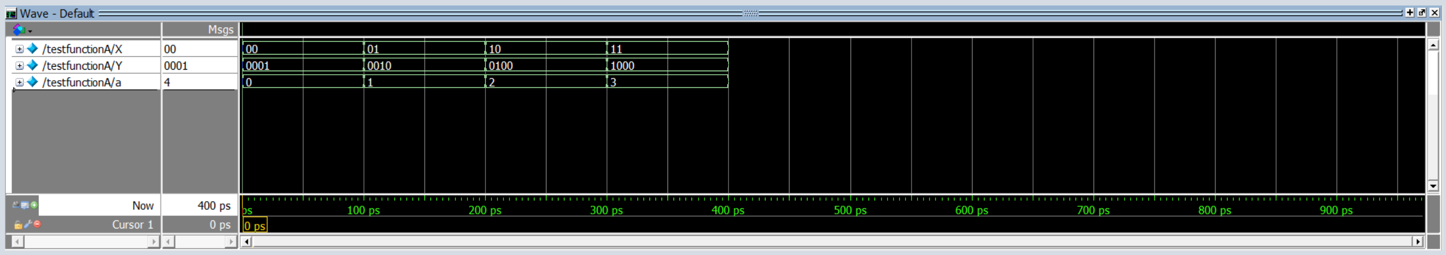
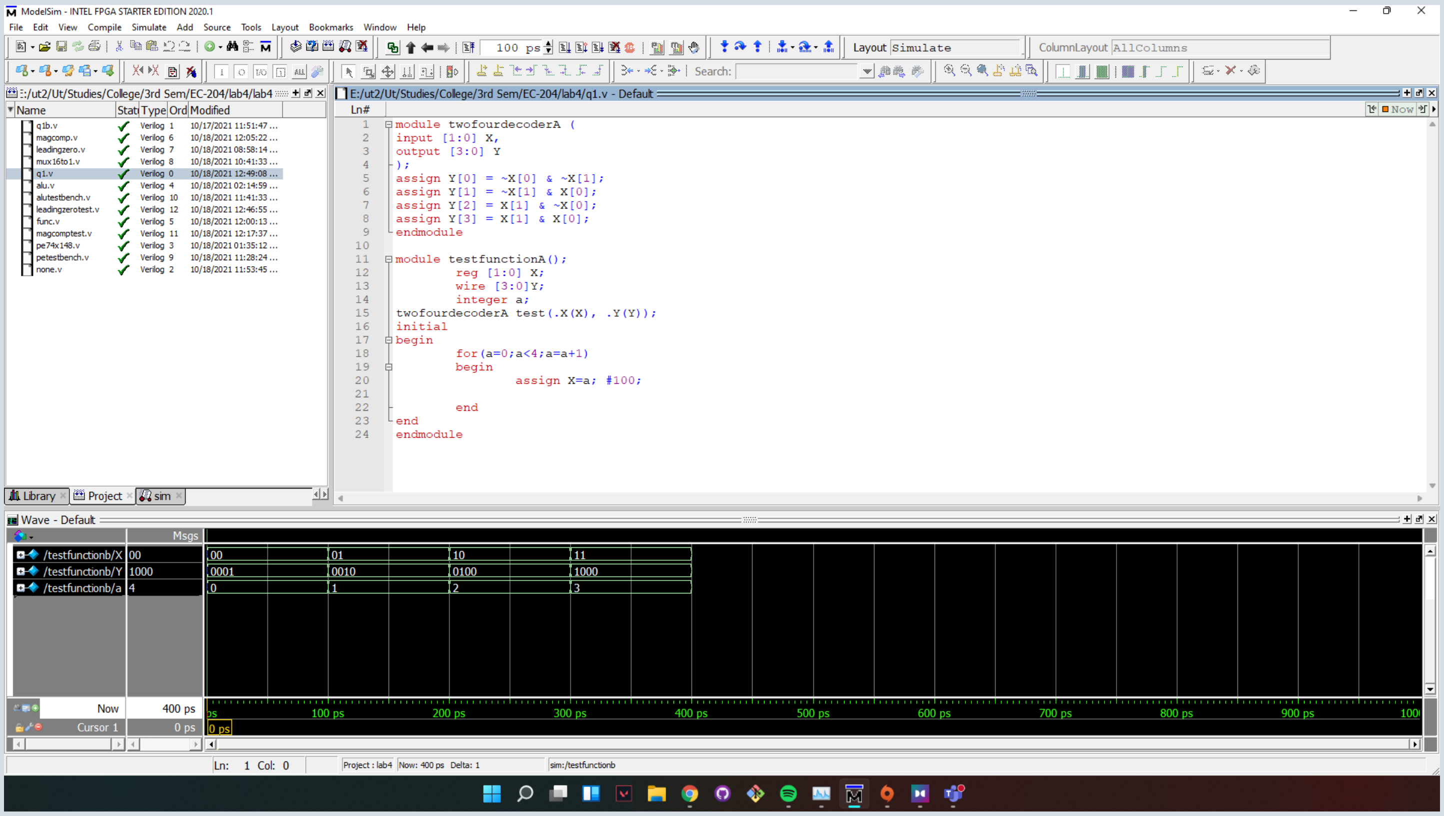
1] 2 to 4 decoder using

(a) concurrent signal assignment statements

```verilog
module twofourdecoderA (
input [1:0] X,
output [3:0] Y
);
assign Y[0] = ~X[0] & ~X[1];
assign Y[1] = ~X[1] & X[0];
assign Y[2] = X[1] & ~X[0];
assign Y[3] = X[1] & X[0];
endmodule

module testfunctionA();
    reg [1:0] X;
    wire [3:0]Y;
    integer a;
twofourdecoderA test(.X(X), .Y(Y));
initial
begin
    for(a=0;a<4;a=a+1)
    begin
        assign X=a; #100;

    end
end
endmodule
```

Wave:




ModelSim - INTEL FPGA STARTER EDITION 2020.1

E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/lab4

Name	Stat	Type	Ord	Modified
q1b.v	✓	Verilog	1	10/17/2021 11:51:47 ...
magcomp.v	✓	Verilog	6	10/18/2021 12:05:22 ...
leadingzero.v	✓	Verilog	7	10/18/2021 08:58:14 ...
mux16to1.v	✓	Verilog	8	10/18/2021 10:41:33 ...
q1.v	✓	Verilog	0	10/18/2021 12:49:08 ...
alu.v	✓	Verilog	4	10/18/2021 02:14:59 ...
alutestbench.v	✓	Verilog	10	10/18/2021 11:41:33 ...
leadingzerotest.v	✓	Verilog	12	10/18/2021 12:46:55 ...
func.v	✓	Verilog	5	10/18/2021 12:00:13 ...
magcomptest.v	✓	Verilog	11	10/18/2021 12:17:37 ...
pe74x148.v	✓	Verilog	3	10/18/2021 01:35:12 ...
petestbench.v	✓	Verilog	9	10/18/2021 11:28:24 ...
none.v	✓	Verilog	2	10/18/2021 11:53:45 ...

