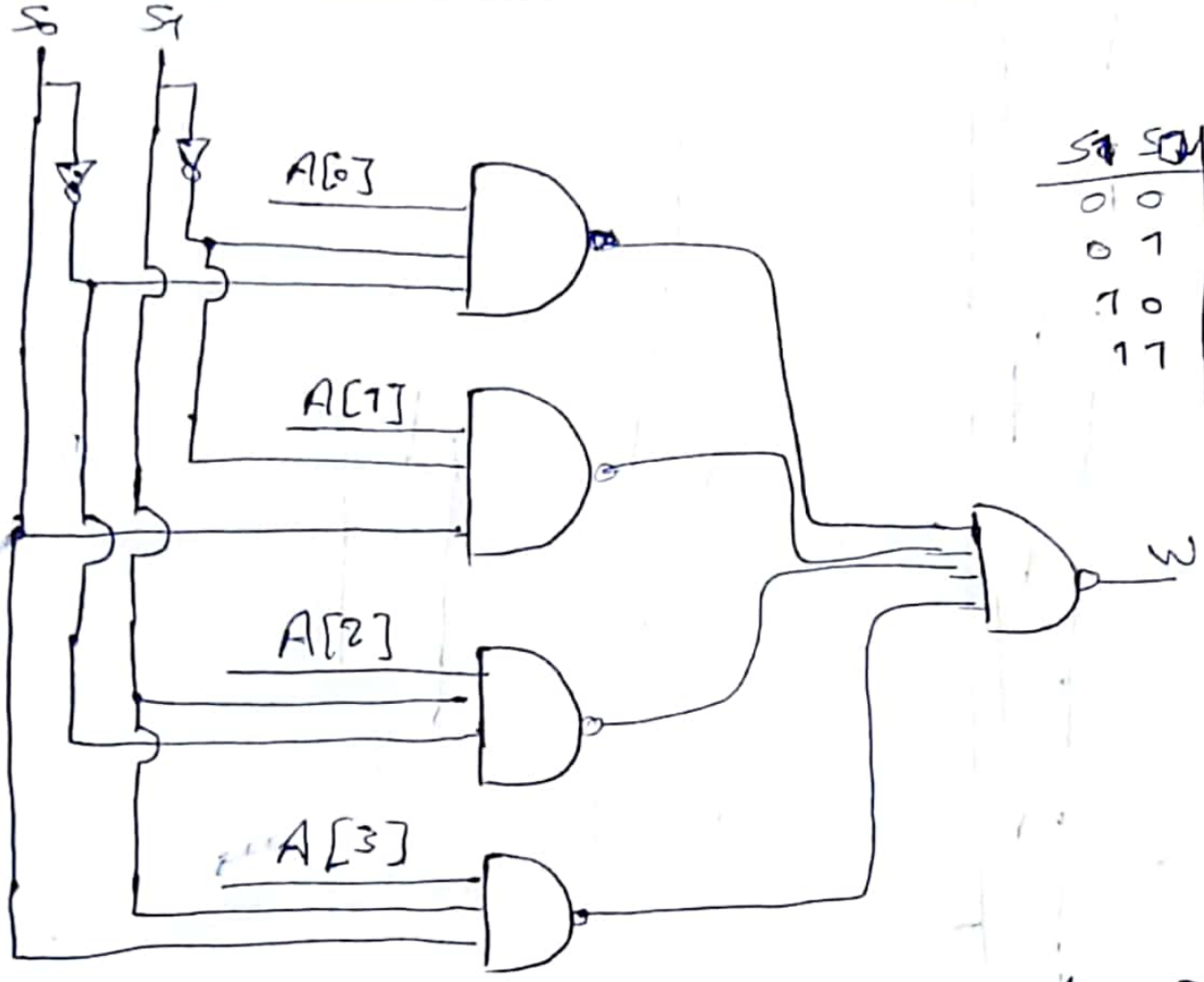
Digital Systems

CA2 - Report

Mahdi Bohloul

810198365

(1)



