```verilog
`timescale 1ns / 1ns

module memory(input[9:0] SW, input[3:0] KEY, output[6:0] HEX0,HEX2,HEX4,HEX5);
    wire[3:0] DATA_IN,DATA_OUT;
    wire[4:0] address;
    wire writeEnable,clock;

    // Input assignments
    assign DATA_IN = SW[3:0];
    assign address = SW[8:4];
    assign writeEnable = SW[9];
    assign clock = KEY[0];

    // Memory module
    ram32x4 ram0(
        .clock(clock),.wren(writeEnable),
        .address(address),
        .data(DATA_IN),.q(DATA_OUT)
    );

    // Output
    displayHEX h0(.BIN(DATA_OUT),.HEX(HEX0));
    displayHEX h2(.BIN(DATA_IN),.HEX(HEX2));
    displayHEX h4(.BIN(address[3:0]),.HEX(HEX4));
    displayHEX h5(.BIN({3'b0,address[4]}),.HEX(HEX5));
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
```