SDR SDRAM Controller

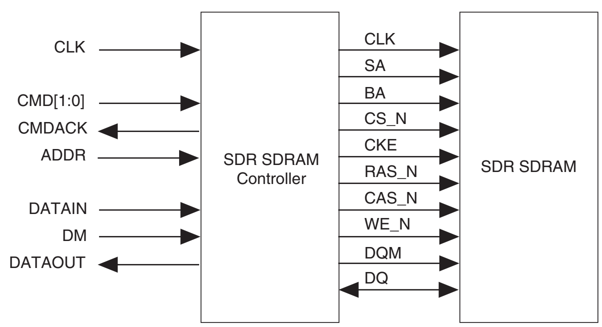
Introduction

The single data rate (SDR) synchronous dynamic random access memory (SDRAM) controller provides a simplified interface to industry standard SDR SDRAM. The SDR SDRAM Controller is available in either Verilog HDL or VHDL and is optimized for the Altera® APEX™ architecture. The SDR SDRAM Controller supports the following features:

- Burst lengths of 1, 2, 4, or 8 data words
- CAS latency of 2 or 3 clock cycles
- 16-bit programmable refresh counter used for automatic refresh
- 2-chip selects for SDRAM devices
- Supports the NOP, READA, WRITEA, AUTO_REFRESH, PRECHARGE, ACTIVATE, BURST_STOP, and LOAD_MR commands
- Support for full-page mode operation
- Data mask line for write operations
- PLL to increase system performance
- Support for data-path widths of 16, 32, and 64 bits

Figure 1 shows a system-level diagram of the SDR SDRAM Controller.

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, which are addressed by row and column. The number of row- and column-address bits and the 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. Table 1 shows the standard SDRAM bus commands.

Table 1. SDRAM Bus Commands

Command	Abbreviation	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

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 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 PCH command.

The primary commands used to access SDRAM are RD and 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 to 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 are required. After the initial RD or WR command, sequential read and writes continue until the burst length is reached or a BT command is issued. SDRAM memory devices support burst lengths of 1, 2, 4, or 8 data cycles. The ARF is issued periodically to ensure data retention. This function is performed by the SDR SDRAM Controller and is transparent to the user.

The LMR is used to configure the SDRAM mode register. which 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 (DIMMs), small-outline DIMMs (SO-DIMMs) and 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.

Functional Description

Table 2 shows the SDR SDRAM Controller interface signals. All signals are synchronous to the system clock and outputs are registered at the SDR SDRAM Controller's outputs.

Table 2. Interface Signals

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 pre-charged 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 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.

SDRAM Controller Command Interface

The SDR SDRAM Controller provides a synchronous command interface to the SDRAM and several control registers. Table 3 shows the commands, which are described in following sections. The following rules apply to the commands:

- All commands, except NOP, are driven by the user onto CMD[2:0]; ADDR and DATAIN are set appropriately for the requested command. The controller registers the command on the next rising clock edge
- To acknowledge the command the controller asserts `CMDACK` for one clock period
- For READA or WRITEA commands, the user should start receiving or writing data on `DATAOUT` and `DATAIN`
- The user must drive NOP onto `CMD[2:0]` by the next rising clock edge after `CMDACK` is asserted

Table 3. Interface Commands

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.
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.

NOP Command

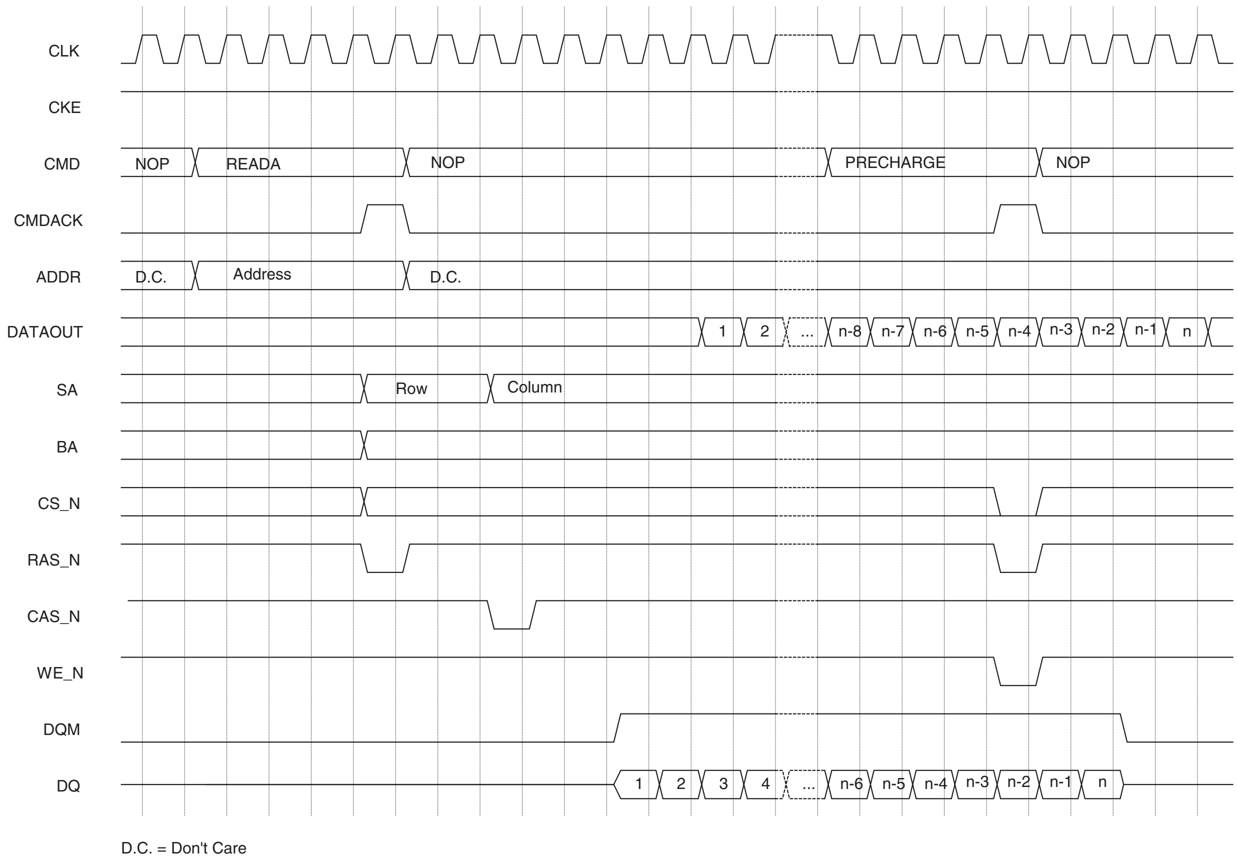
NOP is a no operation command to the controller. When NOP is detected by the controller, it performs a NOP in the following clock cycle. A NOP must be issued the following clock cycle after the controller has acknowledged a command. The NOP command has no affect on SDRAM accesses that are already in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

