VHDL Implementation of nor Flash Controller

Rajkuwar Kanase[1], Prof.Shivdas.S.S[2]
[1] Department of Electronics Engineering, Karmveer Bhaurao Patil, College of Engineering ,
Satara, District-Satara.415001.

***Abstract** –*

Flash memory is an electronic non-volatile storage medium that is developed from EEPROM, which is electrically erased and reprogrammed. NOR flash memories are mainly used in code storage, booting and executable in place codes. Flash memory controller is bridge between user and flash memory as it converts commands by user to required timing specification of flash memory. Operations such as program and erase are controlled by flash memory. Flash memory controller is developed in reconfigured FPGA environment. This paper focuses mainly on design and timing response of flash memory controller inVHDL.

I. INTRODUCTION

Flash controller is used whenever interface needs to interacts with flash memory; it will firstly have to face with flash memory controller. Flash memory has two types NAND and NOR, NAND is used in USB drive storages in that flash controller is on chip. NOR flash are used in booting applications.[1]

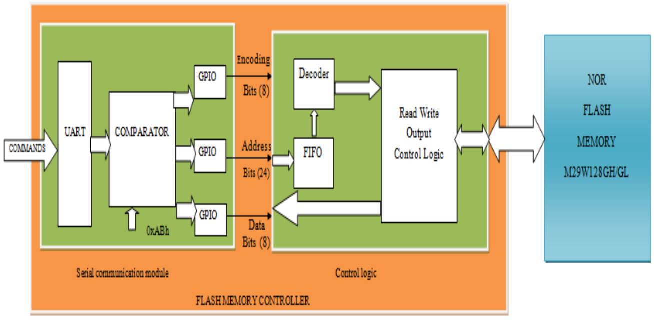

Fig 1: General Diagram of Flash Memory Controller

Block diagram shows design of flash memory controller. Flash controller communicates with parallel NOR flash memory M29W128GH/GL which is micron chip. Flash memory controller give response to commands from command interface. The interfaced system sends information like address and data bits one by one. Read and write operations are with respect to flash memory. Let's discuss its module one by one.[3][4]

II. SERIAL COMMUNICATION MODULE

UART receives command sequence and gives to comparator. There are command start bits which are added at the start of each command. When flash controller first receives these command start bits, we recognize that there is a valid command. Command start bits must be there at the start of each command because of absence of start bit command is discarded as it is a invalid command. 0xABh are selected as command start bits. For the further operation control logic needs data in parallel form. So three GPIO's are used to separate the incoming data bits into three parts which are as follows

- Command encoding bits
- Program address,
- Program data. Encoding bits are of the size 8 bits, program address is of size 24 bits and program data is of the size of 8 bits.[6]

III. FIFO (First In First Out)

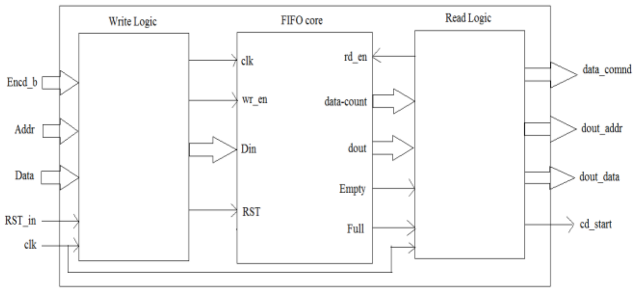

Fig 2: Block Diagram of FIFO

FIFO (first in first out) memory is used for buffering of command sequence which is given by serial logic. FIFO is used to store the incoming encoding bits, address bits and data bits. Incoming commands from UART are need to be stored for further working of flash memory controller and control unit should remains idle when flash memory goes into busy state. FIFO generates enable signal for enabling the command decoder When valid command is received. Start signal it gives three stored operands to the command decoder named as encoding bits, program address bits and program data bits.

