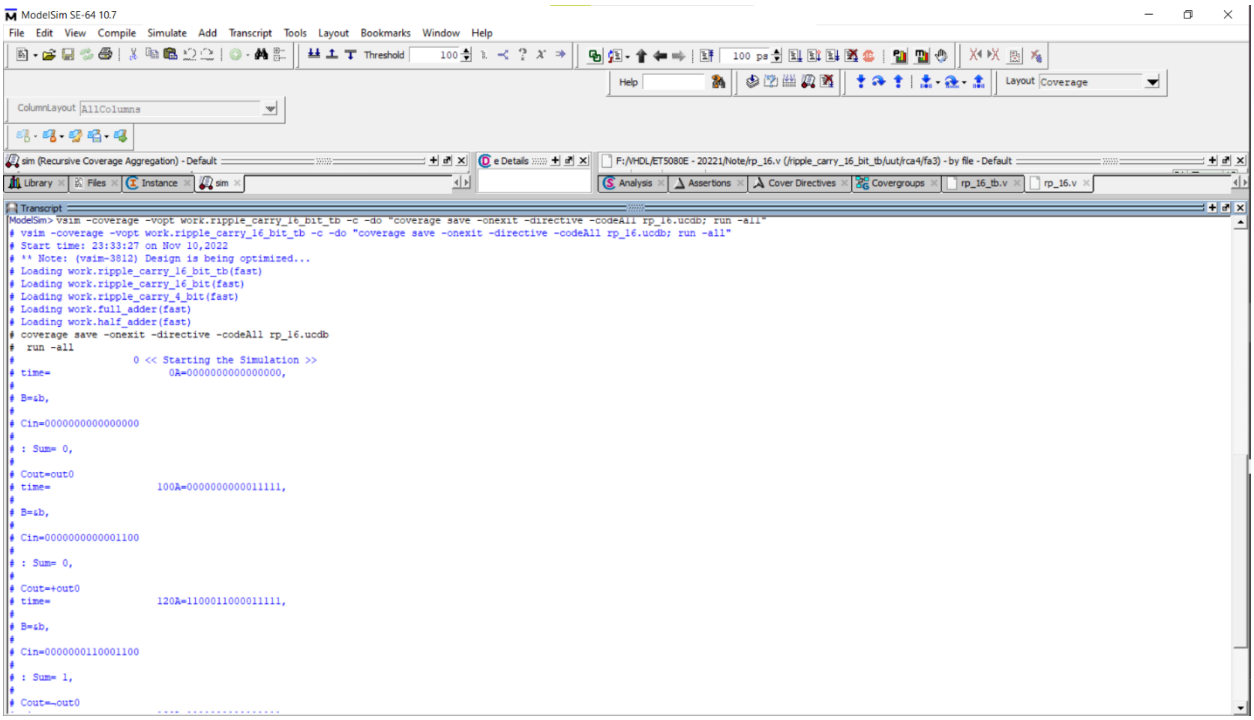
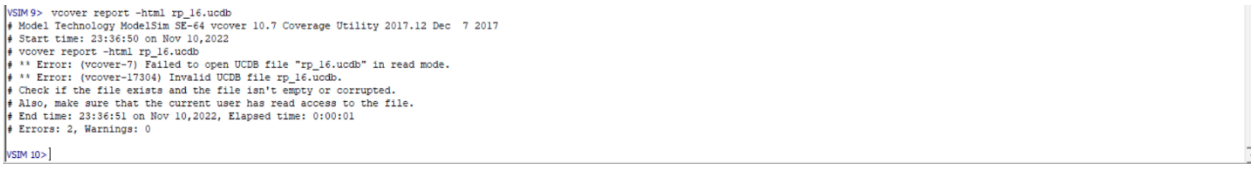
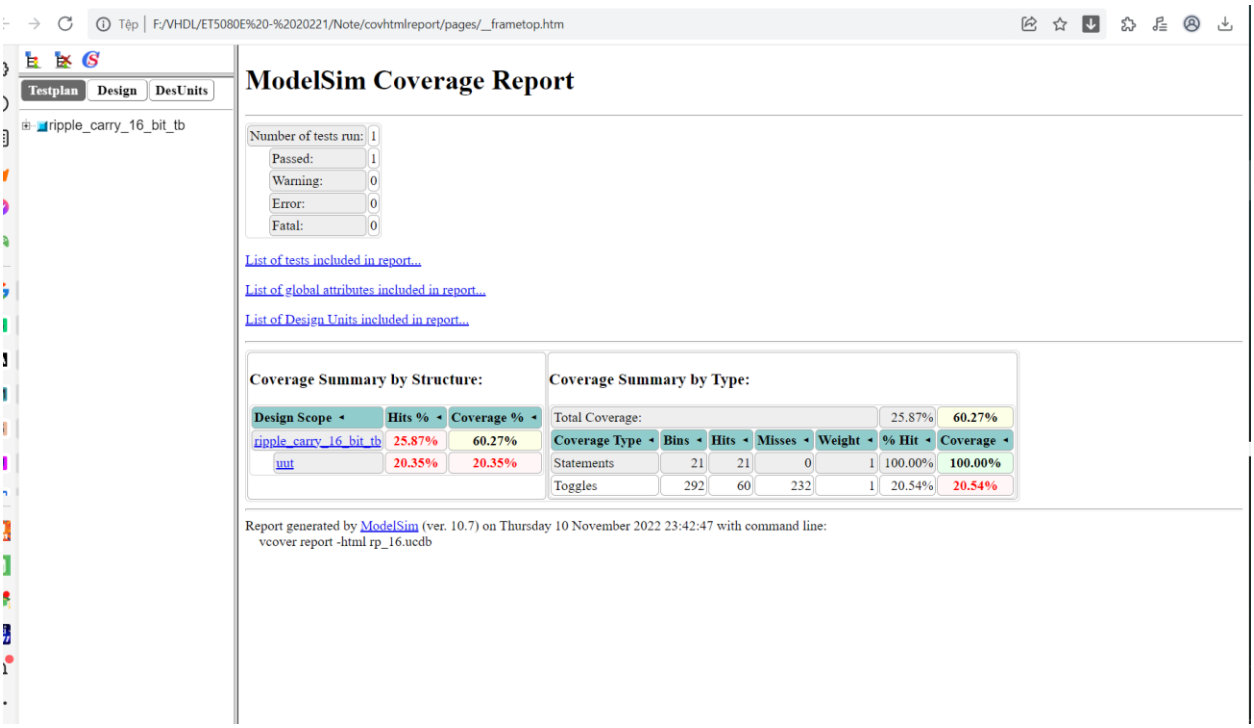
Prob 1


ModelSim SE-64 10.7
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
Transcript
ModelSim> vsim -coverage -vopt work.ripple_carry_16_bit_tb -c -do "coverage save -onexit -directive -codeAll rp_16.ucdb; run -all"
vsim -coverage -vopt work.ripple_carry_16_bit_tb -c -do "coverage save -onexit -directive -codeAll rp_16.ucdb; run -all"
Start time: 23:33:27 on Nov 10,2022
** Note: (vsim-3812) Design is being optimized...
Loading work.ripple_carry_16_bit_tb(fast)
Loading work.ripple_carry_16_bit(fast)
Loading work.ripple_carry_4_bit(fast)
Loading work.full_adder(fast)
Loading work.half_adder(fast)
coverage save -onexit -directive -codeAll rp_16.ucdb
run -all
0 << Starting the Simulation >>
time=   0A=0000000000000000,
#
B=&b,
#
Cin=0000000000000000
#
: Sum= 0,
#
Cout=out0
time=   100A=0000000000011111,
#
B=&b,
#
Cin=0000000000001100
#
: Sum= 0,
#
Cout=+out0
time=   120A=1100011000011111,
#
B=&b,
#
Cin=0000000110001100
#
: Sum= 1,
#
Cout=-out0


