Design and Implementation of 8-Bit Read-Write Memory Using FIFO Algorithm

Ranjan Chakraborty

***Abstract*— The term RAM stands for random access memory, which allows easy access at any memory location. Memory can be classified as having its random access of data, but when the term RAM is used with semiconductor memories it is usually meant for read/write operation as opposed to ROM (read only memory). RAM is a volatile form of computer data storage. The word random thus refers to the fact that any piece of data can be returned in a constant time, regardless of its physical location and whether or not it is related to the previous piece of data. Main purpose of the said project is to design and implementation of a 8-bit read and write memory using single clock. The primary technology to be implemented in the design process of the proposed thesis work is First In First Out (FIFO) algorithm based synchronous memory design. FIFO can be classified as synchronous and asynchronous depending on whether same clock or different (asynchronous) clocks control the read and write operations. In this type of memory system, the data that are written into the RAM storage area are read out in the same order that they are written in.**

I. INTRODUCTION

In paper work the concept of Design and Implementation of the 8-bit Read/Write Memory device has been accomplished using the First In First Out (FIFO) algorithm. For this purpose an Address Generation Unit has been devised which will locate the memory locations in the RAM chips used in this design so as to data word (8-bit) to be written in it. Then has been designed 2:1 bus multiplexer unit which is used for the purpose of deciding the input data words to be written, read and again overwritten. The clock input used in this design is global in nature as the same clock is applied to the address generation unit as well as to the clock input of the RAM chips. Then again the same clock is connected with the select line of the bus multiplexer unit, since the first set of data word written into the memory block will pass through the multiplexer during the first half of a complete clock cycle and then the passing of next set of data word through the same multiplexer during the next half of the clock cycle so as to over write the old data in the same memory location, hence this kind of memory is essentially FIFO type, and the select line of the MUX will choose the set of data word to be written and read as per the functioning of the clock applied to the select line. The entire design has been carried out using the Xilinx 7.1i ISE, Xilinx 10.1i softwares and then has been implemented on the Spartan 3 target device.

Author Details: Assistant Professor, Dept. of ECE, Institute of Technology & Marine Engineering, Dist: 24 Parganas (S), West Bengal India, Email: ranjan.chakraborty@ymail.com, Phone: +919433868956

II. THE FIFO (FIRST IN FIRST OUT) ALGORITHM

In computer programming, FIFO (first-in, first-out) is an approach to handling program work requests from queues or stacks so that the oldest request is handled first. In hardware it is either an array of flops or Read/Write memory that store data given from one clock domain and on request supplies with the same data to other clock domain following the first in first out logic. The clock domain that supplies data to FIFO is often referred as WRITE OR INPUT LOGIC and the clock domain that reads data from the FIFO is often referred as READ OR OUTPUT LOGIC [2]. FIFOs are used in designs to safely pass multi-bit data words from one clock domain to another or to control the flow of data between source and destination side sitting in the same clock domain. If read and write clock domains are governed by same clock signal the FIFO is said to be SYNCHRONOUS and if read and write clock domains are governed by different (asynchronous) clock signals FIFO is said to be ASYNCHRONOUS. FIFO full and FIFO empty flags are of great concern as no data should be written in full condition and no data should be read in empty condition, as it can lead to loss of data or generation of non relevant data. The full and empty conditions of FIFO are controlled using binary or gray pointers. Below is the block diagram to depict this basic FIFO operation [3].

