به نام خدا
دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر

سیستم های دیجیتال 1

--- ECE 894 ---

استاد: پروفسور نوابی

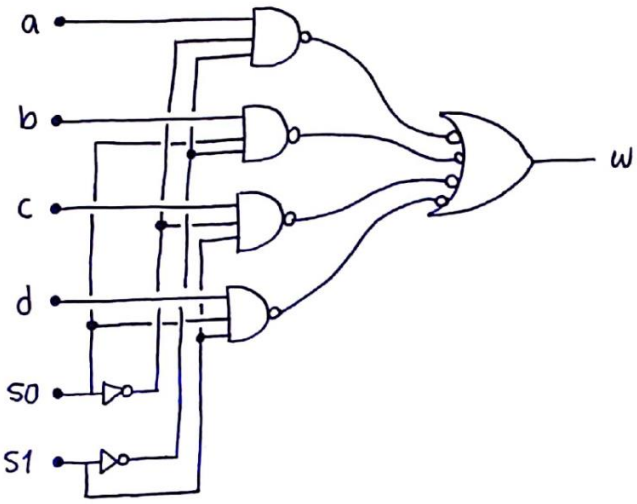
Computer Assignment 2

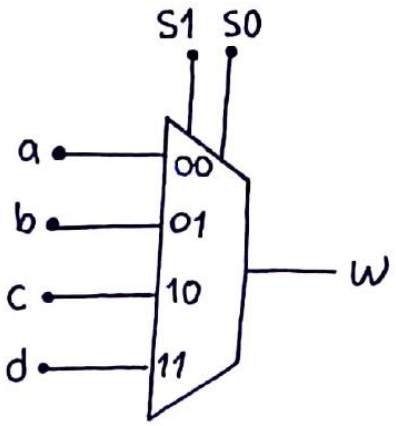
محمدمهدی عبدالحسینی

810 198 434

Part 1

THE MODIFIED VERSION OF 4TO1 MUX OF PART 3 OF CA1 :





Calculating Delay Values :

3-inputs NAND #(15,12) ➔ #(13.5)

4-inputs NAND #(20,16) ➔ #(18)

Inverter : NOT #(5,7) ➔ #(6)

➔➔➔ 4to1 MUX Delay Value = 13.5 + 18 + 6 = **37.5**

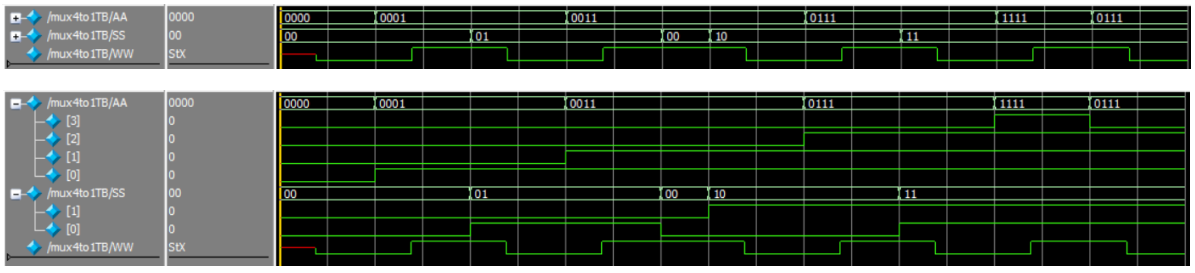
Assign Statement :

```
1   `timescale 1ns/1ns
2   module mux4to1 (input [3:0] A, input [1:0] S, output w);
3
4           assign #37.5 w = A[S];
5
6   endmodule
```

Testbench Code :