READA Command

The READA command instructs the SDR SDRAM Controller to perform a burst read with auto-precharge to the SDRAM at the memory address specified by ADDR. The SDR SDRAM Controller issues an ACTIVATE command to the SDRAM followed by a READA command. The read burst data first appears on `DATAOUT` (RCD + CL + 2) after the SDR SDRAM Controller asserts `CMDACK`. During a READA command the user must keep `DM` low. When the controller is configured for full-page mode, the READA command becomes READ (READ without auto-precharge). Figure 2 shows an example timing diagram for a READA command. The following sequence describes the general operation of the READA command:

- The user asserts READA, `ADDR` and `DM`
- The SDR SDRAM Controller asserts `CMDACK` to acknowledge the command and simultaneously starts issuing commands to the SDRAM devices
- One clock after `CMDACK` is asserted, the user must assert NOP
- The `CMDACK` presents the first read burst value on `DATAOUT`, the remainder of the read bursts follow every clock cycle

Figure 2. READA Timing Diagram



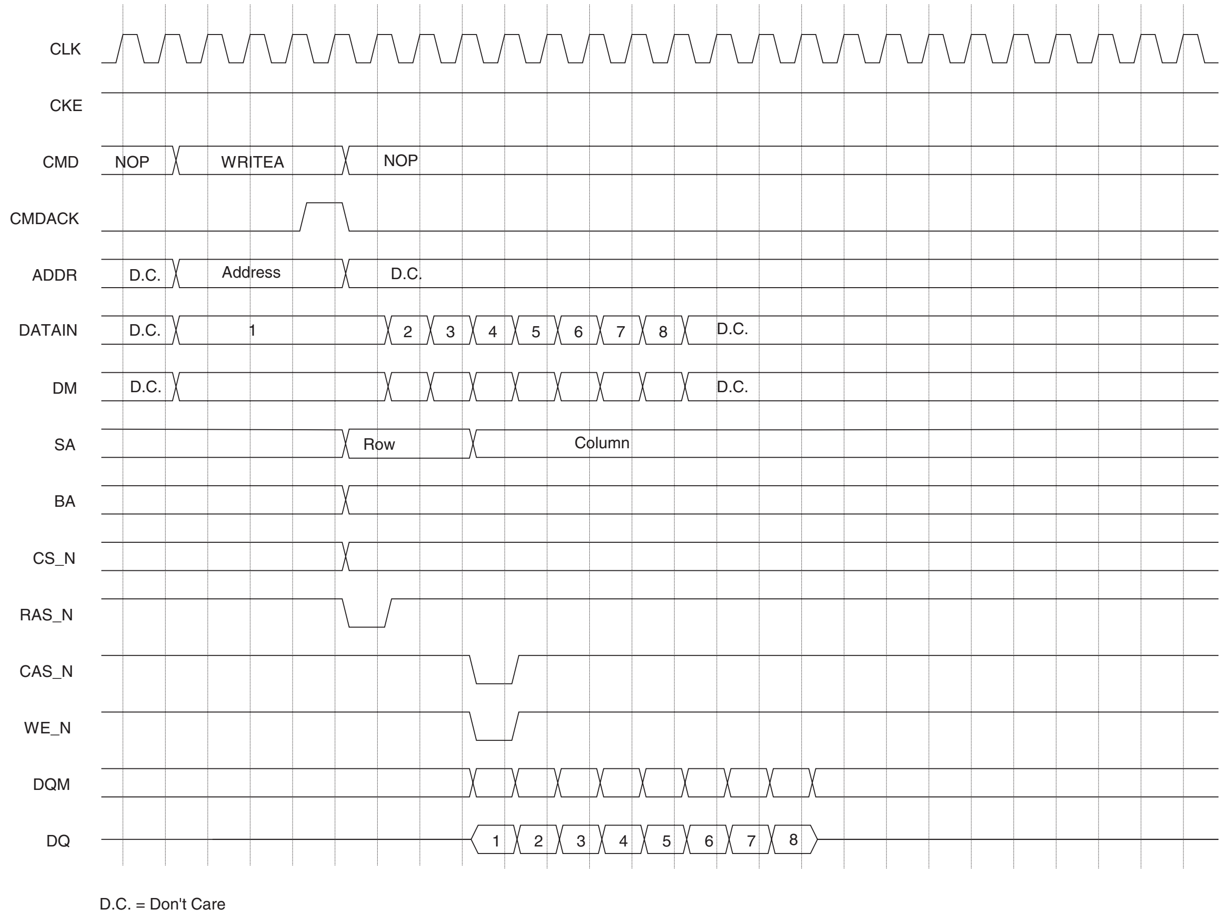
WRITEA Command

The WRITEA command instructs the SDR SDRAM Controller to perform a burst write with auto-precharge to the SDRAM at the memory address specified by ADDR. The SDR SDRAM 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 SDR SDRAM Controller (t_{RCD} – 2) clocks after the SDR SDRAM Controller has acknowledged the WRITEA command. See a SDRAM data sheet for how to use the data mask lines DM/DQM.

