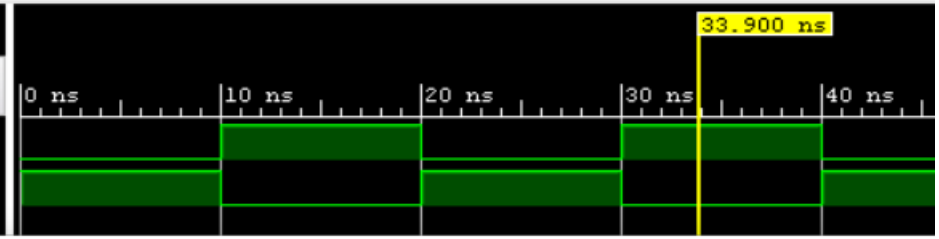
Basys3_Master.xdc | inverter.v | inverter_tb.v | Untitled 1

Name	Value
a	1
b	0



inverter_tb.v + (~\Desktop\Engr433) - GVIM2

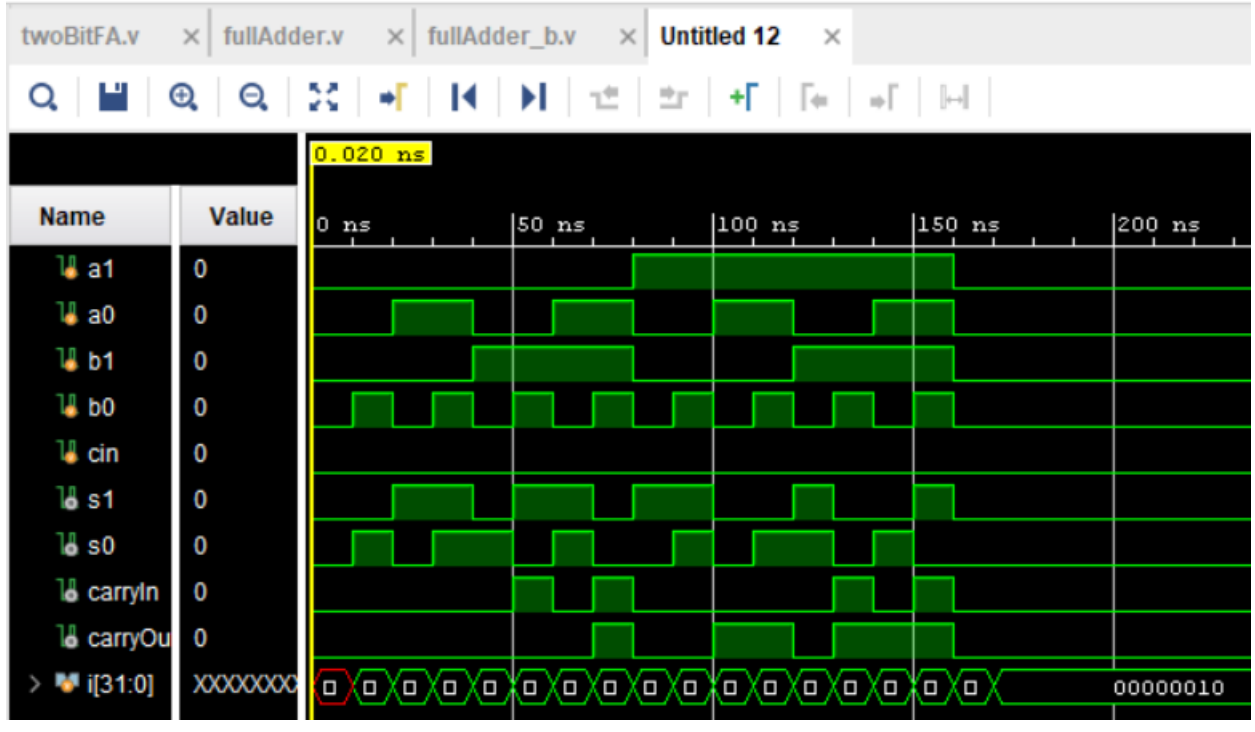
File Edit Tools Syntax Buffers Window Help

```
module invert_tb;
reg a;
wire b;

inverter UUT (.a(a),.b(b));

initial
begin
        a = 1'b0;
        #10;
        a = 1'b1;
        #10;
        a = 1'b0;
    #10;
    a = 1'b1;
    #10;
    a = 1'b0;
    #10;
    a = 1'b1;
    #10;
end
endmodule
~
```

```
module inverter(input a, output b);

assign  b = ~a;

endmodule
~
```

inverter_tb.v [+] 22,1 All inverter.v 5,1 Al

"inverter_tb.v" 23L, 239B


twoBitFA.v
fullAdder.v
fullAdder_b.v
Untitled 12
0.020 ns
Name
Value
0 ns
50 ns
100 ns
150 ns
200 ns
a1 0
a0 0
b1 0
b0 0
cin 0
s1 0
s0 0
carryIn 0
carryOu 0
i[31:0] XXXXXXXX
00000010