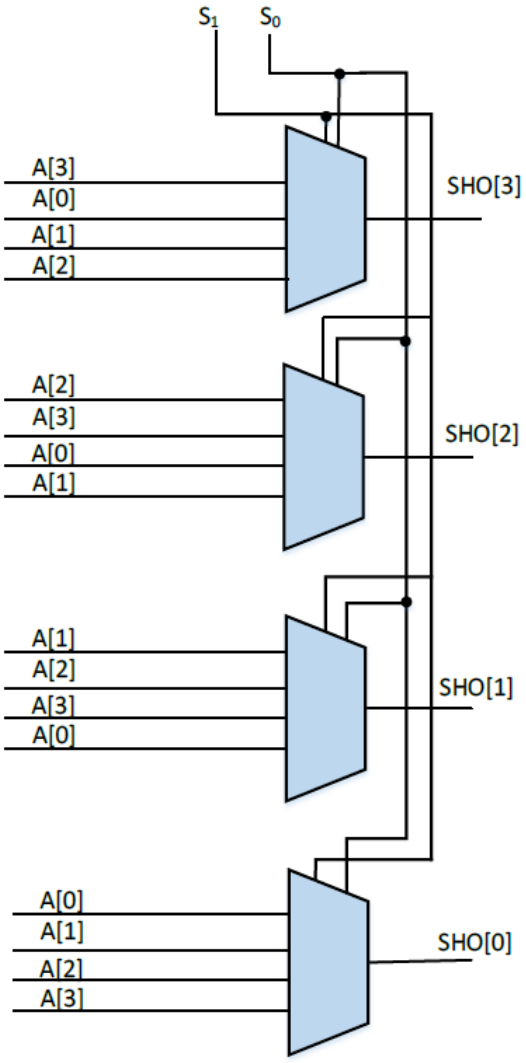
```
1   `timescale 1ns/1ns
2   module mux4to1TB ();
3           reg [3:0] AA = 0;
4           reg [1:0] SS = 0;
5           wire WW;
6           mux4to1 MUX1(AA, SS, WW);
7           initial begin
8           #100 AA[0] = 1;
9           #100 SS[0] = 1;
10          #100 AA[1] = 1;
11          #100 SS[0] = 0;
12          #50 SS[1] = 1;
13          #100 AA[2] = 1;
14          #100 SS[0] = 1;
15          #100 AA[3] = 1;
16          #100 AA[3] = 0;
17          #100 $stop;
18          end
19  endmodule
```

Testbench Simulation Results :



Part 2

4-Bit Barrel Shifter :



Shift Value N[1:0]	Shifted Output (SHO)			
00	A_3	A_2	A_1	A_0
01	A_0	A_3	A_2	A_1
10	A_1	A_0	A_3	A_2
11	A_2	A_1	A_0	A_3

Calculating Delay Values :

Same as The 4to1 MUX Delay Value = **37.5**

(because each 4to1 MUX is on a separate path)

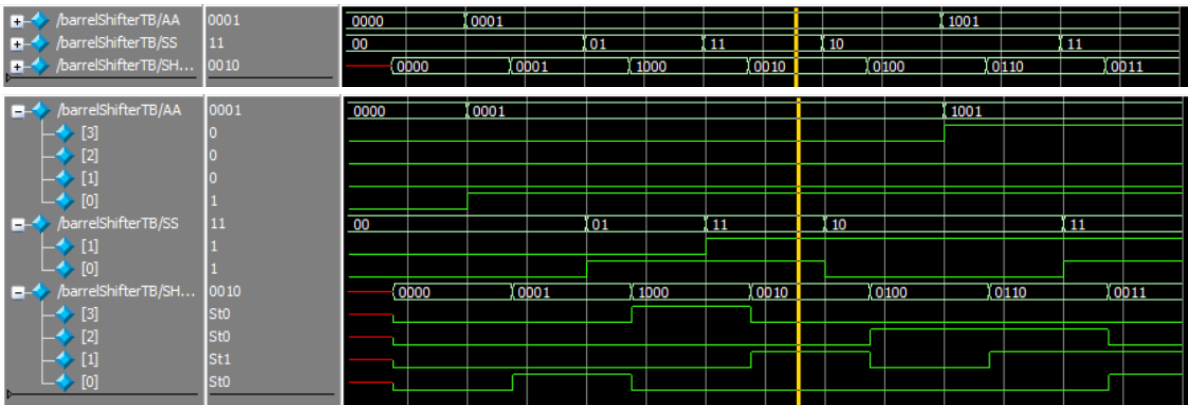
Main Code :

```
1   `timescale 1ns/1ns
2   module barrelShifter (input [3:0] A, input [1:0] S, output [3:0] SHO);
3
4           wire [3:0] J1,J2,J3,J4;
5           assign J1 = {A[2], A[1], A[0], A[3]};
6           mux4to1 M1(J1, S, SHO[3]);
7           assign J2 = {A[1], A[0], A[3], A[2]};
8           mux4to1 M2(J2, S, SHO[2]);
9           assign J3 = {A[0], A[3], A[2], A[1]};
10          mux4to1 M3(J3, S, SHO[1]);
11          assign J4 = {A[3], A[2], A[1], A[0]};
12          mux4to1 M4(J4, S, SHO[0]);
13
14  endmodule
```

