UNIVERSITY OF TEHRAN

Report for Computer Assignment 2

Instructor : Dr. Navabi

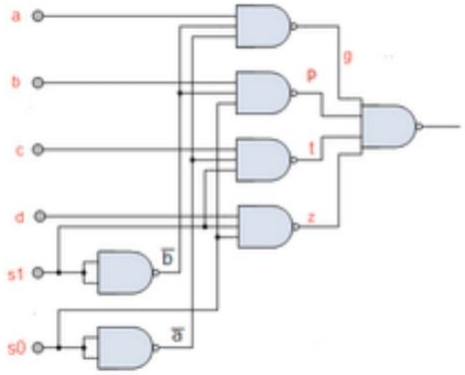
Digital Logic Design, ECE 367 / Digital Systems I, ECE 894

Danial Saeedi

Problem 1

From problem 1 in Computer Assignment 1, we know that the delays for 2, 3 and 4 input NAND gates are #(10,8), #(15,12) and #(20,16).

For simplicity, we assign the average of To0 and To1 delays. So the average delays for 2, 3 and 4 inputs are #9, #13 and #18.



The worst-case delay of this circuit is :

9+13+18 = 40ns

Verilog Code

4 to 1 MUX implementation in Computer Assignment 1 :

```
`timescale 1ns/1ns
module MyMux(
    input s0,s1,a,b,c,d,
    output y
);
    wire s0_not,s1_not,g,p,t,z;
    TwoInputNAND nand_1(s0,s0,s0_not);
    TwoInputNAND nand_2(s1,s1,s1_not);
    ThreeInputsNand nand_3(a,s1_not,s0_not,g);
    ThreeInputsNand nand_4(b,s1_not,s0,p);
    ThreeInputsNand nand_5(c,s1,s0_not,t);
    ThreeInputsNand nand_6(d,s1,s0,z);
    FourInputsNand nand_7(g,p,t,z,y);
endmodule
```

MUX-CA1.v

```
problem1.v   problem1-2.v
`timescale 1ns/1ns
module MUX4to1(input a,b,c,d,s1,s0,output w);
    assign #40 w = (~s1&~s0&a)|(~s1&s0&b)|(s1&~s0&c)|(s1&s0&d);
endmodule
```

Problem1.v

```
`timescale 1ns/1ns
module MUX4to1V2(input [0:3] data,input [1:0] s,output w);
    assign #40 w = (s == 2'b00) ? data[0] :
                   (s == 2'b01) ? data[1] :
                   (s == 2'b10) ? data[2] :
                   (s == 2'b11) ? data[3] :
                   1'bx;
endmodule
```

Problem1-2.v

Worst-case Delay

Every possible input has 40ns delay because we are using assign statement. So the worst-case delay is **40ns** in this problem.

Testbench

The ww1 and ww3 wires are the output of 3 different implementation of MUX.

```
`timescale 1ns/1ns
module MuxTB();
    wire ww;
    reg [1:0] ss;
    reg [0:3] data;
    MUX4to1V2 CUT2(data,ss,ww);
    initial begin
        #11 ss = 2'b00; data = 4'b0001;
        #61 ss = 2'b00; data = 4'b1000;
        #61 ss = 2'b10; data = 4'b0100;
        #61 ss = 2'b10; data = 4'b0010;
        #61 ss = 2'b01; data = 4'b0010;
        #61 ss = 2'b01; data = 4'b0110;
        #61 ss = 2'b11; data = 4'b0110;
        #61 ss = 2'b11; data = 4'b0101;
        #61 $stop;
    end
endmodule
```

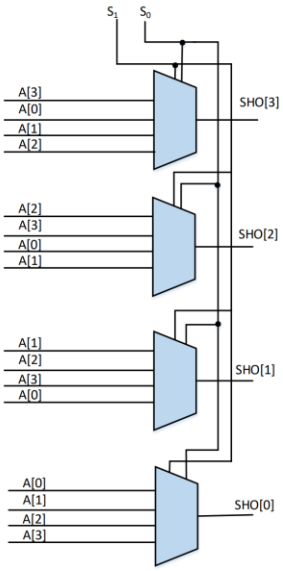
TB-MUX.v

Simulation Result

The worst-case delay is 40ns.