When the SDR SDRAM Controller is in the full-page mode WRITEA becomes WRITE (write without auto-precharge). Figure 3 shows an example timing diagram for a WRITEA command. The following sequence describes the general operation of a WRITEA command:

- The user asserts WRITEA, ADDR, the first write data value on DATAIN, and the desired data mask value on DM
- The SDR SDRAM Controller asserts CMDACK to acknowledge the command and simultaneously starts issuing commands to the SDRAM devices
- One clock after CMDACK was asserted, the user asserts NOP on CMD
- The user clocks data and data mask values into the SDR SDRAM Controller through DATAIN and DM

Figure 3. WRITEA Timing Diagram

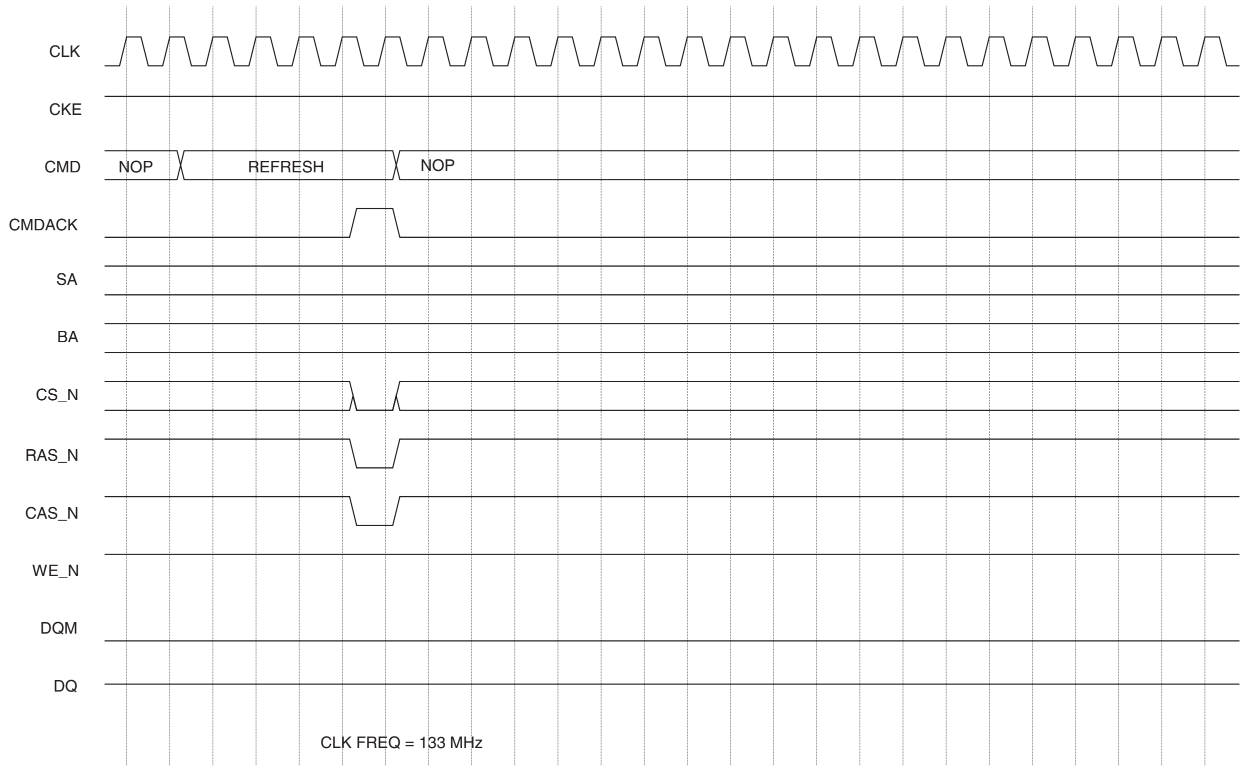


REFRESH Command

The REFRESH command instructs the SDR SDRAM Controller to perform an ARF command to the SDRAM. The SDR SDRAM Controller acknowledges the REFRESH command with `CMDACK`. Figure 4 shows an example timing diagram of the REFRESH command. The following sequence describes the general operation of a REFRESH command:

- The user asserts REFRESH on the `CMD` input
- The SDR SDRAM Controller asserts `CMDACK` to acknowledge the command and simultaneously starts issuing commands to the SDRAM devices
- The user asserts NOP on `CMD`

Figure 4. REFRESH Timing Diagram



PRECHARGE Command

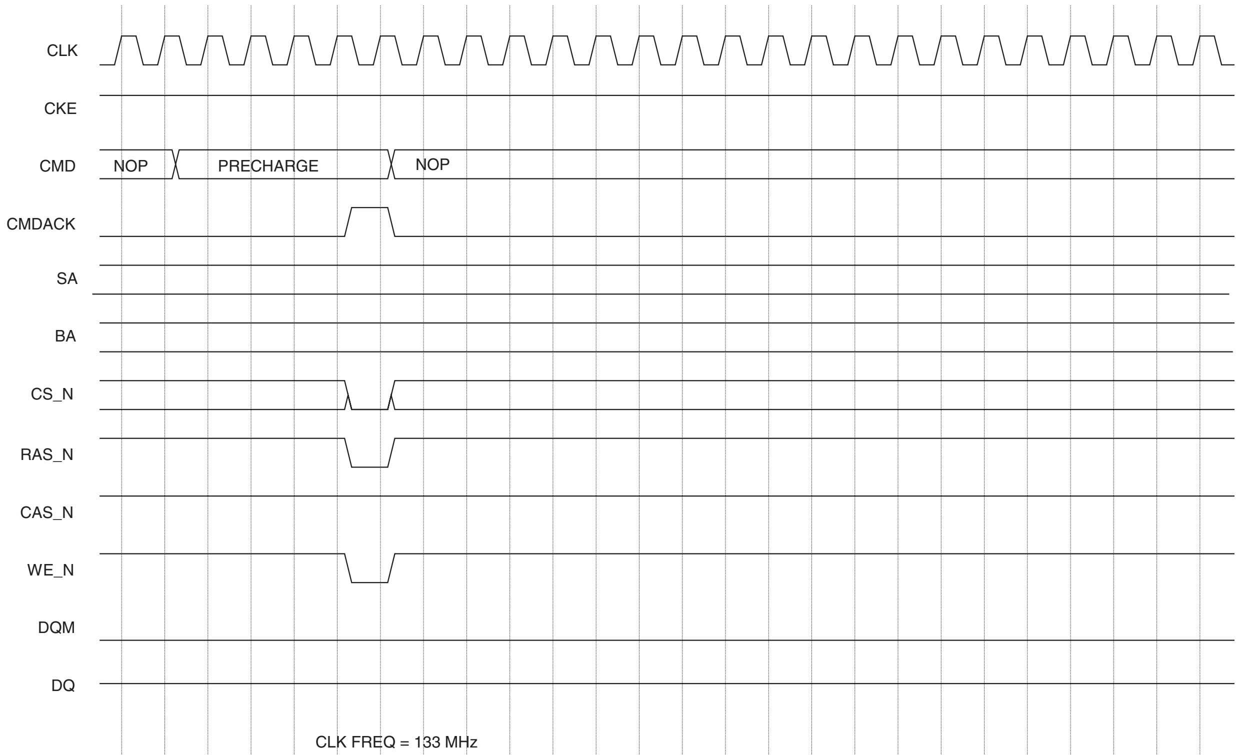
The PRECHARGE command instructs the SDR SDRAM Controller to perform a PCH command to the SDRAM. The SDR SDRAM Controller acknowledges the command with `CMDACK`. The PCH command is also used to generate a burst stop to the SDRAM. Using PRECHARGE to terminate a burst is only supported in the full-page mode.

