Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 10s (2025)

N-bit Multiplexers Using Verilog

Ponnala Lakshmi Prasanna

Seshadri Rao Gudlavalleru Engineering College, JNTUK

Article History:

Received: ***01/01/2025***

Revised: ***06/02/2025***

Accepted: ***10/03/2025***

Abstract

Multiplexer is a digital circuit with multiple inputs and one output. It is a Multi input single output circuit. Multiplexers uses the notation 2^n X 1 or 2^n : 1 or 2^n to 1. n represents select lines.

The paper n-bit Multiplexers Using Verilog includes three Multiplexers 1-bit 8X1 Multiplexer, 8-bit 8X1 Multiplexer and 8-bit 16X1 Multiplexer. The three Multiplexers are designed, simulated and Synthesized Using Verilog. The Verilog Modules of each Multiplexer are developed and Synthesized to obtain RTL and Technology schematics. In the next step, The verilog Test benches are developed for each Multiplexer and simulated using Behavioral simulation to obtain the Output waveforms. Next The Output waveforms are verified as per the given Truth Tables.

The design summary of each Multiplexer can be obtained after synthesization and simulation. The design summary includes Timing Summary, Device Utilization Summary, Primitive and Black Box Usage and Timing Reports etc.

n-bit Multiplexers Using Verilog can be further implemented for various set of Test Benches. A Test Bench contains different set of input combinations. In future n-bit Multiplexers can be further implemented for increased value of n with multiple Test Bench combinations. n-bit Multiplexers can be designed Using VHDL as well as other HDL languages also and the design can be implemented using Field Programmable Gate Array(FPGA).

Keywords: Mux, Verilog, VHDL,HDL, FPGA, RTL,

1. Introduction

Multiplexer is a circuit which has multiple inputs and one output. Multiplexer is a combinational circuit. A combinational circuit is a digital circuit whose output is purely a function of its current inputs. Combinational circuit has no memory (past inputs). Multiplexers have been divided into two types Analog Multiplexers and Digital Multiplexers. Analog Multiplexers are multiplexers that multiplex analog signals to get the resultant output signal. Analog multiplexers are manufactured by the components such as relays and transistor switches. Digital Multiplexers are multiplexer that accepts digital signals to yield the output. Digital multiplexers are built from standard logic gates, when the multiplexer is used for digital applications. It is called a digital multiplexer.

An analog multiplexer accepts both analog as well as digital inputs, whereas a digital multiplexer accepts only digital inputs.

Multiplexer is a Multi input single output circuit. Multiplexers are represented by the notation 2^n X 1 or 2^n : 1 or 2^n to 1 and consists of n select lines. For example an 8X1 Multiplexer has 3 select lines, 8 inputs and 1 output, similarly 16X1 Multiplexer has 4 select lines 16 inputs and 1 output. Multiplexers are called as data selectors. A multiplexer, sometimes called a selector, allows only one of its data inputs I_m to pass through to the output O. Allowing only one of multiple input tracks to connect to a single output track. If there are m data inputs, then there are $\log_2(m)$ select lines S, and an mX1 multiplexer has m data inputs, one data output. For Example an 8X1 Mux has 8- input lines and 1-output line and 3 select lines.

The binary value of S determines which data input passes through;

- 00...00 means I0 may pass,
- 00...01 means I1 may pass,
- 00...10 means I2 may pass, and so on.

For example, an 8x1 multiplexer has 8 data inputs and thus 3 select lines. If s =111, then I_7 will pass through to the output. So if I_7 were 1, then the output would be 1; if I_7were 0, then the output would be 0. Commonly n-bit multiplexer is used which is a more complex device. Each data input and output, consists of n bits (lines). For Example, a 4-bit 8x1 multiplexer. Thus, if I_7were 0110, then the output would be 0110. n is independent of the number of select lines.

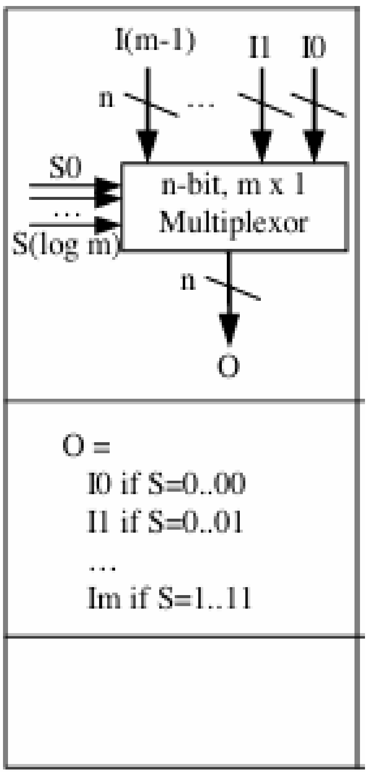


Fig: n-bit mx1 Multiplexer

2. Objective

The objective of the research paper is to develop n-bit Multiplexers: 1-bit 8 to 1 Multiplexer, 8-bit 8X1 Multiplexer and 8-bit 16X1 Multiplexer using Verilog.

To simulate and synthesize the three Multiplexers using Software Xilinx 14.7 and To obtain the corresponding RTL, Technology Schematics, the design summaries, the Output Waveforms for the given Test bench.

3. Methods

1-bit 8 to 1 Multiplexer:-A Multiplexer 8 to 1 multiplexes eight binary inputs into a 1 bit binary output with 3 select lines.