Cout=out0
time=   100A=0000000000011111,
#
B=&b,
#
Cin=0000000000001100
#
: Sum= 0,
#
Cout=+out0
time=   120A=1100011000011111,
#
B=&b,
#
Cin=0000000110001100
#
: Sum= 1,
#
Cout=-out0
time=   130A=1111111111111111,
#
B=&b,
#
Cin=0000000000000000
#
: Sum= 1,
#
Cout=out1


VSIM 9> vcover report -html rp_16.ucdb
Model Technology ModelSim SE-64 vcover 10.7 Coverage Utility 2017.12 Dec 7 2017
Start time: 23:36:50 on Nov 10,2022
vcover report -html rp_16.ucdb
** Error: (vcover-7) Failed to open UCDB file "rp_16.ucdb" in read mode.
** Error: (vcover-17304) Invalid UCDB file rp_16.ucdb.
Check if the file exists and the file isn't empty or corrupted.
Also, make sure that the current user has read access to the file.
End time: 23:36:51 on Nov 10,2022, Elapsed time: 0:00:01
Errors: 2, Warnings: 0
VSIM 10>

covhtmlreport	11/10/2022 11:36 PM	File folder	
work	11/10/2022 11:33 PM	File folder	
modelsim.ini	11/10/2022 11:15 PM	Configuration setti...	96 KB
rp_16.cr.mti	11/6/2022 1:28 PM	MTI File	1 KB
rp_16.mpf	11/6/2022 1:16 PM	MPF File	99 KB
rp_16.v	11/6/2022 11:44 AM	V File	2 KB
rp_16.v.bak	11/6/2022 11:40 AM	BAK File	1 KB
rp_16_tb.v	11/6/2022 11:45 AM	V File	1 KB
rp_16_tb.v.bak	11/6/2022 11:41 AM	BAK File	2 KB



Prob 2

Full Adder

Verilog

F:\VHDL\ET5080E - 20221\HW5\Full Adder n-bit\fa_nbit.v - Default

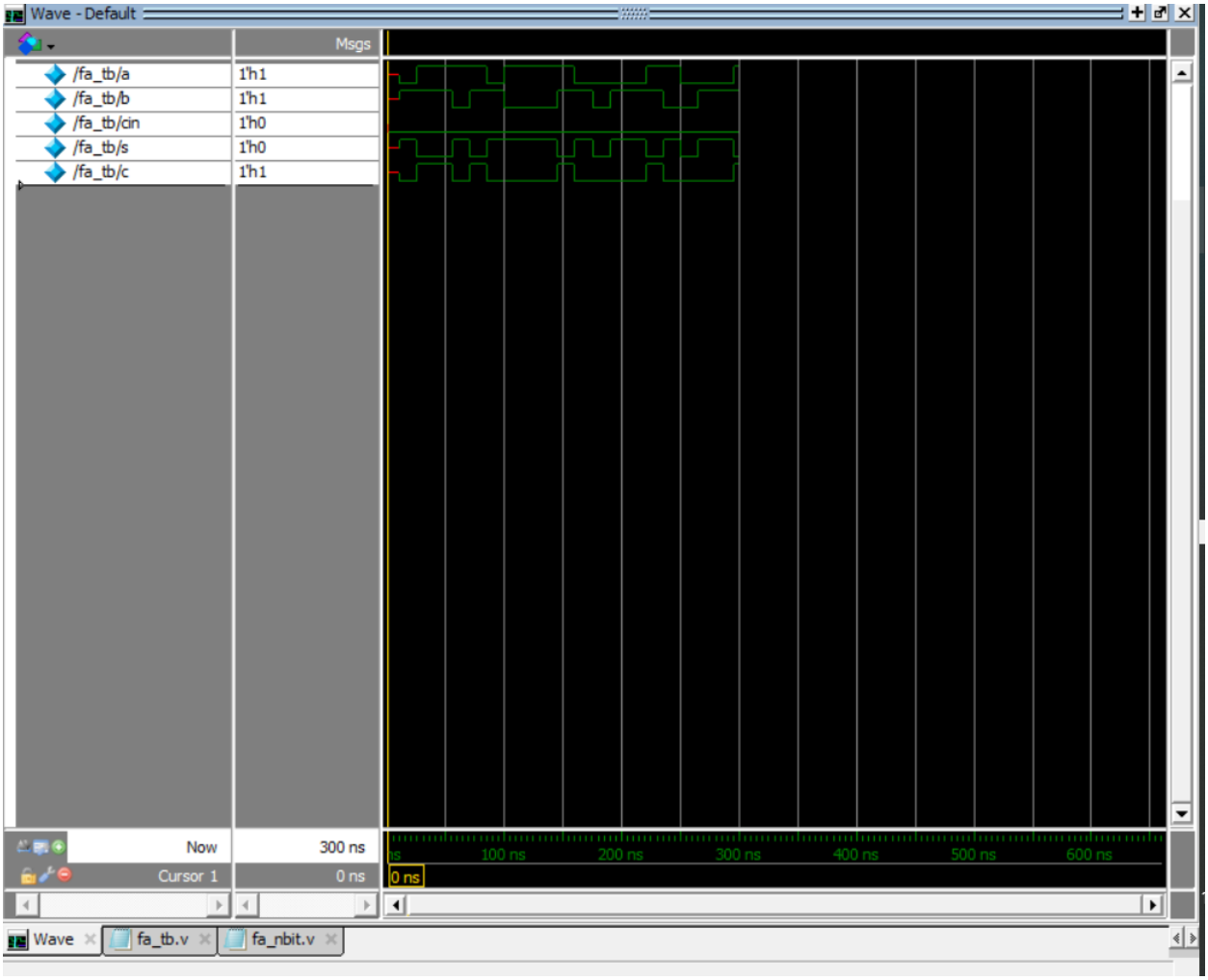
```
Ln#
 1  module fa(a,b,cin,s,c);
 2  input a,b,cin;
 3  output s,c;
 4  wire n1,n2,n3;
 5  xor x1(s,a,b,cin);
 6  and a1(n1,a,b);
 7  and a2(n2,b,cin);
 8  and a3(n3,a,cin);
 9  or o1(c,n1,n2,n3);
10  endmodule : fa
11
```

Testbench

F:/VHDL/ET5080E - 20221/HW5/Full Adder n-bit/fa_tb.v (/fa_tb) - Default

```
Ln#
 1  module fa_tb();
 2  reg a,b,cin;
 3  wire s,c;
 4
 5  fa dut(
 6  .a(a),
 7  .b(b),
 8  .cin(cin),
 9  .s(s),
10  .c(c)
11  );
12
13  initial begin
14  cin =1'b0;
15  repeat(20) begin
16  #10;
17  a = $random();
18  b = $random();
19  #5;
20  $display(":\t\t a =",a,"\t\t b =",b, "\t\t cin= ",cin,"\t\t s = ", s, "\t\t c= ", c);
21  end
22  $stop;
23  end
24
25  endmodule : fa_tb
26
```