Note that the SDR SDRAM Controller adds a latency from when the host issues a command to when the SDRAM sees the PRECHARGE command of 4 clocks. 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.

Figure 5 shows an example timing diagram of the PRECHARGE command. The following sequence describes the general operation of a PRECHARGE command:

- The user asserts PRECHARGE on `CMD`
- The SDR SDRAM Controller asserts `CMDACK` to acknowledge the command and simultaneously starts issuing commands to the SDRAM devices
- The user asserts NOP on `CMD`

Figure 5. PRECHARGE Timing Diagram

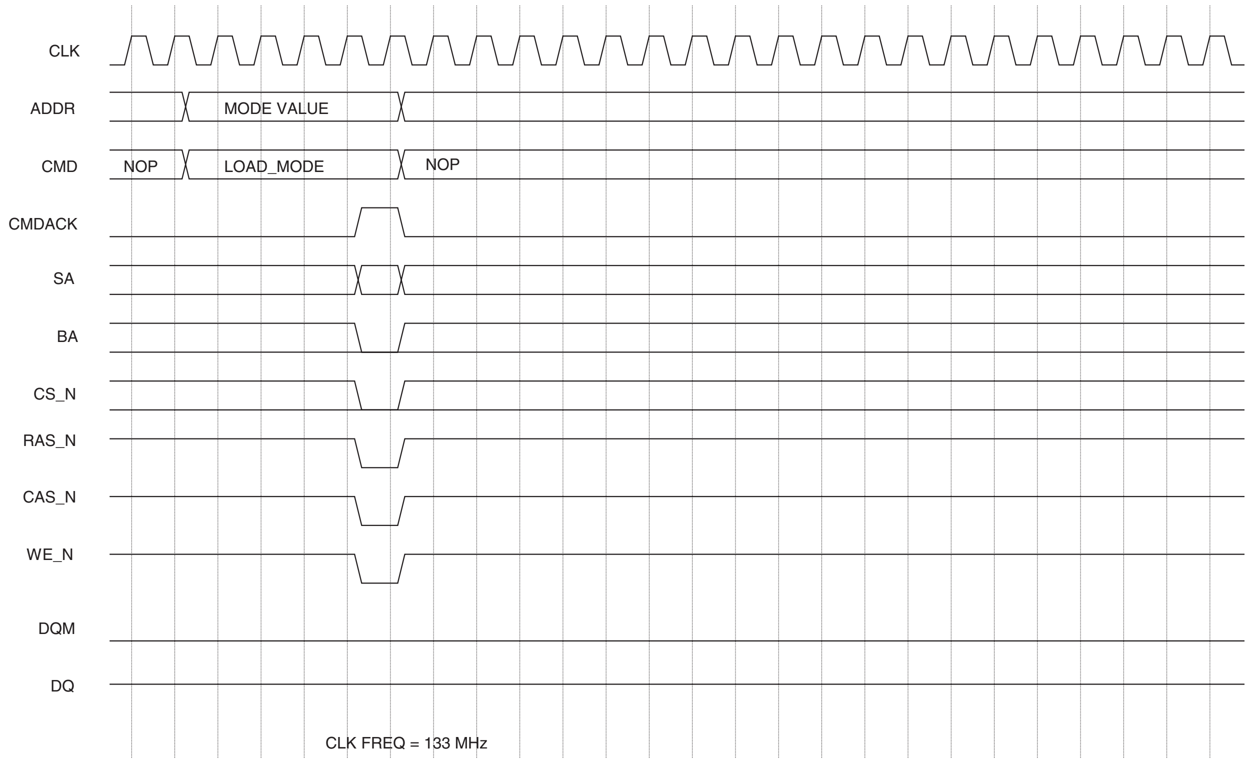


LOAD_MODE Command

The LOAD_MODE command instructs the SDR SDRAM Controller to perform a LMR 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 SDR SDRAM Controller issues the LMR to the SDRAM. Figure 6 shows an example timing diagram. The following sequence describes the general operation of a LOAD_MODE command:

- The users asserts LOAD_MODE on `CMD`
- The SDR SDRAM Controller asserts `CMDACK` to acknowledge the command and simultaneously starts issuing commands to the SDRAM devices
- One clock after the SDR SDRAM Controller asserts `CMDACK`, the users asserts NOP on `CMD`

Figure 6. LOAD_MODE Timing Diagram



LOAD_REG1 Command

The LOAD_REG1 command instructs the SDR SDRAM 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. Table 4 shows the bit assignments for REG1.

Table 4. REG1 Bit Definitions

Label	ADDR Bits	Description
CL	[1:0]	CAS latency setting.
RCD	[3:2]	RAS to CAS delay in number of clocks.
RRD	[7:4]	REFRESH command duration in clocks.
PM	[8]	SDR SDRAM Controller mode, 0 = normal 1 = page mode.
BL	[12:9]	Burst length, valid values are 1, 2, 4, 8.

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.

RCD is the RAS to CAS delay in clock periods and is dependent on the SDRAM speed grade and clock frequency.

RCD = INT(t_{RCD}/*clock_period*), where t_{RCD} is the value from the SDRAM data sheet and *clock_period* is the clock period of the SDR SDRAM Controller and SDRAM clock.

RRD is the refresh to RAS delay in clock periods. RRD is dependent on the SDRAM speed grade and clock frequency.

RRD= INT(tRRD/*clock_period*), where tRRD is the value from the SDRAM data sheet and *clock_period* is the clock period of the SDR SDRAM controller and SDRAM clock.

