```
// 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;

   // Default values, turns off the HEX displays
   assign HEX0 = 7'b1111111;
   assign HEX1 = 7'b1111111;
   assign HEX2 = 7'b1111111;
   assign HEX3 = 7'b1111111;
   assign HEX4 = 7'b1111111;
   assign HEX5 = 7'b1111111;

   // SW[9], SW[8], SW[7] for U, P, C respectively. SW[0] for MARK.
   // LEDR[0], LEDR[1] for DISCOUNTED, STOLEN signals respectively.

   // DISCOUNTED signal implementation: 2 gates used
   assign LEDR[0] = SW[8] | (SW[9] & SW[7]);    // 1 AND, 1 OR gates used --- 2 gates


   // STOLEN signal implementation: 3 gates used
   assign LEDR[1] = ~((~SW[9] & SW[7]) | SW[0] | SW[8]); // 1 NOT, 1 AND, 1 NOR used --- 3 gates

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
      SW[6:1] = 1'b0;
      for(i = 0; i < 16; i++) begin // i-limit set to 16 because 2^4, 4 switches used
         {SW[0], SW[9:7]} = i; #10;
      end
   end
endmodule
```