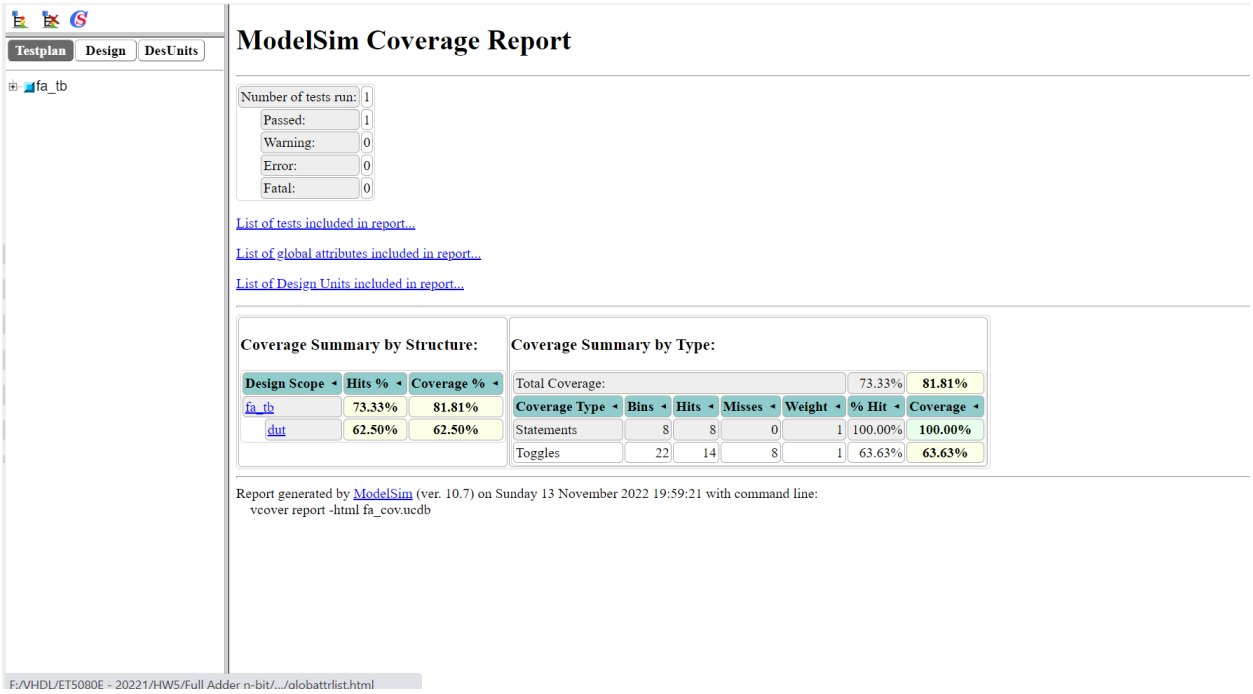
Waveform



Transcript

```
add wave -position insertpoint sim:/fa_tb/*
VSIM 11> run
# :               a =0            b =1            cin= 0          s = 1           c= 0
# :               a =1            b =1            cin= 0          s = 0           c= 1
# :               a =1            b =1            cin= 0          s = 0           c= 1
# :               a =1            b =0            cin= 0          s = 1           c= 0
# :               a =1            b =1            cin= 0          s = 0           c= 1
# :               a =0            b =1            cin= 0          s = 1           c= 0
# :               a =1            b =0            cin= 0          s = 1           c= 0
# :               a =1            b =0            cin= 0          s = 1           c= 0
# :               a =1            b =0            cin= 0          s = 1           c= 0
# :               a =1            b =1            cin= 0          s = 0           c= 1
# :               a =0            b =1            cin= 0          s = 1           c= 0
# :               a =0            b =0            cin= 0          s = 0           c= 0
# :               a =0            b =1            cin= 0          s = 1           c= 0
# :               a =0            b =1            cin= 0          s = 1           c= 0
# :               a =1            b =1            cin= 0          s = 0           c= 1
# :               a =1            b =0            cin= 0          s = 1           c= 0
# :               a =0            b =0            cin= 0          s = 0           c= 0
# :               a =0            b =1            cin= 0          s = 1           c= 0
# :               a =0            b =1            cin= 0          s = 1           c= 0
# :               a =1            b =1            cin= 0          s = 0           c= 1
# ** Note: $stop    : F:/VHDL/ET5080E - 20221/HW5/Full Adder n-bit/fa_tb.v(22)
#    Time: 300 ns  Iteration: 0  Instance: /fa_tb
# Break in Module fa_tb at F:/VHDL/ET5080E - 20221/HW5/Full Adder n-bit/fa_tb.v line 22
```

Report



Testplan | Design | DesUnits

fa_tb

ModelSim Coverage Report

Number of tests run:	1
Passed:	1
Warning:	0
Error:	0
Fatal:	0

List of tests included in report...

List of global attributes included in report...

List of Design Units included in report...

Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
fa_tb	73.33%	81.81%
dut	62.50%	62.50%

Coverage Summary by Type:

Total Coverage:					73.33%	81.81%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Statements	8	8	0	1	100.00%	100.00%
Toggles	22	14	8	1	63.63%	63.63%

Report generated by ModelSim (ver. 10.7) on Sunday 13 November 2022 19:59:21 with command line:
vcover report -html fa_cov.ucdb

F:/VHDL/ET5080E - 20221/HW5/Full Adder n-bit/.../globattrlist.html

Carry Look Ahead Adder

Verilog

CLA Verilog code

```
`timescale 1ns / 1ps



module full_adder_gp(
    input  wire A,
    input  wire B,
    input  wire Ci,
    output wire S,
    output wire Co
    );

    assign S  = A ^ B ^ Ci;

endmodule
```

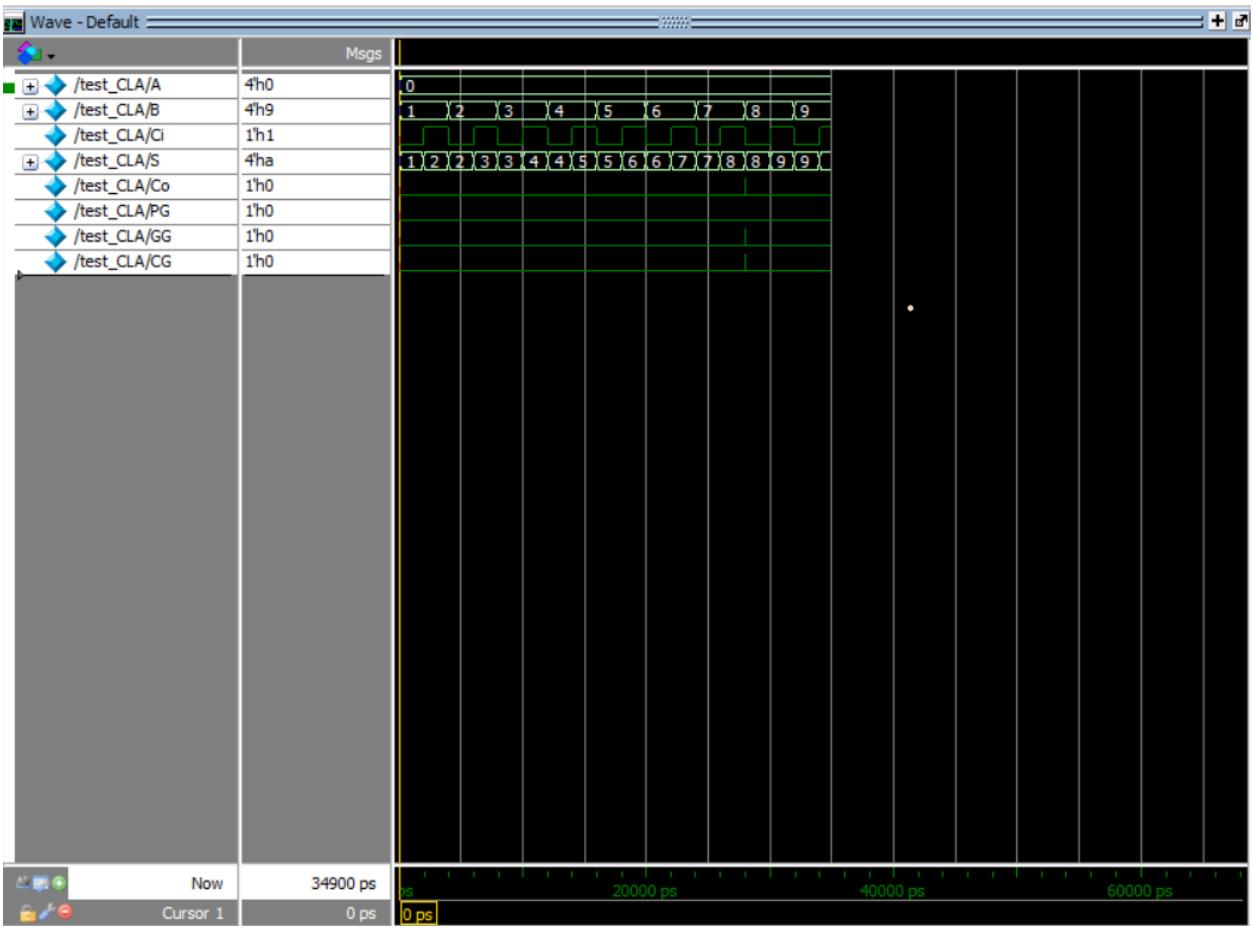
```
`timescale 1ns / 1ps



module GP(
    input  wire A,
    input  wire B,
    input  wire Ci,
    output reg  G,
    output reg  P,
    output reg  C
    );

    always @(*)
        begin
            G = A & B;
            P = A | B;  // A xor B can also be used
            C = G | P & Ci;
        end
endmodule

`timescale 1ns / 1ps

module CLA (A,B,Ci,S,Co,PG,GG,CG);

    parameter N = 4;
    input  wire [N-1:0] A;
    input  wire [N-1:0] B;
    input  wire Ci;
    output wire [N-1:0] S;
```

```
    output wire Co;
    output reg  PG,GG,CG;

    wire [N-1:0] P;
    wire [N-1:0] G;

    wire [N:0] C;



    assign C[0] = Ci;

    genvar i;
    generate
      for (i=0; i < N; i=i+1)
       begin : Build
       full_adder_gp COMP1(.A(A[i]), .B(B[i]), .Ci(C[i]), .S(S[i]));
       GP            COMP2(.A(A[i]), .B(B[i]), .Ci(C[i]), .G(G[i]), .P(P[i]), .C(C[i+1]));
       end
     endgenerate

     assign Co = C[N];

     always@(*)
       begin
          PG = &P;
          GG = C[N];
          CG = GG | PG & Ci;
       end

endmodule
```

