CA1-Report

Digital Systems I

Spring 1402-03
Digital Systems I – ECE894
Amirmortaza Rezaee
810101429

Problem 1

Figure 1.1 illustrates the circuit we are analyzing in this question:

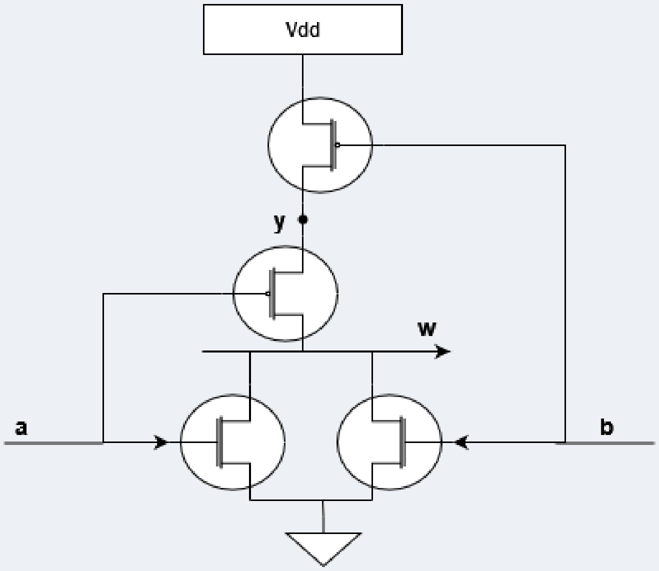


Figure 1.1 A simple NOR gate transistor design with inputs a, b and output w

Consider starting from 00, the inputs a and b of this NOR gate change at 70 ns time intervals as: 00 → 10 → 11 → 01 → 00 → 01.

Now let's find out how the waveform of the NOR gate output, w, changes:

- ***0 to 70 ns:** As it illustrated in Figure 2, after changing both of "a" and "b" values to 0, it takes 5 ns for each of pMOS transistors to propagate 1 (2×5=10 ns); also, it takes 5 ns for both nMOS transistors to propagate the Z-value to the output. So, for 10 ns we'll get Z-value on the output, and after that "w" will be 1.*

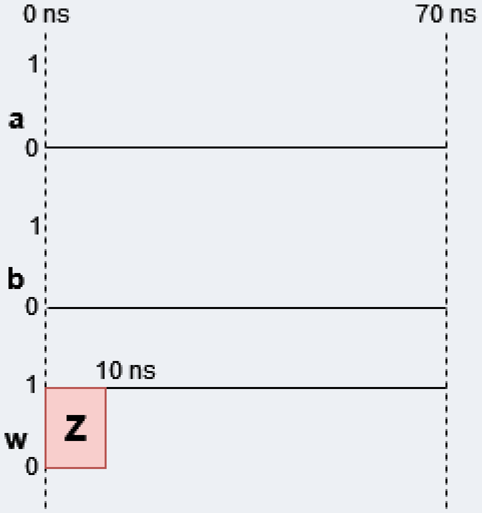


Figure 1.2

- ***70 to 140 ns:*** *As “a” turns to 1, it will take 7 ns for the second pMOS to propagate Z to the output; also, it’ll take 4 ns for the nMOS to propagate 0 to the output. So, there won’t be any changes in the output for 4 ns, but after that, we’ll get a X-value on the output for 3 ns. Then the output will be 0. (fig. 1.3)*

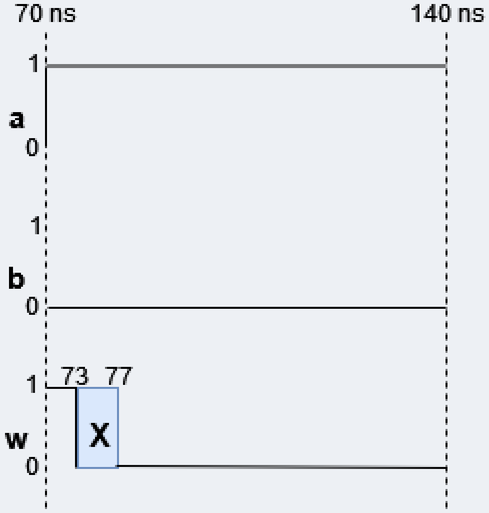


Figure 1.3

- ***140 to 210 ns:*** *As "b" turns to 1, the first pMOS propagates Z to the next one, and there won't be any changes in the output at all. So, the output won't change in this period.*

- ***210 to 280 ns:*** *As "a" turns to 0, the second pMOS conducts, but because its input is still Z, there will be no changes in the output.*

- ***280 to 350 ns:*** *As "b" turns to 0, first pMOS propagates 1 to the next one; so, there'll be two 5 ns delays which will sum up in order to propagating 1 to the output. Also, there is a 4 ns delay for nMOS transistors to propagating Z to the output. So, the output won't change for 5 ns, and for the next 5 ns, we'll get Z, after that the output will be 1 till end. (fig. 1.4)*

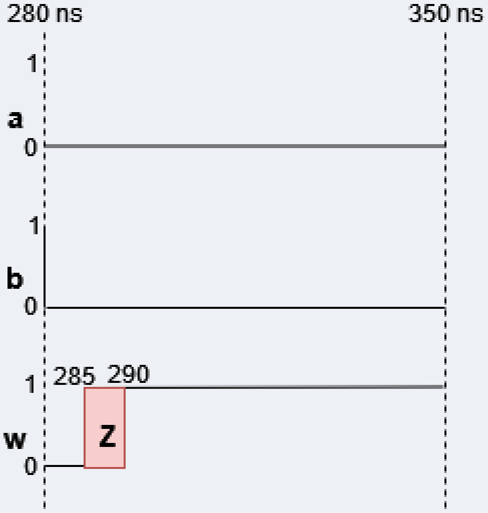


Figure 1.4

*As we saw, the worst-case delay of output making To-1 transition is the time in which both of pMOS transistors propagate 1 (2×5=10 ns) and nMOS transistors propagate Z (5 ns). So, the worst-case delay of output making To-1 transition equals to **10 ns**.*

- ***350 to 420 ns:*** *As "b" turns to 0, it costs 7 ns for the first and 7 ns for the second pMOS transistor to propagate Z to the output. Also, we need a 4 ns for the nMOS in order to propagating 0 to the output. So, the output will remain 1 for 4 ns, then we'll get X on the output for 10 ns till the pull up part propagates Z-value. (fig. 1.5)*

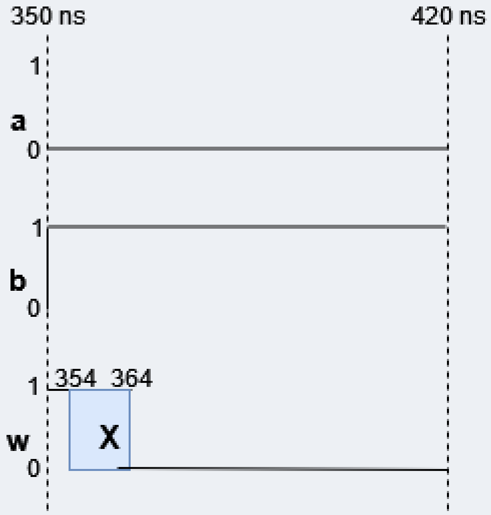
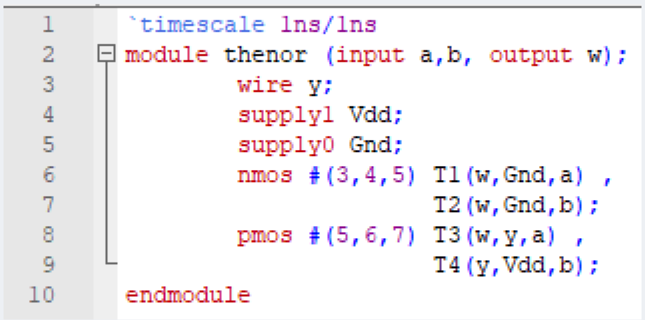


Figure 1.5

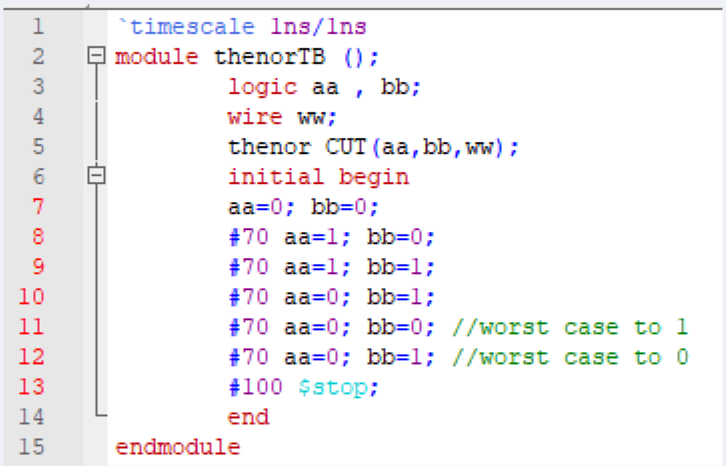
As mentioned, the worst-case delay of output making To-0 transition is the time in which both of pMOS transistors propagate Z (2×7=14 ns) and nMOS transistors propagate 0 (4 ns). and the worst-case delay of output making To-0 transition equals to ***14 ns****.*

Figures 1.6, 1.7 and 1.8 illustrate the Verilog description of this NOR gate and the testbench:



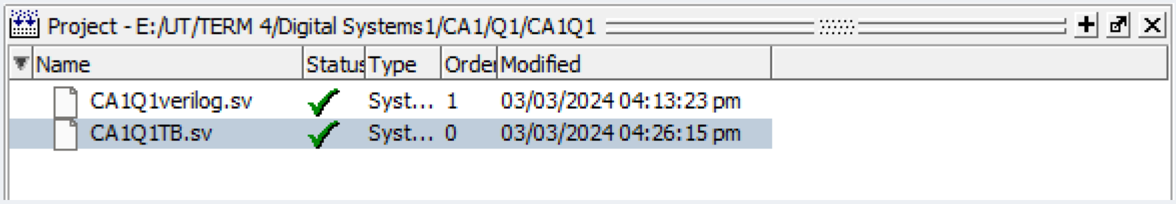
```
`timescale 1ns/1ns
module thenor (input a,b, output w);
        wire y;
        supply1 Vdd;
        supply0 Gnd;
        nmos #(3,4,5) T1(w,Gnd,a) ,
                      T2(w,Gnd,b);
        pmos #(5,6,7) T3(w,y,a) ,
                      T4(y,Vdd,b);
endmodule
```

Figure 1.6 Verilog description



```
`timescale 1ns/1ns
module thenorTB ();
        logic aa , bb;
        wire ww;
        thenor CUT(aa,bb,ww);
        initial begin
        aa=0; bb=0;
        #70 aa=1; bb=0;
        #70 aa=1; bb=1;
        #70 aa=0; bb=1;
        #70 aa=0; bb=0; //worst case to 1
        #70 aa=0; bb=1; //worst case to 0
        #100 $stop;
        end
endmodule
```

Figure 1.7 Verilog Testbench



Project - E:/UT/TERM 4/Digital Systems1/CA1/Q1/CA1Q1

Name	Statu	Type	Orde	Modified
CA1Q1verilog.sv	✓	Syst...	1	03/03/2024 04:13:23 pm
CA1Q1TB.sv	✓	Syst...	0	03/03/2024 04:26:15 pm

Figure 1.8

Figures 1.9 and 1.10 show the Verilog simulation results.

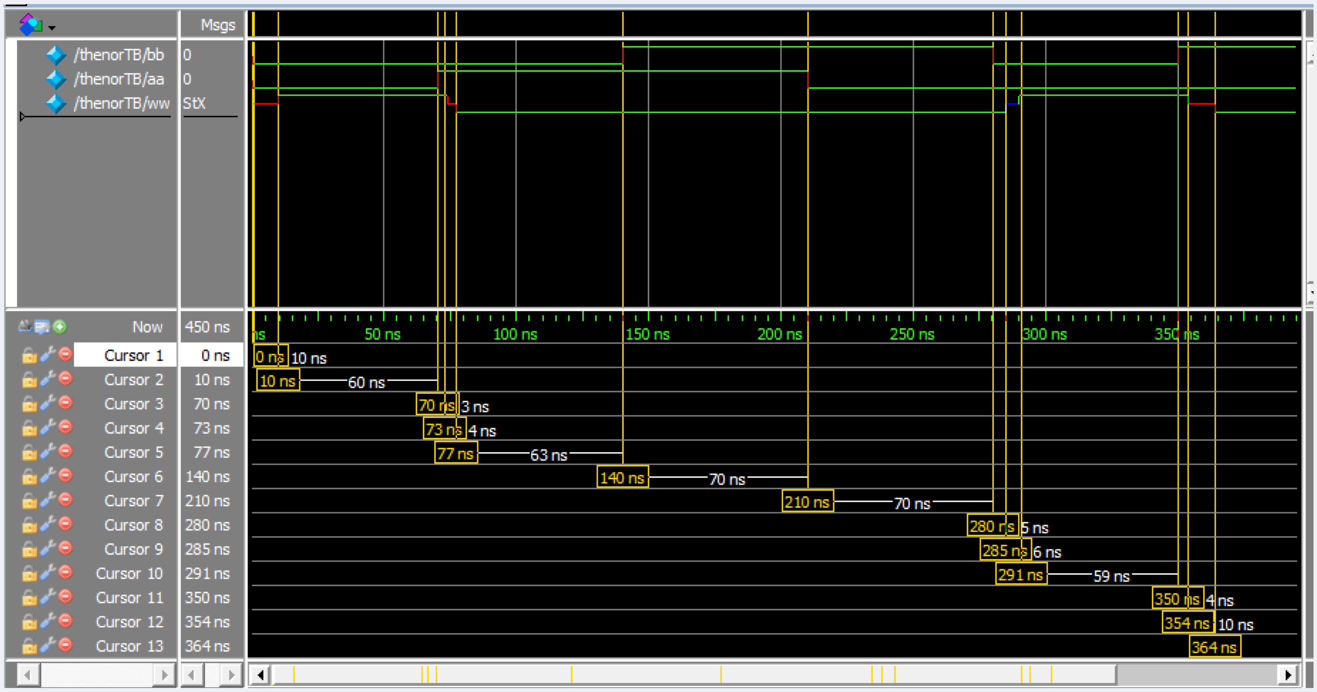


Figure 1.9

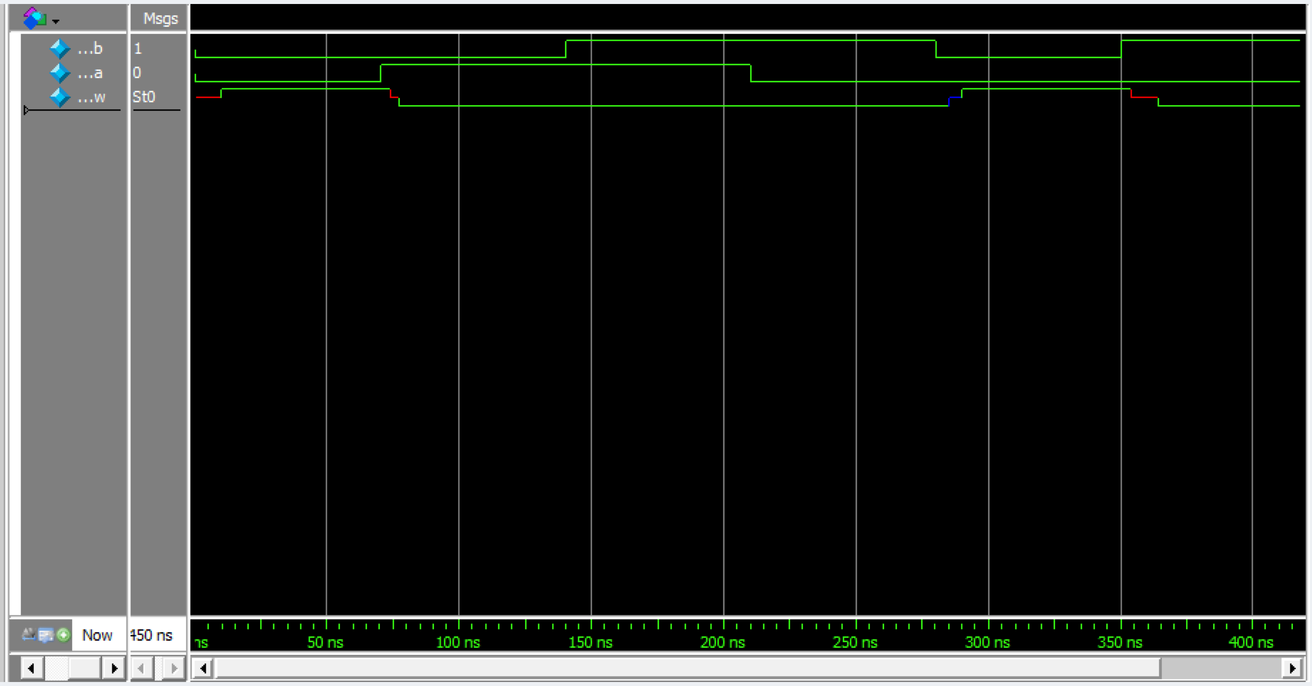


Figure 1.10

Problem 2

Figure 2.1 illustrates the circuit we are analyzing in this question:

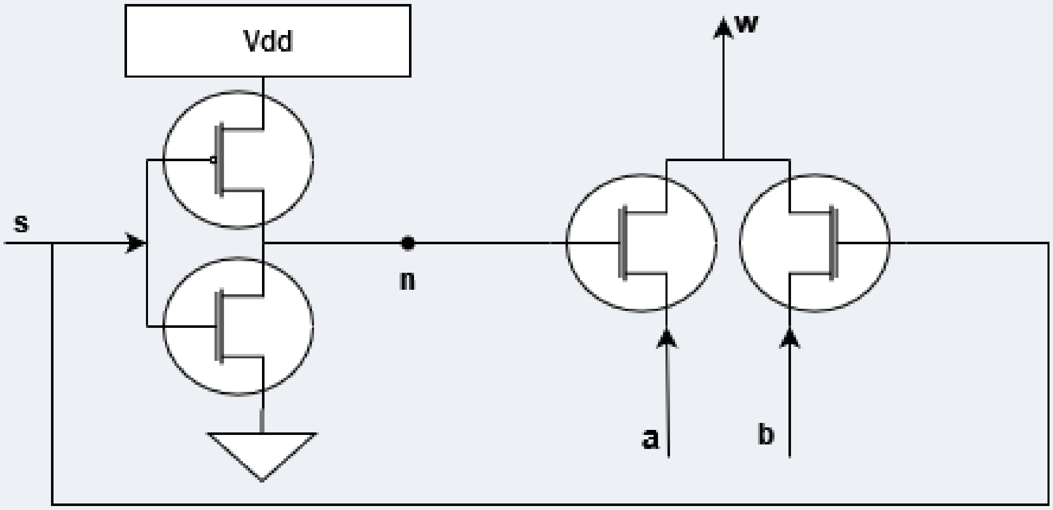


Figure 2.1 A pass-transistor based 2-to-1 multiplexer

(We named nMOS transistors from left to right as: T2, T3, T4)

Consider starting from 000, the inputs s, a and b of this multiplexer change at 70 ns time intervals as: 000 → 010 → 110 → 111 → 101 → 001 → 101 → 111.

