```
1   module Comparator_10b(A, B, out);
2      input  logic [9:0] A, B;
3      output logic out;
4
5      logic [10:0] car, outty;
6
7      genvar i;
8      generate
9         for(i=0; i<11; i++) begin : eachFA
10           if (i == 0)
11              F_Adder FA(.A(~A[i]), .B(B[i]), .c_in(1'b1),       .out(outty[i]), .c_out(car[i
    ]));
12           else if (i == 10)
13              F_Adder FA(.A(1'b1),  .B(0),    .c_in(car[i-1]),   .out(outty[i]), .c_out(car[i
    ]));
14           else
15              F_Adder FA(.A(~A[i]), .B(B[i]), .c_in(car[i-1]),   .out(outty[i]), .c_out(car[i
    ]));
16        end
17     endgenerate
18
19     assign out = ((car[10]&(~car[9]))|(car[10]&outty[10])|((~car[9])&outty[10]));
20  endmodule
21
22
23  module F_Adder(A, B, c_in, out, c_out);
24     input  logic A, B, c_in;
25     output logic out, c_out;
26
27     assign out   = (A^B)^c_in;
28     assign c_out = (c_in&(A^B))|(A&B);
29  endmodule
30
31  module F_Adder_10b(A_in, B_in, c_in, out, c_out);
32     input  logic [9:0] A_in, B_in;
33     output logic [9:0] out;
34     input  logic c_in;
35     output logic c_out;
36
37     logic [8:0] car;
38
39     genvar i;
40     generate
41        for(i=0; i<10; i++) begin : eachFA
42           if (i == 0)
43              F_Adder FA(.A(A_in[i]), .B(B_in[i]), .c_in,           .out(out[i]), .c_out(car[i
    ]));
44           else if (i == 9)
45              F_Adder FA(.A(A_in[i]), .B(B_in[i]), .c_in(car[i-1]), .out(out[i]), .c_out);
46           else
47              F_Adder FA(.A(A_in[i]), .B(B_in[i]), .c_in(car[i-1]), .out(out[i]), .c_out(car[i
    ]));
48        end
49     endgenerate
50
51  endmodule
52
53  module F_Adder_testbench();
54     logic A, B, c_in, out, c_out;
55
56     F_Adder dut(A, B, c_in, out, c_out);
57
58     initial begin
59        A <= 0; B <= 0; c_in <= 0; #10;
60        A <= 0; B <= 0; c_in <= 1; #10;
61        A <= 0; B <= 1; c_in <= 0; #10;
62        A <= 0; B <= 1; c_in <= 1; #10;
63        A <= 1; B <= 0; c_in <= 0; #10;
64        A <= 1; B <= 0; c_in <= 1; #10;
65        A <= 1; B <= 1; c_in <= 0; #10;
66        A <= 1; B <= 1; c_in <= 1; #10;
67     end
68  endmodule
69
70  module F_Adder_10b_testbench();
71     logic [9:0] A, B, out;
```

```
72       logic c_in, c_out;
73
74       F_Adder_10b dut(A, B, c_in, out, c_out);
75
76       integer i, j;
77
78       initial begin
79          for (i = 0; i<1024; i++) begin
80             A = i;
81             for (j = 0; j < 1024; j++) begin
82                B = j;
83                c_in = 0; #10;
84                c_in = 1; #10;
85             end
86          end
87       end
88    endmodule
89
90    module Comparator_10b_testbench ();
91       logic [9:0] A, B;
92       logic out;
93
94       Comparator_10b dut(A, B, out);
95
96       integer i, j;
97
98       initial begin
99          for ( i = 0; i < 1024; i++) begin
100            A = i;
101            for (j = 0; j < 1024; j++) begin
102               B = j; #10;
103            end
104         end
105      end
106   endmodule
107
```