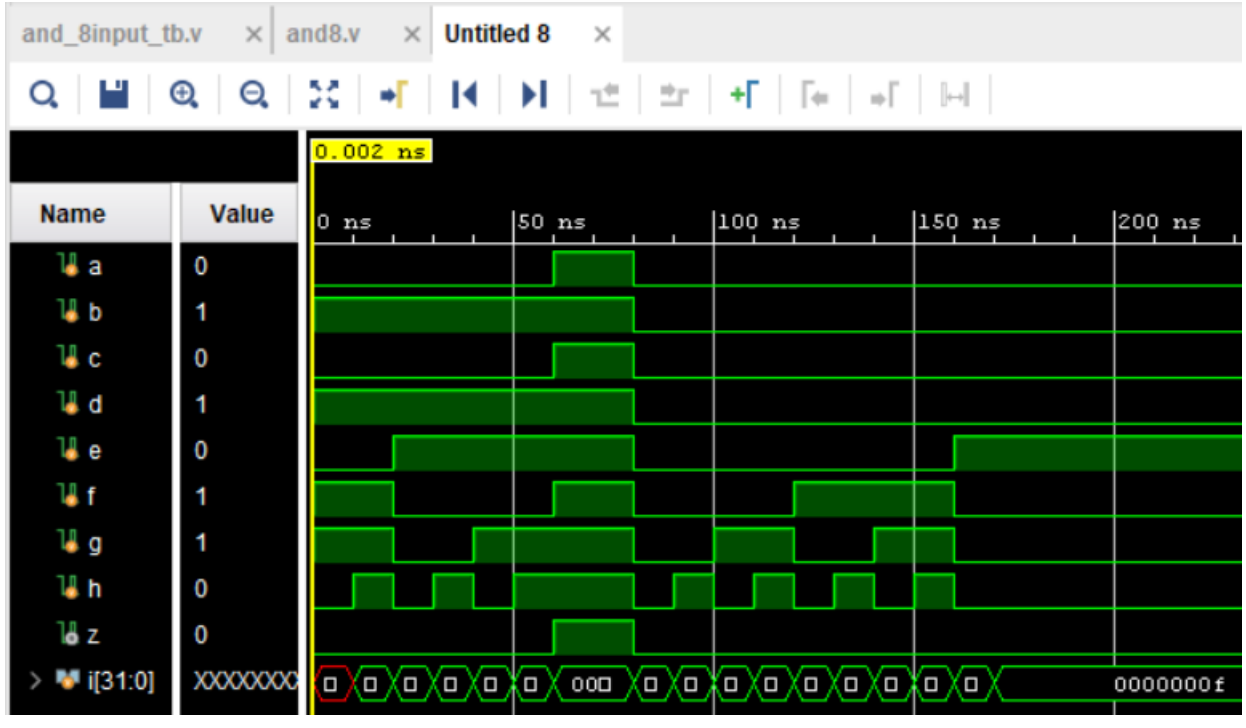
fullAdder_b.v (~\Desktop\Engr433) - GVIM