Fig 2, indicates two part of FIFO core that are Write data logic and Read data logic. It is used to control and data received as a input of this module and to assign the write logic module. FIFO core is controlled by Write logic module and it will apply control signals to the FIFO core. Whereas to control FIFO core and for the read operation Read data logic is used. Stored data from the FIFO is read by the read logic module at that time control unit should have to execute a command sequence. Signals necessary for read cycle are applied by FIFO core. To check that whether the FIFO is full or empty, full and empty are the two output pins used by the read logic module. Data-count is another pin which is output pin from

FIFO core. It gives the count of filled memory location of FIFO. [7]

The Xilinx LogiCORE™ IP FIFO Generator is a fully verified first-in first-out (FIFO) memory queue used for applications requiring in-order storage and retrieval. Detailed description of that signals are shown in table no.1

Table 1: FIFO Signals. [12]

Sr. No	Pin	Status	Description
1	Clk	Input	Common clock for read and write.
2	RST_in	Input	Reset
3	Rd_en	Input	Set this pin to 1 and enable read operation.
4	Wr_en	Input	Set this pin to 1 and enable write operation
5	Din	Input	Input data, address and encoding bit
6	Dout	Output	Output data, address and encoding bit
7	Data_count	Output	Output the bits currently present in fifo
8	Empty	Output	Gives 1, if fifo is empty.
9	Full	Output	Gives 1, if fifo is full.
10	Cd_strt	Output	To start command decoder we set this bit to 1

Data sequence from serial protocol is forwarded to din one by one. It is buffered in FIFO by write operation keeping Wr_en=1, and read by keeping Rd_en=1, by read operation it is forwarded to command decoder for further operation. FSM logic is for deciding pattern of buffering of sequence in FIFO. Write 1 takes encoding bit from serial protocol and put in FIFO by making Clk=1, rst=0, Wr_en=1, and Rd_en=0. State of 1st write is ‘0000’. Write 2 takes address from serial protocol and put in FIFO by keeping Clk=1, rst=0, Wr_en=1, and Rd_en=0. State of 2nd write is ‘0001’. Write 3 takes data from serial protocol by keeping Clk=1, rst=0, Wr_en=1, and Rd_en=0. State of 3rd write is ‘0011’. Read 1 reads encoding bit by making Clk=1, rst=0, Wr_en=0, and Rd_en=1. State of 1st read is ‘0100’. Read 2 reads address by making Clk=1, rst=0, Wr_en=0, and Rd_en=1. State of 1st read is ‘0101’. Read 3 reads data by making Clk=1, rst=0, Wr_en=0, and Rd_en=1. State of 1st read is ‘0110’. FIFO gives output as address data and encoding bit separated from each other, after one clocks it writes data in FIFO. But after every 3 clocks it reads data from it.[7]

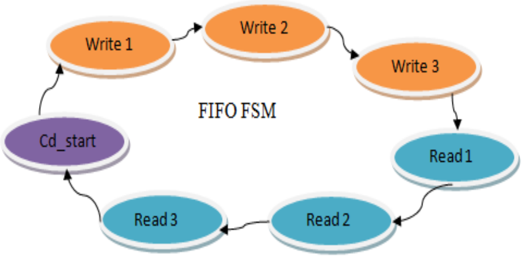

Fig 3: FSM Diagram of FIFO

Algorithm

1. Check for reset. If RST_in = 1, then reset fifo. If no then go to step 2.
2. Execute each step on the edge of clock.
3. Apply signals of write cycles through write logic and store value of encoding bits.
4. Apply signals of write cycles through write logic and store value of address bits
5. Apply signals of write cycles through write logic and store value of data bits.
6. Apply signals of read cycles.
7. Wait for one clock cycle period. Don‘t apply any write or read cycles.
8. Assign dout value to dout_comnd.
9. Apply signals of read cycles.
10. Wait for one clock cycle period. Don‘t apply any write or read cycles.
11. Assign dout value to dout_Addr.
12. Apply signals of read cycles.
13. Wait for one clock cycle period. Don‘t apply any write or read cycles.
14. Assign dout value to dout_data.
15. Enable command decoder to execute the received command.

IV. COMMAND DECODER

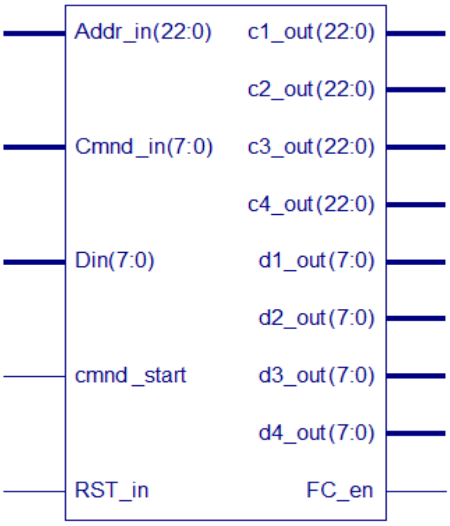

Fig 4: RTL Schematic of Command Decoder.

