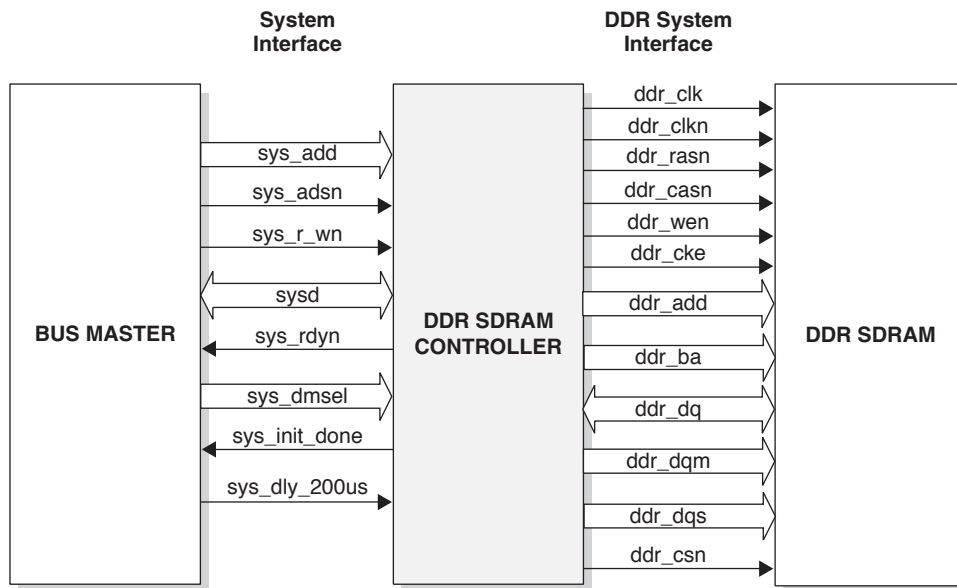


Introduction

The DDR SDRAM uses double data rate architecture to achieve high-speed data transfers. DDR SDRAM (referred to as DDR) transfers data on both the rising and falling edge of the clock. This reference design provides an implementation of the DDR memory controller implemented in a Lattice ORCA[®] Series 4 FPGA device. This DDR controller is typically implemented in a system between the DDR and the bus master. Figure 1 shows the relationship of the controller between the bus master and the DDR. The bus master could be either a microprocessor like the Intel i960 or a user's proprietary module interface.

For illustration purposes, the Micron DDR SDRAM MT46V16M8 (4Meg x 8 x 4 banks) is chosen for this design. This design has been verified using Micron's simulation model. It is highly recommended to download the simulation model from the DDR vendor for timing simulation.

Figure 1. SDR SDRAM Controller System



Features

- Simplifies DDR command interface to standard system read/write interface.
- Internal state machine built for DDR power-on initialization and auto refresh.
- Read/Write cycle access time optimized automatically according to DDR timing spec and the mode it is configured to.
- Auto refresh is done automatically without bus master intervention.
- Easily configurable to support different CAS latency and burst length, by changing the parameters.
- Supports ORCA Series 4 devices.