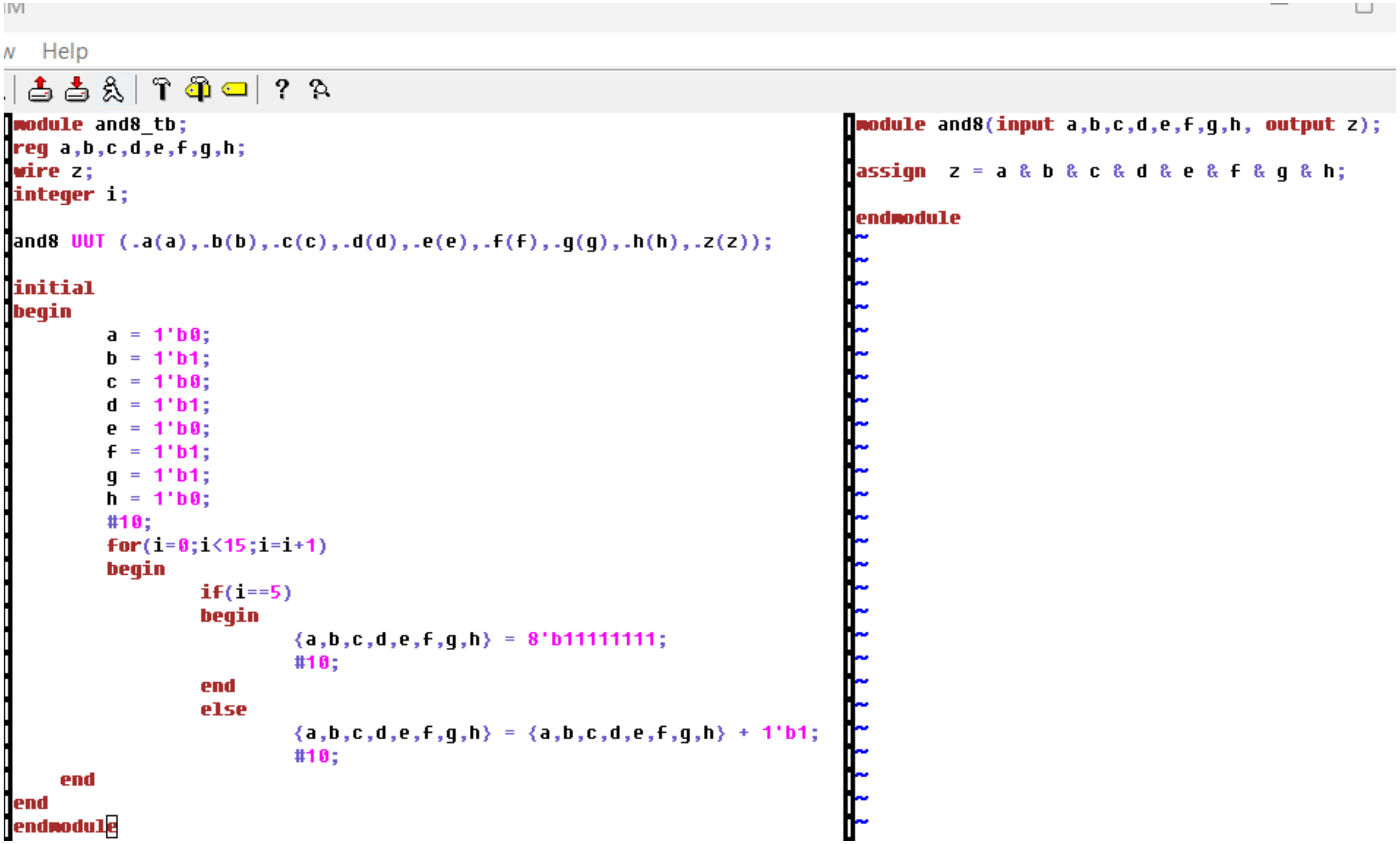
File Edit Tools Syntax Buffers Window Help

```
module oneBitFA(sb,co,x,y,ci);
input x, y, ci;
output sb, co;

assign co = (x&y) | (ci & (x^y));
assign sb = x ^ y ^ ci;

endmodule
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
module oneBitFullAdder(sc,co1,x1,y1,ci1);
input x1, y1, ci1;
output sc, co1;

assign co1 = (x1&y1) | (ci1 & (x1^y1));
assign sc = x1 ^ y1 ^ ci1;

endmodule
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
module twoBitFullAdder_top;
reg a1,a0,b1,b0,cin;
wire s1,s0,carryIn,carryOut;
integer i;

oneBitFA UUT(.x(b0),.y(b1),.ci(cin),.sb(s0),.co(carryIn));
oneBitFullAdder DUT(.x1(a0),.y1(a1),.ci1(carryIn),.sc(s1),.co1(carryOut));

initial begin
        cin = 1'b0;
        a0 = 1'b0;
        b0 = 1'b0;
        a1 = 1'b0;
        b1 = 1'b0;
        #10;
        for(i=0;i<16;i=i+1)
        begin
                {a1,b1,a0,b0} = {a1,b1,a0,b0} + 1'b1;
                #10;
        end
end
endmodule
~
```


and_8input_tb.v
and8.v
Untitled 8
0.002 ns
Name
Value
a 0
b 1
c 0
d 1
e 0
f 1
g 1
h 0
z 0
i[31:0] XXXXXXXX
0 ns
50 ns
100 ns
150 ns
200 ns
0000000f

