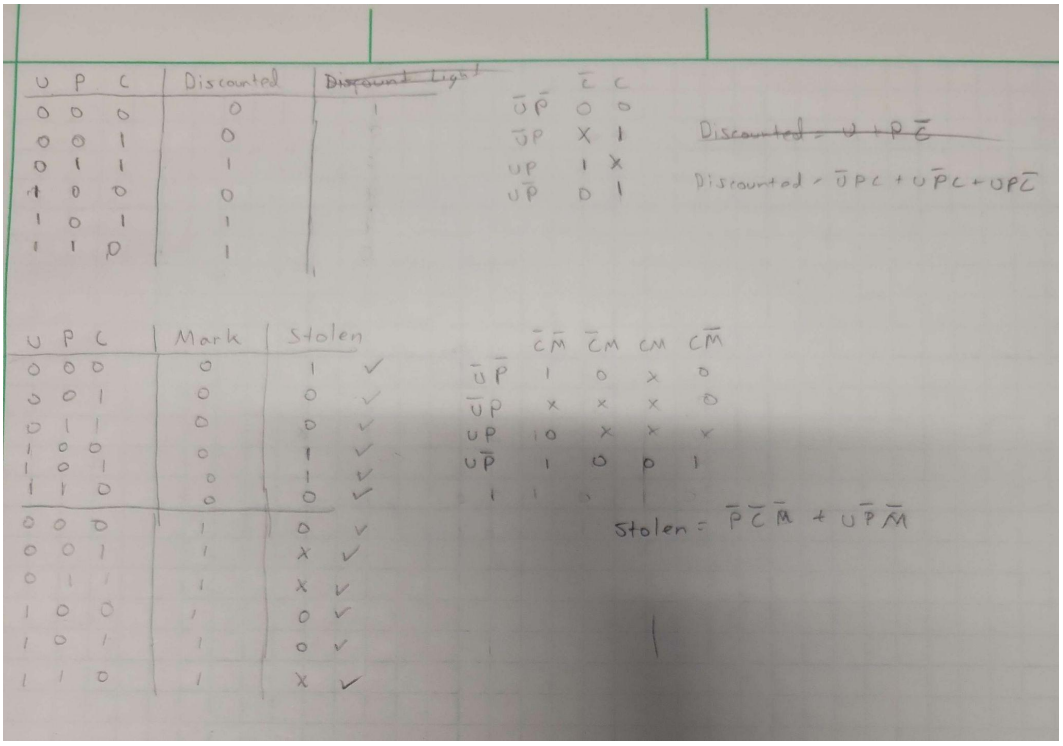
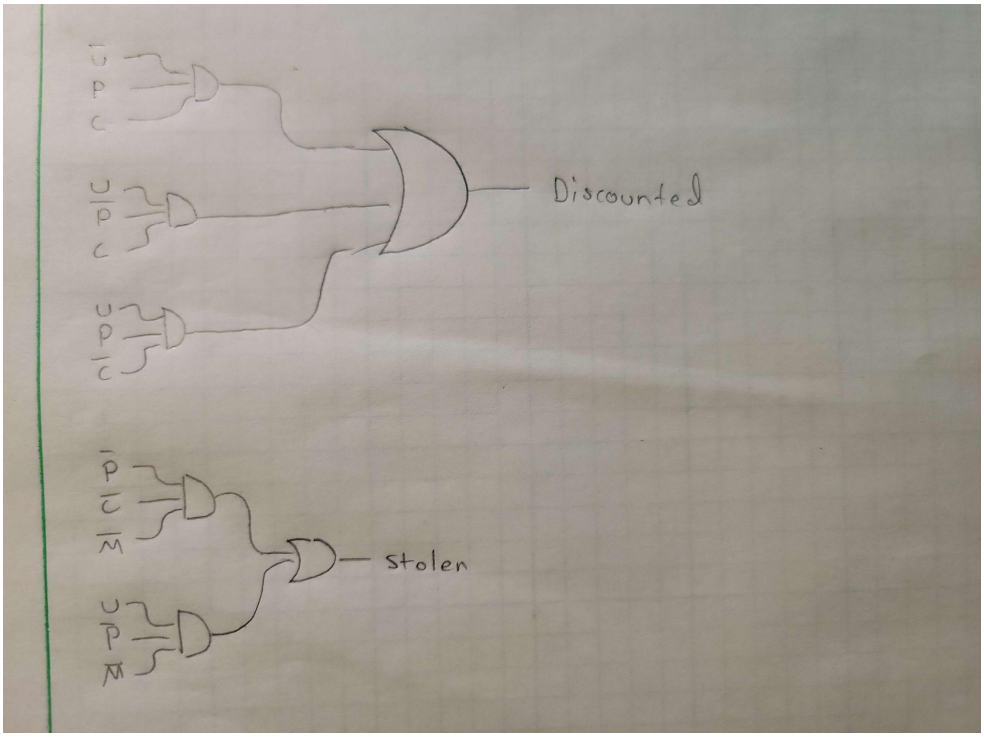
Lab 2: Intro to Digital Logic, Digital Design Using FPGA:

- K-maps and Boolean Simplification
  - 

- Circuit Diagram of Implementation:
  - 

- Verliog Code:
  - ```
    // Top-level module that defines the I/Os for the DE-1 SoC board

    module DE1_SoC (HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, KEY, LEDR, SW);
     output logic  [6:0]    HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
     output logic  [9:0]    LEDR;
     input  logic  [3:0]    KEY;
     input  logic  [9:0]   SW;


     // Logic to check if UPC is discounted
     // "U" = SW[9], "P" = SW[8], "C" = SW[7],
     // Discount = LEDR[9]
     assign LEDR[9] = (~SW[9] & SW[8] & SW[7]) | (SW[9] & ~SW[8] & SW[7]) | (SW[9] & SW[8] & ~SW[7]);
     // Logic to check if item is stolen
     // An expensive that is not marked is stolen
     // "U" = SW[9], "P" = SW[8], "C" = SW[7], "Marked" = SW[0]
     // Stolen = LED[0]
     assign LEDR[0] = (~SW[8] & ~SW[7] & ~SW[0]) | (SW[9] & ~SW[8] & ~SW[0]);

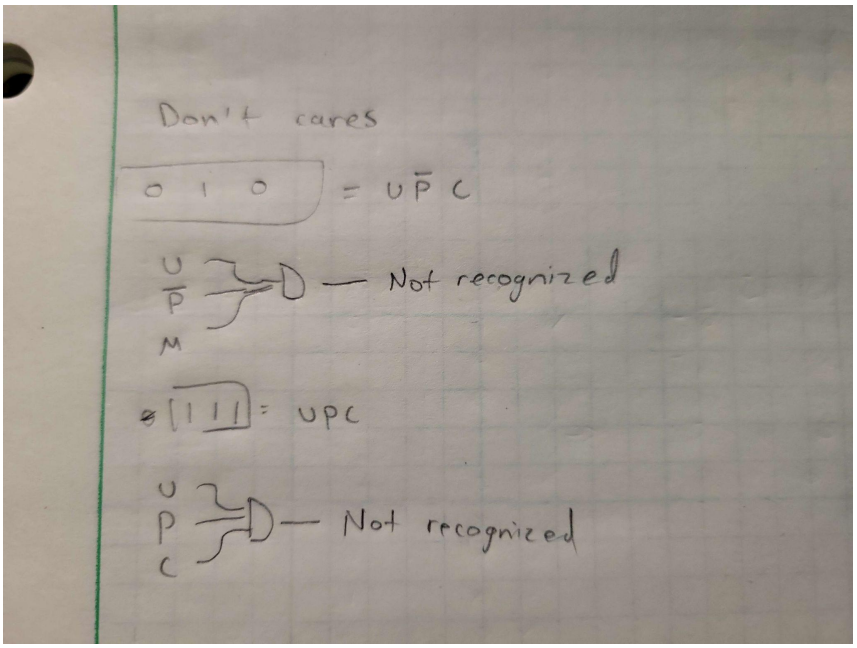
     SSS
     // Logic to control both of the seven segment displays.
     seg7 m0 (.bcd(SW[3:0]), .leds(HEX0));
     seg7 m1 (.bcd(SW[7:4]), .leds(HEX1));

    endmodule


    module DE1_SoC_testbench();
     logic  [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
     logic  [9:0] LEDR;
     logic  [3:0] KEY;
     logic  [9:0] SW;

     DE1_SoC dut (.HEX0, .HEX1, .HEX2, .HEX3, .HEX4, .HEX5, .KEY, .LEDR,
    .SW);

     // Try all combinations of inputs.
     integer i;
     initial begin
      for(i = 0; i <256; i++) begin
        SW[9:7] = i; #10;
        SW[0] = i;
      end
     end
    endmodule
    ```

- Don’t Cares
  - 
    ```