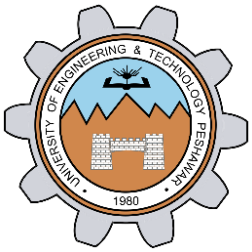
DIGITAL SYSTEM DESIGN LAB

LAB 1



Spring 2021

CSE308L DSD LAB

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

Engr. Madiha Sher

Tuesday, April 20th, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task # 01:

Develop a program to control on Board LED using On board available switch.

Problem Analysis:

Truth Table:

Input	Output
0	0
1	1

Code:

Module:

```
module buff(O1,O2,O3,O4,O5,O6,O7,O8,I1,I2,I3,I4,I5,I6,I7,I8);
        input I1,I2,I3,I4,I5,I6,I7,I8;
        output O1,O2,O3,O4,O5,O6,O7,O8;

        buf (O1,I1);
        buf (O2,I2);
        buf (O3,I3);
        buf (O4,I4);
        buf (O5,I5);
        buf (O6,I6);
        buf (O7,I7);
        buf (O8,I8);
endmodule
```

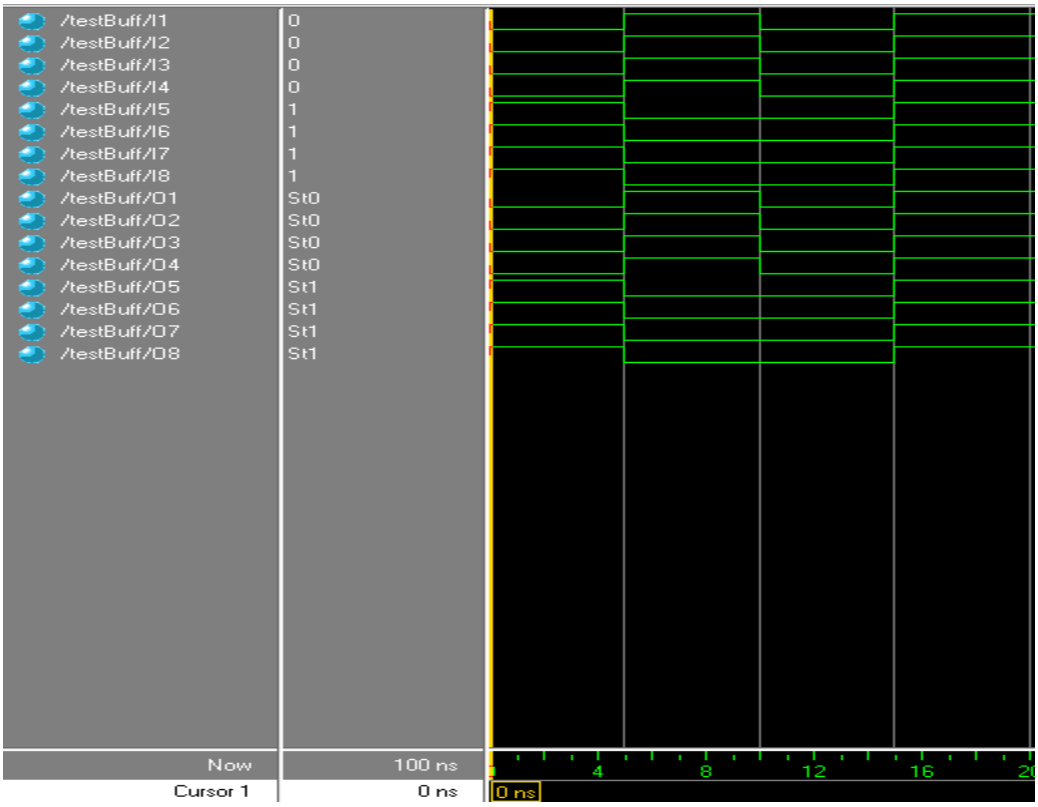
Test Bench:

```
module testBuff;
	reg I1,I2,I3,I4,I5,I6,I7,I8;
	wire O1,O2,O3,O4,O5,O6,O7,O8;

	buff b(O1,O2,O3,O4,O5,O6,O7,O8,I1,I2,I3,I4,I5,I6,I7,I8);

	initial
	begin
		$monitor ("%b %b %b %b %b %b %b %b %b %b %b %b %b %b %b %b %b %b %b %b",I1,I2,I3,I4,I5,I6,I7,I8,O1,O2,O3,O4,O5,O6,O7,O8);
		I1=0;I2=0;I3=0;I4=0;I5=1;I6=1;I7=1;I8=1;
		#5
		I1=1;I2=1;I3=1;I4=1;I5=0;I6=0;I7=0;I8=0;
		#5
		I1=0;I2=0;I3=0;I4=0;I5=0;I6=0;I7=0;I8=0;
		#5
		I1=1;I2=1;I3=1;I4=1;I5=1;I6=1;I7=1;I8=1;
	end
endmodule
```