III. THE ADDRESS GENERATION UNIT

Here in the design, the Address Generation Unit has been constructed for the purpose of locating various locations in the Random Access Memory chips. The RAM, used here, is of the capacity of storing binary information (data) each of 8 bit width in the memory locations beginning from ‘00000000’ up to the location ‘11111111’, that means 2^8 combinations of memory address can be allocated using the Address Generation unit .To achieve this concept, here an up counter of 8 bit has been derived so as to count from ‘00000000’ till ‘11111111’ such that each and every count sequence may result in the process of locating each memory address in the RAM chip in chronological manner.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. THE DELAYED ADDRESS GENERATION UNIT

Here in the design, the Address Generation Unit has been constructed for the purpose of locating various locations in the Random Access Memory chips. The RAM, used here, is of the capacity of storing binary information (data) each of 8 bit width in the memory locations beginning from '00000000' up to the location '11111111', that means 2^8 combinations of memory address can be allocated using the Address Generation unit .To achieve this concept, here an up counter of 8 bit has been derived so as to count from '00000000' till '11111111' such that each and every count sequence may result in the process of locating each memory address in the RAM chip in chronological manner.

V. THE BUS MULTIPLEXER UNIT

A multiplexer, as it is already known widely, is a combinational electronic circuit which allows only one set of data to be chosen out of multiple data taken one at a time as the input depending upon the unique value selected by the select line(s) and then passes that particular data to the output line. In this design it has been maintained that the multiplexer unit is of the 2X1 type, that means, the multiplexer will select only one set of data out of two as per the sect input value and then it will pass that selected input to the output terminal. Here in this design each of the input terminals is a bus of 8-bit data and so is the output terminal while the select line is controlled by the user. The basic building block of this type of multiplexer is 2:1 MUX of single bit input and output lines. In total 8 numbers of 2:1MUX have been used to materialize the idea of the special type of 2:1 MUX with bus type input and output lines **.**

VI. INTERFACING OF RAM CHIPS WITH DELAYED ADDRESSGENERATION UNIT (CREATION OF MEMORY BLOCK)

The Random Access Memory chip cannot work independently unless it is not properly connected with the address generation unit via the connecting buses. The 8-bit counting sequence generated by the delayed address generation unit by means of the up counter are transferred to the RAM16X4S chip blocks by means of the bus of 8 bit [0:7]. The 8 bit bus is again spit bitwise using the bus taps taken from the available resources in the software used in design. Each of the bus tap takes a single bit out of the every combination of the 8-bit sequence generated and pushes it to the address word locator of each of the RAM chip each of capacity 4-bit and in two chips, it is of 8 bit with every pulse of the clock applied to the clock input terminals of the components in the interfaced structure.

VII. FINAL INTERFACING OF MEMORY BLOCK WITH BUS MULTIPLEXER UNIT

In this part of the work, the Memory Block (Interfaced RAM chips with Delayed Address Generation Unit) is connected to the bus multiplexer unit. The 8-bit input data bus is used to feed a set of data word into the memory block then it is passed to the one of the two buses of the bus multiplexer unit. Another set of 8-bit input data bus is connected directly to the other input terminal of the multiplexer. Now, during a complete global clock cycle, which is connected to the memory block, as well as to the select line of the 2:1 bus multiplexer, in one half of the clock one set of data word is read at the output bus terminal of the MUX unit with a unique state of the select line as per the state of the half clock and then another set of new data word will overwrite and will get read out at the output bus, this entire operation will take place according to the data word (8-bit) which is written first into the memory. Thus the First In First Out operation is accomplished.

VIII. RESULTS AND DISCUSSION

Here, the 'clock' is global to the entire design. The terms 'data_ext [0:7]'and 'data_in [0:7]' denote the bus inputs of the two sets of 8-bit data fed into the Read/Write Memory. The preset and inputs are denoting the state s of the JK flip flops used during the design of delayed address generation unit. Write Enable (we) control is kept high always to enable data word to be written into the memory, 'da[7:0]' denotes the address generated by the delayed address generation unit and at last 'dout[0:7]' denotes final data word output from the memory**.**