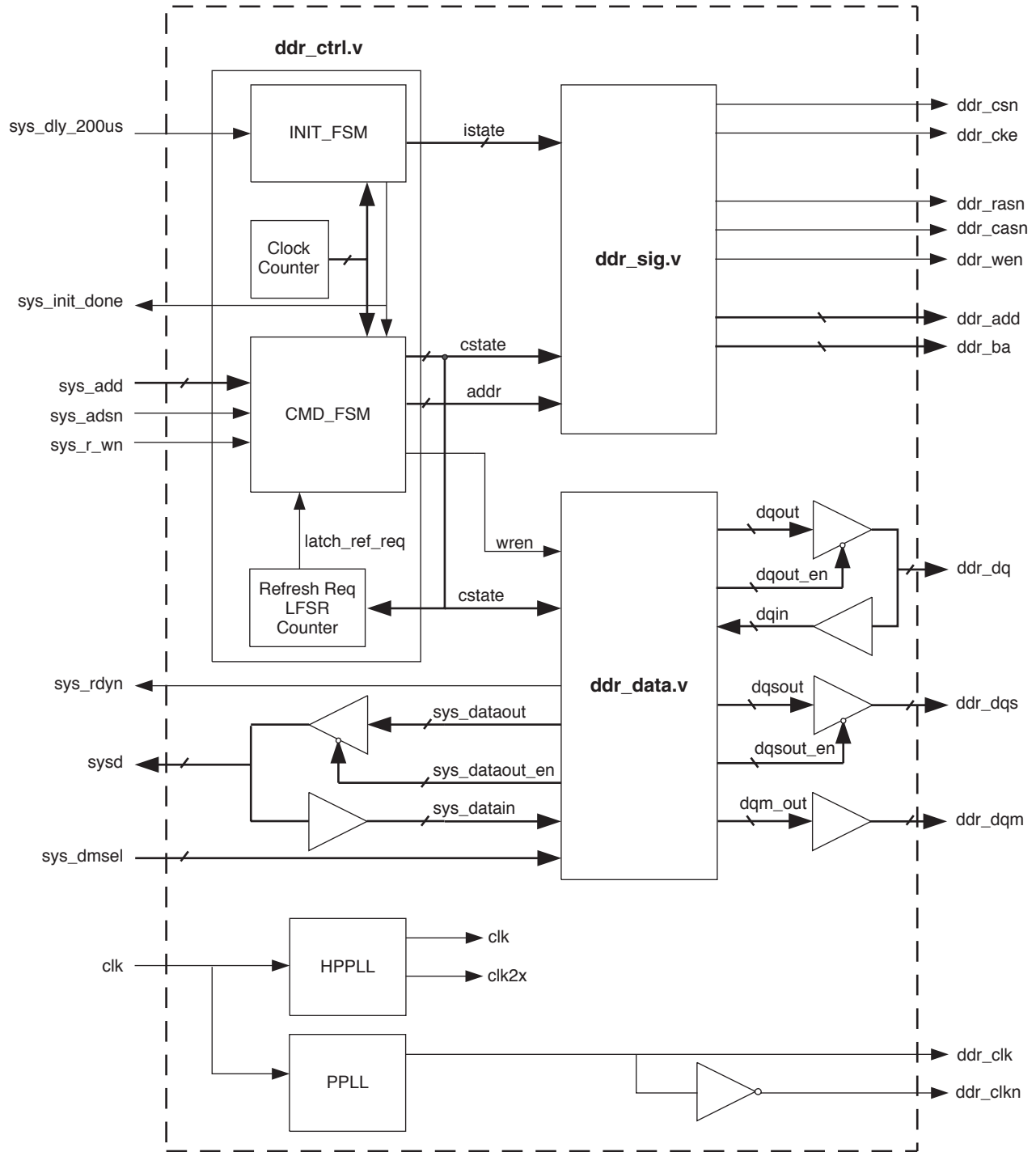
Pin Descriptions

Pin Name	Type	Active	Pin Description
System Interface Signals			
clk	In	NA	System interface clock.
reset_n	In	Low	This active low signal resets the controller to the initial state.
sys_adsn	In	Low	Active low system interface address strobe. This pin indicates the start of a bus master cycle.
sys_add[RA_MSB:CA_LSB]	In	NA	System interface address bus.
sys_r_wn	In	NA	System interface read/write signal. High indicates a read cycle and low indicates a write cycle. When this pin is high, it indicates to the controller that the bus master is performing a read cycle. When low, it indicates that it's a write cycle.
sysd[DSIZE-1:0]	In/Out	NA	Bi-directional system interface data bus.
sys_rdyn	Out	Low	Active low ready signal (for both read/write cycles). Indicates that data on sysd bus is valid when asserted low. System samples this pin on rising edge of clk and if asserted low makes a data transfer.
sys_init_done	Out	High	This active high signal indicates that the DDR SDRAM initialization is completed. Until this signal is asserted no READ/WRITE to be performed.
sys_dly_200us	In	High	This active high signal indicates to the controller that the DDR SDRAM has gone through the 200 μ s delay for power and clock stabilization.
DDR SDRAM Controller			
ddr_clk	Out	NA	DDR SDRAM clock
ddr_clkn	Out	NA	DDR SDRAM negated clock
ddr_cke	Out	High	DDR SDRAM clock enable
ddr_csn	Out	Low	DDR SDRAM command inputs CS#
ddr_dq[DSIZE/2-1:0]	In/Out	NA	DDR SDRAM data bus
ddr_dqs[DSIZE/16-1:0]	Out	NA	DDR SDRAM data strobe bus
ddr_add[DDR_A_WIDTH-1:0]	Out	NA	DDR SDRAM address bus
ddr_ba[DDR_BA_WIDTH-1:0]	Out	NA	DDR SDRAM bank address
ddr_rasn	Out	Low	DDR SDRAM command inputs RAS#
ddr_casn	Out	Low	DDR SDRAM command inputs CAS#
ddr_wen	Out	Low	DDR SDRAM command inputs WE#
ddr_dqm[DSIZE/16-1:0]	Out	High	DDR SDRAM data bus mask When high, masks write data.