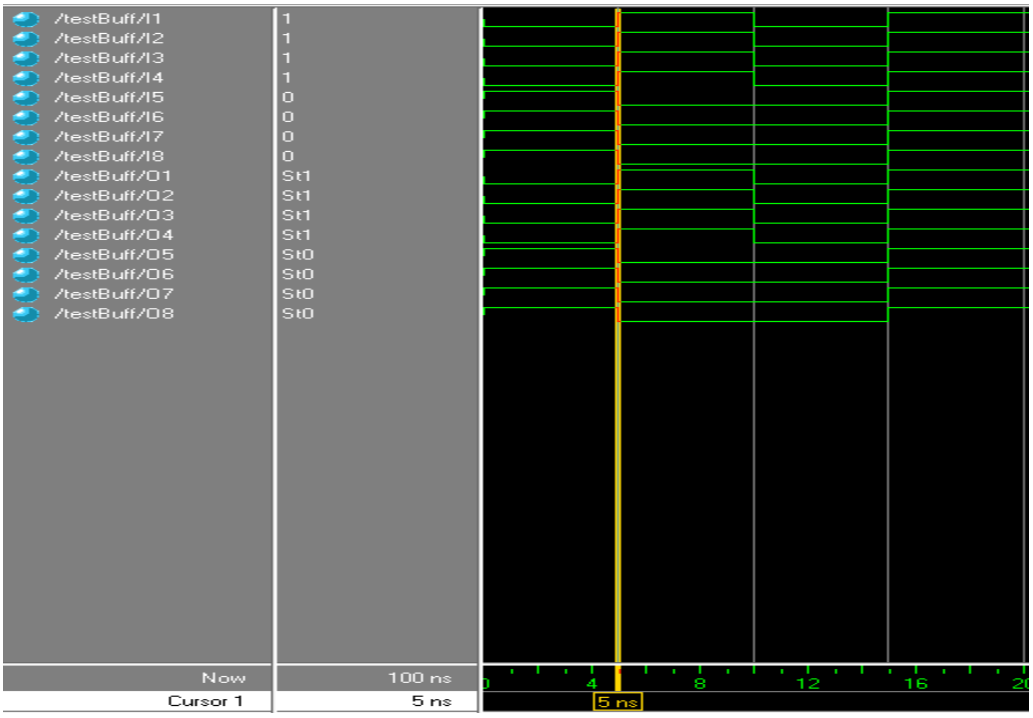
Output:

Waveform:

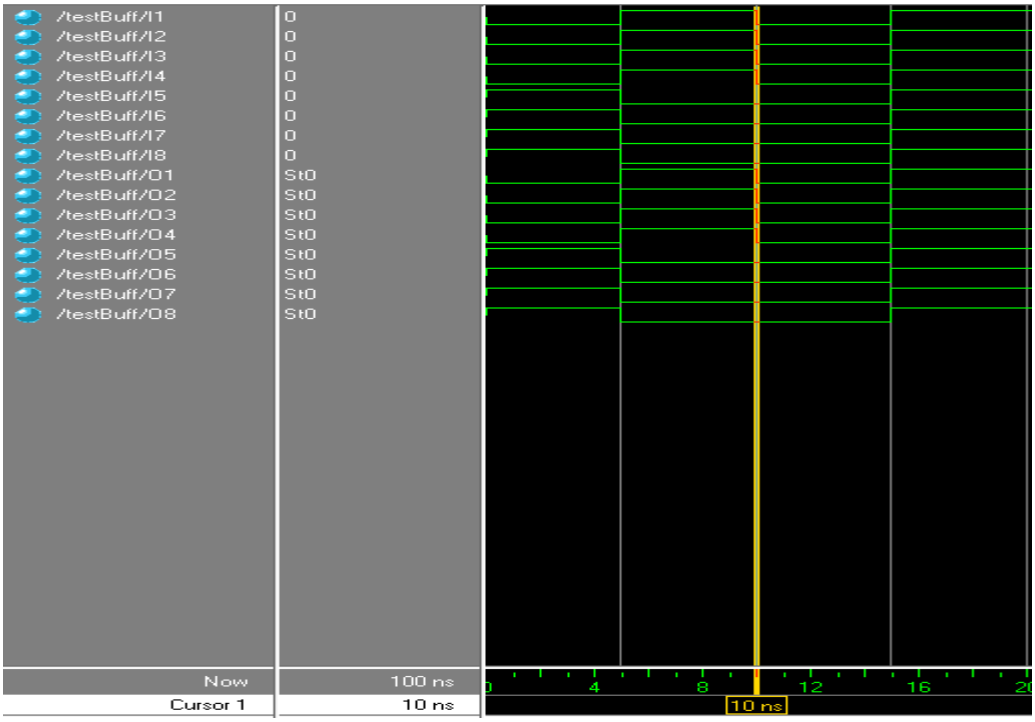
At 0ns:



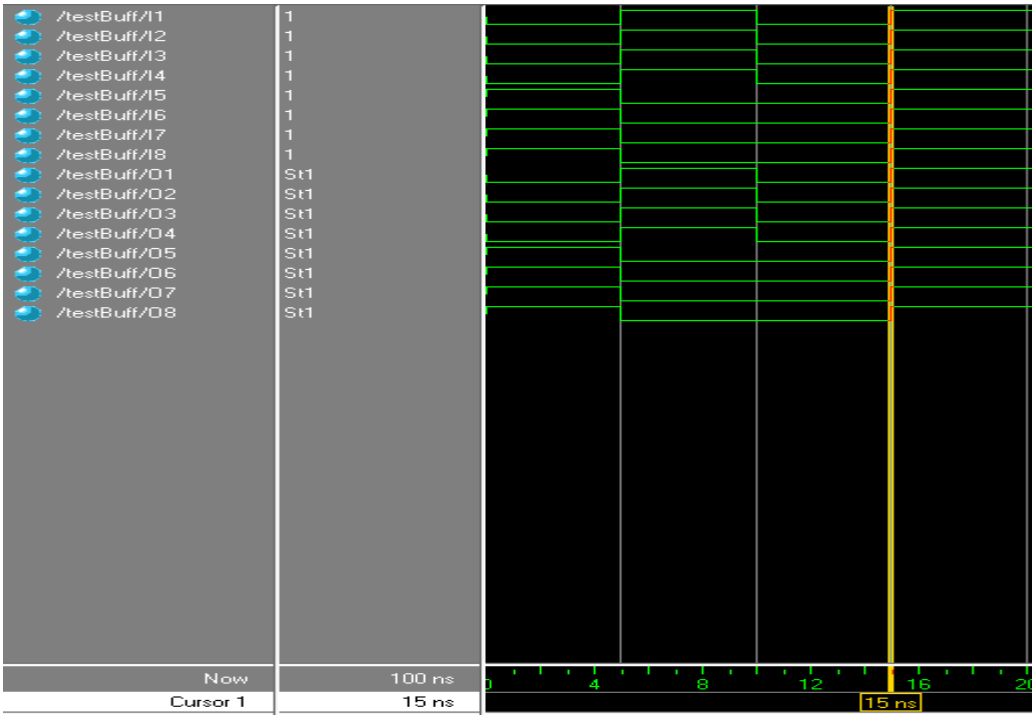
At 5ns:



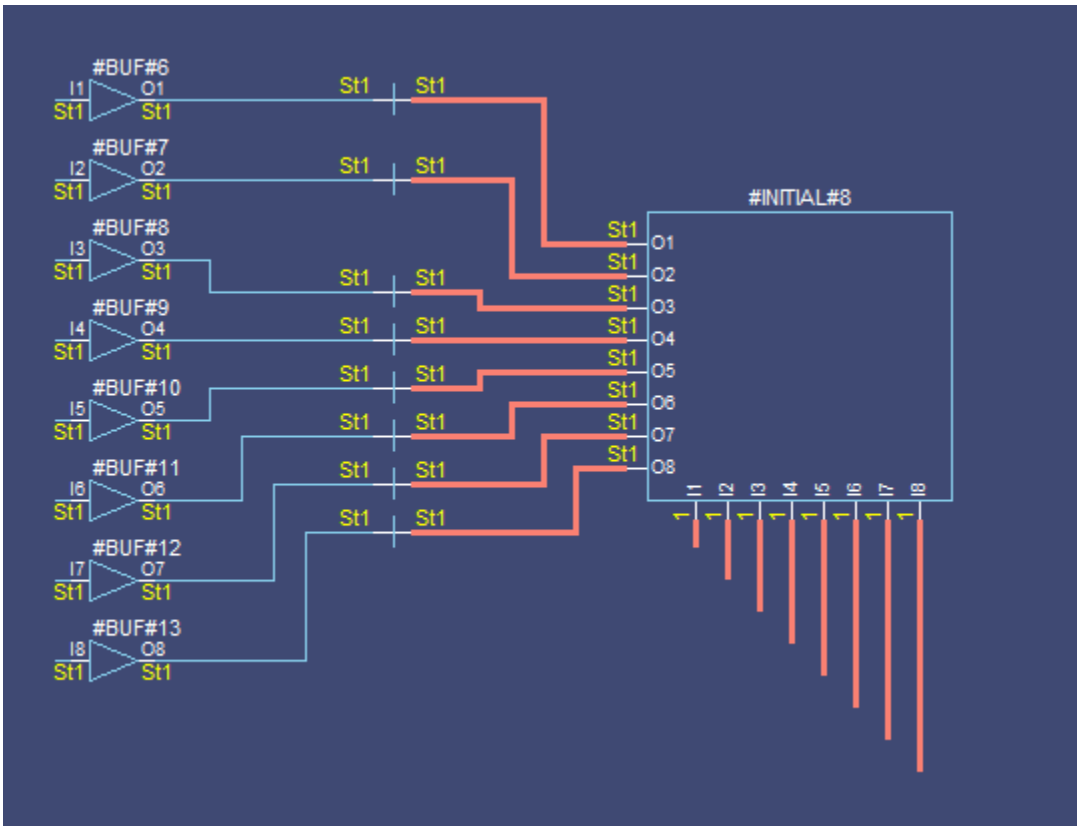
At 10ns:



At 15ns:



Dataflow:


#BUF#6
I1
O1
St1
St1
St1
St1
#BUF#7
I2
O2
St1
St1
St1
St1
#BUF#8
I3
O3
St1
St1
St1
St1
#BUF#9
I4
O4
St1
St1
St1
St1
#BUF#10
I5
O5
St1
St1
St1
St1
#BUF#11
I6
O6
St1
St1
St1
St1
#BUF#12
I7
O7
St1
St1
St1
St1
#BUF#13
I8
O8
St1
St1
St1
St1
#INITIAL#8
St1
O1
St1
O2
St1
O3
St1
O4
St1
O5
St1
O6
St1
O7
St1
O8
I1
I2
I3
I4
I5
I6
I7
I8
1
1
1
1
1
1
1
1

Task # 02:

Develop a program that implements a 2x1 multiplexer using gate level modeling.

Problem Analysis:

Truth Table:

S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Simplified Expression for Z:

$Z = (A.\overline{S}) + (B.S)$

Code:

Module:

```
module Mux2(Z,A,B,S);
        output Z;
        input A,B,S;

        wire So,X,Y;

        not (So,S);
        and (X,A,So);
        and (Y,B,S);
        or (Z,X,Y);
endmodule
```

Test Bench:

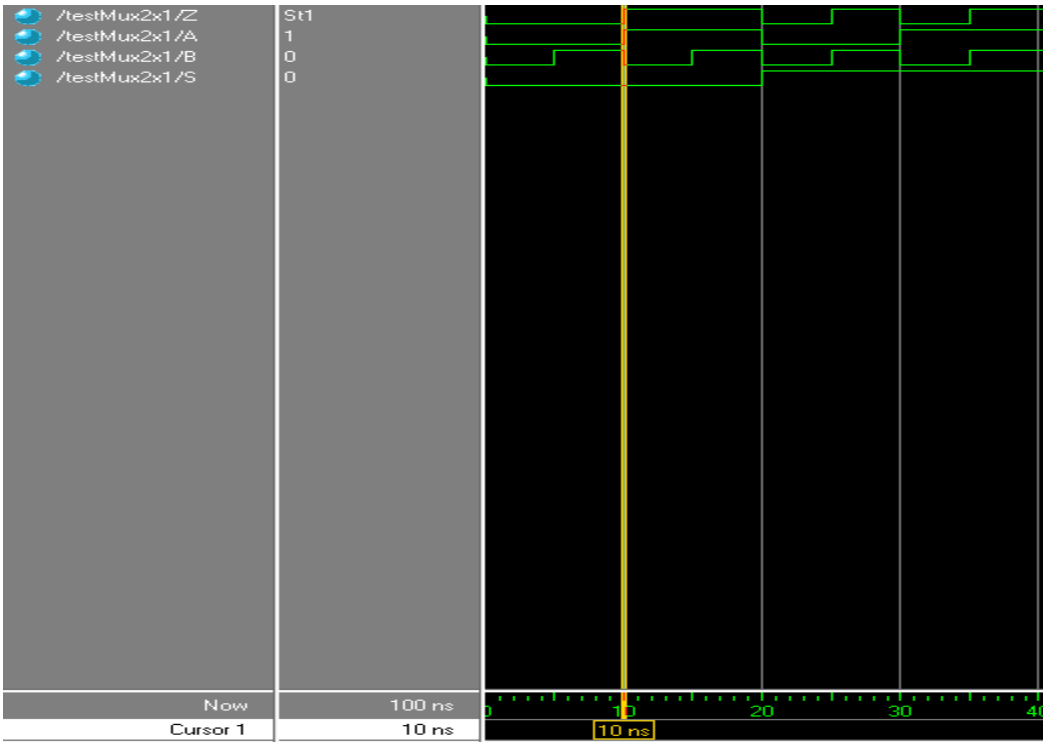
```
module testMux2x1;
        wire Z;
        reg A,B,S;
        Mux2 m(Z,A,B,S);

        initial
        begin
                $monitor ("%b %b %b %b",S,A,B,Z);
                A=0;B=0;S=0;
                #5
                A=0;B=1;S=0;
                #5
                A=1;B=0;S=0;
                #5
                A=1;B=1;S=0;
                #5

                A=0;B=0;S=1;
                #5
                A=0;B=1;S=1;
                #5
                A=1;B=0;S=1;
                #5
                A=1;B=1;S=1;
        end
endmodule
```