Figure1: Block Diagram of 1-bit 8 to 1 Multiplexer

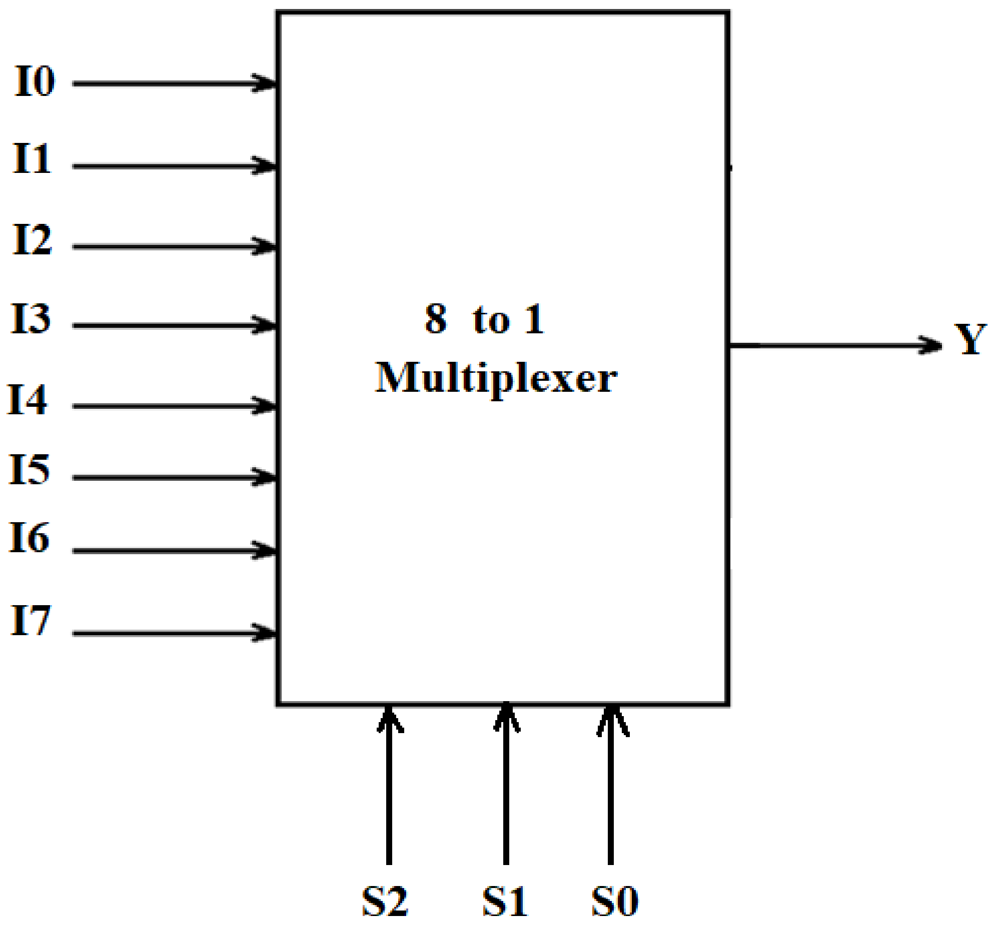


Fig: 8 to 1 Multiplexer

The Block Diagram of 8 to 1 Multiplexer consists of 8 inputs from $I_7 - I_o$ and output Y with select lines $S_2 - S_o$.

Working of 8 to 1 Multiplexer:-

The working of 8 to 1 Multiplexer is as follows,

When the select lines input $S_2S_1S_0 = 000$ and I_o is active high/low ($I_o = 1/0$) then the output $Y = I_o$.

When the select lines input $S_2S_1S_0 = 001$ and I_o is active high/low ($I_1 = 1/0$) then the output $Y = I_1$.

When the select lines input $S_2S_1S_0 = 010$ and I_o is active high/low ($I_2 = 1/0$) then the output $Y = I_2$.

When the select lines input $S_2S_1S_0$ = 011 and I_o is active high/low (I_3 = 1/0) then the output Y = I_3.

When the select lines input $S_2S_1S_0$ = 100 and I_o is active high/low (I_4 = 1/0) then the output Y = I_4.

When the select lines input $S_2S_1S_0$ = 101 and I_o is active high/low (I_5 = 1/0) then the output Y = I_5.

When the select lines input $S_2S_1S_0$ = 110 and I_o is active high/low (I_6 = 1/0) then the output Y = I_6.

When the select lines input $S_2S_1S_0$ = 111 and I_o is active high/low (I_7 = 1/0) then the output Y = I_7.

Truth table and the logic diagram of 8 to 1 Multiplexer are given as follows.

Table1: Truth Table of 8 to 1 Multiplexer

S2	S1	S0	Y
0	0	0	I0
0	0	1	I1
0	1	0	I2
0	1	1	I3
1	0	0	I4
1	0	1	I5
1	1	0	I6
1	1	1	I7

Figure2: Logic Diagram of 8 to 1 Multiplexer

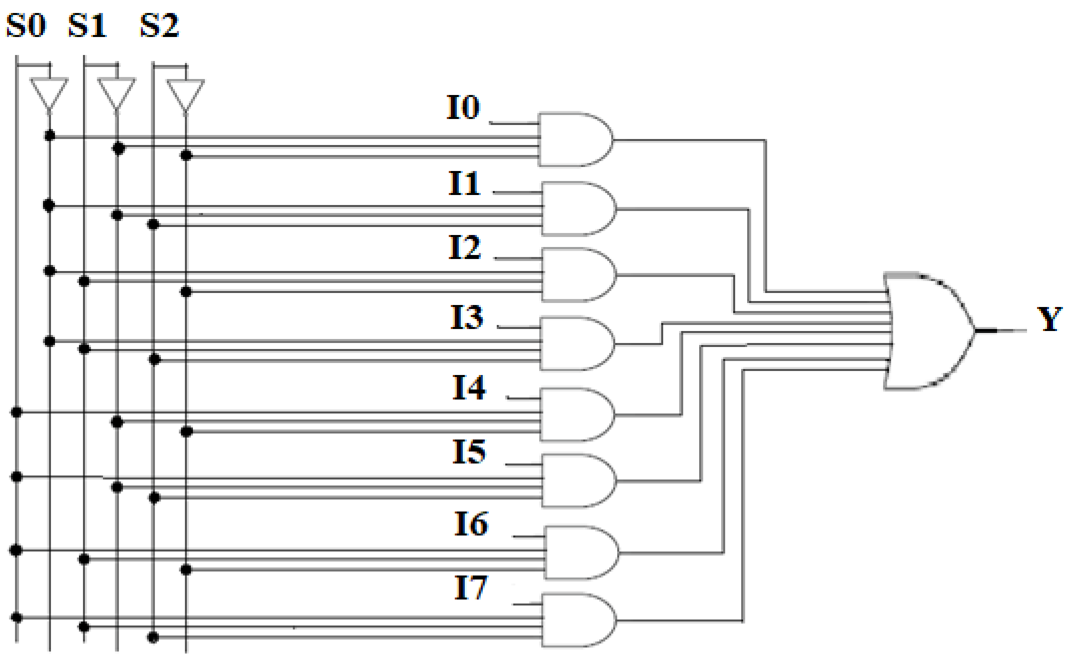


Fig: Logic diagram of 8 to 1 Multiplexer

ISSN: 1074-133X
Vol 32 No. 10s (2025)