E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/q1.v - Default

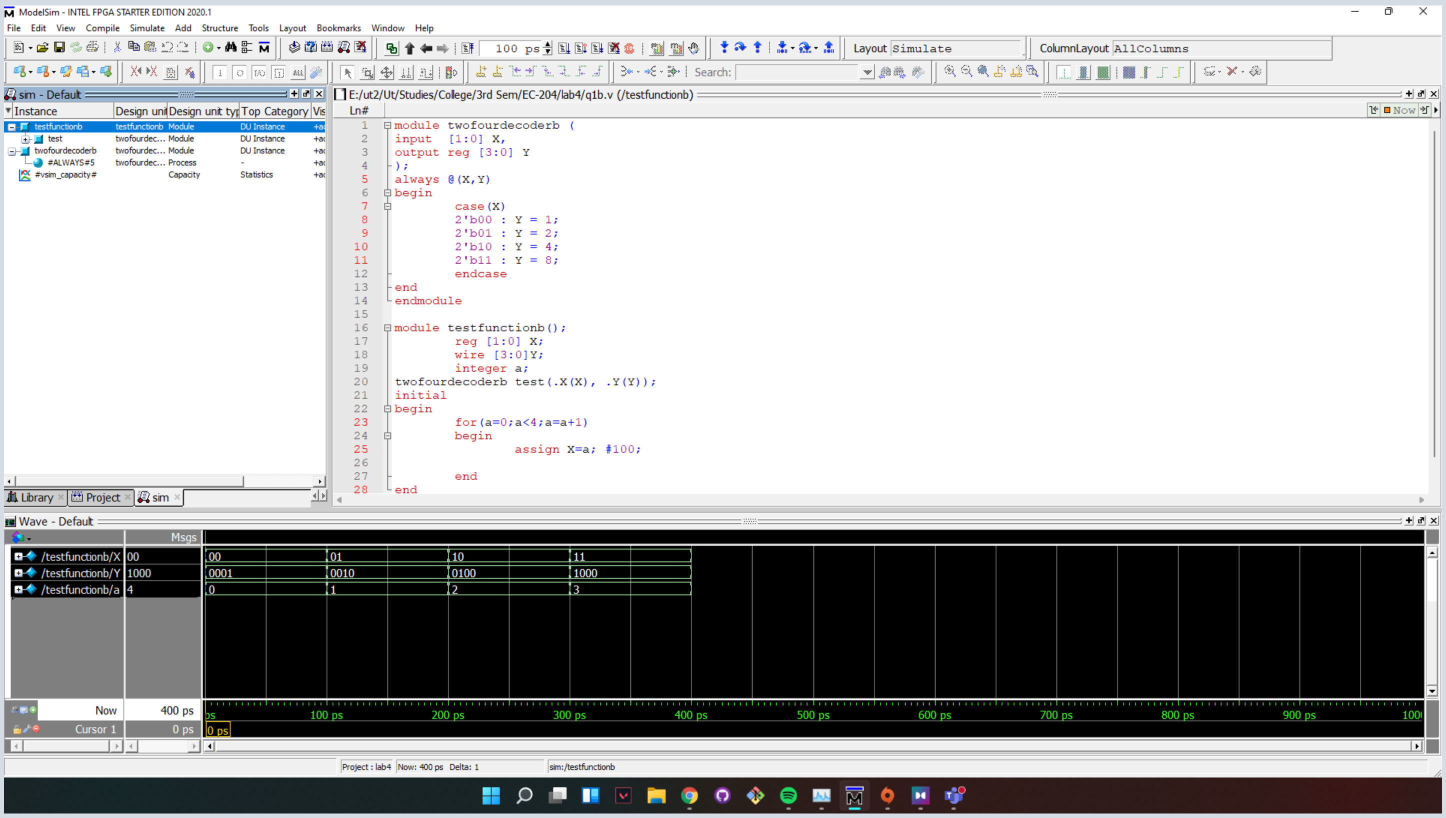
```
module twofourdecoderA (
input [1:0] X,
output [3:0] Y
);
assign Y[0] = ~X[0] & ~X[1];
assign Y[1] = ~X[1] & X[0];
assign Y[2] = X[1] & ~X[0];
assign Y[3] = X[1] & X[0];
endmodule

module testfunctionA();
        reg [1:0] X;
        wire [3:0]Y;
        integer a;
twofourdecoderA test(.X(X), .Y(Y));
initial
begin
        for(a=0;a<4;a=a+1)
        begin
                assign X=a; #100;

        end
end
endmodule
```

Wave - Default

```
/testfunctionb/X  00    00     01     10     11
/testfunctionb/Y  1000  0001   0010   0100   1000
/testfunctionb/a  4     0      1      2      3
Now       400 ps
Cursor 1  0 ps
```

Ln: 1 Col: 0   Project : lab4   Now: 400 ps   Delta: 1   sim:/testfunctionb

(b) using case statement


ModelSim - INTEL FPGA STARTER EDITION 2020.1

sim - Default

Instance	Design unit	Design unit type	Top Category	Vis
testfunctionb	testfunctionb	Module	DU Instance	+ac
test	twofourdec...	Module	DU Instance	+ac
twofourdecoderb	twofourdec...	Module	DU Instance	+ac
#ALWAYS#5	twofourdec...	Process	-	+ac
#vsim_capacity#		Capacity	Statistics	+ac

E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/q1b.v (/testfunctionb)

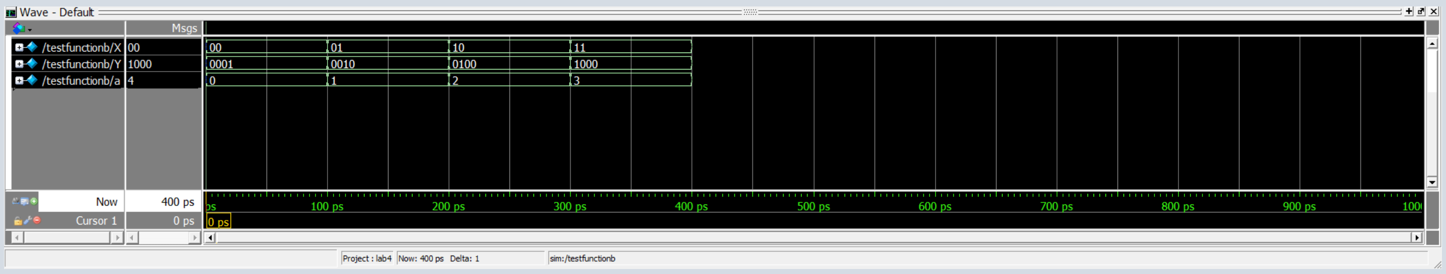
```
module twofourdecoderb (
input   [1:0] X,
output reg [3:0] Y
);
always @(X,Y)
begin
        case(X)
        2'b00 : Y = 1;
        2'b01 : Y = 2;
        2'b10 : Y = 4;
        2'b11 : Y = 8;
        endcase
end
endmodule

module testfunctionb();
        reg [1:0] X;
        wire [3:0]Y;
        integer a;
twofourdecoderb test(.X(X), .Y(Y));
initial
begin
        for(a=0;a<4;a=a+1)
        begin
                assign X=a; #100;

        end
end
```

Wave - Default

```
/testfunctionb/X  00    00     01     10     11
/testfunctionb/Y  1000  0001   0010   0100   1000
/testfunctionb/a  4     0      1      2      3
Now       400 ps
Cursor 1  0 ps
```

Project : lab4   Now: 400 ps   Delta: 1   sim:/testfunctionb

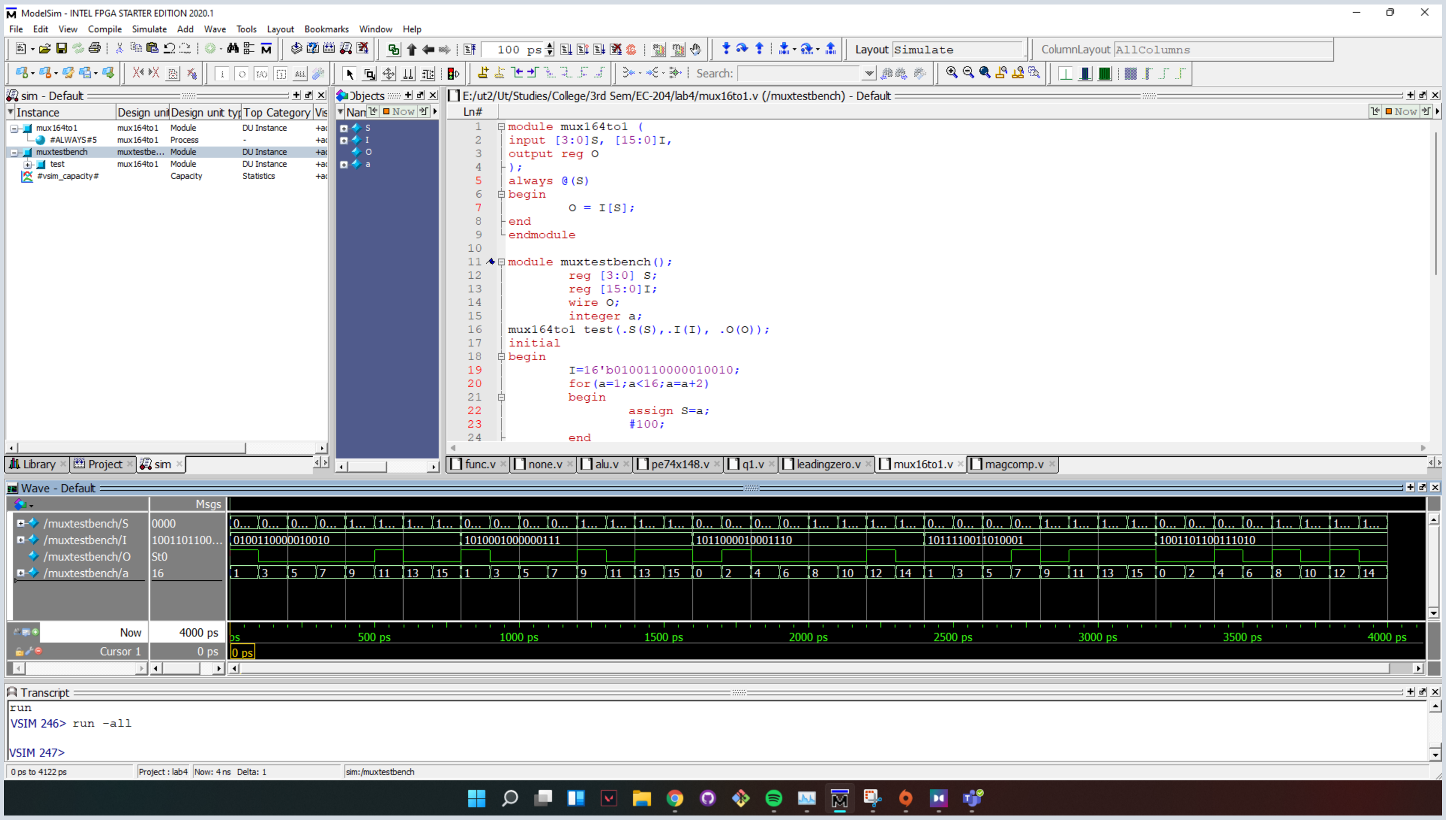
Wave:



```
module twofourdecoderb (
input  [1:0] X,
output reg [3:0] Y
);
always @(X,Y)
begin
    case(X)
    2'b00 : Y = 1;
    2'b01 : Y = 2;
    2'b10 : Y = 4;
    2'b11 : Y = 8;
    endcase
end
endmodule

module testfunctionb();
    reg [1:0] X;
    wire [3:0]Y;
    integer a;
twofourdecoderb test(.X(X), .Y(Y));
initial
begin
    for(a=0;a<4;a=a+1)
    begin
        assign X=a; #100;

    end
end
endmodule
```