Command decoder receives encoding bits from FIFO. Command decoder generates the command sequence of address and data cycle as per the specifications of flash memory depending on encoding bits. There are different standard command definitions for the flash memory such as READ/RESET Command, PROGRAM Command and ERASE command. In every standard command sequence encoding bits are used which is used to by forgets the command decoder to know which command is received and which type of address and data sequences are to be generated.[9] For all the design of flash memory controller, Parallel NOR Flash Memory (M29W128GH) is considered in x8 bit mode and its required standard command definitions are mentioned in table 2.In table 3, it shows the encoding bits used for different commands. 0xABh is the start bit for each command. Start bits are united with each command while sending command from interfaced system. Start bits denotes valid command transmission. If command start bits are not linked with received command, then that frame is treated as a invalid and discarded by the controller.[2]

Table 2: Command Definitions [2].

Command Start Bits	Command Encoding Bits	Command
0xABh	0000 0001	PROGRAM
0xABh	0000 0010	READ
0xABh	0000 0011	RESET

Algorithm

1. Check reset.
2. If reset =1, Disable control unit and no any value at the output of command decoder. And go to step 1
3. If reset =0, check for valid command sequence.
4. Verify the address data cycle and signal selected accordingly.
5. Enable control logic for data operation.

Table 3: Address and Data Cycles [2].

Command Name	A1	D1	A2	D2	A3	D3	A4	D4
PROGRAM command	AAA	AA	555	55	AAA	A0	PA	PD
READ command	X	F0	RA	-	-	-	-	-
RESET command	AAA	AA	555	55	X	F0	-	-

V. CONTROL LOGIC

Control logic is actual flash memory controller which accepts input from serial module and gives it to FIFO memory, It receives address and data cycles from command decoder. There are three states for the controller such as idle state, read state, program state. Every command received by controller has specified algorithm. There are different commands for write data or read data. Write data logic and read data logic controls operation of controller.[5]

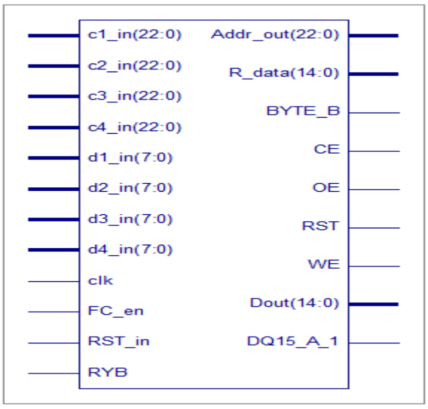

Fig5: RTL Schematic of Control Logic

Algorithm

1. Check reset. If reset =1, go to step 1. If reset =0, check for flash memory ready or not.
2. Assign Address = C1_in (1st address of standard command definition) Assign Data = D1_in (1st address of standard command definition) Disable CE, OE, WE signals.
3. Wait for 20 ns. Apply chip enable (CE) and disable OE signal. Wait for 10 ns.
4. Apply Write enable (WE) signal. Wait for 40 ns.
5. Assign Address = C2_in (2nd address of standard command definition) .Assign Data = D2_in (2nd address of standard command definition). Disable CE, OE, WE signals. Wait for 20 ns.
6. Apply chip enable (CE) signal. Wait for 10 ns.
7. Check for READ command. If it is D1_in = 0xF0h then Enable OE signal and Disable WE signal. If not then Disable OE signal and Enable WE signal. Wait for 10 ns.
8. Assign Address = C1_in (3rd address of standard command definition). Assign Data = D1_in (3rdaddress of standard command definition) .Disable CE, OE, WE signals. Wait for 20 ns.
9. Apply chip enable (CE) signal. Wait for 10 ns, Apply Write enable (WE) signal. Wait for 40 ns.
10. Assign Address = C1_in (4th address of standard command definition) .Assign Data = D1_in (4thaddress of standard command definition) .Disable CE, OE, WE signals. Wait for 20 ns.
11. Apply chip enable (CE) signal. Wait for 10 ns, Apply Write enable (WE) signal. Wait for 40 ns.