Now let's find out how the waveform of the output, w, changes:

- ***0 to 70 ns:** At first, the value of "s" is 0, so the pMOS will conduct and it will propagate 1 after 5 ns; also, the first nMOS propagates Z after 7 ns. So, T3 will conduct 0 to the output after 4 ns and T4 will propagate Z after 5 ns. So, the output will be 0 after 9 ns of being X. (fig. 2.2)*

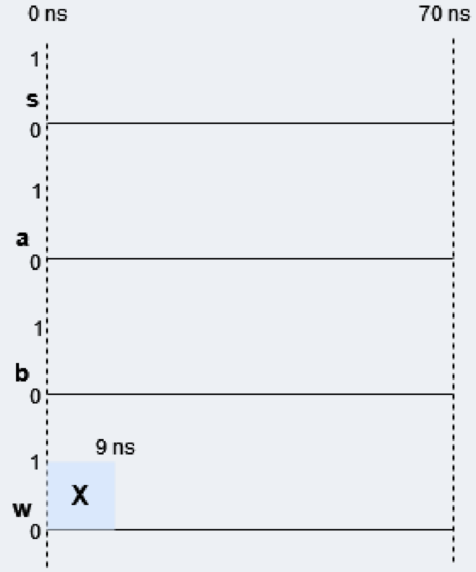


Figure 2.2

- ***70 to 140 ns:*** *As "a" 's value changes to 1, actually there won't be any changes at the node "n", and T3 will propagate 1 to the output after 3 ns. So, there won't be any changes in output till t=73 ns, then the output will be updated with 1. (fig. 2.3)*

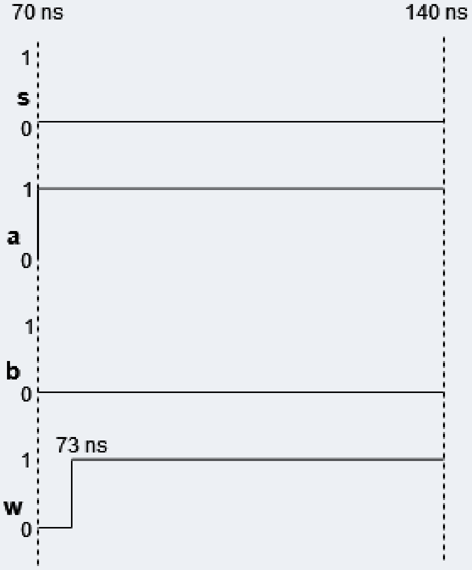


Figure 2.3

- ***140 to 210 ns:*** *As the value of "s" changes to 1, T2 will propagate 0 in 4 ns, but the pMOS will propagate Z after 7 ns. Then, T3 will propagate Z in 5 ns and T4 will propagate 0 in 4 ns. So, the output won't change in first 4 ns till T4 propagates 0, after that we'll get a X-value on the output until T3 propagates Z, this time will be* ***12 ns*** *at all (7 ns for the T1 and 5 ns for the T3). Actually, this is the worst-case delay of output making T0-0 transition. (fig. 2.4)*

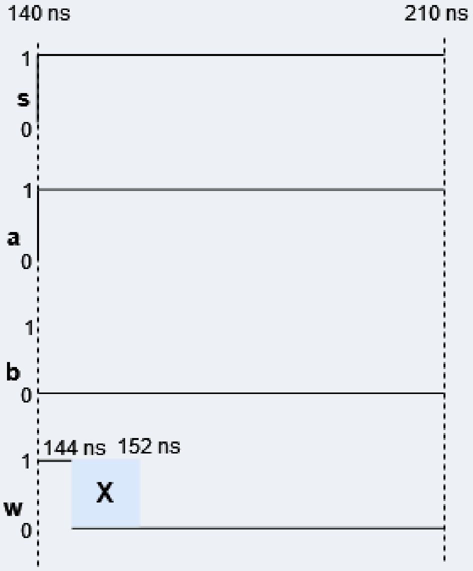


Figure 2.4

- ***210 to 280 ns:*** *As "b" changes to 1, there will be a 3 ns gap till T4 propagates 1 to the output. (fig. 2.5)*

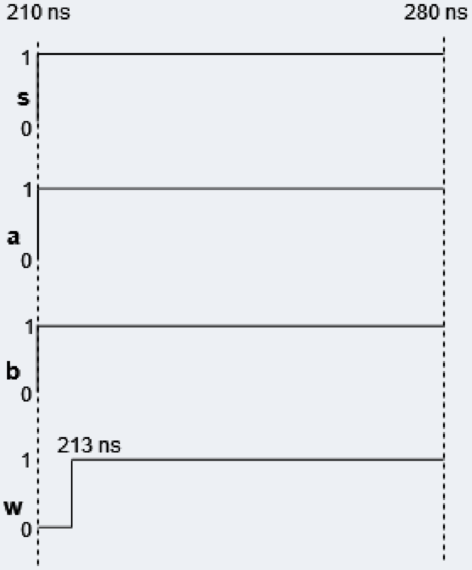


Figure 2.5

- ***280 to 350 ns:*** *As "a" changes to 1, there won't be any changes in the output because the controller of T3 is still 0.*

- ***350 to 420 ns:*** *After changing the value of "s" to 0, in 5 ns T4 will propagate Z and in this time the value of point "n" will be 1 (it takes 5 ns for T1 to propagate 1 and T2 to propagate Z). Then there will be a 4 ns gap till T3 propagates 0 to the output. (fig. 2.6)*

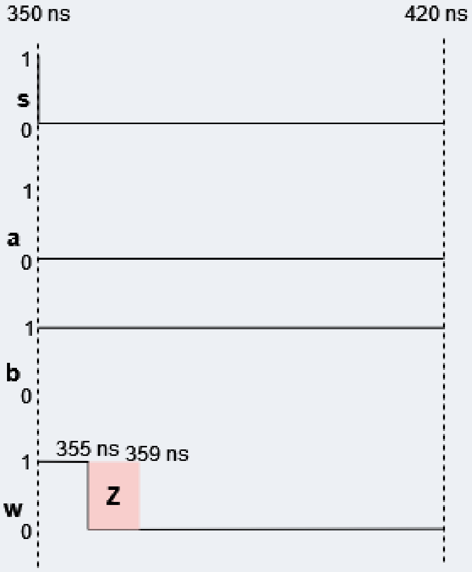


Figure 2.6

- ***420 to 490 ns**: As the value of "s" changes to 1, T2 will propagate 0 in 4 ns, but the pMOS will propagate Z after 7 ns. Then, T3 will propagate Z in 5 ns and T4 will propagate 1 in 3 ns. So, the output won't change in first 3 ns till T4 propagates 0, after that we'll get a X-value on the output until T3 propagates Z, this time will be **12 ns** at all (7 ns for the T1 and 5 ns for the T3). Actually, this is the worst-case delay of output making T0-1 transition. (fig. 2.7)*

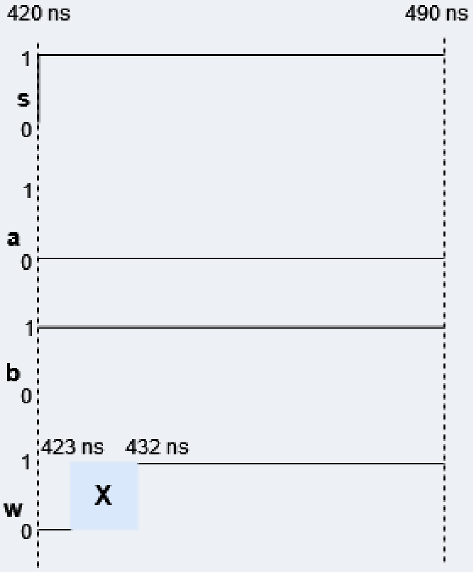
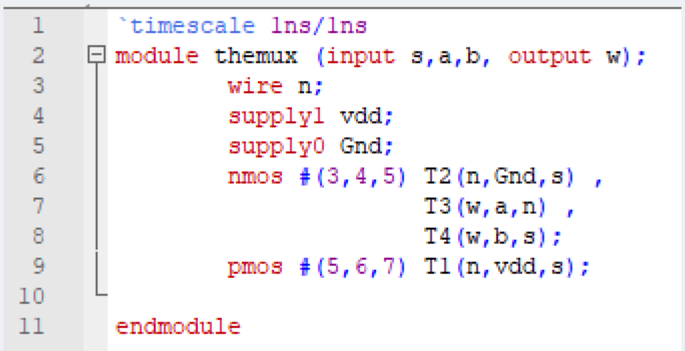