2] 16:1 multiplexer

```verilog
module mux164to1 (
input [3:0]S, [15:0]I,
output reg O
);
always @(S)
begin
    O = I[S];
end
endmodule
```



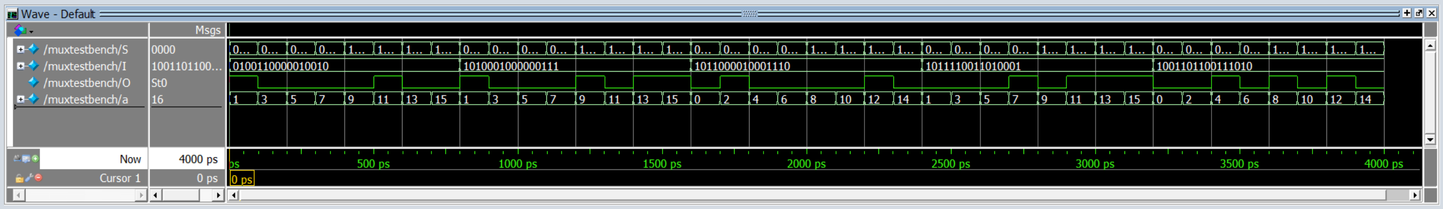
E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/mux16to1.v (/muxtestbench) - Default

```verilog
module mux164to1 (
input [3:0]S, [15:0]I,
output reg O
);
always @(S)
begin
        O = I[S];
end
endmodule

module muxtestbench();
        reg [3:0] S;
        reg [15:0]I;
        wire O;
        integer a;
mux164to1 test(.S(S),.I(I), .O(O));
initial
begin
        I=16'b0100110000010010;
        for(a=1;a<16;a=a+2)
        begin
                assign S=a;
                #100;
        end
```

```
/muxtestbench/S   0000
/muxtestbench/I   1001101100...
/muxtestbench/O   St0
/muxtestbench/a   16
Now               4000 ps
Cursor 1          0 ps
```

```
run
VSIM 246> run -all

VSIM 247>
```

Wave:

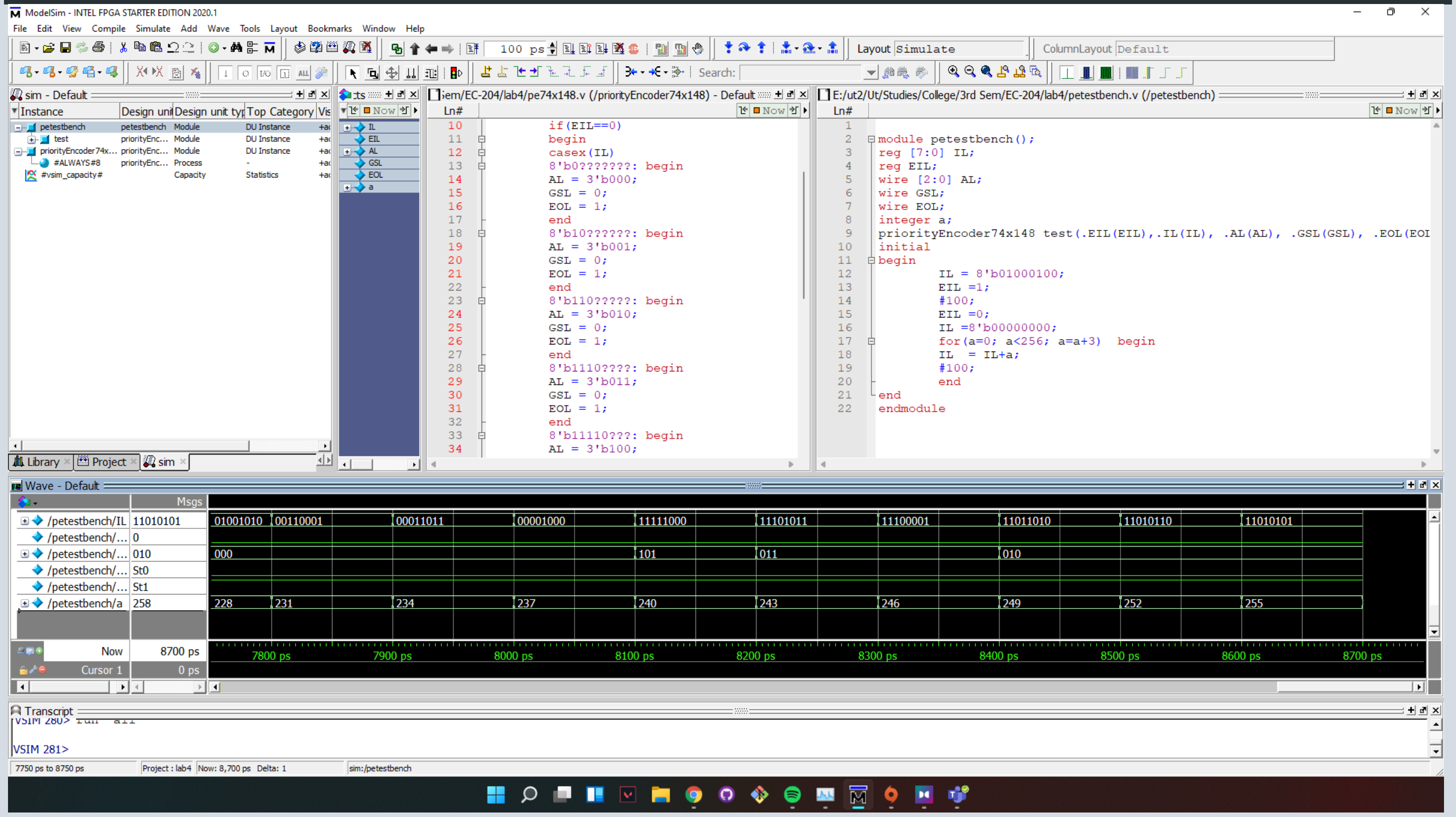


```
/muxtestbench/S   0000
/muxtestbench/I   1001101100...
/muxtestbench/O   St0
/muxtestbench/a   16
Now               4000 ps
Cursor 1          0 ps
```

3] Functionality of 74x148 priority encoder

```
module priorityEncoder74x148 (
input  [7:0] IL,
input EIL,
output reg [2:0] AL,
output reg GSL,
output reg EOL
);
always @(EIL,IL)
begin
    if(EIL==0)
    begin
    casex(IL)
    8'b0???????: begin
    AL = 3'b000;
    GSL = 0;
    EOL = 1;
    end
    8'b10??????: begin
    AL = 3'b001;
    GSL = 0;
    EOL = 1;
    end
    8'b110?????: begin
    AL = 3'b010;
    GSL = 0;
    EOL = 1;
    end
    8'b1110????: begin
    AL = 3'b011;
    GSL = 0;
    EOL = 1;
    end
    8'b11110???: begin
    AL = 3'b100;
    GSL = 0;
    EOL = 1;
    end
    8'b111110??: begin
    AL = 3'b101;
    GSL = 0;
    EOL = 1;
    end
    8'b1111110?: begin
```

```
        AL = 3'b110;
        GSL = 0;
        EOL = 1;
        end
        8'b11111110: begin
        AL = 3'b111;
        GSL = 0;
        EOL = 1;
        end
        8'b11111111: begin
        AL = 3'b111;
        GSL = 1;
        EOL = 0;
        end
        endcase
        end else begin
        AL = 3'b111;
        GSL = 1;
        EOL = 1;
        end
end
endmodule
```