S1	S0	W
0	0	A[0]
0	1	A[1]
1	0	A[2]
1	1	A[3]

nmos # (3,4,5)

Pmos # (5,6,7)

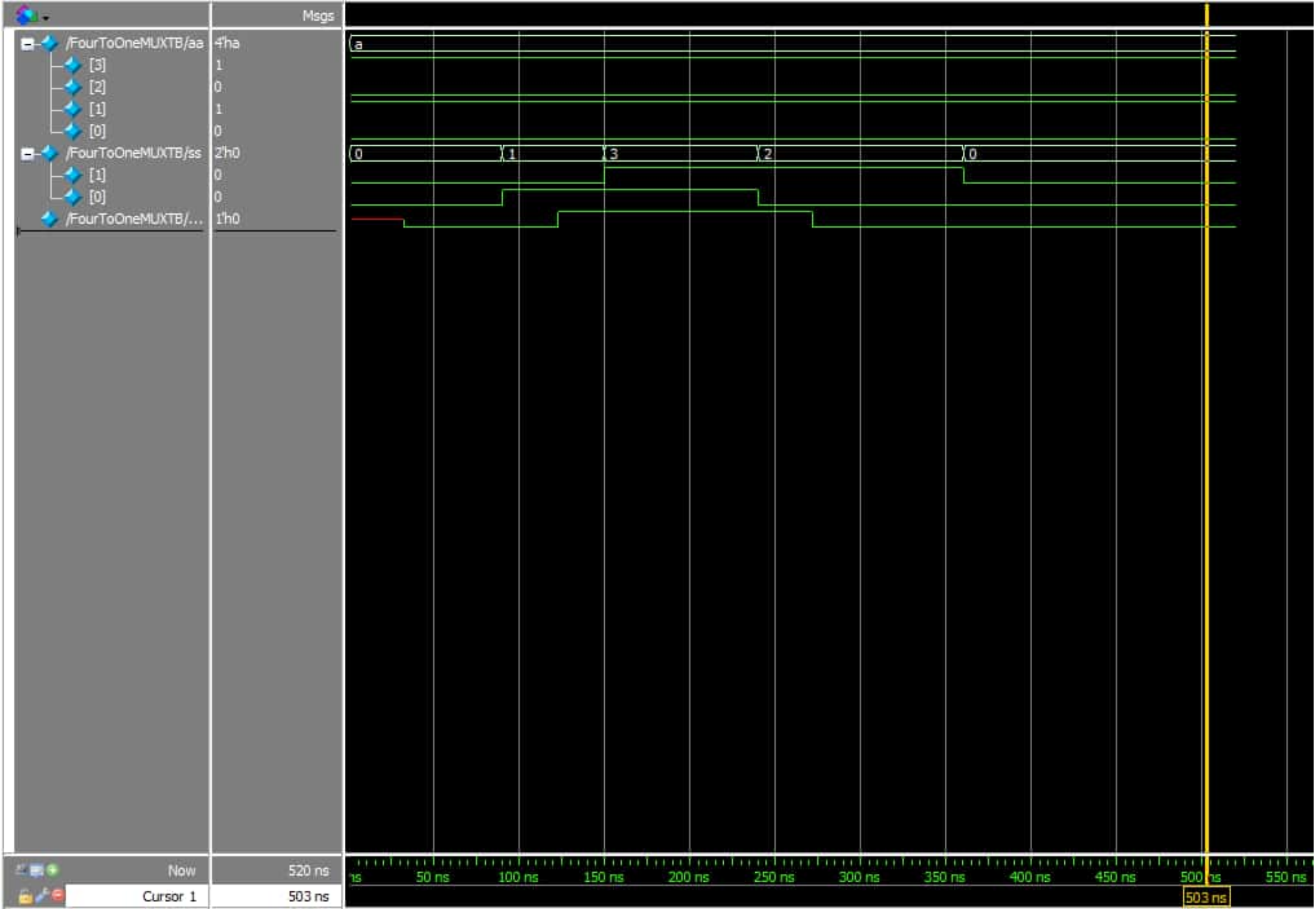
worst-case delay = 33ns

4 To 1 MUX

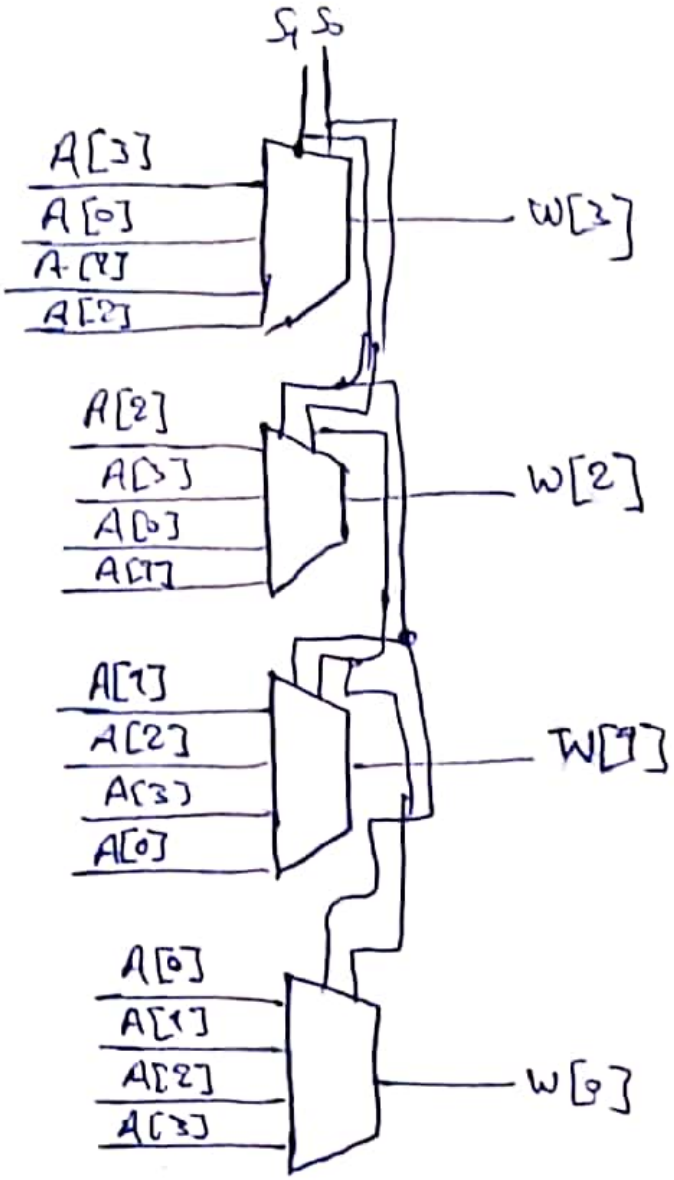
```
`timescale 1ns/1ns


module FourToOneMUX (input [3:0]A, [1:0]S, output w);
    assign #32 w = A[S];
endmodule
```