Results

Figure3: RTL schematic of 1-bit 8 to 1 Multiplexer

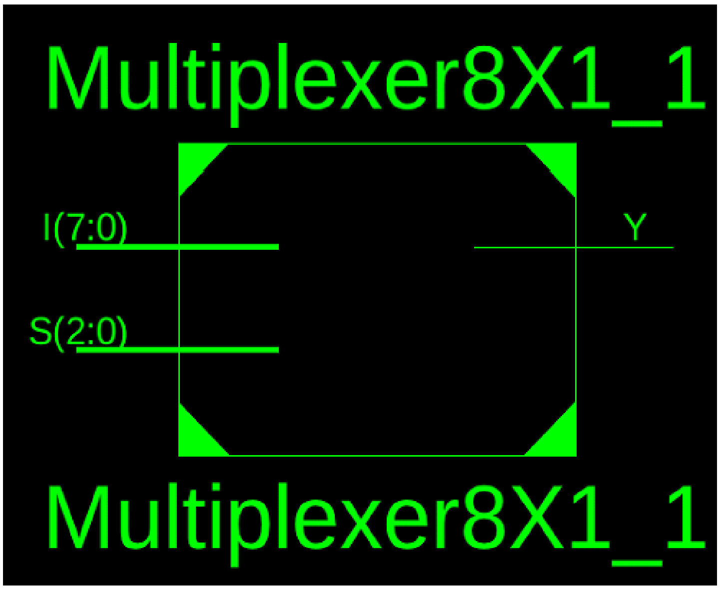


Figure4: Technology schematic of 1-bit 8 to 1 Multiplexer

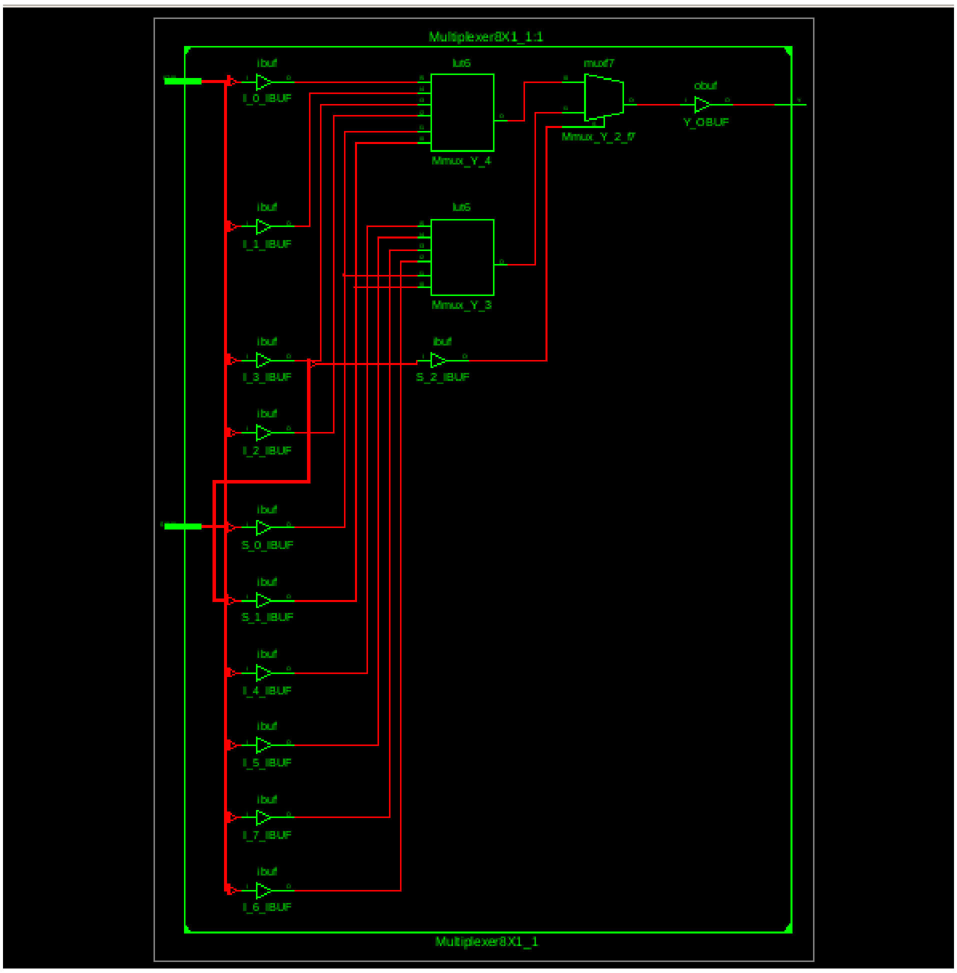
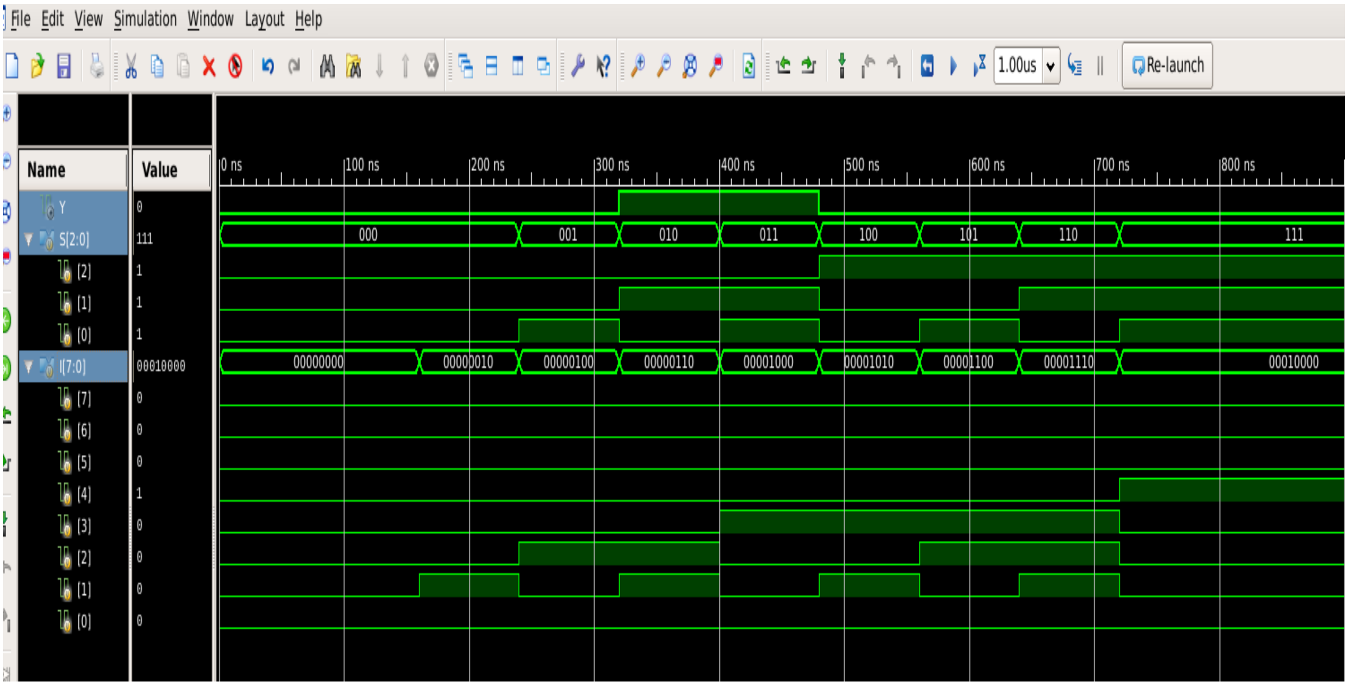