ModelSim - INTEL FPGA STARTER EDITION 2020.1

sem/EC-204/lab4/pe74x148.v (/priorityEncoder74x148) - Default

```
        if(EIL==0)
        begin
        casex(IL)
        8'b0???????: begin
        AL = 3'b000;
        GSL = 0;
        EOL = 1;
        end
        8'b10??????: begin
        AL = 3'b001;
        GSL = 0;
        EOL = 1;
        end
        8'b110?????: begin
        AL = 3'b010;
        GSL = 0;
        EOL = 1;
        end
        8'b1110????: begin
        AL = 3'b011;
        GSL = 0;
        EOL = 1;
        end
        8'b11110???: begin
        AL = 3'b100;
```

E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/petestbench.v (/petestbench)

```

module petestbench();
reg [7:0] IL;
reg EIL;
wire [2:0] AL;
wire GSL;
wire EOL;
integer a;
priorityEncoder74x148 test(.EIL(EIL),.IL(IL), .AL(AL), .GSL(GSL), .EOL(EOI
initial
begin
        IL = 8'b01000100;
        EIL =1;
        #100;
        EIL =0;
        IL =8'b00000000;
        for(a=0; a<256; a=a+3)  begin
        IL  = IL+a;
        #100;
        end
end
endmodule
```

Wave - Default

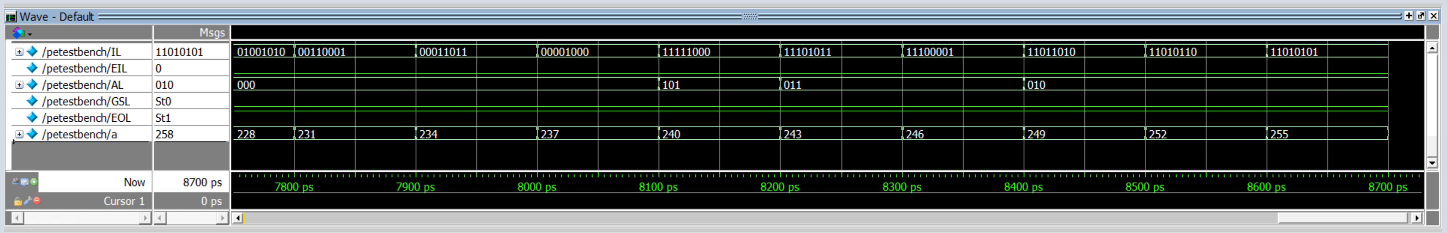
```
                        Msgs
/petestbench/IL     11010101   01001010 00110001 00011011 00001000 11111000 11101011 11100001 11011010 11010110 11010101
/petestbench/...    0
/petestbench/...    010        000                                 101      011               010
/petestbench/...    St0
/petestbench/...    St1
/petestbench/a      258        228      231      234      237      240      243      246      249      252      255

Now                 8700 ps       7800 ps  7900 ps  8000 ps  8100 ps  8200 ps  8300 ps  8400 ps  8500 ps  8600 ps  8700 ps
Cursor 1            0 ps
```

Transcript

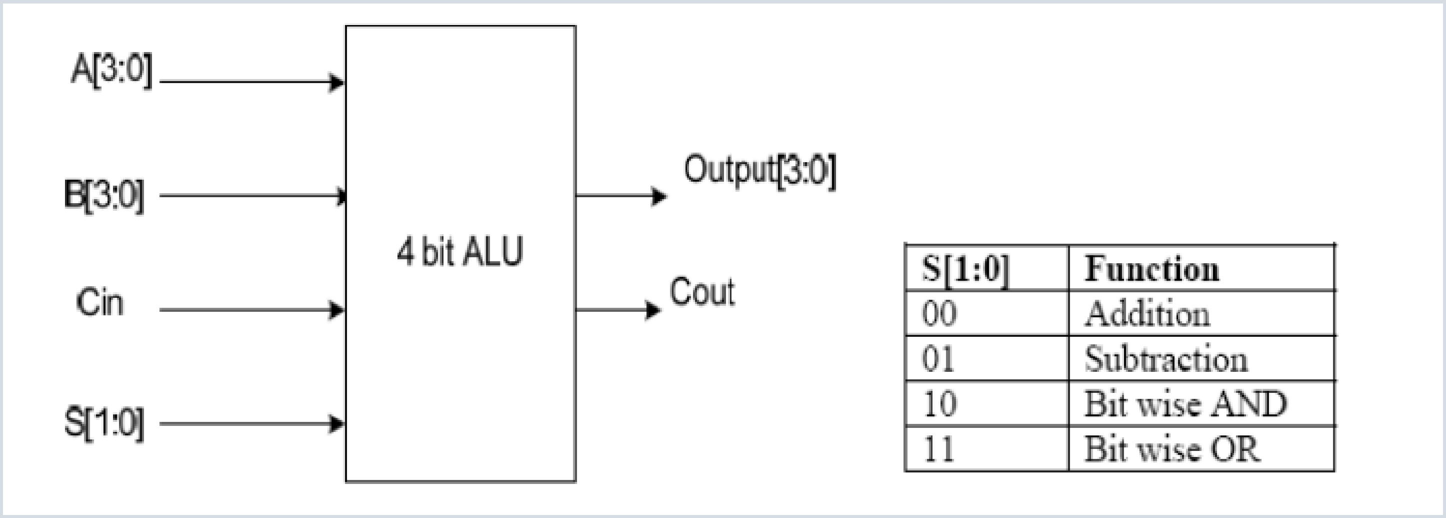
```
VSIM 281>
```

7750 ps to 8750 ps | Project : lab4 | Now: 8,700 ps Delta: 1 | sim:/petestbench

Wave:

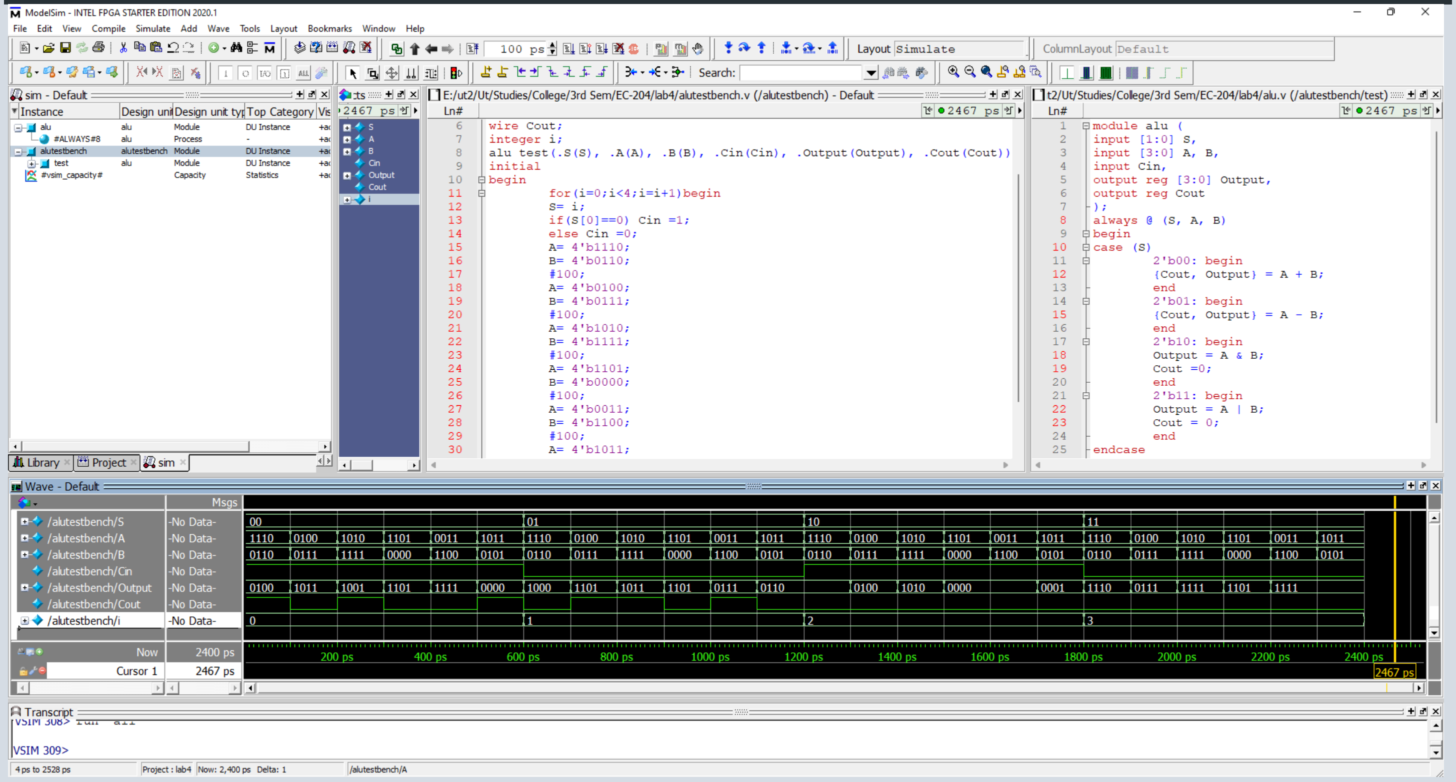


4] ALU design: Design a 4 bit ALU that is capable of performing addition, subtraction, bitwise AND and bit-wise OR instructions on 4 bit operands.