Testbench Code :

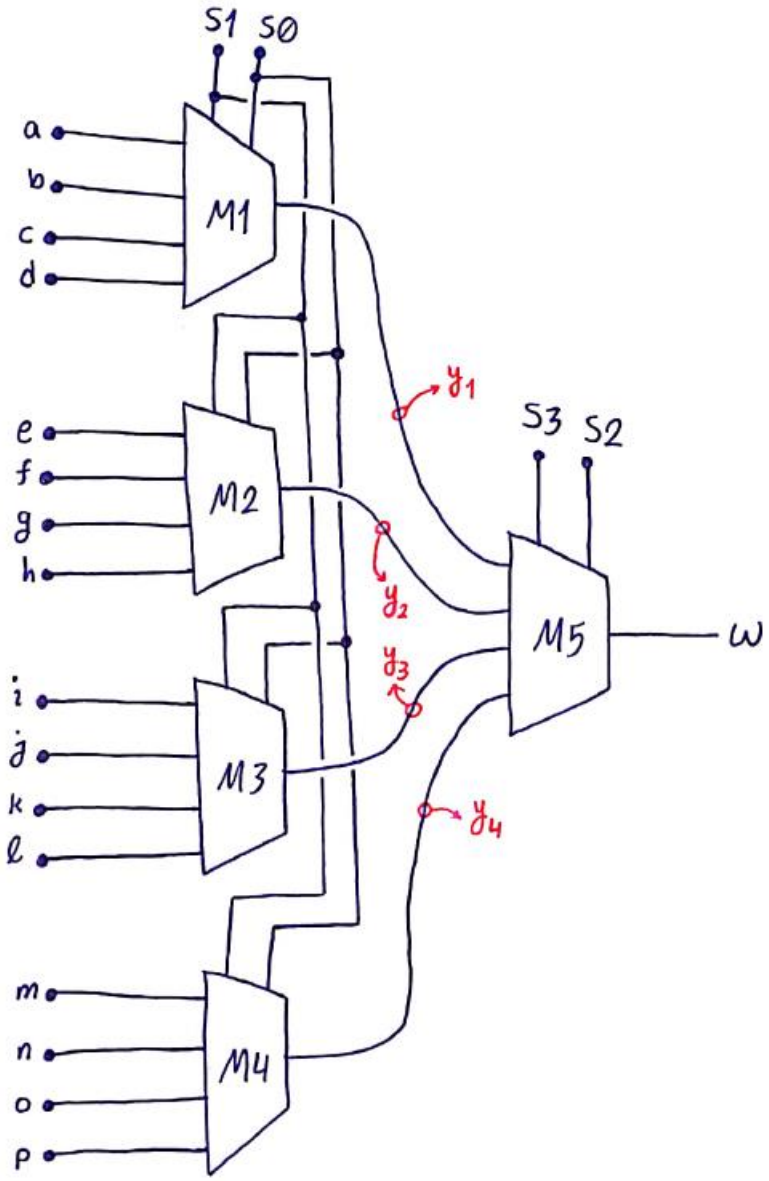
```
1   `timescale 1ns/1ns
2   module barrelShifterTB ();
3           reg [3:0] AA = 0;
4           reg [1:0] SS = 0;
5           wire [3:0] SHOWW;
6           barrelShifter B1(AA, SS, SHOWW);
7           initial begin
8           #100 AA[0] = 1;
9           #100 SS[0] = 1;
10          #100 SS[1] = 1;
11          #100 SS[0] = 0;
12          #100 AA[3] = 1;
13          #100 SS[0] = 1;
14          #100 $stop;
15          end
16  endmodule
```