Functional Description

The functional block diagram of the DDR controller is shown in Figure 2. It consists of three modules, the main control module, the signal generation module and the data path module. The main control module has two state machines and a refresh counter, which generates proper *istate* and *cstate* outputs according to the system interface control signals. The signal generation module generates the address and command signals required for DDR based on *istate* and *cstate*. The data path module performs the data latching and dispatching of the data between the bus master and DDR.

Figure 2. Block Diagram



PLL

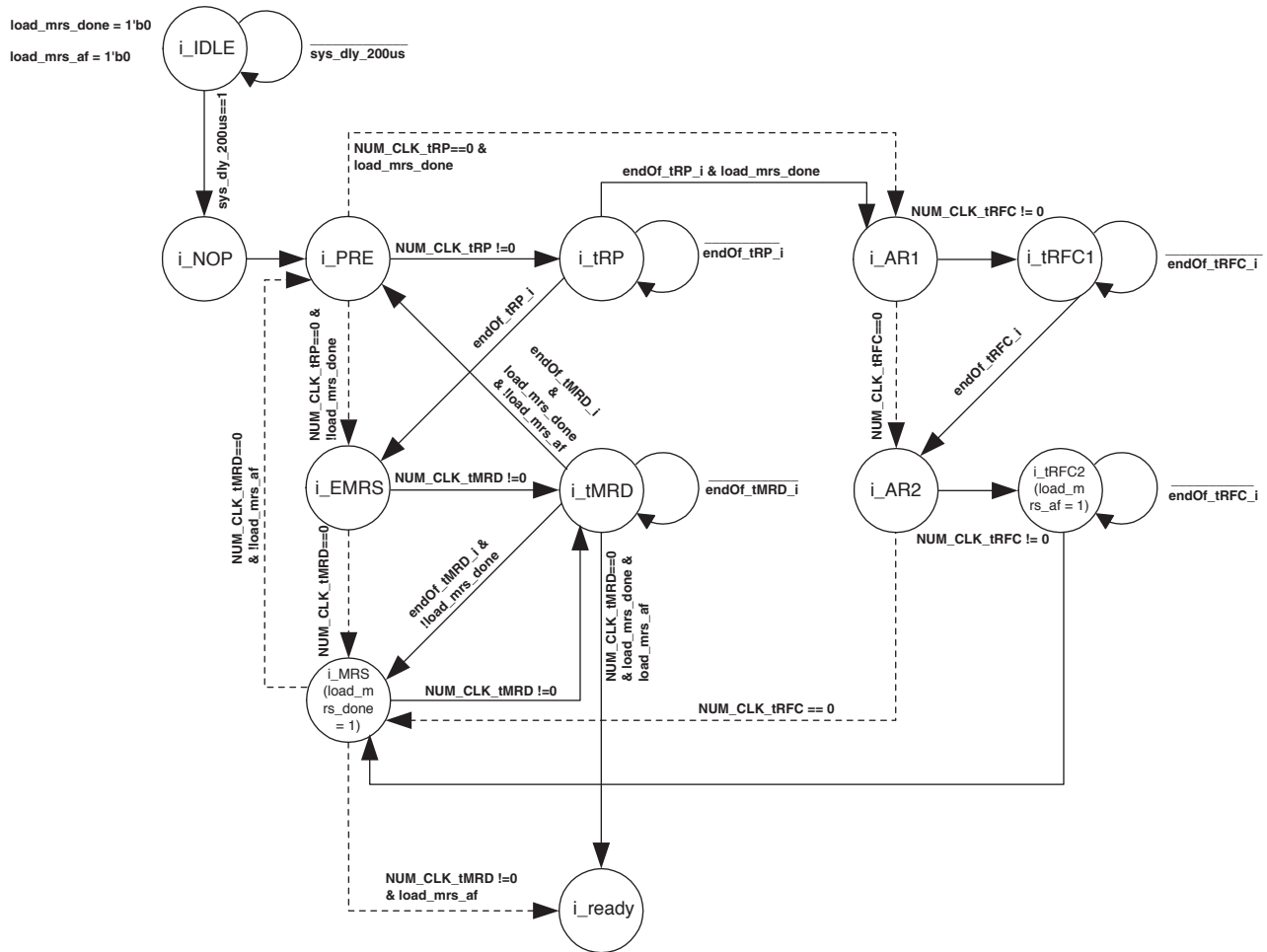
HPPLL (range 60-420MHz) is required for generation clk (133MHz) and clk2x (266MHz). PPLL is required for generating `ddr_clk` and `ddr_clkn`. The reason for PPLL is to control the read data path delay. `ddr_clk` delay can be varied by varying feedback delay (`FB_PDEL`).

