```
Problem 3: [6 points] Drill problem
Filename: hw5prob3.sv
AndrewID: jtbell
```

```
 1 module hw5prob3
 2   (input logic A, clock, reset_L,
 3    output logic B, C);
 4
 5     enum logic [2:0] {state1=3'b010, state2=3'b100,state3= 3'b011}
 6       currState,nextState;
 7
 8     always_comb begin
 9       case(currState)
10         state1: begin
11           if(A == 0) begin
12             nextState = state2;
13           end
14           if(A == 1) begin
15             nextState = state3;
16           end
17         end
18         state2: begin
19           if(A == 0) begin
20             nextState = state3;
21           end
22           if(A==1) begin
23             nextState = state2;
24           end
25         end
26         state3: begin
27           if (A == 0) begin
28             nextState = state1;
29           end
30           if (A == 1) begin
31             nextState = state3;
32           end
33         end
34       endcase
35     end
36
37     always_comb begin
38       case(currState)
39         state1: begin
40           B = 0;
41           C = 1;
42         end
43         state2: begin
44           B = 1;
45           C = 0;
46         end
47         state3: begin
48           B = 0;
49           C = 1;
50         end
51       endcase
52     end
53
54   always_ff @(posedge clock, negedge reset_L)
55     if (~reset_L)
56       currState <= state1; // or whatever the reset state is
57     else
58       currState <= nextState;
59
60 endmodule
61
62 module hw5prob3_tb();
63   logic A, clock, reset_L;
64   logic B, C;
65
66   hw5prob3 DUT(.*);
67
68   initial begin
69     clock = 0;
```

```
 70      reset_L = 0;
 71      reset_L <= 1;
 72
 73      forever #5 clock = ~clock;
 74    end
 75
 76    initial begin
 77      $monitor($time,, "state=%s, A=%b, B=%b,C=%b,reset_L=%b",DUT.currState.name,
 78        A, B, C, reset_L);
 79    end
 80
 81    initial begin
 82      A <= 1'b0;
 83      reset_L <= 1'b1;
 84      @(posedge clock);
 85      A <= 1'b0;
 86      @(posedge clock);
 87      A <= 1'b0;
 88      @(posedge clock);
 89      A <= 1'b1;
 90      @(posedge clock);
 91      A <= 1'b0;
 92      @(posedge clock);
 93      A <= 1'b1;
 94      @(posedge clock);
 95      A <= 1'b1;
 96      @(posedge clock);
 97      reset_L <= 1'b0;
 98      @(posedge clock);
 99    #100 $finish;
100    end
101
102  endmodule
```