Realizing Multiplexers and Comparators in structural and RTL Verilog

Lab Report No. 2

MV407: IC Design Lab-1 (Digital)

Submitted by:

Full Name	**Enrollment No.**
Abhinav M	23PMMT11(@uohyd.ac.in)

Lab Instructor: Dr. Bhawna Gomber

Teaching Assistant: Ms. Bhargavi

Center for Advanced Studies in Electronics Science and Technology
University of Hyderabad
Hyderabad, India
October 4, 2023

Contents

1 | Introduction

This lab report details the implementation of comparators and multiplexers in Verilog Hardware Description Language(HDL) in structural and Register Transfer Level (RTL) / Dataflow paradigms. The code was written and synthesized on Xilinx Vivado v2020 on a Zynq 7000 xc7c020clg484-1 platform with 53200 Look Up Tables (LUTs) and 400 Input Output Blocks (IOBs). This report is a part of the MV407 course at CASEST, University of Hyderabad (2023).

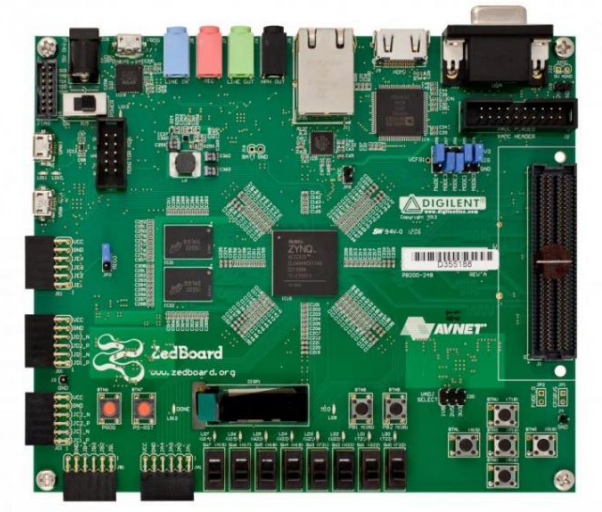

Figure 1.1: Zedboard: A common FPGA platform powered by the Zynq 7000 family.

2 | 4 bit 2:1 Multiplexer

04-10-2023

2.1 | Objectives

- To design a 4 bit, 2:1 Multiplexer using structural and RTL Verilog and verify its functionality using a testbench.

2.2 | Schematic

4 bit 2x1 mux works in the same way as its 1 bit counterpart only difference being that the inputs and hence output will be 4 bit words. Fig 2.1 shows the block diagram where A_0 and A_1 are 4-bits each.

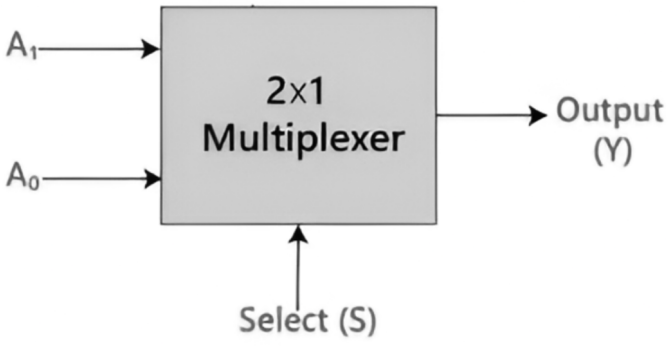

Figure 2.1: Block diagram of the 2:1 MUX.

2.3 | Truth table

The truth table can be used to derive the Boolean expression:

Input 1	Input 2	S (Select)	Y (Output)
A_0	A_1	0	A_0
A_0	A_1	1	A_1

2.4 | Code

2.4.1 | Structural Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 28.09.2023 07:30:18
// Design Name:
// Module Name: BS_4_MUX21
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module BS_4_MUX21(output [3:0]data,input s,input [3:0]a,b);

mux_2_1 m[3:0](s,a,b,data); //used 1 bit 2x1 mux to make 4 bit 2x1 mux

endmodule
```

2.4.2 | Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 01.10.2023 22:50:56
// Design Name:
// Module Name: tb_BS_4_MUX21
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_BS_4_MUX21();
wire [3:0] data;
reg s;
reg [0:3]a,b;
BS_4_MUX21 uut(data,s,a,b);

initial
    begin

        a=4'b0001;b=4'b1111;
        s=1'b0;
    #10 s=1'b1;


    #10 $finish;
    end
```

```

endmodule
```

2.4.3 | RTL Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 28.09.2023 07:30:46
// Design Name:
// Module Name: BR_4_MUX21
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module BR_4_MUX21(output [3:0]data,input s,input [3:0]a,b);

    assign data=s?a:b;

endmodule
```

2.4.4 | Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 01.10.2023 22:18:47
// Design Name:
// Module Name: tb_BR_4_MUX21
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_BR_4_MUX21();
wire [0:3]data;
reg s;
reg [0:3]a,b;
BR_4_MUX21 uut(data, s,a,b);

     initial
    begin

     a=4'b0001;b=4'b1000;

     s=1;
    #10 s=0;

    #10 $finish;
    end


endmodule
```

2.5 | Elaborated design

2.5.1 | Structural

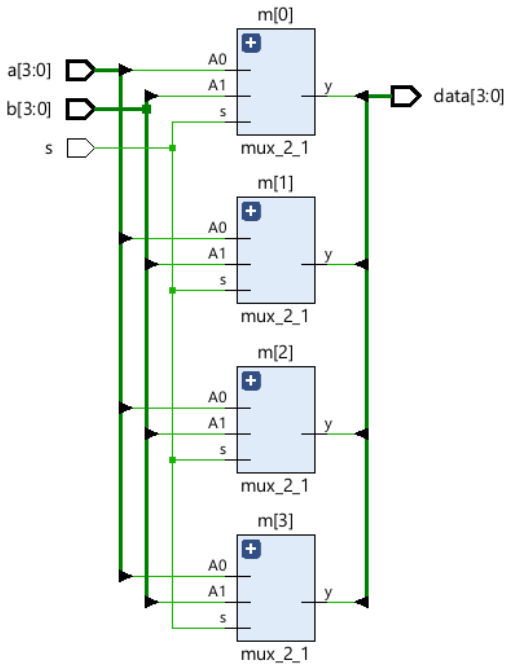

Figure 2.2: Elaborated design of 4 bit 2x1 mux using 1 bit 2x1 muxes.