DDR Initialization

Before normal memory accesses can be performed, DDR needs to be initialized by a sequence of commands. The `INIT_FSM` state machine handles this initialization. Figure 3 shows the state diagram of the `INIT_FSM` state machine. During reset, the `INIT_FSM` is forced to the `i_IDLE` state. After reset, the `sys_dly_200US` signal will be sampled to determine if the 200 μ s power/clock stabilization delay is completed. After the power/clock stabilization is complete, the DDR initialization sequence will begin and the `INIT_FSM` will switch from `i_IDLE` to `i_NOP` state and in the next clock to `I_PRE`. The initialization starts with the PRECHARGE ALL command. Next a LOAD MODE REGISTER command will be applied for the Extended mode register to enable the DLL inside DDR, followed by another LOAD MODE REGISTER command to the mode register to reset the DLL. Then a PRECHARGE command will be applied to make all banks in the device to idle state. Then two, AUTO REFRESH commands, and then the LOAD MODE REGISTER command to configure DDR to a specific mode of operation. After issuing the LOAD MODE REGISTER command and the t_{MRD} timing delay is satisfied, `INIT_FSM` goes to `i_ready` state and remains there for the normal memory access cycles unless `reset_n` is asserted. Also, signal `sys_init_done` is set to high to indicate the DDR initialization is completed. The `i_PRE`, `i_AR1`, `i_AR2`, `i_EMRS` and `i_MRS` states are used for issuing DDR commands.

The LOAD MODE REGISTER command configures the DDR by loading data into the mode register through the address bus. The data present on the address bus (`ddr_add`) during the LOAD MODE REGISTER command is loaded to the mode register. The mode register contents specify the burst length, burst type, CAS latency, etc. Refer to the DDR vendor's data sheet for more detailed information about the mode register field definitions. A PRECHARGE/AUTO PRECHARGE command moves all banks to idle state. As long as all banks of the DDR are in idle state, mode register can be reloaded with different value thereby changing the mode of operation. However, in most applications the mode register value will not be changed after initialization. This design assumes the mode register stays the same after initialization. However, the user can change parameters like `cas_latency`, burst length etc. in the `ddr_par.v` file as necessary, without changing any code in the Verilog files.