Figure5: Output Waveform of 1-bit 8 to 1 Multiplexer



Discussion of Results:-

From the above output waveform it is clear that, the output Y becomes high based on select lines $S_2S_1S_o$ input and the corresponding inputs I_o to I_7. Table shows the values of select lines and input combinations with the corresponding output.

Table2: Truth Table of 1-bit 8 to 1 Multiplexer with 8 input combinations

S_2	S_1	S_0	$I_7I_6I_5I_4I_3I_2I_1I_0$	Y
0	0	0	00000000	0
0	0	1	00000010	1
0	1	1	00000100	1
1	0	0	00000110	1
1	0	1	00001000	1
1	1	0	00001100	1
1	1	1	00010000	1

Table3: Timing Summary of 1-bit 8 to 1 Multiplexer

Speed Grade: -3

Minimum period	No path found
Minimum input arrival time before clock	No path found
Maximum output required time after clock	No path found
Maximum combinational path delay	1.354ns

Table4: Device utilization summary of 1-bit 8 to 1 Encoder

Selected Device: 7a100tcsg324-3

Slice Logic Utilization:	
Number of Slice LUTs:	2 out of 63400 0%
Number used as Logic:	2 out of 63400 0%
Slice Logic Distribution:	
Number of LUT Flip Flop pairs used	2
Number with an unused Flip Flop	2 out of 2 100%
Number with an unused LUT	0 out of 2 0%
Number of fully used LUT-FF pairs	0 out of 2 0%
Number of unique control sets	0
IO Utilization:	
Number of IOs	12
Number of bonded IOBs	12 out of 210 5%

Table5: Primitive and Black Box Usage of 1- bit 8 to 1 Multiplexer

#BELS	3
#LUT6	2
#MUXF7	1
# IO Buffers	12
# IBUF	11
#OBUF	1

8-bit 8 to 1 Multiplexer:-

An 8-bit 8 to 1 Multiplexer multiplexes eight 8-bit binary inputs into an 8 bit binary output with 3 select lines.

Figure6: Block Diagram of 8-bit 8 to 1 Multiplexer

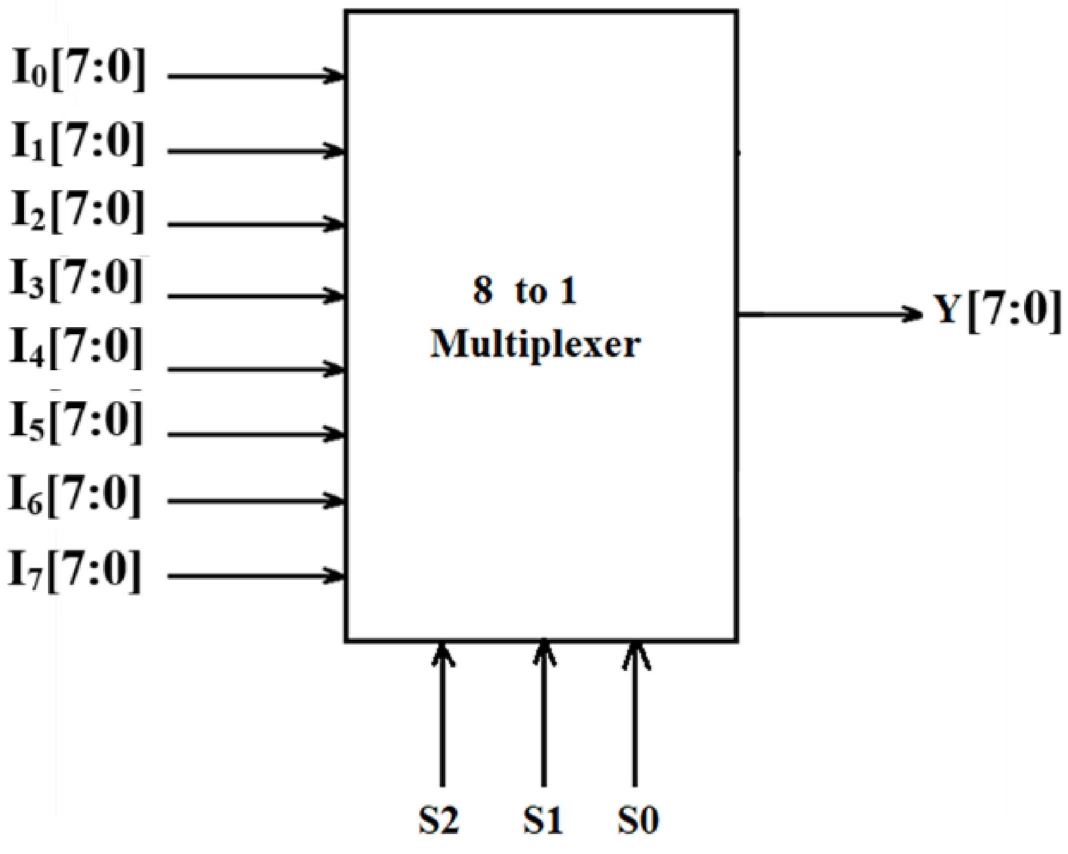


Fig: 8-bit 8 to 1 Multiplexer

The Block Diagram of 8-bit 8 to 1 Multiplexer consists of 8 inputs from $I_7 – I_o$ and 8-bit output Y with select lines $S_2 – S_o$.

