

module wire\_demo;

wire net1, net2;

assign net1 = 1'b1;

assign net2 = net1;

initial begin

$display(" net1=%b net2=%b", $time, net1, net2);

end

endmodule

2.module wirew(output wire y, input wire a, b);

assign y = a & b;

endmodule

module regw(output reg y, input reg a,b);

always@\*y=a&b;

endmodule

module tb\_compare\_wire\_reg;

reg a, b;

wire y\_wire;

wire y\_reg;

wirew U1(.y(y\_wire), .a(a), .b(b));

regw U2(.y(y\_reg), .a(a), .b(b));

initial begin

$monitor("%b %b | %b %b", $time, a, b, y\_wire, y\_reg);

a=0; b=0; #5;

a=0; b=1; #5;

a=1; b=0; #5;

a=1; b=1; #5;

end

endmodule

3. module wirew(output wire y,h,i,f,g, input wire a, b);

wand f;

wor g;

tri h;

triand i;

assign y = a & b;

assign y=a|b;

assign h=a;

assign i=b;

endmodule

module tb\_compare\_wire\_reg;

reg a, b;

wire y\_wire;

wire f,i;

wire h;

wire i;

wirew U1(.y(y\_wire), .a(a), .b(b), .h(h),.i(i),.f(f),.g(g));

initial begin

$monitor(" %b %b | %b %b %b %b", $time, a, b, y\_wire,f,g);

a=0; b=0; #5;

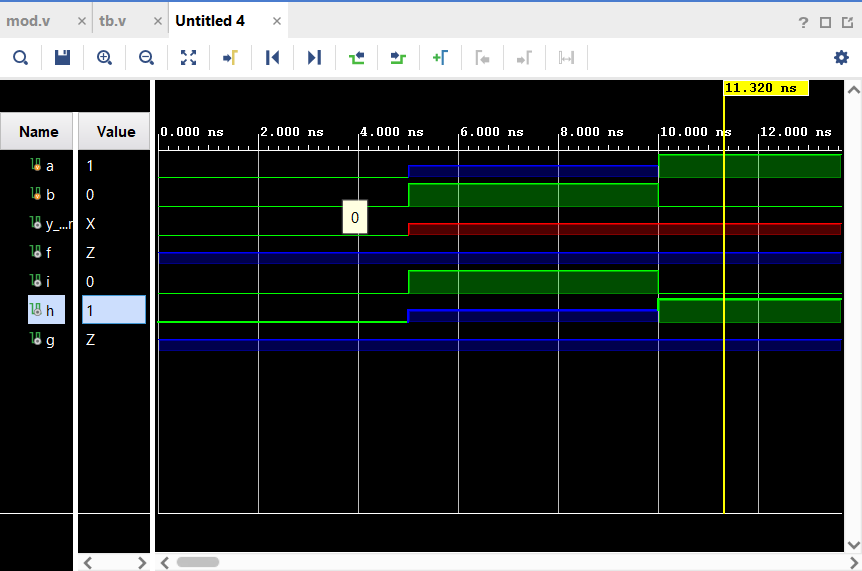
a=1'bz; b=1; #5;

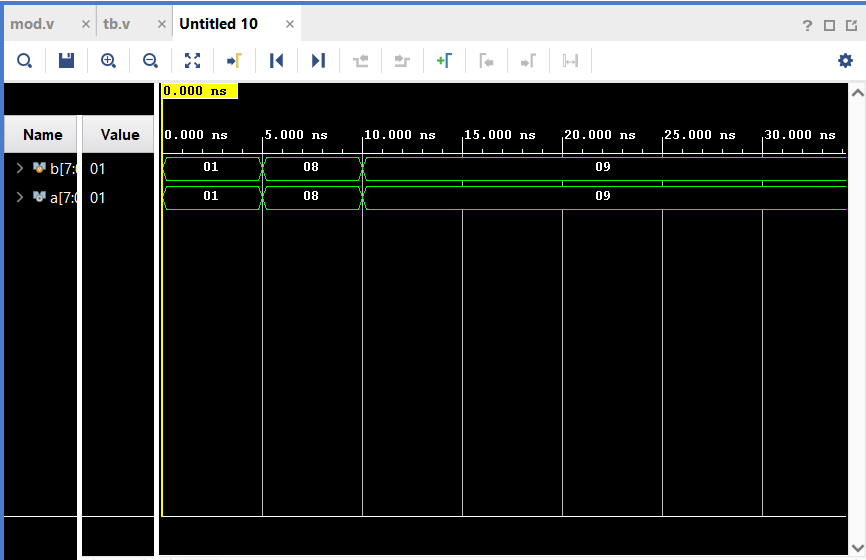
a=1; b=0; #5;

a=1; b=1; #5;

end

endmodule



5. 