Figure 3. INIT_FSM State Diagram

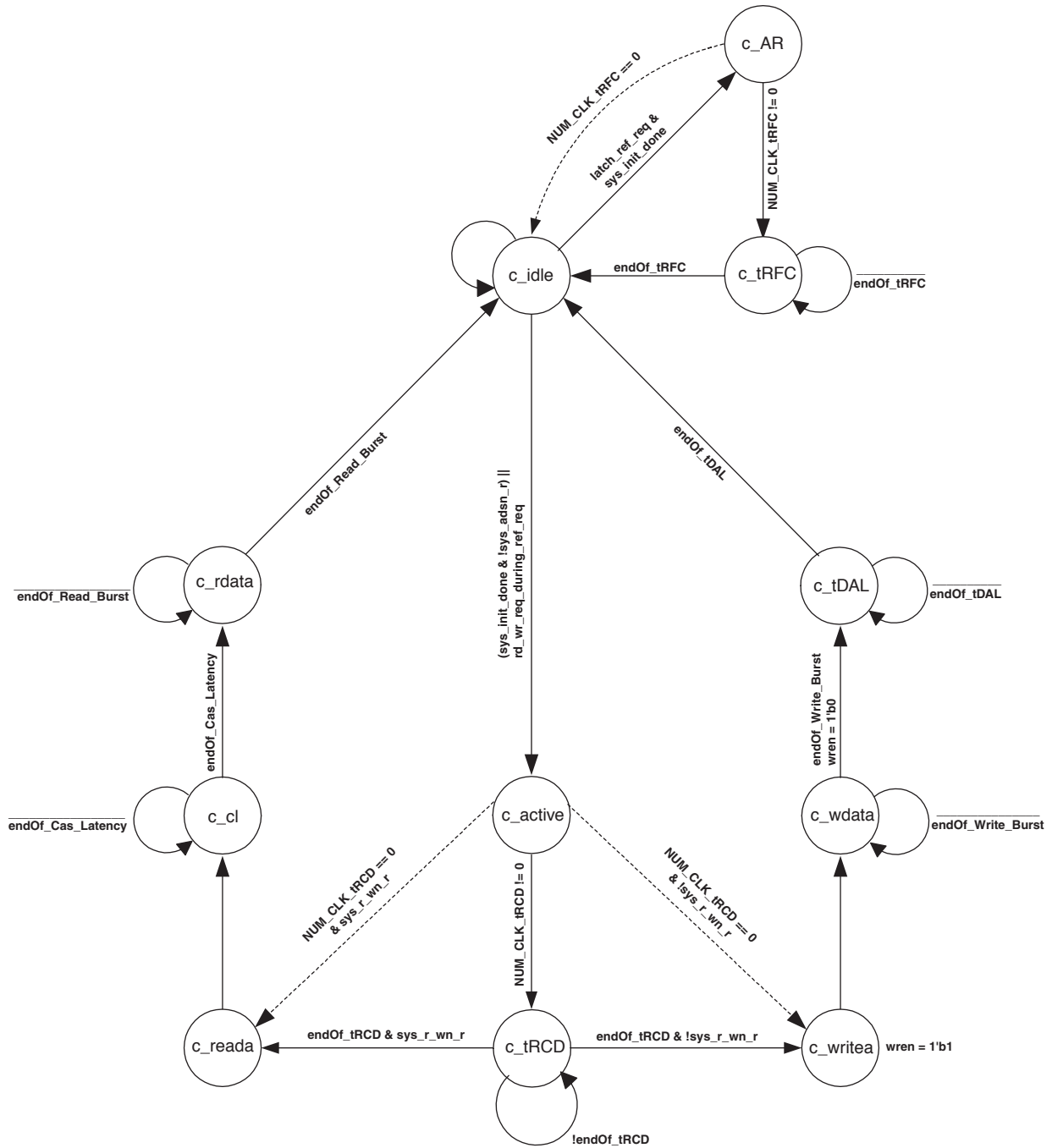


As mentioned above, certain timing delays (like t_{RP} , t_{RFC} , t_{MRD}) need to be satisfied before another non-NOP command can be issued. These DDR delays vary from speed grade to speed grade and sometimes from vendor to vendor. To accommodate this without sacrificing performance, the designer needs to modify the `ddr_par.v` file for the specific delays and clock period (t_{CK}). According to these timing values, the number of clocks, the state machine will stay at **i_tRP**, **i_tRFC1**, **i_tRFC2**, **i_tMRD** states will be determined. In the case when t_{CK} is larger than the timing delay, the state machine doesn't need to switch to the timing delay states and can go directly to the command states. The dashed lines in Figure 3 show the possible state switching paths.

Read/Write Cycle

Figure 4 shows the state diagram of **CMD_FSM**, which handles read, write and refresh of the DDR. The **CMD_FSM** state machine is initialized to `c_idle` during reset. After reset, **CMD_FSM** stays in `c_idle` as long as `sys_init_done` is low which indicates the DDR initialization sequence is not yet completed. From this state, a READA/WRITEA/REFRESH cycle starts depending on `sys_adsn/rd_wr_req_during_ref_req` and `latch_ref_req` signals as shown in the state diagram.

Figure 4. CMD_FSM State Diagram



All rows are in the “closed” status after the DDR initialization. The rows need to be “opened” before they can be accessed. However, only one row in the same bank can be opened at a time. Since there are four banks, there can be at most four rows opened at the same time. If a row in one bank, is currently opened, it needs to be closed before another row in the same bank can be opened. ACTIVE command is used to open the rows and PRE-CHARGE (or the AUTO PRECHARGE hidden in the WRITE and READ commands as used in this design) is used to close the rows. When issuing the commands for opening or closing the rows, both row address and bank address need to be provided.

In this design, the ACTIVE command will be issued for each read or write access to open the row. After a t_{RCD} delay is satisfied, READA or WRITEA commands will be issued with a high `ddr_add[10]` to enable the AUTO REFRESH for closing the row after access. Therefore, the clocks required for read/write cycle are fixed and the access can be random over the full address range.

Read or write is determined by the `sys_r_wn` status sampled at the rising edge of the clock before the t_{RCD} delay is satisfied. If a logic high is sampled, the state machine switches to `c_READA`. If a logic low is sampled, the state machine switches to `c_WRITEA`.

For read cycles, the state machine switches from `c_READA` to `c_cl` for CAS latency, then switches to `c_rdata` for transferring data from DDR to bus master. The burst length determines the number of clocks the state machine stays in `c_rdata` state. After the data is transferred, it switches back to `c_idle`.

For write cycles, the state machine switches from `c_WRITEA` to `c_wdata` for transferring data from bus master to DDR, then switches to `c_tDAL`. Similar to read, the number of clocks the state machine stays in `c_wdata` state is determined by the burst length. The time delay t_{DAL} is the sum of WRITE recovery time t_{WR} and the AUTO PRE-CHARGE timing delay t_{RP} . After the clock rising edge of the last data in the burst sequence, no commands other than NOP can be issued to DDR before t_{DAL} is satisfied.

The dashed lines indicate possible state switching paths when the t_{CK} period is larger than the timing delay specification.