Problem 2

Circuit Diagram



Verilog Code

```verilog
`timescale 1ns/1ns
module FourBitBarrelShifter(input [3:0] A,input [1:0] N,output [3:0] SHO);
    reg [0:3] J,K,L,M;
    wire o0,o1,o2,o3;

    assign J = {A[3],A[0],A[1],A[2]};
    MUX4to1V2 CUT1(J,N,o3);

    assign K = {A[2],A[3],A[0],A[1]};
    MUX4to1V2 CUT2(K,N,o2);

    assign L = {A[1],A[2],A[3],A[0]};
    MUX4to1V2 CUT3(L,N,o1);

    assign M = {A[0],A[1],A[2],A[3]};
    MUX4to1V2 CUT4(M,N,o0);

    assign SHO = {o3,o2,o1,o0};

endmodule
```

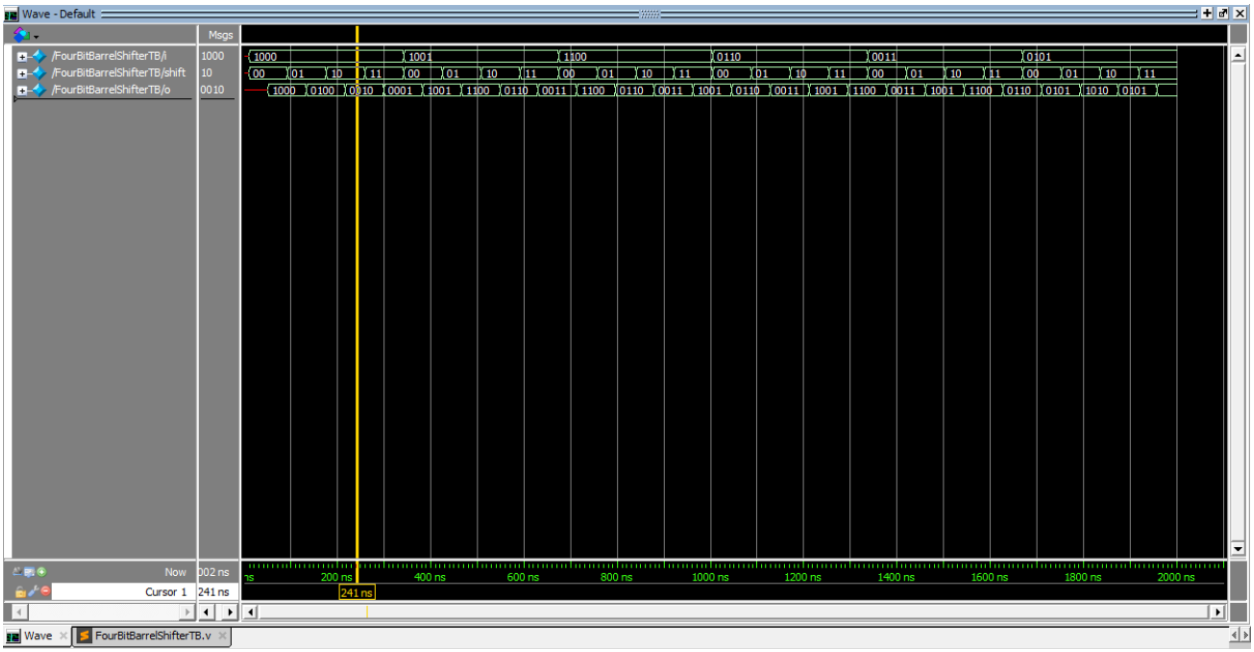
FourBitBarrelShifter.v

Testbench

i,shift and o represent input, shift value and output.

```
`timescale 1ns/1ns
module FourBitBarrelShifterTB();
    reg [3:0] i;
    reg [1:0] shift;
    wire [3:0] o;
    FourBitBarrelShifter CUT1(i,shift,o);
    initial begin
        #10 i = 4'b1000; shift = 2'b00;
        #83 i = 4'b1000; shift = 2'b01;
        #83 i = 4'b1000; shift = 2'b10;
        #83 i = 4'b1000; shift = 2'b11;

        #83 i = 4'b1001; shift = 2'b00;
