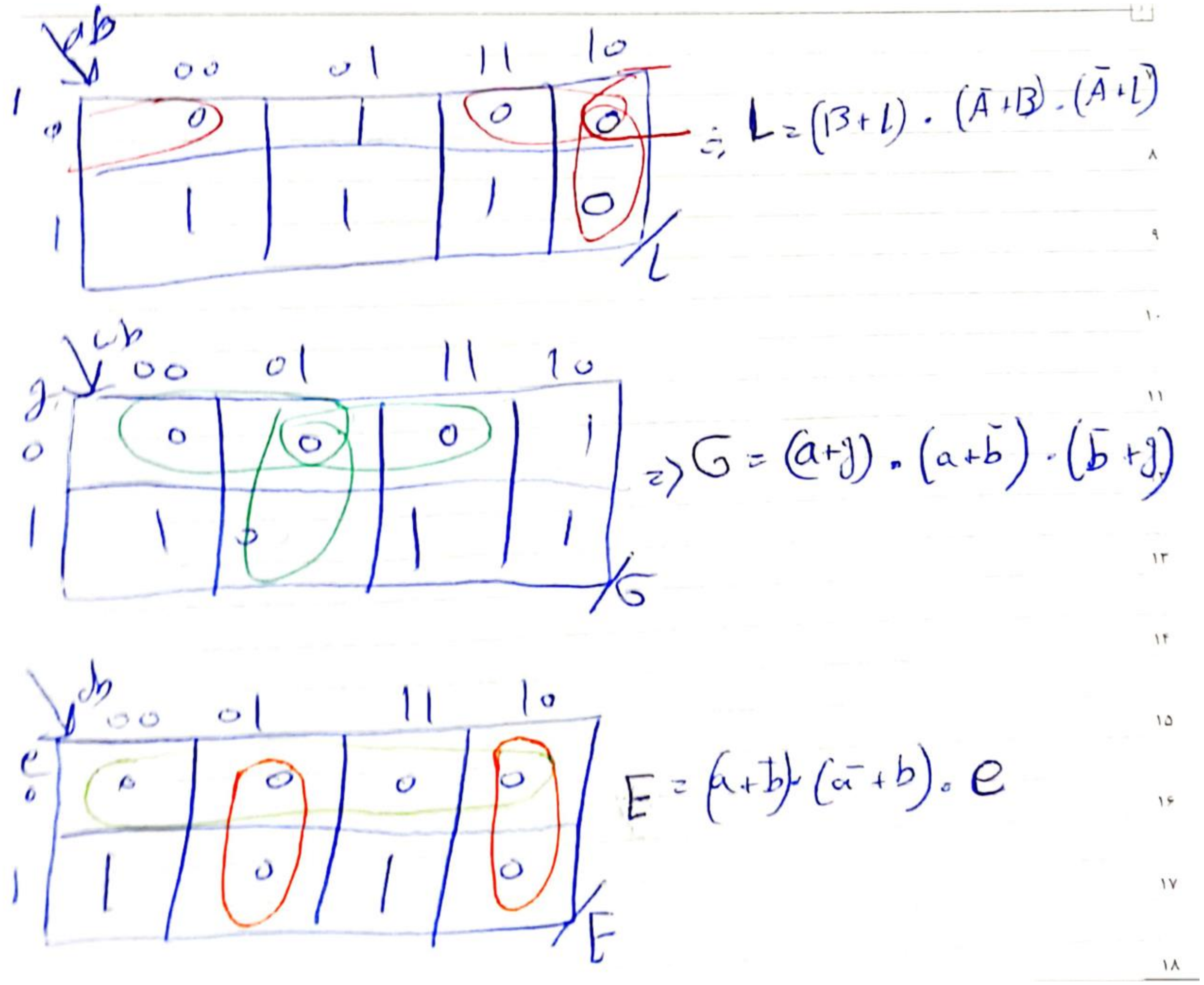
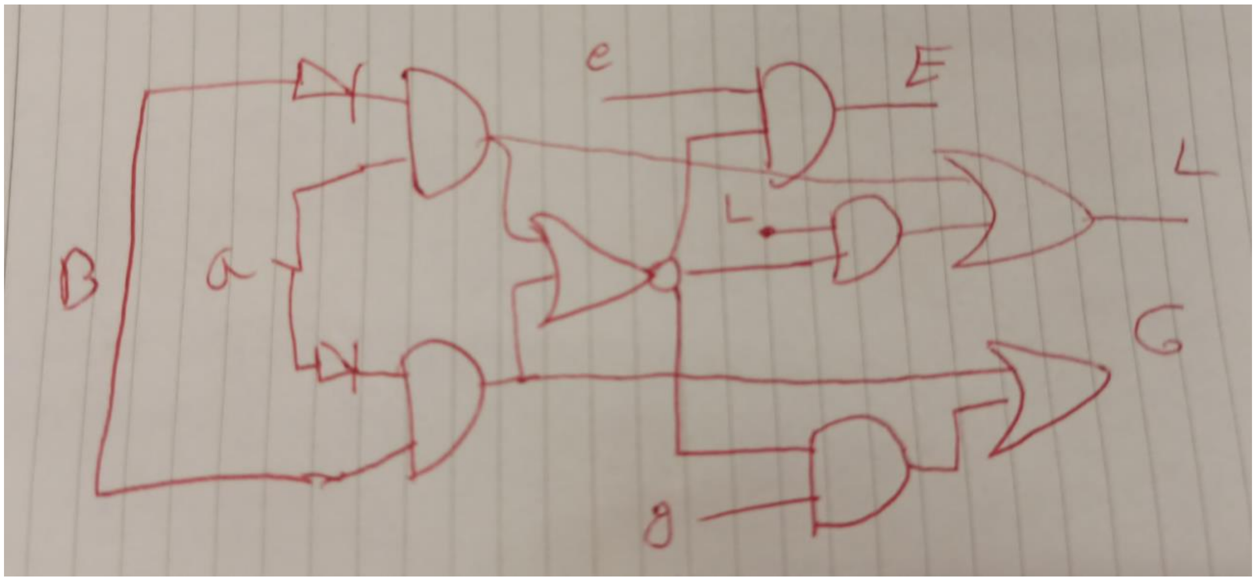
1-a

kaurna map :

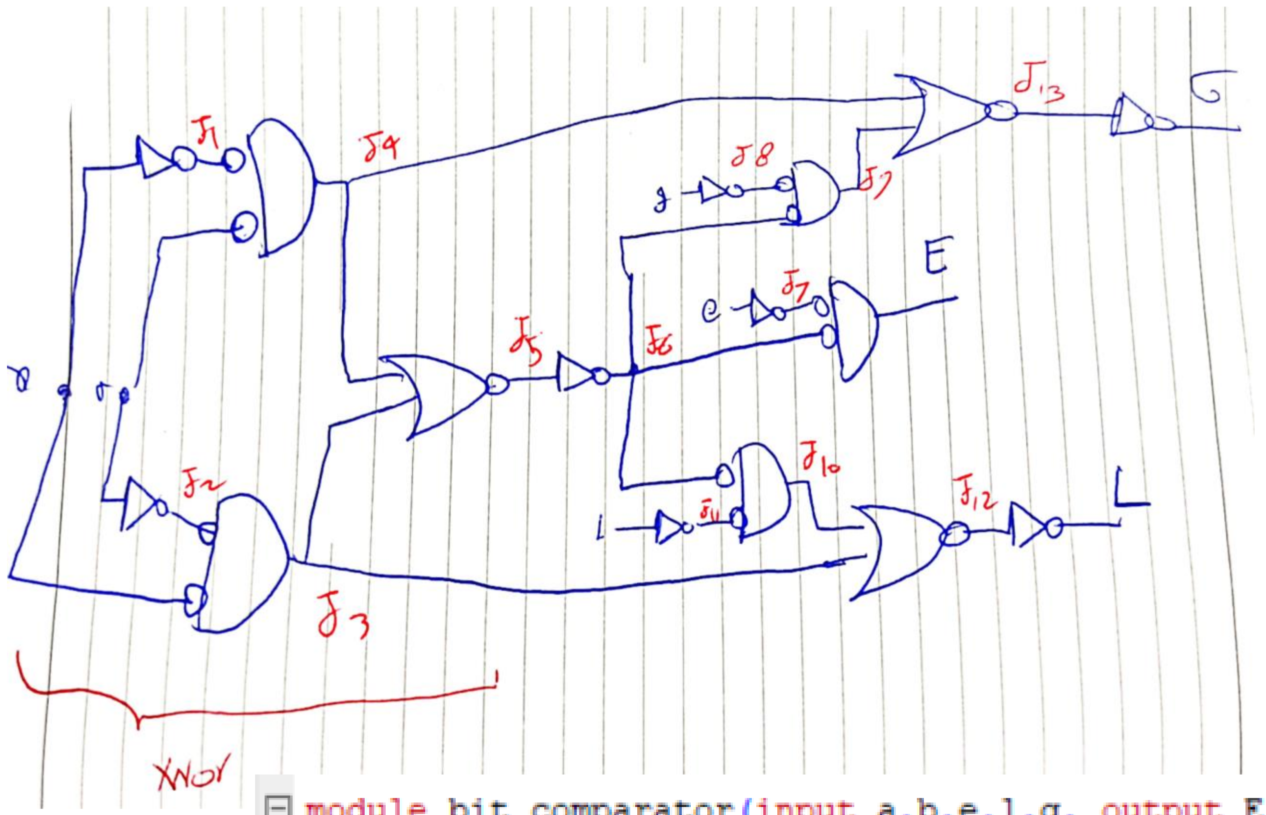


00 01 11 10
$L = (B+L)\cdot(\bar{A}+B)\cdot(\bar{A}+\bar{L})$
$G = (a+g)\cdot(a+\bar{b})\cdot(\bar{b}+g)$
$E = (a+b)\cdot(\bar{a}+b)\cdot e$
L
G
E

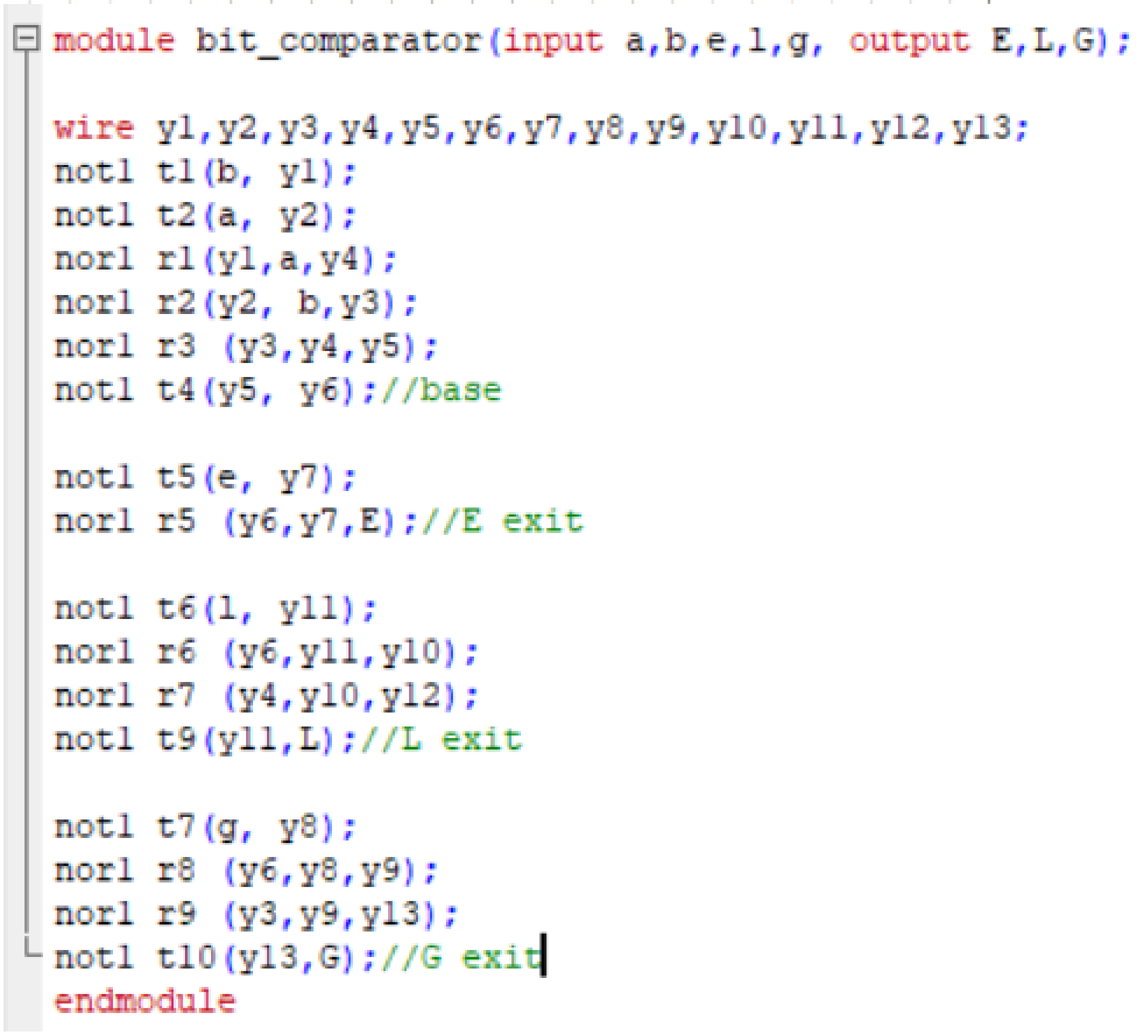
gate level:



Gate level with 2 input nor and inveter from CA1 :



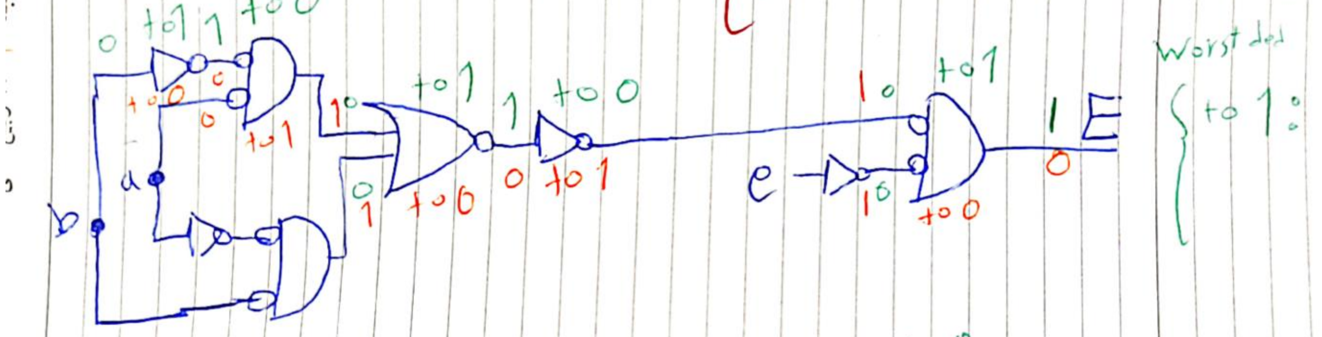
Verilog Code :

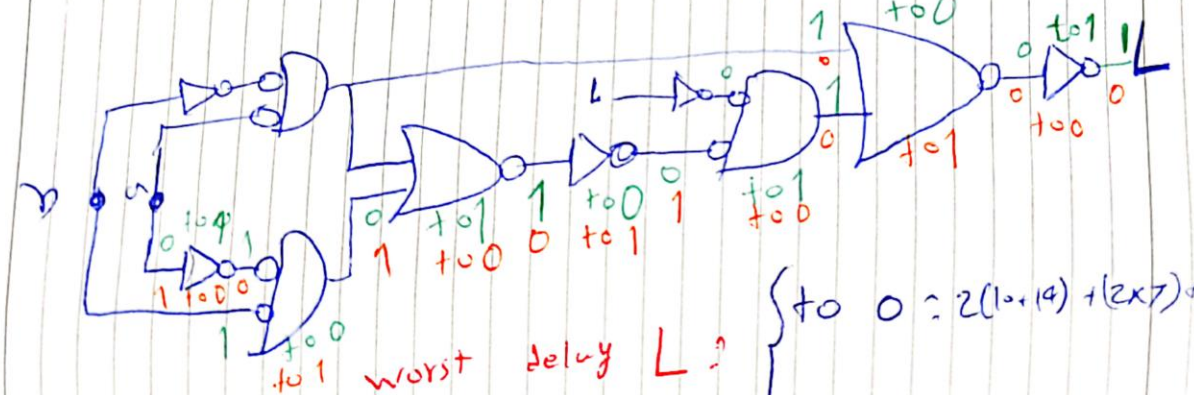


```
module bit_comparator(input a,b,e,l,g, output E,L,G);

wire y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13;
not1 t1(b, y1);
not1 t2(a, y2);
nor1 r1(y1,a,y4);
nor1 r2(y2, b,y3);
nor1 r3 (y3,y4,y5);
not1 t4(y5, y6);//base

not1 t5(e, y7);
nor1 r5 (y6,y7,E);//E exit

not1 t6(l, y11);
nor1 r6 (y6,y11,y10);
nor1 r7 (y4,y10,y12);
not1 t9(y11,L);//L exit

not1 t7(g, y8);
nor1 r8 (y6,y8,y9);
nor1 r9 (y3,y9,y13);
not1 t10(y13,G);//G exit
endmodule
```

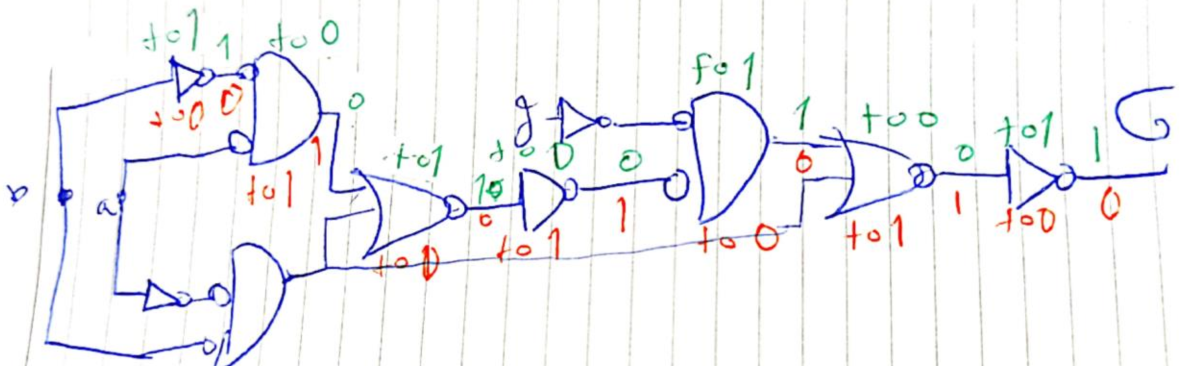
1-b-caculating worst case delay

worst delay E
$$\begin{cases} \text{to } 0: 1\times10 + 2\times14 + 5 + 7 = \boxed{60ns} & 31ns \\ \text{to } 1: 2\times10 + 1\times14 + 5 + 7 = \boxed{46ns} & 29n \end{cases}$$