2.5.2 | RTL

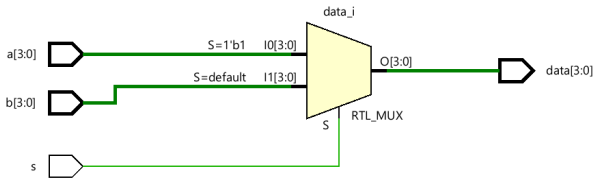

Figure 2.3: Elaborated design of 4 bit 2x1 mux in RTL style.

2.6 | Timing diagram

2.6.1 | Structural

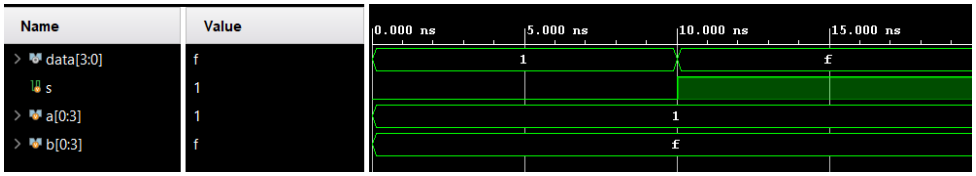

Figure 2.4: Timing diagram of the 4 bit 2x1 mux obtained after simulating the testbench.

2.6.2 | RTL

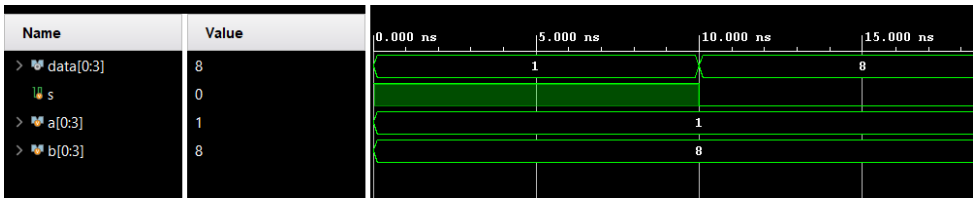

Figure 2.5: Timing diagram of the 4 bit 2x1 mux obtained after simulating the testbench.

2.7 | Utilization summary

2.7.1 | Structural

Resource	Utilization	Available	Utilization %
LUT	2	53200	0.00
IO	13	200	6.50

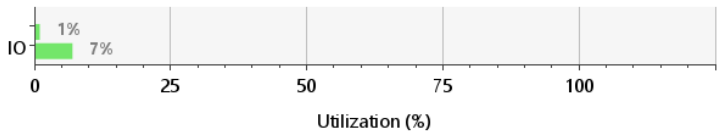

Figure 2.6: Resource utilization summary of the synthesized design.

2.7.2 | RTL

Resource	Utilization	Available	Utilization %
LUT	2	53200	0.00
IO	13	200	6.50

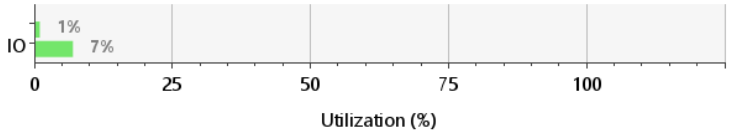

Figure 2.7: Resource utilization summary of the synthesized design.

2.8 | Remarks

The design and testing of 4 bit 2x1 mux in structural (using 1 bit 2x1 mux) and RTL Verilog has been carried out. The device is showing expected behaviour.

3 | 4bit, 4:1 Multiplexer by instantiating 2:1 MUX

04-10-2023

3.1 | Objectives

- To use the designed multiplexer in 2 to design a 4bit, 4:1 Multiplexer by instantiating as well as RTL and verify its functionality using a testbench.

3.2 | Schematic

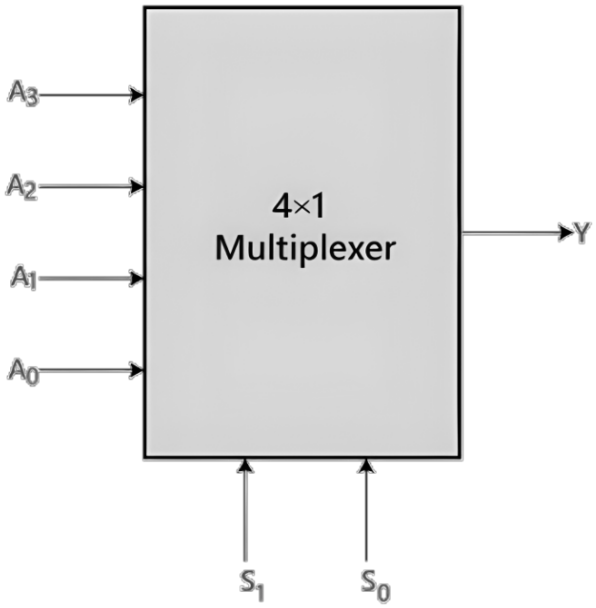

Figure 3.1: Block diagram of the full adder.

