RESEARCH ARTICLE OPEN ACCESS

Design and implementation of High Speed Pipelined DDR SDRAM memory Controller

G. Divya[1] P. Bhagya lakshmi[2] L. Srinivas[3]
[1,2,3]Assitant Professor, Electronics and Communication Engineering, CMR Engineering College, JNTU, Hyderabad.
(dha.grs@gmail.com,bhagyapotukanuma@gmail.com,seenulgsm@gmail.com)

Abstract –
We are living in an age of Electronics. Electronics is more and more dominating our daily life. These days most people use at least one personal computing device, like a pc, a tablet or a Smartphone. Memory is the most essential part of these devices. A dedicated memory controller is a very important thing in applications, and it does not contain microprocessors. The Memory Controller provides command signals for the device such as memory refresh, read and write operation and initialization of SDRAM. The Controller is used to simplify the SDRAM command interface to standard system read/write interface and also to optimize the access time of read/write cycle. The DDR SDRAM uses double data rate architecture to achieve high-speed data transfers. In this paper, Double Data Rate SDRAM Controller is implemented. The R/W control of DDR SDRAM is realized through Verilog HDL.

***Keywords* – *DDR* SDRAM Controller, Read/Write Data path**

I. INTRODUCTION

Memory devices are almost found in all systems and nowadays high speed and high performance memories are in great demand. For better throughput and speed, the controllers are to be designed with clock frequency in the range of megahertz. As the clock speed of the controller is increasing, the design challenges are also becoming complex. Therefore the next generation memory devices require very high speed controllers like double data rate and quad data rate memory controllers.

In this paper, the double data rate SDRAM Controller is implemented using ASIC methodology. Synchronous DRAM (SDRAM) is preferred in embedded system memory design because of its speed and pipelining capability. In high-end applications, like microprocessors there will be specific built in peripherals to provide the interface to the SDRAM. But for other applications, the system designer must design a specific memory controller to provide command signals for memory refresh, read and write operation and initialization of SDRAM.

In this paper, the SDRAM controller, located between the SDRAM and the bus master, minimizes the effort to deal with the SDRAM memory by providing a simple system to interact with the bus master. Figure 1 is the block diagram of the DDR SDRAM Memory Controller that is connected between the bus master and SDRAM [1].

SDRAM's are classified based on their data transfer rates. In Single data rate SDRAM, the data is transferred on every rising edge of the clock whereas in double data rate (DDR) SDRAM's the data is transferred on every rising edge and every falling edge of the clock and as a result the throughput is increased. DDR SDRAM Controllers are faster and efficient than its counterparts. They allow data transfer at a faster rate without much increase in clock frequency and bus width.

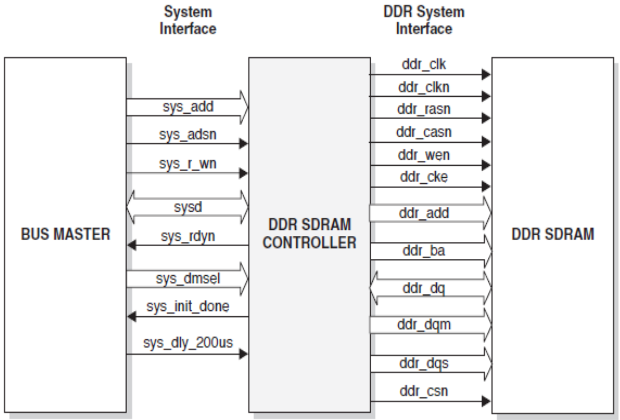

Fig. 1. DDR SDRAM controller system [1]

II. IMPLEMENTATION METHODOLODY OF DDR SDRAM

DDR SDRAM Controller architecture is implemented using Verilog HDL. The RTL

Synthesis and Simulations are performed using ModelSim-Altera 6.5b Edition.

III. ARCHITECTURE OF DDR SDRAM CONTROLLER

DDR SDRAM Controller module receives addresses and control signals from the BUS Master. The Controller generates command signals and based on these signals the data is either read or written to a particular memory location. The DDR SDRAM Controller architecture is shown in Figure 2. It consists of three modules: 1) Main control module 2) signal generation module 3) data path module. The main control module has two state machines and a refresh counter. The two state machines are for initialization of the SDRAM and for generating the commands to the SDRAM. They generate iState and cState outputs according to the system interface control signals. The signal generation module now generates the address and command signals depending upon the iState and cState. The data path module performs the read and writes operations between the bus master and DDR. Following are some of the important features of DDR SDRAM Controller:

i. The DDR SDRAM Read and Write operations are simplified by the controller.
ii. For initializing the DDR SDRAM controller, separate state machines are designed internally.
iii. The access time for read and the write cycle is optimized based on the CAS latency and burst length of the DDR SDRAM.
iv. The auto refresh for the DDR SDRAM is done by the controller.

