

BITS-Pilani Dubai Campus
I Sem 2020-21
Digital Design Laboratory / ECE/INSTR/CS F215
Submission Report
Experiment No.- 6 (Multiplexer and Demultiplexer)

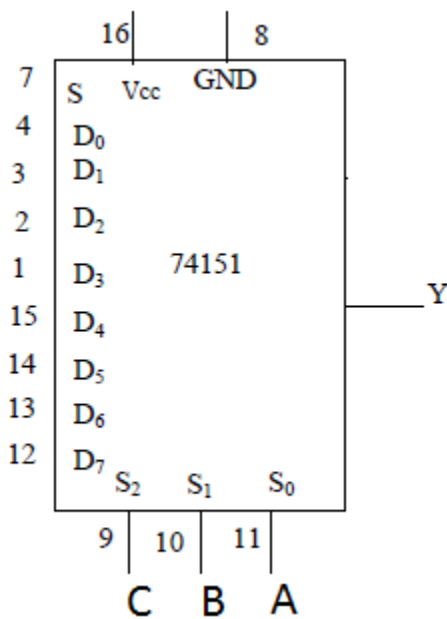
Name: Dhruv Jain

ID Number: 2019A7PS0260U

Hardware runs

Run 1: Multiplexer

Diagram



Truth Table

S	C	B	A	Y	Y'
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1

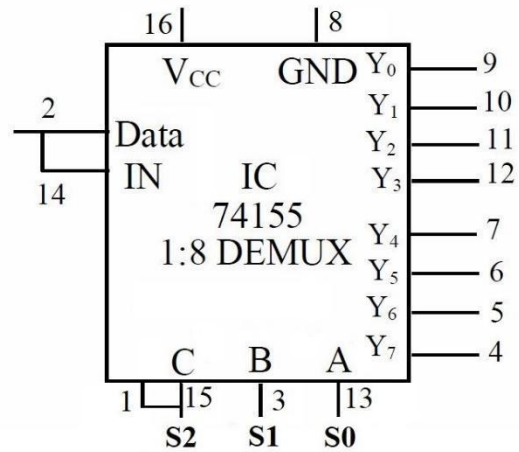
Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

Run 2: Demultiplexer

Diagram



Truth Table

Data	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0

Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

Software runs

Run 3: Multiplexer

1. Write the Verilog code and testbench for 2:1 Multiplexer using data flow modeling

A: Verilog Code and testbench-

//Verilog HDL for "LAB6", "run3" "functional"

```
module run3 (y,i,s0 );
    input[1:0] i;
    input s0;
    output y;
    wire p,q;

    assign p = i[0] & ~s0;
    assign q = i[1] & s0;
    assign y = p|q;
```

```
endmodule
```

```
module testbench();
    reg[1:0] i;
    reg s0;
    wire y;
```

```
initial begin
    #000 i=2'b00; s0=1;
    #100 i=2'b00; s0=0;
    #100 i=2'b01; s0=0;
    #100 i=2'b10; s0=0;
    #100 i=2'b11; s0=0;
    #100 i=2'b00; s0=1;
    #100 i=2'b01; s0=1;
    #100 i=2'b10; s0=1;
    #100 i=2'b11; s0=1;
    # 100;
    $stop;
end
run3 g1 (y,i,s0);
endmodule
```