Figure 2.7

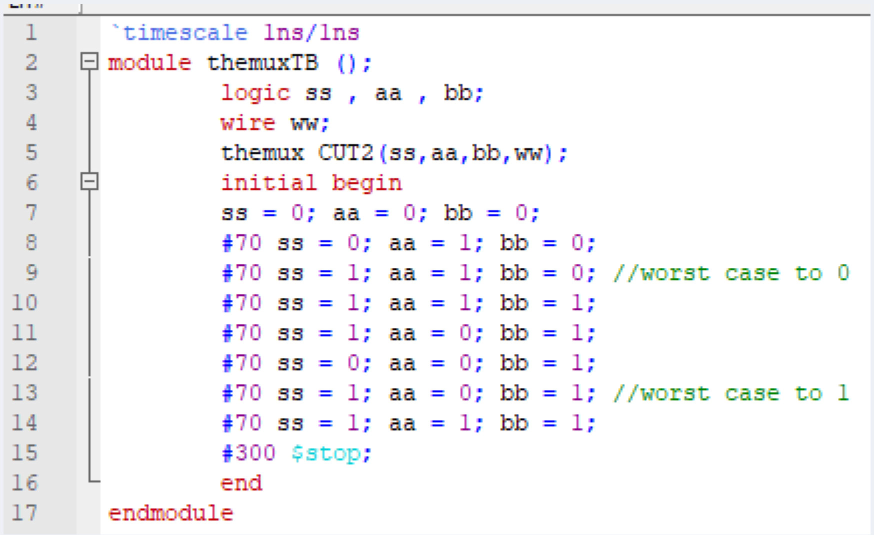
- ***490 to 560 ns:** As "a" changes to 1, there won't be any changes in the output because the controller of T3 is still 0.*

Figures 2.8, 2.9 and 2.10 illustrate the Verilog description of this multiplexer and the testbench:



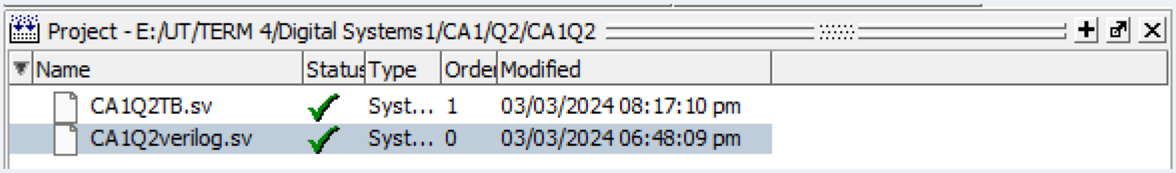
```
`timescale 1ns/1ns
module themux (input s,a,b, output w);
        wire n;
        supply1 vdd;
        supply0 Gnd;
        nmos #(3,4,5) T2(n,Gnd,s) ,
                      T3(w,a,n) ,
                      T4(w,b,s);
        pmos #(5,6,7) T1(n,vdd,s);

endmodule
```

Figure 2.8 Verilog description



```
`timescale 1ns/1ns
module themuxTB ();
        logic ss , aa , bb;
        wire ww;
        themux CUT2(ss,aa,bb,ww);
        initial begin
        ss = 0; aa = 0; bb = 0;
        #70 ss = 0; aa = 1; bb = 0;
        #70 ss = 1; aa = 1; bb = 0; //worst case to 0
        #70 ss = 1; aa = 1; bb = 1;
        #70 ss = 1; aa = 0; bb = 1;
        #70 ss = 0; aa = 0; bb = 1;
        #70 ss = 1; aa = 0; bb = 1; //worst case to 1
        #70 ss = 1; aa = 1; bb = 1;
        #300 $stop;
        end
endmodule
```

Figure 2.9 Verilog Testbench



Project - E:/UT/TERM 4/Digital Systems1/CA1/Q2/CA1Q2

Name	Status	Type	Order	Modified
CA1Q2TB.sv	✓	Syst...	1	03/03/2024 08:17:10 pm
CA1Q2verilog.sv	✓	Syst...	0	03/03/2024 06:48:09 pm

Figure 2.10

Figures 2.11 and 2.12 show the Verilog simulation results.

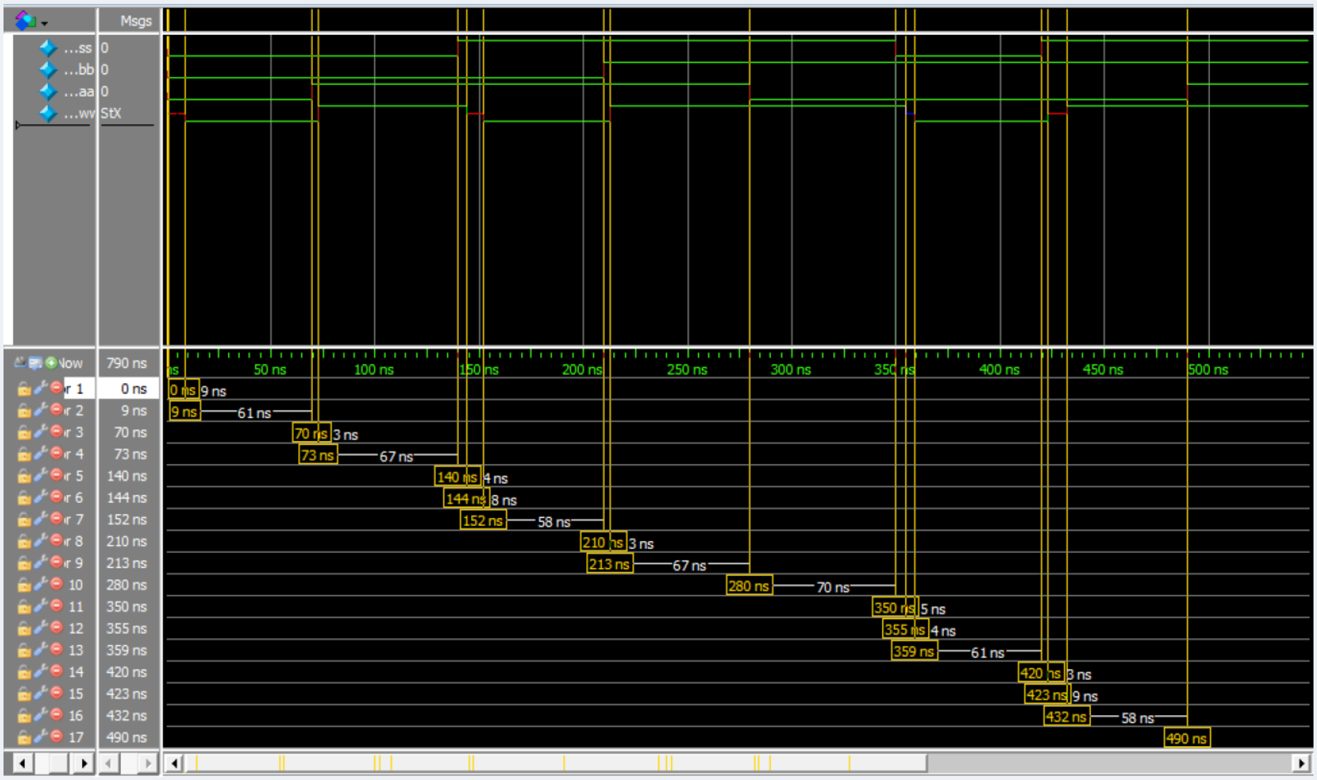


Figure 2.11

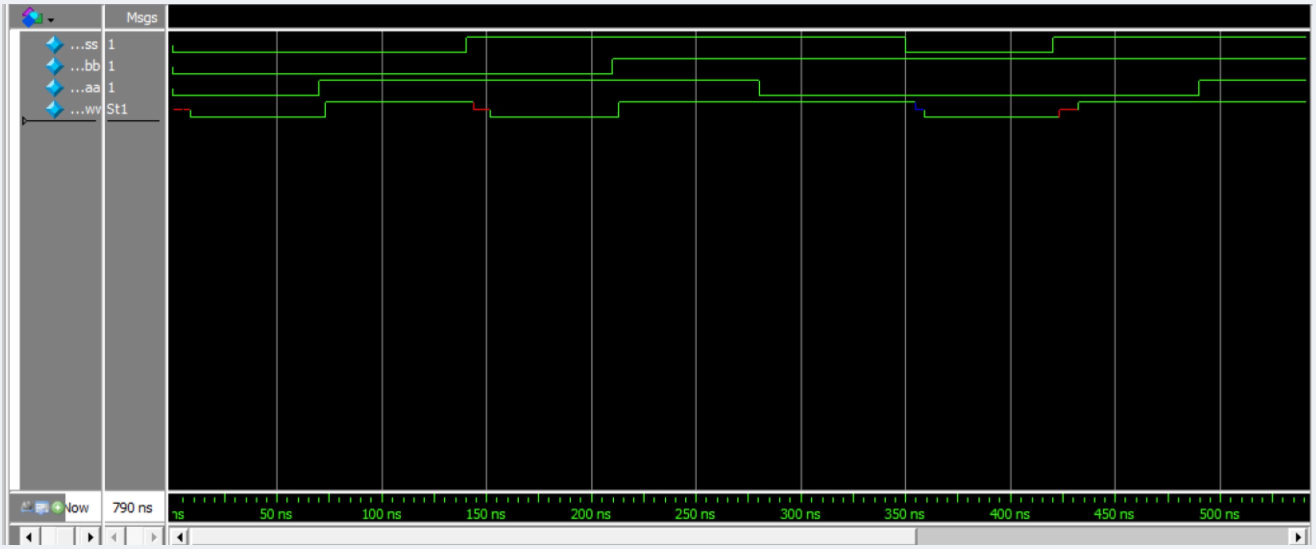


Figure 2.12

Problem 3

Figure 3.1 shows the truth table of a 4-to-1 MUX with two select inputs, s1 and s0, and four data inputs a, b, c, and d and the output y:

S1	S0	y
0	0	a
0	1	b
1	0	c
1	1	d

Figure 3.1

As in the table, the output "a" occurs when both of s1 and s0 are 0. So, in our structure, "a" should be Nored with both of s1 and s0. Although, the output will be "b" if s1 is 0 and s0 is 1; then we should NOR it with s1 and ~s0. Likewise, we have to NOR "c", ~s1 and s0 together and "d" with ~s1 and ~s0. At the end, we should NOR the outputs together.