module wirew(output reg[7:0] a, input [7:0] b );

always@\* a=b;

endmodule

module tb\_compare\_wire\_reg;

reg [7:0] b;

wire [7:0] a;

wirew U1( .a(a),.b(b));

initial begin

$monitor(" %b %b ", $time, a, b);

end

initial begin

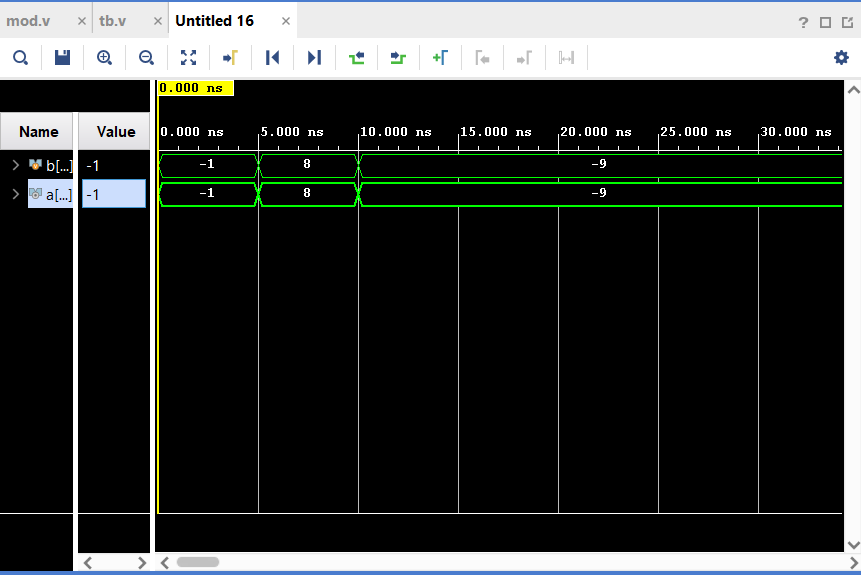
b=1; #5;

b=8; #5;

b=9; #5;

end

endmodule

6. 

module wirew(output integer a ,input signed[31:0] b );

always@\* a=b;

endmodule

module tb\_compare\_wire\_reg;

integer b;

wire signed[31:0] a;

wirew U1( .a(a),.b(b));

initial begin

$monitor(" %d %d ", $time, a, b);

end

initial begin

b=-1; #5;

b=8; #5;

b=-9; #5;

end

endmodule

7. module tb\_compare\_wire\_reg;

/\*real b;

real a;\*/

//wirew U1( .a(a),.b(b));

integer b;

initial begin

$monitor(" %d ", $time, b);

end

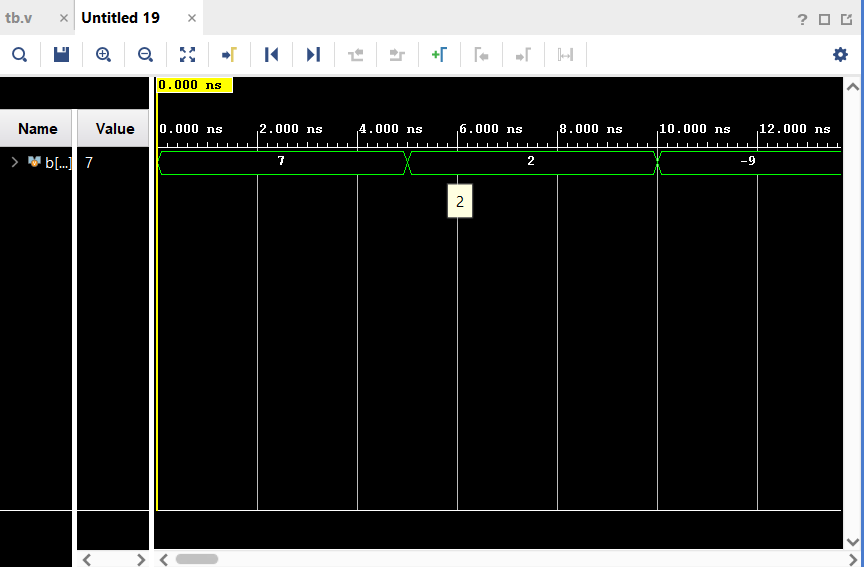
initial begin

b=7.2; #5;

