```
1   /* Name: Eugene Ngo
2      Date: 1/13/2023
3      Class: EE 371
4      Lab 6
5      Taken from Lab 1 and adapted for lab 6 task 1 */
6
7   // carSensor takes inputs from two sensors, a and b, and output "1" to either enter or exit
    for 1 clock cycle
8   // whenever an entering or exiting vehicle is detected.
9   module carSensor (a, b, enter, exit, clk, reset);
10     input logic a, b, clk, reset;
11     output logic enter; // car entering
12     output logic exit; //  car exiting
13
14     enum {none, entering01, exiting01, entering11, exiting11, entering10, exiting10, idle} ps
    , ns;
15
16     // Logic for next state
17     always_comb begin
18        case(ps)
19           none: if (~a & ~b) ns = none;
20                 else if (~a & b) ns = entering01;
21                 else if (a & ~b) ns = exiting10;
22                 else ns = idle;
23           entering01: if (~a & ~b) ns = none;
24                 else if (~a & b) ns = entering01;
25                 else if (a & ~b) ns = idle;
26                 else ns = entering11;
27           exiting01: if (~a & ~b) ns = none;
28                 else if (~a & b) ns = exiting01;
29                 else if (a & ~b) ns = idle;
30                 else ns = exiting11;
31           entering11: if (~a & ~b) ns = none;
32                 else if (~a & b) ns = entering01;
33                 else if (a & ~b) ns = entering10;
34                 else ns = entering11;
35           exiting11: if (~a & ~b) ns = none;
36                 else if (~a & b) ns = exiting01;
37                 else if (a & ~b) ns = exiting10;
38                 else ns = exiting11;
39           entering10: if (~a & ~b) ns = none;
40                 else if (~a & b) ns = idle;
41                 else if (a & ~b) ns = entering10;
42                 else ns = entering11;
43           exiting10: if (~a & ~b) ns = none;
44                 else if (~a & b) ns = idle;
45                 else if (a & ~b) ns = exiting10;
46                 else ns = exiting11;
47           idle: if (~a & ~b) ns = none;
48                 else ns = idle;
49        endcase
50     end // always_comb
51
52     //output logic for exiting: outputs 1 to exit when an exiting vehicle is detected.
53     always_comb begin
54        case(ps)
55           exiting01: if (~a & ~b) exit = 1'b1;
56                      else exit = 1'b0;
57           default: exit = 1'b0;
58        endcase
59     end // always_comb
60
61     //DFFs
62     always_ff @(posedge clk) begin
63        if (reset)
64           ps <= none;
65        else
66           ps <= ns;
67     end // always_ff
68  endmodule // carSensor
69
70  // carSensor_testbench tests all expected, unexpected, and edgecase behaviors
71  module carSensor_testbench();
```

```
72      logic a, b, clk, reset, enter, exit;
73      logic CLOCK_50;
74
75      carSensor dut (.a(b), .b(a), .clk(CLOCK_50), .reset, .enter, .exit);
76
77      // Setting up a clock.
78      parameter CLOCK_PERIOD = 100;
79      initial begin
80         CLOCK_50 <= 0;
81         forever #(CLOCK_PERIOD/2) CLOCK_50 <= ~CLOCK_50; // toggle the clock forever
82      end // initial
83
84      initial begin
85         // reset
86         reset <= 1;                      repeat(3) @(posedge CLOCK_50);
87
88         //enters
89         reset <= 0;       a <= 0; b <= 0; repeat(2) @(posedge CLOCK_50);
90                           a <= 1; b <= 0; repeat(2) @(posedge CLOCK_50);
91                           a <= 1; b <= 1; repeat(2) @(posedge CLOCK_50);
92                           a <= 0; b <= 1; repeat(2) @(posedge CLOCK_50);
93                           a <= 0; b <= 0; repeat(2) @(posedge CLOCK_50);
94         //exits
95                           a <= 0; b <= 0; repeat(2) @(posedge CLOCK_50);
96                           a <= 0; b <= 1; repeat(2) @(posedge CLOCK_50);
97                           a <= 1; b <= 1; repeat(2) @(posedge CLOCK_50);
98                           a <= 1; b <= 0; repeat(2) @(posedge CLOCK_50);
99                           a <= 0; b <= 0; repeat(2) @(posedge CLOCK_50);
100
101        // direction changes while entering
102                          a <= 0; b <= 0; repeat(2) @(posedge CLOCK_50);
103                          a <= 1; b <= 0; repeat(2) @(posedge CLOCK_50);
104                          a <= 1; b <= 1; repeat(2) @(posedge CLOCK_50);
105                          a <= 1; b <= 0; repeat(2) @(posedge CLOCK_50);
106                          a <= 1; b <= 1; repeat(2) @(posedge CLOCK_50);
107                          a <= 0; b <= 1; repeat(2) @(posedge CLOCK_50);
108                          a <= 1; b <= 1; repeat(2) @(posedge CLOCK_50);
109                          a <= 0; b <= 1; repeat(2) @(posedge CLOCK_50);
110                          a <= 0; b <= 0; repeat(2) @(posedge CLOCK_50);
111
112        // direction changes while exiting
113                          a <= 0; b <= 0; repeat(2) @(posedge CLOCK_50);
114                          a <= 0; b <= 1; repeat(2) @(posedge CLOCK_50);
115                          a <= 1; b <= 1; repeat(2) @(posedge CLOCK_50);
116                          a <= 0; b <= 1; repeat(2) @(posedge CLOCK_50);
117                          a <= 1; b <= 1; repeat(2) @(posedge CLOCK_50);
118                          a <= 1; b <= 0; repeat(2) @(posedge CLOCK_50);
119                          a <= 1; b <= 1; repeat(2) @(posedge CLOCK_50);
120                          a <= 1; b <= 0; repeat(2) @(posedge CLOCK_50);
121                          a <= 0; b <= 0; repeat(2) @(posedge CLOCK_50);
122        $stop;
123     end // intial
124
125  endmodule // carSensor_testbench
126
```