PM is the page mode bit. If PM = 0, the SDR SDRAM Controller operates in non-page mode. If PM = 1, the SDR SDRAM Controller operates in page-mode. See Section “Full-Page Mode Operation” on page 14 for more information.

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

LOAD_REG2 Command

The LOAD_REG2 command instructs the SDR SDRAM Controller to load the internal configuration register REG2. REG2 is a 16-bit value that represents the period between REFRESH commands that the SDR SDRAM Controller issues. The value is set by the equation int(*refresh_period*/*clock_period*).

For example, if a SDRAM device connected to the SDR SDRAM Controller has a 64-ms, 4096-cycle refresh requirement the device must have a REFRESH command issued to it at least every

64 ms/4096 = 15.625 09 µs.

If the SDRAM and SDR SDRAM Controller are clocked by a 100 MHz clock, the maximum value of REG2 is 15.625 µs/0.01 µs = 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 SDR SDRAM Controller Configuration

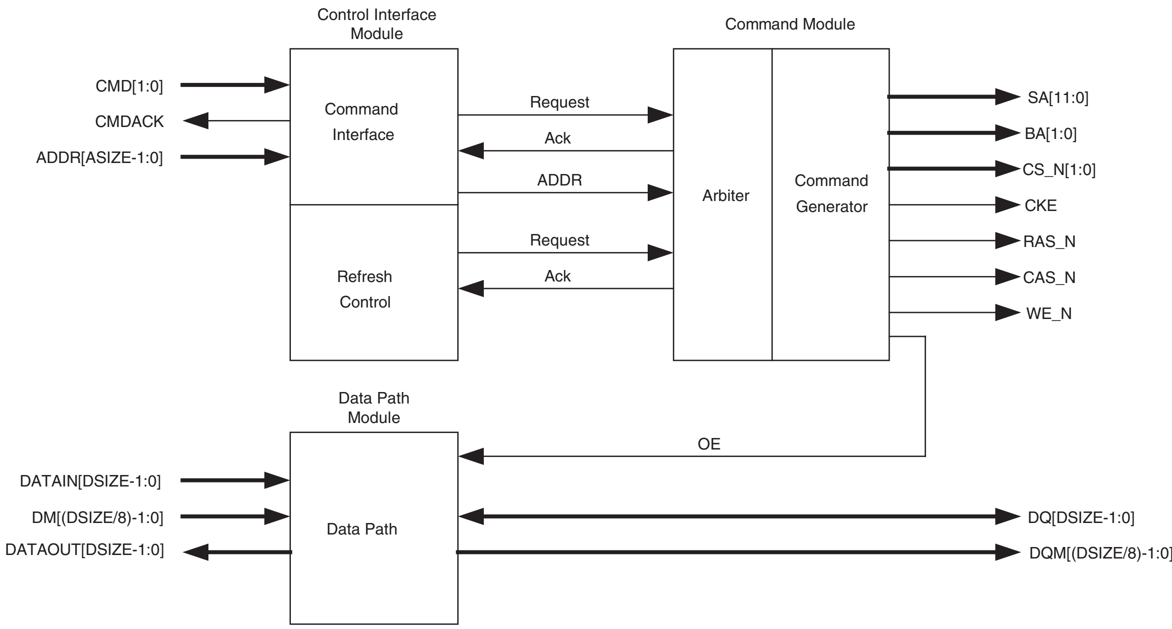
The SDRAM devices that are connected to the SDR SDRAM Controller must be initialized before they can be accessed. This initialization process sets the burst length, CAS latency, burst type, and operation mode for the SDRAM devices. After the SDRAM devices are initialized, the SDR SDRAM Controller’s configuration registers must be set. To initialize the SDRAM devices and the SDR SDRAM Controller, perform the following steps:

- Assert the PRECHARGE command (see “PRECHARGE Command” on page 7)
- Assert a LOAD_MODE command (see “LOAD_MODE Command” on page 8)
- Assert a LOAD_REG2 command (see “LOAD_REG2 Command” on page 10)
- Assert a LOAD_REG1 command (see “LOAD_REG1 Command” on page 9)

Architecture

The SDR SDRAM Controller 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 commands and addresses from the control interface module, and generates the proper commands to the SDRAM. The data path module handles the data path operations during WRITEA and READA commands. The SDRAM controller 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 SDR SDRAM Controller and can be easily removed. Figure 7 shows the SDR SDRAM Controller block diagram.

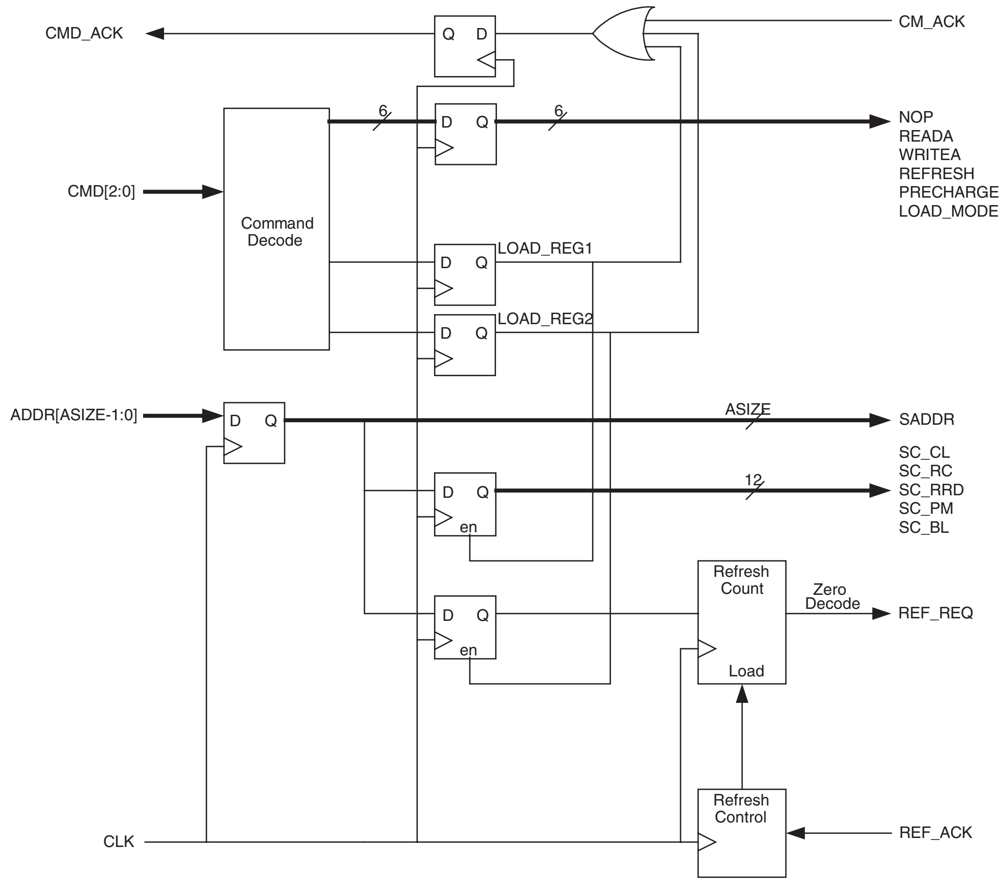
Figure 7. SDRAM Controller Block Diagram