4 To 1 MUX TB

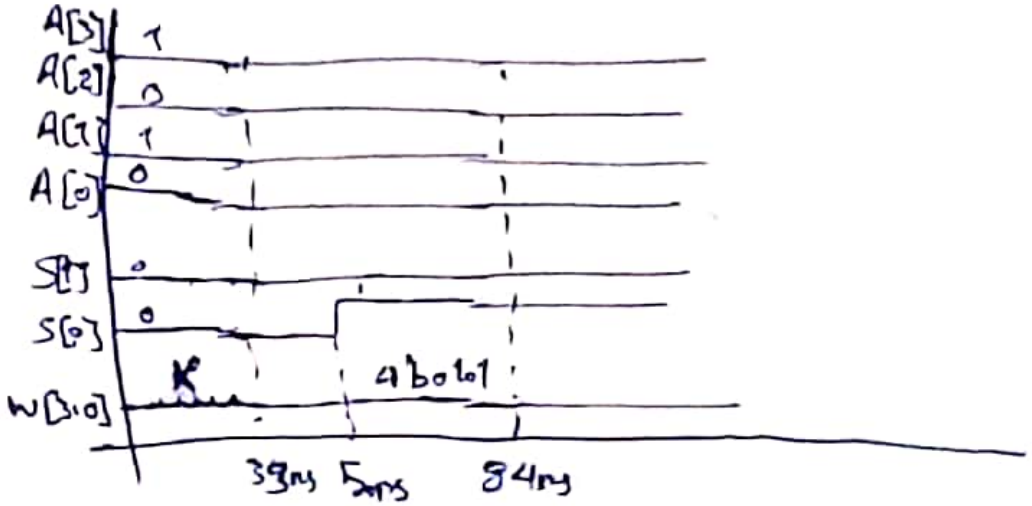
```
`include "FourToOneMUX.v"
`timescale 1ns/1ns

module FourToOneMUXTB ();
    reg [3:0] aa = 4'b1010;
    reg [1:0] ss = 2'b00;
    wire ww;
    FourToOneMUX mux(aa, ss, ww);
    initial begin
        #50
        #40 ss = 2'b01;
        #60 ss = 2'b11;
        #90 ss = 2'b10;
        #120 ss = 2'b00;
        #160 $stop;
    end
endmodule
```



(2)



The worst case of this circuit is equal to the worst case of one of the MUX, which is equal to 33ns