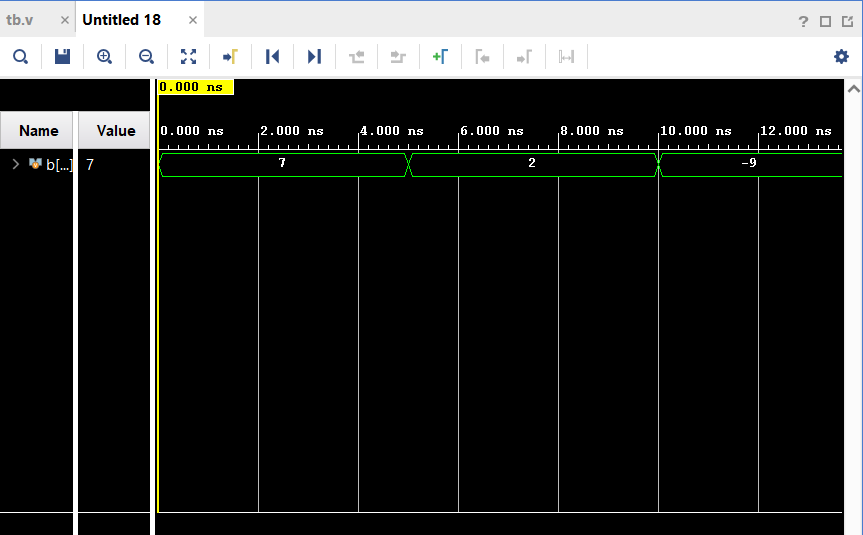
b=1.8; #5;

b=-9; #5;

end

endmodule

8.



module tb\_compare\_wire\_reg;

/\*real b;

real a;\*/

//wirew U1( .a(a),.b(b));

integer b;

initial begin

$monitor(" %d ", $realtime, b);

end

initial begin

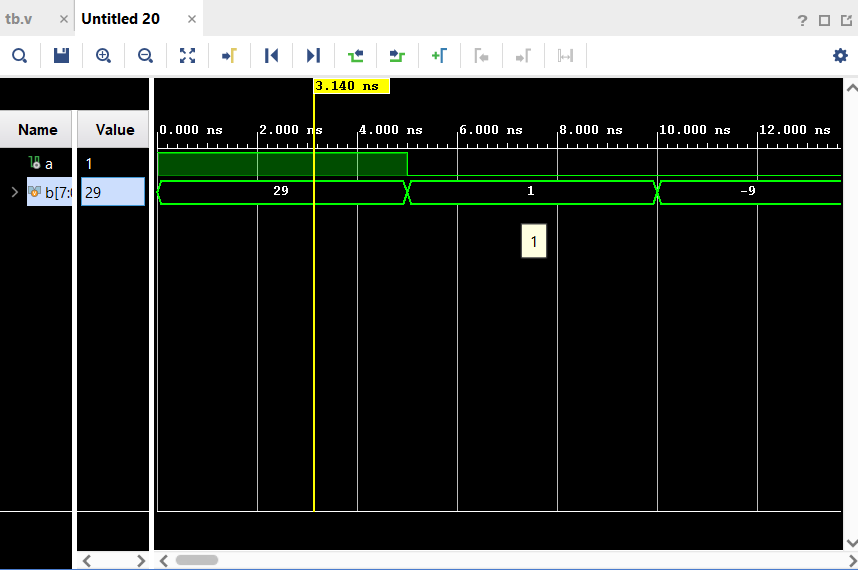
b=7.2; #5;

b=1.8; #5;

b=-9; #5;

end

endmodule

9. 

module tb\_compare\_wire\_reg;

/\*real b;

real a;\*/

wire a;

//wirew U1( .a(a),.b(b));

reg[7:0] b;

assign a= b[3];

initial begin

$monitor(" %d %d ", $realtime, b,a);