3.3 | Truth table

The truth table of the 4x1 mux is shown below.

Input 1	Input 2	Input 3	Input 4	S_0	S_1	Y (Output)
A_0	A_1	A_2	A_3	0	0	A_0
A_0	A_1	A_2	A_3	0	1	A_1
A_0	A_1	A_2	A_3	1	0	A_2
A_0	A_1	A_2	A_3	1	1	A_3

Table 3.1: Truth table for the 4x1 mux.

3.4 | Code

3.4.1 | Structural Design

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company: CASEST, University of Hyderabad
4  // Author: Abhinav M
5  //
6  // Create Date: 01.10.2023 19:36:17
7  // Design Name:
8  // Module Name: BS_4_MUX_41
9  // Project Name:
10 // Target Devices:
```

```
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module BS_4_MUX41(output [3:0]y,input s0, s1,input [3:0]A0,A1,A2,A3);

 wire [3:0]y0, y1;
BR_4_MUX21 m1(y1,s1, A2, A3);
BR_4_MUX21 m2(y0, s1, A0,A1);
BR_4_MUX21 m3(y, s0, y0,y1);


endmodule
```

3.4.2 | Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 01.10.2023 22:23:57
// Design Name:
// Module Name: tb_BS_4_MUX41
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module tb_BS_4_MUX41();
reg s0,s1;
reg [3:0]A0,A1,A2,A3;
wire [3:0]data;
BS_4_MUX41 uut(data,s0,s1,A0,A1,A2,A3);
  initial
    begin

    A0=4'b0001;A1=4'b0010;A2=4'b0100;A3=4'b1000;
     s0=0;s1=0;
    #10 s0=0;s1=1;
    #10 s0=1;s1=0;
    #10 s0=1;s1=1;
    #10 $finish;
    end

endmodule
```

3.4.3 | RTL Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 01.10.2023 15:01:41
// Design Name:
// Module Name: BR_4_MUX_41
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
```

```
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module BR_4_MUX_41(output [3:0]data,input [1:0]s,input [3:0]A0,A1,A2,A3);

assign data = (s==2'b00)?A0:
        (s == 2'b01) ? A1 :
        (s == 2'b10) ? A2 :
        (s == 2'b11) ? A3 :
        2'b00;

endmodule
```

3.4.4 | Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 01.10.2023 19:12:50
// Design Name:
// Module Name: tb_BR_4_MUX_41
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_BR_4_MUX_41();
reg [1:0]s;
reg [3:0]A0,A1,A2,A3;
wire [3:0]data;
BR_4_MUX_41 uut(data,s,A0,A1,A2,A3);
  initial
    begin

    #10 A0=4'b0001;A1=4'b0010;A2=4'b0100;A3=4'b1000;
    #10 s=2'b00;
    #10 s=2'b01;
    #10 s=2'b10;
    #10 s=2'b11;




    #10 $finish;
    end


endmodule
```

3.5 | Elaborated design

3.5.1 | Structural

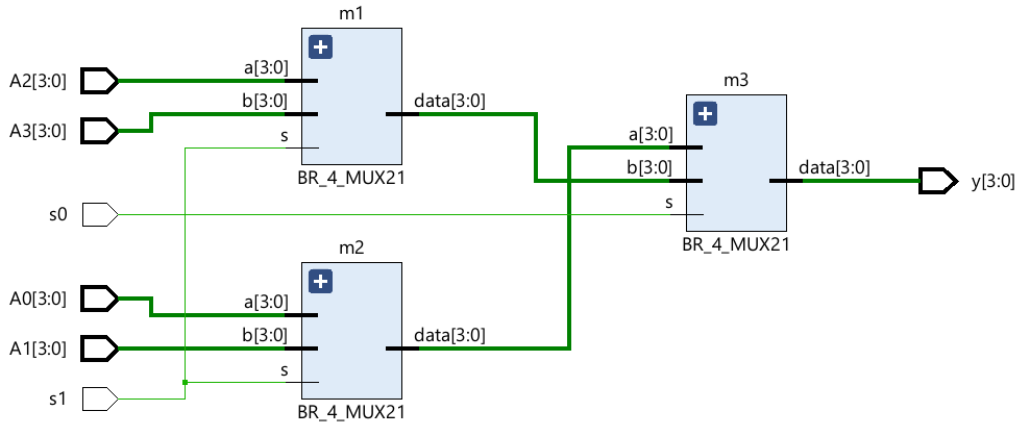

Figure 3.2: Elaborated design of the 4 bit 4x1 mux using three 4 bit 2x1 muxes.

3.6 | Elaborated design

3.6.1 | RTL

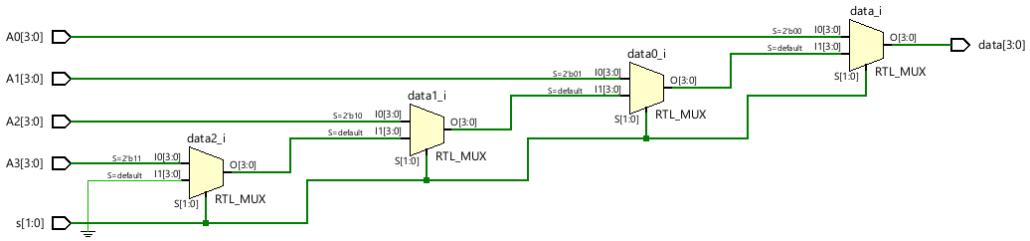

Figure 3.3: Elaborated design of the 4 bit 4x1 mux in RTL style.