So, the gate-level design of this MUX should be as described below:

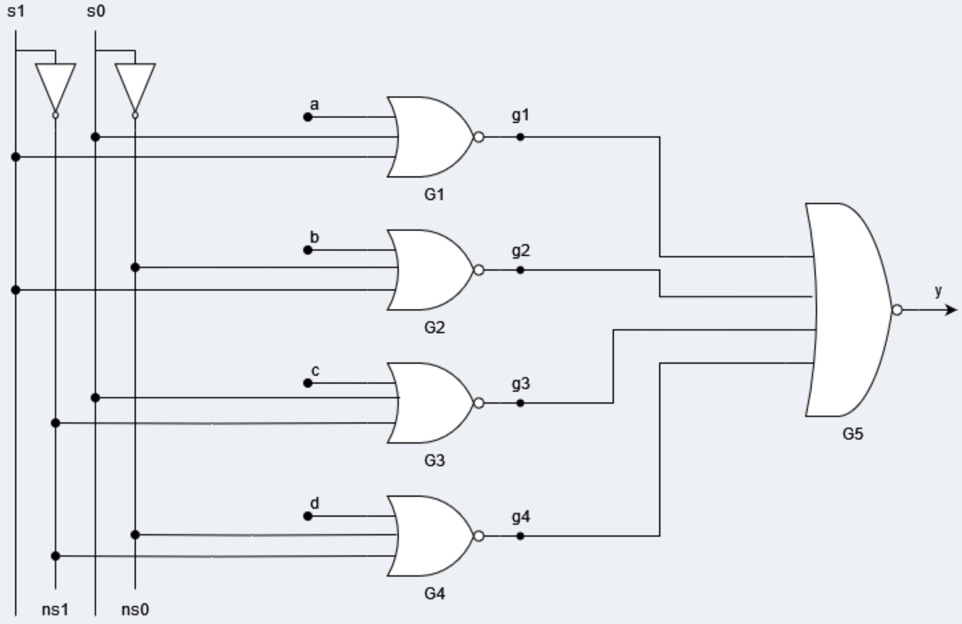


Figure 3.2 4-to-1 MUX

As described in figure 3.2, we can make a 4-to-1 MUX using 5 NOR gates and 2 NOT gates. But we can create a NOT structure using a single NOR gate. (fig. 3.3)

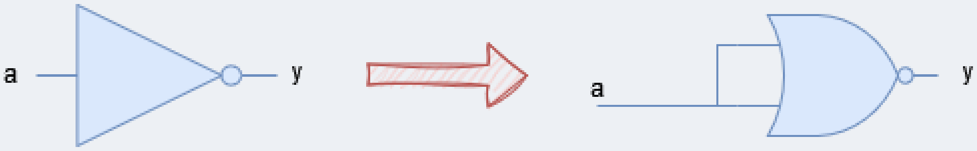


Figure 3.3 a NOR gate used as a NOT gate

So, we can regenerate our 4-to-1 MUX. (fig. 3.4)

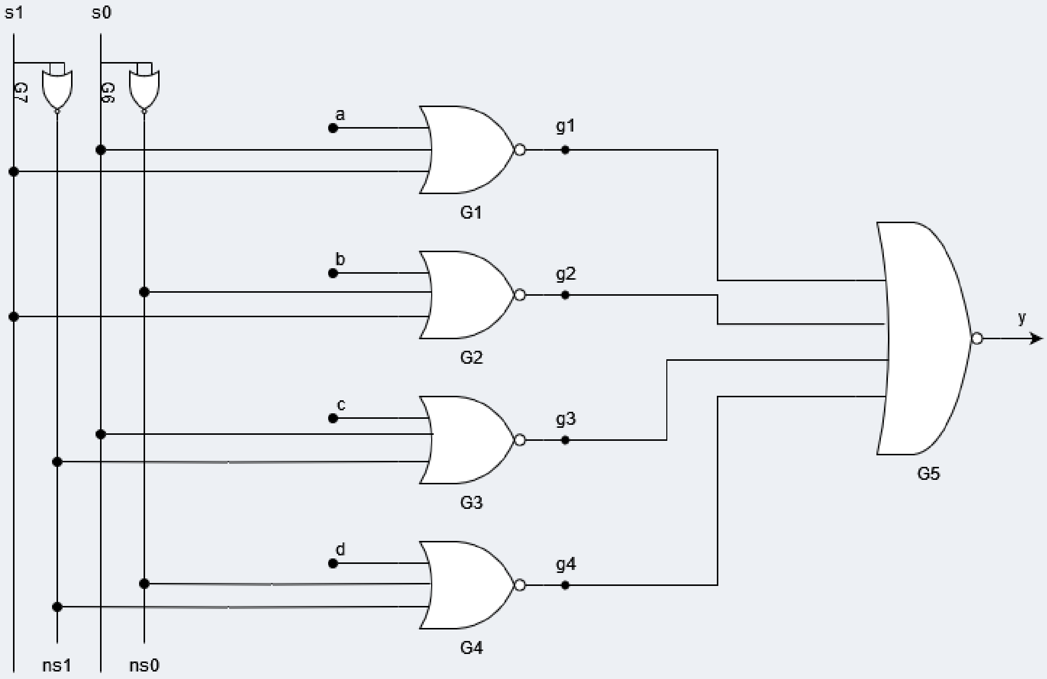


Figure 3.4 4-to-1 MUX using nor gates

Actually, we've designed a 4-to-1 MUX only using NOR gates. But there is still a problem: Timings!

So, let's discover worst-case delays for NOR gates with different input numbers.

I) *As solved before (problem 1), for a NOR gate with 2 inputs, the worst-case delay of output making T0-1 transition equals to* ***10 ns****. And the worst-case delay of output making T0-0 transition equals to* ***14 ns****.*

II) *The switch-level structure of a 3-input NOR gate looks like figure 3.5:*

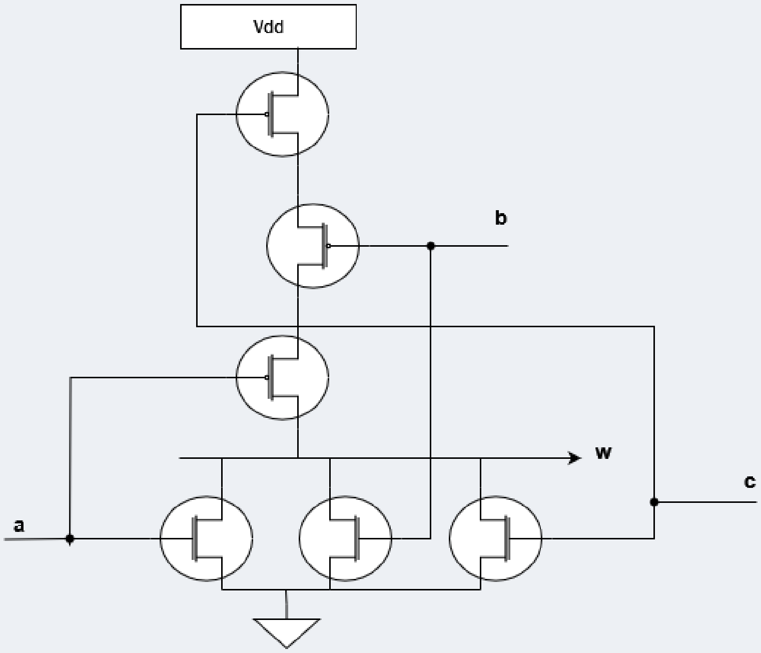


Figure 3.5 switch-level structure of a 3-input NOR

the worst-case delay of output making T0-1 transition is the time in which all of pMOS transistors propagate 1 (3×5=15 ns) and nMOS transistors propagate Z (5 ns). So, the worst-case delay of output making T0-1 transition equals to ***15 ns****.*

Also, the worst-case delay of output making T0-0 transition is the time in which all of pMOS transistors propagate Z (3×7=21 ns) and nMOS transistors propagate 0 (4 ns).and the worst-case delay of output making T0-0 transition equals to ***21 ns****.*

III) Just like previous part, we can easily derive worst-case delays for a 4-input NOR gate as well.

*So, the worst-case delay of output making To-1 transition is the time in which all of pMOS transistors propagate 1 (4×5=20 ns) and nMOS transistors propagate Z (5 ns). So, the worst-case delay of output making To-1 transition equals to **20 ns**.*

*Also, the worst-case delay of output making To-0 transition is the time in which all of pMOS transistors propagate Z (4×7=28 ns) and nMOS transistors propagate 0 (4 ns).and the worst-case delay of output making To-0 transition equals to **28 ns**.*

Now let's calculate the structures worst-case delay of output making To-0 transition; The output will change from 1 to 0 when all of the gates of the second layer, result in 0 then one of them changes to 1, this means one of its inputs changes to 0. at the worst case, this will happen when that input is "ns0" or "ns1". So, the final output changes into 0 after:

1. ***14 ns:** the time in which the inverters output changes to 0*
2. ***15 ns:** the time it costs for one of the 3-input NOR gates changing to 1*
3. ***28 ns:** the time that the last gate's output changes to 0*

*So, the worst-case delay of output making To-0 transition equals to (14+15+28=) **57 ns**.*

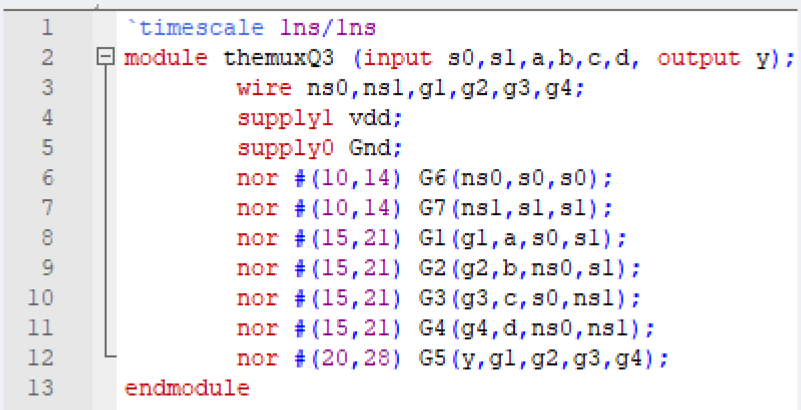
(this delay derived by the testbench and the result will be figured later)