The main control module consists of three sub modules: 1) Initialization FSM module (INIT_FSM). 2) Command FSM module (CMD_FSM) 3) Counter module.

A. Main control module

The DDR SDRAM Controller has to undergo an initialization process by a sequence of command signals before the normal memory access. The initialization finite state machine in the main control module is responsible for the initialization of the DDR SDRAM controller.

Figure 3 shows the state diagram of the initialization FSM (INIT_FSM). Whenever reset signal is high, the initialization FSM will switch to i_IDLE state. Once the reset signal goes low, the controller has to wait for 200us clock stabilization delay. This is constantly checked by sys_dly_200us signal and a high on the sys_dly_200us will indicate that the clock stabilization delay is complete. The DDR initialization sequence will begin immediately after the clock/power stabilization is complete and then the INIT_FSM will change its state from i_IDLE to i_NOP state. From the i_NOP state, the initialization FSM will switch to the i_PRE state on the next clock cycle.

In the figure 2, in the i_PRE state, the main control module will generate the PRECHARGE command. The PRECHARGE command generated during this state will be applied to all the banks in the device. Once the PRECHARGE command is generated by the initialization FSM, it will switch to the next state.

The next state in the design of initialization FSM is two AUTO REFRESH commands. These refresh commands are generated to refresh the DRAM memory. After the two refresh state, the initialization FSM will switch to i_MRS state. During this state LOAD MODE REGISTER command is generated to configure the DDR SDRAM to a specific mode of operation. After satisfying the i_tMRD timing delay the initialization FSM will switch to i_ready state. The initialization FSM will remain in the i_ready state for normal memory access. And also, when the initialization FSM switches to i_ready state signal sys_INIT_DONE is set to high to indicate that DDR SDRAM controller initialization is completed. The i*_PRE*, *i_AR1*, *i_AR2*, *i_EMRS* and *i_MRS* states are used for issuing DDR commands.

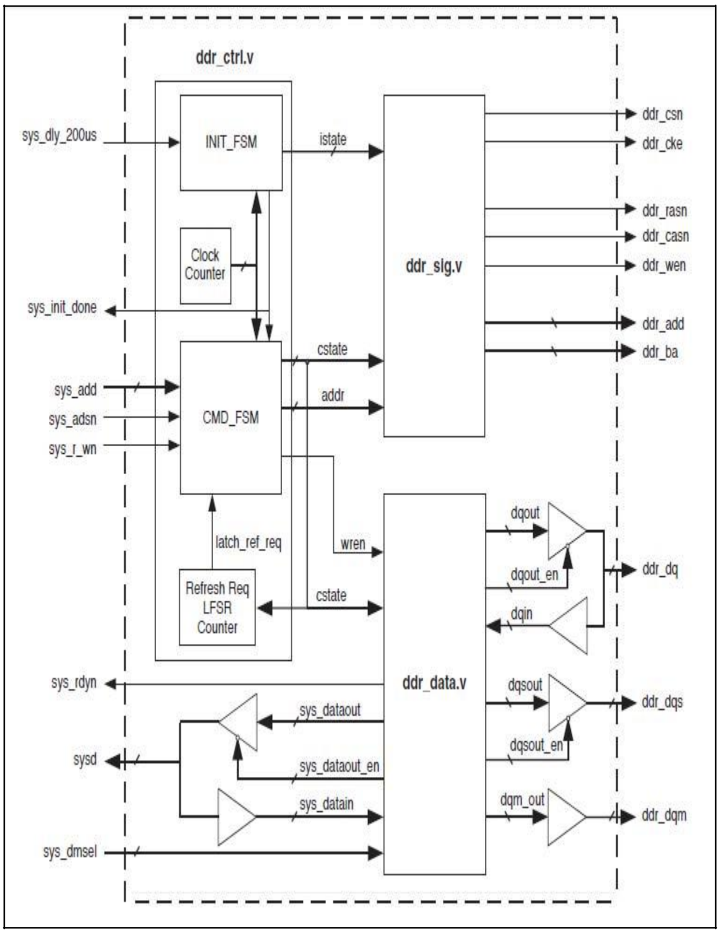

Fig. 2. Functional block diagram of DDR SDRAM controller [2]

CMD_FSM handles the read, write and refresh of the SDRAM. 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 SDRAM initialization sequence is not yet completed.