S[1:0]	Function
00	Addition
01	Subtraction
10	Bit wise AND
11	Bit wise OR

```
module alu (
input [1:0] S,
input [3:0] A, B,
input Cin,
output reg [3:0] Output,
output reg Cout
);
always @ (S, A, B)
begin
case (S)
    2'b00: begin
    {Cout, Output} = A + B;
    end
    2'b01: begin
    {Cout, Output} = A - B;
    end
    2'b10: begin
    Output = A & B;
    Cout =0;
    end
    2'b11: begin
    Output = A | B;
    Cout = 0;
    end
endcase
end
endmodule
```



ModelSim - INTEL FPGA STARTER EDITION 2020.1

E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/alutestbench.v (/alutestbench) - Default

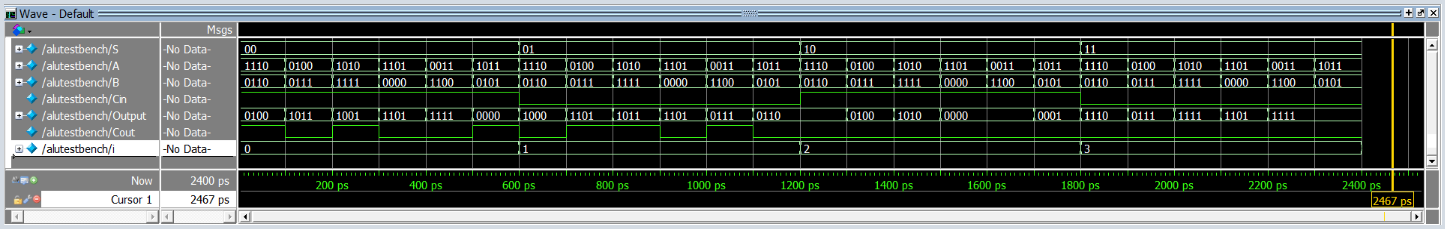
```
wire Cout;
integer i;
alu test(.S(S), .A(A), .B(B), .Cin(Cin), .Output(Output), .Cout(Cout))
initial
begin
        for(i=0;i<4;i=i+1)begin
        S= i;
        if(S[0]==0) Cin =1;
        else Cin =0;
        A= 4'b1110;
        B= 4'b0110;
        #100;
        A= 4'b0100;
        B= 4'b0111;
        #100;
        A= 4'b1010;
        B= 4'b1111;
        #100;
        A= 4'b1101;
        B= 4'b0000;
        #100;
        A= 4'b0011;
        B= 4'b1100;
        #100;
        A= 4'b1011;
```

t2/Ut/Studies/College/3rd Sem/EC-204/lab4/alu.v (/alutestbench/test)

```
module alu (
input [1:0] S,
input [3:0] A, B,
input Cin,
output reg [3:0] Output,
output reg Cout
);
always @ (S, A, B)
begin
case (S)
        2'b00: begin
        {Cout, Output} = A + B;
        end
        2'b01: begin
        {Cout, Output} = A - B;
        end
        2'b10: begin
        Output = A & B;
        Cout =0;
        end
        2'b11: begin
        Output = A | B;
        Cout = 0;
        end
endcase
```

```
/alutestbench/S       00 01 10 11
/alutestbench/A       1110 0100 1010 1101 0011 1011 (repeated for each S)
/alutestbench/B       0110 0111 1111 0000 1100 0101 (repeated for each S)
/alutestbench/Cin
/alutestbench/Output  0100 1011 1001 1101 1111 0000 1000 1101 1011 1101 0111 0110 0100 1010 0000 0001 1110 0111 1111 1101 1111
/alutestbench/Cout
/alutestbench/i       0 1 2 3
Now 2400 ps   Cursor 1 2467 ps
VSIM 309>
```

Wave:

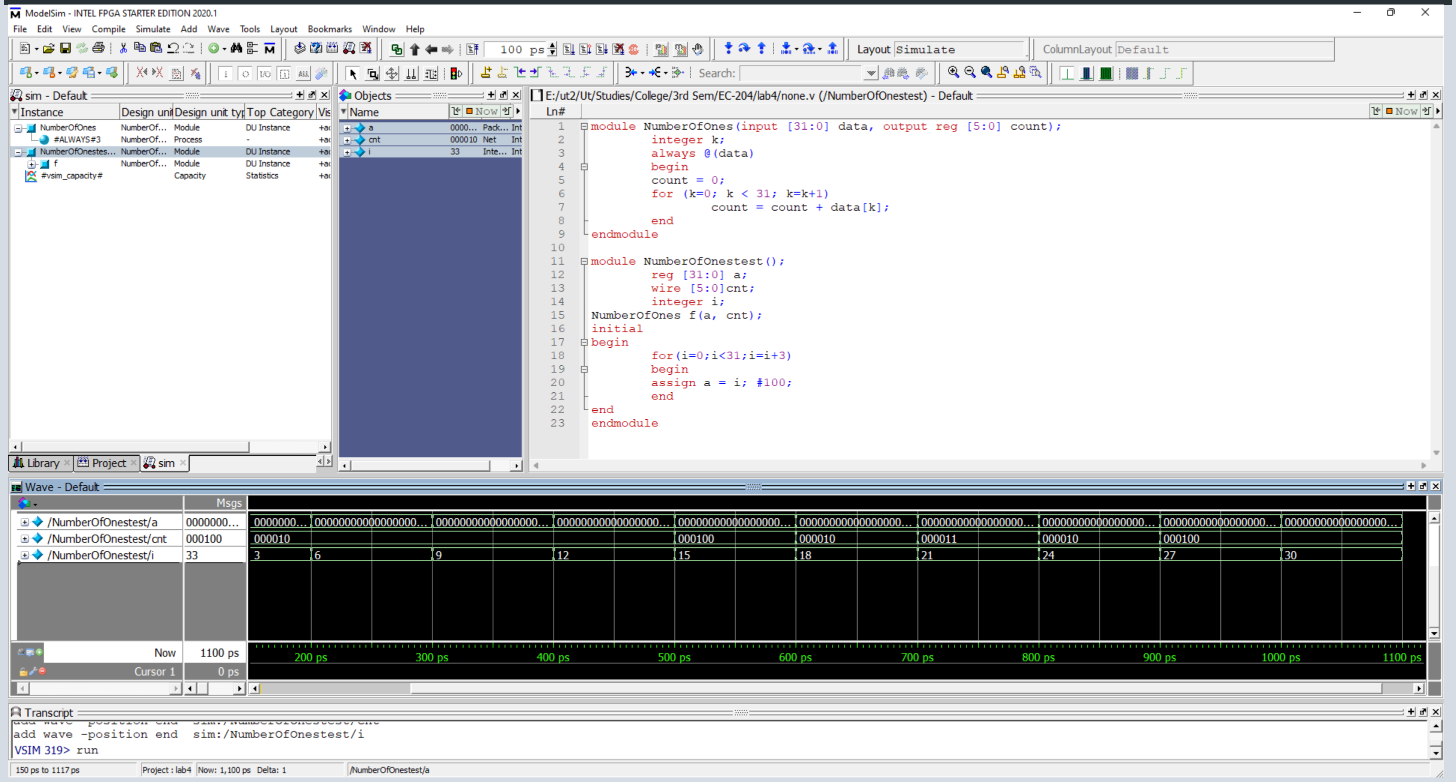


```
/alutestbench/S       00 01 10 11
/alutestbench/A       1110 0100 1010 1101 0011 1011 (repeated for each S)
/alutestbench/B       0110 0111 1111 0000 1100 0101 (repeated for each S)
/alutestbench/Cin
/alutestbench/Output  0100 1011 1001 1101 1111 0000 1000 1101 1011 1101 0111 0110 0100 1010 0000 0001 1110 0111 1111 1101 1111
/alutestbench/Cout
/alutestbench/i       0 1 2 3
Now 2400 ps   Cursor 1 2467 ps
```

5] Count the number of 1s in a 32 bit number

```
module NumberOfOnes(input [31:0] data, output reg [5:0] count);
    integer k;
    always @(data)
    begin
    count = 0;
    for (k=0; k < 31; k=k+1)
        count = count + data[k];
    end
endmodule
```

