



```
module part4 (SW, HEX0);  
    input [9:0] SW;  
    output [0:6] HEX0;  
  
    assign HEX0[0] = SW[1] | (~SW[0]);  
    assign HEX0[1] = SW[0];  
    assign HEX0[2] = SW[0];  
    assign HEX0[3] = SW[1];  
    assign HEX0[4] = SW[1];  
    assign HEX0[5] = SW[1] | (~SW[0]);  
    assign HEX0[6] = SW[1];  
  
endmodule
```

```

module part5 (SW, LEDR, HEX0, HEX1, HEX2);
    input [9:0] SW;
    output [9:0] LEDR;
    output [0:6] HEX0;
    output [0:6] HEX1;
    output [0:6] HEX2;

    assign LEDR[9:8] = SW[9:8];

    wire [1:0] M0, M1, M2;

    mux_2bit_3to1 U0 (SW[9:8], SW[5:4], SW[3:2], SW[1:0], M0);
    char_7seg H0 (M0, HEX0);

    mux_2bit_3to1 U1 (SW[9:8], SW[1:0], SW[5:4], SW[3:2], M1);
    char_7seg H1 (M1, HEX1);

    mux_2bit_3to1 U2 (SW[9:8], SW[3:2], SW[1:0], SW[5:4], M2);
    char_7seg H2 (M2, HEX2);

endmodule

//2-bit wide 3 to 1 mux
module mux_2bit_3to1 (S, U, V, W, M);
    input [1:0] S, U, V, W;
    output [1:0] M;

    wire [1:0] f;

    assign f[0] = (~S[0] & U[0]) | (S[0] & V[0]);
    assign f[1] = (~S[0] & U[1]) | (S[0] & V[1]);

    assign M[0] = (~S[1] & f[0]) | (S[1] & W[0]);
    assign M[1] = (~S[1] & f[1]) | (S[1] & W[1]);

endmodule

//7-seg decoder for d, E, 1 and 'blank'
module char_7seg(C, Display);
    input [1:0] C;
    output [0:6] Display;

    assign Display[0] = C[1] | (~C[0]);
    assign Display[1] = C[0];
    assign Display[2] = C[0];
    assign Display[3] = C[1];
    assign Display[4] = C[1];
    assign Display[5] = C[1] | (~C[0]);
    assign Display[6] = C[1];

endmodule

```