```
module MorseEncoder(SW, KEY, LEDR, CLOCK_50);
   input [2:0] SW;
   input CLOCK_50;
   input [1:0] KEY;
   output [0:0] LEDR;
   reg [2:0] select;
   reg reset, shift;
   wire [12:0] morseCode,conn;
   wire[25:0] counter;
   wire en;

   always@(*)
   begin
      reset= KEY[0];
      shift = KEY[1];
      select = SW;
   end

   mux8to1 m0(select, morseCode);
   rateDivider r(CLOCK_50, en, counter);

   subcircuit s12(morseCode[12], 1'b0, shift, en, reset, CLOCK_50, conn[12]);
   subcircuit s11(morseCode[11], conn[12], shift, en, reset, CLOCK_50, conn[11]);
   subcircuit s10(morseCode[10], conn[11], shift, en, reset, CLOCK_50, conn[10]);
   subcircuit s9(morseCode[9], conn[10], shift, en, reset, CLOCK_50, conn[9]);
   subcircuit s8(morseCode[8], conn[9], shift, en, reset, CLOCK_50, conn[8]);
   subcircuit s7(morseCode[7], conn[8], shift, en, reset, CLOCK_50, conn[7]);
   subcircuit s6(morseCode[6], conn[7], shift, en, reset, CLOCK_50, conn[6]);
   subcircuit s5(morseCode[5], conn[6], shift, en, reset, CLOCK_50, conn[5]);
   subcircuit s4(morseCode[4], conn[5], shift, en, reset, CLOCK_50, conn[4]);
   subcircuit s3(morseCode[3], conn[4], shift, en, reset, CLOCK_50, conn[3]);
   subcircuit s2(morseCode[2], conn[3], shift, en, reset, CLOCK_50, conn[2]);
   subcircuit s1(morseCode[1], conn[2], shift, en, reset, CLOCK_50, conn[1]);
   subcircuit s0(morseCode[0], conn[1], shift, en, reset, CLOCK_50, conn[0]);

   assign LEDR[0] = conn[0];

endmodule

module mux8to1(input [2:0] select, output reg [12:0] morseCode);
   always @(*)
      case(select[2:0])
         3'b000: morseCode= 13'b101; //i
         3'b001: morseCode= 13'b1110111011101; //j
         3'b010: morseCode= 13'b111010111; //k
         3'b011: morseCode= 13'b101011101; //l
         3'b100: morseCode= 13'b1110111; //m
         3'b101: morseCode= 13'b10111; //n
         3'b110: morseCode= 13'b11101110111; //o
         3'b111: morseCode= 13'b10111011101; //p
         default: morseCode=13'b0;
      endcase
endmodule

module rateDivider(input clock, output reg en, output reg [25:0] counter);
   always@(posedge clock)
      if (counter == 26'd24999999)
      begin
         en = 1'b1;
         counter <= 26'b0;
      end
      else
      begin
         en = 1'b0;
         counter <= counter + 1 ;
      end
endmodule


module mux2to1(input x,y, select, output out);
   assign out = select ? y : x;

endmodule

module subcircuit(morse, mux0, shift, en, reset, clock, conn);
   input morse, mux0, shift, en, reset, clock;
```

```
77      output reg conn;
78      wire D;
79      mux2to1 m(morse, mux0, shift, D);
80      always@(posedge clock)
81      begin
82         if (reset == 1'b1)
83            conn<=0;
84         else if (en == 1'b1)
85            conn<=D;
86      end
87   endmodule
```