Testbench

```
`timescale 1ns / 1ps


module test_CLA();
    parameter N = 4;
// inputs to the DUT
    reg [N-1:0] A = 4'h0;
    reg [N-1:0]B = 4'h0;
    reg Ci = 0;
// outputs from the DUT
    wire [N-1:0]S;
    wire Co;
    wire PG,GG,CG;

CLA DUT ( .A(A), .B(B), .Ci(Ci), .S(S), .Co(Co), .PG(PG), .GG(GG), .CG(CG) );

initial
    begin
        $display($time, " << Simulation Results >>");
        $monitor($time, "A = %b, B = %b, Ci = %b, S = %b, Co = %b, PG = %b, GG = %b, CG = %b", A, B, Ci, S, Co,PG,GG,CG);
    end

initial
    begin
        forever begin
          if (A <= 4'b1111)
            begin
                if(B < 4'b1111)
                    begin
                        B  = B + 4'b0001;
                        Ci = 0;
                        #2;
                        Ci =1;
                        #2;
                    end else begin
                        B = 4'b0000;
                        A = A + 4'b0001;
                    end
            end else begin
                $finish;
            end
        end
    end

endmodule

```

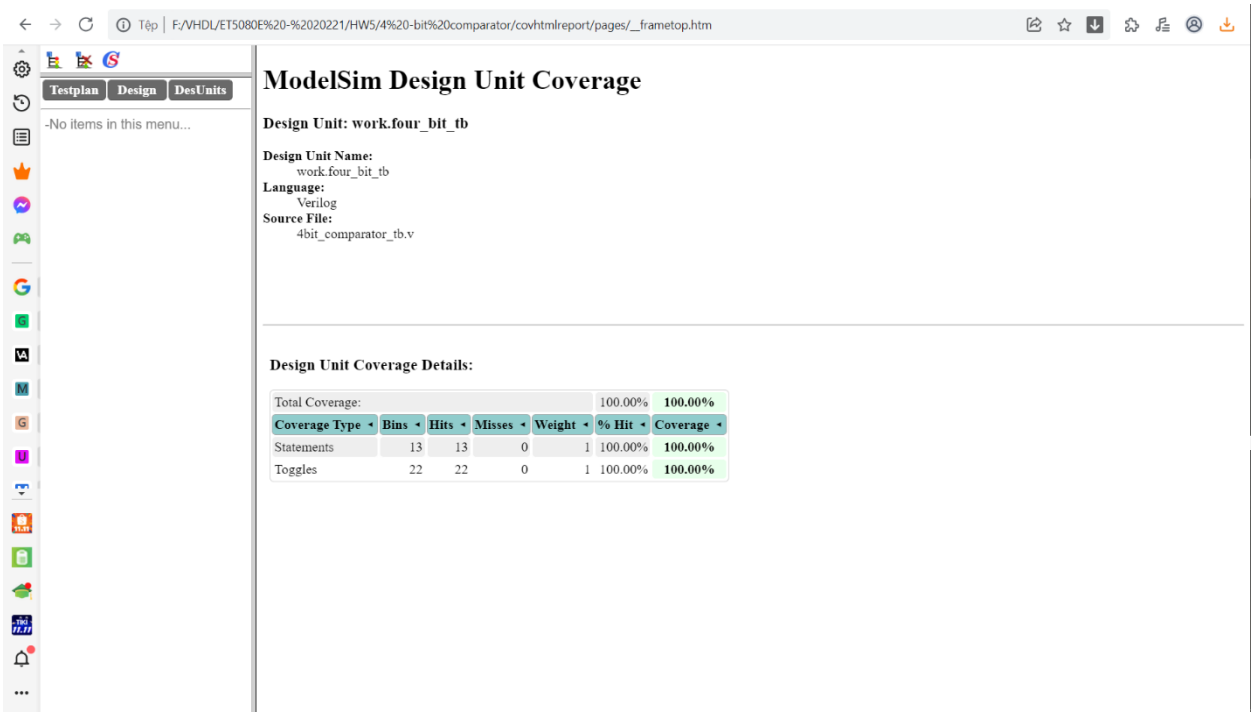
Wave



Transcipt

```
add wave -position insertpoint sim:/test_CLA/*
VSIM 6> run
#                    0 << Simulation Results >>
#                    0A = 0000, B = 0001, Ci = 0, S = 0001, Co = 0, PG = 0, GG = 0, CG = 0
run
run
run
run
#                    2A = 0000, B = 0001, Ci = 1, S = 0010, Co = 0, PG = 0, GG = 0, CG = 0
run
run
#                    4A = 0000, B = 0010, Ci = 0, S = 0010, Co = 0, PG = 0, GG = 0, CG = 0
run
#                    6A = 0000, B = 0010, Ci = 1, S = 0011, Co = 0, PG = 0, GG = 0, CG = 0
#                    8A = 0000, B = 0011, Ci = 0, S = 0011, Co = 0, PG = 0, GG = 0, CG = 0
run
#                   10A = 0000, B = 0011, Ci = 1, S = 0100, Co = 0, PG = 0, GG = 0, CG = 0
#                   12A = 0000, B = 0100, Ci = 0, S = 0100, Co = 0, PG = 0, GG = 0, CG = 0
#                   14A = 0000, B = 0100, Ci = 1, S = 0101, Co = 0, PG = 0, GG = 0, CG = 0
run
#                   16A = 0000, B = 0101, Ci = 0, S = 0101, Co = 0, PG = 0, GG = 0, CG = 0
#                   18A = 0000, B = 0101, Ci = 1, S = 0110, Co = 0, PG = 0, GG = 0, CG = 0
run
#                   20A = 0000, B = 0110, Ci = 0, S = 0110, Co = 0, PG = 0, GG = 0, CG = 0
#                   22A = 0000, B = 0110, Ci = 1, S = 0111, Co = 0, PG = 0, GG = 0, CG = 0
#                   24A = 0000, B = 0111, Ci = 0, S = 0111, Co = 0, PG = 0, GG = 0, CG = 0
run
#                   26A = 0000, B = 0111, Ci = 1, S = 1000, Co = 0, PG = 0, GG = 0, CG = 0
#                   28A = 0000, B = 1000, Ci = 0, S = 1000, Co = 0, PG = 0, GG = 0, CG = 0
VSIM 7> run
#                   30A = 0000, B = 1000, Ci = 1, S = 1001, Co = 0, PG = 0, GG = 0, CG = 0
#                   32A = 0000, B = 1001, Ci = 0, S = 1001, Co = 0, PG = 0, GG = 0, CG = 0
#                   34A = 0000, B = 1001, Ci = 1, S = 1010, Co = 0, PG = 0, GG = 0, CG = 0

VSIM 7>
```