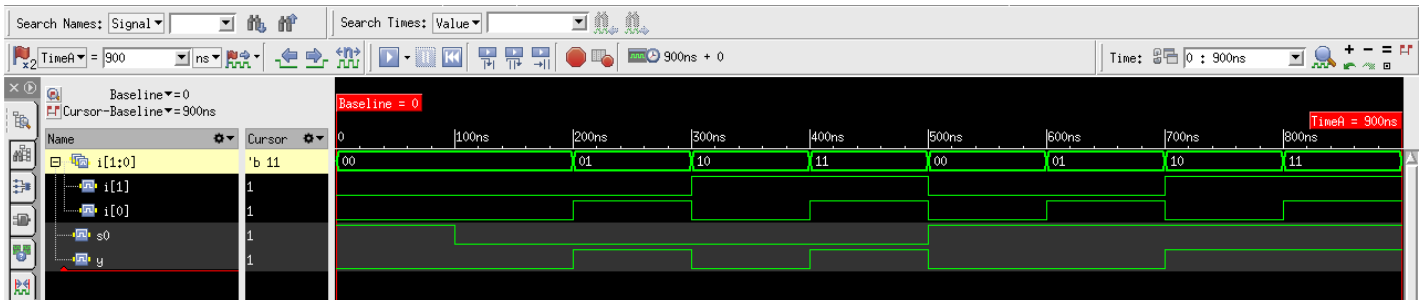
Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

Q: Paste the screenshot of waveform window where you get the waveforms for the above code. (online students also paste the URL)

A:



Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

2. Write the Verilog code and testbench for 2:1 Multiplexer using behavioral modeling.

A: Verilog Code and testbench -

```
//Verilog HDL for "LAB6", "run32" "functional"
```

```
module run32 (input i0,i1,i2,i3, s0,s1, output reg y);
always @ (*)
begin

case ({s0,s1})
2'b00 : y = i0;
2'b01 : y = i1;
2'b10 : y = i2;
2'b11 : y = i3;
endcase
end
endmodule
```

```
module testbench();
reg i0,i1,i2,i3,s0,s1;
wire y; // check this because output is reg
```

```
initial begin
```

```
#000 i0=0; i1=0; i2=0; i3=0; s0=0; s1=0;
#100 i0=0; i1=1; i2=0; i3=1; s0=0; s1=0;
#100 i0=0; i1=1; i2=0; i3=1; s0=0; s1=1;
#100 i0=0; i1=1; i2=0; i3=1; s0=1; s1=0;
#100 i0=0; i1=1; i2=0; i3=1; s0=1; s1=1;
#100;
end
run32 g1 ( i0,i1,i2,i3, s0,s1, y);
endmodule
```

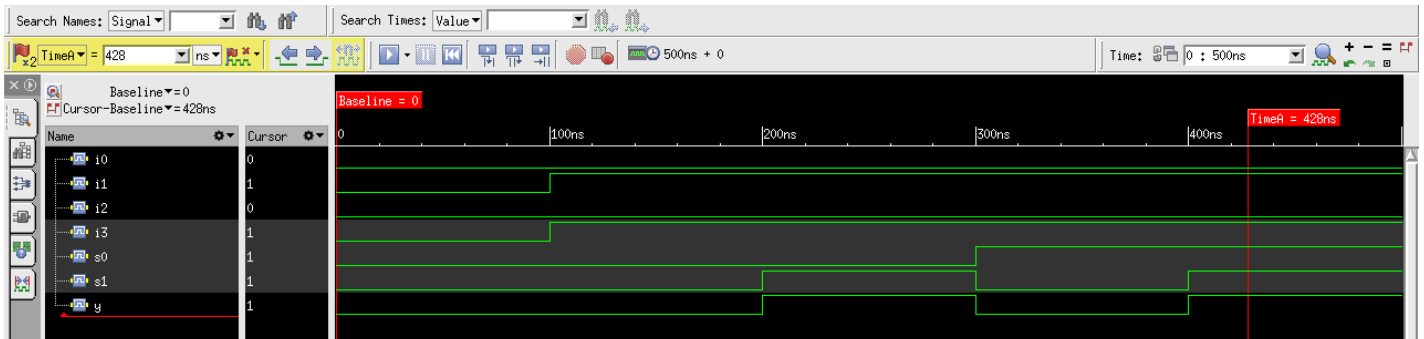
Q: Paste the screenshot of waveform window where you get the waveforms for the above code. (online students also paste the URL)

A:

Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U



Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

3. Write the Verilog code and testbench for 4:1 Multiplexer using structural modeling.

A: Verilog Code and testbench -

//Verilog HDL for "LAB6", "run33" "functional"

```
module run33 (d,q,s);
    output q;
    input[3:0] d;
    input[1:0] s;
    wire one,two;

    t2x1mux g1 (one,d[0], d[1], s[0]);
    t2x1mux g2 (two,d[2], d[3], s[0]);
    t2x1mux g3 (q,one, two, s[1]);
```

endmodule

```
module t2x1mux (y,i0,i1,s0 );
    input i0,i1;
    input s0;
    output y;
    wire p,q;

    assign p = i0 & ~s0;
    assign q = i1 & s0;
    assign y = p|q;
```

endmodule

```
module testbench();
    wire q;
    reg[3:0] d;
    reg[1:0] s;
```

```
    initial begin
        #000 d=4'b0000; s=2'b00;
        #100 d=4'b0101; s=2'b01;
        #100 d=4'b0101; s=2'b10;
        #100 d=4'b0101; s=2'b11;
        #100;
        $stop;
    end
    run33 g1 (d,q,s);
endmodule
```

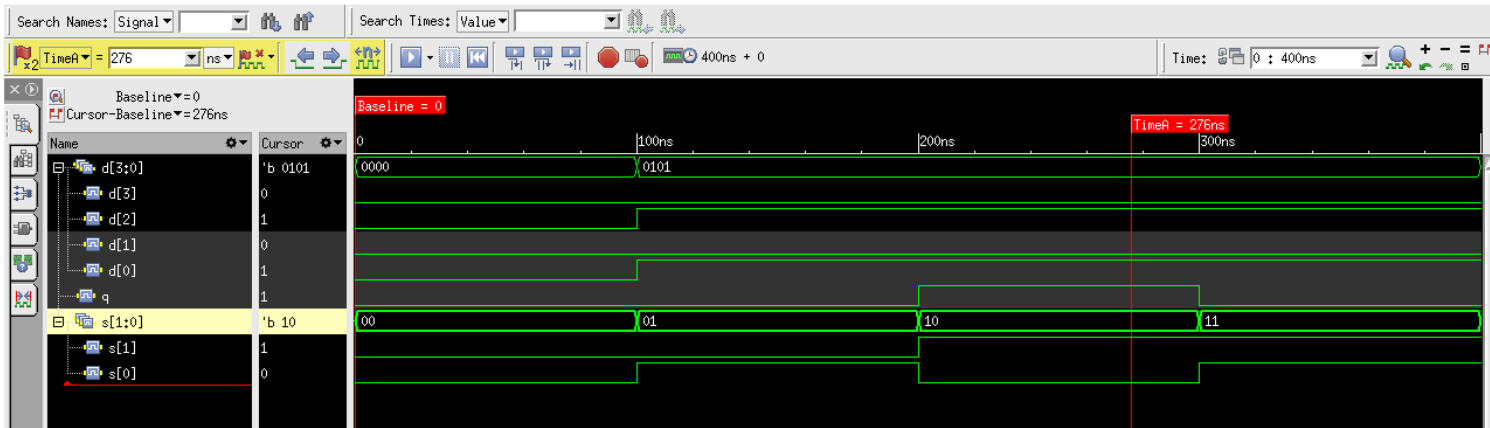
Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

Q: Paste the screenshot of waveform window where you get the waveforms for the above code. (online students also paste the URL)

A:



Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

Run 4: Demultiplexer

1. Write the Verilog code and testbench for 1:4 demultiplexer using behavioral modeling.

A: Verilog Code and testbench -

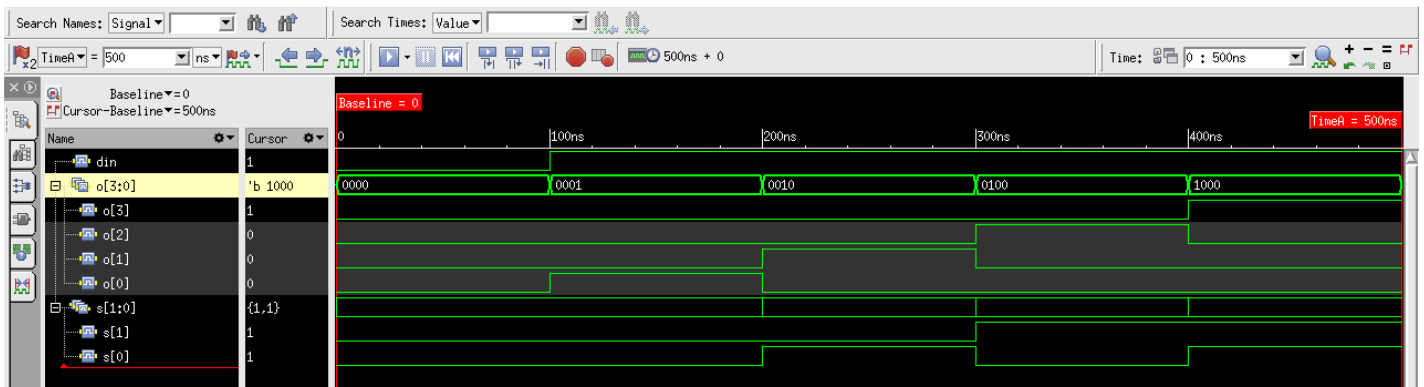
//Verilog HDL for "LAB6", "run41" "functional"

```
module run41 (input din, s0,s1, output o0, o1, o2, o3);  
assign o0 = din & (~s1 & ~s0);  
assign o1 = din & (~s1 & s0);  
assign o2 = din & (s1 & ~s0);  
assign o3 = din & (s1 & s0);  
endmodule
```

```
module testbench();  
reg din;  
reg s[1:0];  
wire[3:0] o;  
  
initial begin  
    #000 din = 0; s[1] = 0; s[0] = 0;  
    #100 din = 1; s[1] = 0; s[0] = 0;  
    #100 din = 1; s[1] = 0; s[0] = 1;  
    #100 din = 1; s[1] = 1; s[0] = 0;  
    #100 din = 1; s[1] = 1; s[0] = 1;  
    #100;  
    $stop;  
end  
run41 g1 (din,s[0],s[1],o[0],o[1],o[2],o[3]);  
endmodule
```

Q: Paste the screenshot of waveform window where you get the waveforms for the above code. (online students also paste the URL)

A:



Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

2. Write the Verilog code and testbench for 1:8 demux using data flow modeling.

A: Verilog Code and testbench-

//Verilog HDL for "LAB6", "run42" "functional"

```
module run42 ( input din, input [2:0] s, output [7:0] q);  
// assign q = din<<s;  
assign q[0] = din & ~s[2] & (~s[1] & ~s[0]);  
assign q[1] = din & ~s[2] & (~s[1] & s[0]);  
assign q[2] = din & ~s[2] & (s[1] & ~s[0]);  
assign q[3] = din & ~s[2] & (s[1] & s[0]);  
assign q[4] = din & s[2] & (~s[1] & ~s[0]);  
assign q[5] = din & s[2] & (~s[1] & s[0]);  
assign q[6] = din & s[2] & (s[1] & ~s[0]);  
assign q[7] = din & s[2] & (s[1] & s[0]);
```

endmodule

```
module testbench();  
reg din;  
reg [2:0] s;  
wire[7:0] q;
```

```
initial begin  
#000 din = 0; s[2] = 0; s[1] = 0; s[0] = 0;  
#100 din = 1; s[2] = 0; s[1] = 0; s[0] = 1;  
#100 din = 1; s[2] = 0; s[1] = 1; s[0] = 0;  
#100 din = 1; s[2] = 0; s[1] = 1; s[0] = 1;  
#100 din = 1; s[2] = 1; s[1] = 0; s[0] = 0;  
#100 din = 1; s[2] = 1; s[1] = 0; s[0] = 1;  
#100 din = 1; s[2] = 1; s[1] = 1; s[0] = 0;  
#100 din = 1; s[2] = 1; s[1] = 1; s[0] = 1;  
#100;  
$stop;  
end  
run42 g1 (din,s,q);  
endmodule
```