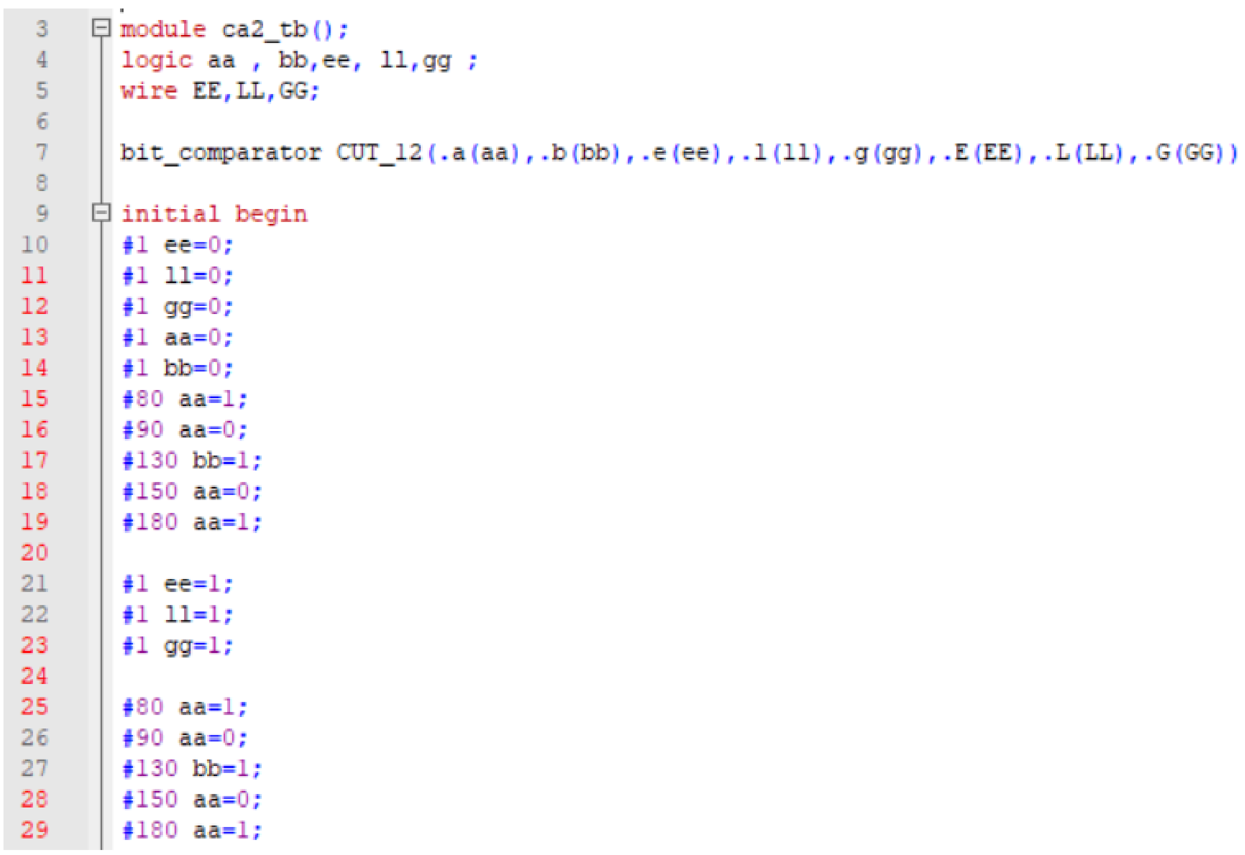


worst delay L :
$$\begin{cases} \text{to } 0: 2(10+14) + (2\times7) + 5 = \boxed{67ns} \\ \text{to } 1: \underbrace{2(10+14)} + 2\times5 + 7 = \boxed{65ns} \end{cases}$$

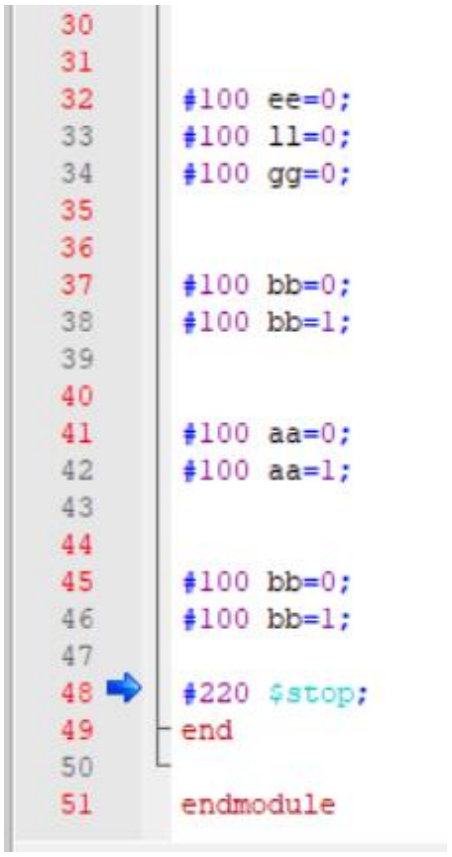


worst case delay G
$$\begin{cases} \text{to } 1: 2(10+14) + (2\times5) + 7 = 65ns \\ \text{to } 0: 2(10+14) + (2\times7) + 5 = 67ns \end{cases}$$

1-c-test bench:



```
module ca2_tb();
logic aa , bb,ee, ll,gg ;
wire EE,LL,GG;

bit_comparator CUT_12(.a(aa),.b(bb),.e(ee),.l(ll),.g(gg),.E(EE),.L(LL),.G(GG))

initial begin
#1 ee=0;
#1 ll=0;
#1 gg=0;
#1 aa=0;
#1 bb=0;
#80 aa=1;
#90 aa=0;
#130 bb=1;
#150 aa=0;
#180 aa=1;

#1 ee=1;
#1 ll=1;
#1 gg=1;

#80 aa=1;
#90 aa=0;
#130 bb=1;
#150 aa=0;
#180 aa=1;
```



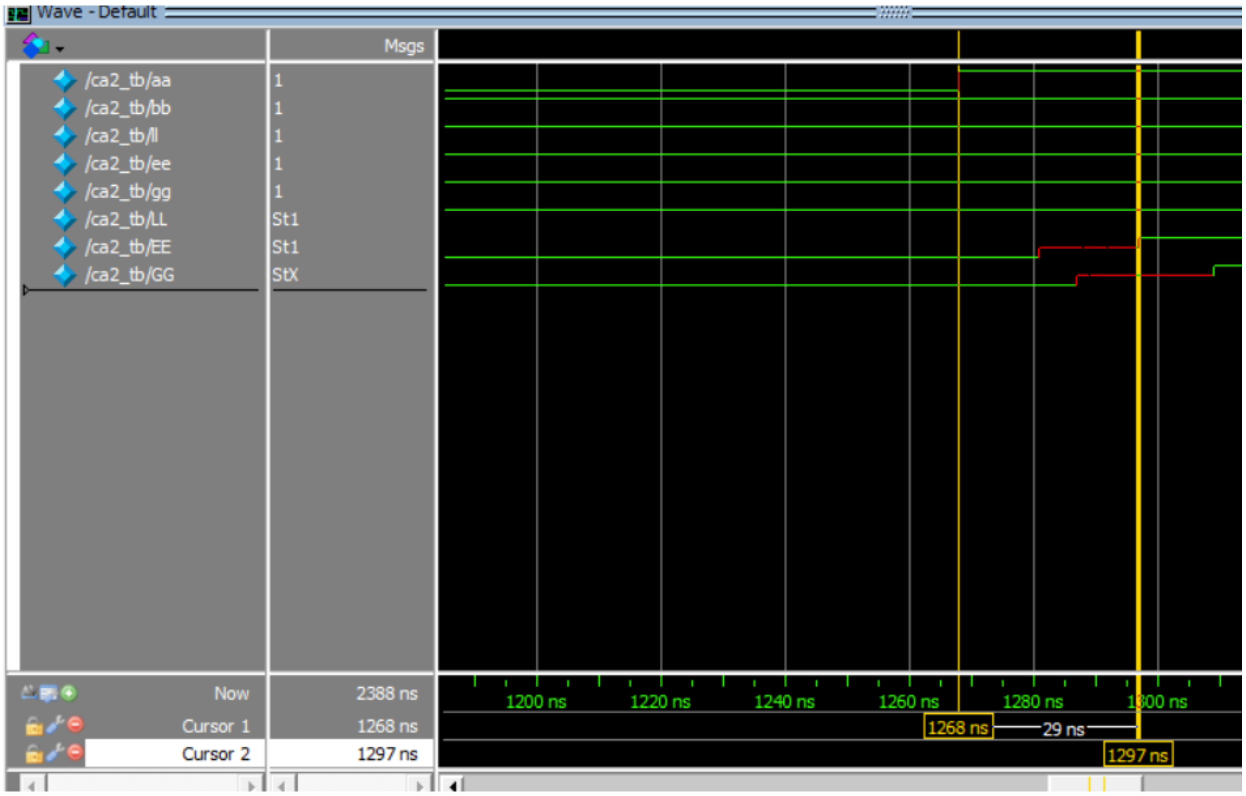
```


#100 ee=0;
#100 ll=0;
#100 gg=0;


#100 bb=0;
#100 bb=1;


#100 aa=0;
#100 aa=1;


#100 bb=0;
#100 bb=1;

#220 $stop;
end

endmodule
```

Waves worst case delay:

E_output:

To 1:

29ns

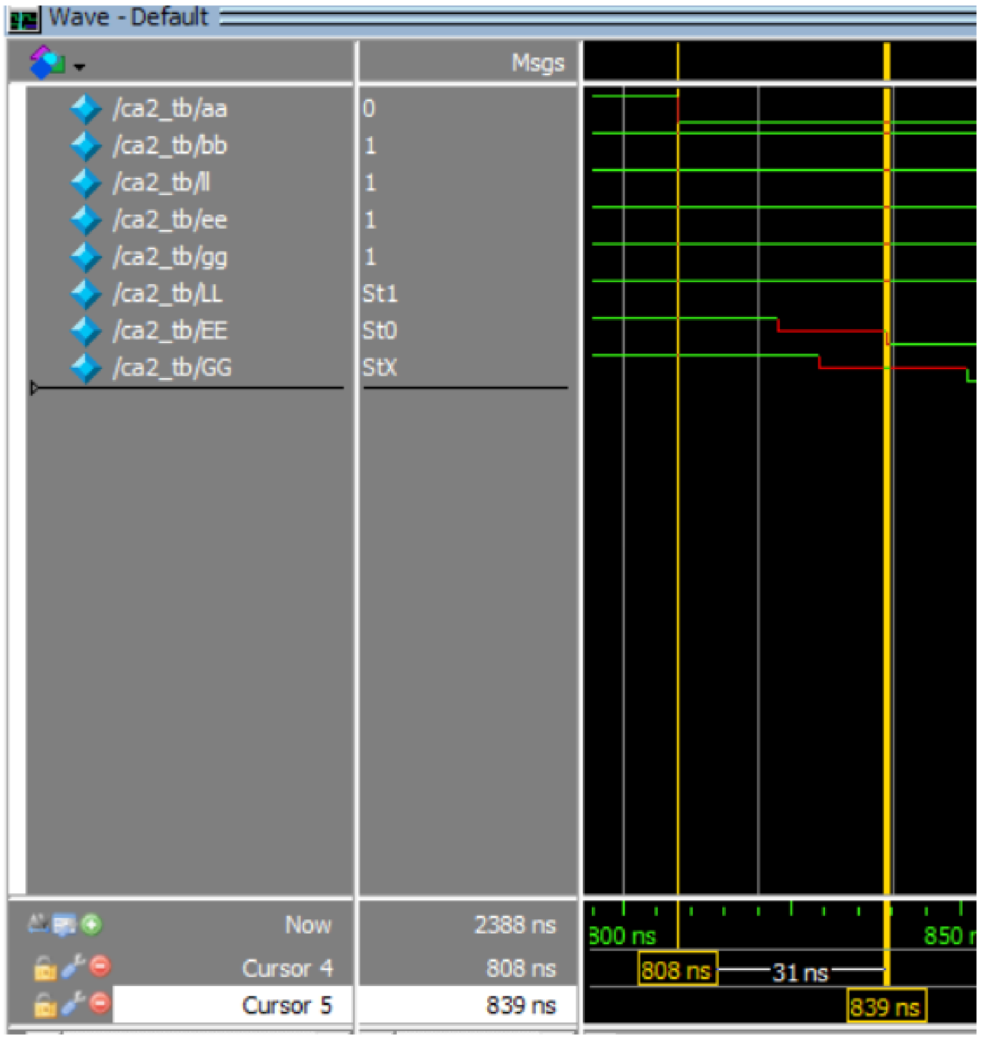


Wave - Default

	Msgs
/ca2_tb/aa	1
/ca2_tb/bb	1
/ca2_tb/ll	1
/ca2_tb/ee	1
/ca2_tb/gg	1
/ca2_tb/LL	St1
/ca2_tb/EE	St1
/ca2_tb/GG	StX
Now	2388 ns
Cursor 1	1268 ns
Cursor 2	1297 ns

To 0 :

31ns



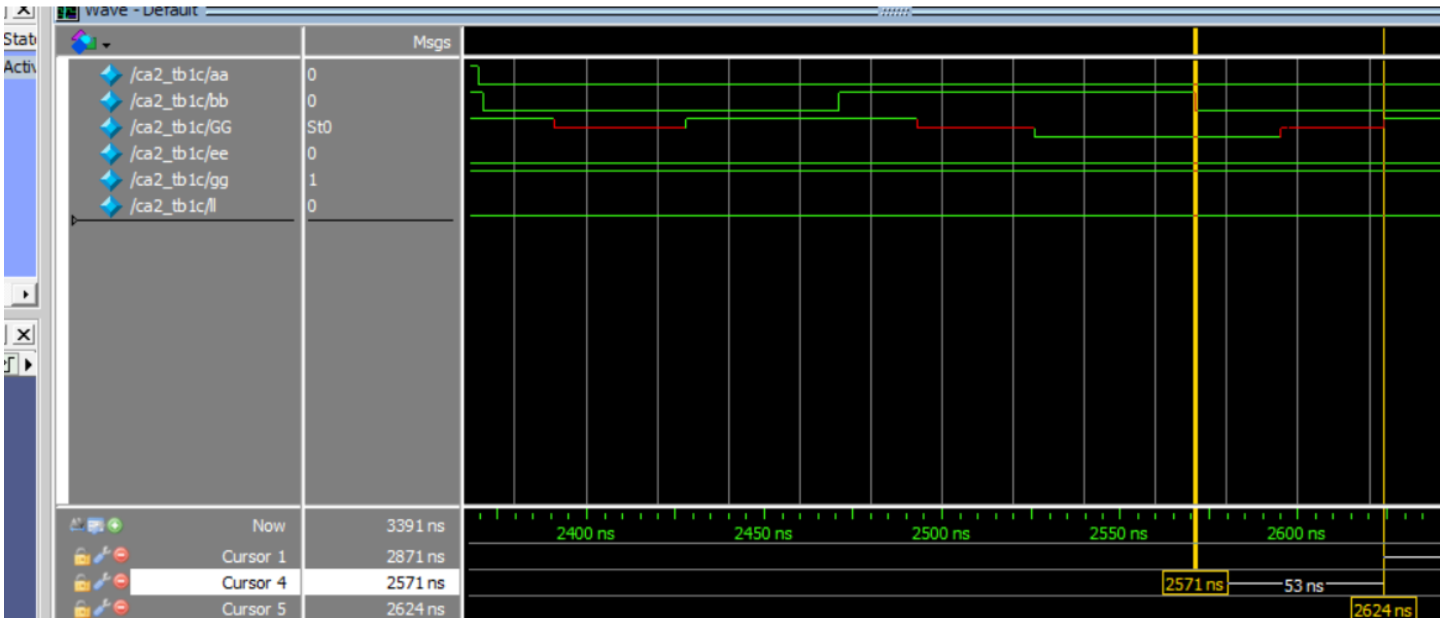
Wave - Default

	Msgs
/ca2_tb/aa	0
/ca2_tb/bb	1
/ca2_tb/ll	1
/ca2_tb/ee	1
/ca2_tb/gg	1
/ca2_tb/LL	St1
/ca2_tb/EE	St0
/ca2_tb/GG	StX
Now	2388 ns
Cursor 4	808 ns
Cursor 5	839 ns

G_output:

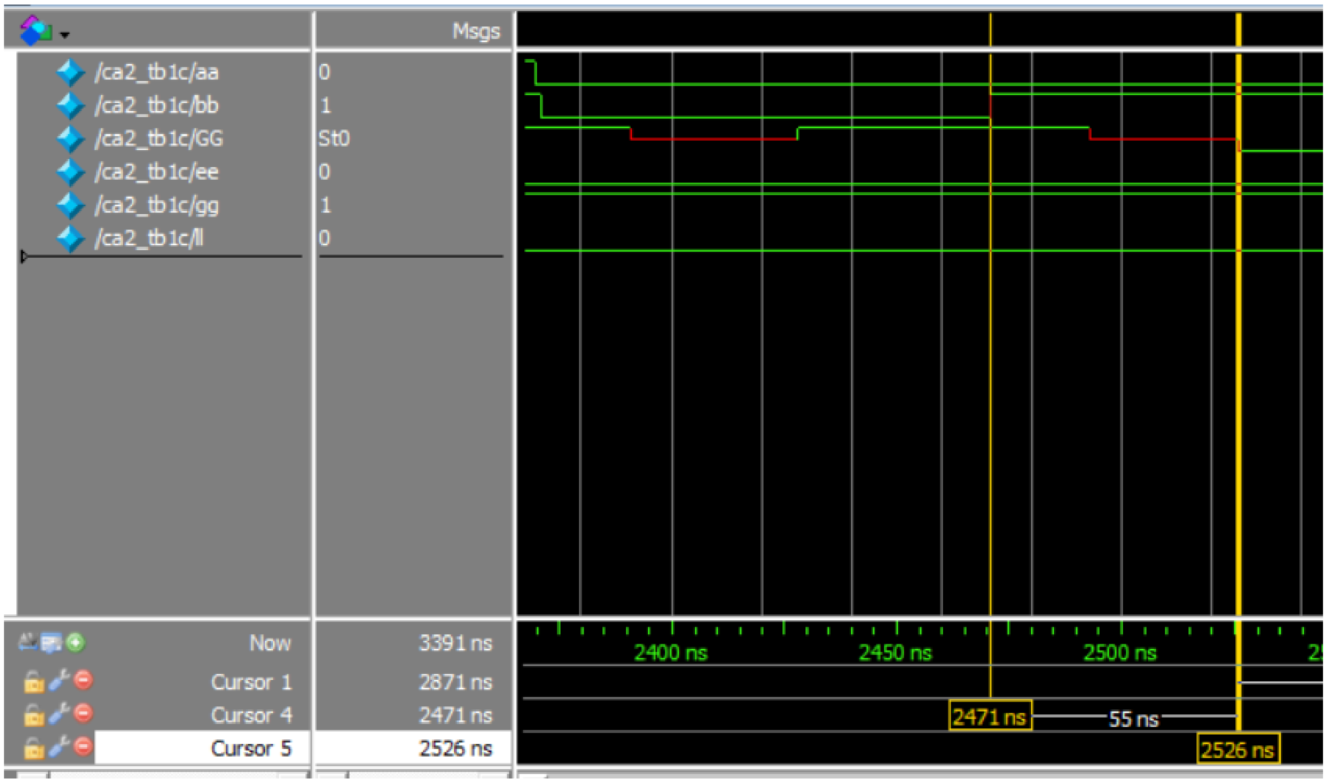
To 1:

53ns



To 0 :

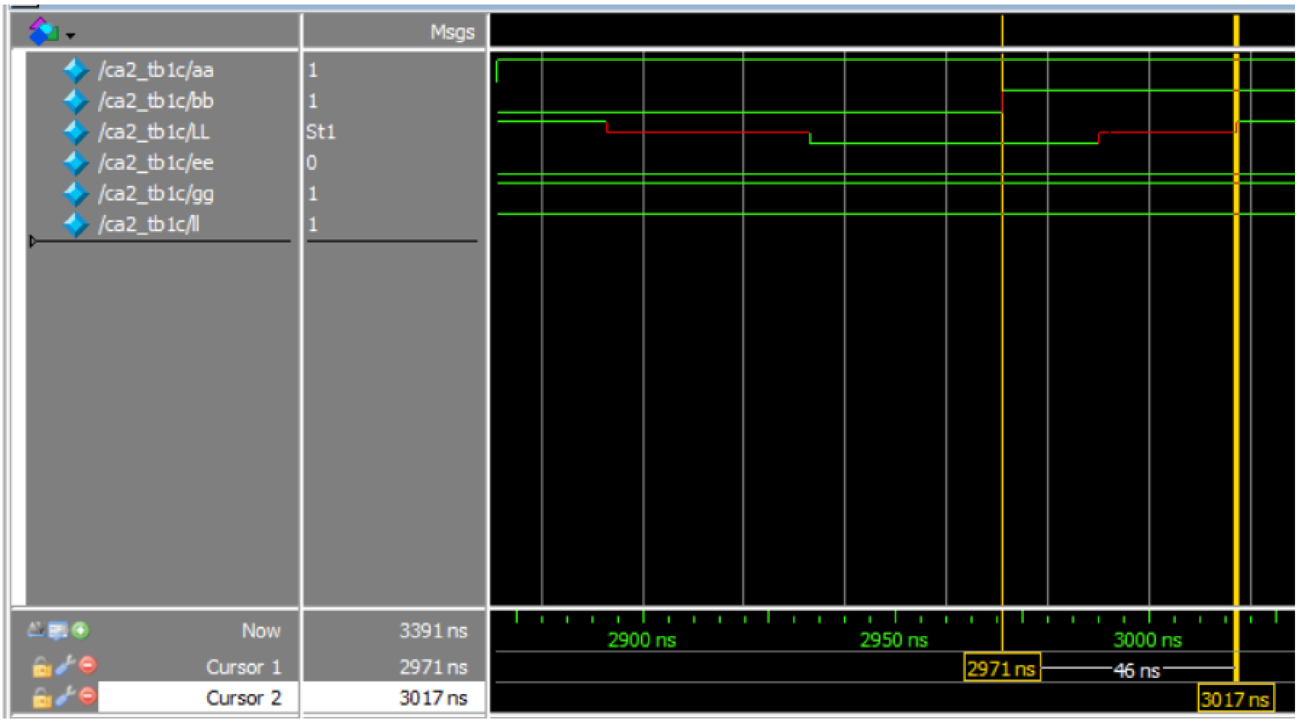
75ns



L_output:

To 1:

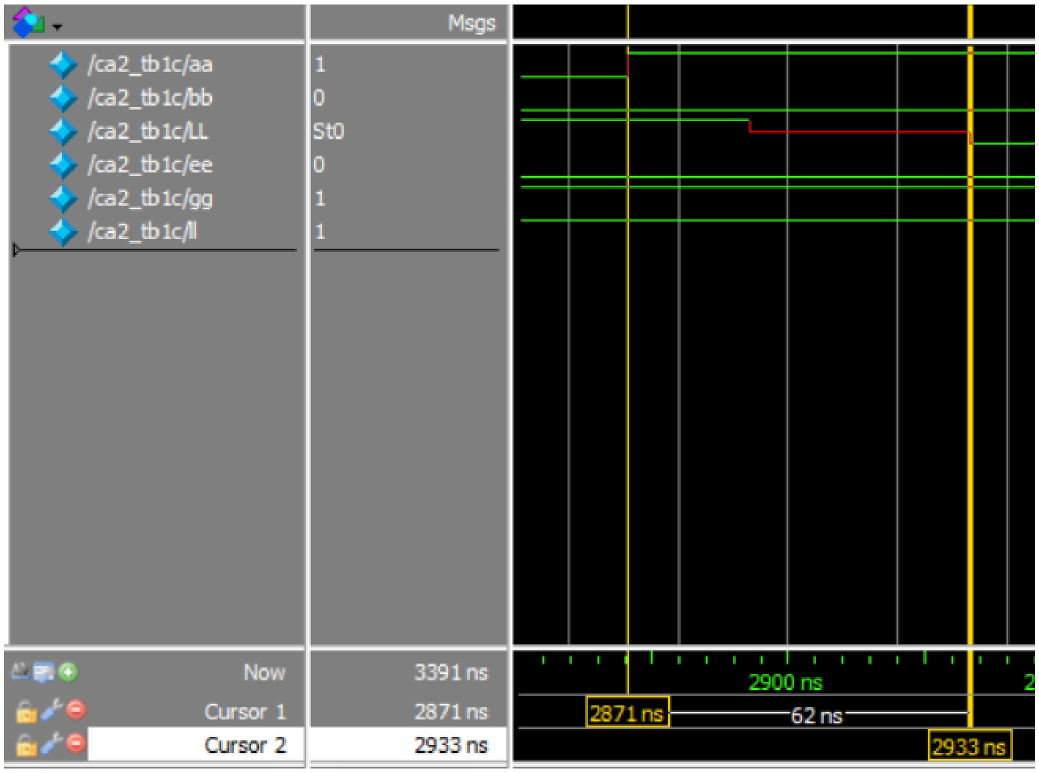
46ns



```
/ca2_tb1c/aa   1
/ca2_tb1c/bb   1
/ca2_tb1c/LL   St1
/ca2_tb1c/ee   0
/ca2_tb1c/gg   1
/ca2_tb1c/ll   1
Now            3391 ns
Cursor 1       2971 ns
Cursor 2       3017 ns
2971 ns  46 ns  3017 ns
```

To 0 :

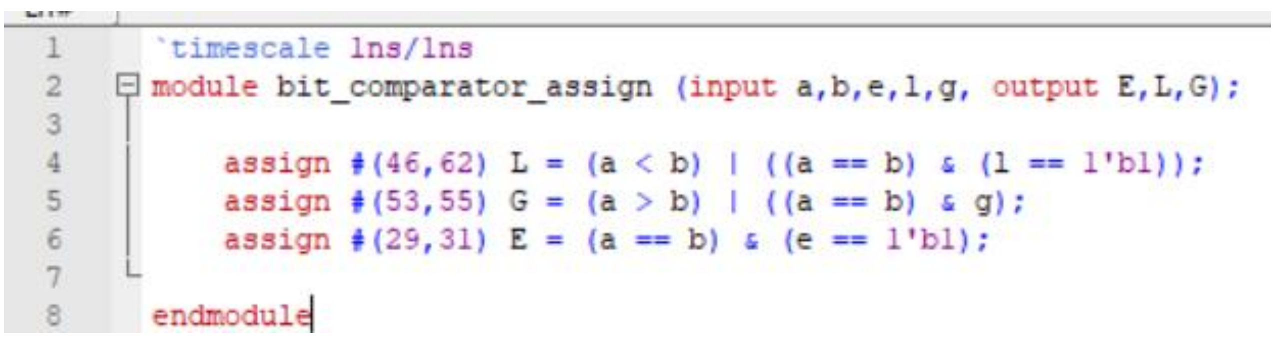
62ns



```
/ca2_tb1c/aa   1
/ca2_tb1c/bb   0
/ca2_tb1c/LL   St0
/ca2_tb1c/ee   0
/ca2_tb1c/gg   1
/ca2_tb1c/ll   1
Now            3391 ns
Cursor 1       2871 ns
Cursor 2       2933 ns
2871 ns  62 ns  2933 ns
```

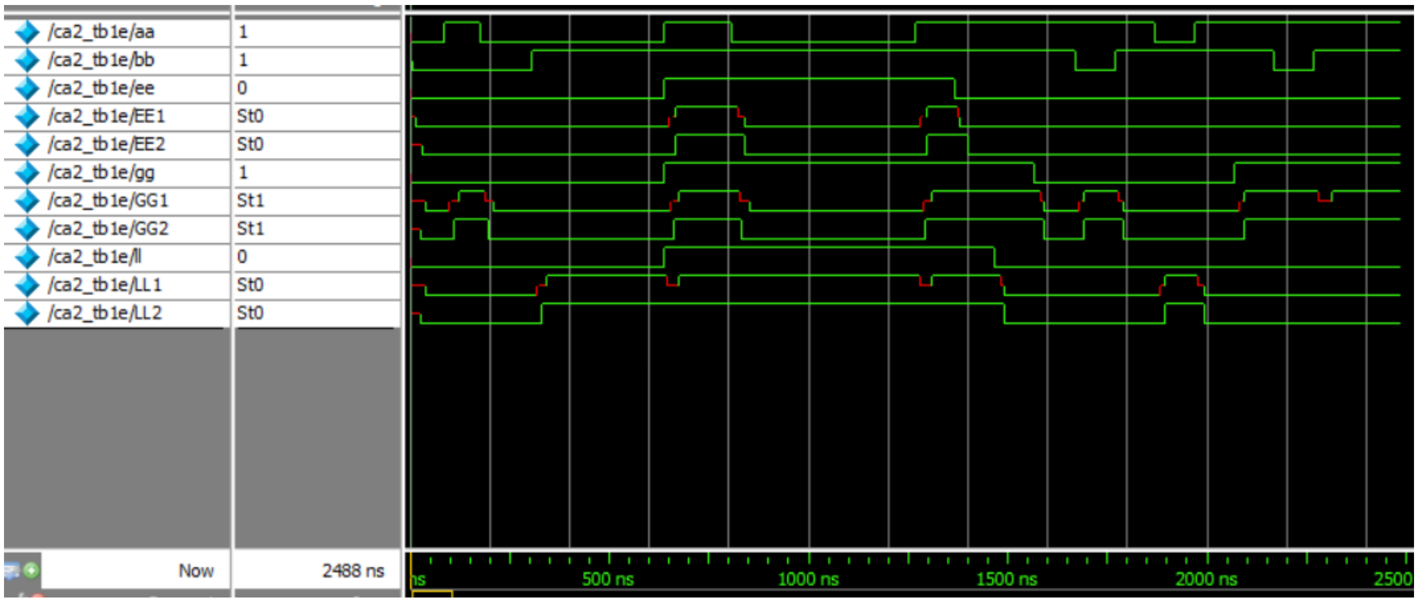
With this picture we can say G L worst delay are near to each other unlike E output and that's because of the design that L and G get used more gate.

1-d



```
`timescale 1ns/1ns
module bit_comparator_assign (input a,b,e,l,g, output E,L,G);

    assign #(46,62) L = (a < b) | ((a == b) & (l == 1'b1));
    assign #(53,55) G = (a > b) | ((a == b) & g);
    assign #(29,31) E = (a == b) & (e == 1'b1);

endmodule
```

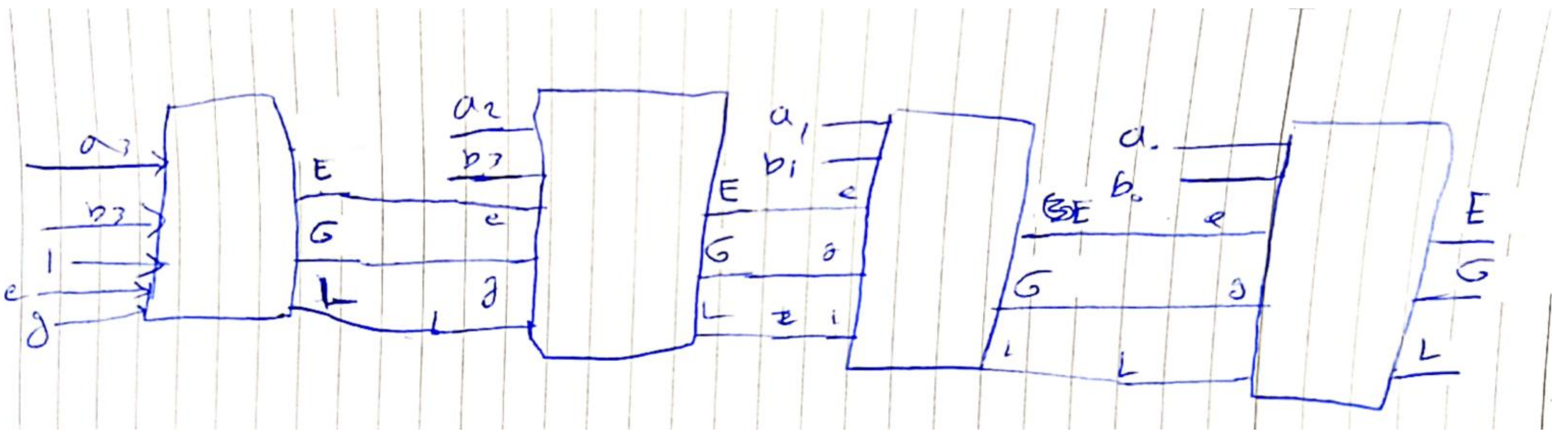
1-e

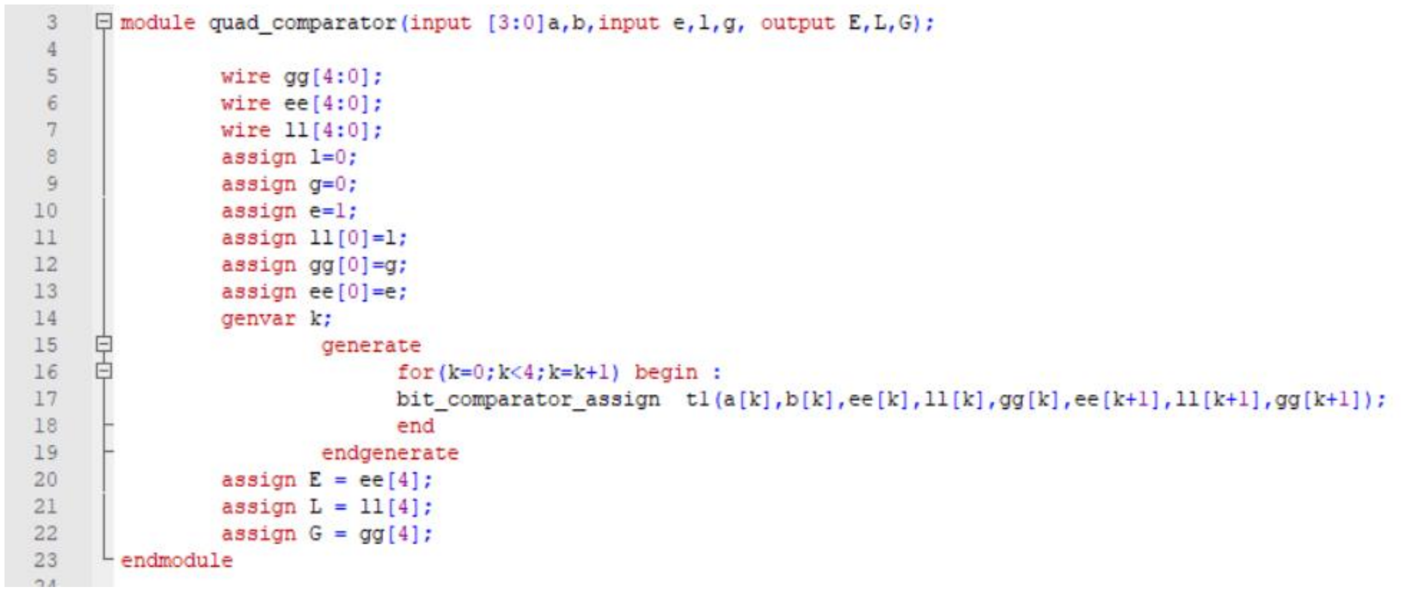


1-Second circuit is better (it doesn't include X in its waves)

2-both delays are the same (and some parts are different and that's because of l g e input that we didn't give them any value in e part,)