        #83 i = 4'b1001; shift = 2'b01;
        #83 i = 4'b1001; shift = 2'b10;
        #83 i = 4'b1001; shift = 2'b11;

        #83 i = 4'b1100; shift = 2'b00;
        #83 i = 4'b1100; shift = 2'b01;
        #83 i = 4'b1100; shift = 2'b10;
        #83 i = 4'b1100; shift = 2'b11;

        #83 i = 4'b0110; shift = 2'b00;
        #83 i = 4'b0110; shift = 2'b01;
        #83 i = 4'b0110; shift = 2'b10;
        #83 i = 4'b0110; shift = 2'b11;

        #83 i = 4'b0011; shift = 2'b00;
        #83 i = 4'b0011; shift = 2'b01;
        #83 i = 4'b0011; shift = 2'b10;
        #83 i = 4'b0011; shift = 2'b11;

        #83 i = 4'b0101; shift = 2'b00;
        #83 i = 4'b0101; shift = 2'b01;
        #83 i = 4'b0101; shift = 2'b10;
        #83 i = 4'b0101; shift = 2'b11;
        #83 $stop;
    end
endmodule
```

Simulation Result



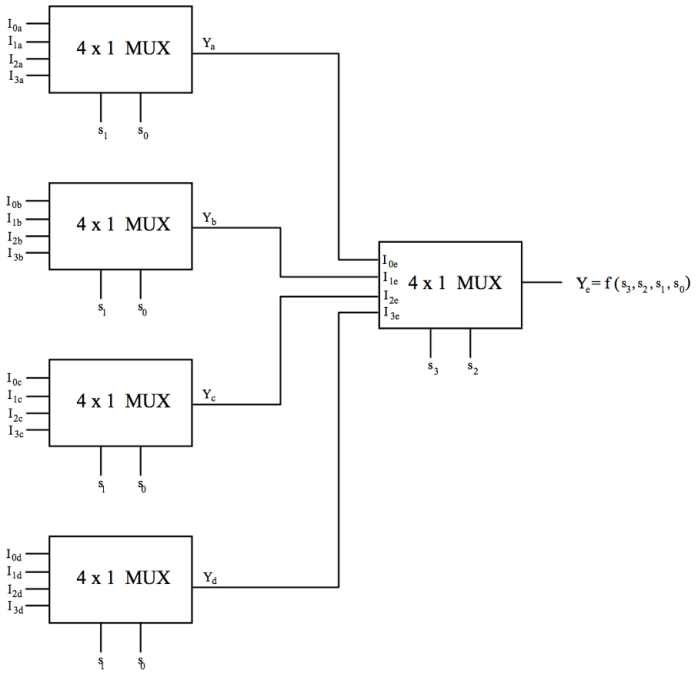
Worst-case Delay

Each 4 to 1 MUX has 40ns delay. So we expect worst-case delay would be **around 40ns**.

The waveform proves that the worst-case delay is **40ns**.

