setting "Automatic fast input registers=ON" which results in setup times on the FPGA inputs from the SDRAM devices of 1.2ns. A Tsu of 1.2ns at 133mhz leaves 7.5 – 1.2 – 5.4 = .9ns for board delay. Using the PLL results in Tco times of better than 4.8ns on the FPGA outputs to the SDRAM devices. At a clock rate of 133mhz and a SDRAM Tsu of 1.5ns, 7.5 – 4.8 – 1.5 = 1.5ns for PCB board delay. See Altera Application Note 75, High-Speed Board Designs, for more information on PCB board design for High-Speed applications.

Results

The post route performance results for the SDR SDRAM controller are shown in Table 6. Throughput is calculated using a 32 bit data path to SDRAM and a 20 clock READA cycle time for an 8 cycle SDRAM burst. READA cycle time in clocks represents the critical path as WRITEA cycle time is 15 clocks. The formula for calculating the throughput in Mbytes/sec is

1 / (clk period ns * number of clocks per access) * (data path width in Bytes) * (burst length of the access)

Device	Internal Fmax	Tco/Tsu	Throughput	Total LEs
20K200E-1X	133Mhz	4.8ns/1.2ns	212.8 Mbytes/sec	223
20K200-1X	100	6.6ns/1.6	160 Mbytes/sec	223

Table 6 Post Route Performance comparison