Refresh Cycle

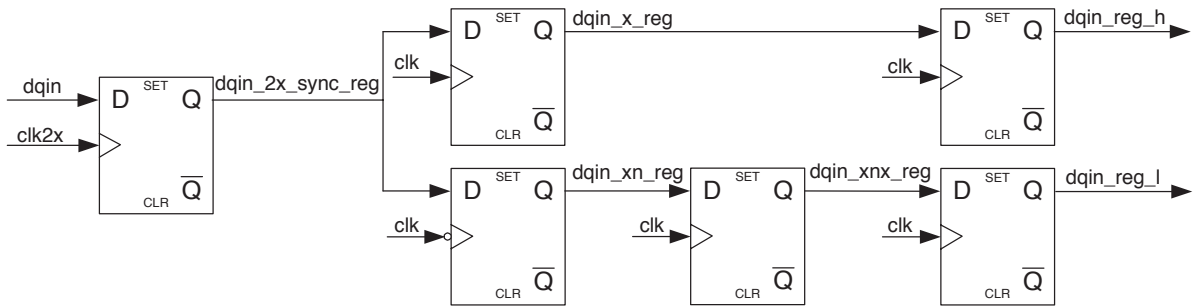
DDR memory needs a periodic refresh to hold the data. This periodic refresh is done using AUTO REFRESH command. All banks must be idle before an AUTO REFRESH command is issued. In this design all banks will be in idle state, as every read/write operation uses auto pre charge. A 128Mb DDR SDRAM requires an AUTO REFRESH command every $15.625\mu\text{s}$. At 133MHz, this results in 2078 clocks ($15.625\mu\text{s}$). LFSR counter is used to count for 2000 clocks and generates a refresh request to control state machine (`CMD_FSM`). `CMD_FSM` generates auto refresh command and resets the refresh request. For 256Mb/512Mb/1Gb DDR devices, AUTO REFRESH command is required every $7.8125\mu\text{s}$. At 133MHz this results to 1039 clocks. Hence according to the device and frequency of operation select the appropriate values for `REF_INTERVAL` in `ddr_par.v` file. There are four pre-defined values for `REF_INTERVAL`. They are `REF_INT_128MBIT_100MHZ` (for 128Mb devices running at 100MHz), `REF_INT_NON128MBIT_100MHZ` (for 256Mb/512Mb/1Gb devices running at 100MHz), `REF_INT_128MBIT_133MHZ` (for 128Mb parts running at 133MHz) and `REF_INT_NON128MBIT_133MHZ` (for 256Mb/512Mb/1Gb parts running at 133MHz).

Data Path

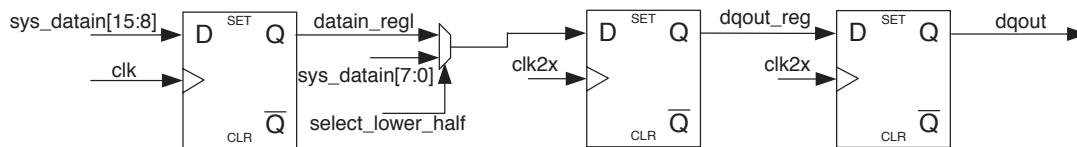
Figure 5 shows the data flow design between the DDR and the system interface. The module in this reference design interfaces between the DDR with an 8-bit data bus, and the bus master with a 16-bit data bus. The user should be able to modify this module to customize to fit the system bus requirements, by changing the `DSIZE` parameter in the `ddr_par.v` file.

Figure 5. Data Path Module

READ DATA PATH



WRITE DATA PATH



Timing Diagrams

Figure 6 and Figure 7 are the read cycle and write cycle timing diagrams of the reference design with the 2.5 CAS latency cycles and the burst length of eight. The timing diagrams may be different due to the values of the timing delays $t_{MRD}/t_{RP}/t_{RFC}/t_{RCD}/t_{RCD}/t_{WR}$, the clock period t_{CK} , the CAS latency and the burst length. The total number of clocks for read and writes cycles are decided by these factors.

The state variable c_state of CMD_FSM is also shown in these figures. Note that the ACTIVE, READ and WRITE commands are asserted one and half clock after the c_ACTIVE , c_READA and c_WRITEA states respectively.

The values of the region filed with slashes in the system interface input signals of these figures are “don’t care”.