Problem 3

Circuit Diagram



Verilog Code

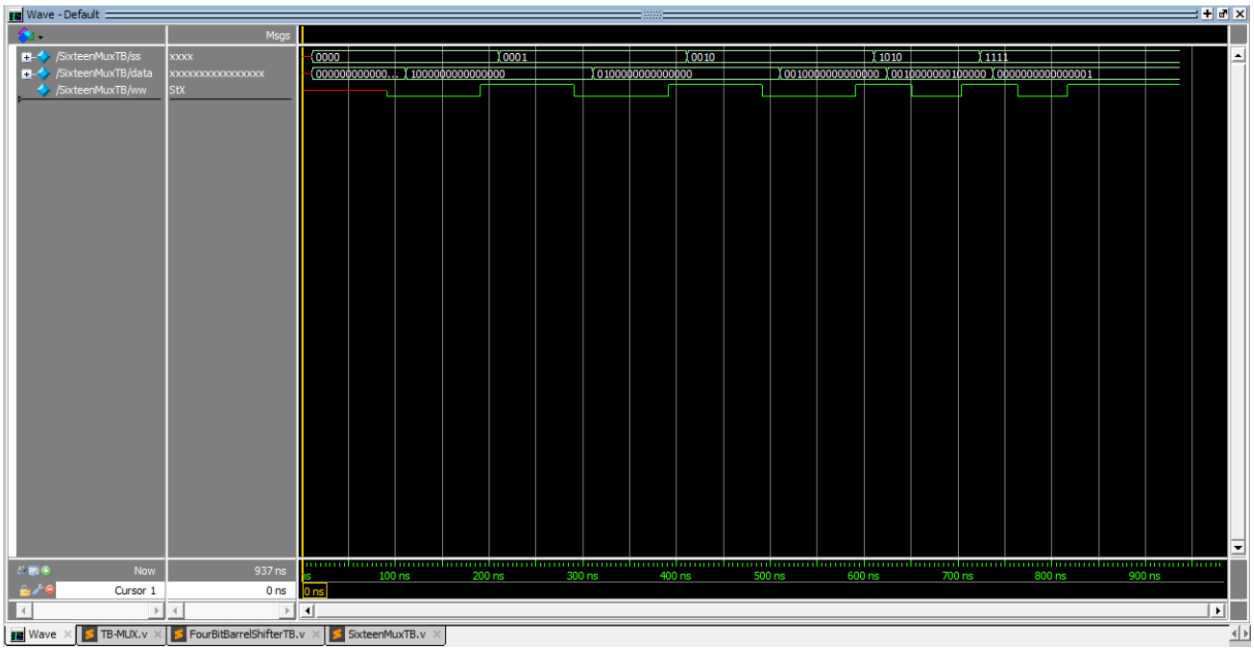
```verilog
`timescale 1ns/1ns
module SixteenMux(input [0:15] data,input [3:0] s,output w);
    wire g,p,t,z;
    MUX4to1V2 CUT1({data[0],data[1],data[2],data[3]},{s[1],s[0]},g);
    MUX4to1V2 CUT2({data[4],data[5],data[6],data[7]},{s[1],s[0]},p);
    MUX4to1V2 CUT3({data[8],data[9],data[10],data[11]},{s[1],s[0]},t);
    MUX4to1V2 CUT4({data[12],data[13],data[14],data[15]},{s[1],s[0]},z);
    MUX4to1V2 CUT5({g,p,t,z},{s[3],s[2]},w);
endmodule
```

SixteenMux.v

Testbench

```
1  `timescale 1ns/1ns
2  module SixteenMuxTB();
3      wire ww;
4      reg [3:0] ss;
5      reg [0:15] data;
6      SixteenMux CUT1(data,ss,ww);
7      initial begin
8          #11 ss = 4'b0; data = 16'b0;
9          #100 ss = 4'b0; data = 16'b1000000000000000;
10         #100 ss = 4'b0001;
11         #100 data = 16'b0100000000000000;
12         #100 ss = 4'b0010;
13         #100 data = 16'b0010000000000000;
14         #100 ss = 4'b1010;
15         #13 data = 16'b0001000000100000;
16         #100 ss = 4'b1111;
17         #13 data = 16'b0000000000000001;
18         #200 $stop;
19     end
20 endmodule
```

Simulation Result



SixteenMuxTB.v

Worst-case Delay

Every path goes through 2 Muxes. So the worst-case delay is :

40 + 40 = 80ns

We can see from waveform, the worst-case delay is also 80ns.

Problem 4

16-bit Barrel Shifter

We need sixteen pieces of 16 to 1 MUX in parallel in order to generate 16-bit barrel shifter. We can design the circuit according to this table:

s3,s2,s1,s0	input
0000	A15,A14,A13,A12,A11,A10,A9,A8,A7, A6, A5,A4,A3,A2,A1,A0
0001	A0,A15,A14,A13,A12,A11,A10,A9,A8,A7, A6, A5,A4,A3,A2,A1
0010	A1,A0,A15,A14,A13, A12,A11, A10, A9, A8,A7,A6,A5,A4,A3,A2
0011	A2,A1,A0,A15 ,A14,A13, A12,A11, A10, A9, A8,A7,A6,A5,A4,A3
0100	A3,A2,A1,A0,A15,A14,A13, A12,A11, A10, A9, A8,A7,A6,A5,A4
0101	A4,A3,A2,A1,A0,A15 ,A14,A13, A12,A11, A10, A9, A8,A7,A6,A5
0110	A5,A4,A3,A2,A1,A0,A15,A14,A13, A12,A11, A10, A9, A8,A7,A6
0111	A6,A5,A4,A3,A2,A1,A0,A15, A14,A13, A12,A11, A10, A9, A8,A7
1000	A7,A6,A5,A4,A3,A2,A1,A0,A15 , A14,A13, A12,A11, A10, A9, A8
1001	A8,A7,A6,A5,A4,A3,A2,A1,A0,A15 , A14,A13, A12,A11, A10, A9
1010	A9,A8,A7,A6,A5,A4,A3,A2,A1,A0,A15 ,A14,A13, A12,A11, A10
1011	A10,A9,A8,A7,A6,A5,A4,A3,A2,A1, A0,A15 ,A14,A13, A12,A11
1100	A11,A10,A9,A8,A7,A6,A5,A4,A3,A2,A1,A0,A15 ,A14,A13, A12
1101	A12,A11,A10,A9,A8,A7,A6,A5,A4,A3,A2,A1,A0,A15, A13
1110	A13,A12,A11,A10,A9,A8,A7,A6,A5,A4,A3,A2,A1,A0, A15 , A14
1111	A14,A13,A12,A11,A10,A9,A8,A7,A6,A5,A4,A3,A2,A1,A0, A15

Verilog Code

```
`timescale 1ns/1ns
module SixteenBarrelShifter(input [15:0] A,input [3:0] N,output [15:0] w);
    SixteenMux CUT1({A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14]},N,w[15]);
    SixteenMux CUT2({A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13]},N,w[14]);
    SixteenMux CUT3({A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12]},N,w[13]);
    SixteenMux CUT4({A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11]},N,w[12]);
    SixteenMux CUT5({A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10]},N,w[11]);
    SixteenMux CUT1({A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9]},N,w[10]);
    SixteenMux CUT1({A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8]},N,w[9]);
    SixteenMux CUT1({A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7]},N,w[8]);
    SixteenMux CUT1({A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5],A[6]},N,w[7]);
    SixteenMux CUT1({A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4],A[5]},N,w[6]);
    SixteenMux CUT1({A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3],A[4]},N,w[5]);
    SixteenMux CUT1({A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2],A[3]},N,w[4]);
    SixteenMux CUT1({A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1],A[2]},N,w[3]);
    SixteenMux CUT1({A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0],A[1]},N,w[2]);
    SixteenMux CUT1({A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15],A[0]},N,w[1]);
    SixteenMux CUT1({A[0],A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11],A[12],A[13],A[14],A[15]},N,w[0]);
endmodule
```

Testbench

```
`timescale 1ns/1ns
module SixteenBShifterTB();
    reg [15:0] i;
    reg [3:0] shift;
    wire [15:0] o;
    SixteenBarrelShifter CUT1(i,shift,o);
    initial begin
        #11 i = 16'b1000000000000000;
        #101 shift = 4'b0000;
        #101 shift = 4'b0001;
        #101 shift = 4'b0010;
        #101 shift = 4'b0011;
        #101 shift = 4'b0100;
        #101 shift = 4'b0101;
        #101 shift = 4'b0110;
        #101 shift = 4'b0111;
        #101 shift = 4'b1000;
        #101 shift = 4'b1001;
        #101 shift = 4'b1010;
        #101 shift = 4'b1011;
        #101 shift = 4'b1100;
        #101 shift = 4'b1101;
        #101 shift = 4'b1110;
        #101 shift = 4'b1111;

        #101 i = 16'b1000000001000000;
        #101 shift = 4'b0000;
        #101 shift = 4'b0001;
        #101 shift = 4'b0010;
        #101 shift = 4'b0011;
        #101 shift = 4'b0100;
        #101 shift = 4'b0101;
        #101 shift = 4'b0110;
        #101 shift = 4'b0111;
        #101 shift = 4'b1000;
        #101 shift = 4'b1001;
        #101 shift = 4'b1010;
        #101 shift = 4'b1011;
        #101 shift = 4'b1100;
        #101 shift = 4'b1101;
        #101 shift = 4'b1110;
        #101 shift = 4'b1111;
        #101 $stop;
    end
endmodule
```

Simulation Result

Worst-case Delay

The design of 16-bit Barrel Shifter is very similar to 4-bit Barrel Shifter.

Every path from goes through one 16 to 1Mux. And from problem 3, we know that the delay of 16 to 1 Mux is **80ns**.

We can see from waveform, the worst-case delay is also **80ns**