Report



Majority

Verilog

F:/VHDL/ET5080E - 20221/HW5/Majority n-bit/major.v (/tb/dut) - Default

```
Ln#
 1  module majority(a,b,c,z);
 2  output z;
 3  input a,b,c;
 4  wire n1,n2,n3;
 5
 6  assign n1 = a & b;
 7  assign n2 = b & c;
 8  assign n3 = a & c;
 9  assign z = n1 | n2 | n3;
10  endmodule : majority
11
12
13
```

Testbench

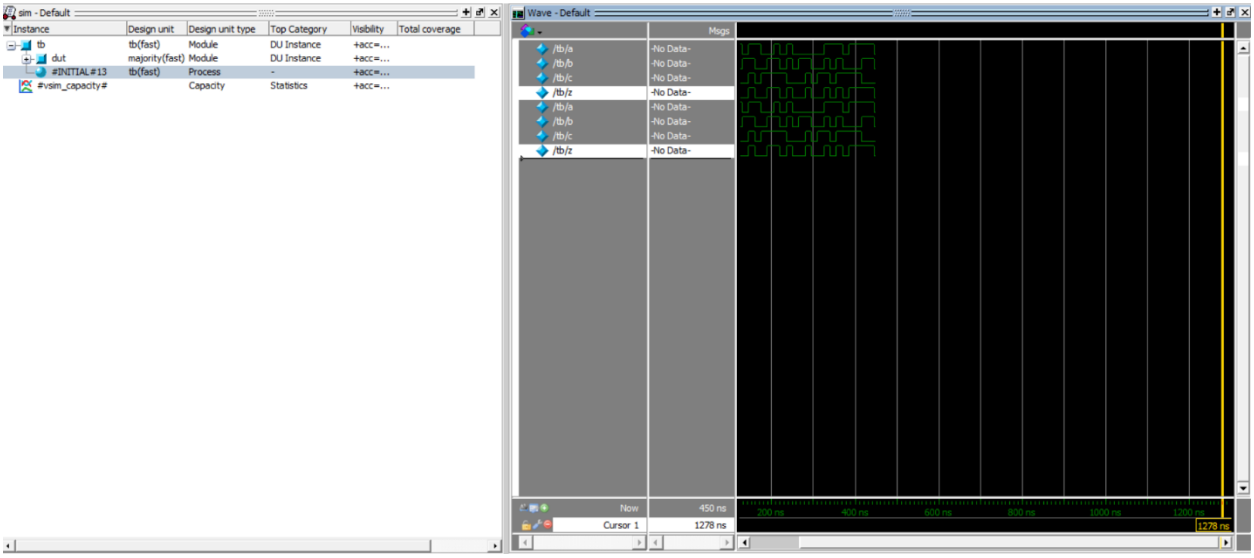
F:/VHDL/ET5080E - 20221/HW5/Majority n-bit/majority_tb.v (/tb) - Default

```
Ln#
 1  module tb ();
 2
 3  reg a, b, c;
 4  wire z;
 5
 6  majority dut(
 7    .a(a),
 8    .b(b),
 9    .c(c),
10    .z(z)
11  );
12
13  initial begin
14    repeat(30) begin
15      #10;
16      a = $random();
17      b = $random();
18      c = $random();
19    #5;
20    $display($time,":\t\t a = ",a,"\t\t b = ",b,"\t\t c= ",c,"\t\t z =",z);
21  end
22    $stop;
23  end
24
25  endmodule : tb
26
```

Script

```
VSIM 9> run
#                   15:   a = 0   b = 1          c= 1          z =1
#                   30:   a = 1   b = 1          c= 1          z =1
#                   45:   a = 1   b = 0          c= 1          z =1
#                   60:   a = 1   b = 0          c= 1          z =1
#                   75:   a = 1   b = 0          c= 1          z =1
#                   90:   a = 0   b = 1          c= 0          z =0
#                  105:   a = 1   b = 1          c= 0          z =1
#                  120:   a = 1   b = 0          c= 0          z =0
#                  135:   a = 0   b = 1          c= 0          z =0
#                  150:   a = 1   b = 1          c= 1          z =1
#                  165:   a = 1   b = 0          c= 0          z =0
#                  180:   a = 0   b = 0          c= 1          z =0
#                  195:   a = 0   b = 1          c= 1          z =1
#                  210:   a = 1   b = 1          c= 1          z =1
#                  225:   a = 0   b = 0          c= 1          z =0
#                  240:   a = 1   b = 1          c= 0          z =1
#                  255:   a = 0   b = 0          c= 0          z =0
#                  270:   a = 0   b = 1          c= 0          z =0
#                  285:   a = 0   b = 1          c= 1          z =1
#                  300:   a = 0   b = 0          c= 0          z =0
#                  315:   a = 0   b = 0          c= 1          z =0
#                  330:   a = 1   b = 1          c= 1          z =1
#                  345:   a = 1   b = 0          c= 0          z =0
#                  360:   a = 1   b = 1          c= 1          z =1
#                  375:   a = 0   b = 0          c= 1          z =0
#                  390:   a = 1   b = 0          c= 1          z =1
#                  405:   a = 1   b = 0          c= 1          z =1
#                  420:   a = 1   b = 1          c= 0          z =1
#                  435:   a = 1   b = 1          c= 0          z =1
#                  450:   a = 0   b = 0          c= 0          z =0
# ** Note: $stop    : F:/VHDL/ET5080E - 20221/HW5/Majority n-bit/majority_tb.v(22)
#    Time: 450 ns  Iteration: 0  Instance: /tb
```

Wave



Report

Testplan | Design | DesUnits

tb

ModelSim Coverage Report

Number of tests run:	1
Passed:	1
Warning:	0
Error:	0
Fatal:	0

List of tests included in report...

List of global attributes included in report...

List of Design Units included in report...

Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
tb	100.00%	100.00%
dut	100.00%	100.00%

Coverage Summary by Type:

Total Coverage:					100.00%	100.00%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Statements	12	12	0	1	100.00%	100.00%
FEC Expressions	9	9	0	1	100.00%	100.00%
Toggles	20	20	0	1	100.00%	100.00%

Report generated by ModelSim (ver. 10.7) on Thursday 17 November 2022 23:25:40 with command line:
vcover report -html majority_cov.ucdb