end

initial begin

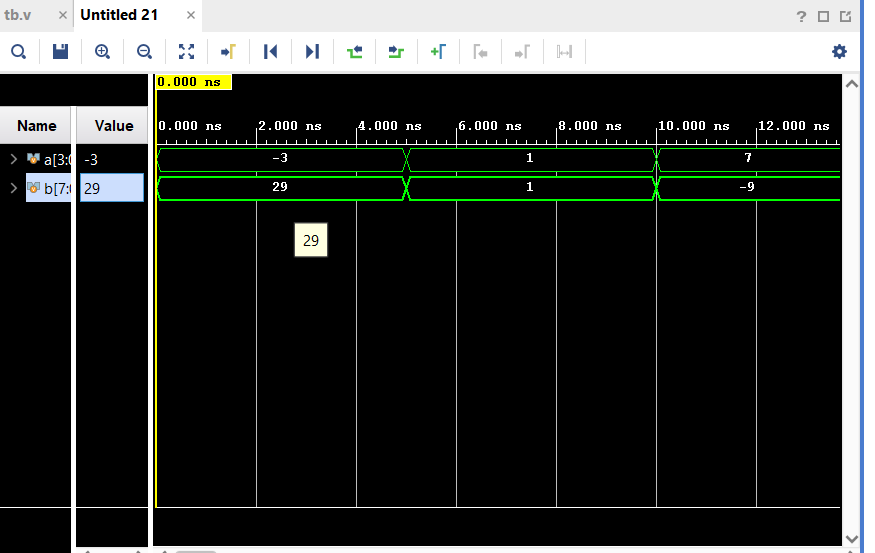
b=8'b00011101; #5;

b=1; #5;

b=-9; #5;

end

endmodule

10. 

module tb\_compare\_wire\_reg;

/\*real b;

real a;\*/

reg [3:0]a;

//wirew U1( .a(a),.b(b));

reg[7:0] b;

always@\*

assign a= b[3:0];

initial begin

$monitor(" %d %d ", $realtime, b,a);

end

initial begin

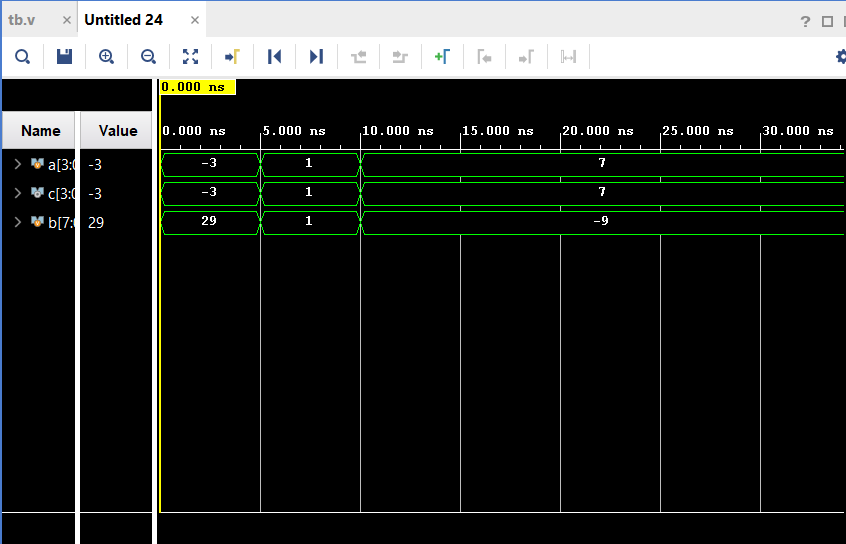
b=8'b00011101; #5;

b=1; #5;

b=-9; #5;

end

endmodule

11. 

module tb\_compare\_wire\_reg;

/\*real b;

real a;\*/

reg [3:0]a;

wire[3:0] c;

//wirew U1( .a(a),.b(b));

reg[7:0] b;

always@\*

assign a= b[3:0];

assign c[3]=b[3];

assign c[2:0]=b[2:0];

initial begin

$monitor(" %d %d %b ", $realtime, b,a,c);

end

initial begin

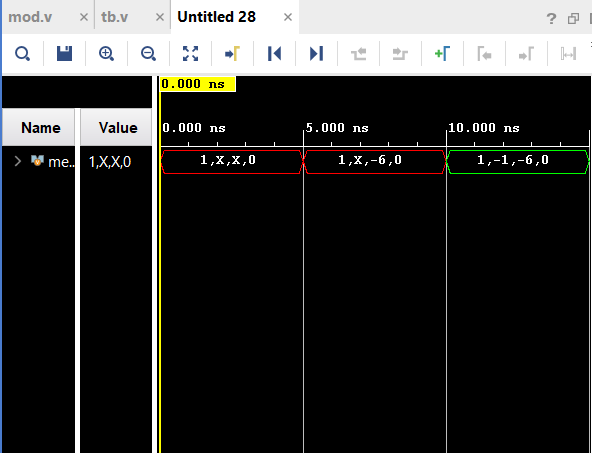
b=8'b00011101; #5;

b=1; #5;

b=-9; #5;

end

endmodule

12. 