Testbench Simulation Results :



Part 3

16TO1 MULTIPLEXER :



CALCULATING DELAY VALUES :

4TO1 MUX #(37.5)

➔➔➔ 16TO1 MULTIPLEXER DELAY VALUE = 37.5 + 37.5 = **75**

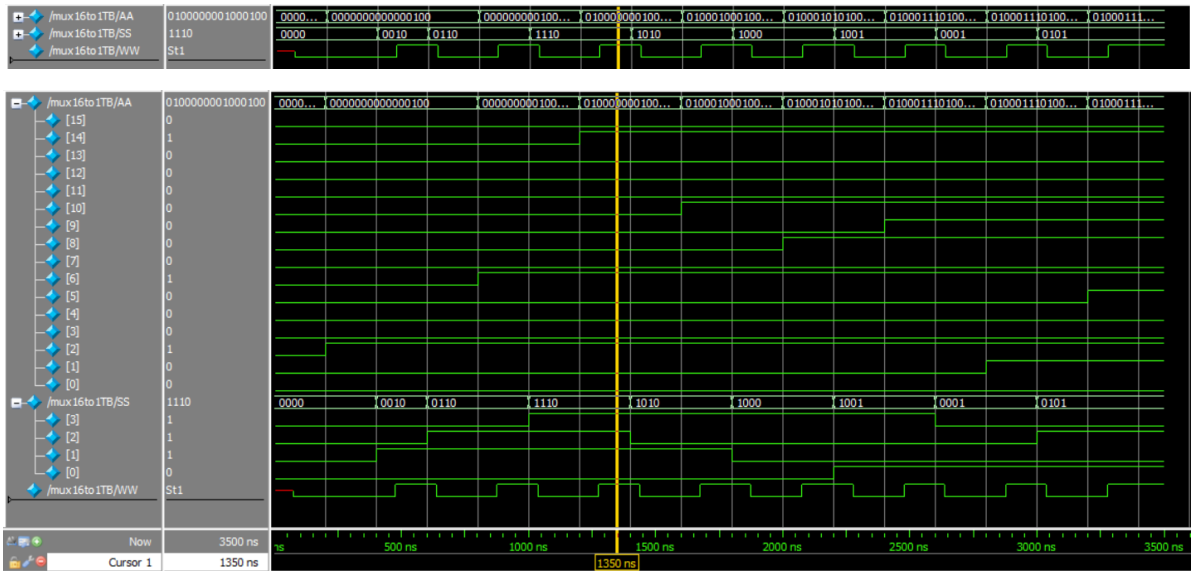
MAIN CODE :

```
1   `timescale 1ns/1ns
2   module mux16to1 (input [15:0] A, input [3:0] S, output w);
3
4           wire y1,y2,y3,y4,y5,y6;
5           mux4to1 M1(A[15:12],S[1:0],y1);
6           mux4to1 M2(A[11:8],S[1:0],y2);
7           mux4to1 M3(A[7:4],S[1:0],y3);
8           mux4to1 M4(A[3:0],S[1:0],y4);
9           wire [3:0] J;
10          assign J = {y1,y2,y3,y4};
11          mux4to1 M5(J,S[3:2],w);
12
13  endmodule
```

TESTBENCH CODE :

```
1   `timescale 1ns/1ns
2   module mux16to1TB ();
3           reg [15:0] AA = 0;
4           reg [3:0] SS = 0;
5           wire WW;
6           mux16to1 M16(AA, SS, WW);
7           initial begin
8           #200 AA[2] = 1;
9           #200 SS[1] = 1;
10          #200 SS[2] = 1;
11          #200 AA[6] = 1;
12          #200 SS[3] = 1;
13          #200 AA[14] = 1;
14          #200 SS[2] = 0;
15          #200 AA[10] = 1;
16          #200 SS[1] = 0;
17          #200 AA[8] = 1;
18          #200 SS[0] = 1;
19          #200 AA[9] = 1;
20          #200 SS[3] = 0;
21          #200 AA[1] = 1;
22          #200 SS[2] = 1;
23          #200 AA[5] = 1;
24          #300 $stop;
25          end
26  endmodule
```