When the sys_INIT_DONE is high, it indicates the system initialization is complete. The controller will now wait for *latch_ref_req, sys_INIT_DONE* signals and will enter auto refresh, read and write mode depending upon these signals. When the initialization is complete and when the *latch_ref_req* goes high the controller will refresh by entering into refresh state. After the refresh is complete, when the *latch_ref_req* and *sys_ADSn* signal goes low, the controller will go to active state. The ACTIVE command will be issued for each read or write access to open the row.

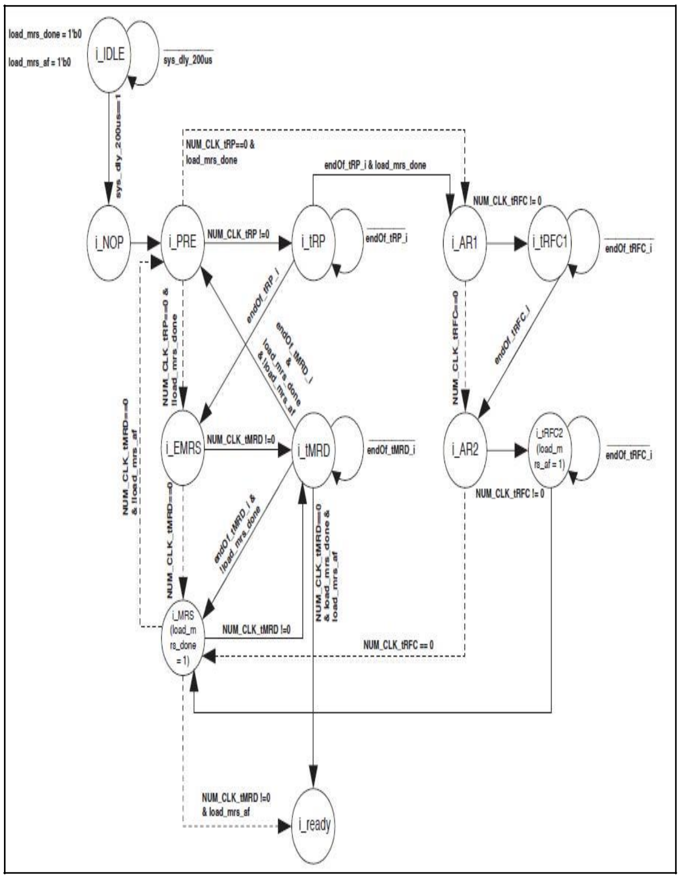

Fig. 3. DDR SDRAM Initialization FSM (INIT_FSM) state diagram [1].

After a specified delay is satisfied, READ or WRITE commands will be issued. Read or write is determined by the *sys_R_Wn* signal. If 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 specified delay, then switches to *c_rdata* for transferring data from SDRAM to bus master.

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 SDRAM, and then switches to c_tDAL. After writing the data, it switches back to c_IDLE state.

B. Data path module

The DDR SDRAM Controller design interfaces between the 8-bit data bus, and the bus master with a 16-bit data bus. The data path module for read and write are shown in figure. The data path module depends on cState for its read/write operation. The cState is generated by the CMD_FSM present in the Main Control module.