module wirew;

reg [3:0] mem [3:0];

initial begin

mem[3] = 4'b0001;

mem[0] = 4'b0000;

$display("t=%0t mem[3]=%b mem[0]=%b", $time, mem[3], mem[0]);

#5 mem[1] = 4'b1010;

$display("t=%0t mem[1]=%b", $time, mem[1]);

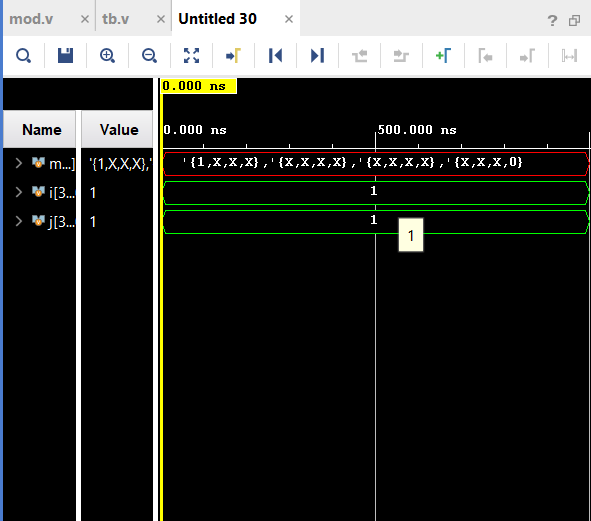
#5 mem[2] = 4'b1111;

$display("t=%0t mem[2]=%b", $time, mem[2]);

#5 $finish;

end

endmodule

13. 

module wirew;

reg [3:0] mem [3:0][3:0];

integer i;

integer j;

initial begin

mem[3][3] = 4'b0001;

for(i=0;i<1;i=i+1)begin

for(j=0;j<1;j=j+1)begin

mem[i][j]=4'b0000;

end

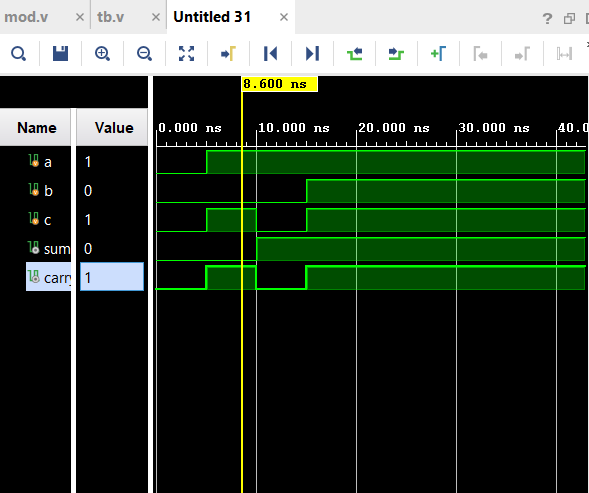
end

$display("t=%0t mem[3][3]=%b ", $time, mem[3][3]);

$display("t=%0t mem[0][0]=%b", $time, mem[0][0]);

end

endmodule

14. 

module fulladder(a,b,c,sum,carry);

input a,b,c;

output sum,carry;

wire x,y,z,d;

xor(d,a,b);

xor(sum,d,c);

and(x,a,b);

and(y,b,c);

and(z,c,a);

or(carry,x,y,z);

endmodule

module ha\_tb;

reg a,b,c;

wire sum,carry;

fulladder uut(a,b,c,sum,carry);

initial begin

a=0;b=0;c=0;

#5 a=1;b=0;c=1;

#5 a=1;b=0;c=0;

#5 a=1;b=1;c=1;

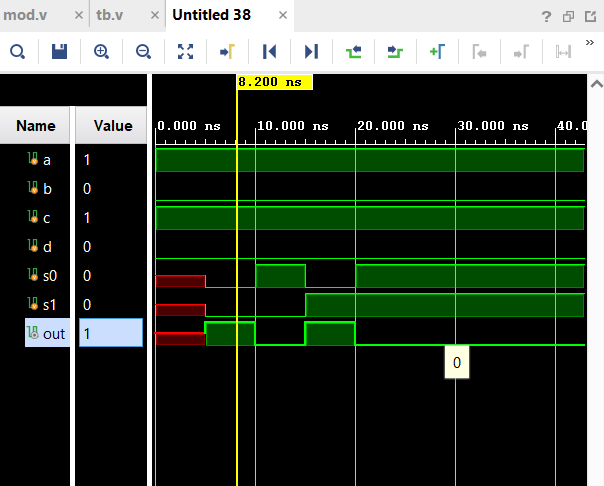
end

initial begin

$monitor($time,"a=%b b=%b c=%b sum=%b carry=%c",a,b,c,sum,carry);

end

endmodule

15. 