Output:

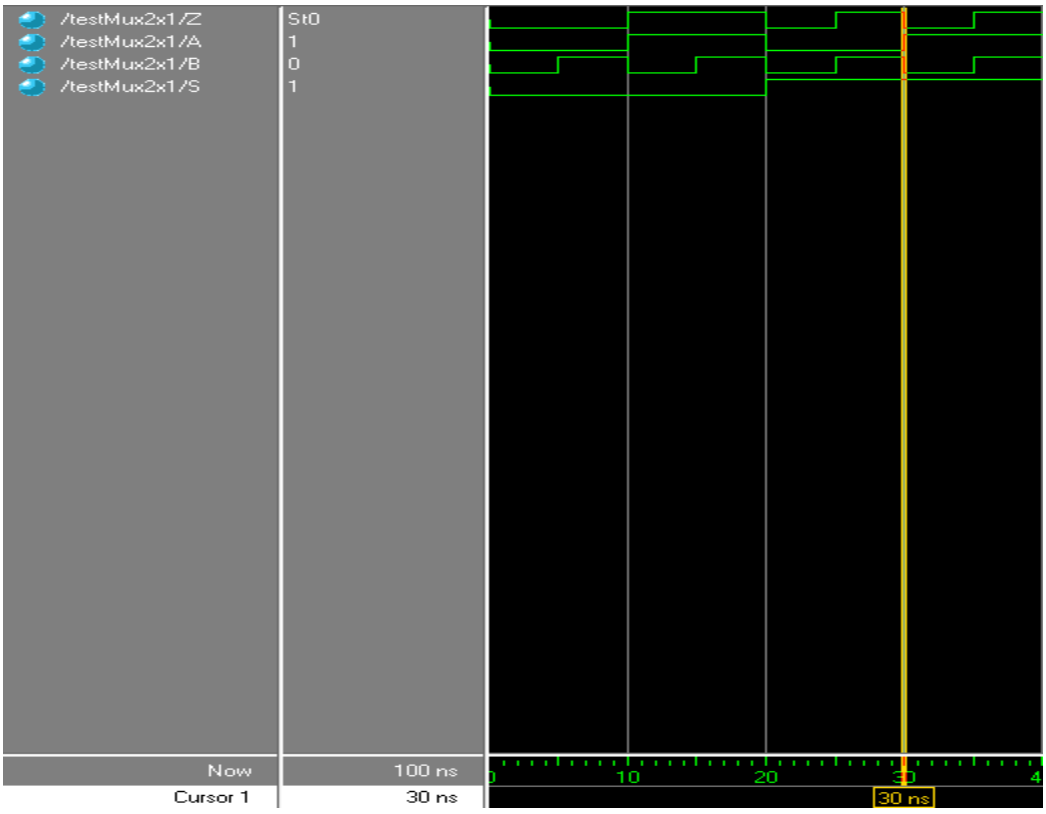
```
run
#0000
#0010
#0101
#0111
#1000
#1011
#1100
#1111
```

Waveform:

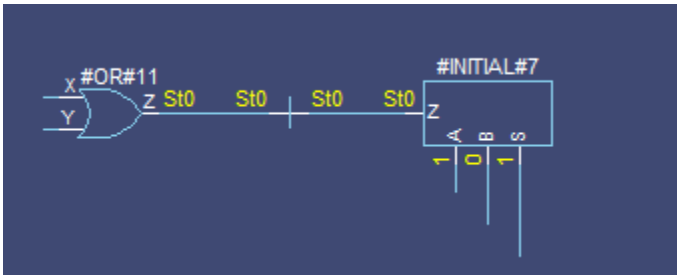
At 10ns:



At 30ns:



Dataflow:



Task # 03:

Develop a program that implements a 4x1 multiplexer using gate level modeling.