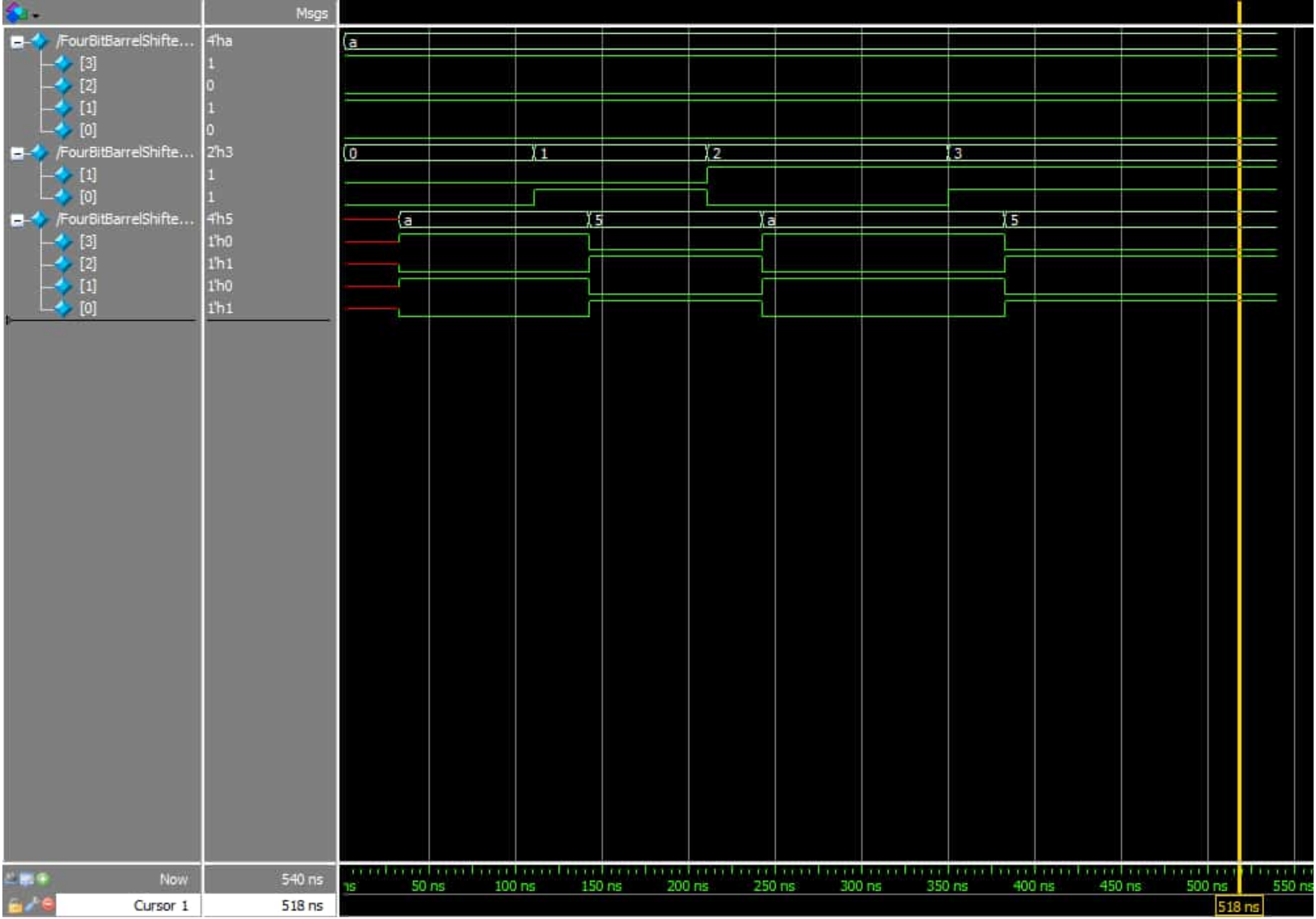


4 Bit Barrel Shifter TB

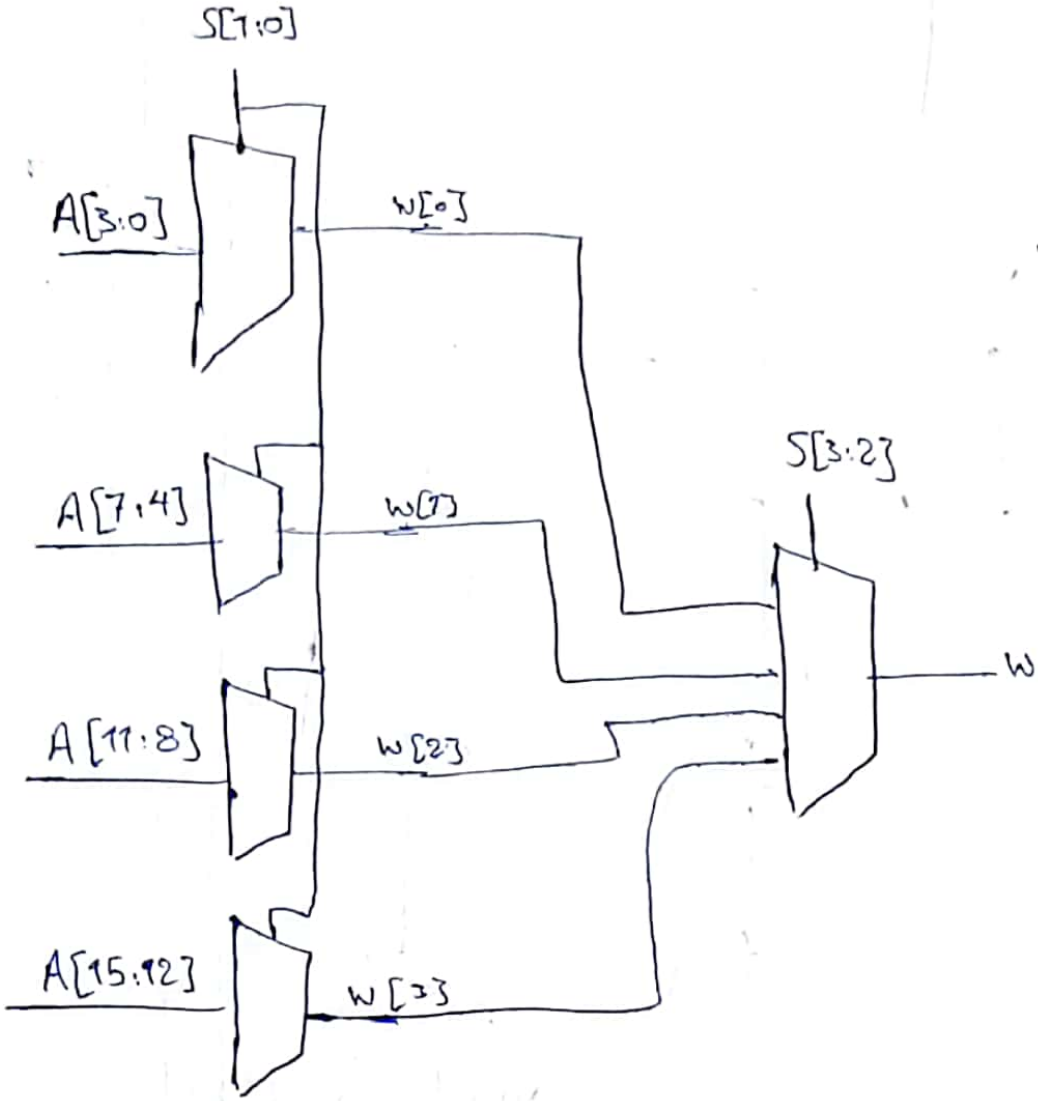
```
`include "FourBitBarrelShifter.v"
`timescale 1ns/1ns


module FourBitBarrelShifterTB ();
    reg [3:0] aa = 4'b1010;
    reg [1:0] nn = 2'b00;
    wire [3:0] ww;
    FourBitBarrelShifter barrel_shifter(aa, nn, ww);
    initial begin
        #50
        #60 nn = 2'b01;
        #100 nn = 2'b10;
        #140 nn = 2'b11;
        #190 $stop;
    end
endmodule
```