3.7 | Timing diagram

3.7.1 | Structural

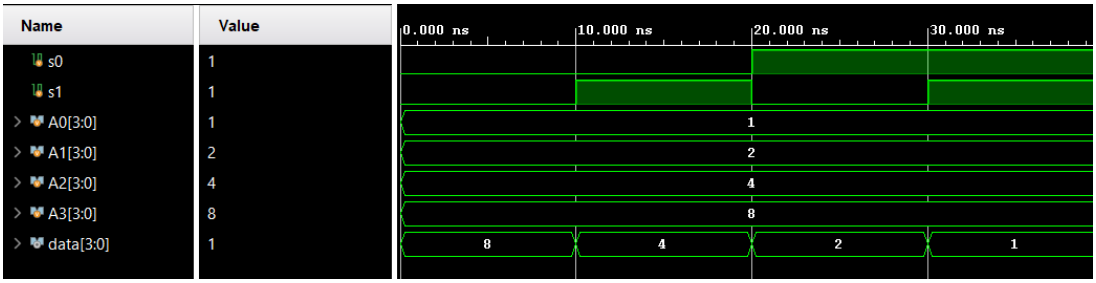

Figure 3.4: Timing diagram of the 4 bit 4x1 mux obtained after simulating the testbench.

3.7.2 | RTL

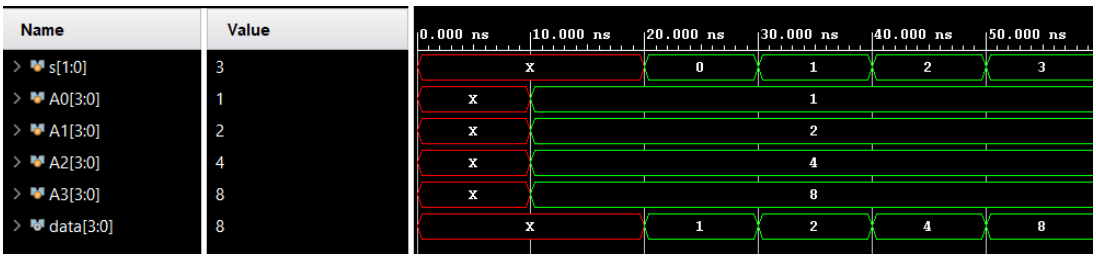

Figure 3.5: Timing diagram of the 4 bit 4x1 mux obtained after simulating the testbench.

3.8 | Utilization summary

3.8.1 | Structural

Resource	Utilization	Available	Utilization %
LUT	4	53200	0.01
IO	22	200	11.00

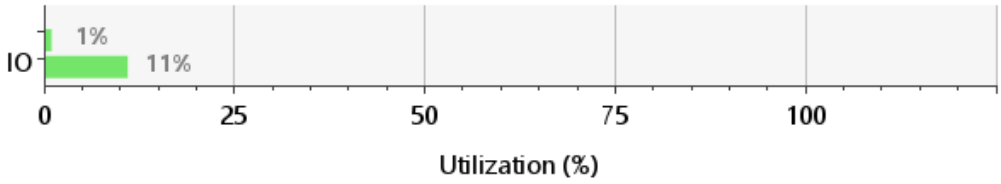

Figure 3.6: Resource utilization summary of the synthesized design.

3.8.2 | RTL

Resource	Utilization	Available	Utilization %
LUT	4	53200	0.01
IO	22	200	11.00

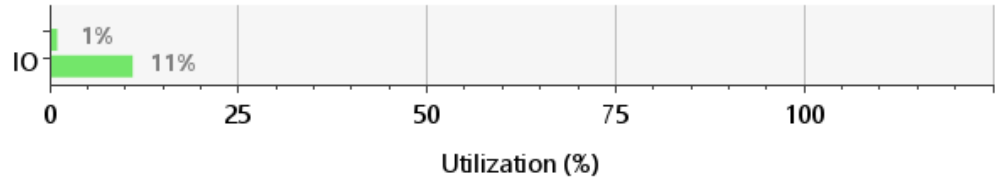

Figure 3.7: Resource utilization summary of the synthesized design.

3.9 | Remarks

The design and testing of 4 bit 4x1 mux in structural (using 4bit 2x1 mux) and RTL Verilog has been carried out. The device is showing expected behaviour.

4 | 4 bit 8:1 Multiplexer by instantiating 4:1, 2:1 MUXs

04-10-2023

4.1 | Objectives

- To design a 4 bit 8:1 Multiplexer by instantiating earlier designed multiplexers and verify its functionality using a testbench.

4.2 | Schematic

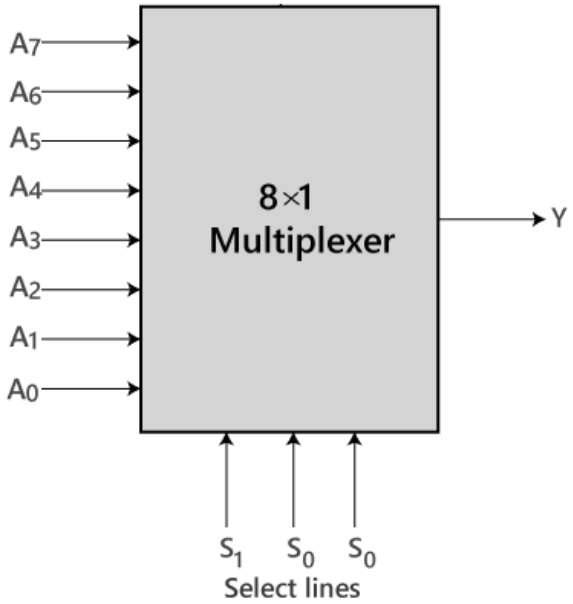

Figure 4.1: Block diagram of the 8x1 mux.