Working of 8-bit 8 to 1 Multiplexer:-

The working of 8-bit 8 to 1 Multiplexer is as follows,

When the input $S_2S_1S_0$ = 000 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 00000110 then the output Y = 00000110

When the input $S_2S_1S_0$ = 001 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 00101010 then the output Y = 00101010

When the input $S_2S_1S_0$ = 010 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 01000000 then the output Y = 01000000

When the input $S_2S_1S_0$ = 011 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 00010010 then the output Y = 00010010

When the input $S_2S_1S_0$ = 100 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 01111000 then the output Y = 01111000

When the input $S_2S_1S_0$ = 101 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 11011000 then the output Y = 11011000

When the input $S_2S_1S_0$ = 110 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 10010100 then the output Y = 10010100

When the input $S_2S_1S_0$ = 111 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 01110100 then the output Y = 01110100.

Table6:-Truth Table of 8-bit 8 to 1 Multiplexer

The Truth Table of 8-bit 8 to 1 Multiplexer is given in the following Table. It has 8-bit inputs I_0- I_7 and three select lines S_2 –S_0 and 8-bit output Y. Based on select lines input corresponding input flows into output Y. For Example $S_2S_1S_0$ =100 then Y=01111000 and for $S_2S_1S_0$ =111 then Y=01110100 and so on.

I_i[7:0]	S_2	S_1	S_0	Y[7:0]
00000110	0	0	0	00000110
00101010	0	0	1	00101010
01000000	0	1	0	01000000
00010010	0	1	1	00010010
01111000	1	0	0	01111000
11011000	1	0	1	11011000
10010100	1	1	0	10010100
01110100	1	1	1	01110100

Results

Figure7: RTL schematic of 8-bit 8 to 1 Multiplexer

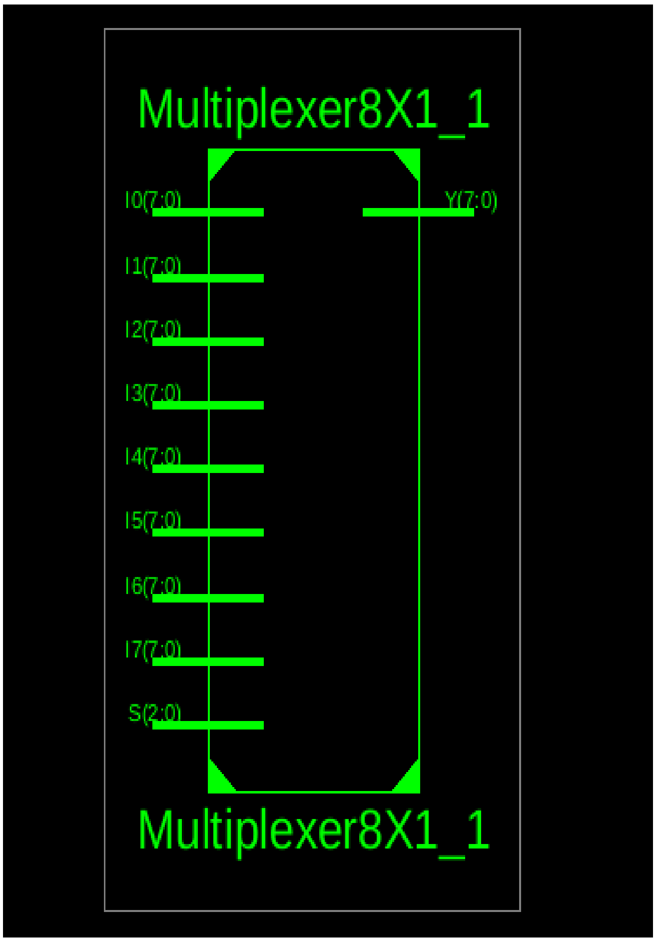
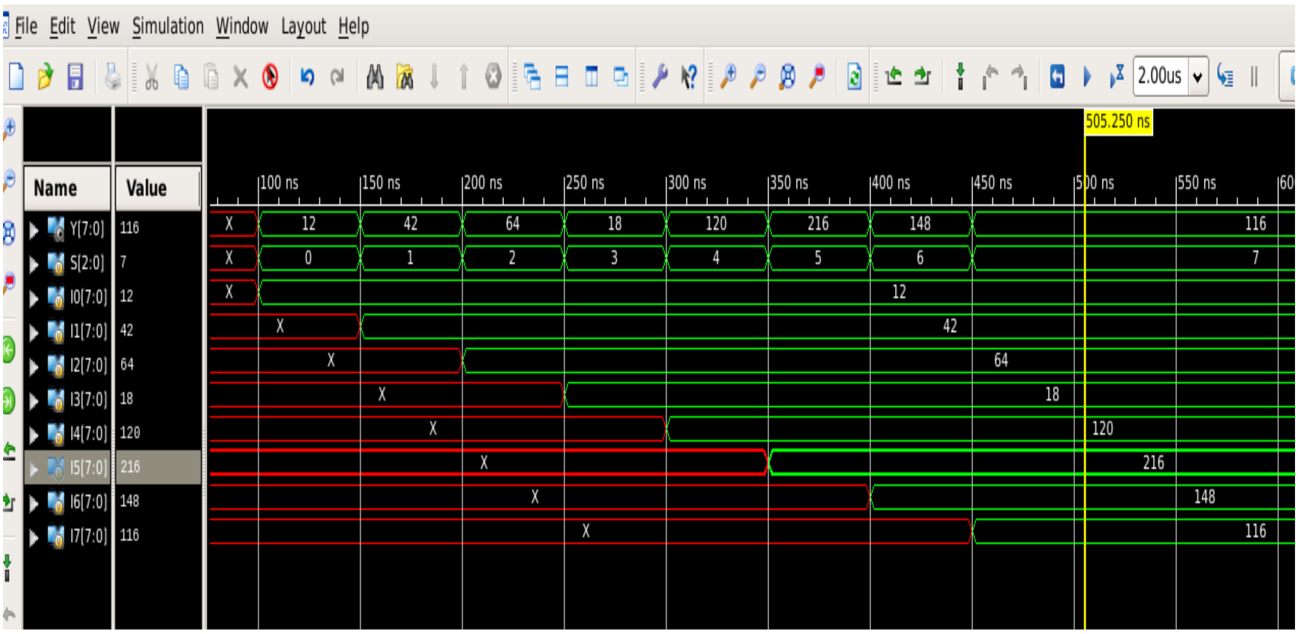


Figure8: Output Waveform of 8-bit 8 to 1 Multiplexer



Discussion of Results:-

The output waveforms are shown in Decimal values. From the above output waveform it is clear that, the output Y becomes one of the inputs I_o to I_7 based on select lines $S_2S_1S_o$. Table shows the values of select lines and input combinations with the corresponding output. For Example $S_2S_1S_0$ =100 then Y=120, $S_2S_1S_0$ =101 then Y= 216 , $S_2S_1S_0$ =110 then Y= 148 and for $S_2S_1S_0$ =111 then Y= 116 and so on.