16 Bit Barrel Shifter TB

```
`include "SixteenBitBarrelShifter.v"
`timescale 1ns/1ns

module SixteenBitBarrelShifterTB ();
    reg [15:0] A = 16'hd;
    reg [3:0] N = 4'b0;
    wire [15:0] ww;

    SixteenBitBarrelShifter barrel_shifter(A, N, ww);
    initial begin
        #60
        repeat(5) #60 N = $random;
        repeat(15) #60 A = A + 1;
        #100 $stop;
    end

endmodule
```



(3)



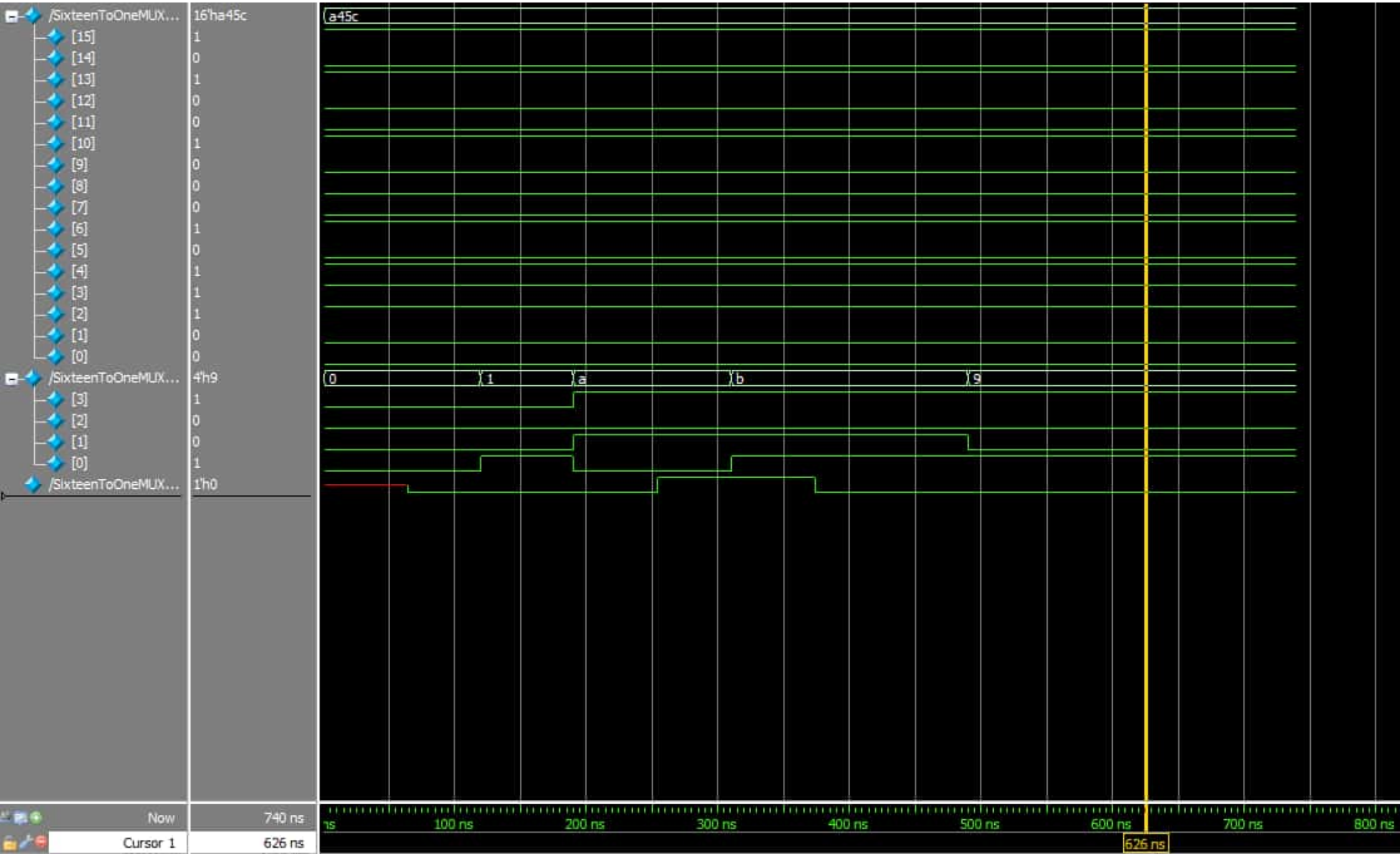
The worst case of this circuit is equal to the worst case of two of the multiplexers in series, which is equal to 66ns

