

```

1  /* Name: Eugene Ngo
2  Date: 3/7/2023
3  Class: EE 371
4  Lab 6: Parking Lot 3D Simulation
5  */
6  // DE1_SoC combines all the modules together, pipelining the control and
7  // various increment modules to Switches and VGPIIO to then
8  // send them to the datapath module to output to the HEX and RAM modules.
9  `timescale 1 ps / 1 ps
10 module DE1_SoC (CLOCK_50, HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, KEY, SW, LEDR, V_GPIIO);
11     // define ports
12     input logic CLOCK_50;
13     output logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
14     input logic [3:0] KEY;
15     input logic [9:0] SW;
16     output logic [9:0] LEDR;
17     inout logic [35:23] V_GPIIO;
18
19     logic clk;
20     assign clk = CLOCK_50;
21
22     logic [1:0] counter_out;
23     logic [3:0] hour_out, incr_out, addr_out;
24     logic [31:0] divided_clocks;
25
26     // FPGA input
27     assign V_GPIIO[26] = V_GPIIO[28]; // LED parking 1
28     assign V_GPIIO[27] = V_GPIIO[29]; // LED parking 2
29     assign V_GPIIO[32] = V_GPIIO[30]; // LED parking 3
30
31     // Parking spot 1 occupied and Parking spot 2 occupied and Parking spot 3 occupied
32     // = parking lot = full.
33     assign V_GPIIO[34] = V_GPIIO[28] & V_GPIIO[29] & V_GPIIO[30]; // LED full
34     // Parking lot isnt full and presence at entrance = open entrance gate
35     assign V_GPIIO[31] = ~V_GPIIO[34] & V_GPIIO[23]; // Open entrance
36     // If anyone at exit sensor, open exit gate.
37     assign V_GPIIO[33] = V_GPIIO[24]; // Open exit
38
39
40     // Helper logic, passed along to different modules.
41     logic zeroOccupied;
42     // logic [6:0] rushHourBegin, rushHourEnd;
43     assign zeroOccupied = !(V_GPIIO[28] | V_GPIIO[29] | V_GPIIO[30]);
44
45     logic startRush;
46     logic rushEnded;
47     logic stopRush;
48     logic endGameHexOut;
49
50     // FPGA output, for debugging
51     assign LEDR[0] = V_GPIIO[28]; // Presence parking 1
52     assign LEDR[1] = V_GPIIO[29]; // Presence parking 2
53     assign LEDR[2] = V_GPIIO[30]; // Presence parking 3
54     assign LEDR[3] = V_GPIIO[23]; // Presence entrance
55     assign LEDR[4] = V_GPIIO[24]; // Presence exit
56     assign LEDR[9] = endGameHexOut; // Presence exit
57     assign LEDR[8] = startRush;
58     assign LEDR[7] = stopRush;
59
60     // clockDivide generates slower clocks to process different inputs
61     clock_divider clockDivide (.clock(clk), .reset(SW[9]), .divided_clocks(divided_clocks));
62
63     // hourCount counter takes in the KEY[0] signal and increases the
64     // current hour, progressing the day in the parking lot
65     hourCount timeCounter (.inc(~KEY[0]), .clk(clk), .reset(SW[9]), .out(hour_out));
66
67     // carCount counter takes in the parking spot signals from VGPIIO 28-30 and outputs
68     // the number of spots left
69     carCount #(3) counter (.park1(V_GPIIO[28]), .park2(V_GPIIO[29]), .park3(V_GPIIO[30]), .full(
70 V_GPIIO[34]), .clk(clk), .reset(SW[9]), .out(counter_out));
71
72     // The carIncrCounter, used for saving the values to RAM
73     carIncrCounter incrCounter (.inc(V_GPIIO[31]), .buttonReset(~KEY[0]), .clk(clk), .reset(SW

```

Page 2 of 3

137    endmodule