2-a

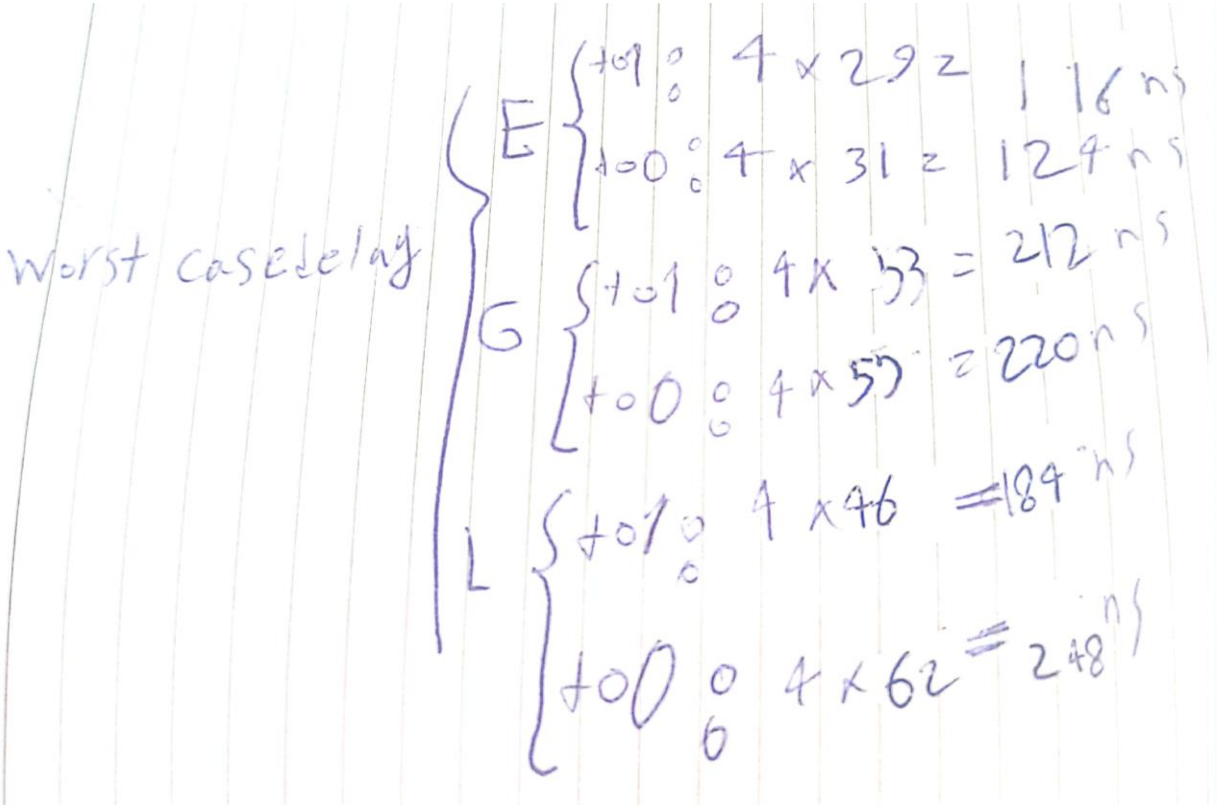




```
module quad_comparator(input [3:0]a,b,input e,l,g, output E,L,G);

        wire gg[4:0];
        wire ee[4:0];
        wire ll[4:0];
        assign l=0;
        assign g=0;
        assign e=1;
        assign ll[0]=l;
        assign gg[0]=g;
        assign ee[0]=e;
        genvar k;
                generate
                      for(k=0;k<4;k=k+1) begin :
                      bit_comparator_assign  t1(a[k],b[k],ee[k],ll[k],gg[k],ee[k+1],ll[k+1],gg[k+1]);
                      end
                endgenerate
        assign E = ee[4];
        assign L = ll[4];
        assign G = gg[4];
endmodule
```

2-b

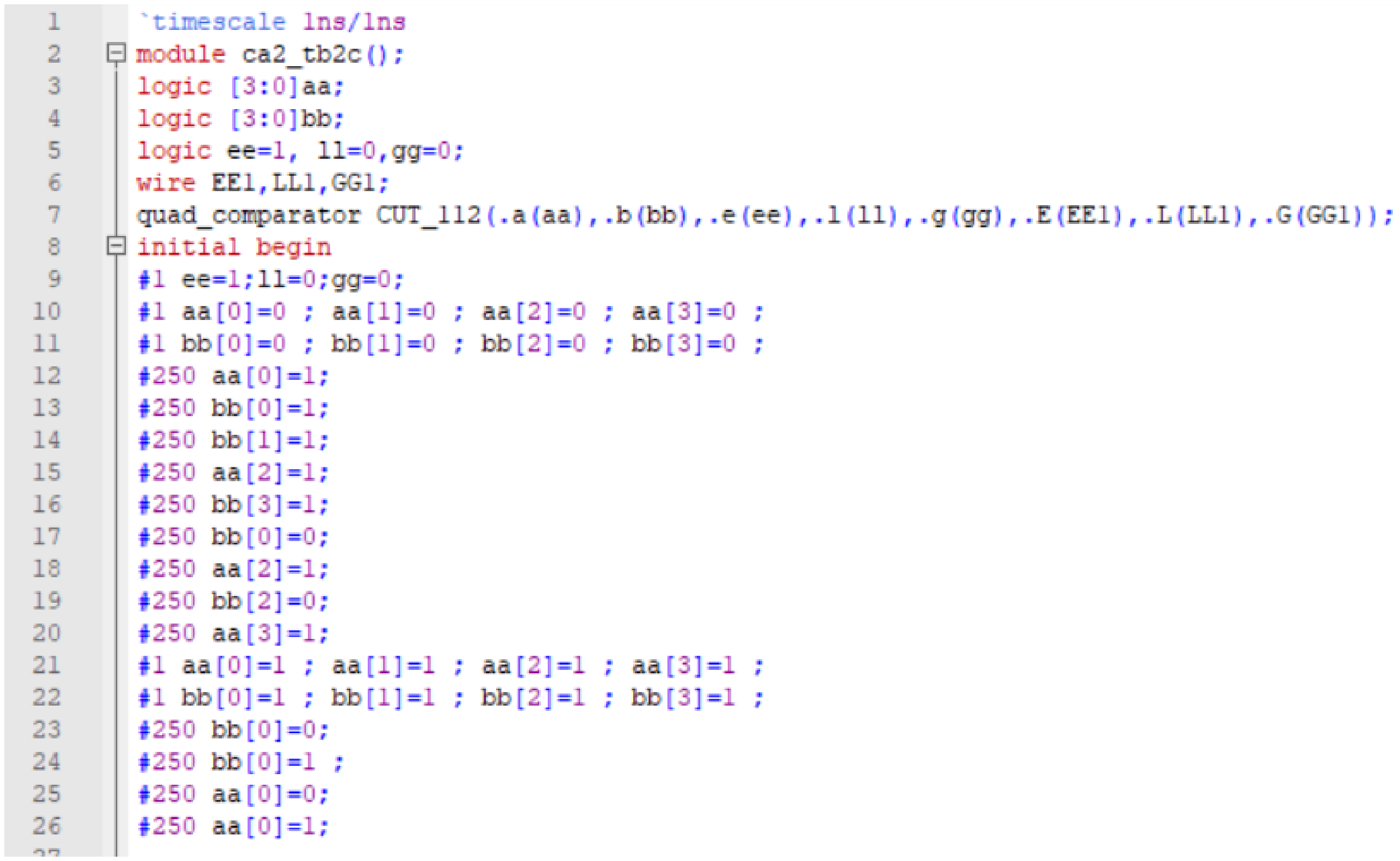
From part a we expect that every output delay multiply by 4

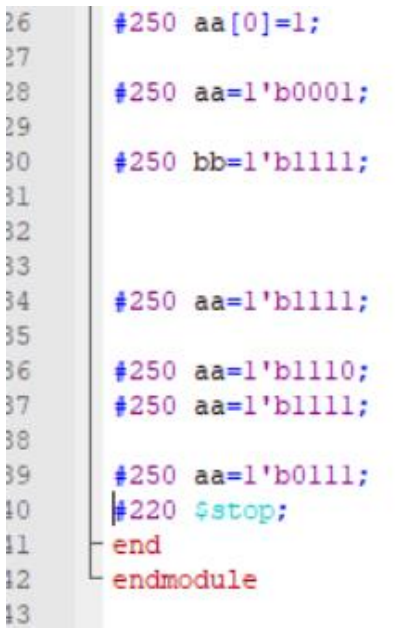


$$\text{Worst case delay}\begin{cases} E\begin{cases} t_{01} : 4\times 29 = 116\text{ ns} \\ t_{10} : 4\times 31 = 124\text{ ns} \end{cases} \\ G\begin{cases} t_{01} : 4\times 53 = 212\text{ ns} \\ t_{10} : 4\times 55 = 220\text{ ns} \end{cases} \\ L\begin{cases} t_{01} : 4\times 46 = 184\text{ ns} \\ t_{10} : 4\times 62 = 248\text{ ns} \end{cases} \end{cases}$$

2-c

Test bench code:

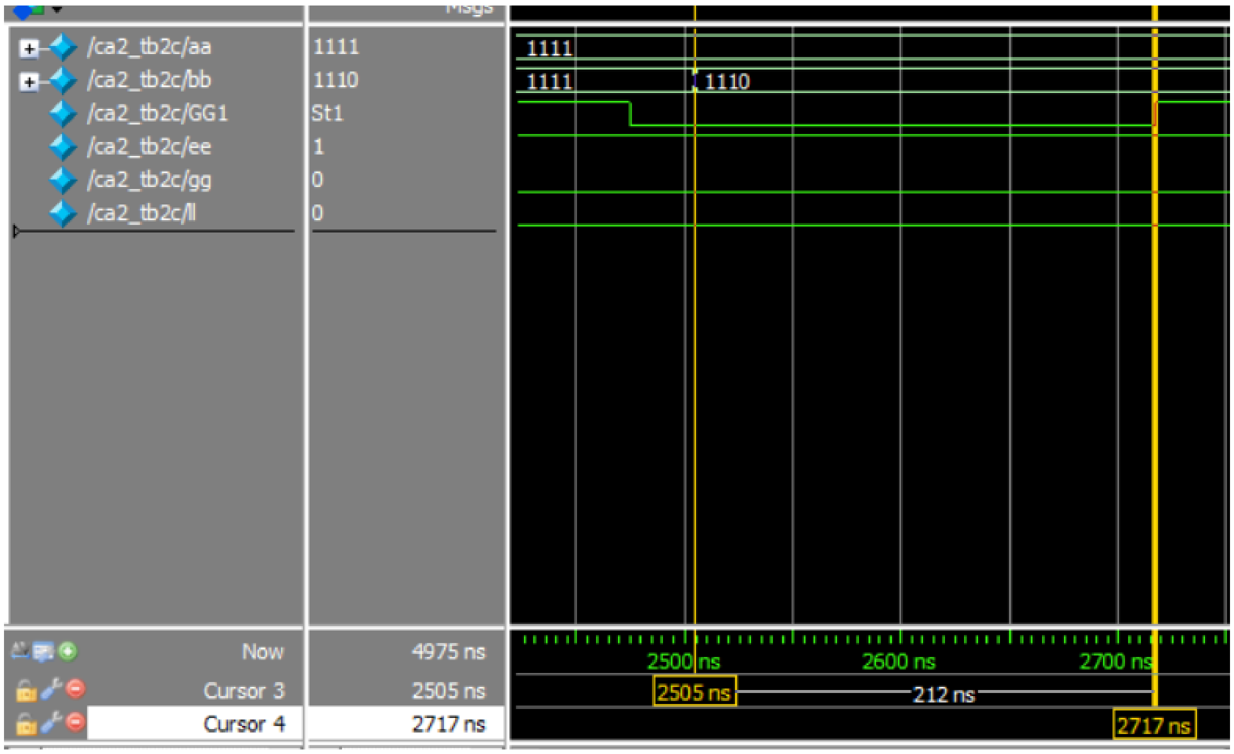

```
`timescale 1ns/1ns
module ca2_tb2c();
logic [3:0]aa;
logic [3:0]bb;
logic ee=1, ll=0,gg=0;
wire EE1,LL1,GG1;
quad_comparator CUT_112(.a(aa),.b(bb),.e(ee),.l(ll),.g(gg),.E(EE1),.L(LL1),.G(GG1));
initial begin
#1 ee=1;ll=0;gg=0;
#1 aa[0]=0 ; aa[1]=0 ; aa[2]=0 ; aa[3]=0 ;
#1 bb[0]=0 ; bb[1]=0 ; bb[2]=0 ; bb[3]=0 ;
#250 aa[0]=1;
#250 bb[0]=1;
#250 bb[1]=1;
#250 aa[2]=1;
#250 bb[3]=1;
#250 bb[0]=0;
#250 aa[2]=1;
#250 bb[2]=0;
#250 aa[3]=1;
#1 aa[0]=1 ; aa[1]=1 ; aa[2]=1 ; aa[3]=1 ;
#1 bb[0]=1 ; bb[1]=1 ; bb[2]=1 ; bb[3]=1 ;
#250 bb[0]=0;
#250 bb[0]=1 ;
#250 aa[0]=0;
#250 aa[0]=1;
```


```
#250 aa[0]=1;

#250 aa=1'b0001;

#250 bb=1'b1111;



#250 aa=1'b1111;

#250 aa=1'b1110;
#250 aa=1'b1111;

#250 aa=1'b0111;
#220 $stop;
end
endmodule
```

G output

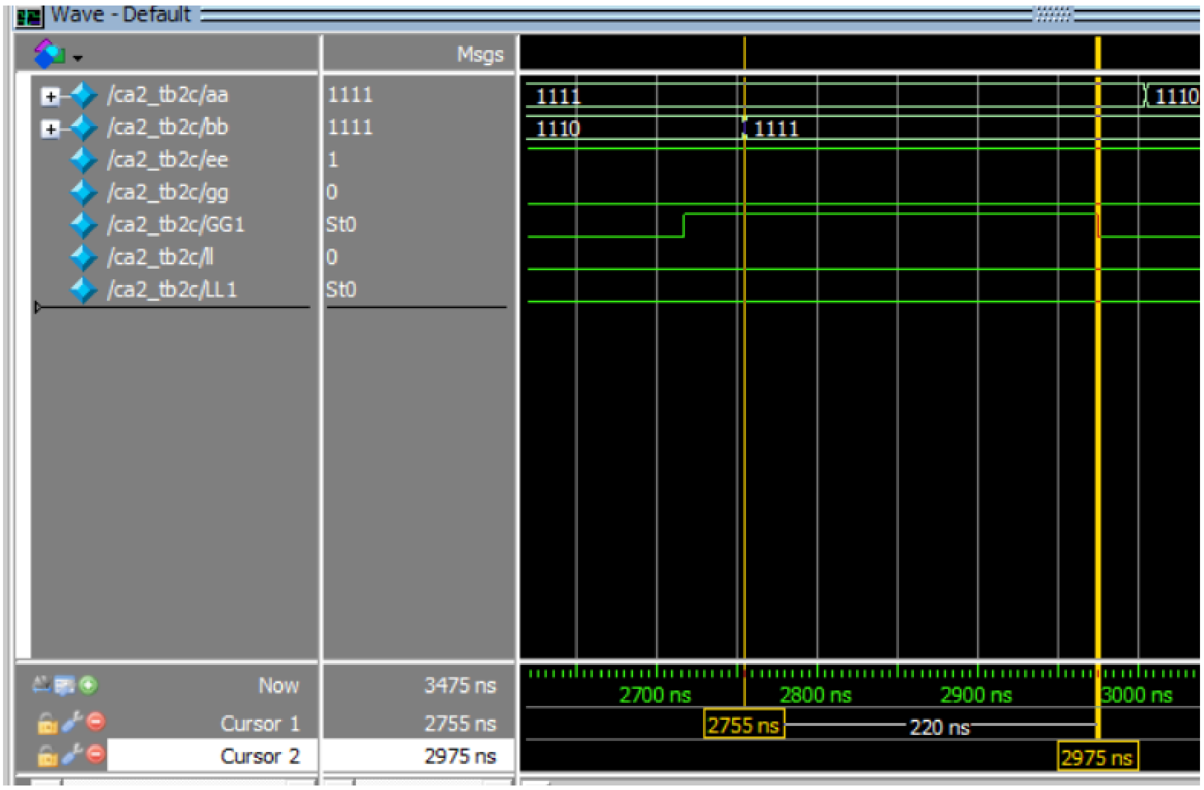
To 1:

212 ns



To 0:

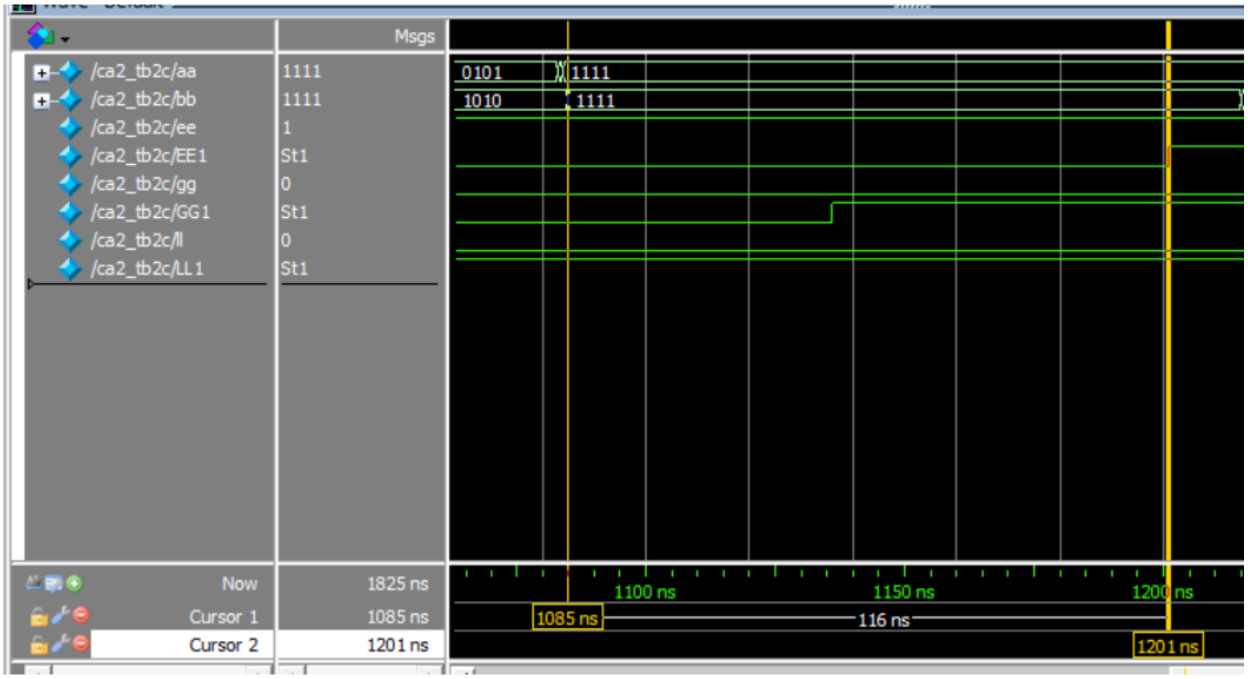
220ns



E output

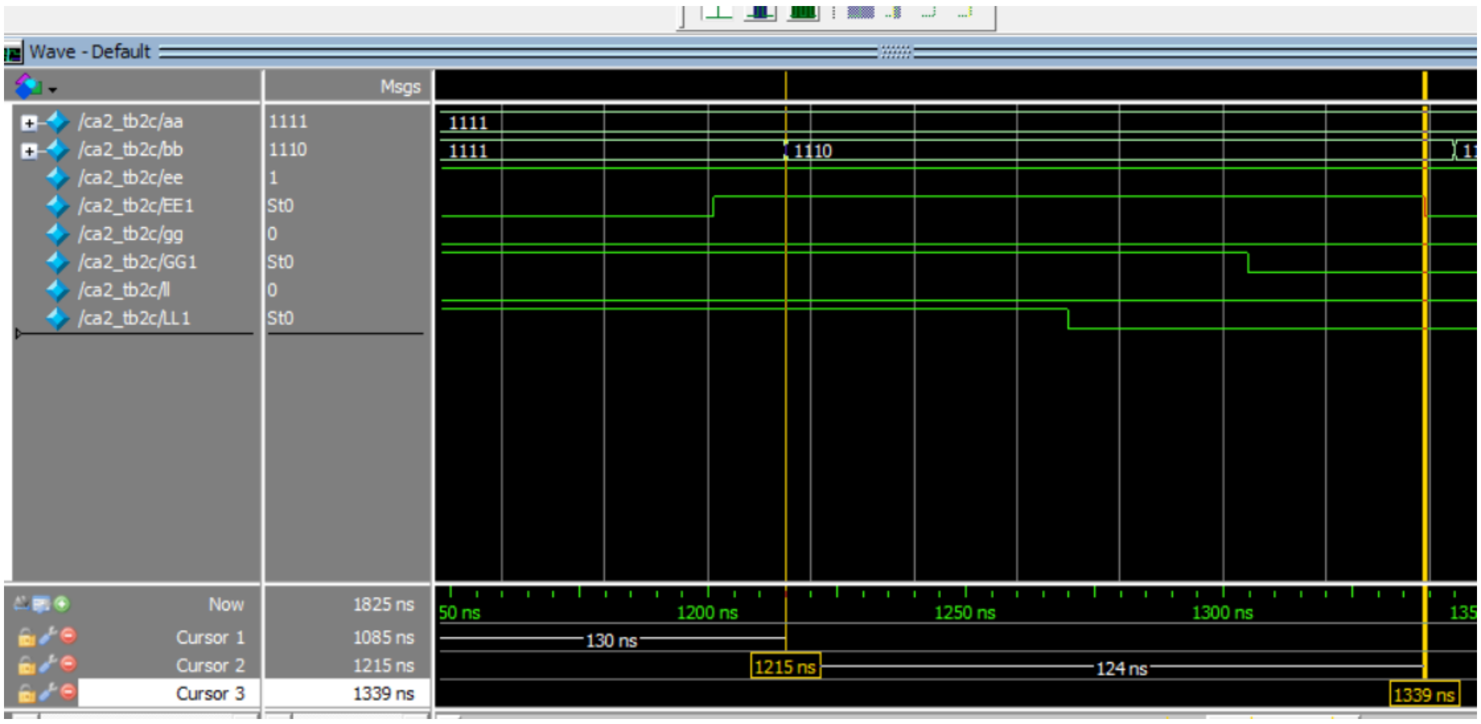
To 1:

116 ns



To 0:

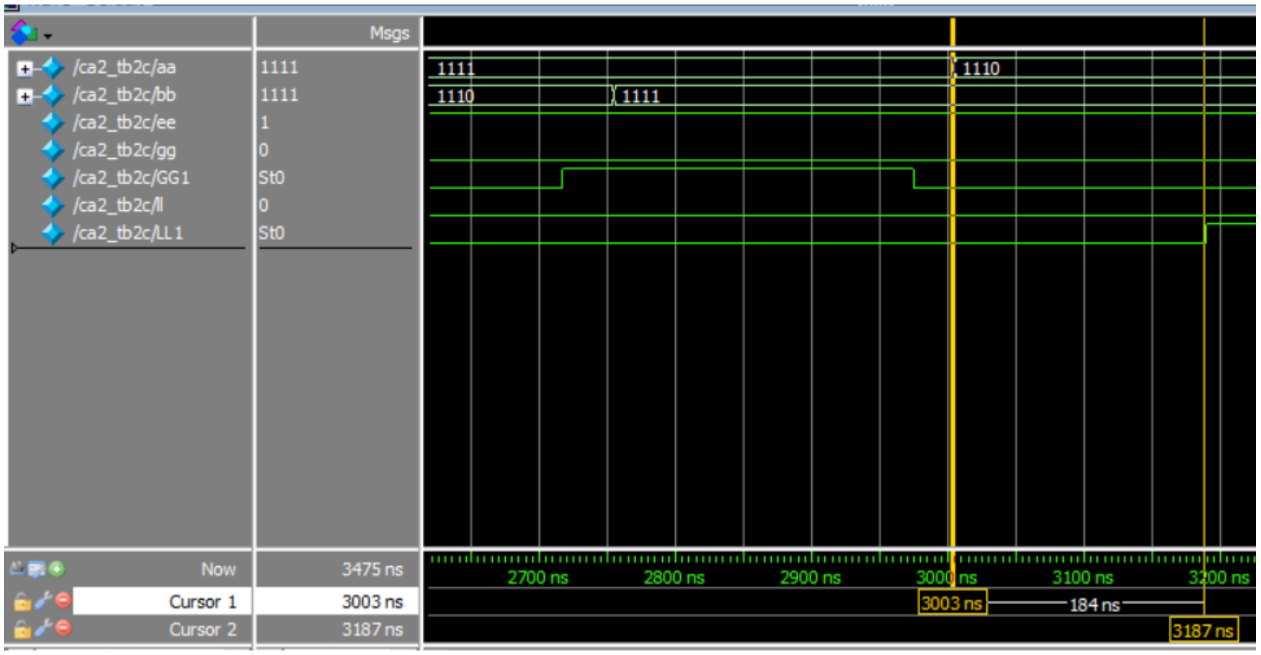
124 ns



L output

To 1:

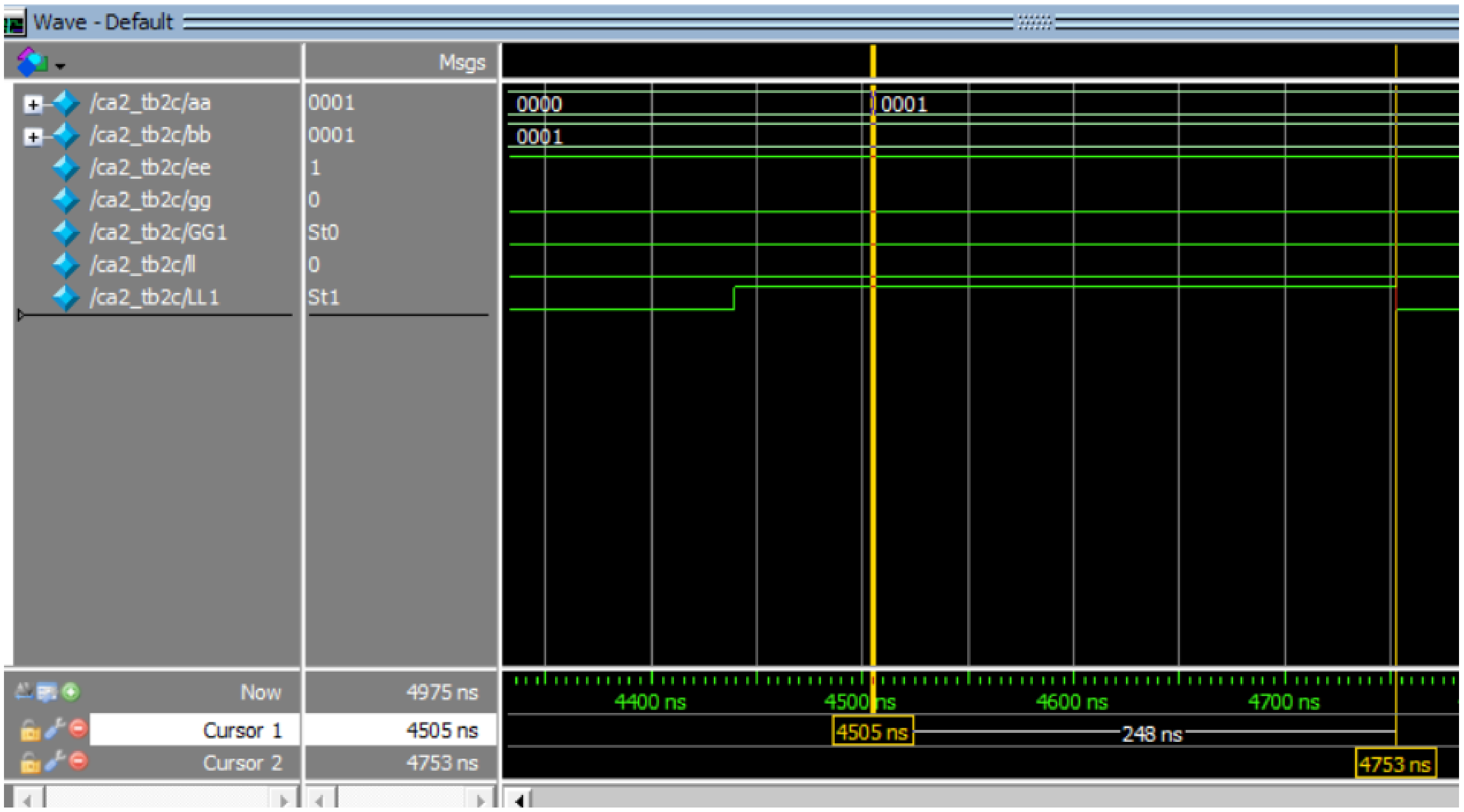
184 ns



```
                     Msgs
/ca2_tb2c/aa         1111
/ca2_tb2c/bb         1111
/ca2_tb2c/ee         1
/ca2_tb2c/gg         0
/ca2_tb2c/GG1        St0
/ca2_tb2c/ll         0
/ca2_tb2c/LL1        St0

Now                  3475 ns
Cursor 1             3003 ns
Cursor 2             3187 ns
```

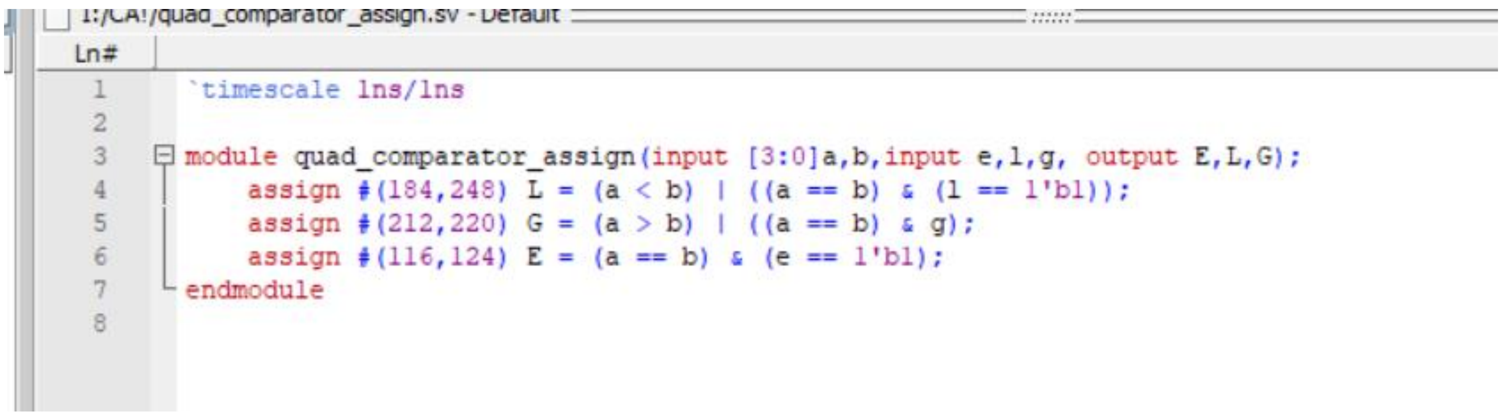
To 0:

248 ns



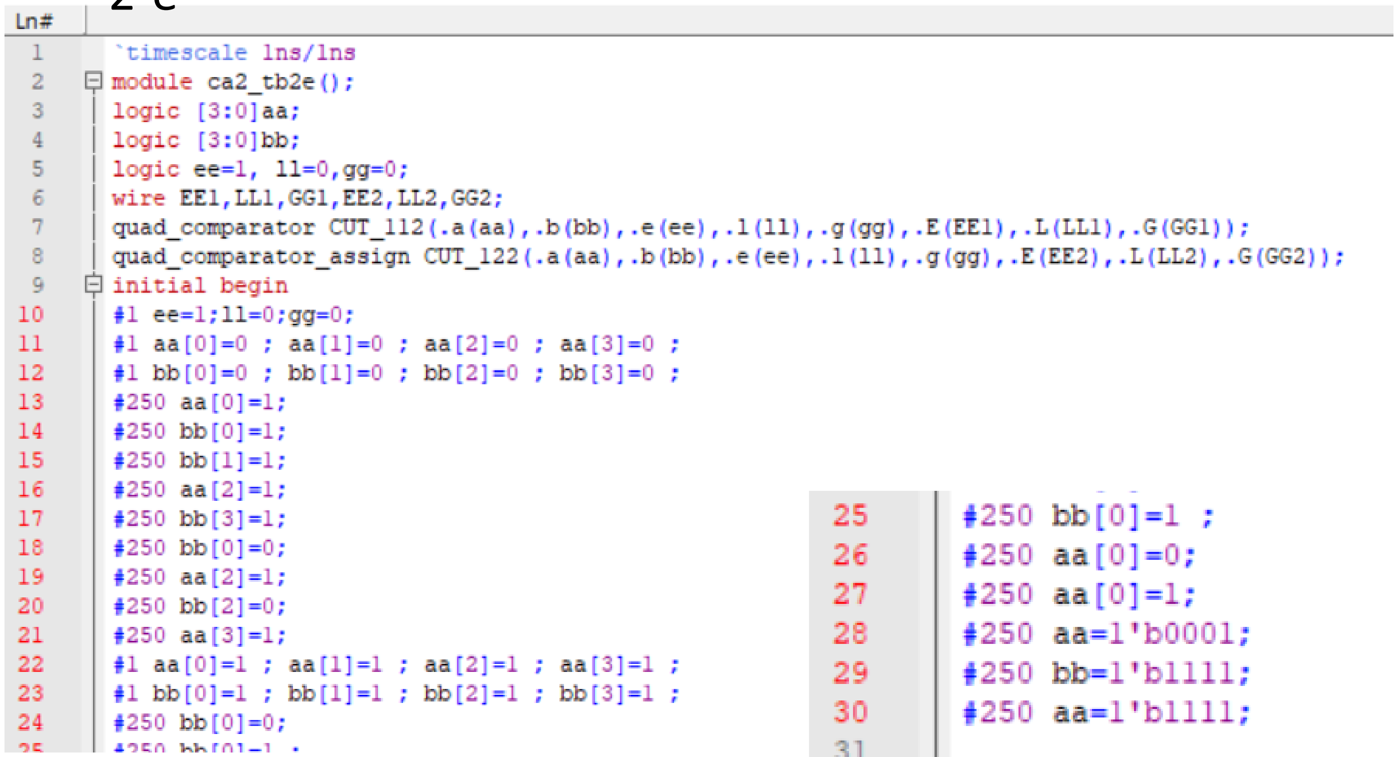
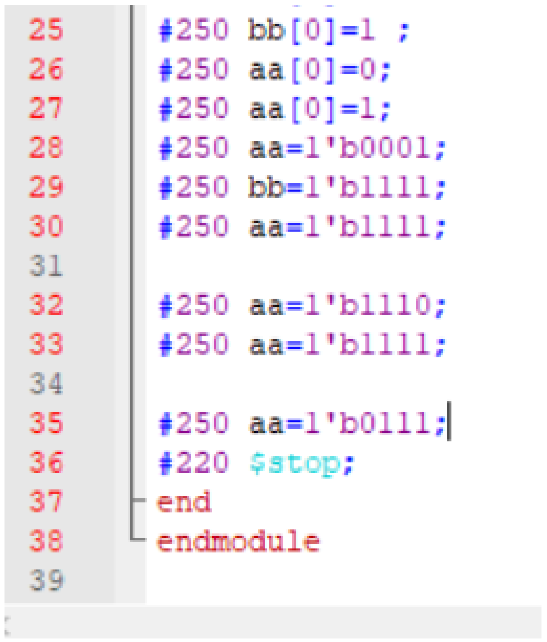
```
Wave - Default
                     Msgs
/ca2_tb2c/aa         0001
/ca2_tb2c/bb         0001
/ca2_tb2c/ee         1
/ca2_tb2c/gg         0
/ca2_tb2c/GG1        St0
/ca2_tb2c/ll         0
/ca2_tb2c/LL1        St1

Now                  4975 ns
Cursor 1             4505 ns
Cursor 2             4753 ns
```

2-d:

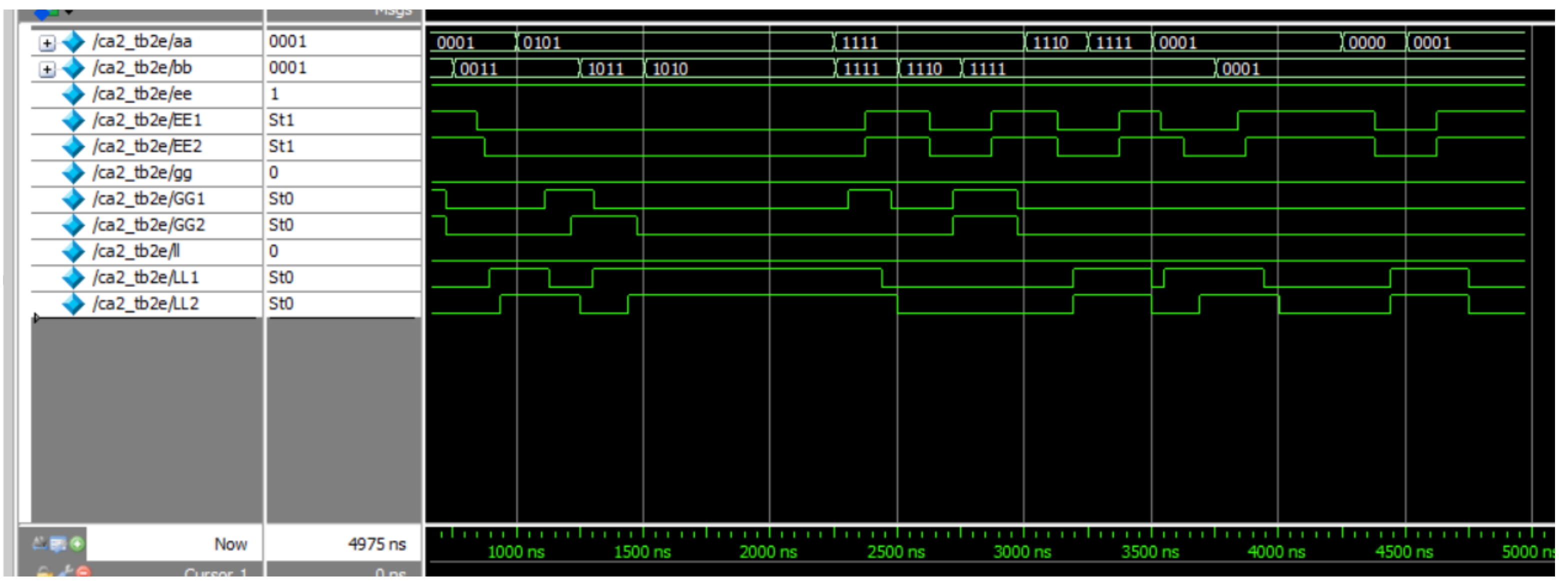


```
`timescale 1ns/1ns

module quad_comparator_assign(input [3:0]a,b,input e,l,g, output E,L,G);
    assign #(184,248) L = (a < b) | ((a == b) & (l == 1'b1));
    assign #(212,220) G = (a > b) | ((a == b) & g);
    assign #(116,124) E = (a == b) & (e == 1'b1);
endmodule
```

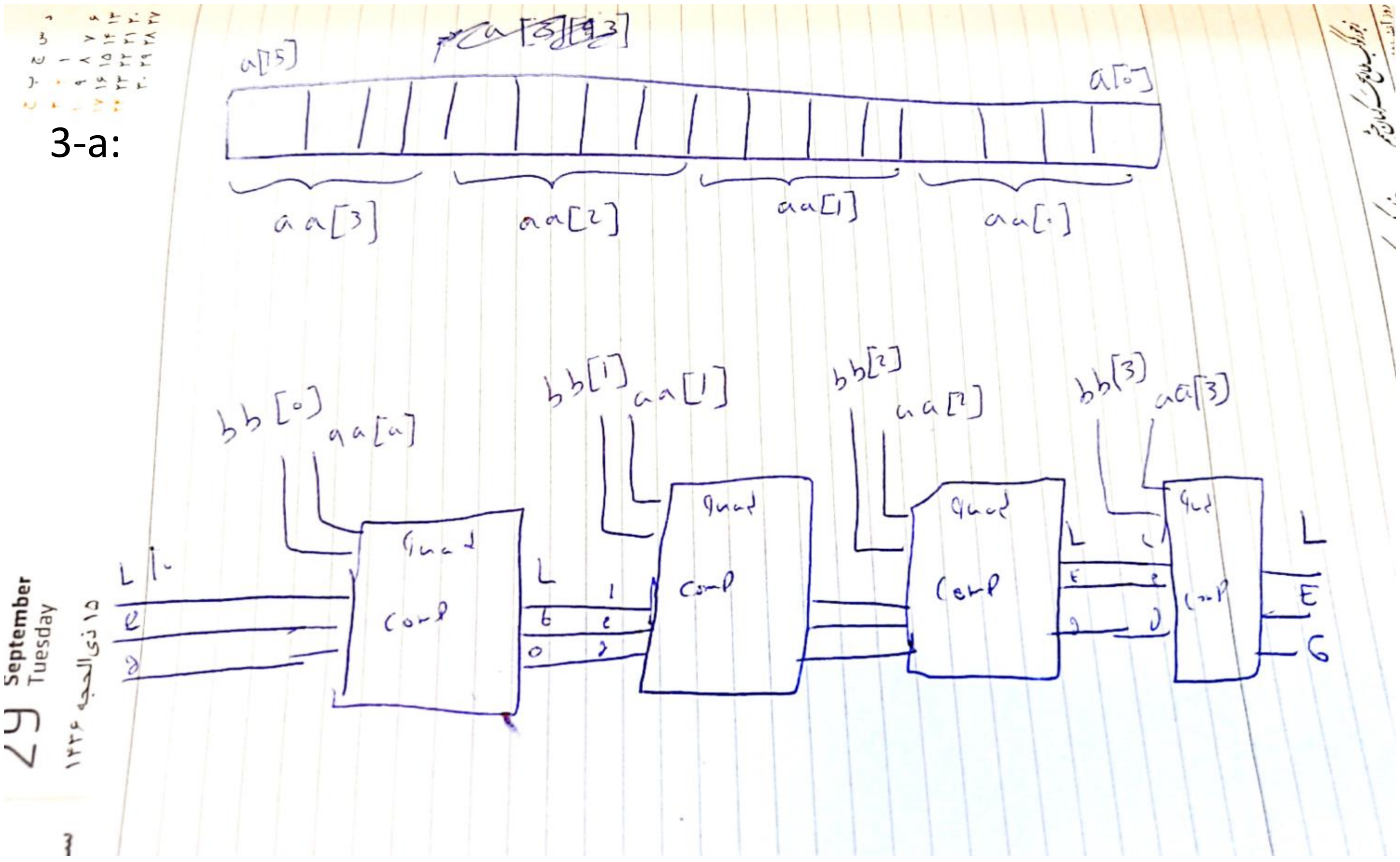
2-e

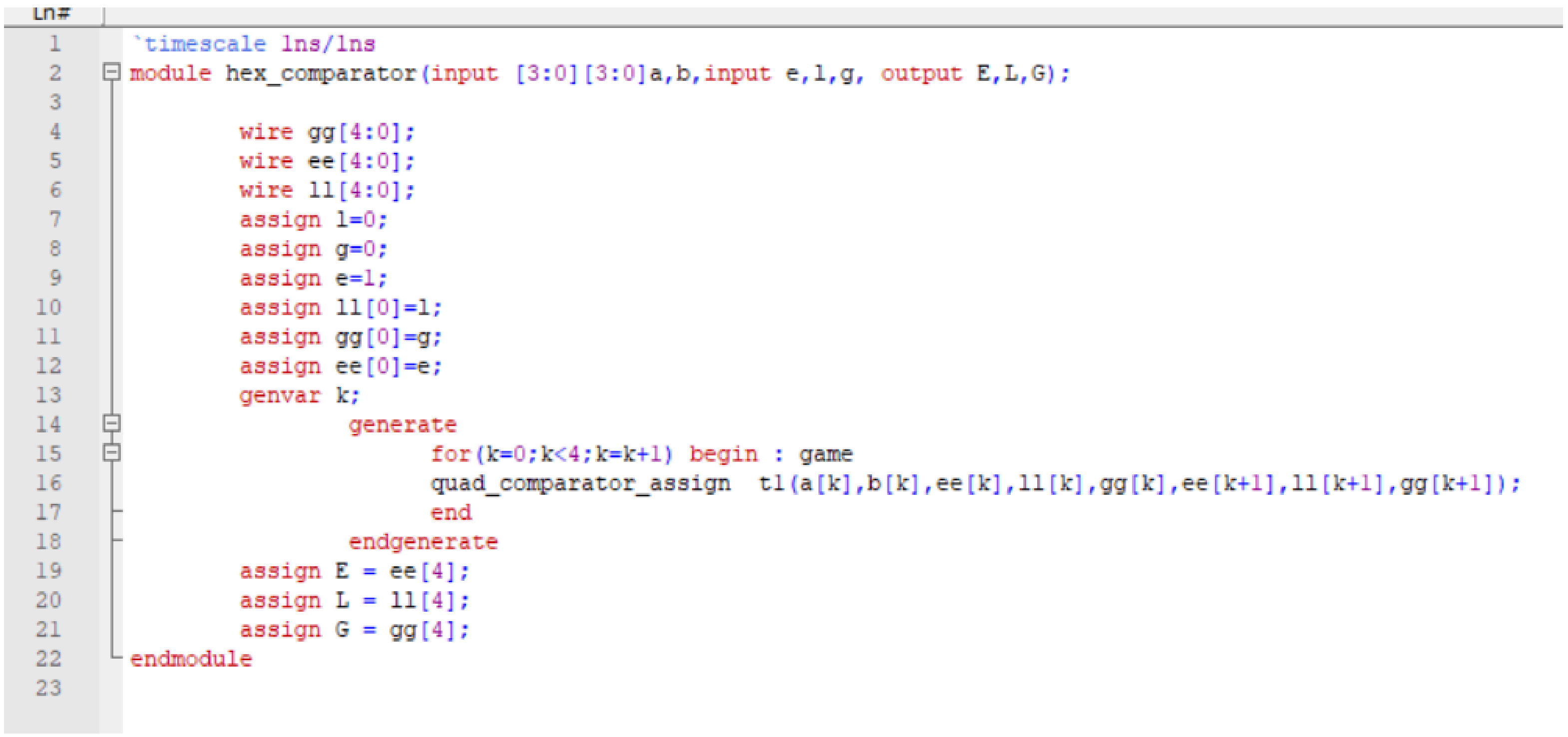
```
`timescale 1ns/1ns
module ca2_tb2e();
logic [3:0]aa;
logic [3:0]bb;
logic ee=1, ll=0,gg=0;
wire EE1,LL1,GG1,EE2,LL2,GG2;
quad_comparator CUT_112(.a(aa),.b(bb),.e(ee),.l(ll),.g(gg),.E(EE1),.L(LL1),.G(GG1));
quad_comparator_assign CUT_122(.a(aa),.b(bb),.e(ee),.l(ll),.g(gg),.E(EE2),.L(LL2),.G(GG2));
initial begin
#1 ee=1;ll=0;gg=0;
#1 aa[0]=0 ; aa[1]=0 ; aa[2]=0 ; aa[3]=0 ;
#1 bb[0]=0 ; bb[1]=0 ; bb[2]=0 ; bb[3]=0 ;
#250 aa[0]=1;
#250 bb[0]=1;
#250 bb[1]=1;
#250 aa[2]=1;
#250 bb[3]=1;
#250 bb[0]=0;
#250 aa[2]=1;
#250 bb[2]=0;
#250 aa[3]=1;
#1 aa[0]=1 ; aa[1]=1 ; aa[2]=1 ; aa[3]=1 ;
#1 bb[0]=1 ; bb[1]=1 ; bb[2]=1 ; bb[3]=1 ;
#250 bb[0]=0;
#250 bb[0]=1 ;
#250 aa[0]=0;
#250 aa[0]=1;
#250 aa=1'b0001;
#250 bb=1'b1111;
#250 aa=1'b1111;

#250 aa=1'b1110;
#250 aa=1'b1111;

#250 aa=1'b0111;
#220 $stop;
end
endmodule
```


/ca2_tb2e/aa 0001
/ca2_tb2e/bb 0001
/ca2_tb2e/ee 1
/ca2_tb2e/EE1 St1
/ca2_tb2e/EE2 St1
/ca2_tb2e/gg 0
/ca2_tb2e/GG1 St0
/ca2_tb2e/GG2 St0
/ca2_tb2e/ll 0
/ca2_tb2e/LL1 St0
/ca2_tb2e/LL2 St0
0001
0101
1111
1110
1111
0001
0000
0001
0011
1011
1010
1111
1110
1111
0001
Now 4975 ns
1000 ns
1500 ns
2000 ns
2500 ns
3000 ns
3500 ns
4000 ns
4500 ns
5000 ns

3-a:

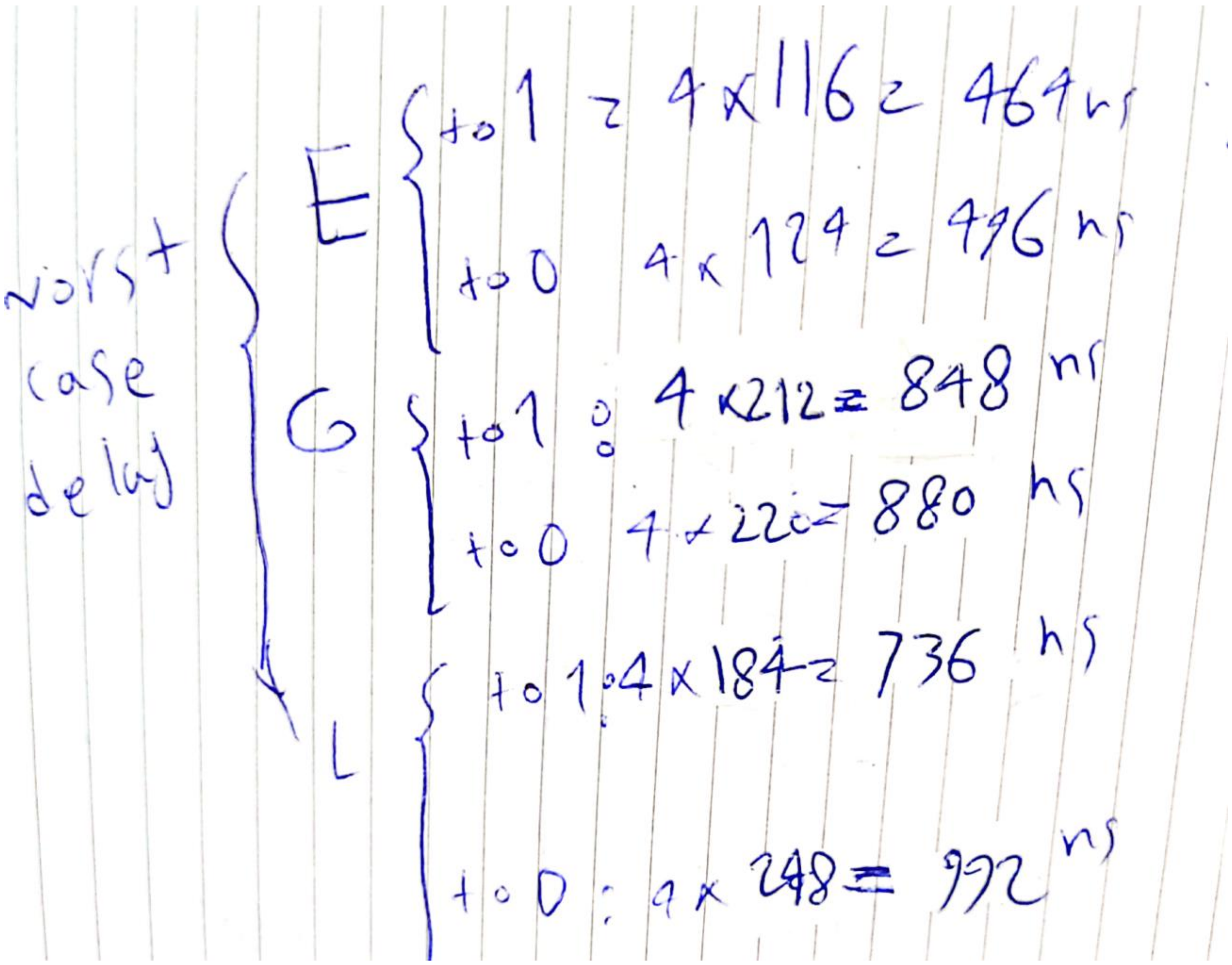

a[15]
a[0]
aa[3]
aa[2]
aa[1]
aa[0]
bb[0]
aa[0]
bb[1]
aa[1]
bb[2]
aa[2]
bb[3]
aa[3]
Quad Comp
L
E
G

Verilog code:



```
`timescale 1ns/1ns
module hex_comparator(input [3:0][3:0]a,b,input e,l,g, output E,L,G);

        wire gg[4:0];
        wire ee[4:0];
        wire ll[4:0];
        assign l=0;
        assign g=0;
        assign e=1;
        assign ll[0]=l;
        assign gg[0]=g;
        assign ee[0]=e;
        genvar k;
                generate
                      for(k=0;k<4;k=k+1) begin : game
                      quad_comparator_assign  t1(a[k],b[k],ee[k],ll[k],gg[k],ee[k+1],ll[k+1],gg[k+1]);
                      end
                endgenerate
        assign E = ee[4];
        assign L = ll[4];
        assign G = gg[4];
endmodule
```

3-b

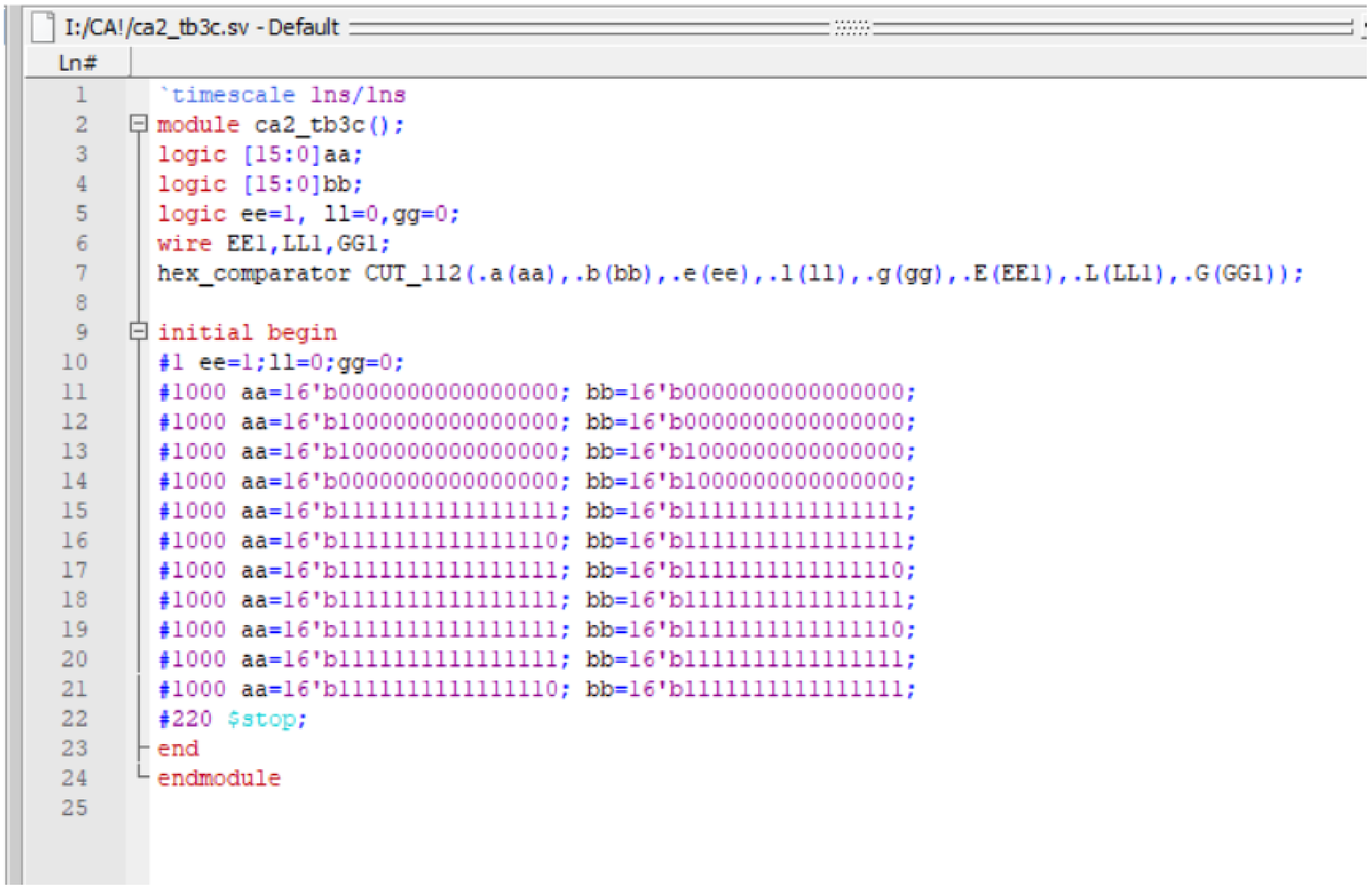


worst case delay

- E
  - to 1 = $4 \times 116 = 464$ ns
  - to 0 $4 \times 124 = 496$ ns
- G
  - to 1 : $4 \times 212 = 848$ ns
  - to 0 $4 \times 220 = 880$ ns
- L
  - to 1 : $4 \times 184 = 736$ ns
  - to 0 : $4 \times 248 = 992$ ns