Figure 6. Read Cycle Timing Diagram

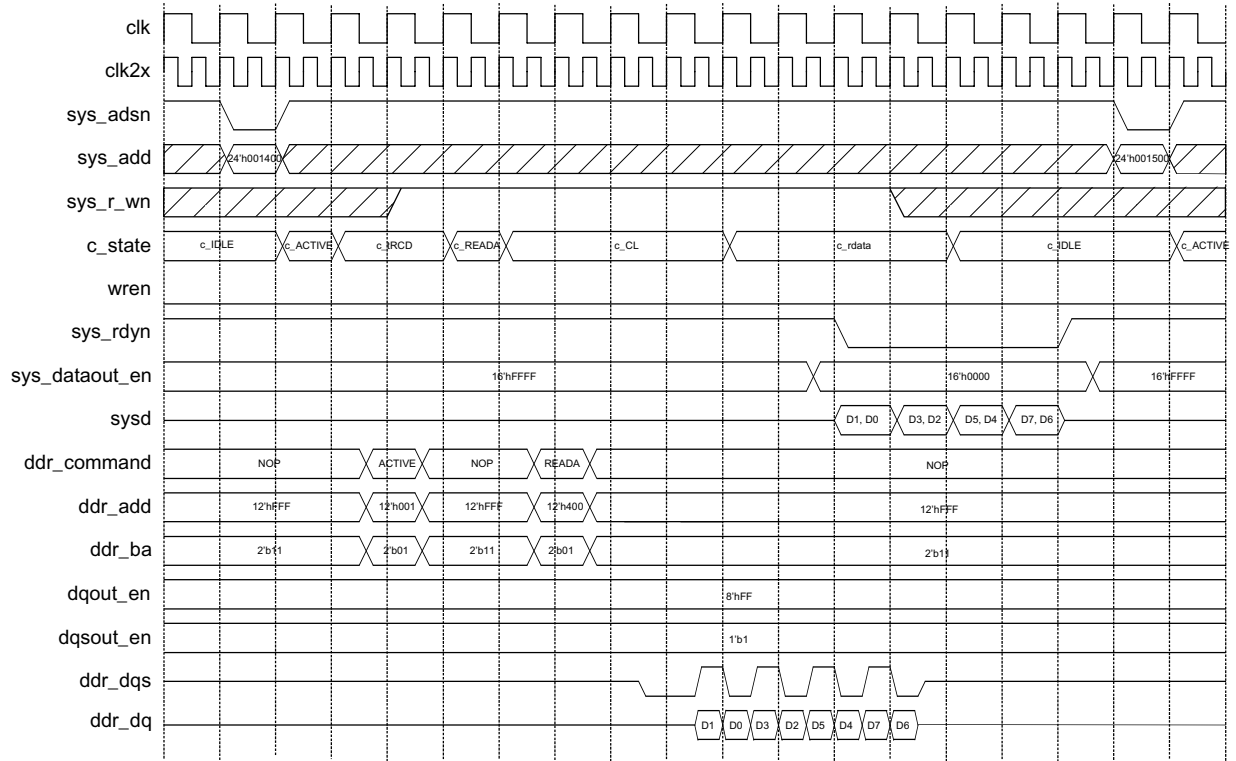
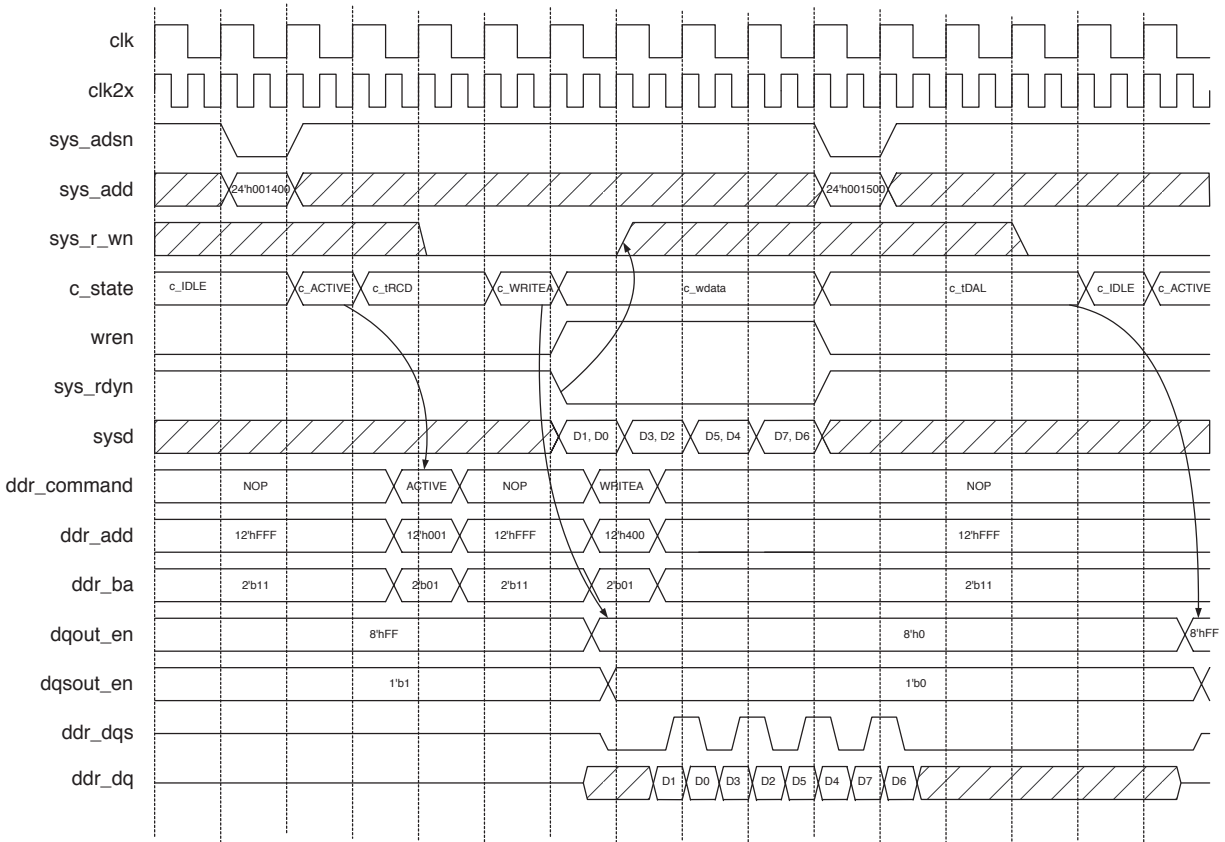