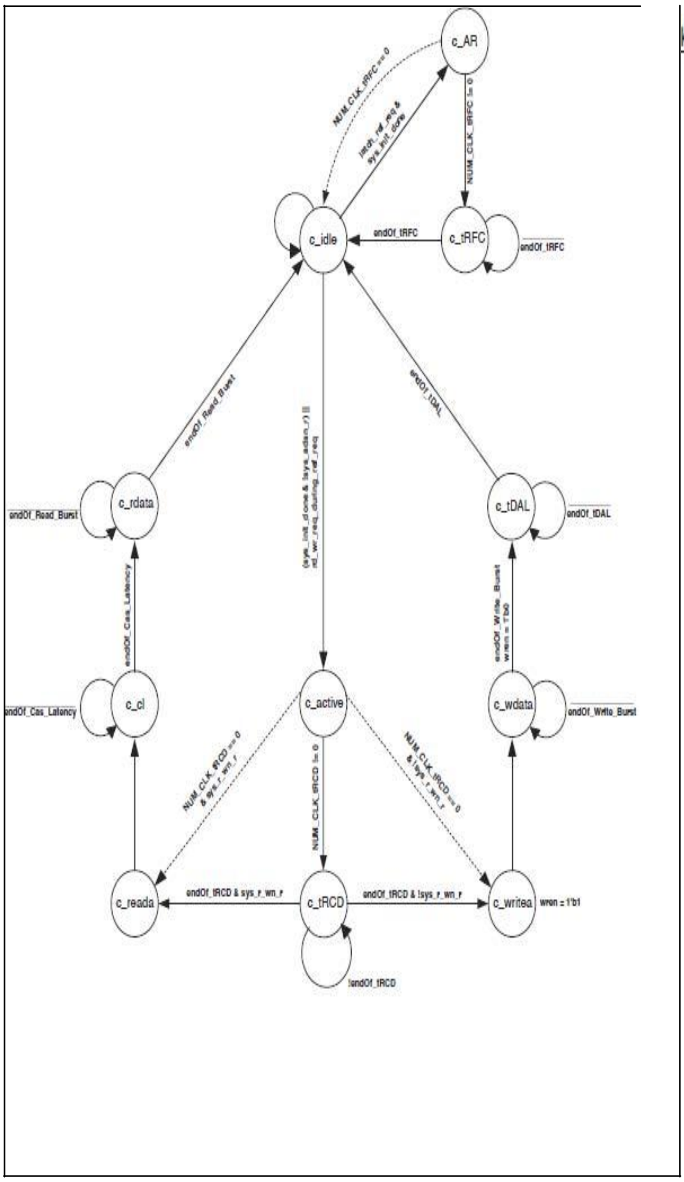

Fig. 4. DDR SDRAM command FSM (CMD_FSM) [1].

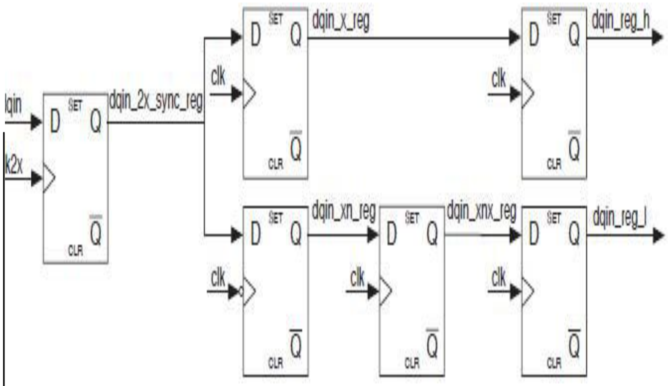

Fig. 5. Read data path for DDR SDRAM Controller [1].

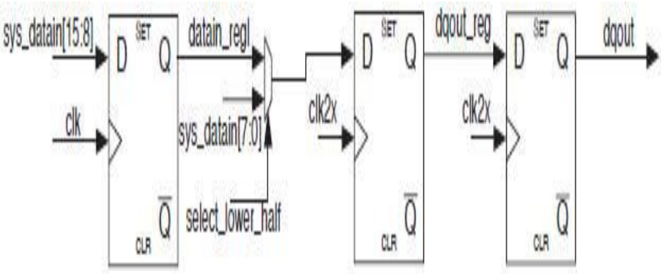

Fig. 6. Write data path for DDR SDRAM Controller [1].

C. Signal Generation Module

The DDR SDRAM Signal generation module should generate the command signals to the DDR SDRAM. The command signals include ddr_add; for generating addresses, ddr_casn and ddr_rasn; for selecting particular column and row address. These signals are generated based on the iState and cState received from the CMD_FSM and INIT_FSM present in the Main Control module.

IV. EXPERIMENTAL RESULTS

The Design is simulated after combining the BUS Master, DDR SDRAM Controller and the DDR SDRAM. Micron's MT46V16M8 [3] is chosen as the DDR SDRAM for this design. The Controller is designed using Verilog HDL. The simulation is done using ModelSim software. Figures below show the timing diagram of Controller Initialization, Read Cycle and Write Cycle. In figure 7, once the initialisation sequence for the DDR SDRAM Controller is completed, sys_INIT_DONE becomes high. In figure 8 and 9, on both positive edge and negative edge of the clock, the read and write operations are performed.