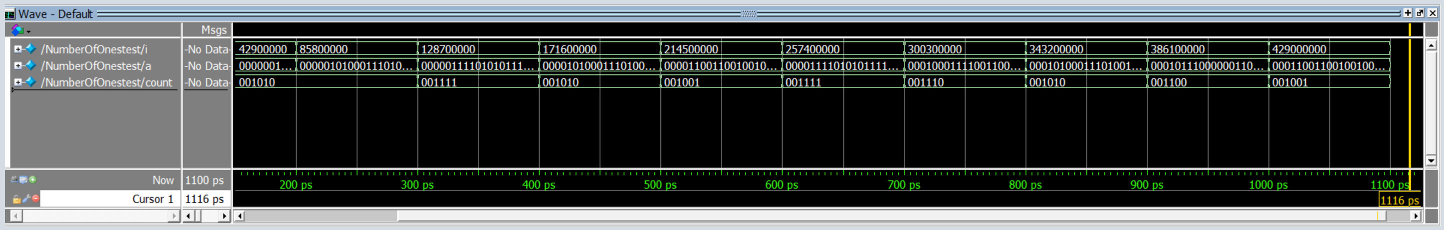
```
module NumberOfOnestest();
    reg [31:0] a;
    wire [5:0]count;
    integer i;
NumberOfOnes f(.data(a), .count(count));
initial
begin
    for(i=0;i<429496729;i=i+42900000)
    begin
    assign a = i; #100;
    end
end
endmodule
```



```
module NumberOfOnes(input [31:0] data, output reg [5:0] count);
        integer k;
        always @(data)
        begin
        count = 0;
        for (k=0; k < 31; k=k+1)
                count = count + data[k];
        end
endmodule

module NumberOfOnestest();
        reg [31:0] a;
        wire [5:0]cnt;
        integer i;
NumberOfOnes f(a, cnt);
initial
begin
        for(i=0;i<31;i=i+3)
        begin
        assign a = i; #100;
        end
end
endmodule
```

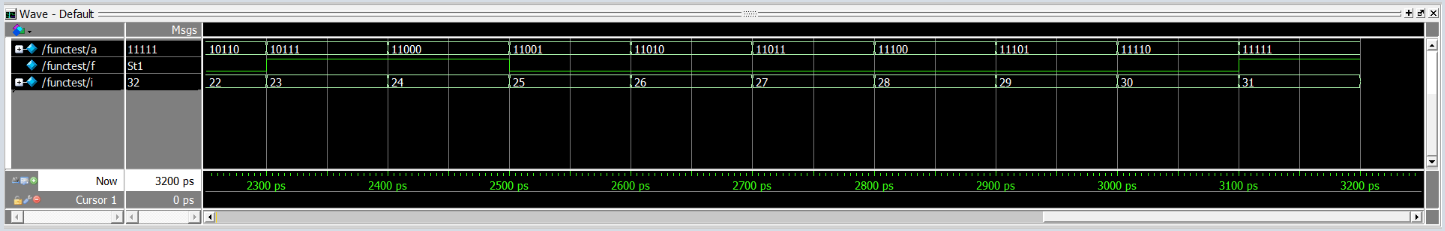
```
add wave -position end  sim:/NumberOfOnestest/i
VSIM 319> run
```

Wave:

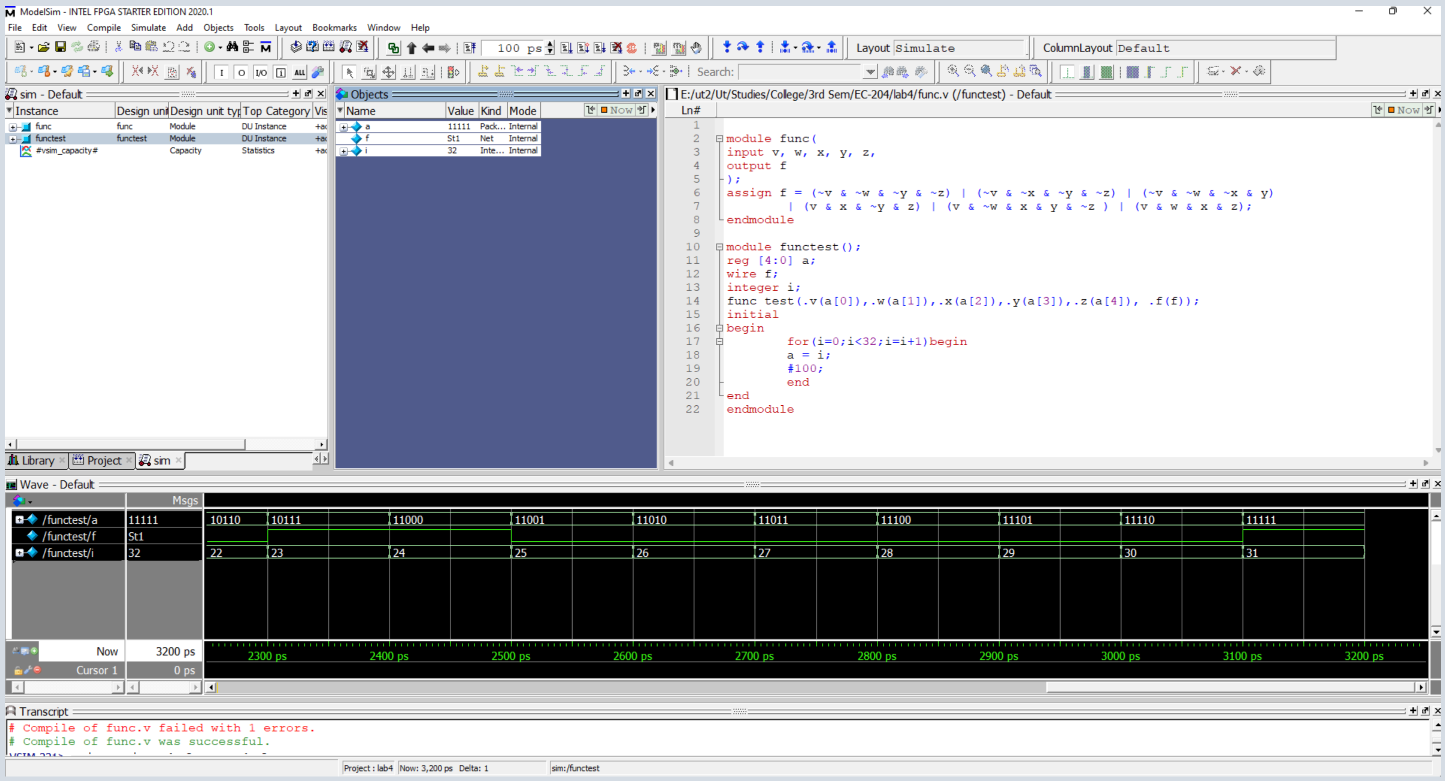


6] Implement F(v,w,x,y,z) = ∑ (0, 2, 3, 4, 8, 21, 22, 29, 31)

```verilog
module func(
input v, w, x, y, z,
output f
);
assign f = (~v & ~w & ~y & ~z) | (~v & ~x & ~y & ~z) | (~v & ~w &
~x & y)
    | (v & x & ~y & z) | (v & ~w & x & y & ~z ) | (v & w & x & z);
endmodule

module functest();
reg [4:0] a;
wire f;
integer i;
func test(.v(a[0]),.w(a[1]),.x(a[2]),.y(a[3]),.z(a[4]), .f(f));
initial
begin
    for(i=0;i<32;i=i+1)begin
    a = i;
    #100;
    end
end
endmodule
```



```
Wave - Default
                    Msgs
/functest/a    11111    10110 | 10111 | 11000 | 11001 | 11010 | 11011 | 11100 | 11101 | 11110 | 11111
/functest/f    St1
/functest/i    32       22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31

Now            3200 ps     2300 ps  2400 ps  2500 ps  2600 ps  2700 ps  2800 ps  2900 ps  3000 ps  3100 ps  3200 ps
Cursor 1       0 ps
```



E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/func.v (/functest) - Default

```verilog

module func(
input v, w, x, y, z,
output f
);
assign f = (~v & ~w & ~y & ~z) | (~v & ~x & ~y & ~z) | (~v & ~w & ~x & y)
        | (v & x & ~y & z) | (v & ~w & x & y & ~z ) | (v & w & x & z);
endmodule

module functest();
reg [4:0] a;
wire f;
integer i;
func test(.v(a[0]),.w(a[1]),.x(a[2]),.y(a[3]),.z(a[4]), .f(f));
initial
begin
        for(i=0;i<32;i=i+1)begin
        a = i;
        #100;
        end
end
endmodule
```