VI. RESULT AND DISCUSSIONS

Encoding bits, Program address, program data are given as an input to the control logic module. It generates the signals which are required for flash memory. Encoding bit is responsible for which command sequence is to be generated. Hardware description language is used to implement control logic module and checked for functionality correctness. The code is implemented using VHDL language and XILINX simulation tool is used to simulate the design and generate the waveforms. Numbers of test benches are used for verification of design. Encoding bits, address bits and data bits are input of FIFO module. Write logic and read logic performs operation on FIFO core and execute the operation. One complete command read through the read logic module sent the stored bits to command decoder.[2]

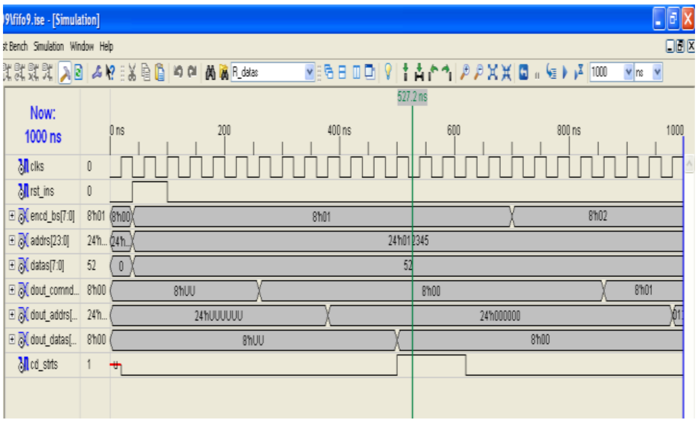

Fig 6: Simulation of FIFO

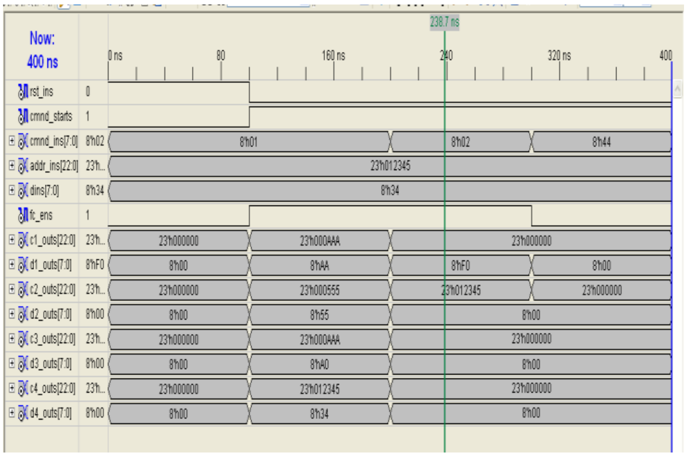

Fig 7: Simulation of Command Decoder

FIFO module sends enable signal for command decoder named as cd_strt. It goes high for a time when encoding bits, address bits; data bits are available at the output of FIFO. Command decoder decodes the incoming command and enables control logic. If valid command is received, because of valid command encoding bits, it generates standard address – data cycles. All address- data cycle values are kept zero for an invalid command. Address – data cycles are inputs for this module. When these address-data cycles are appears, control unit generates necessary signals required by the flash memory. As per timing specifications WE, CE, OE signals are generated. PROGRAM command and second one is the READ commands are discussed in this paper. Control logic module used to generate signals with the reference of timing specification of flash memory chip.[2]