Gray counter

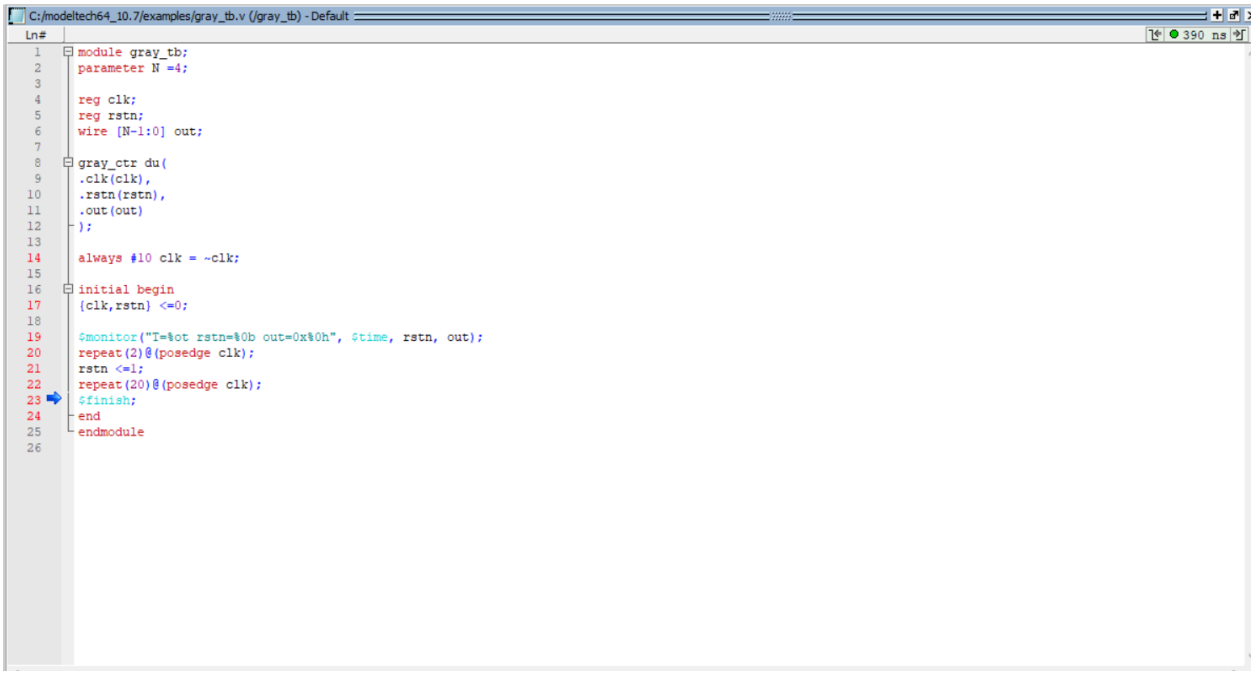
Verilog

C:/modeltech64_10.7/examples/gray1.v (/gray_tb/du) - Default

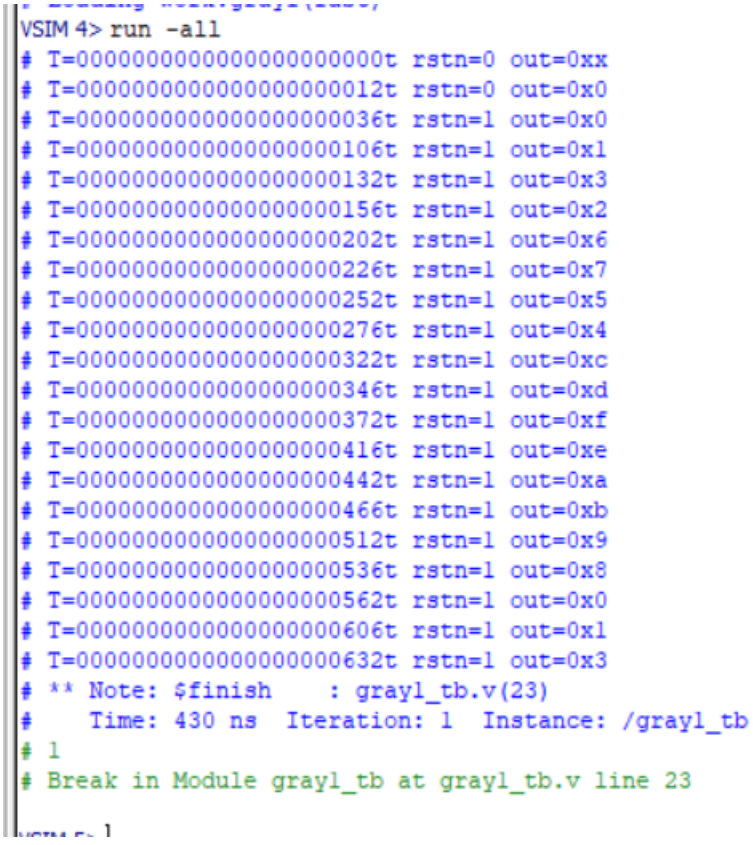
Ln# 390 ns

```
1  module gray_ctr
2  #(parameter N = 4)
3  (input clk,
4   input rstn,
5   output reg [N-1:0] out);
6
7  reg [N-1:0] q;
8
9  always @(posedge clk) begin
10 if (!rstn) begin
11 q <= 0;
12 out <= 0;
13 end else begin
14 q <= q+1;
15
16 `ifdef FOR_LOOP
17 for(int i=0; i < N-1; i = i+1) begin
18 out[i] <= q[i+1] ^ q[i];
19 end
20 out[N-1] <= q[N-1];
21 `else
22 out <= {q[N-1], q[N-1:1] ^ q[N-2:0]};
23 `endif
24 end
25 end
26 endmodule
27
28
```

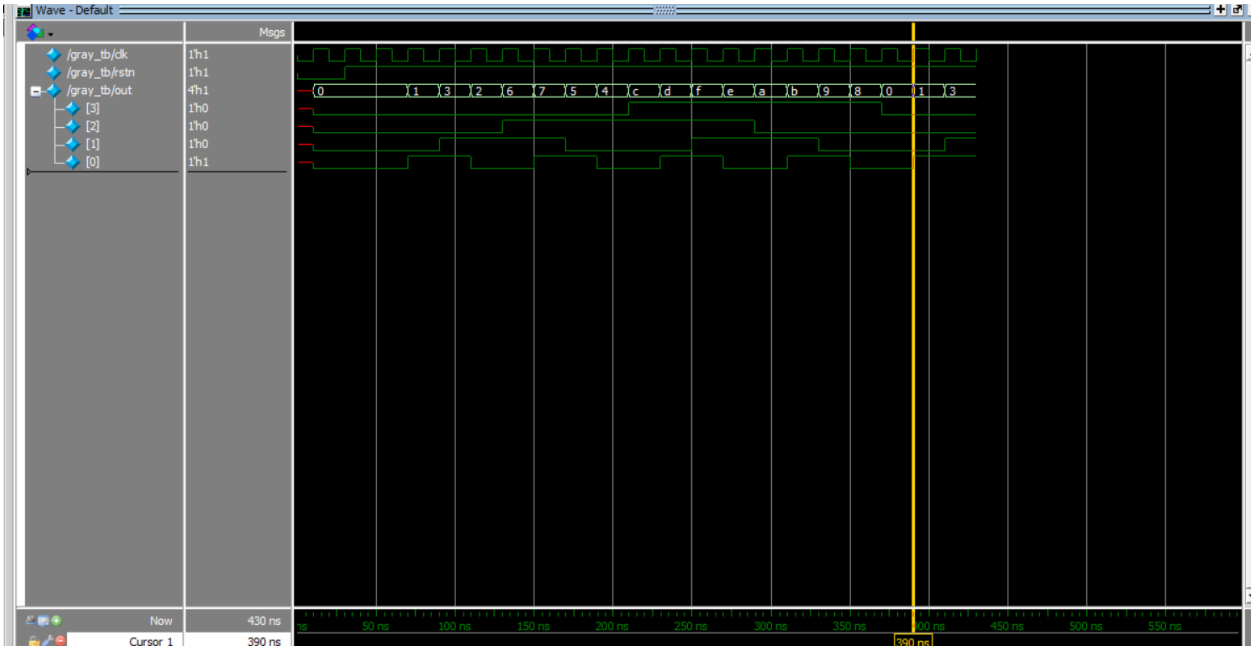
Testbench


C:/modeltech64_10.7/examples/gray_tb.v (/gray_tb) - Default
390 ns
module gray_tb;
parameter N =4;
reg clk;
reg rstn;
wire [N-1:0] out;
gray_ctr du(
.clk(clk),
.rstn(rstn),
.out(out)
);
always #10 clk = ~clk;
initial begin
{clk,rstn} <=0;
$monitor("T=%0t rstn=%0b out=0x%0h", $time, rstn, out);
repeat(2)@(posedge clk);
rstn <=1;
repeat(20)@(posedge clk);
$finish;
end
endmodule

