

```
1 // Multi-level logic on the FPGA
2 // Michael Humphrey 1/30/2018 SID:1536326
3
4 module fred (HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, KEY, LEDR, SW);
5     output logic [6:0]    HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
6     output logic [9:0]    LEDR;
7     input logic [3:0]     KEY;
8     input logic [9:0]     SW;
9
10    // U,P,C go to SW[9],SW[8],SW[7] respectively. Mark goes to SW[0].
11    // Discount light goes to LEDR[1].
12    logic uc;
13    and a1 (uc, SW[9], SW[7]);
14    or o1 (LEDR[1], uc, SW[8]);
15
16    // Stolen light goes to LEDR[0].
17    logic notc, temp;
18    not n1 (notc, SW[7]);
19    nor no1 (temp, notc, SW[9]);
20    nor no2 (LEDR[0], temp, SW[0], SW[8]);
21
22    // Create product patterns for the display
23    always_comb begin
24
25        case(SW[9:7])
26
27            3'b000: begin
28                HEX5 = ~7'b1111100; //b
29                HEX4 = ~7'b1111011; //e
30                HEX3 = ~7'b1110111; //A
31                HEX2 = ~7'b1010000; //r
32                HEX1 = ~7'b0000000; //
33                HEX0 = ~7'b0000000; //
34            end
35
36            3'b001: begin
37                HEX5 = ~7'b1111100; //b
38                HEX4 = ~7'b0011100; //u
39                HEX3 = ~7'b1010100; //n
40                HEX2 = ~7'b1010100; //n
41                HEX1 = ~7'b1101110; //y
42                HEX0 = ~7'b0000000; //
43            end
44
45            3'b011: begin
46                HEX5 = ~7'b1111100; //b
47                HEX4 = ~7'b0111000; //l
48                HEX3 = ~7'b0011100; //u
49                HEX2 = ~7'b1111100; //b
50                HEX1 = ~7'b0000000; //
51                HEX0 = ~7'b0000000; //
52            end
53
54            3'b100: begin
55                HEX5 = ~7'b1111100; //b
56                HEX4 = ~7'b1011100; //o
57                HEX3 = ~7'b1011100; //o
58                HEX2 = ~7'b1011011; //z
59                HEX1 = ~7'b1111011; //e
60                HEX0 = ~7'b0000000; //
61            end
62
63            3'b101: begin
64                HEX5 = ~7'b1010000; //r
65                HEX4 = ~7'b0011100; //u
66                HEX3 = ~7'b1101111; //g
67                HEX2 = ~7'b0000000; //
68                HEX1 = ~7'b0000000; //
69                HEX0 = ~7'b0000000; //
70            end
71
72            3'b110: begin
73                HEX5 = ~7'b0111001; //c
```

```
74         HEX4 = ~7'b1110111; //A
75         HEX3 = ~7'b1010000; //r
76         HEX2 = ~7'b1010000; //r
77         HEX1 = ~7'b1011100; //o
78         HEX0 = ~7'b1111000; //t
79     end
80
81     default: begin
82         HEX0 = 7'bx;
83         HEX1 = 7'bx;
84         HEX2 = 7'bx;
85         HEX3 = 7'bx;
86         HEX4 = 7'bx;
87         HEX5 = 7'bx;
88     end
89
90 endcase
91 end
92
93
94 endmodule
95
96 module fred_testbench();
97     logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
98     logic [9:0] LEDR;
99     logic [3:0] KEY;
100    logic [9:0] SW;
101
102    fred dut (.HEX0, .HEX1, .HEX2, .HEX3, .HEX4, .HEX5, .KEY, .LEDR, .SW);
103
104    // Try all combinations of inputs.
105    integer i;
106    initial begin
107        SW[0] = 1'b0;
108        for(i = 0; i < 8; i++) begin
109            SW[9:7] = i; #10;
110        end
111        SW[0] = 1'b1;
112        for(i = 0; i < 8; i++) begin
113            SW[9:7] = i; #10;
114        end
115    end
116 endmodule
```