Control Interface Module

The control interface module decodes and registers commands from the host, and passes the decoded NOP, WRITEA, READA, REFRESH, PRECHARGE, and LOAD_MODE commands, and 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. Figure 8 shows the control interface module block diagram.

Figure 8. Control Interface Module Block Diagram



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 when 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 SDR SDRAM Controller issues. The value is set by the equation int(*refresh_period*/*clock_period*).

For example, if an SDRAM device that is connected to the SDR SDRAM Controller has a 64-ms, 4096-cycle refresh requirement, the device must have a REFRESH command issued to it at least every

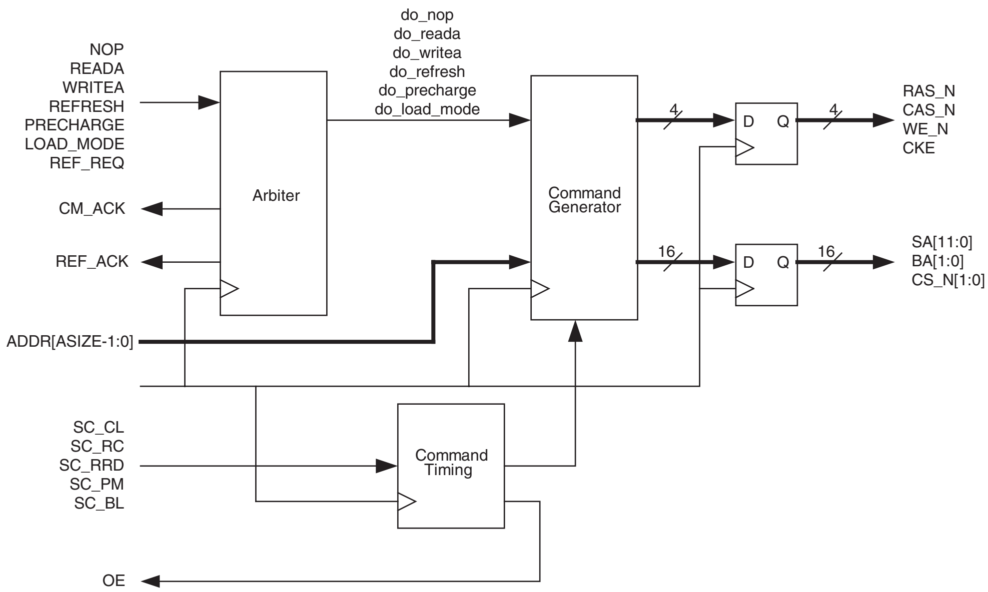
64 ms/4096 = 15.625 µs.

If the SDRAM and SDR SDRAM Controller are clocked by a 100-MHz clock, the maximum value of REG2 is 15.625 µs/0.01µs = 1562d.

Command Module

The command module accepts decoded commands from the control interface module, 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, the hidden refresh is held off until the host operation is complete. Figure 9 shows the command module block diagram.

Figure 9. Command Module Block Diagram



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; a second is used to control the positioning of the READA or WRITEA commands; a third is used to time command durations, which allows the arbiter to determine if the last requested operation has been completed.

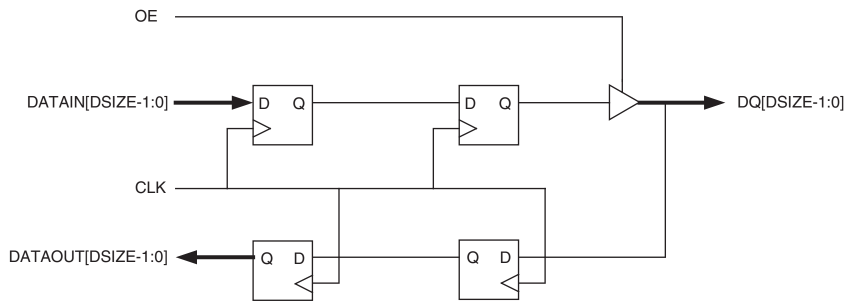
The command module also performs the multiplexing of the address to the SDRAM. The row portion of the address is multiplexed out to the SDRAM outputs A[11:0] during the ACTIVATE(RAS) command. The column portion is then multiplexed out to the SDRAM address outputs during a READA(CAS) or WRITEA command.

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.

Data Path Module

The data path module provides the SDRAM data interface to the host. Host data is accepted on DATAIN for WRITEA commands and data is provided to the host on DATAOUT during READA commands. Figure 10 shows the data path module block diagram. 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. DATAOUT consists of a 2-stage pipeline that registers data from the SDRAM during a READA command. DATAOUT pipeline delay can be reduced to one or even zero registers, with the only affect that the relationship of DATAOUT to CMDACK changes.