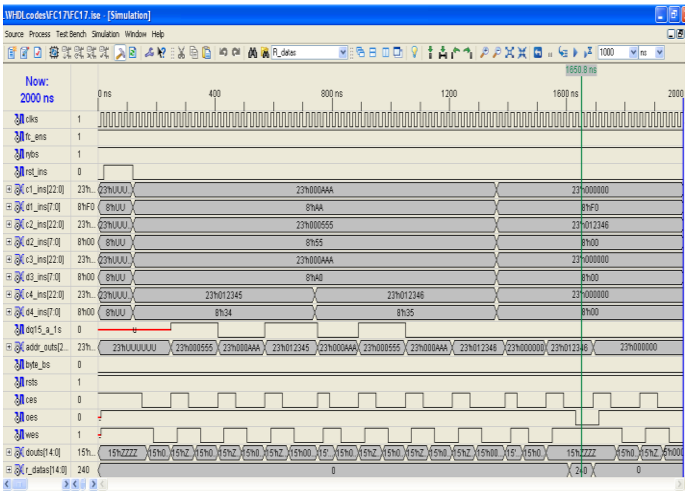

Fig 8: Simulation of Control Logic

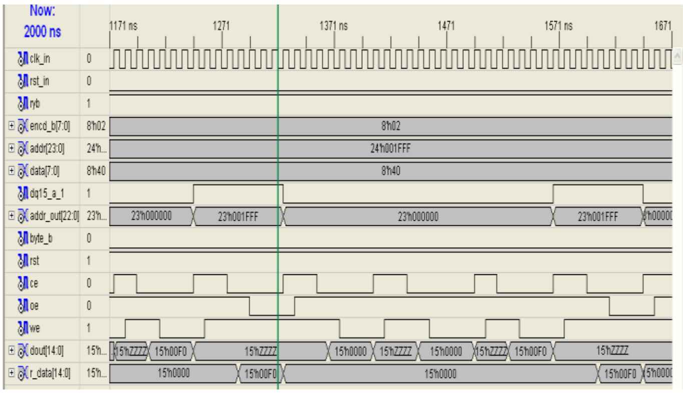

Fig 9: Simulation of READ Command

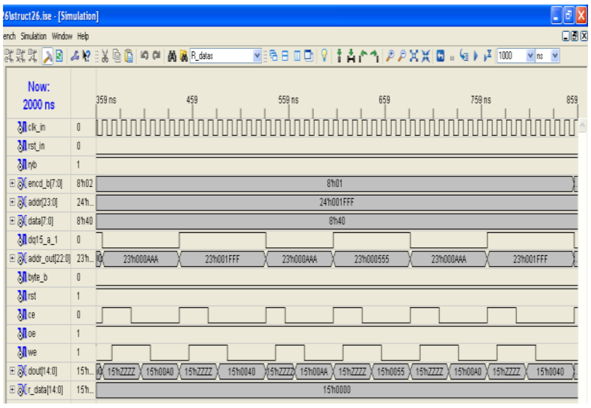

Fig 10: Simulation of PROGRAM Command

VII.CONCLUSION

NOR flash memory controller is implemented in VHDL, is discussed in this paper. When any command is sent through interfaced system it verifies the all logic blocks explained in paper. Serial communication block checks for the valid reception of a command. If valid command is received, it converts incoming bit stream into three inputs which are required by the control logic module. After reception of valid command it is first of all saved into FIFO. Then command decoder checks whether there is any command for execution. If there is command present for the execution, command decoder reads this command from FIFO and generate necessary address-data cycles. Further required signals such as OE WE,CE etc signals are generated as per their timing parameters of address and data cycle. Thus controller executes command received to it. PROGRAM command, READ command are implemented in the flash memory controller.

REFERENCES

[1] NOR/NAND Flash Guide_pdf http://www. Micron .com/products/nor-flash/ parallel-Nor- Flash/ download-guide.

[2] Parallel NOR Flash Embedded Memory M29W128, M29W128GL-datasheethttp:// www. Micron com/Parts/nor-flash/parallel-nor-flash/ m29w128gh70n6e.

[3] Hoeseun Jung, Sanghyuk Jung, Yong Song “Architecture Exploration of Flash Memory Storage Controller through a Cycle Accurate Profiling Proc. IEEE Transactions on Consumer” Electronics,Vol. 57, No. 4, Novembers 2011.

[4] Koichi S E N, Hitoshi Kume, YuzuruOhji—An 80-ns1-Mb” Flash Memory with On-Chip Erase/Erase Verify controller” IEEE Journal of Solid-State Circuits, Vol - 25, NO. 5.

[5] Soya Treesa Jose. Prof. Pradeep.C “Design of a Multichannel NAND Flash Memory Controller for Efficient Utilization of Bandwidth in SSDs”, 978-1-4673-5090-7/13 ©2013 IEEE.

[6] Rajkuwar Kanase, Prof.Shivdas S.S” FPGA Implementation Of Nor Flash Storage Controller” International Journal Of Engineering Sciences & Research Technology, 3(10): October, 2014.

[7] LogiCORE IP FIFO Generator v8.2.pdf www.xilinx.com DS317 June 22, 2011.