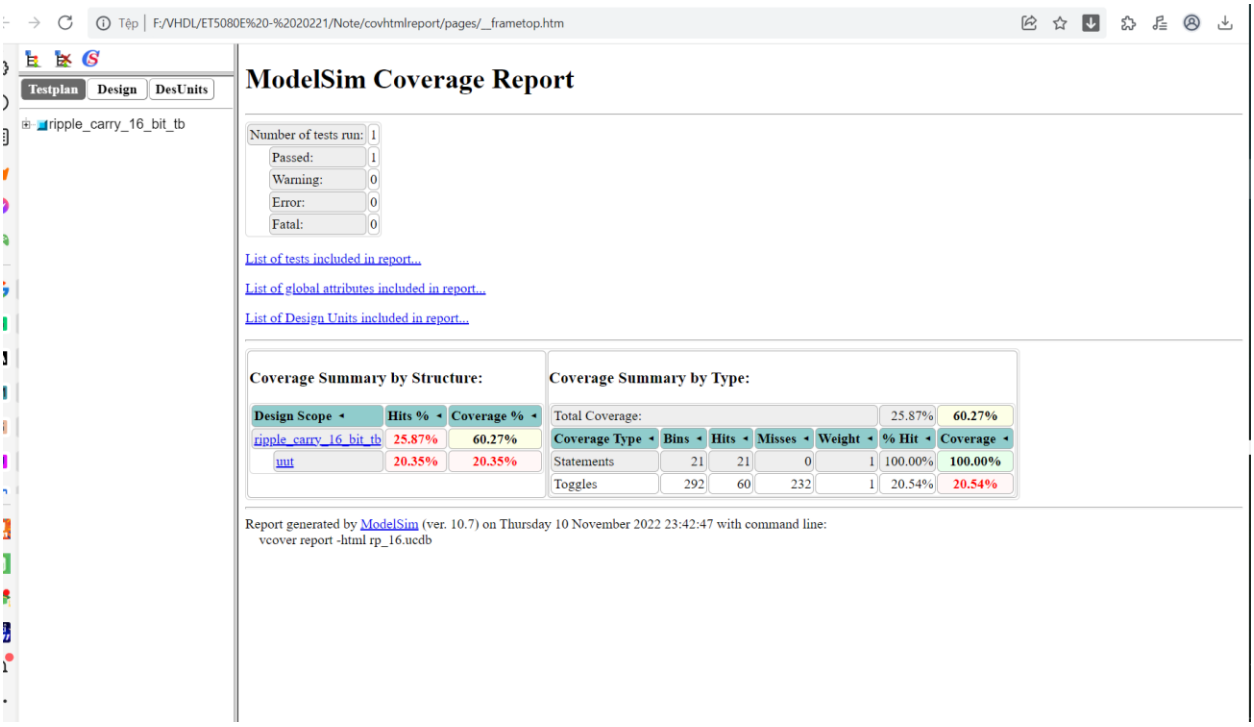
Script


VSIM 4> run -all
T=00000000000000000000t rstn=0 out=0xx
T=00000000000000000012t rstn=0 out=0x0
T=00000000000000000036t rstn=1 out=0x0
T=00000000000000000106t rstn=1 out=0x1
T=00000000000000000132t rstn=1 out=0x3
T=00000000000000000156t rstn=1 out=0x2
T=00000000000000000202t rstn=1 out=0x6
T=00000000000000000226t rstn=1 out=0x7
T=00000000000000000252t rstn=1 out=0x5
T=00000000000000000276t rstn=1 out=0x4
T=00000000000000000322t rstn=1 out=0xc
T=00000000000000000346t rstn=1 out=0xd
T=00000000000000000372t rstn=1 out=0xf
T=00000000000000000416t rstn=1 out=0xe
T=00000000000000000442t rstn=1 out=0xa
T=00000000000000000466t rstn=1 out=0xb
T=00000000000000000512t rstn=1 out=0x9
T=00000000000000000536t rstn=1 out=0x8
T=00000000000000000562t rstn=1 out=0x0
T=00000000000000000606t rstn=1 out=0x1
T=00000000000000000632t rstn=1 out=0x3
** Note: $finish : grayl_tb.v(23)
Time: 430 ns Iteration: 1 Instance: /grayl_tb
1
Break in Module grayl_tb at grayl_tb.v line 23

Wave



Report



4 -bit comparator

Verilog

```
`timescale 1ns / 1ps

module four_bit_comp(G, L, E, a0, a1, a2, a3, b0, b1, b2, b3);
    input a0, a1, a2, a3, b0, b1, b2, b3;
    output G, L, E;

    wire g1, g2, l1, l2, e1, e2;

    two_bit_comp comp1(g2, l2, e2, a3, a2, b3, b2);
    two_bit_comp comp2(g1, l1, e1, a1, a0, b1, b0);

    assign G = (g2)|(e2&g1);
    assign L = (l2)|(e2&l1);
    assign E = e1&e2;
endmodule

module two_bit_comp(G, L, E, a1, a0, b1, b0);
    input a1, a0, b1, b0;
    output G, L, E;

    assign G = ((a1)&(~b1))|(((a1)~^(b1))&((a0)&(~b0)));
    assign L = ((~a1)&(b1))|(((a1)~^(b1))&((~a0)&(b0)));
    assign E = (a1~^b1)&(a0~^b0);
endmodule
```

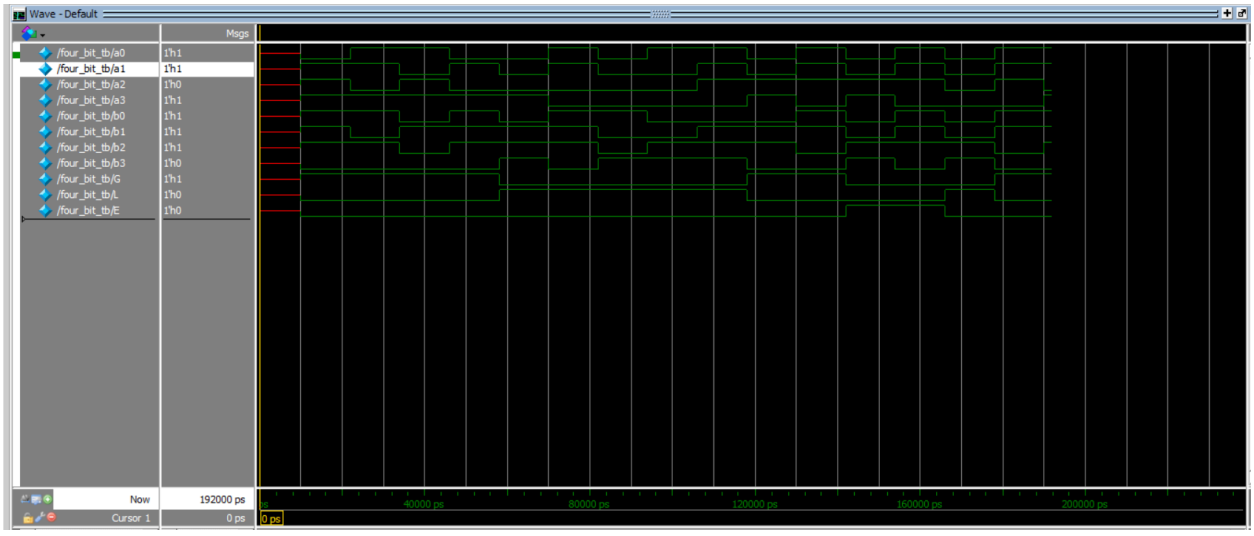
Testbench

```
`timescale 1ns / 1ps

module four_bit_tb();
    reg a0, a1, a2, a3;
    reg b0, b1, b2, b3;
    wire G, L, E;

    four_bit_comp dut(G, L, E, a0, a1, a2, a3, b0, b1, b2, b3);

    initial begin
        repeat(16) begin
            #10;
            a0 = $random();
            a1 = $random();
            a2 = $random();
            a3 = $random();
            b0 = $random();
            b1 = $random();
            b2 = $random();
            b3 = $random();
            #2;
        $display($time,":\t\t a0 = ",a0,"\t\t a1 = ",a1,"\t\t a2 = ",a2,"\t\t a3 = ",a3,"\t\t b0 = ",b0,"\t\t b1 = ",b1,"\t\t b2 = ",b2,"\t\t b3 = ",b3,"\t\t G= ",G,"\t\t L =",L,"\t\t E=",E);
    end
$stop;
end
endmodule : four_bit_tb
```