4.3 | Truth table

The truth table of the 8x1 mux:

Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	S_0	S_1	S_2	Y (Output)
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	0	0	0	A_0
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	0	0	1	A_1
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	0	1	0	A_2
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	0	1	1	A_3
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	1	0	0	A_4
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	1	0	1	A_5
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	1	1	0	A_6
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	1	1	1	A_7

Table 4.1: Truth table for the 8x1 mux.

4.4 | Code

4.4.1 | RTL Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 28.09.2023 07:33:43
// Design Name:
// Module Name: BS_4_MUX81
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module BS_4_MUX81(output [3:0]data, input [2:0]s, input [3:0]A0,A1,A2,A3,A4,A5,A6,A7);

wire [3:0]w1,w2;

BR_4_MUX_41 m1(w1,s[1:0],A0,A1,A2,A3);

BR_4_MUX_41 m2(w2,s[1:0],A4,A5,A6,A7);

BS_4_MUX21 m3(data,s[2],w1,w2);

endmodule
```

4.4.2 | Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 01.10.2023 19:00:53
// Design Name:
// Module Name: tb_BS_4_MUX81
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_BS_4_MUX81();
    reg [3:0]A0,A1,A2,A3,A4,A5,A6,A7;
    wire [3:0]data;
    reg [2:0] s;
BS_4_MUX81 uut(data,s,A0,A1,A2,A3,A4,A5,A6,A7);

     initial
    begin

    #10 A0=4'b0001;A1=4'b0010;A2=4'b0100;A3=4'b1000;A4=4'b1100;A5=4'b1110;A6=4'b0111;A7=4'b1111;
    #10 s=3'b000;
    #10 s=3'b001;
    #10 s=3'b010;
    #10 s=3'b011;
    #10 s=3'b100;
    #10 s=3'b101;
    #10 s=3'b110;
    #10 s=3'b111;
```

```
44      #10 $finish;
45      end
46
47
48
49  endmodule
```

4.5 | Elaborated design

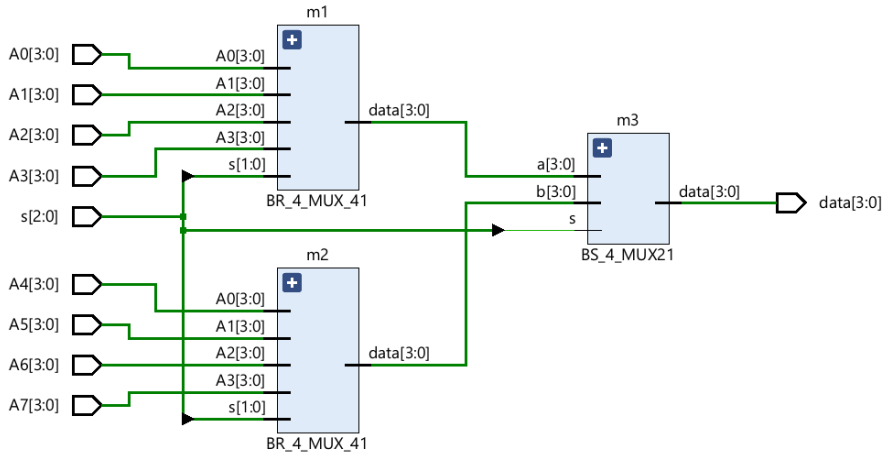

Figure 4.2: Elaborated design of the 4bit 8x1 mux using two 4bit 4x1 mux and one 4 bit 2x1 mux

.

4.6 | Timing diagram

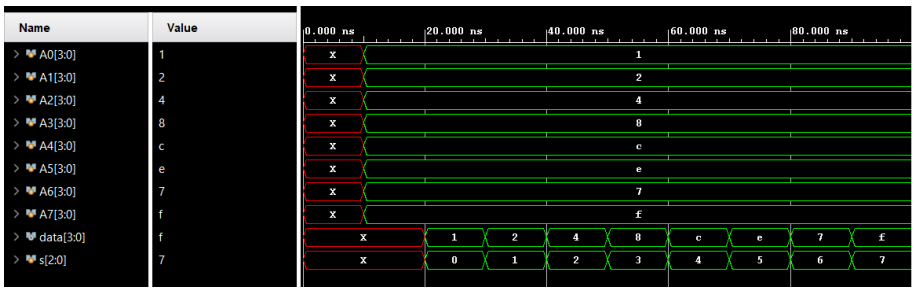

Figure 4.3: Timing diagram of 4bit 8x1 mux using 4x1 and 2x1 muxes.

4.7 | Utilization summary

Resource	Utilization	Available	Utilization %
LUT	8	53200	0.02
IO	39	200	19.50

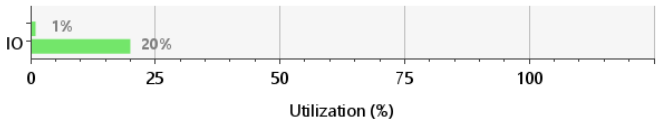

Figure 4.4: Utilization summary of the 4bit 8x1 mux using 4x1 and 2x1 muxes.

4.8 | Remarks

The design and testing of 4bit 8x1 mux using 4x1 and 2x1 muxes in structural Verilog has been carried out. The device is showing expected behaviour.

5 | 1 bit comparator

20-09-2023

5.1 | Objectives

- To design a 1 bit comparator in Structural and RTL Verilog and test its functionality.

