

```
// File: display.v
// John Hubbard, 16 Feb 2015
```

```
module simple_counter(clk, arst,q) ;
    parameter N = 7 ;
    input clk,arst;
    output reg [N-1:0] q ;

    always @(posedge clk or posedge arst)
    if (arst == 1'b1)
        q <= 0 ;
    else
        q <= q + 1 ;
endmodule
```

```
module decoder(text,s,y,val) ;
    input [15:0] text;
    input [1:0] s ;
    output reg [3:0] y ;
    output reg [3:0] val ;

    always @(*)
    begin
        case (s)
            0: begin
                y <= 4'b1110 ;
                val <= text[3:0] ;
            end
            1:begin
                y <= 4'b1101 ;
                val <= text[7:4] ;
            end
            2:begin
                y <= 4'b1011 ;
                val <= text[11:8] ;
            end
            3: begin
                y <= 4'b0111 ;
                val <= text[15:12] ;
            end
            default:begin
                y <= 4'bx ;
                val <= 4'bx ;
            end
        endcase
    end
endmodule
```

```
module display(text,clk, arst, seg, an) ;
    parameter C = 21;
    parameter N7 = 7 ;
    parameter N4 = 4 ;
    parameter N2 = 2 ;
    parameter ANODE_FREQ = 20 ; //20 = 47 HZ; 19 is 95 HZ 95-47 = 48HZ
```

```

input [15:0] text;
input clk, arst;
output [0:N7-1] seg ;
output [N4-1:0] an ;

wire [C-1:0] q ;
wire [N2-1:0] sel ;
wire [N4-1:0] zero_to_f_counter ;

assign sel[1] = q[ANODE_FREQ] ;
assign sel[0] = q[ANODE_FREQ-1] ;

//assign sel[1] = q[15] ; 23 hz
//assign sel[0] = q[20] ; 762 hz
//Run with above number and see display
simple_counter #C U(clk,arst,q) ;
decoder D(text,sel,an,zero_to_f_counter) ;
hex2_7seg H(zero_to_f_counter,seg);
endmodule

module display_test(clk, btnU, seg, an) ;
parameter D = 16'h8602;
parameter N7 = 7;
parameter N4 = 4;
input clk, btnU;
output [0:N7-1] seg ;
output [N4-1:0] an ;

wire [15:0] text;
assign text = D ;
display T(text,clk,btnU, seg, an) ;
endmodule

```