`timescale 100ms/10ms

module BCD\_counter(count,TC,clr\_, enb, clk) ;

output [3:0] count ;

output TC ;

reg [3:0] count ;

input clr\_, enb, clk ;

reg TC ;

always @ (\*)

    if(enb && count >= 9)

        TC = 1;

    else

        TC = 0;

always @ (posedge clk or negedge clr\_) // combinational+sequential

    if (~clr\_)

        count = 0 ;

    else if (enb)

        if (TC)

            count = 0;

        else

            count = count + 1;

endmodule

module Hex\_counter(count,TC,clr\_, enb, clk) ;

output [2:0] count ;

output TC ;

reg [2:0] count ;

input clr\_, enb, clk ;

reg TC ;

always @ (\*)

    if(enb && count >= 5)

        TC = 1;

    else

        TC = 0;

always @ (posedge clk or negedge clr\_) // combinational+sequential

    if (~clr\_)

        count = 0 ;

    else if (enb)

        if (TC)

            count = 0;

        else

            count = count + 1 ;

endmodule

module BCD\_counter\_2(count,TC,clr\_, enb , h1\_is\_2, clk) ;

output [3:0] count ;

output TC ;

reg [3:0] count ;

input clr\_, enb , h1\_is\_2, clk ;

reg TC ;

always @ (\*)

    if((enb && count >= 9) || (enb && h1\_is\_2 && count>=3))

        TC = 1;

    else

        TC = 0;

always @ (posedge clk or negedge clr\_) // combinational+sequential

    if (~clr\_)

        count = 0 ;

    else if (enb)

        if (TC)

            count = 0;

        else

            count = count + 1;

endmodule

module Hex\_counter\_2(count,TC,clr\_, enb , h1\_is\_2, clk) ;

output [2:0] count ;

output TC , h1\_is\_2;

reg [2:0] count ;

input clr\_, enb, clk ;

reg TC ;

reg h1\_is\_2 ;

always @ (\*)

    if(enb && count >= 2)

        TC = 1;

    else

        TC = 0;

always @(\*)

    if(count>=2)

        h1\_is\_2 = 1;

    else

        h1\_is\_2 = 0;

always @ (posedge clk or negedge clr\_) // combinational+sequential

    if (~clr\_)

        count = 0 ;

    else if (enb)

        if (TC)

            count = 0;

        else

            count = count + 1 ;

endmodule

module Testfixture ;

wire Vdd = 1'b1;

reg clk, clr\_;

wire [3:0] m0 ;

wire [2:0] m1 ;

wire [3:0] h0 ;

wire [2:0] h1 ;

BCD\_counter M0 (m0,m0\_tc,clr\_, Vdd, clk) ;   //BCD\_counter(count,TC,clr\_, enb, clk)

Hex\_counter M1 (m1,m1\_tc,clr\_, m0\_tc, clk) ; //Hex\_counter(count,TC,clr\_, enb, clk)

BCD\_counter\_2 H0 (h0,h0\_tc,clr\_, m1\_tc , h1\_is\_2, clk) ;  //BCD\_counter\_2(count,TC,clr\_, enb , h1\_is\_2, clk)

Hex\_counter\_2 H1 (h1,h1\_tc,clr\_, h0\_tc, h1\_is\_2 , clk) ; //Hex\_counter\_2(count,TC,clr\_, enb , h1\_is\_2, clk)

//Stimulus

initial

begin

    clk = 1'b0 ;

    repeat(4000)

        #5 clk = ~clk ;

    $finish ;

end

initial

begin

  clr\_ = 0;