Now let’s calculate the structures worst-case delay of output making To-1 transition; The output will change from 0 to 1 when all of the gates of the second layer except one of them, result in 0. Then the exception changes to 0, this means one of its input’s changes to 1 when two other inputs are still 0. at the worst case, this will happen when that input is “ns0” or “ns1”. So, the final output changes into 0 after:

*1- **10 ns:** the time in which the inverters output changes to 1*
*2- **21 ns:** the time it costs for one of the 3-input NOR gates changing to 0*
*3- **20 ns:** the time that the last gate’s output changes to 1*

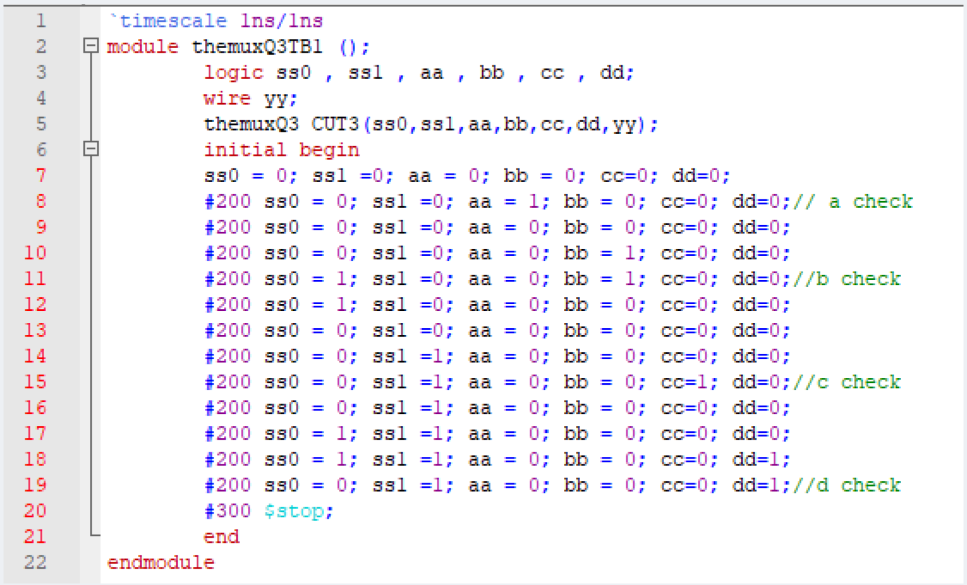
*So, the worst-case delay of output making To-0 transition equals to (10+21+20=) **51 ns**.*
(this delay derived by the testbench and the result will be figured later)

Figures 3.6, 3.7,3.8 and 3.9 illustrate the Verilog description of this multiplexer and the testbench:



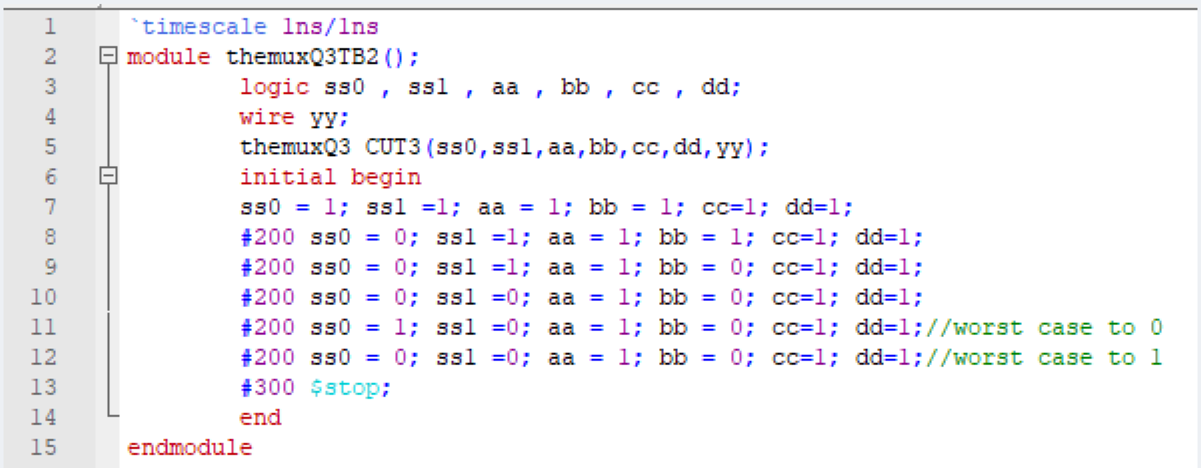
```
`timescale 1ns/1ns
module themuxQ3 (input s0,s1,a,b,c,d, output y);
        wire ns0,ns1,g1,g2,g3,g4;
        supply1 vdd;
        supply0 Gnd;
        nor #(10,14) G6(ns0,s0,s0);
        nor #(10,14) G7(ns1,s1,s1);
        nor #(15,21) G1(g1,a,s0,s1);
        nor #(15,21) G2(g2,b,ns0,s1);
        nor #(15,21) G3(g3,c,s0,ns1);
        nor #(15,21) G4(g4,d,ns0,ns1);
        nor #(20,28) G5(y,g1,g2,g3,g4);
endmodule
```

Figure 3.6 Verilog description



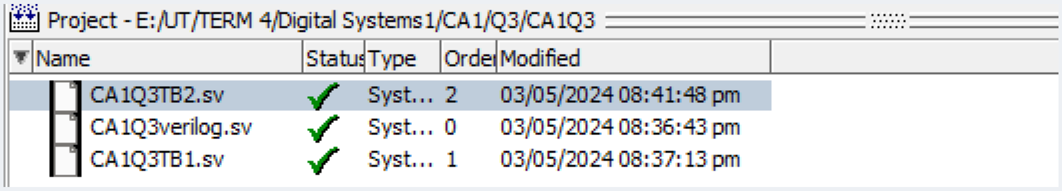
```
`timescale 1ns/1ns
module themuxQ3TB1 ();
        logic ss0 , ss1 , aa , bb , cc , dd;
        wire yy;
        themuxQ3 CUT3(ss0,ss1,aa,bb,cc,dd,yy);
        initial begin
        ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 1; bb = 0; cc=0; dd=0;// a check
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 1; cc=0; dd=0;
        #200 ss0 = 1; ss1 =0; aa = 0; bb = 1; cc=0; dd=0;//b check
        #200 ss0 = 1; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=1; dd=0;//c check
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 1; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 1; ss1 =1; aa = 0; bb = 0; cc=0; dd=1;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=1;//d check
        #300 $stop;
        end
endmodule
```

Figure 3.7 Verilog first Testbench



```
`timescale 1ns/1ns
module themuxQ3TB2();
        logic ss0 , ss1 , aa , bb , cc , dd;
        wire yy;
        themuxQ3 CUT3(ss0,ss1,aa,bb,cc,dd,yy);
        initial begin
        ss0 = 1; ss1 =1; aa = 1; bb = 1; cc=1; dd=1;
        #200 ss0 = 0; ss1 =1; aa = 1; bb = 1; cc=1; dd=1;
        #200 ss0 = 0; ss1 =1; aa = 1; bb = 0; cc=1; dd=1;
        #200 ss0 = 0; ss1 =0; aa = 1; bb = 0; cc=1; dd=1;
        #200 ss0 = 1; ss1 =0; aa = 1; bb = 0; cc=1; dd=1;//worst case to 0
        #200 ss0 = 0; ss1 =0; aa = 1; bb = 0; cc=1; dd=1;//worst case to 1
        #300 $stop;
        end
endmodule
```

Figure 3.8 Verilog Testbench for worst-case delays



Project - E:/UT/TERM 4/Digital Systems1/CA1/Q3/CA1Q3

Name	Status	Type	Order	Modified
CA1Q3TB2.sv	✓	Syst...	2	03/05/2024 08:41:48 pm
CA1Q3verilog.sv	✓	Syst...	0	03/05/2024 08:36:43 pm
CA1Q3TB1.sv	✓	Syst...	1	03/05/2024 08:37:13 pm

Figure 3.9

Figures 3.10 and 3.11 show the Verilog simulation results.

