```
// Multi-level logic on the FPGA
// Michael Humphrey 1/30/2018 SID:1536326

module fred (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;

   // U,P,C go to SW[9],SW[8],SW[7] respectively. Mark goes to SW[0].
   // Discount light goes to LEDR[1].
   logic uc;
   and a1 (uc, SW[9], SW[7]);
   or o1 (LEDR[1], uc, SW[8]);

   // Stolen light goes to LEDR[0].
   logic notc, temp;
   not n1 (notc, SW[7]);
   nor no1 (temp, notc, SW[9]);
   nor no2 (LEDR[0], temp, SW[0], SW[8]);

   // Create product patterns for the display
   always_comb begin

      case(SW[9:7])

         3'b000: begin
            HEX5 = ~7'b1111100; //b
            HEX4 = ~7'b1111011; //e
            HEX3 = ~7'b1110111; //A
            HEX2 = ~7'b1010000; //r
            HEX1 = ~7'b0000000; //
            HEX0 = ~7'b0000000; //
         end

         3'b001: begin
            HEX5 = ~7'b1111100; //b
            HEX4 = ~7'b0011100; //u
            HEX3 = ~7'b1010100; //n
            HEX2 = ~7'b1010100; //n
            HEX1 = ~7'b1101110; //y
            HEX0 = ~7'b0000000; //
         end

         3'b011: begin
            HEX5 = ~7'b1111100; //b
            HEX4 = ~7'b0111000; //l
            HEX3 = ~7'b0011100; //u
            HEX2 = ~7'b1111100; //b
            HEX1 = ~7'b0000000; //
            HEX0 = ~7'b0000000; //
         end

         3'b100: begin
            HEX5 = ~7'b1111100; //b
            HEX4 = ~7'b1011100; //o
            HEX3 = ~7'b1011100; //o
            HEX2 = ~7'b1011011; //z
            HEX1 = ~7'b1111011; //e
            HEX0 = ~7'b0000000; //
         end

         3'b101: begin
            HEX5 = ~7'b1010000; //r
            HEX4 = ~7'b0011100; //u
            HEX3 = ~7'b1101111; //g
            HEX2 = ~7'b0000000; //
            HEX1 = ~7'b0000000; //
            HEX0 = ~7'b0000000; //
         end

         3'b110: begin
            HEX5 = ~7'b0111001; //C
            HEX4 = ~7'b1110111; //A
            HEX3 = ~7'b1010000; //r
            HEX2 = ~7'b1010000; //r
            HEX1 = ~7'b1011100; //o
            HEX0 = ~7'b1111000; //t
         end

         default: begin
            HEX0 = 7'bX;
            HEX1 = 7'bX;
            HEX2 = 7'bX;
            HEX3 = 7'bX;
            HEX4 = 7'bX;
            HEX5 = 7'bX;
         end

      endcase
   end


endmodule

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

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

   // Try all combinations of inputs.
   integer i;
   initial begin
      SW[0] = 1'b0;
      for(i = 0; i <8; i++) begin
         SW[9:7] = i; #10;
      end
      SW[0] = 1'b1;
      for(i = 0; i <8; i++) begin
         SW[9:7] = i; #10;
      end
   end
endmodule
```