Table7: Truth Table of 8-bit 8 to 1 Multiplexer with inputs $I_0 – I_7$.

I_i[7:0]	S_2	S_1	S_0	Y[7:0]
I_0 -12	0	0	0	12
I_1-42	0	0	1	42
I_2-64	0	1	0	64
I_3-18	0	1	1	18
I_4-120	1	0	0	120
I_5-216	1	0	1	216
I_6-148	1	1	0	148
I_7-116	1	1	1	116

Table8: Timing Summary of 8-bit 8 to 1 Multiplexer

Speed Grade: -3

Minimum period	No path found
Minimum input arrival time before clock	No path found
Maximum output required time after clock	No path found
Maximum combinational path delay	1.418ns

Table9: Device utilization summary of 8-bit 8 to 1 Multiplexer

Selected Device: 7a100tcsg324-3

Slice Logic Utilization:	
Number of Slice LUTs:	16 out of 63400 0%
Number used as Logic:	16 out of 63400 0%
Slice Logic Distribution:	
Number of LUT Flip Flop pairs used	16
Number with an unused Flip Flop	16 out of 16 100%
Number with an unused LUT	0 out of 16 0%
Number of fully used LUT-FF pairs	0 out of 16 0%
Number of unique control sets	0
IO Utilization:	
Number of IOs	75
Number of bonded IOBs	75 out of 210 35%

Table10: Primitive and Black Box Usage of 8-bit 8 to 1 Multiplexer

#BELS	24
#LUT6	16
#MUXF7	8
# IO Buffers	75
# IBUF	67
#OBUF	8

8-bit 16 to 1 Multiplexer:-

An 8-bit 16 to 1 Multiplexer multiplexes eight 8-bit binary inputs into an 8 bit binary output with 4 select lines.

Figure9: Block Diagram of 8-bit 16 to 1 Multiplexer

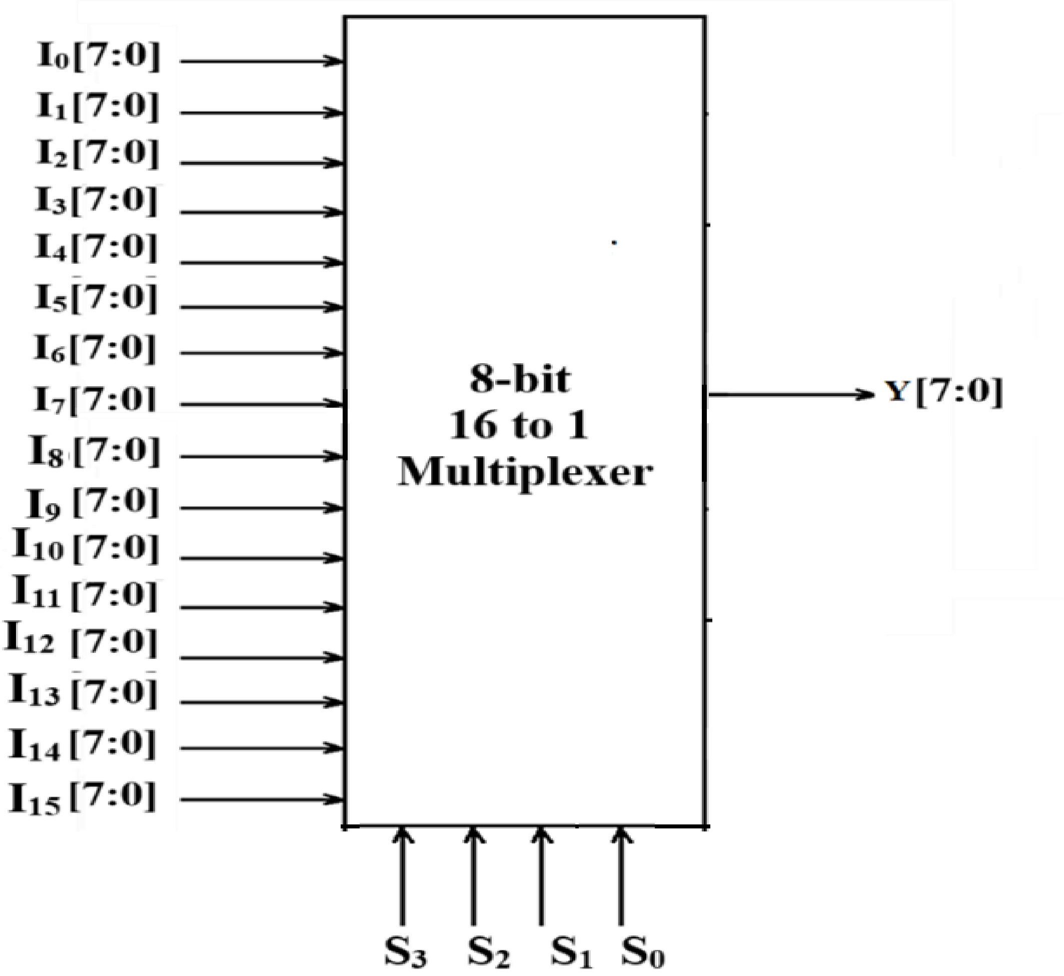


Fig: 8-bit 16 to 1 Multiplexer

The Block Diagram of 8-bit 16 to 1 Multiplexer consists of 8 inputs from $I_7 - I_o$ and 8-bit output Y with select lines $S_3 - S_o$.