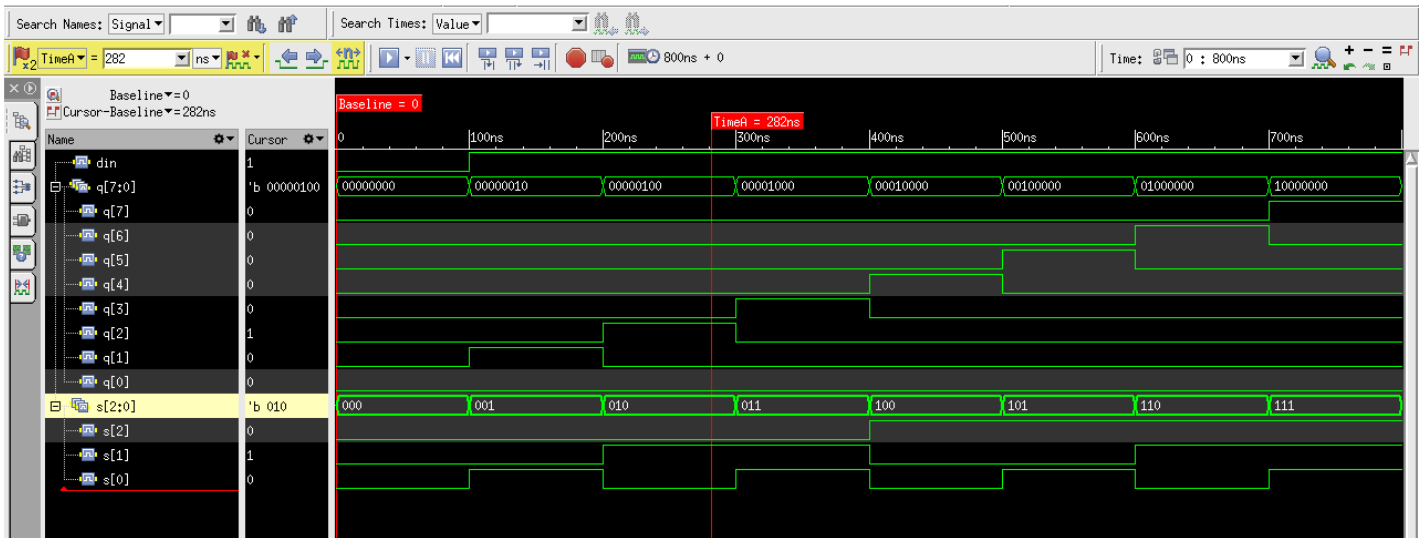
Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

Q: Paste the screenshot of waveform window where you get the waveforms for the above code. (online students also paste the URL)

A:



Experiment No.- 6 (Multiplexer and Demultiplexer)

Name : Dhruv Jain

ID Number : 2019A7PS0260U

Assignment All assignments are to be submitted strictly before start of next lab session through online only. Late assignments will not be entertained and will be awarded '0' marks.

1. Write the Verilog code and testbench for 4:1 Multiplexer using data flow modeling.

Ans: Link1: <https://www.edaplayground.com/x/CLP4>

2. Satisfy yourself that below code is 1:8 demux. (hint: write the testbench for the same and check)

Ans: Link2: <https://www.edaplayground.com/x/PK7i>

3. Verilog code and testbench for implementing 2-input XOR gate using 2x1 multiplexer in structural modeling.

Ans: Link3: <https://www.edaplayground.com/x/pcA6>

Self-Practice and self-evaluation

1. Verilog code and testbench for implementing 2-input NAND gate using 2x1 multiplexer in structural modeling.
2. Verilog code and testbench for a 4-input 4-output 1-channel communication system using 4x1 multiplexer and 1x4 demultiplexer, control the select lines of multiplexer and demultiplex to switch the channel data between 4-input lines.
3. Verilog code and testbench for 8x1 multiplexer using 4x1 multiplexer and 2x1 multiplexer in structural modeling.