IX. FIGURES

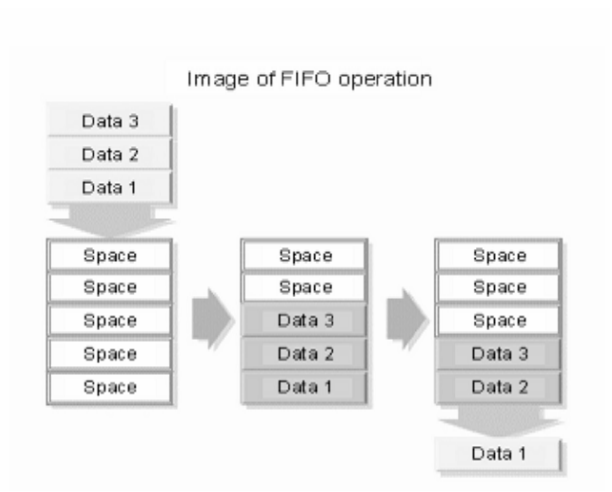


Fig1. *The Basic FIFO Algorithm Technique*

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

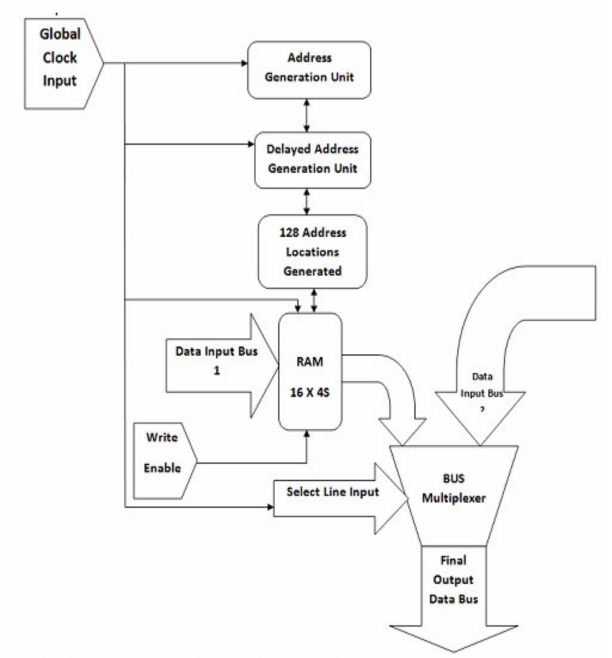


Fig2. *Block Diagram of Complete Design Overview*

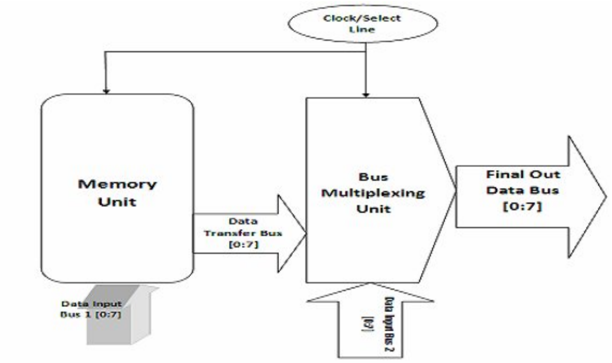


Fig3. *Block Diagram Representation of Final Interfacing*

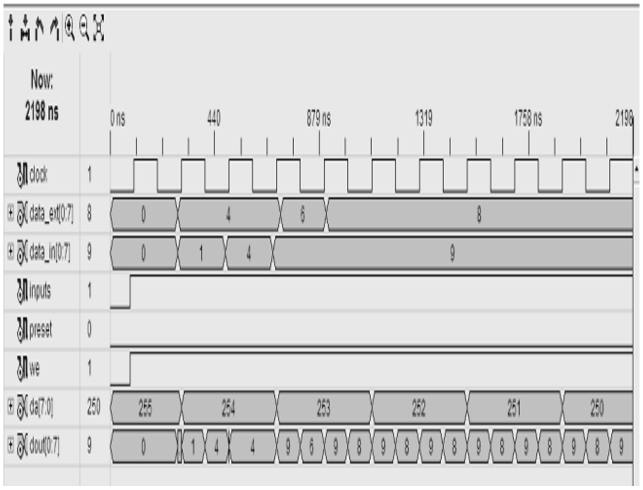
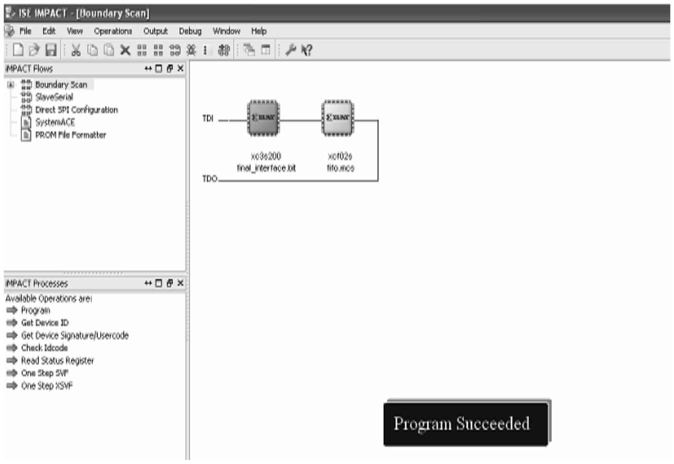
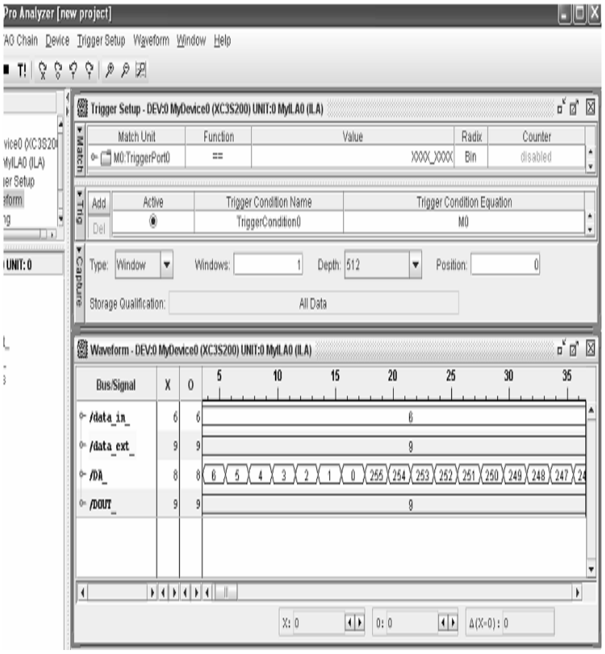


Fig4. *Output of the Final Interfacing Module of the Read/Write Memory using Xilinx7.1i*



Program Succeeded

Fig5. *Output obtained after Implementation on Spartan 3*


Pro Analyzer [new project]
JTAG Chain Device Trigger Setup Waveform Window Help
Trigger Setup - DEV:0 MyDevice0 (XC3S200) UNIT:0 MyILA0 (ILA)
Waveform - DEV:0 MyDevice0 (XC3S200) UNIT:0 MyILA0 (ILA)
Match
Trig
Capture
Add
Del
5 10 15 20 25 30 35
6 5 4 3 2 1 0 255 254 253 252 251 250 249 248 247

Match Unit	Function	Value	Radix	Counter
M0:TriggerPort0	==	XXXX_XXXX	Bin	disabled

