

```

1  // Data written to registers R0 to R5 are sent to the H digits
2  module seg7_scroll (Data, Addr, Sel, Resetn, Clock, H5, H4, H3, H2, H1, H0);
3      input [6:0] Data;
4      input [2:0] Addr;
5      input Sel, Resetn, Clock;
6      output [6:0] H5, H4, H3, H2, H1, H0;
7
8      reg [5:0] enable; // Enable signal to decide which 7 segment to write to
9
10     // One hot encode the 7 segment to be written to
11     always @(*)
12     begin
13         case (Addr)
14             0: enable = 6'b000001;
15             1: enable = 6'b000010;
16             2: enable = 6'b000100;
17             3: enable = 6'b001000;
18             4: enable = 6'b010000;
19             5: enable = 6'b100000;
20             default: enable = 6'b000000;
21         endcase
22     end
23
24     // Registers for the data written to each 7 segment display
25     regne HEX0 (.R(~Data), .Clock(Clock), .Resetn(Resetn), .E(enable[0] & Sel), .Q(H0));
26     regne HEX1 (.R(~Data), .Clock(Clock), .Resetn(Resetn), .E(enable[1] & Sel), .Q(H1));
27     regne HEX2 (.R(~Data), .Clock(Clock), .Resetn(Resetn), .E(enable[2] & Sel), .Q(H2));
28     regne HEX3 (.R(~Data), .Clock(Clock), .Resetn(Resetn), .E(enable[3] & Sel), .Q(H3));
29     regne HEX4 (.R(~Data), .Clock(Clock), .Resetn(Resetn), .E(enable[4] & Sel), .Q(H4));
30     regne HEX5 (.R(~Data), .Clock(Clock), .Resetn(Resetn), .E(enable[5] & Sel), .Q(H5));
31
32 endmodule
33
34 module regne (R, Clock, Resetn, E, Q);
35     parameter n = 7;
36     input [n-1:0] R;
37     input Clock, Resetn, E;
38     output [n-1:0] Q;
39     reg [n-1:0] Q;
40
41     always @(posedge Clock)
42         if (Resetn == 0)
43             Q <= {n{1'b1}};
44         else if (E)
45             Q <= R;
46 endmodule
47

```