Working of 8-bit 16 to 1 Multiplexer:-

The working of 8-bit 16 to 1 Multiplexer is as follows,

When the input $S_3S_2S_1S_0$ = 0000 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 11100100 then the output Y = 11100100

When the input $S_3S_2S_1S_0$ = 0001 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 11001100 then the output Y = 11001100

When the input $S_3S_2S_1S_0$ = 0010 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 01110100 then the output Y = 01110100

When the input $S_3S_2S_1S_0$ = 0011 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 11100000 then the output Y = 11100000

When the input $S_3S_2S_1S_0$ = 0100 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 01101010 then the output Y = 01101010

When the input $S_3S_2S_1S_0$ = 0101 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 01101000 then the output Y = 01101000

When the input $S_3S_2S_1S_0$ = 0110 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 10111100 then the output Y = 10111100

When the input $S_3S_2S_1S_0$ = 0111 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 10000100 then the output Y = 10000100

When the input $S_3S_2S_1S_0$ = 1000 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 10110000 then the output Y =10110000

When the input $S_3S_2S_1S_0$ = 1001 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 10100100 then the output Y =10100100

When the input $S_3S_2S_1S_0$ = 1010 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 11000000 then the output Y = 11000000

When the input $S_3S_2S_1S_0$ = 1011 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 01110000 then the output Y = 01110000

When the input $S_3S_2S_1S_0$ = 1100 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 10000000 then the output Y = 10000000

When the input $S_3S_2S_1S_0$ = 1101 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 01101100 then the output Y = 01101100

When the input $S_3S_2S_1S_0$ = 1110 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 10011000 then the output Y = 10011000

When the input $S_3S_2S_1S_0$ = 1111 and $I_7I_6I_5I_4I_3I_2I_1I_o$ = 10110100 then the output Y = 10110100

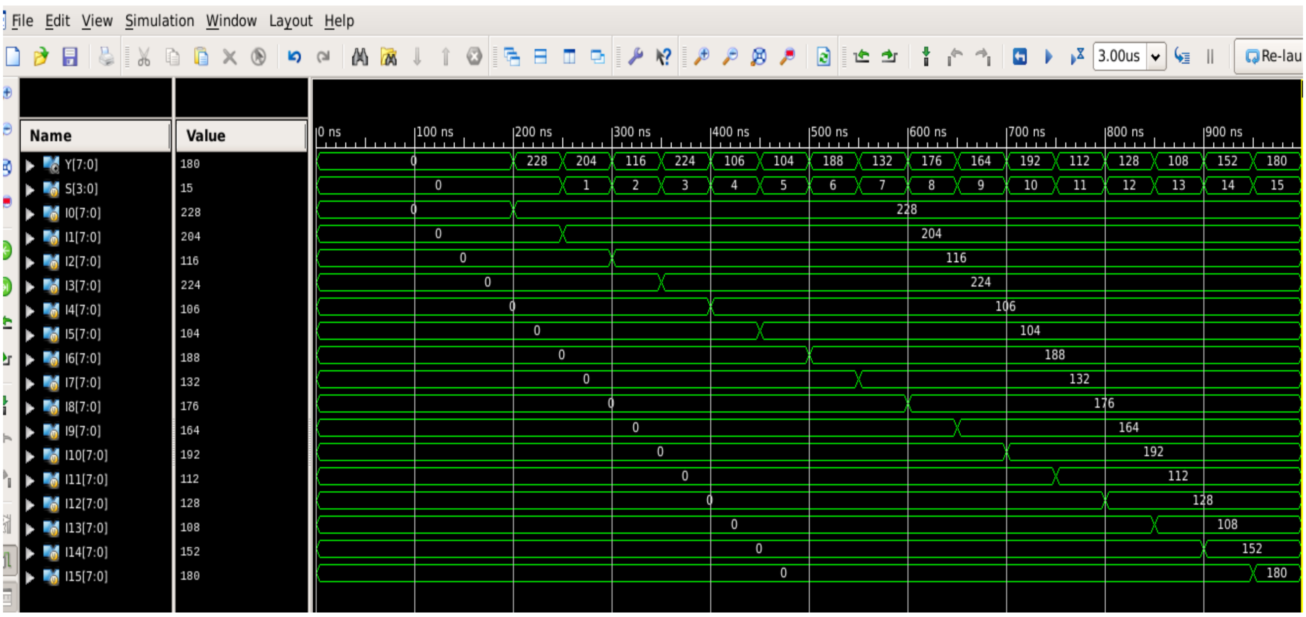
Table11:-Truth Table of 8-bit 16 to 1 Multiplexer

I_i[7:0]	S_3	S_2	S_1	S_0	Y[7:0]
11100100	0	0	0	0	11100100
11001100	0	0	0	1	11001100
01110100	0	0	1	0	01110100
11100000	0	0	1	1	11100000
01101010	0	1	0	0	01101010
01101000	0	1	0	1	01101000
10111100	0	1	1	0	10111100
10000100	0	1	1	1	10000100
10110000	1	0	0	0	10110000
10100100	1	0	0	1	10100100
11000000	1	0	1	0	11000000
01110000	1	0	1	1	01110000
10000000	1	1	0	0	10000000
01101100	1	1	0	1	01101100
10011000	1	1	1	0	10011000
10110100	1	1	1	1	10110100

Results

Figure10: RTL schematic of 8-bit 16 to 1 Multiplexer

Figure11: Output Waveform of 8-bit 16 to 1 Multiplexer



Discussion of Results:- From the above output waveform it is clear that, the output Y becomes one of the inputs I_o to I_{15} based on select lines $S_3S_2S_1S_o$. Table shows the values of select lines and input combinations with the corresponding output.

Table12: Truth Table of 8-bit 16 to 1 Multiplexer with inputs $I_0 - I_{15}$.

$I_i[7:0]$	S_3	S_2	S_1	S_0	Y[7:0]
I0=228	0	0	0	0	228
I1=204	0	0	0	1	204
I2=116	0	0	1	0	116
I3=224	0	0	1	1	224
I4=106	0	1	0	0	106
I5=104	0	1	0	1	104
I6=188	0	1	1	0	188
I7=132	0	1	1	1	132
I8=176	1	0	0	0	176
I9=164	1	0	0	1	164
I10=192	1	0	1	0	192
I11=112	1	0	1	1	112
I12=128	1	1	0	0	128
I13=108	1	1	0	1	108
I14=152	1	1	1	0	152
I15=180	1	1	1	1	180

Table13: Timing Summary of 8-bit 16 to 1 Multiplexer

Speed Grade: -3

Minimum period	No path found
Minimum input arrival time before clock	No path found
Maximum output required time after clock	No path found
Maximum combinational path delay	1.674ns

Table14: Device utilization summary of 8-bit 16 to 1 Multiplexer

Selected Device: 7a100tcsg324-3

Slice Logic Utilization:	
Number of Slice LUTs:	32 out of 63400 0%
Number used as Logic:	32 out of 63400 0%
Slice Logic Distribution:	
Number of LUT Flip Flop pairs used	32
Number with an unused Flip Flop	32 out of 32 100%
Number with an unused LUT	0 out of 32 0%
Number of fully used LUT-FF pairs	0 out of 32 0%
Number of unique control sets	0
IO Utilization:	
Number of IOs	140
Number of bonded IOBs	140 out of 210 66%

Table15: Primitive and Black Box Usage of 8-bit 16 to 1 Multiplexer

#BELS	56
#LUT6	32
# MUXF7	16
# MUXF8	8
# IO Buffers	140
# IBUF	132
# OBUF	8

Conclusions

Thus n-bit Multiplexers are synthesized and simulated Using Verilog with different Test Benches and the Results are obtained.

