

module seven\_seg\_decoder(input [3:0] x, output [6:0] hex\_LEDs);

reg [6:0] reg\_LEDs;

assign hex\_LEDs[0]=(~input[a]&input[b]&~input[c]&~input[d])|

(~input[a]&~input[b]&~input[c]&input[d])|(input[a]&input[b]&input[d])|

(input[a]&input[b]&input[c]) |(input[a]&input[c]&input[d]);

assign hex\_LEDs[1]=(input[b]&~input[c]&input[d]) |(input[a]&input[c]&input[d]) |(~input[a]&input[b]&input[c]&~input[d]);

assign hex\_LEDs[6:2]=reg\_LEDs[6:2];

always @(\*)

begin

case (x)

4'b0000: reg\_LEDs[6:2]=7'b10000; //7'b1000000 decimal 0

4'b0001: reg\_LEDs[6:2]=7'b11110; //7'b1111001 decimal 1

4'b0010: reg\_LEDs[6:2]=7'b01001; //7'b0100100 decimal 2

4'b0011: reg\_LEDs[6:2]=7'b01001; //7'b0110000 decimal 3

4'b0100: reg\_LEDs[6:2]=7'b01001; //7'b0011001 decimal 4

4'b0101: reg\_LEDs[6:2]=7'b01001; //7'b0010010 decimal 5

4'b0110: reg\_LEDs[6:2]=7'b01001; //7'b0000010 decimal 6

4'b0111: reg\_LEDs[6:2]=7'b01001; //7'b1111000 decimal 7

4'b1000: reg\_LEDs[6:2]=7'b01001; //7'b0000000 decimal 8

4'b1001: reg\_LEDs[6:2]=7'b01001; //7'b0010000 decimal 9

4'b1010: reg\_LEDs[6:2]=7'b01001; //7'b0011000 decimal q

4'b1011: reg\_LEDs[6:2]=7'b01001; //7'b1100011 decimal u

4'b1100: reg\_LEDs[6:2]=7'b01001; //7'b0001000 decimal a

4'b1101: reg\_LEDs[6:2]=7'b01001; //7'b0101011 decimal n

4'b1110: reg\_LEDs[6:2]=7'b01001; //7'b0010001 decimal y

4'b1111: reg\_LEDs[6:2]=7'b01001; //7'b1111111 decimal OFF

endcase

end

endmodule