3-c

Test bench code:



I:/CA!/ca2_tb3c.sv - Default

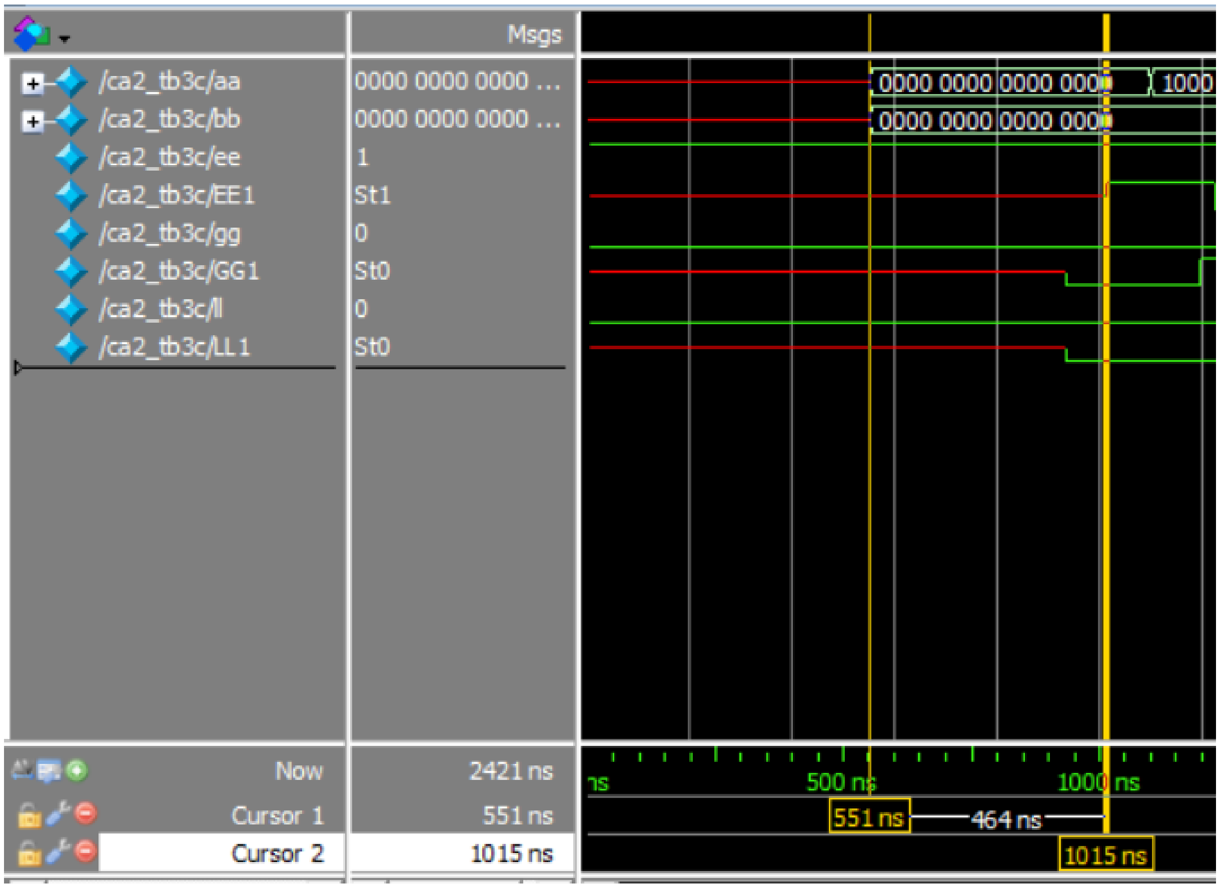
```
`timescale 1ns/1ns
module ca2_tb3c();
logic [15:0]aa;
logic [15:0]bb;
logic ee=1, ll=0,gg=0;
wire EE1,LL1,GG1;
hex_comparator CUT_112(.a(aa),.b(bb),.e(ee),.l(ll),.g(gg),.E(EE1),.L(LL1),.G(GG1));

initial begin
#1 ee=1;ll=0;gg=0;
#1000 aa=16'b0000000000000000; bb=16'b0000000000000000;
#1000 aa=16'b1000000000000000; bb=16'b0000000000000000;
#1000 aa=16'b1000000000000000; bb=16'b1000000000000000;
#1000 aa=16'b0000000000000000; bb=16'b1000000000000000;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111111;
#1000 aa=16'b1111111111111110; bb=16'b1111111111111111;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111110;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111111;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111110;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111111;
#1000 aa=16'b1111111111111110; bb=16'b1111111111111111;
#220 $stop;
end
endmodule
```

Worst case delay:

E output:

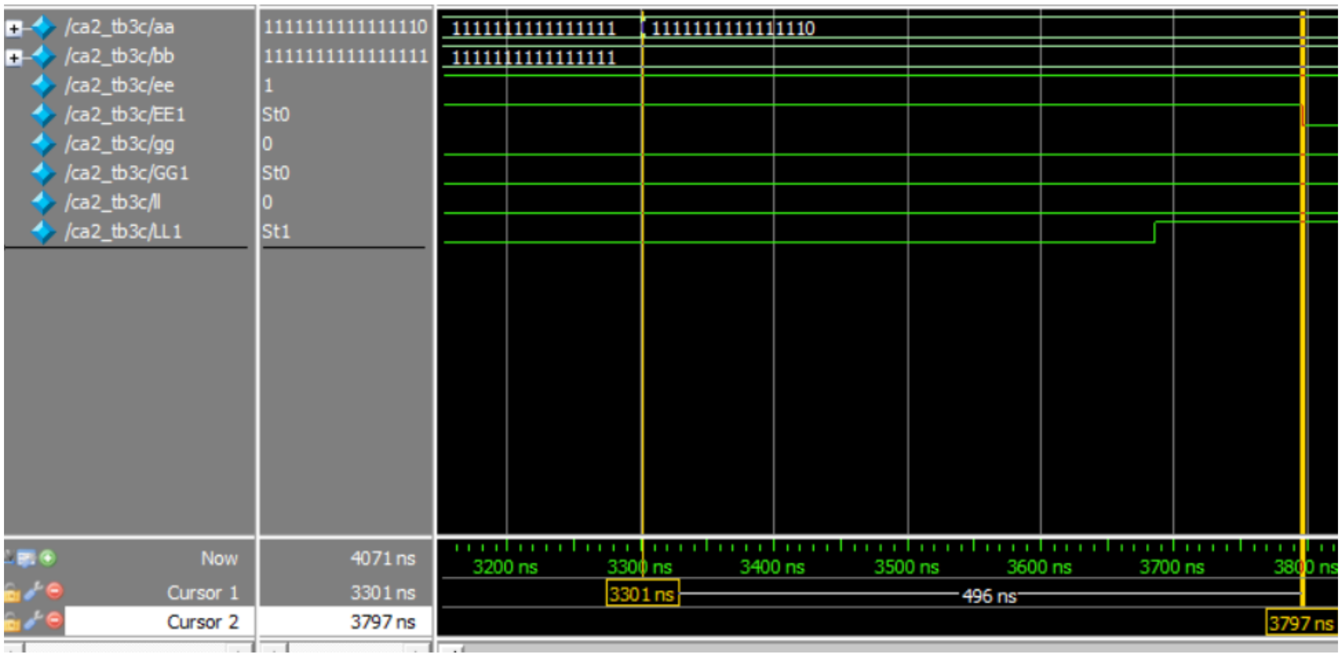
To 1:

464ns



To 0:

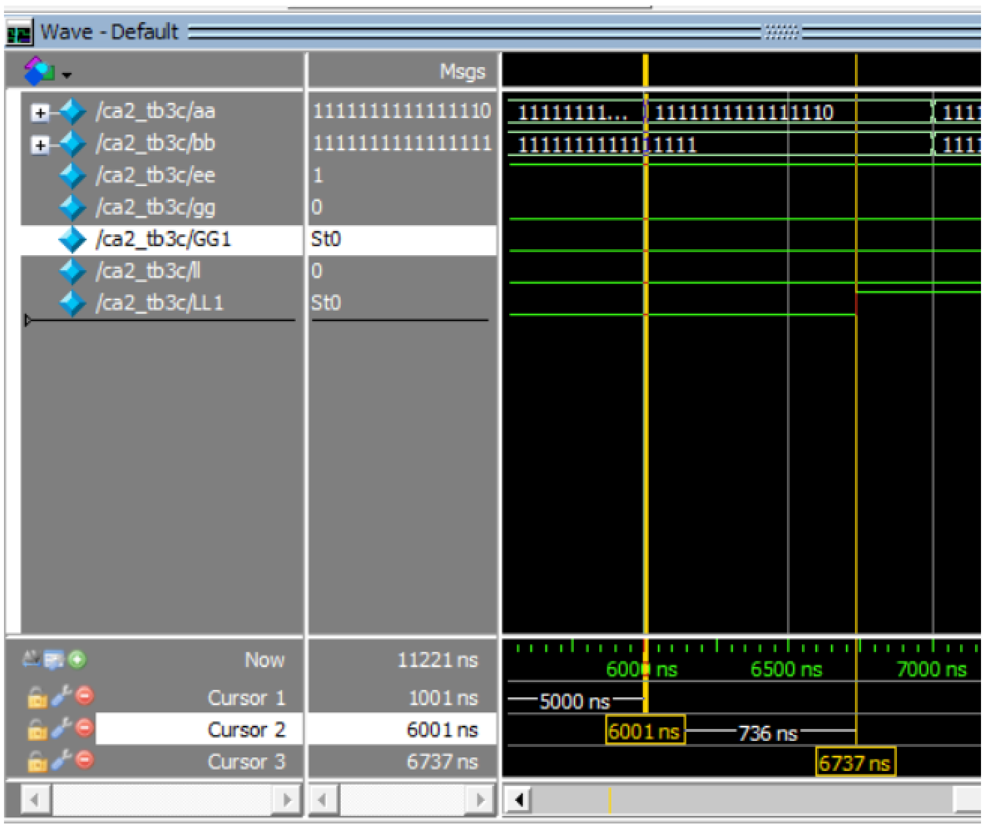
496ns



G output:

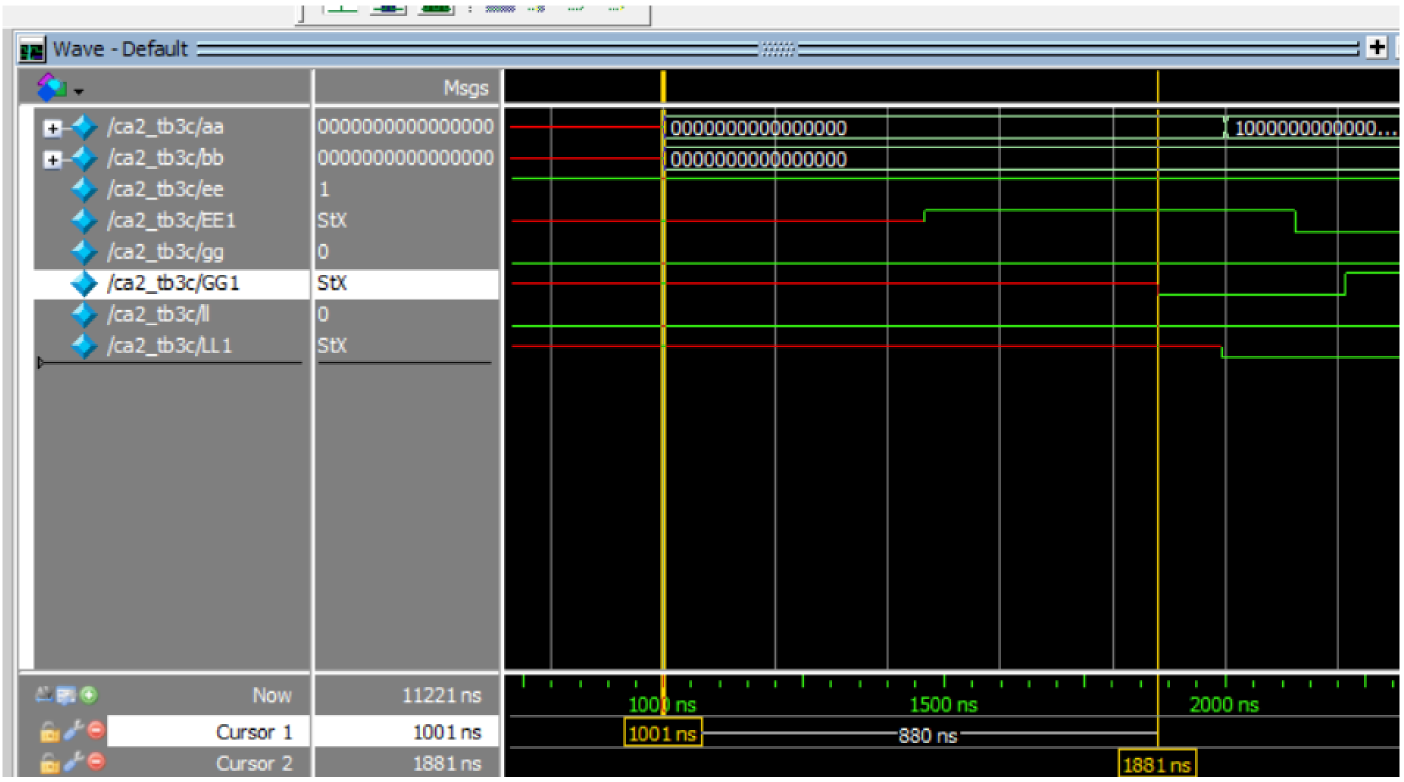
To 1:

848ns



To 0:

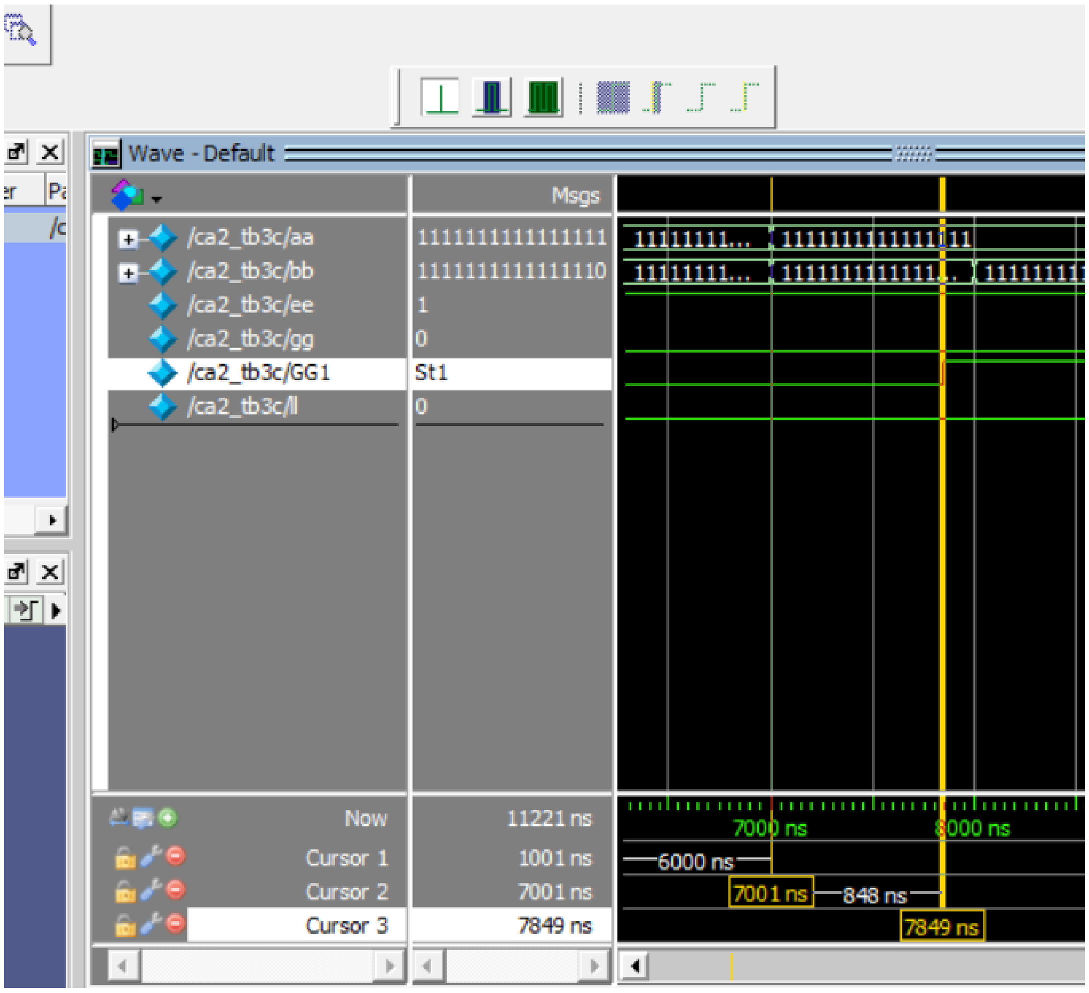
880ns



L output:

To 1:

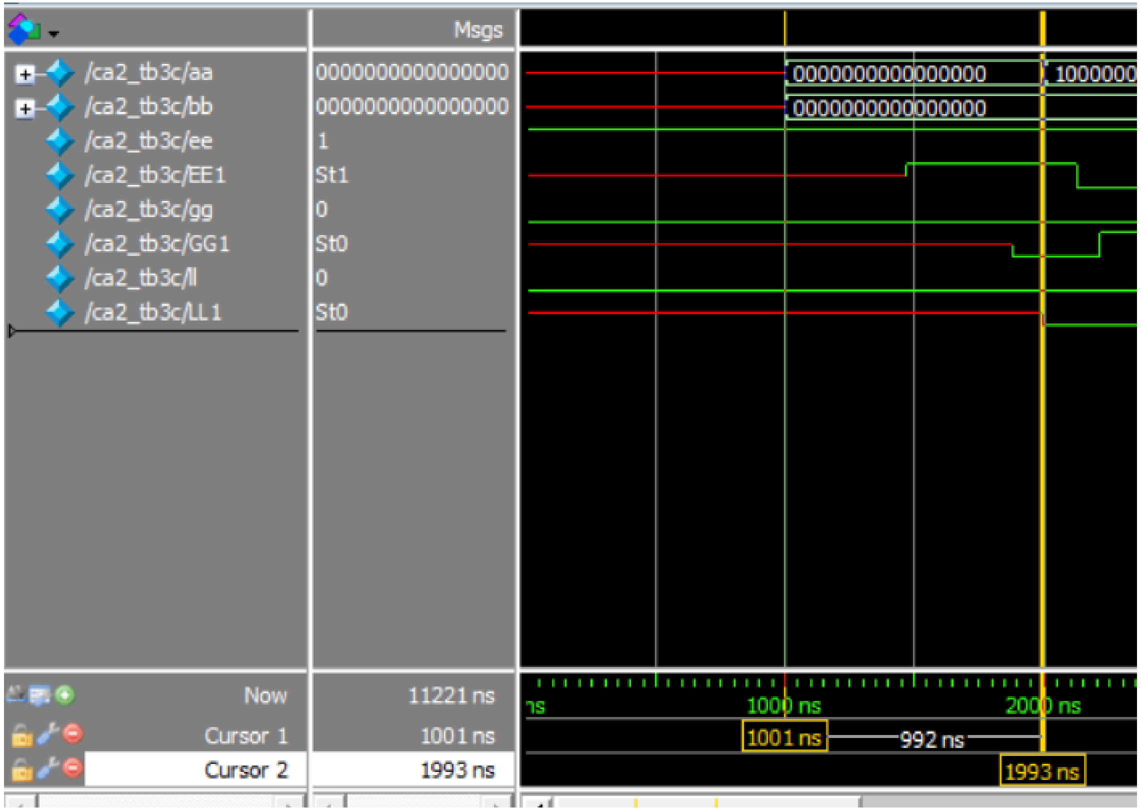
736ns



```
/ca2_tb3c/aa    1111111111111111
/ca2_tb3c/bb    1111111111111110
/ca2_tb3c/ee    1
/ca2_tb3c/gg    0
/ca2_tb3c/GG1   St1
/ca2_tb3c/ll    0

Now       11221 ns
Cursor 1  1001 ns
Cursor 2  7001 ns
Cursor 3  7849 ns
```

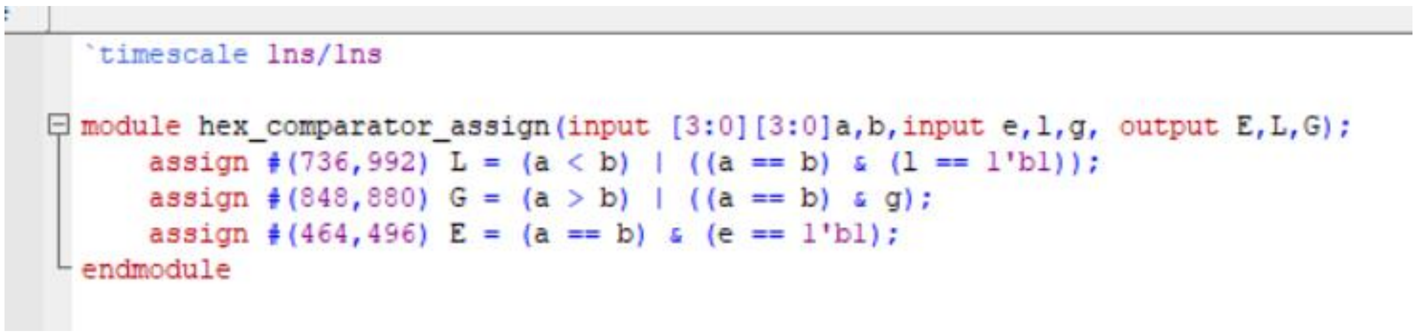
To 0:

992ns



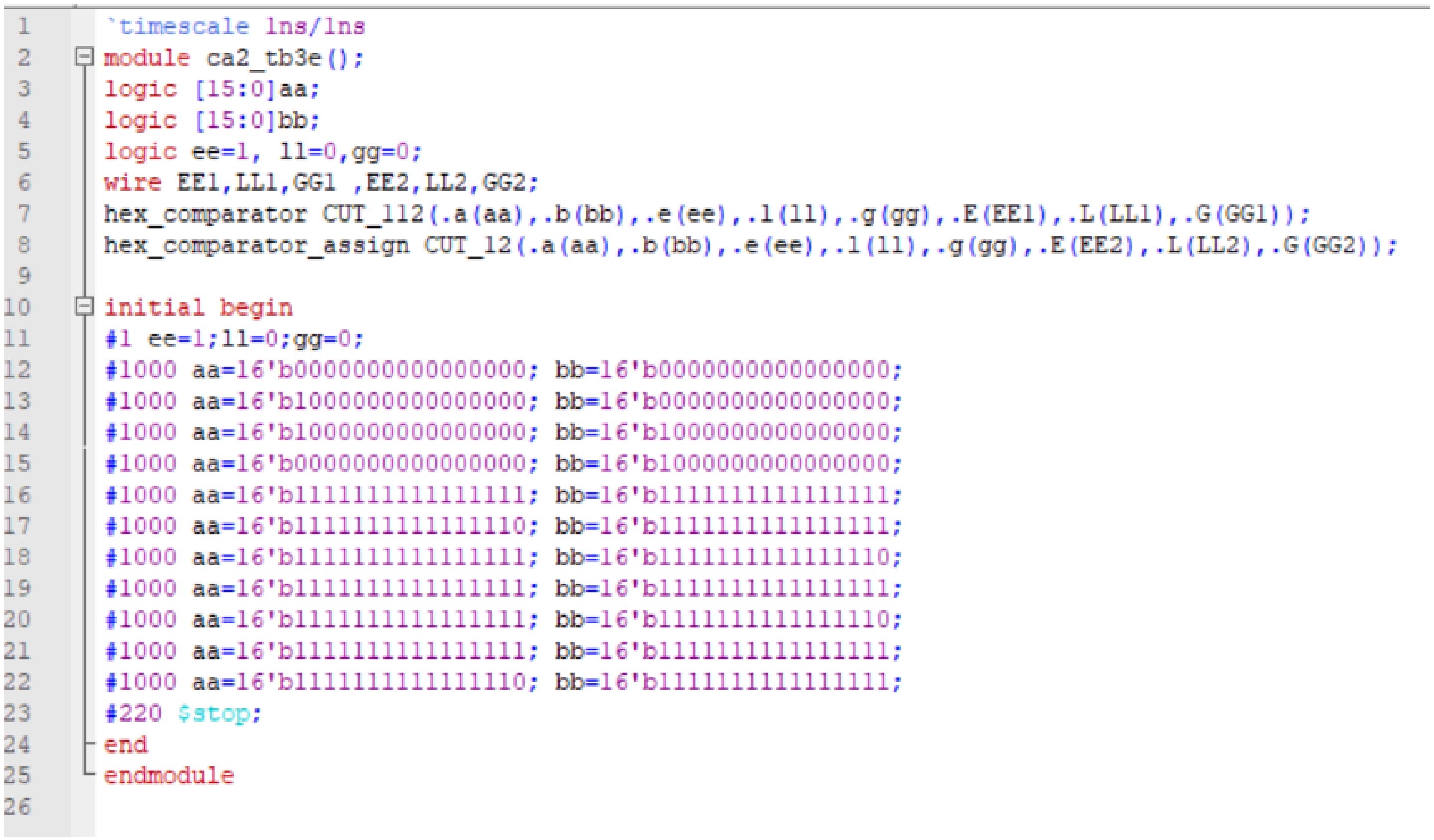
```
/ca2_tb3c/aa    0000000000000000
/ca2_tb3c/bb    0000000000000000
/ca2_tb3c/ee    1
/ca2_tb3c/EE1   St1
/ca2_tb3c/gg    0
/ca2_tb3c/GG1   St0
/ca2_tb3c/ll    0
/ca2_tb3c/LL1   St0

Now       11221 ns
Cursor 1  1001 ns
Cursor 2  1993 ns
```

3-d

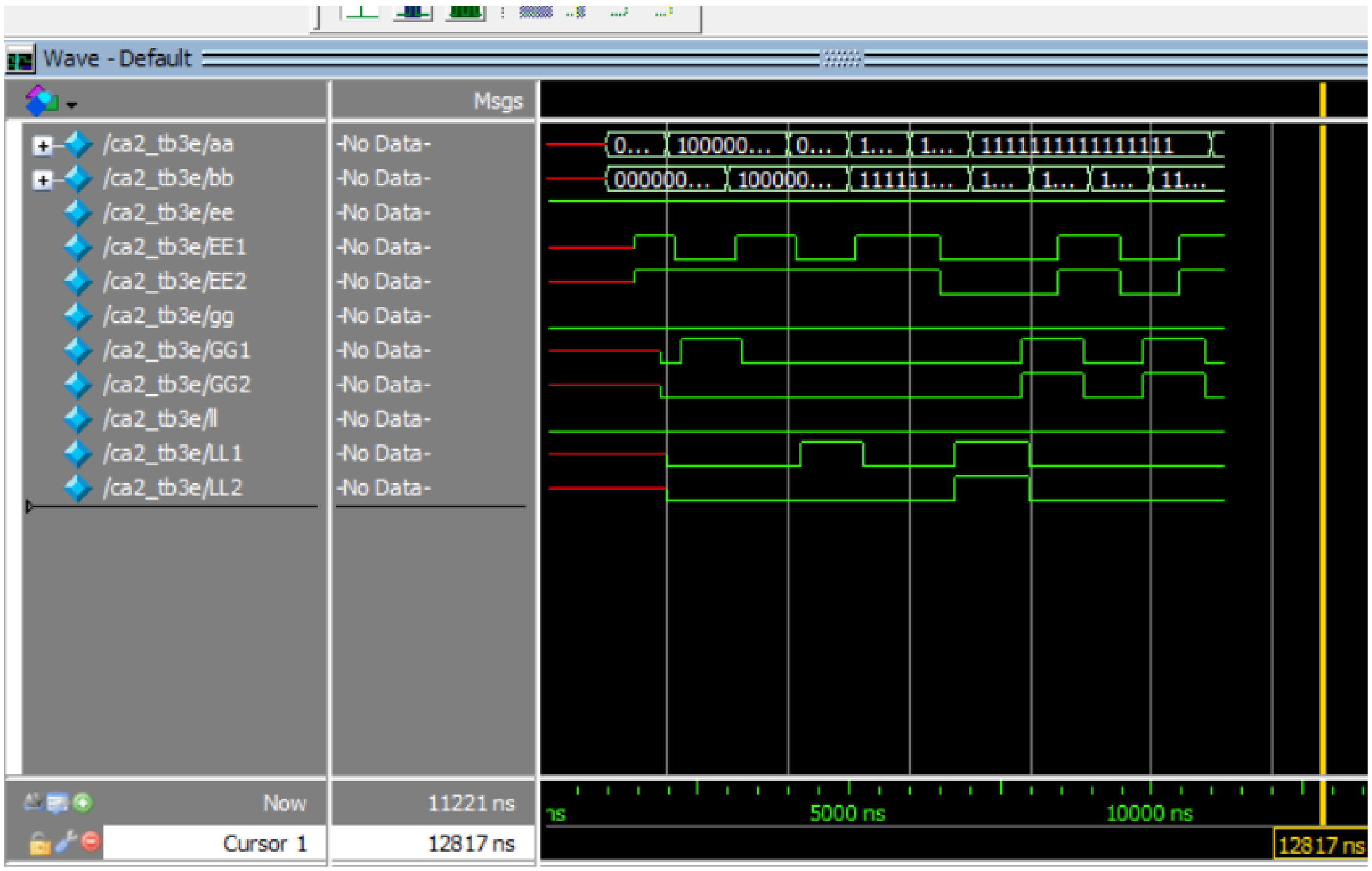


```
`timescale 1ns/1ns

module hex_comparator_assign(input [3:0][3:0]a,b,input e,l,g, output E,L,G);
    assign #(736,992) L = (a < b) | ((a == b) & (l == 1'b1));
    assign #(848,880) G = (a > b) | ((a == b) & g);
    assign #(464,496) E = (a == b) & (e == 1'b1);
endmodule
```

3-e

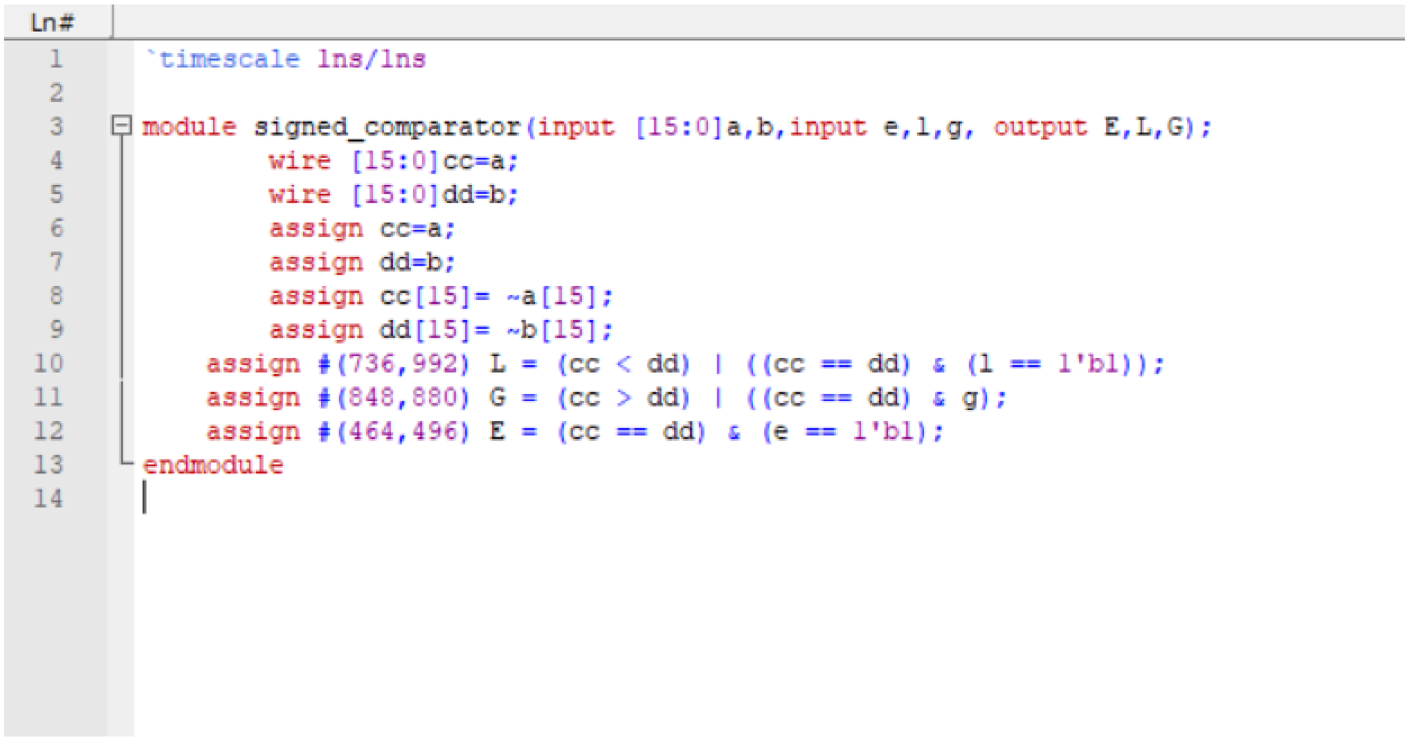


```
`timescale 1ns/1ns
module ca2_tb3e();
logic [15:0]aa;
logic [15:0]bb;
logic ee=1, ll=0,gg=0;
wire EE1,LL1,GG1 ,EE2,LL2,GG2;
hex_comparator CUT_112(.a(aa),.b(bb),.e(ee),.l(ll),.g(gg),.E(EE1),.L(LL1),.G(GG1));
hex_comparator_assign CUT_12(.a(aa),.b(bb),.e(ee),.l(ll),.g(gg),.E(EE2),.L(LL2),.G(GG2));

initial begin
#1 ee=1;ll=0;gg=0;
#1000 aa=16'b0000000000000000; bb=16'b0000000000000000;
#1000 aa=16'b1000000000000000; bb=16'b0000000000000000;
#1000 aa=16'b1000000000000000; bb=16'b1000000000000000;
#1000 aa=16'b0000000000000000; bb=16'b1000000000000000;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111111;
#1000 aa=16'b1111111111111110; bb=16'b1111111111111111;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111110;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111111;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111110;
#1000 aa=16'b1111111111111111; bb=16'b1111111111111111;
#1000 aa=16'b1111111111111110; bb=16'b1111111111111111;
#220 $stop;
end
endmodule
```



3-f

For this part we must just invert the sign bit and then compare them.



```
`timescale 1ns/1ns

module signed_comparator(input [15:0]a,b,input e,l,g, output E,L,G);
	wire [15:0]cc=a;
	wire [15:0]dd=b;
	assign cc=a;
	assign dd=b;
	assign cc[15]= ~a[15];
	assign dd[15]= ~b[15];
    assign #(736,992) L = (cc < dd) | ((cc == dd) & (l == 1'b1));
    assign #(848,880) G = (cc > dd) | ((cc == dd) & g);
    assign #(464,496) E = (cc == dd) & (e == 1'b1);
endmodule
```