```verilog
`timescale 1ns / 1ns // `timescale time_unit/time_precision

// 8 bit counter using T flipflops
module TFFcounter8bit(input[9:0] SW, input[3:0] KEY, output[6:0] HEX0,HEX1);
    wire clock,enable,reset;

    wire[7:0] Q;
    wire[6:0] QT;

    // Input assignment
    assign clock = KEY[0];
    assign enable = SW[1];
    assign reset = SW[0];

    // Putting 6 counter slices together with a single TFFs at the ends
    T_flipflop bit0(.clock(clock),.reset(reset),.T(enable),.Q(Q[0]));
    counterSlice bit1(.clock(clock),.reset(reset),.enable(Q[0]),.Q(Q[1]),.QandT(QT[0]));
    counterSlice bit2(.clock(clock),.reset(reset),.enable(QT[0]),.Q(Q[2]),.QandT(QT[1]));
    counterSlice bit3(.clock(clock),.reset(reset),.enable(QT[1]),.Q(Q[3]),.QandT(QT[2]));
    counterSlice bit4(.clock(clock),.reset(reset),.enable(QT[2]),.Q(Q[4]),.QandT(QT[3]));
    counterSlice bit5(.clock(clock),.reset(reset),.enable(QT[3]),.Q(Q[5]),.QandT(QT[4]));
    counterSlice bit6(.clock(clock),.reset(reset),.enable(QT[4]),.Q(Q[6]),.QandT(QT[5]));
    T_flipflop bit7(.clock(clock),.reset(reset),.T(QT[5]),.Q(Q[7]));

    // Output
    displayHEX h0(.BIN(Q[3:0]),.HEX(HEX0));
    displayHEX h1(.BIN(Q[7:4]),.HEX(HEX1));
endmodule

// Counter slice, not applicable to last slice (no AND gate)
// Just a TFF + an AND gate
module counterSlice(input enable,clock,reset, output Q,QandT);
    T_flipflop tff(.T(enable),.clock(clock),.reset(reset),.Q(Q));
    assign QandT = Q & enable;
endmodule

// Active-low, synchronous reset positive edge triggered TFF
module T_flipflop(input T,clock,reset, output Q);
    reg q;

    // Triggered on clock rises -> button presses
    always @(posedge clock)
    begin
        if(!reset) // Set to 0
            q <= 0;
        else if(!T) // Pass through Q
            q <= Q;
        else if(T) // Pass through Q complement
            q <= ~Q;
    end

    assign Q = q;
endmodule

// Turns decimal to HEX
module displayHEX(input [3:0] BIN, output [6:0] HEX);
     wire x = BIN[3], y = BIN[2], z = BIN[1], w = BIN[0];
     //assign each segment with appropriate formula
     assign HEX[0] = ~((x|y|z|~w)&(x|~y|z|w)&(~x|y|~z|~w)&(~x|~y|z|~w));
     assign HEX[1] =
     ~((x|~y|z|~w)&(x|~y|~z|w)&(~x|y|~z|~w)&(~x|~y|z|w)&(~x|~y|~z|w)&(~x|~y|~z|~w));
     assign HEX[2] = ~((x|y|~z|w)&(~x|~y|z|w)&(~x|~y|~z|w)&(~x|~y|~z|~w));
     assign HEX[3] =
     ~((x|y|z|~w)&(x|~y|z|w)&(x|~y|~z|~w)&(~x|y|z|~w)&(~x|y|~z|w)&(~x|~y|~z|~w));
     assign HEX[4] =
     ~((x|y|z|~w)&(x|y|~z|~w)&(x|~y|z|w)&(x|~y|z|~w)&(x|~y|~z|~w)&(~x|y|z|~w));
     assign HEX[5] = ~((x|y|z|~w)&(x|y|~z|w)&(x|y|~z|~w)&(x|~y|~z|~w)&(~x|~y|z|~w));
     assign HEX[6] = ~((x|y|z|w)&(x|y|z|~w)&(x|~y|~z|~w)&(~x|~y|z|w));
endmodule
```