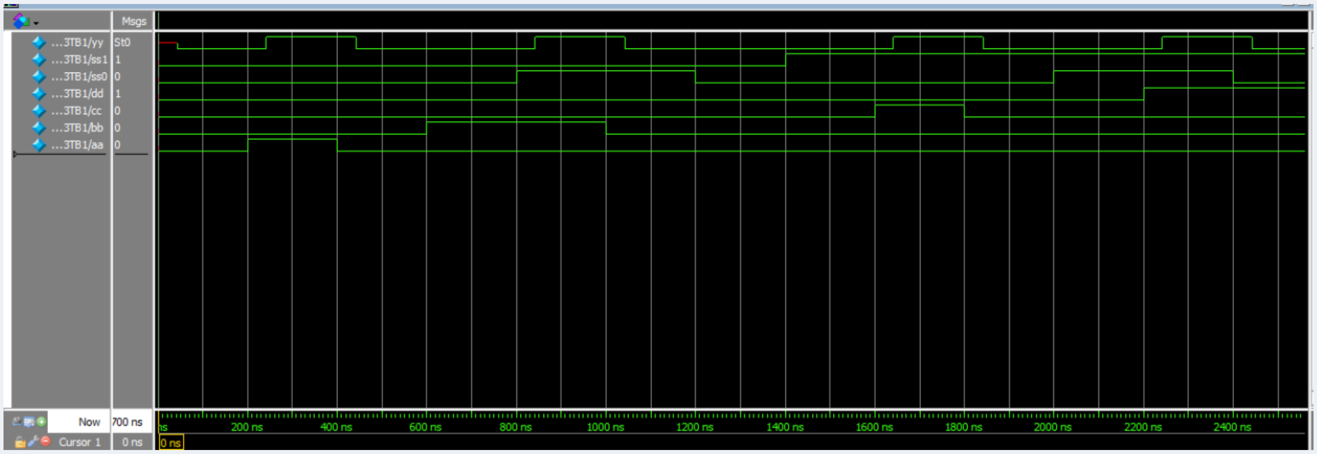


Figure 3.10 Verilog simulation for the first testbench

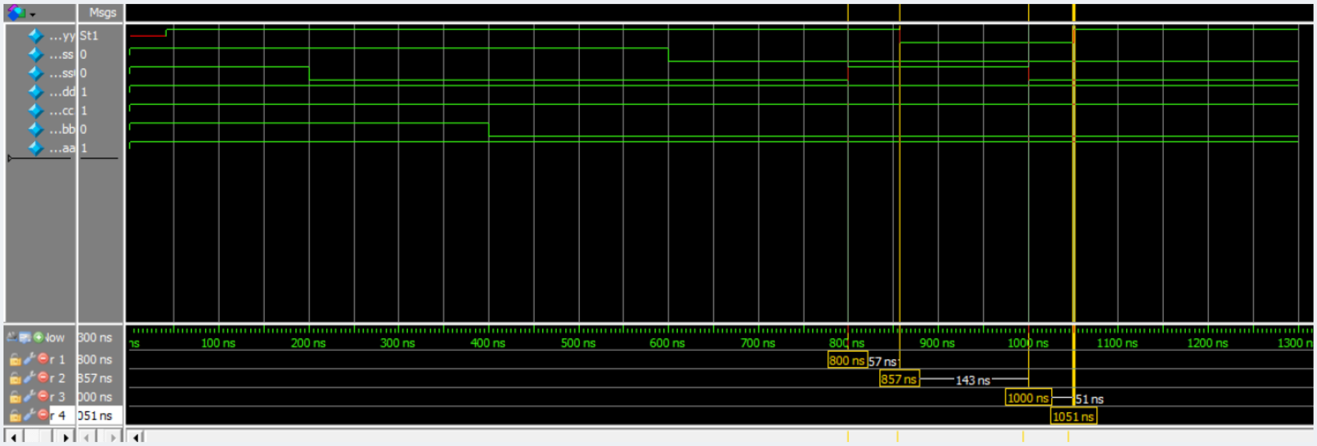


Figure 3.11 Verilog simulation for the worst-case delays testbench

Problem 4

Figure 4.1 illustrates the structure of a 4-to-1 MUX with two select inputs, s1 and s0, and four data inputs a, b, c, and d Using several 2-to-1 multiplexers:

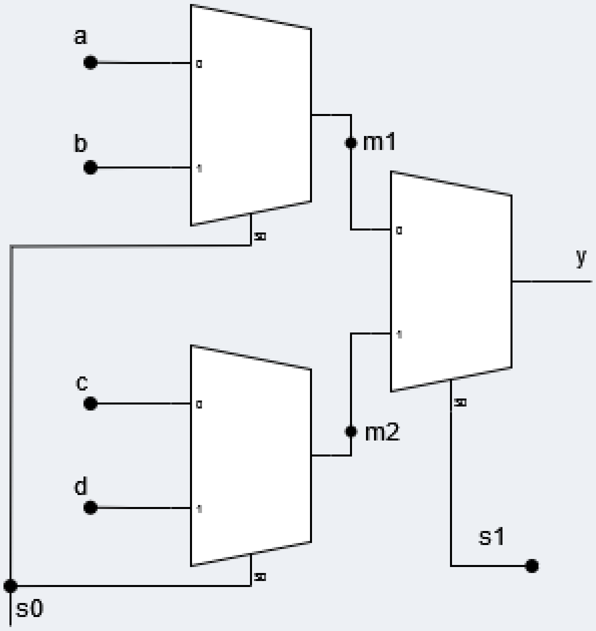


Figure 4.1 a 4-to-1 MUX with two select inputs and four data inputs

Figure 4.2 shows the truth table of this structure.

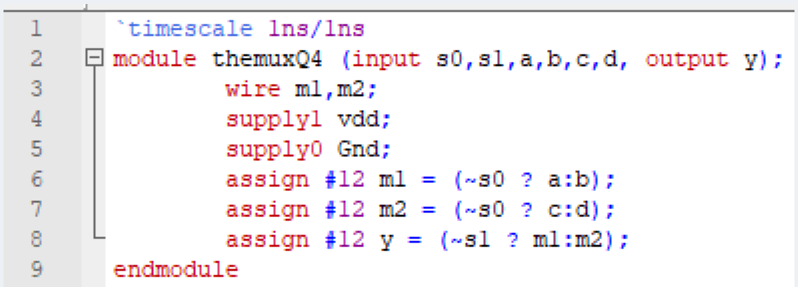
S1	S0	y
0	0	a
0	1	b
1	0	c
1	1	d

Figure 4.2

*As calculated in problem 2, for each of the 2-to-1 MUX's, both worst-case delay of output making To-1 and To-0 transition is **12 ns**.*

*So, the worst-case delay of output making To-1 and To-0 transition for this 4-to-1 MUX structure will be apparently (12 + 12=) **24 ns**.*

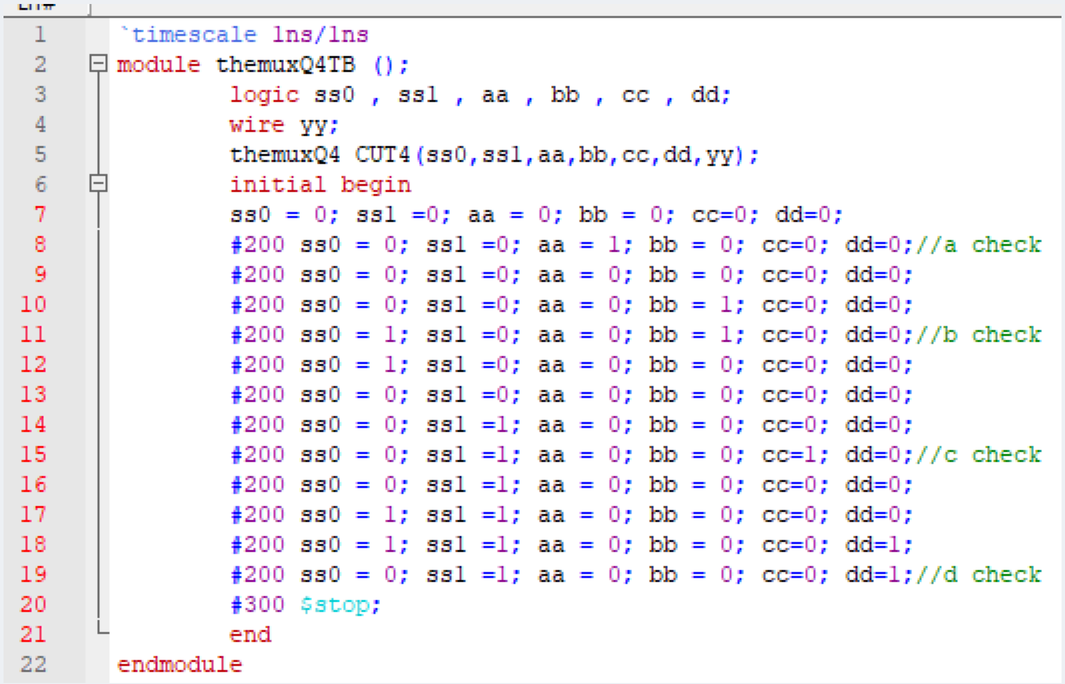
Here we can describe our 2-to-1 MUX gates with "assign" and the delay value of 12 ns in Verilog as it is illustrated in figure 4.3.



```
`timescale 1ns/1ns
module themuxQ4 (input s0,s1,a,b,c,d, output y);
        wire m1,m2;
        supply1 vdd;
        supply0 Gnd;
        assign #12 m1 = (~s0 ? a:b);
        assign #12 m2 = (~s0 ? c:d);
        assign #12 y = (~s1 ? m1:m2);
endmodule
```

Figure 4.3 Verilog description

Also figure 4.4 shows the test bench for this structure:



```
`timescale 1ns/1ns
module themuxQ4TB ();
        logic ss0 , ss1 , aa , bb , cc , dd;
        wire yy;
        themuxQ4 CUT4(ss0,ss1,aa,bb,cc,dd,yy);
        initial begin
        ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 1; bb = 0; cc=0; dd=0;//a check
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 1; cc=0; dd=0;
        #200 ss0 = 1; ss1 =0; aa = 0; bb = 1; cc=0; dd=0;//b check
        #200 ss0 = 1; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=1; dd=0;//c check
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 1; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 1; ss1 =1; aa = 0; bb = 0; cc=0; dd=1;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=1;//d check
        #300 $stop;
        end
endmodule
```

Figure 4.4 Verilog Testbench

Figure 4.5 shows the Verilog simulation results. As it is obvious, the worst-case delay times are 12 ns for both To-1 and To-0 transitions:

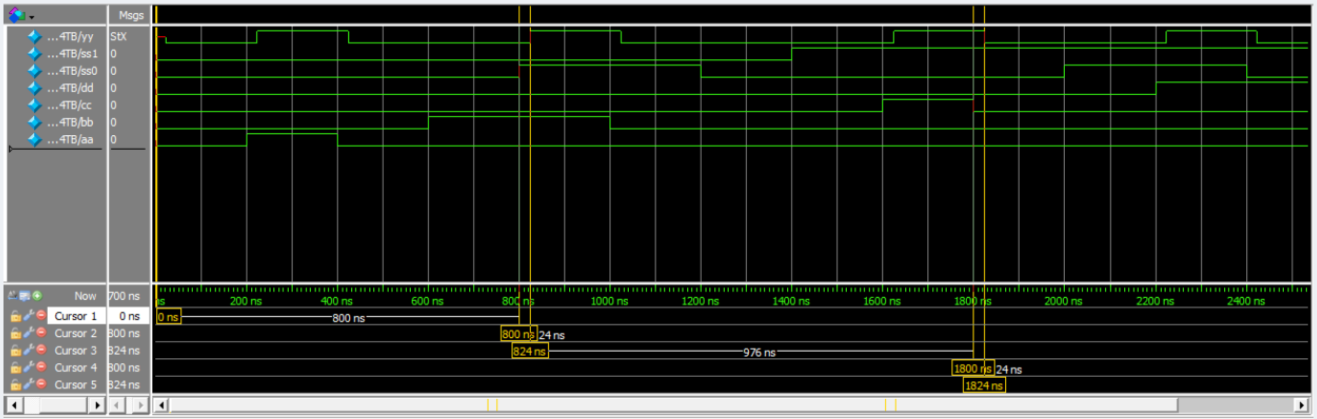
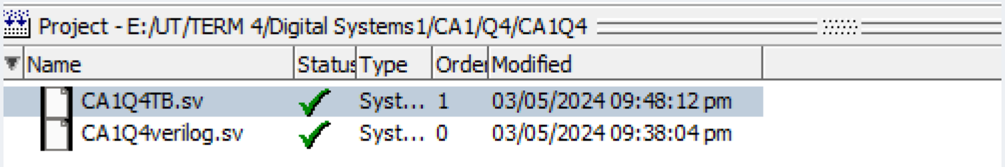


Figure 4.5



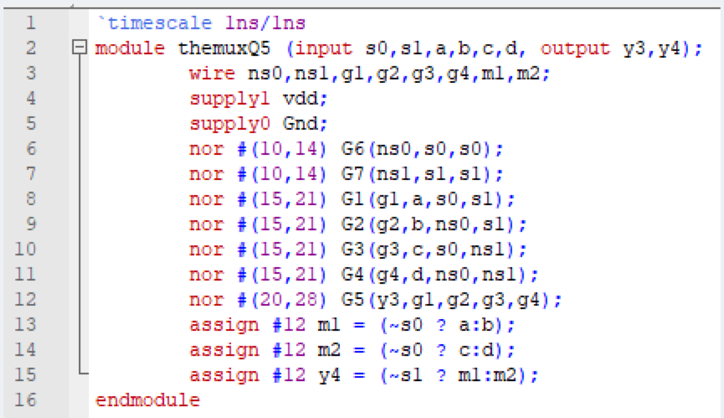
Project - E:/UT/TERM 4/Digital Systems1/CA1/Q4/CA1Q4

Name	Status	Type	Order	Modified
CA1Q4TB.sv	✓	Syst...	1	03/05/2024 09:48:12 pm
CA1Q4verilog.sv	✓	Syst...	0	03/05/2024 09:38:04 pm

Figure 4.6

Problem 5

For this problem, a single Verilog code file has been created in which, both of problem3 and problem4 structures are implemented. (fig. 5.1)

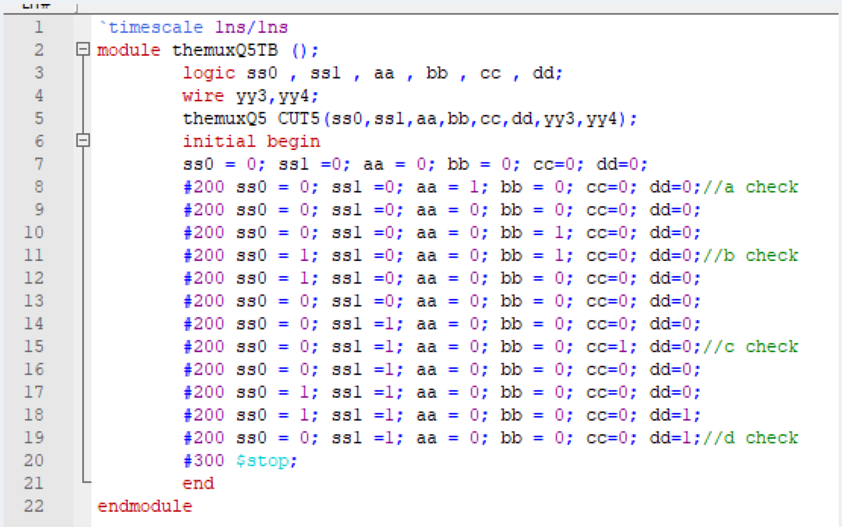


```verilog
`timescale 1ns/1ns
module themuxQ5 (input s0,s1,a,b,c,d, output y3,y4);
        wire ns0,ns1,g1,g2,g3,g4,m1,m2;
        supply1 vdd;
        supply0 Gnd;
        nor #(10,14) G6(ns0,s0,s0);
        nor #(10,14) G7(ns1,s1,s1);
        nor #(15,21) G1(g1,a,s0,s1);
        nor #(15,21) G2(g2,b,ns0,s1);
        nor #(15,21) G3(g3,c,s0,ns1);
        nor #(15,21) G4(g4,d,ns0,ns1);
        nor #(20,28) G5(y3,g1,g2,g3,g4);
        assign #12 m1 = (~s0 ? a:b);
        assign #12 m2 = (~s0 ? c:d);
        assign #12 y4 = (~s1 ? m1:m2);
endmodule
```

Figure 5.1 Verilog description

In the code above, the inputs of two structures are same and there are two outputs: "y3" and "y4"; where "y3" is the output of the structure from problem3 and "y4" is the output of the problem4's structure.

The testbench for this project comes below (fig. 5.2). In this testbench, inputs are same and the change of two outputs, examined by the change of inputs.



```verilog
`timescale 1ns/1ns
module themuxQ5TB ();
        logic ss0 , ss1 , aa , bb , cc , dd;
        wire yy3,yy4;
        themuxQ5 CUT5(ss0,ss1,aa,bb,cc,dd,yy3,yy4);
        initial begin
        ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 1; bb = 0; cc=0; dd=0;//a check
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 1; cc=0; dd=0;
        #200 ss0 = 1; ss1 =0; aa = 0; bb = 1; cc=0; dd=0;//b check
        #200 ss0 = 1; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =0; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=1; dd=0;//c check
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 1; ss1 =1; aa = 0; bb = 0; cc=0; dd=0;
        #200 ss0 = 1; ss1 =1; aa = 0; bb = 0; cc=0; dd=1;
        #200 ss0 = 0; ss1 =1; aa = 0; bb = 0; cc=0; dd=1;//d check
        #300 $stop;
        end
endmodule
```

Figure 5.2 Verilog Testbench

The simulation's results, illustrated in figure 5.3:

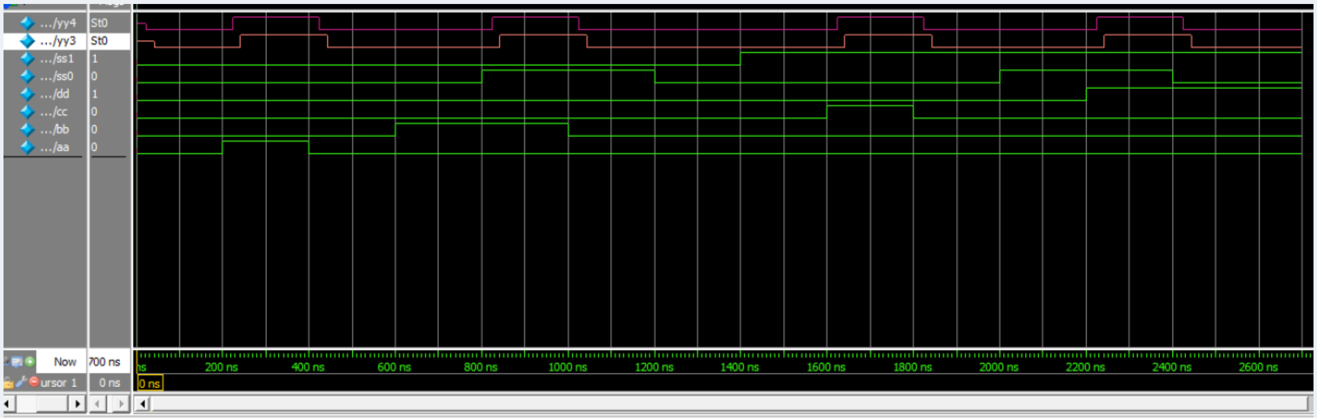


Figure 5.3

As expected, the output of the 4th problem's structure, changes with a less delay in comparison with the 3rd structure.

In the NOR gate structure, we have to use two 2-input, four 3-input and a 4-input NOR gate; which means there are (2×4 + 4×6 + 8 =) ***40*** *transistors at all in this struct. On the other hand, we have three 2-to-1 MUX gates that there are 4 transistors in each. So, there are* ***12*** *in total.*

So, the MUX with NOR gates, consumes more power than the other one in the result of the difference in the number of transistors.