5.2 | Schematic

Schematic for the 1 bit comparator is given below.

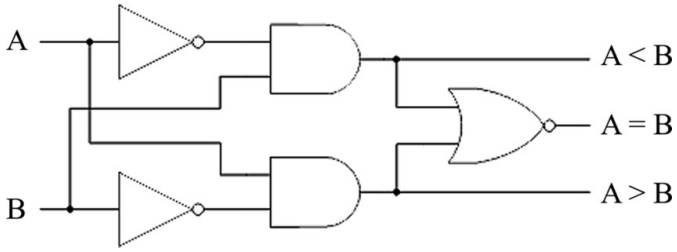

Figure 5.1: 1 bit comparator logic.

5.3 | Truth table

Truth table for 1 bit comparator.

A	B	LT	GT	EQ
0	0	0	0	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

Table 5.1: Truth table for a 1-bit comparator.

5.4 | Code

5.4.1 | Structural Design

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company: CASEST, University of Hyderabad
4  // Author: Abhinav M
5  //
6  // Create Date: 28.09.2023 07:34:41
7  // Design Name:
8  // Module Name: BS_1_comp1
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22 module BS_1_comp1(output LT, EQ, GT, input a,b);
23
24 wire w1, w2;
25
```

```
not X1(w1, a);
not X2 (w2, b);
and X3 (LT,w1, b);
and X4 (GT,w2, a);
xnor X5 (EQ, a, b);

endmodule
```

5.4.2 | Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 01.10.2023 17:00:37
// Design Name:
// Module Name: tb_BS_comp_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_BS_comp1();
reg a,b;
wire LT,EQ,GT;
BS_1_comp1 uut(LT, EQ, GT,a,b);
 initial
    begin
    #5
    a=1;b=0; #10
    a=0;b=0; #10
    a=0;b=1; #10
    a=1;b=1; #10



    #10 $finish;
    end

endmodule
```

5.4.3 | RTL Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 28.09.2023 07:34:55
// Design Name:
// Module Name: BR_1_comp1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////



module BR_1_comp1(output LT,EQ,GT, input a,b);

assign {LT, EQ, GT} = (a < b)?  {1'b1, 1'b0, 1'b0} :
                      (a==b) ?  {1'b0, 1'b1, 1'b0} :
                                 {1'b0, 1'b0, 1'b1};
```

```
29
30 endmodule
```

5.4.4 | Testbench

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company: CASEST, University of Hyderabad
4  // Author: Abhinav M
5  //
6  // Create Date: 01.10.2023 22:56:09
7  // Design Name:
8  // Module Name: tb_BR_1_comp1
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module tb_BR_1_comp1();
24 wire LT,EQ,GT;
25 reg a,b;
26 BR_1_comp1 uut(LT,EQ,GT,a,b);
27
28 initial
29     begin
30
31         a=1;b=0;
32     #10 a=1;b=1;
33     #10 a=0;b=1;
34
35
36
37     #10 $finish;
38     end
39
40
41 endmodule
```

5.5 | Elaborated design

5.5.1 | Structural

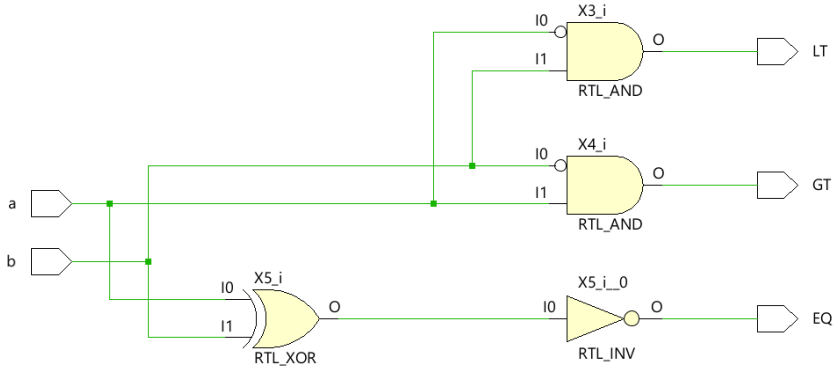

Figure 5.2: Elaborated design of 1 bit comparator in Structural Verilog.

5.5.2 | RTL

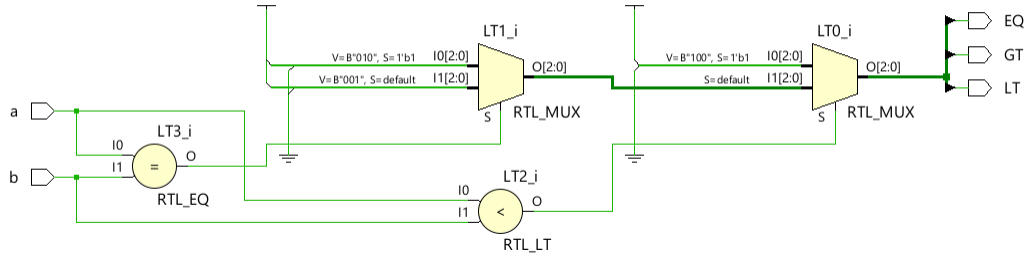

Figure 5.3: Elaborated design of 1 bit comparator in RTL Verilog.