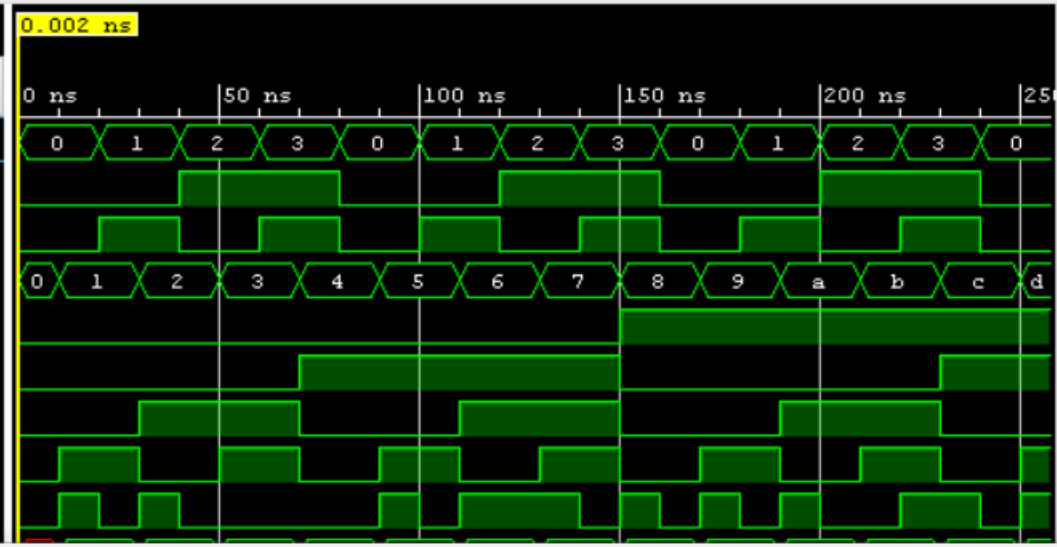


```
module and8_tb;
reg a,b,c,d,e,f,g,h;
wire z;
integer i;

and8 UUT (.a(a),.b(b),.c(c),.d(d),.e(e),.f(f),.g(g),.h(h),.z(z));

initial
begin
        a = 1'b0;
        b = 1'b1;
        c = 1'b0;
        d = 1'b1;
        e = 1'b0;
        f = 1'b1;
        g = 1'b1;
        h = 1'b0;
        #10;
        for(i=0;i<15;i=i+1)
        begin
                if(i==5)
                begin
                        {a,b,c,d,e,f,g,h} = 8'b11111111;
                        #10;
                end
                else
                        {a,b,c,d,e,f,g,h} = {a,b,c,d,e,f,g,h} + 1'b1;
                        #10;
    end
end
endmodule
```

```
module and8(input a,b,c,d,e,f,g,h, output z);

assign  z = a & b & c & d & e & f & g & h;

endmodule
```

four_one_mux.v | four_one_mux_testbench.v | Untitled 9

Name	Value
s[1:0]	0
[1]	0
[0]	0
x[3:0]	0
[3]	0
[2]	0
[1]	0
[0]	0
y	0



four_one_mux_testbench.v + (~\Desktop\Engr433) - GVIM

File Edit Tools Syntax Buffers Window Help

```
module four_one_mux_tb;

reg [1:0] s;
reg [3:0] x;
wire y;
integer i;

four_one_mux UUT(.x(x),.s(s),.y(y));

initial begin
        x = 4'b0000;
        s = 2'b00;
        #10
        for (i=0;i<16;i=i+1)
        begin
                {x} = {x} + 1'b1;
                #10;
                {s} = {s} + 1'b1;
                #10;
        end
end
endmodule
```

```
module four_one_mux(x,y,s);

input [1:0] s;
input [3:0] x;

output y;

assign y = (x[0] & ~s[1] & ~s[0]) | (x[1] & ~s[1] & s[0]) |
           (x[2] & s[1] & ~s[0]) | (x[3] & s[1] & s[0]);

endmodule
~
~
~
~
~
~
~
~
~
~
~
```