module fulladder(a,b,c,d,s0,s1,out);

input a,b,c,d,s0,s1;

output out;

wire x,y,z,e,s1b,s0b;

not(s1b,s1);

not(s0b,s0);

and(x,s0b,s1b,a);

and(y,s0,s1b,b);

and(z,s0b,s1,c);

and(e,s0,s1,d);

or(out,x,y,z,e);

endmodule

module ha\_tb;

reg a,b,c,d,s0,s1;

wire out;

fulladder uut(a,b,c,d,s0,s1,out);

initial begin

a=1;b=0;c=1;d=0;

#5 s0=0;s1=0;

#5 s0=1;s1=0;

#5 s0=0;s1=1;

#5 s0=1;s1=1;

end

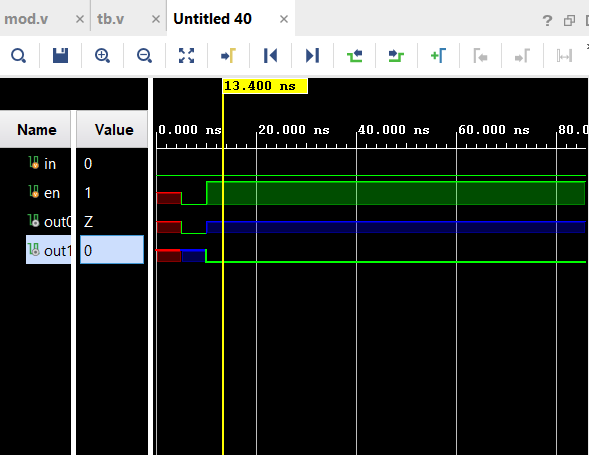
initial begin

$monitor($time,"a=%b b=%b c=%b d=%b s0=%b s1=%b",a,b,c,d,s0,s1,out);

end

endmodule

16.



module ha(in,out0,out1,en);

input in,en;

output out0,out1;

bufif0(out0,in,en);

bufif1(out1,in,en);

endmodule

module ha\_tb;

reg in,en;

wire out0,out1;

ha uut(in,out0,out1,en);

initial begin

in=0;

#5 en=0;

#5 en=1;

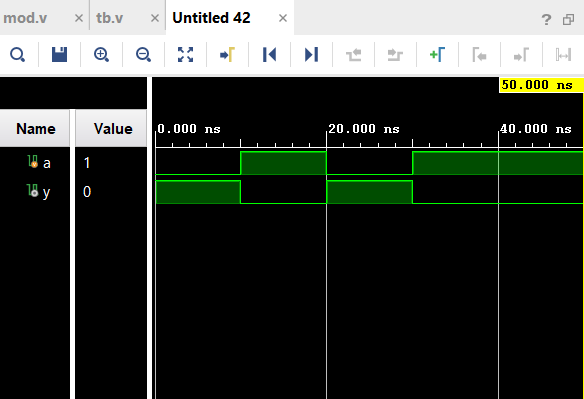
end

initial begin

$monitor($time,"in=%b out0=%b out1=%b en=%b ",in,out0,out1,en);

end

endmodule

17. 

module cmos\_inverter (input a,output y );

supply1 vdd; supply0 gnd;

pmos p1 (y, vdd, a);

nmos n1 (y, gnd, a);

endmodule

module tb\_cmos\_inverter;

reg a;

wire y;

cmos\_inverter uut (.a(a), .y(y));

initial begin

$monitor("t=%0t a=%b -> y=%b", $time, a, y);

a = 0; #10;

a = 1; #10;

a = 0; #10;

a = 1; #10;

#10 $finish;

end

endmodule

18. module nand\_gate\_cmos (

input A, B,

output Y

);

supply1 Vdd;

supply0 Gnd;

wire w;

pmos (Y, Vdd, A);