16 To 1 MUX

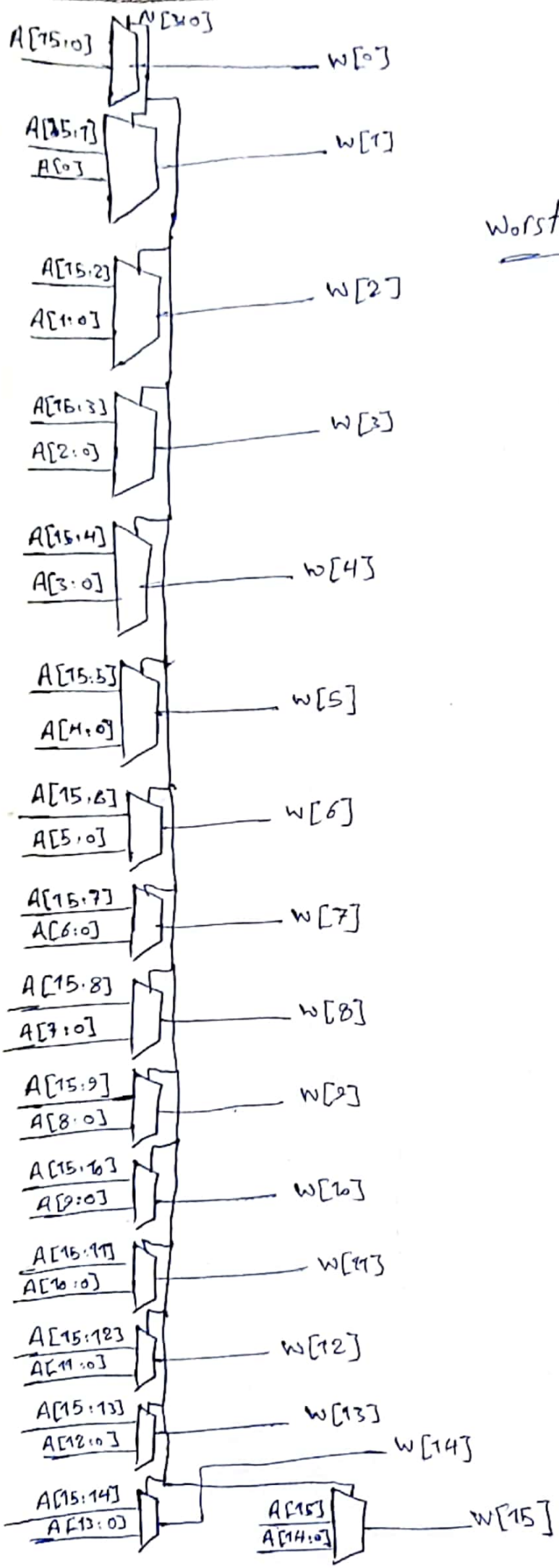
```
`include "FourToOneMUX.v"
`timescale 1ns/1ns

module SixteenToOneMUX(input [15:0]A, [3:0]S, output w);
    wire [3:0] W;
    FourToOneMUX mux1(A[3:0], S[1:0], W[0]);
    FourToOneMUX mux2(A[7:4], S[1:0], W[1]);
    FourToOneMUX mux3(A[11:8], S[1:0], W[2]);
    FourToOneMUX mux4(A[15:12], S[1:0], W[3]);
    FourToOneMUX res(W, S[3:2], w);
endmodule
```

