

Lab 2: Intro to Digital Logic, Digital Design Using FPGA:

- K-maps and Boolean Simplification

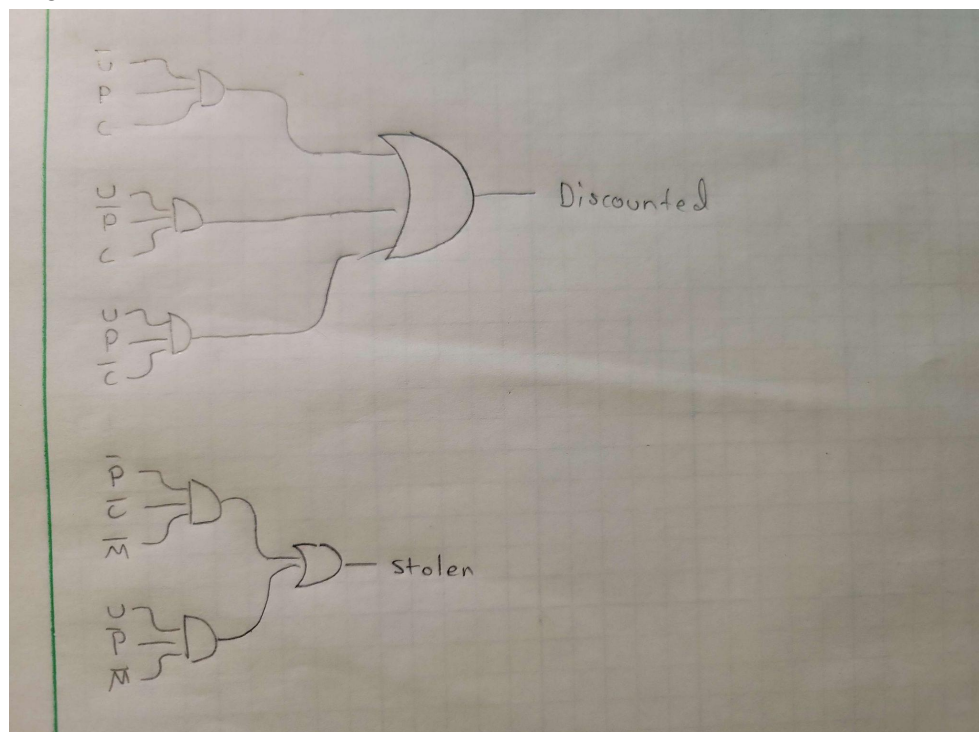
U	P	C	Discounted	Discount Light	$\bar{U}\bar{P}$	$\bar{U}P$	$U\bar{P}$	UP	\bar{C}	C	
0	0	0	0	1	0	0			0	0	
0	0	1	0		X	1			1		$Discounted = U + P\bar{C}$
0	1	1	1				1	X			
1	0	0	0				0	1			$Discounted = \bar{U}P\bar{C} + U\bar{P}C + UP\bar{C}$
1	0	1	1								
1	1	0	1								

U	P	C	Mark	Stolen	$\bar{C}\bar{M}$	$\bar{C}M$	CM	$C\bar{M}$
0	0	0	0	1	✓			
0	0	1	0	0	✓			
0	1	1	0	0	✓			
1	0	0	0	1	✓			
1	0	1	0	1	✓			
1	1	0	0	0	✓			
0	0	0	1	0	✓			
0	0	1	1	X	✓			
0	1	1	1	X	✓			
1	0	0	1	0	✓			
1	0	1	1	0	✓			
1	1	0	1	X	✓			

$\bar{U}\bar{P}$	$\bar{U}P$	$U\bar{P}$	UP
1	0	X	0
X	X	X	0
0	X	X	X
1	0	0	1

$Stolen = \bar{P}\bar{C}\bar{M} + U\bar{P}\bar{M}$

- Circuit Diagram of Implementation:



- Verilog Code:

```

1 // Top-level module that defines the I/Os for the DE-1 SoC board
2
3 module DE1_SoC (HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, KEY, LEDR, SW);
4   output logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
5   output logic [9:0] LEDR;
6   input logic [3:0] KEY;
7   input logic [9:0] SW;
8
9
10  // Logic to check if UPC is discounted
11  // "U" = SW[9], "P" = SW[8], "C" = SW[7],
12  // Discount = LEDR[9]
13  assign LEDR[9] = (~SW[9] & SW[8] & SW[7]) | (SW[9] & ~SW[8] & SW[7]) | (SW[9] & SW[8] & ~SW[7]);
14  // Logic to check if item is stolen
15  // An expensive that is not marked is stolen
16  // "U" = SW[9], "P" = SW[8], "C" = SW[7], "Marked" = SW[0]
17  // Stolen = LED[0]
18  assign LEDR[0] = (~SW[8] & ~SW[7] & ~SW[0]) | (SW[9] & ~SW[8] & ~SW[0]);
19
20  SSS;
21  // Logic to control both of the seven segment displays.
22  seg7 m0 (.bcd(SW[3:0]), .leds(HEX0));
23  seg7 m1 (.bcd(SW[7:4]), .leds(HEX1));
24
25 endmodule
26
27
28 module DE1_SoC_testbench();
29   logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
30   logic [9:0] LEDR;
31   logic [3:0] KEY;
32   logic [9:0] SW;
33
34   DE1_SoC dut (.HEX0, .HEX1, .HEX2, .HEX3, .HEX4, .HEX5, .KEY, .LEDR,
35   .SW);
36
37   // Try all combinations of inputs.
38   integer i;
39   initial begin
40     for(i = 0; i < 256; i++) begin
41       SW[9:7] = i; #10;
42       SW[0] = i;
43     end
44   end
45 endmodule

```

- Don't Cares