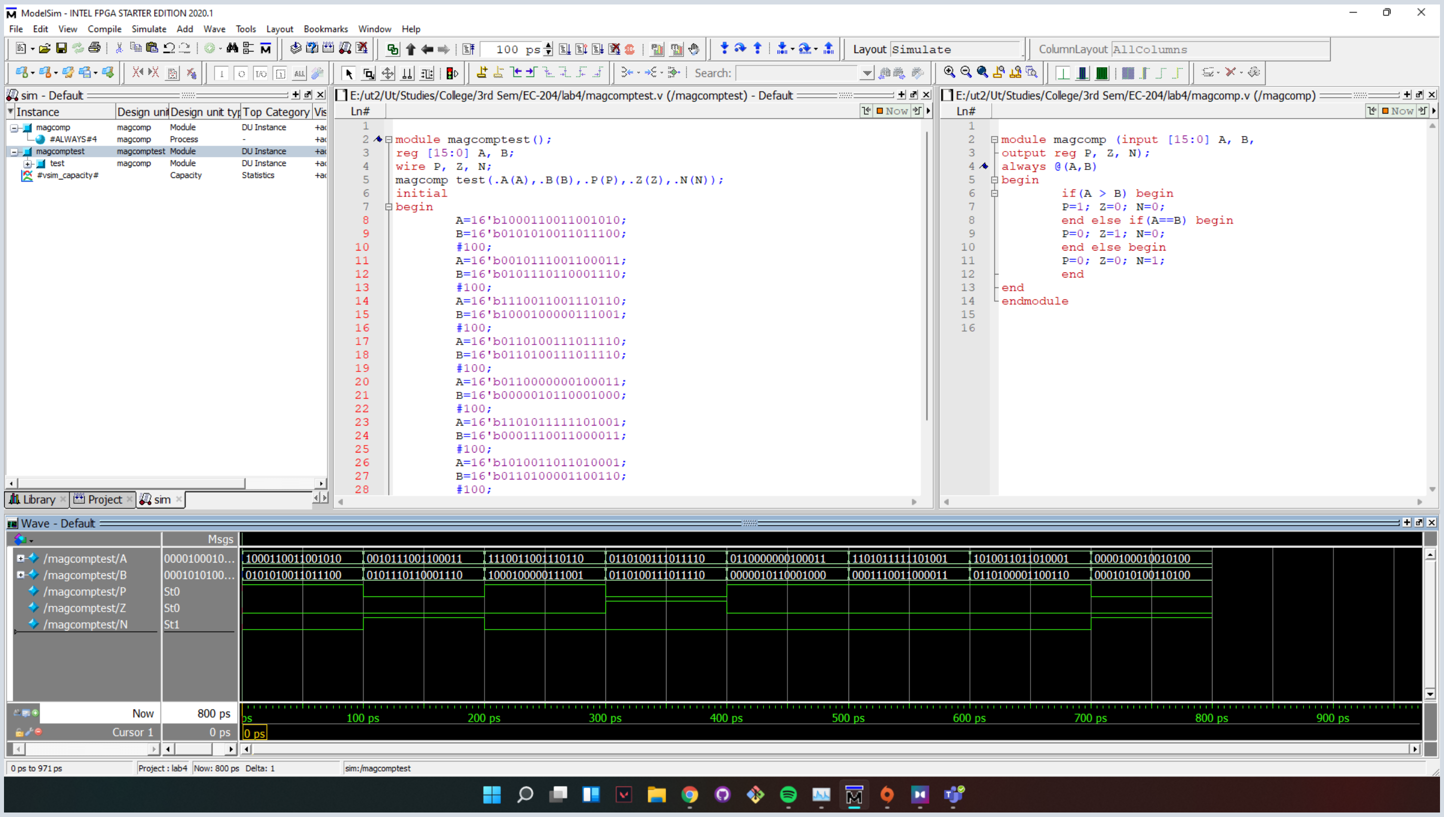
```
/functest/a   11111   10110 10111 11000 11001 11010 11011 11100 11101 11110 11111
/functest/f   St1
/functest/i   32      22    23    24    25    26    27    28    29    30    31

Now      3200 ps   2300 ps  2400 ps  2500 ps  2600 ps  2700 ps  2800 ps  2900 ps  3000 ps  3100 ps  3200 ps
Cursor 1    0 ps
```

```
# Compile of func.v failed with 1 errors.
# Compile of func.v was successful.
```

Project : lab4   Now: 3,200 ps   Delta: 1   sim:/functest

7] 16 bit magnitude comparator

```verilog
module magcomp (input [15:0] A, B,
output reg P, Z, N);
always @(A,B)
begin
    if(A > B) begin
    P=1; Z=0; N=0;
    end else if(A==B) begin
    P=0; Z=1; N=0;
    end else begin
    P=0; Z=0; N=1;
    end
end
endmodule
```

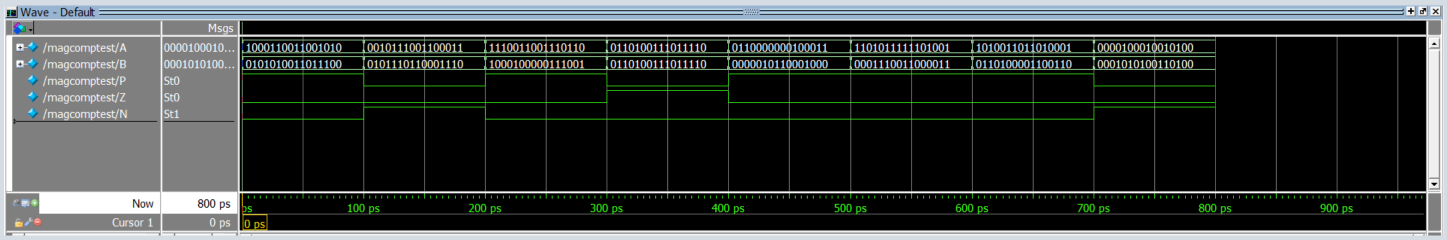


E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/magcomptest.v (/magcomptest) - Default

```
module magcomptest();
reg [15:0] A, B;
wire P, Z, N;
magcomp test(.A(A),.B(B),.P(P),.Z(Z),.N(N));
initial
begin
        A=16'b1000110011001010;
        B=16'b0101010011011100;
        #100;
        A=16'b0010111001100011;
        B=16'b0101110110001110;
        #100;
        A=16'b1110011001110110;
        B=16'b1000100000111001;
        #100;
        A=16'b0110100111011110;
        B=16'b0110100111011110;
        #100;
        A=16'b0110000000100011;
        B=16'b0000010110001000;
        #100;
        A=16'b1101011111101001;
        B=16'b0001110011000011;
        #100;
        A=16'b1010011011010001;
        B=16'b0110100001100110;
        #100;
```

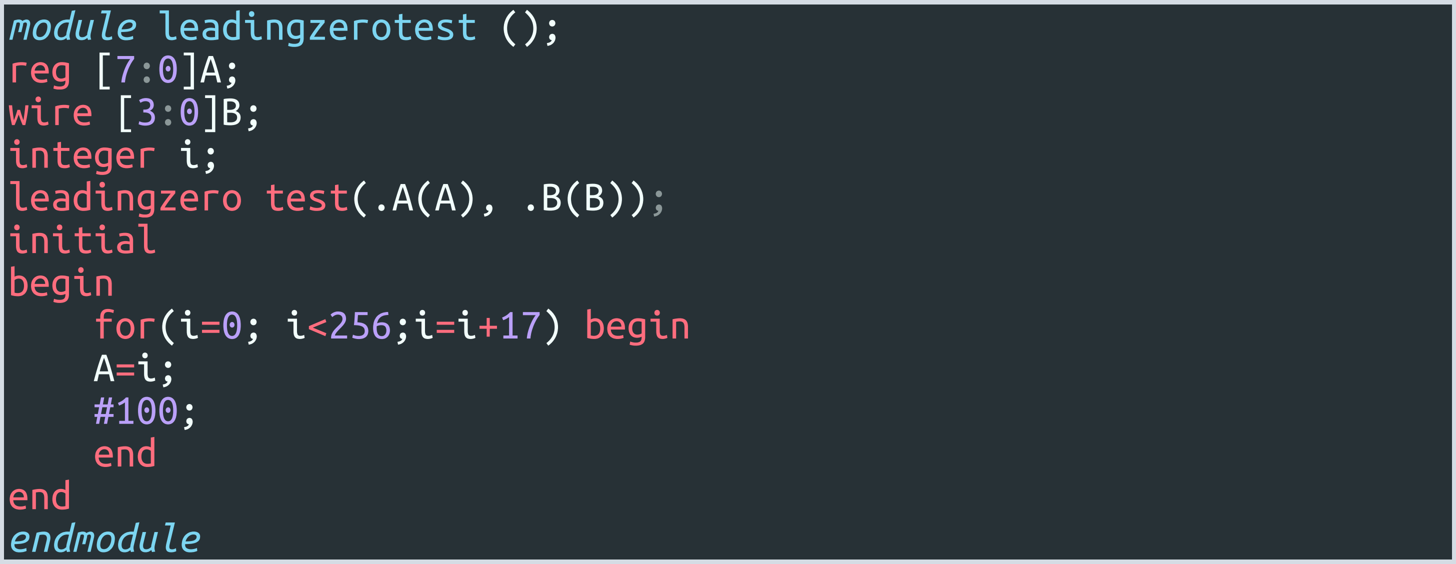
E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/magcomp.v (/magcomp)

```
module magcomp (input [15:0] A, B,
output reg P, Z, N);
always @(A,B)
begin
        if(A > B) begin
        P=1; Z=0; N=0;
        end else if(A==B) begin
        P=0; Z=1; N=0;
        end else begin
        P=0; Z=0; N=1;
        end
end
endmodule
```

