```
/* Name: Eugene Ngo
   Date: 1/13/2023
   Class: EE 371
   Lab 6
   Taken from Lab 1 and adapted for lab 6 task 1 */

// seg7 outputs correct decimal value to hex displays based on the output given by the counter.
// It also displays "FULL" and "CLEAR" according to the indicator outputs given by the counter.
 module seg7 (in, full, clear, hexout0, hexout1, hexout2, hexout3, hexout4, hexout5);

   input logic full, clear;
   input logic [4:0] in;
   output logic [6:0] hexout0, hexout1, hexout2, hexout3, hexout4, hexout5;

   // Assigning hex display variables on necessary numbers.
   logic [6:0] hex0, hex1, hex2, hex3, hex4, hex5, hex6, hex7, hex8, hex9;
   assign hex0 = 7'b1000000; // 0
   assign hex1 = 7'b1111001; // 1
   assign hex2 = 7'b0100100; // 2
   assign hex3 = 7'b0110000; // 3
   assign hex4 = 7'b0011001; // 4
   assign hex5 = 7'b0010010; // 5
   assign hex6 = 7'b0000010; // 6
   assign hex7 = 7'b1111000; // 7
   assign hex8 = 7'b0000000; // 8
   assign hex9 = 7'b0010000; // 9

   // Assigning hex display variables on necessary letters.
   logic [6:0] hexf, hexu, hexl, hexc, hexe, hexa, hexr, hexoff;
   assign hexf = 7'b0001110; // F
   assign hexu = 7'b1000001; // U
   assign hexl = 7'b1000111; // L
   assign hexc = 7'b1000110; // C
   assign hexe = 7'b0000110; // E
   assign hexa = 7'b0001000; // A
   assign hexr = 7'b0101111; // R
   assign hexoff = 7'b1111111; // off

   // Logic for hexout0: 26 different cases for 26 numbers. (0-25)
   always_comb begin
      case(in)
         5'b00000: hexout0 = hex0;
         5'b00001: hexout0 = hex1;
         5'b00010: hexout0 = hex2;
         5'b00011: hexout0 = hex3;
         5'b00100: hexout0 = hex4;
         5'b00101: hexout0 = hex5;
         5'b00110: hexout0 = hex6;
         5'b00111: hexout0 = hex7;
         5'b01000: hexout0 = hex8;
         5'b01001: hexout0 = hex9;
         5'b01010: hexout0 = hex0;
         5'b01011: hexout0 = hex1;
         5'b01100: hexout0 = hex2;
         5'b01101: hexout0 = hex3;
         5'b01110: hexout0 = hex4;
         5'b01111: hexout0 = hex5;
         5'b10000: hexout0 = hex6;
         5'b10001: hexout0 = hex7;
         5'b10010: hexout0 = hex8;
         5'b10011: hexout0 = hex9;
         5'b10100: hexout0 = hex0;
         5'b10101: hexout0 = hex1;
         5'b10110: hexout0 = hex2;
         5'b10111: hexout0 = hex3;
         5'b11000: hexout0 = hex4;
         5'b11001: hexout0 = hex5;
         default: hexout0 = 7'bX;
      endcase
   end // always_comb

   // Logic for hexout1: 26 different cases for 26 numbers. (0-25)
   always_comb begin
      case(in)
         5'b00000: hexout1 = hexr;
         5'b00001: hexout1 = hexoff;
         5'b00010: hexout1 = hexoff;
         5'b00011: hexout1 = hexoff;
         5'b00100: hexout1 = hexoff;
         5'b00101: hexout1 = hexoff;
         5'b00110: hexout1 = hexoff;
         5'b00111: hexout1 = hexoff;
         5'b01000: hexout1 = hexoff;
         5'b01001: hexout1 = hexoff;
         5'b01010: hexout1 = hex1;
         5'b01011: hexout1 = hex1;
         5'b01100: hexout1 = hex1;
         5'b01101: hexout1 = hex1;
         5'b01110: hexout1 = hex1;
         5'b01111: hexout1 = hex1;
         5'b10000: hexout1 = hex1;
         5'b10001: hexout1 = hex1;
         5'b10010: hexout1 = hex1;
         5'b10011: hexout1 = hex1;
         5'b10100: hexout1 = hex2;
         5'b10101: hexout1 = hex2;
         5'b10110: hexout1 = hex2;
         5'b10111: hexout1 = hex2;
         5'b11000: hexout1 = hex2;
         5'b11001: hexout1 = hex2;
         default: hexout1 = 7'bX;
      endcase
   end // always_comb

   // Logic for hexout5 - hexout2: display letters when full or clear, turn off otherwise.
   always_comb begin
      if (full) begin
         hexout5 = hexf;
         hexout4 = hexu;
         hexout3 = hexl;
         hexout2 = hexl;
      end
      else if (clear) begin
         hexout5 = hexc;
         hexout4 = hexl;
         hexout3 = hexe;
         hexout2 = hexa;
      end
      else begin
         hexout5 = hexoff;
         hexout4 = hexoff;
         hexout3 = hexoff;
         hexout2 = hexoff;
      end
   end // always_comb

 endmodule // seg7

 // seg7_testbench tests all expected, unexpected, and edgecase behaviors
 module seg7_testbench();
   logic full, clear;
   logic [4:0] in;
   logic [6:0] hexout0, hexout1, hexout2, hexout3, hexout4, hexout5;

   seg7 dut (.in, .full, .clear, .hexout0, .hexout1, .hexout2, .hexout3, .hexout4, .hexout5);

   initial begin
      in = '0; clear = 1; full = 0; #10; // testing clear output
      in = 5'b11001; clear = 0; full = 1; #10; // testing full output
      in = 5'b10011; clear = 0; full = 0; #10; // testing regular output
      $stop;
   end // initial
endmodule // seg7_testbench
```