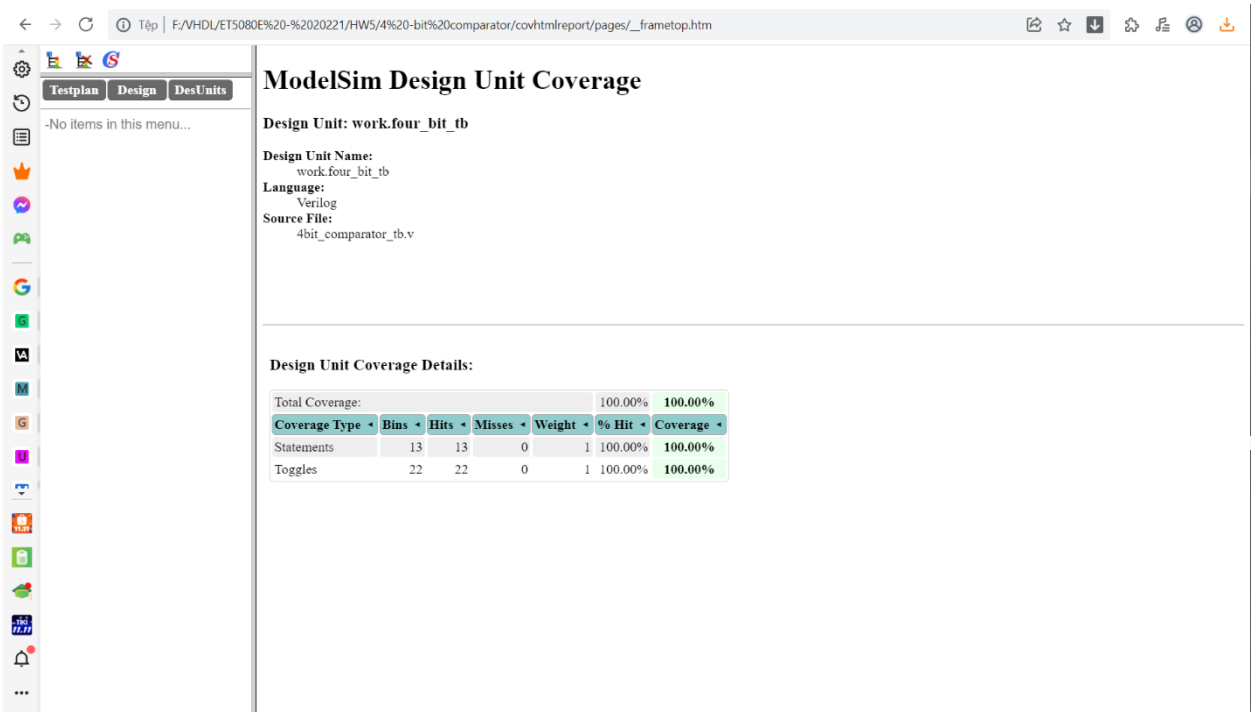
Script

```
VSIM 50> run
#                  12:   a0 = 0   a1 = 1        a2 = 1          a3 = 1          b0 = 1          b1 = 1          b2 = 1
 b3 = 0          G= 1             L =0          E=0
#                  24:   a0 = 1   a1 = 1        a2 = 0          a3 = 1          b0 = 1          b1 = 0          b2 = 1
 b3 = 0          G= 1             L =0          E=0
#                  36:   a0 = 1   a1 = 0        a2 = 1          a3 = 1          b0 = 0          b1 = 1          b2 = 0
 b3 = 0          G= 1             L =0          E=0
#                  48:   a0 = 0   a1 = 1        a2 = 0          a3 = 1          b0 = 1          b1 = 1          b2 = 1
 b3 = 0          G= 1             L =0          E=0
#                  60:   a0 = 0   a1 = 0        a2 = 0          a3 = 1          b0 = 0          b1 = 1          b2 = 1
 b3 = 1          G= 0             L =1          E=0
#                  72:   a0 = 1   a1 = 1        a2 = 0          a3 = 0          b0 = 1          b1 = 1          b2 = 1
 b3 = 0          G= 0             L =1          E=0
#                  84:   a0 = 0   a1 = 0        a2 = 0          a3 = 0          b0 = 1          b1 = 0          b2 = 0
 b3 = 1          G= 0             L =1          E=0
#                  96:   a0 = 1   a1 = 0        a2 = 0          a3 = 0          b0 = 0          b1 = 0          b2 = 1
 b3 = 1          G= 0             L =1          E=0
#                 108:   a0 = 1   a1 = 1        a2 = 1          a3 = 0          b0 = 0          b1 = 1          b2 = 1
 b3 = 1          G= 0             L =1          E=0
#                 120:   a0 = 0   a1 = 0        a2 = 1          a3 = 1          b0 = 0          b1 = 1          b2 = 1
 b3 = 0          G= 1             L =0          E=0
#                 132:   a0 = 1   a1 = 1        a2 = 1          a3 = 0          b0 = 1          b1 = 1          b2 = 0
 b3 = 0          G= 1             L =0          E=0
#                 144:   a0 = 0   a1 = 0        a2 = 1          a3 = 1          b0 = 0          b1 = 0          b2 = 1
 b3 = 1          G= 0             L =0          E=1
#                 156:   a0 = 1   a1 = 1        a2 = 1          a3 = 0          b0 = 1          b1 = 1          b2 = 1
 b3 = 0          G= 0             L =0          E=1
#                 168:   a0 = 0   a1 = 0        a2 = 0          a3 = 0          b0 = 0          b1 = 0          b2 = 1
 b3 = 1          G= 0             L =1          E=0
#                 180:   a0 = 1   a1 = 1        a2 = 1          a3 = 0          b0 = 1          b1 = 1          b2 = 0
 b3 = 0          G= 1             L =0          E=0
#                 192:   a0 = 1   a1 = 1        a2 = 0          a3 = 1          b0 = 1          b1 = 1          b2 = 1
 b3 = 0          G= 1             L =0          E=0
# ** Note: $stop    : C:/modeltech64_10.7/examples/4bit_comparator_tb.v(24)
#    Time: 192 ns  Iteration: 0  Instance: /four_bit_tb
```

Wave



Report


Tệp | F:/VHDL/ET5080E%20-%2020221/HW5/4%20-bit%20comparator/covhtmlreport/pages/__frametop.htm
Testplan
Design
DesUnits
-No items in this menu...
ModelSim Design Unit Coverage
Design Unit: work.four_bit_tb
Design Unit Name:
work.four_bit_tb
Language:
Verilog
Source File:
4bit_comparator_tb.v
Design Unit Coverage Details:
Total Coverage: 100.00% 100.00%
Coverage Type | Bins | Hits | Misses | Weight | % Hit | Coverage
Statements 13 13 0 1 100.00% 100.00%
Toggles 22 22 0 1 100.00% 100.00%