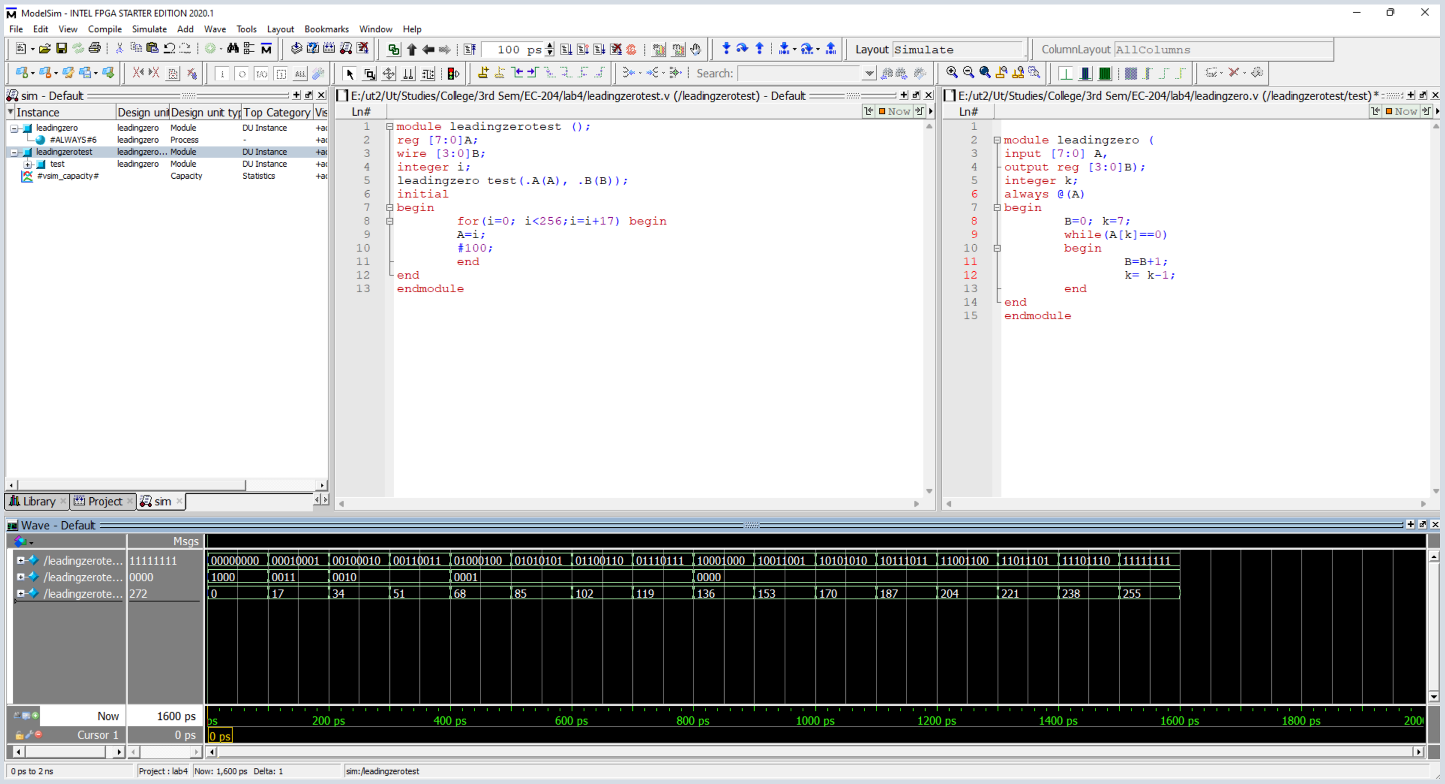


8] Count the number of leading 0s in an 8 bit number

```
module leadingzero (
input [7:0] A,
output reg [3:0]B);
integer k;
always @(A)
begin
    B=0; k=7;
    while(A[k]==0)
    begin
        B=B+1;
        k= k-1;
    end
end
endmodule
```



```verilog
module leadingzerotest ();
reg [7:0]A;
wire [3:0]B;
integer i;
leadingzero test(.A(A), .B(B));
initial
begin
    for(i=0; i<256;i=i+17) begin
    A=i;
    #100;
    end
end
endmodule
```



ModelSim - INTEL FPGA STARTER EDITION 2020.1

Instance	Design unit	Design unit type	Top Category	Vis
leadingzero	leadingzero	Module	DU Instance	+ac
#ALWAYS#6	leadingzero	Process	-	+ac
leadingzerotest	leadingzero...	Module	DU Instance	+ac
test	leadingzero	Module	DU Instance	+ac
#vsim_capacity#		Capacity	Statistics	+ac

E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/leadingzerotest.v (/leadingzerotest) - Default

```verilog
module leadingzerotest ();
reg [7:0]A;
wire [3:0]B;
integer i;
leadingzero test(.A(A), .B(B));
initial
begin
        for(i=0; i<256;i=i+17) begin
        A=i;
        #100;
        end
end
endmodule
```

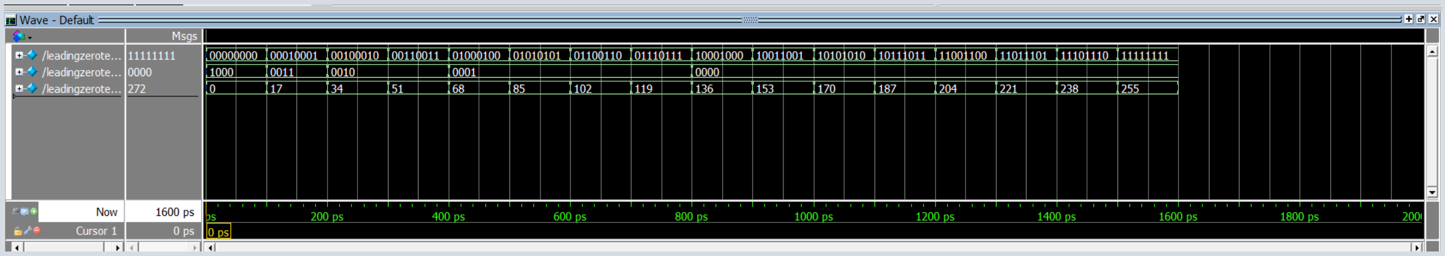
E:/ut2/Ut/Studies/College/3rd Sem/EC-204/lab4/leadingzero.v (/leadingzerotest/test) *

```verilog

module leadingzero (
input [7:0] A,
output reg [3:0]B);
integer k;
always @(A)
begin
        B=0; k=7;
        while(A[k]==0)
        begin
                B=B+1;
                k= k-1;
        end
end
endmodule
```

Wave - Default

```
/leadingzerote... 11111111  00000000 00010001 00100010 00110011 01000100 01010101 01100110 01110111 10001000 10011001 10101010 10111011 11001100 11011101 11101110 11111111
/leadingzerote... 0000      1000     0011     0010              0001                                  0000
/leadingzerote... 272       0        17       34       51       68       85       102      119      136      153      170      187      204      221      238      255

Now 1600 ps        ps  200 ps  400 ps  600 ps  800 ps  1000 ps  1200 ps  1400 ps  1600 ps  1800 ps  2000
Cursor 1 0 ps      0 ps

0 ps to 2 ns   Project : lab4   Now: 1,600 ps  Delta: 1   sim:/leadingzerotest
```

Wave:



Wave - Default

```
/leadingzerote... 11111111  00000000 00010001 00100010 00110011 01000100 01010101 01100110 01110111 10001000 10011001 10101010 10111011 11001100 11011101 11101110 11111111
/leadingzerote... 0000      1000     0011     0010              0001                                  0000
/leadingzerote... 272       0        17       34       51       68       85       102      119      136      153      170      187      204      221      238      255

Now 1600 ps        ps  200 ps  400 ps  600 ps  800 ps  1000 ps  1200 ps  1400 ps  1600 ps  1800 ps  2000
Cursor 1 0 ps      0 ps
```