Problem Analysis:

Truth Table:

S0	S1	A	B	C	D	Z
0	0	A	x	x	x	A
0	1	x	B	x	x	B
1	0	x	x	C	x	C
1	1	x	x	x	D	D

Simplified Expression for Z:

Z = (A.S0n.S1n) + (B.S0n.S1) + (C.S0.S1n) + (D.S0.S1)

Code:

Module:

```
module Mux4x1(Z,A,B,C,D,S0,S1);
        input A,B,C,D,S0,S1;
        output Z;
        wire S0n,S1n,V,W,X,Y;

        not (S0n,S0);
        not (S1n,S1);

        and (V,A,S0n,S1n);
        and (W,B,S0n,S1);
        and (X,C,S0,S1n);
        and (Y,D,S0,S1);

        or (Z,V,W,X,Y);
endmodule
```

Test Bench:

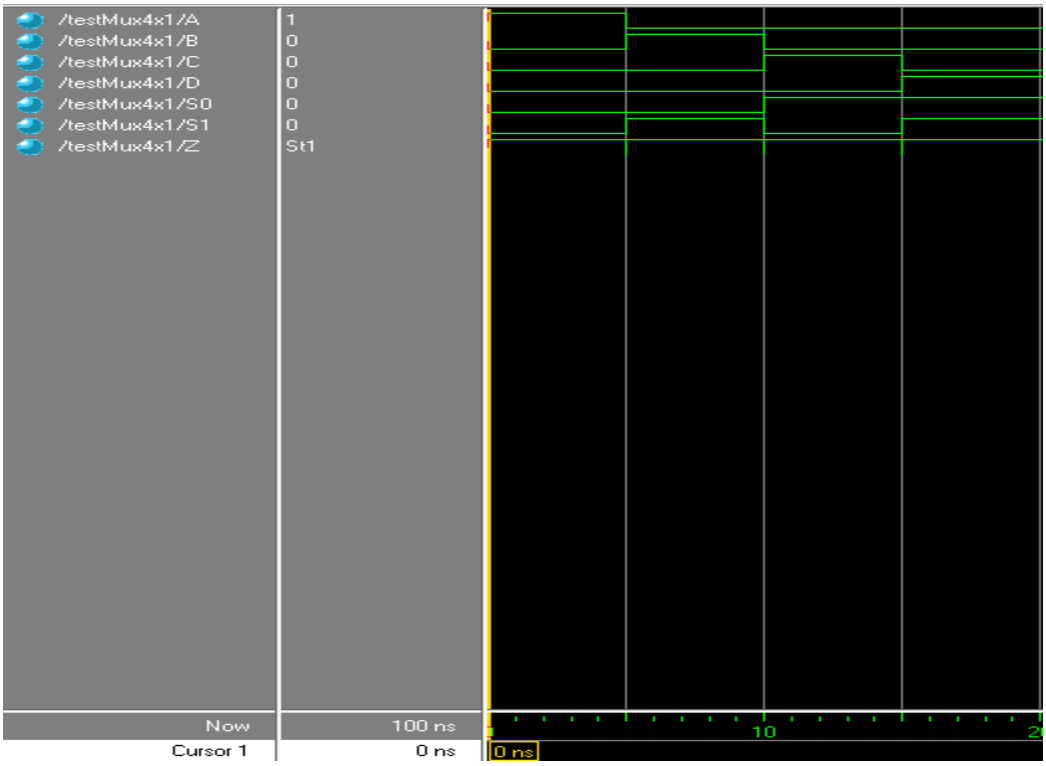
```
module testMux4x1;

        reg A,B,C,D,S0,S1;
        wire Z;

        Mux4x1 m4(Z,A,B,C,D,S0,S1);

        initial
        begin
                $monitor ("%b %b %b %b %b %b %b",S0,S1,A,B,C,D,Z);

                S0=0;S1=0;A=1;B=0;C=0;D=0;
                #5

                S0=0;S1=1;A=0;B=1;C=0;D=0;
                #5

                S0=1;S1=0;A=0;B=0;C=1;D=0;
                #5

                S0=1;S1=1;A=0;B=0;C=0;D=1;
        end

endmodule
```

Output:

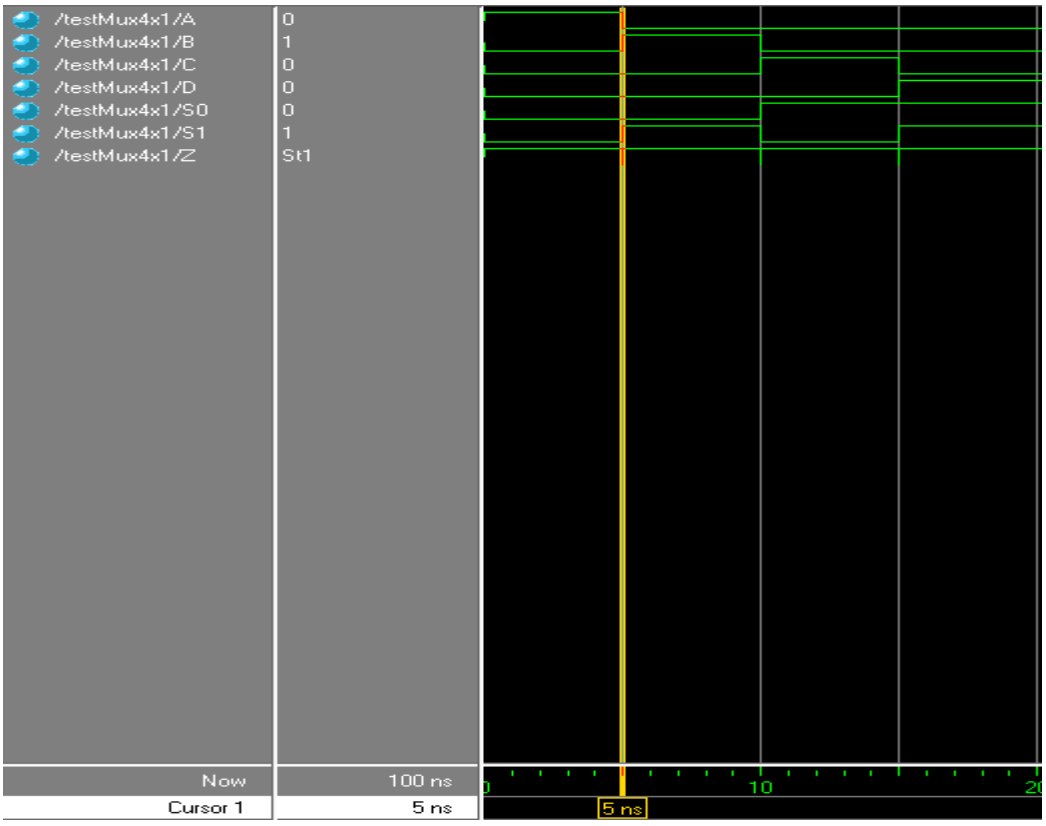
```
run
#0010001
#0101001
#1000101
#1100011
```

Waveform:

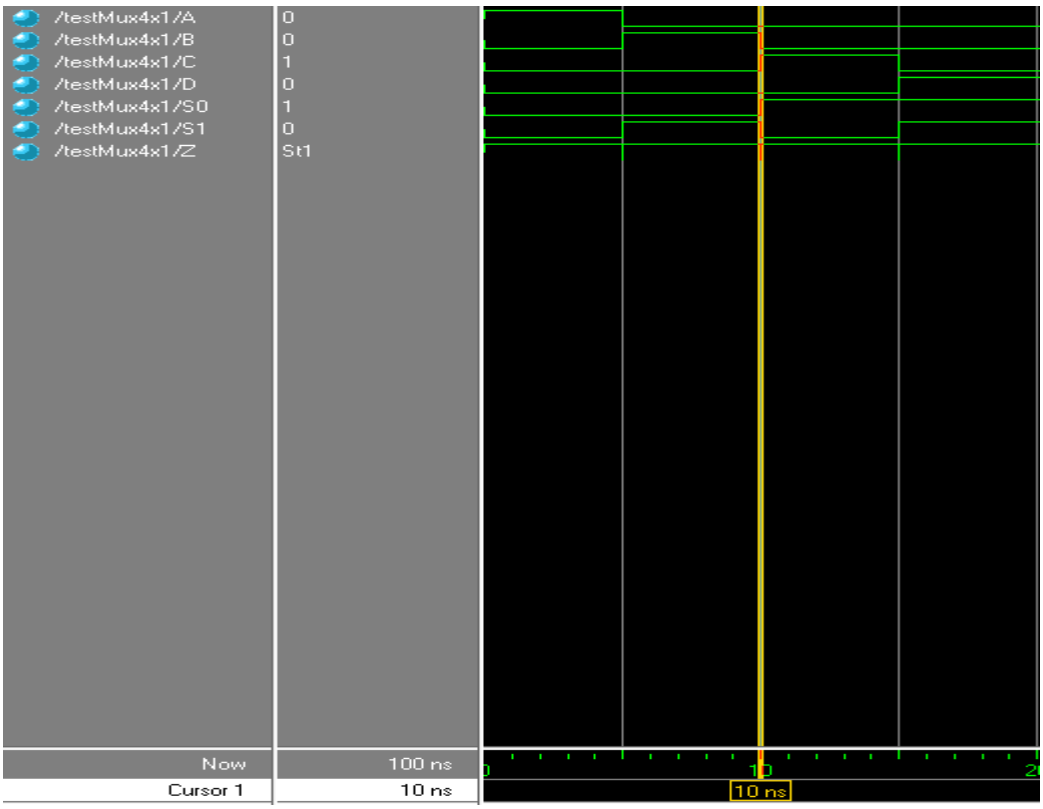
At 0ns:



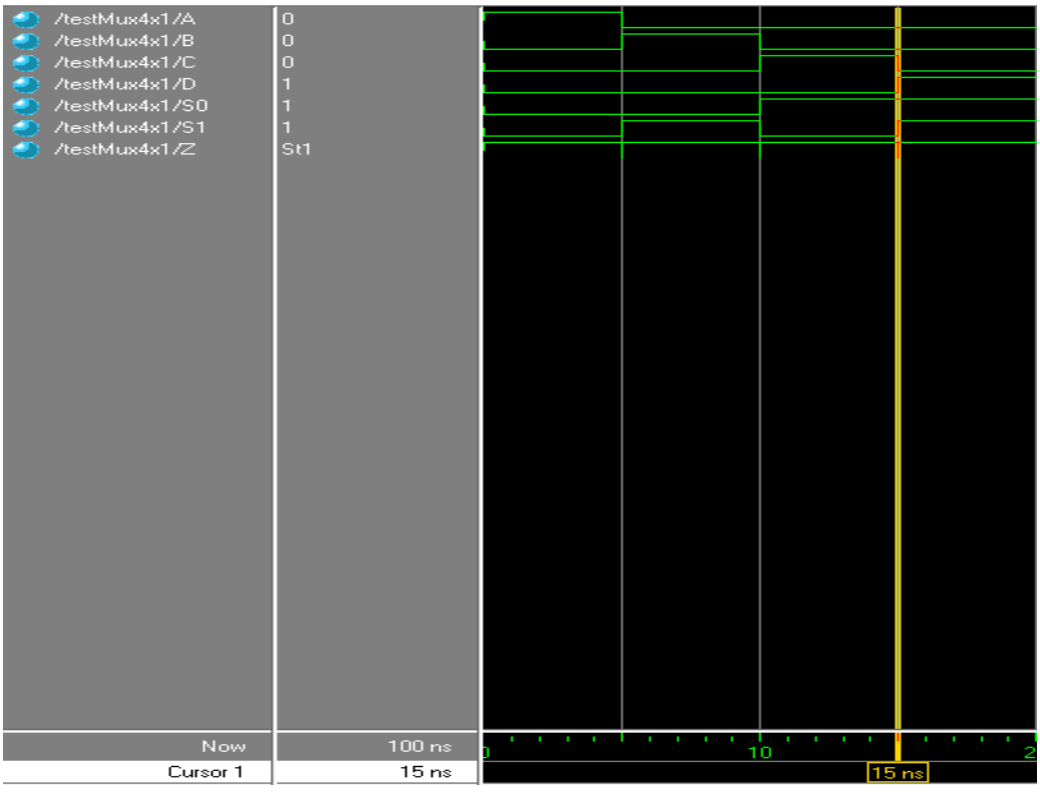
At 5ns:



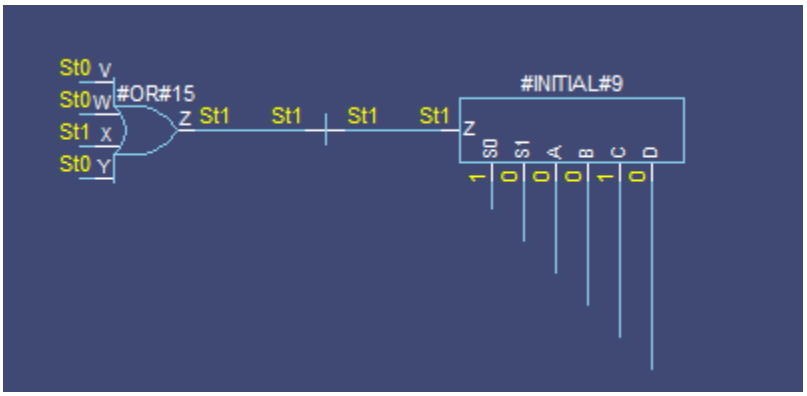
At 10ns:



At 15ns:



Dataflow:


St0 V
St0 W
St1 X
St0 Y
#OR#15
Z St1
St1
St1
St1
#INITIAL#9
Z
S0
S1
A
B
C
D
1
0
0
0
1
0