5.6 | Timing diagram

5.6.1 | Structural

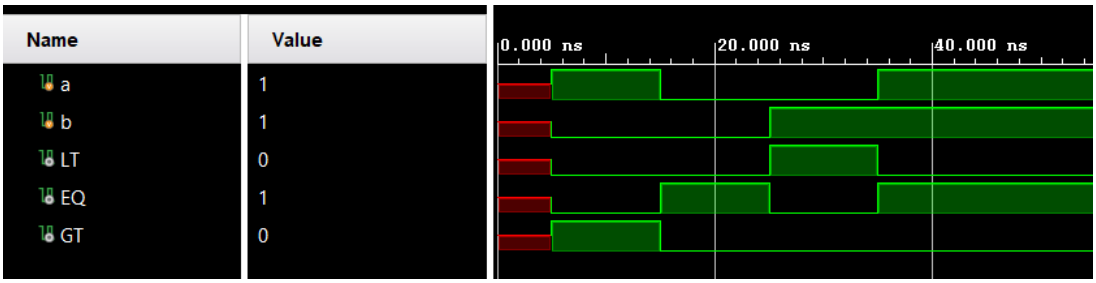

Figure 5.4: Timing diagram for 1 bit comparator in Structural Verilog.

5.6.2 | RTL

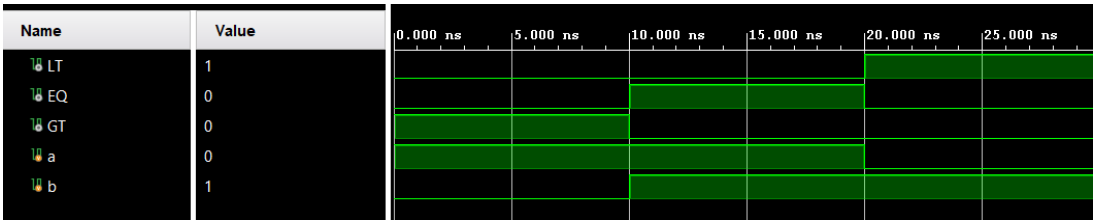

Figure 5.5: Timing diagram for 1 bit comparator in RTL Verilog.

5.7 | Utilization summary

Resource	Utilization	Available	Utilization %
LUT	2	53200	0.00
IO	5	200	2.50

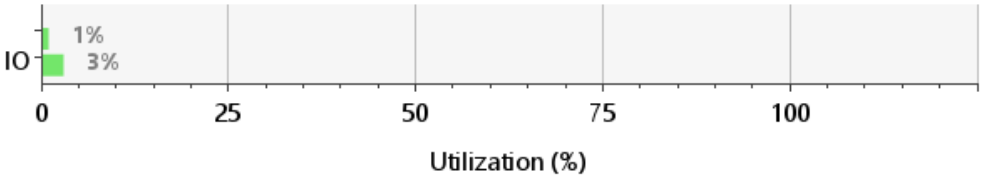

Figure 5.6: Utilization summary of 1 bit comparator in Structural Verilog.

Resource	Utilization	Available	Utilization %
LUT	2	53200	0.00
IO	5	200	2.50

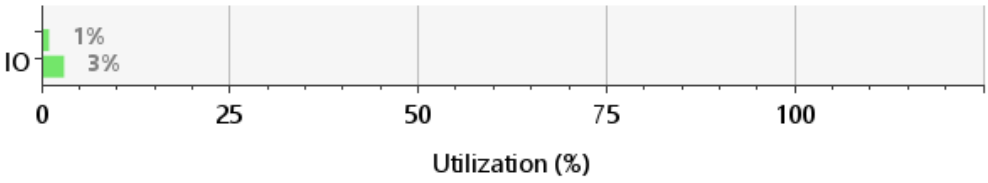

Figure 5.7: Utilization summary of 1 bit comparator in RTL Verilog.

5.8 | Remarks

The design and testing of 1 bit comparator in Structural and RTL Verilog has been carried out. The device is showing expected behaviour.

6 | 4 bit comparator using 1 bit comparator

20-09-2023

6.1 | Objectives

- To design a 4 bit comparator using 1 bit comparator in structural Verilog and test its functionality.

6.2 | Logic

In a 4-bit comparator, the condition of $A > B$ can be possible in the following four cases:

1. If $A3 = 1$ and $B3 = 0$
2. If $A3 = B3$ and $A2 = 1$ and $B2 = 0$
3. If $A3 = B3$, $A2 = B2$ and $A1 = 1$ and $B1 = 0$
4. If $A3 = B3$, $A2 = B2$, $A1 = B1$ and $A0 = 1$ and $B0 = 0$

Similarly, the condition for $A < B$ can be possible in the following four cases:

1. If $A3 = 0$ and $B3 = 1$
2. If $A3 = B3$ and $A2 = 0$ and $B2 = 1$
3. If $A3 = B3$, $A2 = B2$ and $A1 = 0$ and $B1 = 1$
4. If $A3 = B3$, $A2 = B2$, $A1 = B1$ and $A0 = 0$ and $B0 = 1$

The condition of $A = B$ is possible only when all the individual bits of one number exactly coincide with corresponding bits of another number.