Figure 7. Write Cycle Timing Diagram



Implementation Results

The design software used for this implementation is Lattice ispLEVER® version 3.0 and the Synplicity Synplify synthesis tool, version 7.1.

Device utilization and performance summary for OR4E02-2 is given below.

Resource	Resource Used
I/O	80/410
PFU	50/624
LUT4	185
REG	249
GSR	1
HPPLL	1
PPLL	1

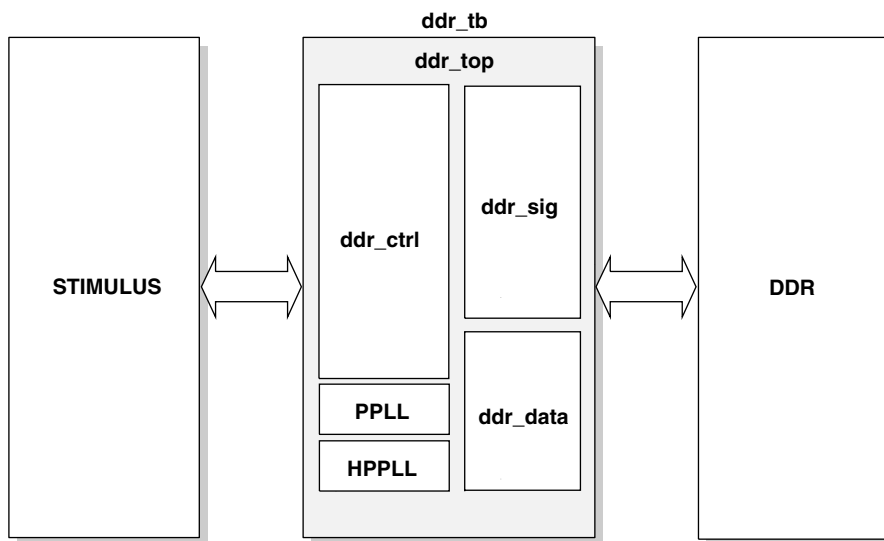
Performance:

Clocks in the Design	Reported Frequency
clk (pll_mclk)	146.649MHz
clk2x (pll_nclk)	314.358MHz

Test bench:

The Verilog source code and test fixture for this DDR controller design is provided as part of this reference design

Figure 8. Test Bench



Source code:

- ddr_top.v : Top level source code
- ddr_ctrl.v : Main control module
- ddr_sig.v : Signal generation module
- ddr_data.v : Data path module
- ddr_par.v : Parameter definition
- ddr_pll_orca.v : HPPLL instantiation
- ddr_pll_orca_sp.v : PPLL instantiation

Test bench:

- `ddr_tb.v` : Test bench top file
- `stimulus.v` : Generates stimulus to the design

Models:

- `mt46v16m8.v` : Micron DDR simulation model. Not included as part of this reference design.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-408-826-6002 (Outside North America)

e-mail: techsupport@latticesemi.com

Internet: www.latticesemi.com