Active	Trigger Condition Name	Trigger Condition Equation
●	TriggerCondition0	M0

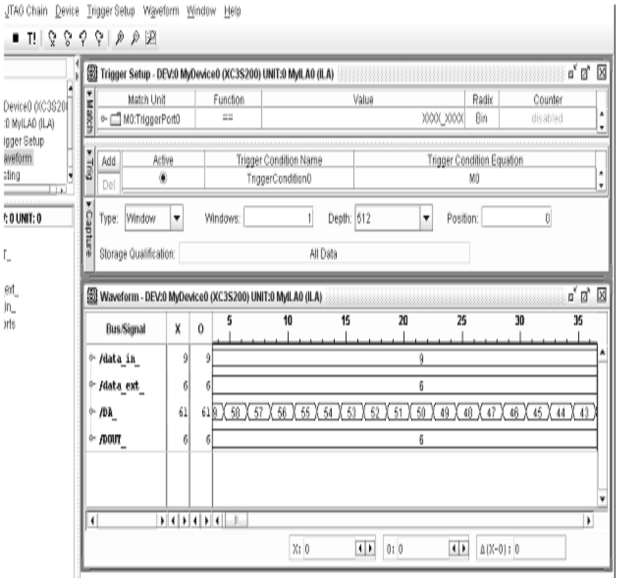
Type: Window Windows: 1 Depth: 512 Position: 0

Storage Qualification: All Data

Bus/Signal	X	0
/data_in_	6	6
/data_ext_	9	9
/DA_	8	8
/DOUT_	9	9

X: 0 O: 0 Δ(X-O): 0

Fig5. *First Output obtained after Implementation on Spartan 3*



Trigger Setup - DEV:0 MyDevice0 (XC3S200) UNIT:0 MyILA0 (ILA)

Match Unit	Function	Value	Radix	Counter
M0:TriggerPort0	==	XXXX_XXXX	Bin	disabled

Active	Trigger Condition Name	Trigger Condition Equation
●	TriggerCondition0	M0

Type: Window Windows: 1 Depth: 512 Position: 0

Storage Qualification: All Data

Waveform - DEV:0 MyDevice0 (XC3S200) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O
/data_in_	9	9
/data_ext_	6	6
/DA_	61	61
/DOUT_	6	6

X: 0 O: 0 Δ(X-O): 0

Fig6. *Second Output obtained after Implementation on Spartan 3*

Here in the above figures it may be noted that, the clock is not shown as the Spartan 3 device always takes its internal clock from PIN # T9, hence the global clock used in the design is not visible here in the waveform.

X. CONCLUSION

Overall what has been discussed in this thesis is the design of a 8-bit First In First Out Memory which may be utilized in various electronic components and gadgets. At first, in the first stage of the design, the RAM 16X8S was utilized but using this type of chip, it was observed that the latency in the output waveform was much higher than expected, so, it was abandoned and instead of this kind of chip two RAM 16X4S chips were introduced in the design but here the numbers of bits were reduced from 16 to 8 as it was not being possible to carry the previous number of bits in this renewed design. Now, the latency in the output is no more and the output waveform is almost perfect in nature.

The First In First Out Memory designed in this paper work is a part of the continuous process of application and there is enough scope to extend further this thesis work beyond the limit of the requirement of the present curriculum. The FIFO memory is essentially built of some logic gates and the most striking feature of any digital circuit is its power consumption. Hence there is always a scope to minimize the power consumption of the constituent parts of the address generation unit by redesigning it using low power techniques. Moreover, this simulation has been primarily done on Xilinx7.1i platform which may be designed using higher versions of the same software, namely, Xilinx11.1i in future course of work.

ACKNOWLEDGEMENT

The author is immensely thankful to Mrs. Depanwita Debnath, Assistant Professor, Department of ECE, Techno India, West Bengal, India for her supervision into the research work and also greatly thankful to Prof. (Dr.) Amlan Chakrabarti, University of Calcutta, India for providing technical guidance and support in the research laboratory at the University of Calcutta, Rajabazar Science College campus, India.

REFERENCES

[1] R J Tocci; “Digital Systems- Principles & Applications”; sixth edition; August, 2001, page: 650

[2] M. Morris Mano; “Computer System Architecture” third edition,;2008, pp : 450-452 & 690

[3] “Design, Verification and Synthesis of Synchronous FIFO”; Manipal Institute of Technology, India; Science Direct ; 2002.