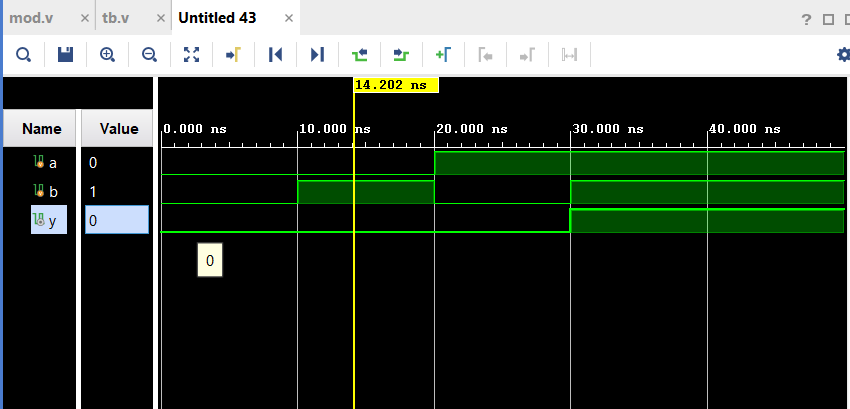
pmos (Y, w, B);

nmos (Y, Gnd, A);

nmos (Y, Gnd, B);

endmodule

19.



primitive udp\_and2 (y, a, b);

output y;

input a, b;

table

// a b : y

0 0 : 0;

0 1 : 0;

1 0 : 0;

1 1 : 1;

endtable

endprimitive

module tb\_cmos\_inverter;

reg a,b;

wire y;

udp\_and2 uut(y,a,b);

initial begin

$monitor("t=%0t a=%b b=%b y=%b", $time, a,b, y);

a = 0; b=0; #10;

a = 0; b=1;#10;

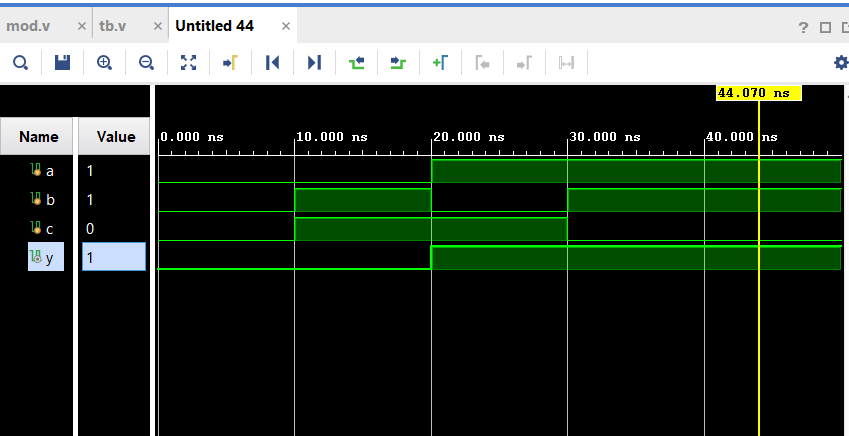
a = 1; b=0;#10;

a = 1; b=1; #10;

#10 $finish;

end

20.



primitive udp\_and2 (y, a, b,c);

output y;

input a, b,c;

table

// a b c : y

0 0 0: 0;

0 0 1: 0;

0 1 0: 1;

0 1 1: 0;

1 0 0: 1;

1 0 1: 1;

1 1 0: 1;

1 1 1: 1;// x+yz`

endtable

endprimitive

module tb\_cmos\_inverter;

reg a,b,c;

wire y;

udp\_and2 uut(y,a,b,c);

initial begin

$monitor("t=%0t a=%b b=%b c=%b y=%b", $time, a,b,c, y);

a = 0; b=0;c=0; #10;

a = 0; b=1;c=1;#10;

a = 1; b=0;c=1;#10;

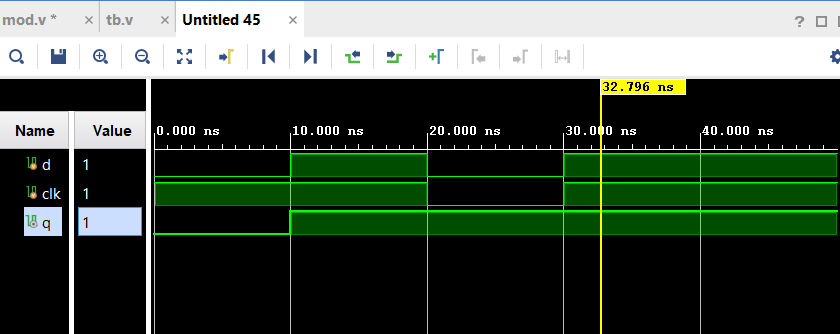
a = 1; b=1;c=0; #10;