16 To 1 MUX TB

```
`include "SixteenToOneMUX.v"
`timescale 1ns/1ns

module SixteenToOneMUXTB ();
    reg [15:0] aa = 16'ha45c;
    reg [3:0] ss = 4'b0;
    wire ww;
    SixteenToOneMUX mux(aa, ss, ww);
    initial begin
        #70
        #50 ss = 4'b0001;
        #70 ss = 4'b1010;
        #120 ss = 4'b1011;
        #180 ss = 4'b1001;
        #250 $stop;
    end

endmodule
```



4)



Worst case: 66ns

16 Bit Barrel Shifter

```
`include "SixteenToOneMUX.v"
`timescale 1ns/1ns


module SixteenBitBarrelShifter (input [15:0]A, input [3:0]N, output [15:0]W);
    SixteenToOneMUX mux1(A, N, W[0]);
    SixteenToOneMUX mux2({A[0], A[15:1]}, N, W[1]);
    SixteenToOneMUX mux3({A[1:0], A[15:2]}, N, W[2]);
    SixteenToOneMUX mux4({A[2:0], A[15:3]}, N, W[3]);
    SixteenToOneMUX mux5({A[3:0], A[15:4]}, N, W[4]);
    SixteenToOneMUX mux6({A[4:0], A[15:5]}, N, W[5]);
    SixteenToOneMUX mux7({A[5:0], A[15:6]}, N, W[6]);
    SixteenToOneMUX mux8({A[6:0], A[15:7]}, N, W[7]);
    SixteenToOneMUX mux9({A[7:0], A[15:8]}, N, W[8]);
    SixteenToOneMUX mux10({A[8:0], A[15:9]}, N, W[9]);
    SixteenToOneMUX mux11({A[9:0], A[15:10]}, N, W[10]);
    SixteenToOneMUX mux12({A[10:0], A[15:11]}, N, W[11]);
    SixteenToOneMUX mux13({A[11:0], A[15:12]}, N, W[12]);
    SixteenToOneMUX mux14({A[12:0], A[15:13]}, N, W[13]);
    SixteenToOneMUX mux15({A[13:0], A[15:14]}, N, W[14]);
    SixteenToOneMUX mux16({A[14:0], A[15]}, N, W[15]);
endmodule
```

16 Bit Barrel Shifter TB

```
`include "SixteenBitBarrelShifter.v"
`timescale 1ns/1ns

module SixteenBitBarrelShifterTB ();
    reg [15:0] A = 16'hd;
    reg [3:0] N = 4'b0;
    wire [15:0] ww;

    SixteenBitBarrelShifter barrel_shifter(A, N, ww);
    initial begin
        #60
        repeat(5) #60 N = $random;
        repeat(15) #60 A = A + 1;
        #100 $stop;
    end

endmodule
```