6.3 | Code

6.3.1 | Structural Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 28.09.2023 07:35:56
// Design Name:
// Module Name: BS_4_comp4
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module BS_4_comp4(output LT, EQ, GT, input [3:0]a,b);

wire [3:0]LT1,EQ1,GT1;

BS_1_comp1 c0(LT1[0],EQ1[0],GT1[0],a[0],b[0]); //bit-wise comparison
BS_1_comp1 c1(LT1[1],EQ1[1],GT1[1],a[1],b[1]);
BS_1_comp1 c2(LT1[2],EQ1[2],GT1[2],a[2],b[2]);
BS_1_comp1 c3(LT1[3],EQ1[3],GT1[3],a[3],b[3]);

or o2(LT,LT1[3],(EQ1[3]&LT1[2]), (EQ1[3]&EQ1[2]&LT1[1]),(EQ1[3]&EQ1[2]&EQ1[1]&LT1[0])); //a<b

or o1(GT,GT1[3],(EQ1[3]&GT1[2]), (EQ1[3]&EQ1[2]&GT1[1]),(EQ1[3]&EQ1[2]&EQ1[1]&GT1[0])); //a>b


and a1(EQ,EQ1[0],EQ1[1],EQ1[2],EQ1[3]); //a==b

endmodule
```

6.3.2 | Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: CASEST, University of Hyderabad
// Author: Abhinav M
//
// Create Date: 01.10.2023 16:32:12
// Design Name:
// Module Name: tb_BS_comp4
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_BS_comp4();
reg [3:0]a,b;
wire LT,EQ,GT;
BS_4_comp4 uut(LT, EQ, GT,a,b);
 initial
    begin
    #5
    a=4'b1010;b=4'b0100;#10 //g

    a=4'b0010;b=4'b1100;#10  //l

    a=4'b1110;b=4'b1110;#10  //e

    // a=4'b1010;b=4'b0100;#10

   // a=4'b0010;b=4'b1100;#10

   // a=4'b1110;b=4'b1110;#10


    #10 $finish;
    end

endmodule
```

6.4 | Elaborated design

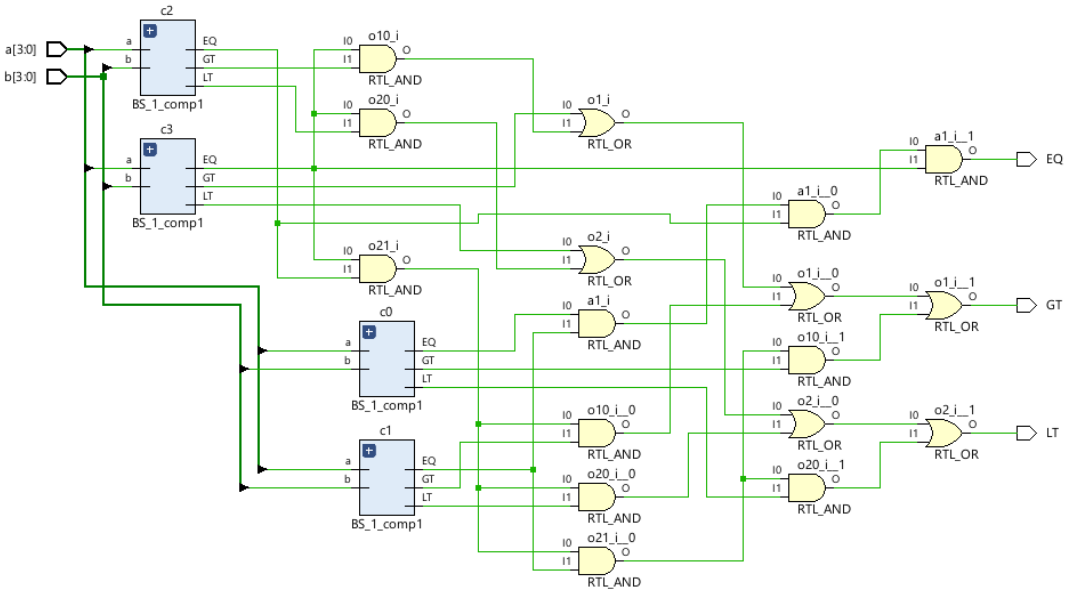

Figure 6.1: Elaborated design of 4 bit comparator using 1 bit comparator.

6.5 | Timing diagram

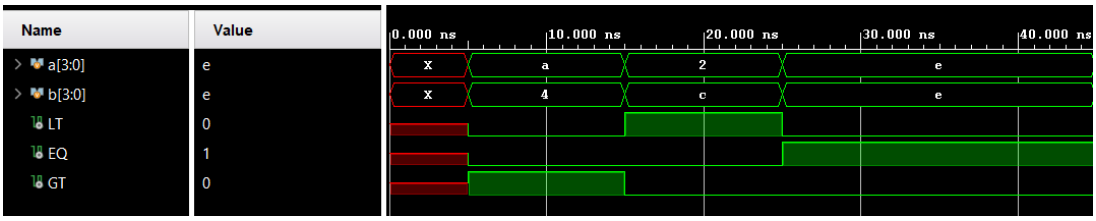

Figure 6.2: Timing diagram of 4 bit comparator using 1 bit comparator.

6.6 | Utilization summary

Resource	Utilization	Available	Utilization %
LUT	5	53200	0.01
IO	11	200	5.50

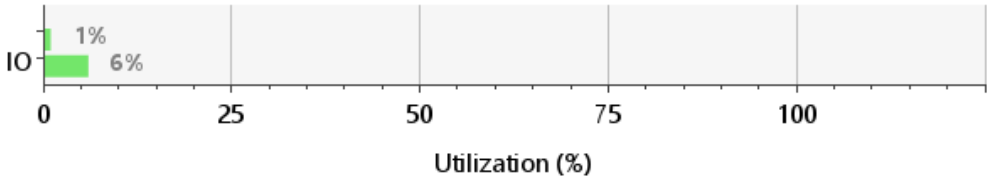

Figure 6.3: Utilization summary of 4 bit comparator using 1 bit comparator.

6.7 | Remarks

The design and testing of 4 bit comparator using 1 bit comparator in Verilog has been carried out. The device is showing expected behaviour.

7 | References

A | Appendix A: GitHub Repo

The GitHub repo referenced below contains the Verilog code and testbenches mentioned in this report.
https://github.com/AbhinavM2000/MV401D/