Figure 10. Data Path Module Block Diagram



Full-Page Mode Operation

The SDR SRAM Controller supports full-page mode operation of the SDRAM devices that it controls. The SDR SDRAM Controller is configured for full-page operation by setting bit 8 of REG 1 (see section "LOAD_REG1 Command" on page 9). When the SDR SDRAM Controller is in full-page mode, the auto-hidden refresh function is disabled and READA and WRITEA commands become READ/WRITE (without auto-precharge). 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. Figure 11 shows a full-page READ burst with a PRECHARGE to terminate the burst timing diagram. The following list details the full-page READ operation:

- The user asserts the READ Command on `CMD` and `ADDR`
- The SDR SDRAM Controller asserts `CMDACK` to acknowledge the command and simultaneously starts issuing commands to the SDRAM devices
- One clock after the SDR SDRAM Controller asserts `CMDACK`, the user asserts NOP on `CMD`
- (RCD + CL + 2) clocks after the SDR SDRAM Controller asserted `CMDACK`, the SDR SDRAM 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 user asserts PRECHARGE on `CMD`
- 4 clocks later, the SDR SDRAM Controller asserts `CMDACK` to acknowledge the PRECHARGE command and simultaneously issues a PRECHARGE command to the SDRAM devices
- (CL – 1 + 2) clocks later, the READ burst terminates on `DATAOUT`

Figure 11. Full-Page READ Burst Timing Diagram

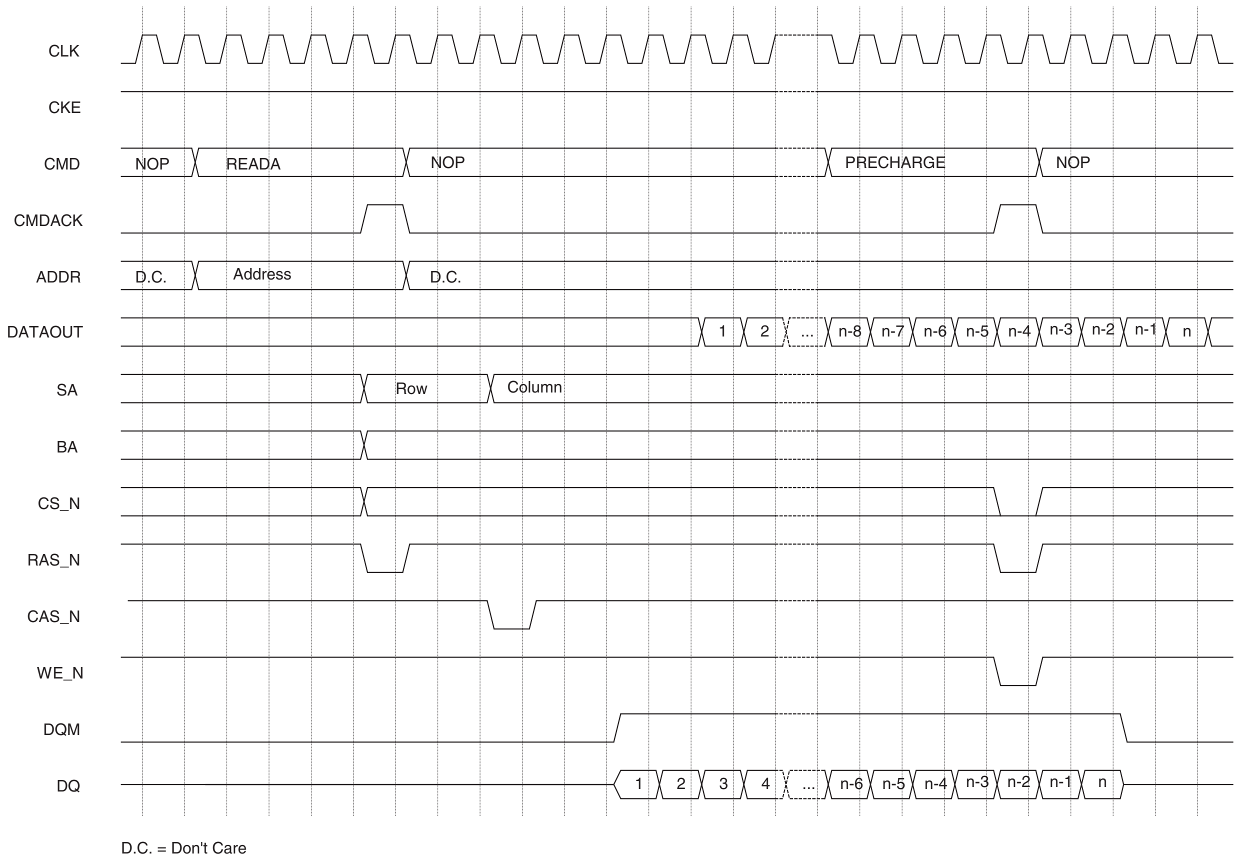
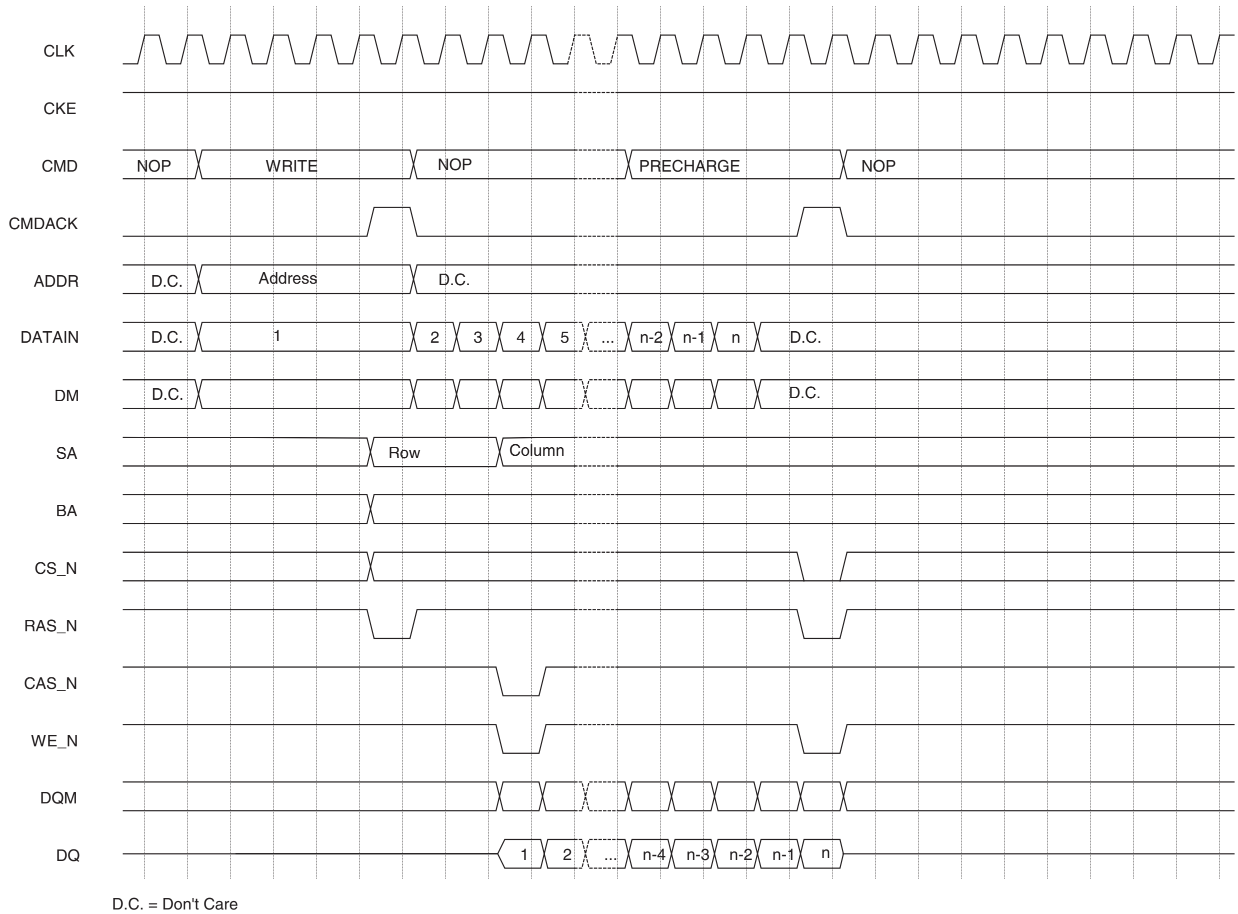