Future scope

In future n-bit Multiplexers can be further implemented for increased value of n with multiple Test Bench combinations. n-bit Multiplexers can be designed Using VHDL as well as other HDL languages also and the design can be implemented using Field Programmable Gate Array(FPGA).

Refrences

[1] Cihun-Siyong Alex Gong, Muh-Tian Shiue, Ci-Tong Hong, and KaliWen Yao, "Analysis and Design of an Efficient Irreversible Energy Recovery Logic in 0.18m CMOS," *IEEE Trans. Circuits Syst. I*, vol. 55, No. 9, pp. 2595–2607, Oct. 2008.

[2] Hirokatsu Shirahama and Takahiro Hanyu, "Design of High-Performance Quaternary Adders Based on Output-Generator Sharing ", *Proceedings of the 38th International Symposium on Multiple Valued Logic*, pp. 8–13. 2008.

[3] S.M. Basir, G.A. Mhdiraji, A. Malekmohammadi, N.H. Ibrahim, A.F. Abas and M.K. Abdullah, "Demonstration of Duty Cycle Division Multiplexing with bit error rate estimation," in *2009 IEEE 9th Malaysis International Conference on Communications (MICC)*, 2009.

[4] S. Vijayakumar and B. Karthikeyan, "Mixed style of Low Power Multiplexer Design for Arithmetic Architectures using 90nm Technology ", *Proceedings of the 12th international conference on Networking, VLSI and signal processing (ICNVS)*, pp. 83–87, February 2010.

[5] Ji-Seop Song, Sang-Hyuk Yang, Eung-ju Kim, Shin-II Lim and Suki Kim, "A High Speed and Low Power 4:1 Multiplexer Architecture for Graphic Memory Interface " accepted to 2010 International Conference on Electronics, Information, and Communication.

[6] Vasundara Patel K.S., K.S. Gurumurthy, "Arithmetic operations in multi-valued logic ", *International journal of VLSI design and communication system (VLSICS)*, vol. 1, no. 1, pp. 21–32, March 2010.

[7] Anindya Dal, Ifat Jahangir and Masud Hasan, "Design of Quaternary Serial and Parallel Adders ", *ICECE 2010, 6th International Conference on Electrical and Computer Engineering*, 18-20 December 2010.

[8] Y. Zhang, L. Okamura, and T. Yoshihara, "An energy efficiency 4-bit multiplier with two-phase non-overlap clock driven charge recovery logic," *Electronics, IEICE Transactions on*, vol. E94-C, no. 4, pp. 605–612, April 2011.

[9] J.C. Kao, W.H. Ma, V.S. Sathe, and M. Papaefthymiou, "Energy efficient low-latency 600 mhz fir with high-overdrive charge recovery logic," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–12, 2011.

[10] A. E. Brouwer & W. H. Haemers, *Spectra of Graphs*. New York, NY : Springer, 2012.

[11] Mukesh Tiwari, Jai Karan Singh, Yashasvi Vaidhya, "Adiabatic Positive Feedback Charge Recovery Logic For Low Power CMOS Design ", *IJCTEE*, Volume 2, Issue 5, October 2012.

[12] B. Dilli Kumar, M. Bharathi, "Design of Energy Efficient Arithmetic Circuits Using Charge Recovery Adiabatic Logic ", *International Journal of Engineering Trends and Technology*, Volume 4, Issue 1, 2013.

[13] Varun Singh, Maan Singh, Tarun Arora, Ankit Kaushik and Praveer Saxena, "Design Of 2:1 Multiplexer Using Two Phase Drive Adiabatic Dynamic CMOS Logic For Low Power Applications ", *International Journal of Engineering Research and Apllications (IJERA)*, vol. 3, Issue 1, pp. 1582–1590, January–February 2013.

[14] Deepti Shinghal, Amit Saxena, Arti Noor, "Adiabatic Logic Circuits: A Retrospect ", *MIT International Journal of Electronics and Communication Engineering*, vol. 3, No. 2, pp. 108–114, August. 2013.

[15] Arun Kumar, Manoj Sharma, "Design and Analysis of Multiplexer using Adiabatic techniques ECRL and PFAL ", International Conference On Advances In Computing, Communications And Informatics (ICACCI), 2013.

[16] "*Nonelectronic Parts Reliability Data Publication (NPRD-2016)*," Quanterion Solutions Incorporated 2016.

[17] Saumya Pandey and Dr. Nidhi Goel "A Power Efficient 2:1 Multiplexer for Low Power VLSI Application," *IJARECE*, Volume 5, Issue 1, January 2016.

[18] R. Diestel, *Graph Theory*. Berlin, Germany : Springer, 2017.

[19] Swapnil S. Patil, Sagar S. Pathak, Rahul R. Kathar and D. S. Patil "Low Power Based Dual Mode Logic gates using Power gating Technique," *IRJET*, 2017, pp. 1462.

[20] E. Dertien, P. Regtien, *Sensors for Mechatronics*. Elsevier, 2018.

[21] A. Anitha, S. Rooban and M. Sujatha "Implementation of Energy Efficient Gate using Adiabatic Logic for Low Power Applications," *IJRTE*, Volume- 8, Issue 3, September – 2019.

[22] Pujit Kumar Nahata, Ashad Ahmed, Shivkant Yadav, Nivedita Nair, Sanmukh Kaur, "All optical Full-Adder and Full-Subtractor using Semiconductor Optical Amplifiers and All-Optical Logic Gates," 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN).

[23] Nivedita Nair, Sanmukh Kaur, Hardeep Singh, "All-optical integrated 2-to-4 decoder and 1-to-4 demultiplexer circuit with enable using SOA based MZI." February 14, 2020 ; accepted August 17, 2020 ; published online October 14, 2020.

[24] Manohari Ramachandran, Shanthi Prince Archana Kambham and Naga Maruthi, "Design and simulation of all optical shift registers using D flip-flop ", *Microw Opt Technol Lett.*, 62 (7), 1 - 12, (2020).

[25] F. P. Lanter, A. Sutinjo, and J. Morgan, "Ionospheric measurement using a linearly polarized tri-band small satellite beacon," *Submitted to Advances in Space Research*, 1 2022.