TESTBENCH SIMULATION RESULTS :



Part 4

16-BIT BARREL SHIFTER:

CALCULATING DELAY VALUES :

SAME AS THE 16TO1 MUX DELAY VALUE = **75**

(BECAUSE EACH PATH HAS ONLY ONE 16TO1 MUX)

MAIN CODE :

```
`timescale 1ns/1ns
module barrelShifter16Bit (input [15:0] A, input [3:0] S, output [15:0] SHO);

        wire [15:0] J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,J14,J15,J16;
        assign J1 = {A[14],  A[13], A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15]};
        assign J2 = {A[13],  A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14]};
        assign J3 = {A[12],  A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13]};
        assign J4 = {A[11],  A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12]};
        assign J5 = {A[10],  A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11]};
        assign J6 = {A[9],   A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10]};
        assign J7 = {A[8],   A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9]};
        assign J8 = {A[7],   A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8]};
        assign J9 = {A[6],   A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8],  A[7]};
        assign J10 = {A[5],  A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6]};
        assign J11 = {A[4],  A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5]};
        assign J12 = {A[3],  A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4]};
        assign J13 = {A[2],  A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3]};
        assign J14 = {A[1],  A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2]};
        assign J15 = {A[0],  A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1]};
        assign J16 = {A[15], A[14], A[13], A[12], A[11], A[10], A[9],  A[8],  A[7],  A[6],  A[5],  A[4],  A[3],  A[2],  A[1],  A[0]};
        mux16to1 M1(J1, S, SHO[15]);
        mux16to1 M2(J2, S, SHO[14]);
        mux16to1 M3(J3, S, SHO[13]);
        mux16to1 M4(J4, S, SHO[12]);
        mux16to1 M5(J5, S, SHO[11]);
        mux16to1 M6(J6, S, SHO[10]);
        mux16to1 M7(J7, S, SHO[9]);
        mux16to1 M8(J8, S, SHO[8]);
        mux16to1 M9(J9, S, SHO[7]);
        mux16to1 M10(J10, S, SHO[6]);
        mux16to1 M11(J11, S, SHO[5]);
        mux16to1 M12(J12, S, SHO[4]);
        mux16to1 M13(J13, S, SHO[3]);
        mux16to1 M14(J14, S, SHO[2]);
        mux16to1 M15(J15, S, SHO[1]);
        mux16to1 M16(J16, S, SHO[0]);

endmodule
```

TESTBENCH CODE :

```
1   `timescale 1ns/1ns
2   module barrelShifter16BitTB ();
3           reg [15:0] AA = 0;
4           reg [3:0] SS = 0;
5           wire [15:0] WW;
6           barrelShifter16Bit BS16(AA, SS, WW);
7           initial begin
8           #200 AA[2] = 1;
9           #200 SS[1] = 1;
10          #200 SS[2] = 1;
11          #200 AA[6] = 1;
12          #200 SS[3] = 1;
13          #200 AA[14] = 1;
14          #200 SS[2] = 0;
15          #200 AA[10] = 1;
16          #200 SS[1] = 0;
17          #200 AA[8] = 1;
18          #200 SS[0] = 1;
19          #200 AA[9] = 1;
20          #200 SS[3] = 0;
21          #200 AA[1] = 1;
22          #200 SS[2] = 1;
23          #200 AA[5] = 1;
24          #200 AA[0] = 1;
25          #200 SS[3] = 1;
26          #200 AA[1] = 1;
27          #200 SS[0] = 0;
28          #200 SS[1] = 1;
29          #200 AA[2] = 0;
30          #200 SS[0] = 1;
31          #200 AA[3] = 1;
32          #200 AA[8] = 0;
33          #200 AA[9] = 1;
34          #200 SS[2] = 0;
35          #200 SS[1] = 0;
36          #200 SS[0] = 0;
37          #200 AA[3] = 1;
38          #200 SS[0] = 1;
39          #300 $stop;
40          end
41  endmodule
```

TESTBENCH SIMULATION RESULTS :