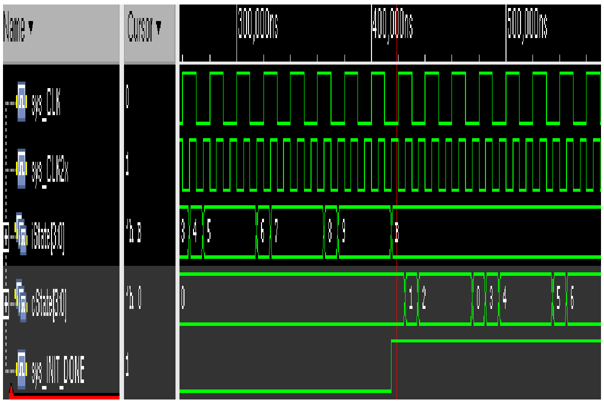

Fig. 7. Simulation result of the Initialization sequence of DDR SDRAM Controller

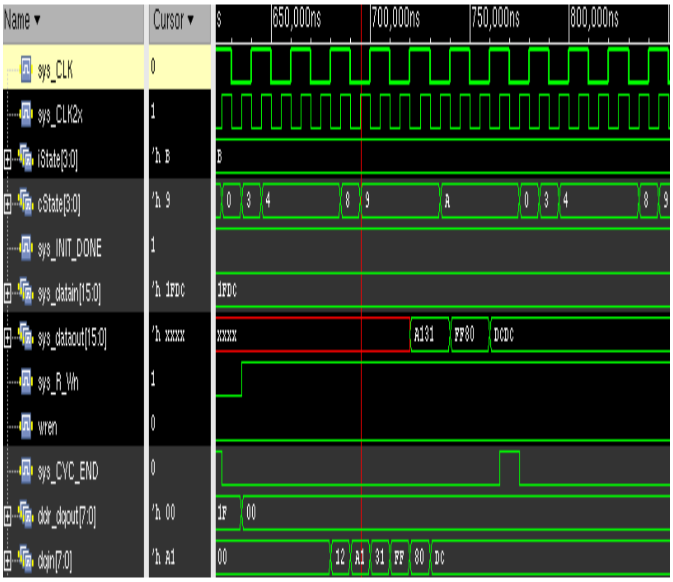

Fig. 8. Simulation result of the Read Cycle of DDR SDRAM Controller.

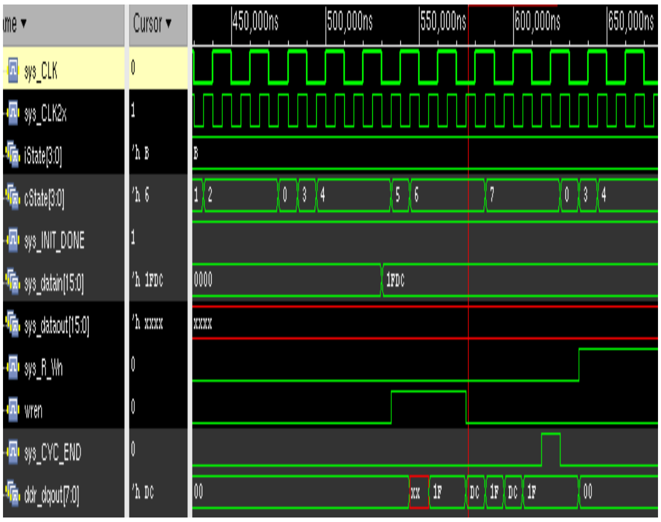

Fig. 9. Simulation result of the Write Cycle of DDR SDRAM Controller

V. CONCLUSION

The high speed pipelined DDR SDRAM Controller architecture is implemented in Verilog HDL. The design software used for this implementation is ModelSim-Altera 6.5b Edition. DDR is a result of collaboration between many companies. DDR is a simple evolution of SDRAM technology. Its configuration options available for different applications. The future of DDR is in evolutionary steps. Future work may include comparison of DDR/DDR1/DDR2/DDR3 architectures.

REFERENCES

[1] DDR SDRAM Controller white paper, *Lattice Semiconductor Corporation,* Reference Design: RD1020, April 2004.

[2] SDR SDRAM Controller white paper, *Lattice Semiconductor Corporation,* Reference Design: RD1010, April 2011.

[3] Micron Technology, Micron's Synchronous DRAM (2002)..

[4] Michael John Sebastian Smith, "*Application Specific Integrated Circuits", 2004.*

[5] http://en.wikipedia.org/wiki/DDR_SDRAM