Figure 12 shows a full-page WRITE burst with a PRECHARGE to terminate the burst timing diagram. The following list details the full-page WRITE operation is as follows:

- The user asserts WRITE on `CMD` and `ADDR`
- The SDR SDRAM Controller asserts `CMDACK` to acknowledge the command and simultaneously starts issuing commands to the SDRAM devices
- One clock after the SDR SDRAM Controller asserts `CMDACK`, the user asserts NOP on `CMD`
- (RCD – 2) clocks after the SDR SDRAM Controller asserts `CMDACK`, the user starts clocking data into the SDR SDRAM Controller on `DATAIN`
- 3 clocks before the last WRITE data value is to appear on `DATAOUT`, the user asserts PRECHARGE on `CMD`. This delay is (5 clock command delay – t_{WR}), which is 2 in this example
- 4 clocks later, the SDR SDRAM Controller asserts `CMDACK` to acknowledge the PRECHARGE command and simultaneously issues a PRECHARGE command to the SDRAM devices
- The full-page WRITE burst is terminated

Figure 12. Full-Page WRITE Burst Timing Diagram



Simulation, Synthesis, Place-&-Route, & Results

The SDR SDRAM Controller includes source files, a testbench, synthesis scripts, and support files for place-and-route using the Quartus® II software version 2.1 targeting an EP20K200EFC672-1X device. Table 5 shows the SDR SDRAM Controller directory structure.

Table 5. SDR SDRAM Controller Directory Structure

Directory	Description
\doc	Contains the documentation.
\simulation	Contains the SDR SDRAM controller testbench.
\source	Contains the source files for the SDR SDRAM controller.

Simulation

The SDR SDRAM Controller can be simulated by using the behavioral source file **sdr_sdram_tb** (**/simulation** directory). The testbench performs the following actions:

- Instantiates the SDR SDRAM Controller and two SDRAM memory models
- Configures the SDRAM devices and SDR SDRAM Controller
- Runs a ramp pattern test using all combinations of CAS latency, burst length, and RAS to CAS delays for a given clock frequency

Higher clock frequencies (e.g., 133 MHz vs. 100 MHz) limit the RAS to CAS delays and CAS latency choices. The test assumes an interface to a Micron MT481C8M16A2 SDRAM device (not supplied). For more information on the Micron models, see http://www.micron.com.

Place & Route

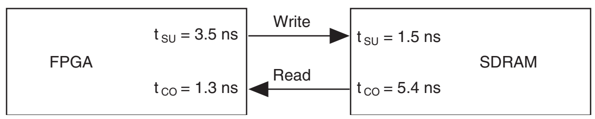
The t_{CO} and t_{SU} values are important parameters to consider, if you want to run the SDR SDRAM Controller at higher speeds. In addition Alter a recommends placing data pins in one bank and control pins in another bank.

PC133 SDRAM devices typically have a t_{CO} of 5.4 ns and a t_{SU} of 1.5 ns.

PCB delay is given by the following equation:

PCB delay = (clock (ns)) – (FPGA setup time (ns)) – (SDRAM t_{CO} (ns)) shows typical PCB delays.

Figure 13. PCB Delays



For a write operation at 133 MHz, the PCB delay = (7.5 – 1.5 – 3.5) = 0.9 ns.

For a read operation at 133 MHz, the PCB delay = (7.5 – 1.3 – 5.4) = 1.5 ns.

For more information on PCB design for high-speed applications, see *Application Note 75: High-Speed Board Designs*.

Throughput

Maximum throughput is achieved with page bursts. For random operation throughput is given by:

Throughput = 1/clock period (ns) × data path width (bytes) × burst length of access/number of clock cycles per access

For example, if data path width = 32 bits (4 bytes), clock cycles = 20, burst length of access = 8,
throughput = 1/7.5 × 4 × 8/20 = 212 Mbytes/s

Note, a write throughput is higher than a read throughput, because a WRITEA operation requires 15 clock cycles

Performance

Table 6 shows the performance results for the SDR SDRAM Controller. The results were generated with the Quartus II software version 2.1 SP1.

Table 6. Post Route Performance Comparison

Device	Internal f_{MAX} (MHz)	t_{CO}/t_{SU} (ns/ns)	Total LEs
APEX 20KE EP20K60EBC356-1X	100	3.3/1.25	219
APEX II EP2A15B724C7	133	3.3/1.3	219
Stratix EP1S10F780C5	133	1.6/1.5	175
Cyclone EP1C6Q240C6	133	1.6/1.4	173

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com