#10 $finish;

end

endmodule

21.



primitive udp\_and2 (q, d, clk);

output reg q;

input t, clk;

table

// d clk : q : qnext

0 1 : ? : 0;

1 1 : ? : 1;

? 0 : ? : -;

? x : ? : -;

endtable

endprimitive

module tb\_cmos\_inverter;

reg d,clk;

wire q;

udp\_and2 uut(q, d, clk);

initial begin

$monitor("t=%0t q=%b d=%b clk=%b ", $time, q, d, clk);

d = 0; clk=1; #10;

d = 1; clk=1;#10;

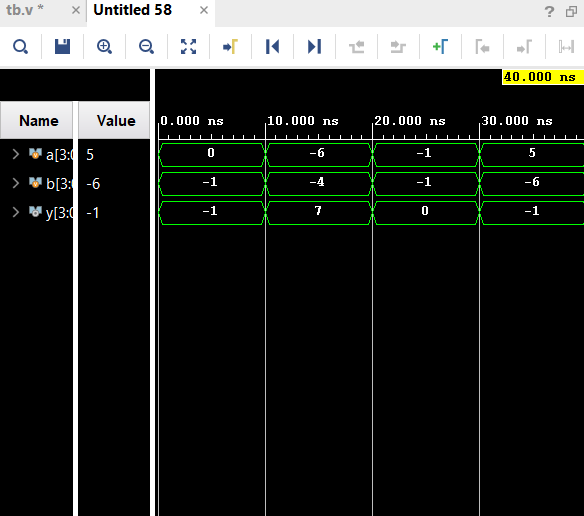
d = 0; clk=0;#10;

d = 1; clk=1; #10;

#10 $finish;

end

endmodule

23. 

**module array;**

**reg [3:0] a, b;**

**wire[3:0] y;**

**nand n1 [3:0] (y, a, b);**

**initial begin**

**a = 4'b0000; b = 4'b1111; #10;**

**a = 4'b1010; b = 4'b1100; #10;**

**a = 4'b1111; b = 4'b1111; #10;**

**a = 4'b0101; b = 4'b1010; #10;**

**$finish;**

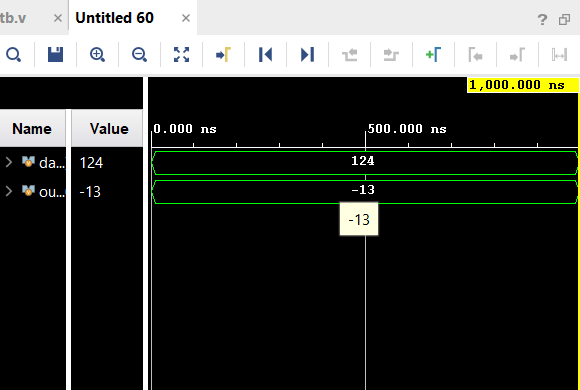
**end**

**initial begin**

**$monitor($time, " a=%b b=%b y=%b", a, b, y);**

**end**

**endmodule**

**24.** ****

**module data1;**

**reg [7:0] data;**

**reg [4:0] out;**

**initial begin**

**data = 124;**

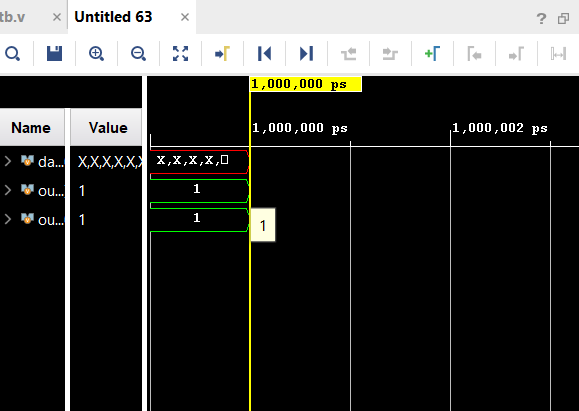
**out = data[7:4] + data[3:0];**

**$display("out = %d , upper = %b, lower = %b",**

**out, out, data[7:4], data[3:0]);**

**end**

**endmodule**

****

**module data1;**

**reg [7:0] data[7:0];**

**reg [7:0] out1,out;**

**initial begin**

**data[4][7:4]=4'b0001;**

**data [5][3]=1'b1;**

**out1=data[4][7:4];**

**out = data[5][3];**

**$display("out1 = %d , out=%d",**

**out1, out);**

**end**

**endmodule**