  #6  clr\_ = 1'b1;

end

initial

begin

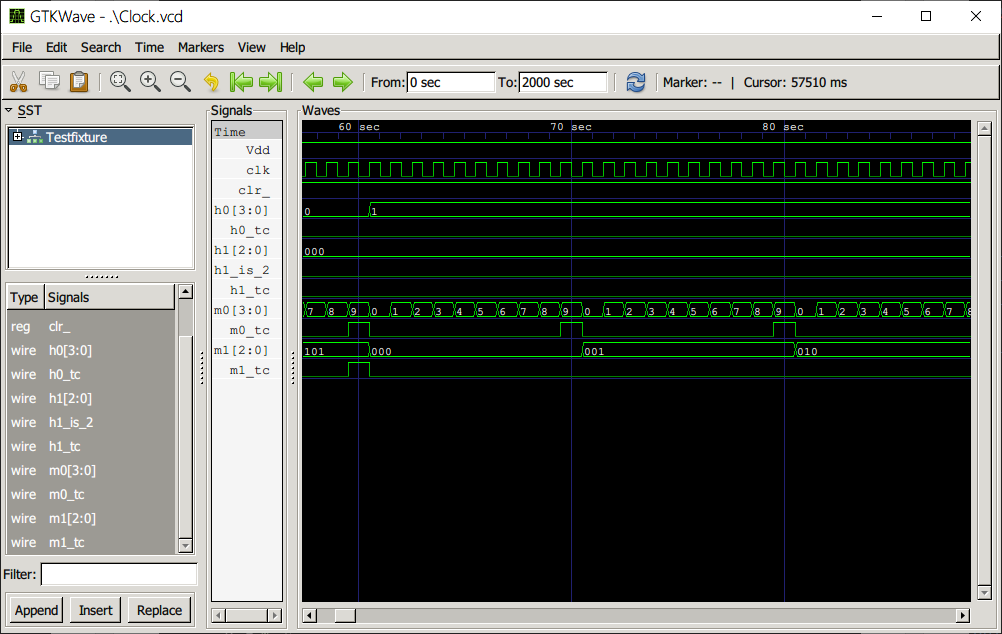
$dumpfile("Clock.vcd");

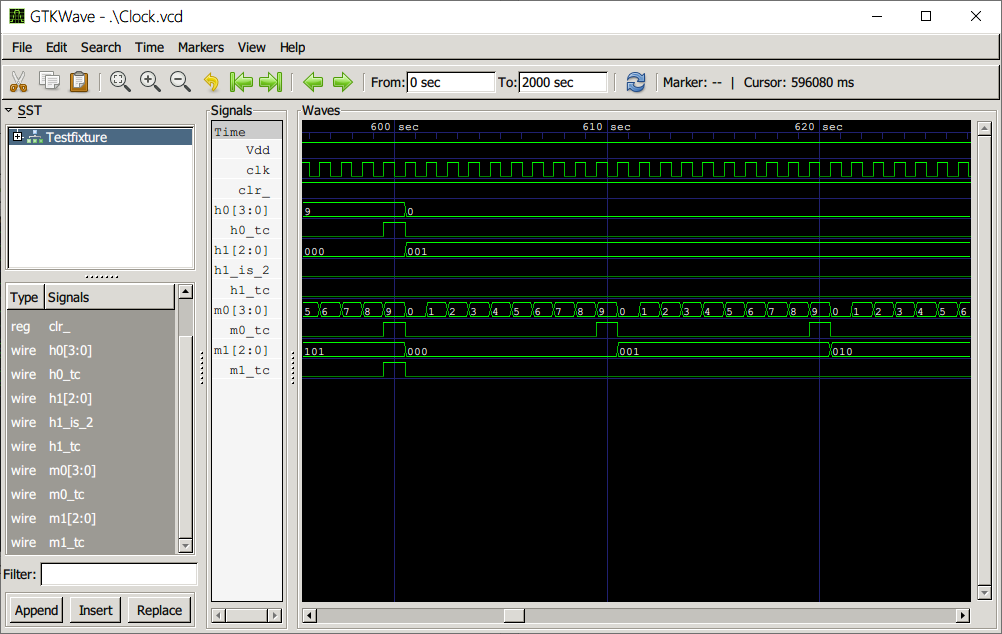
$dumpvars;

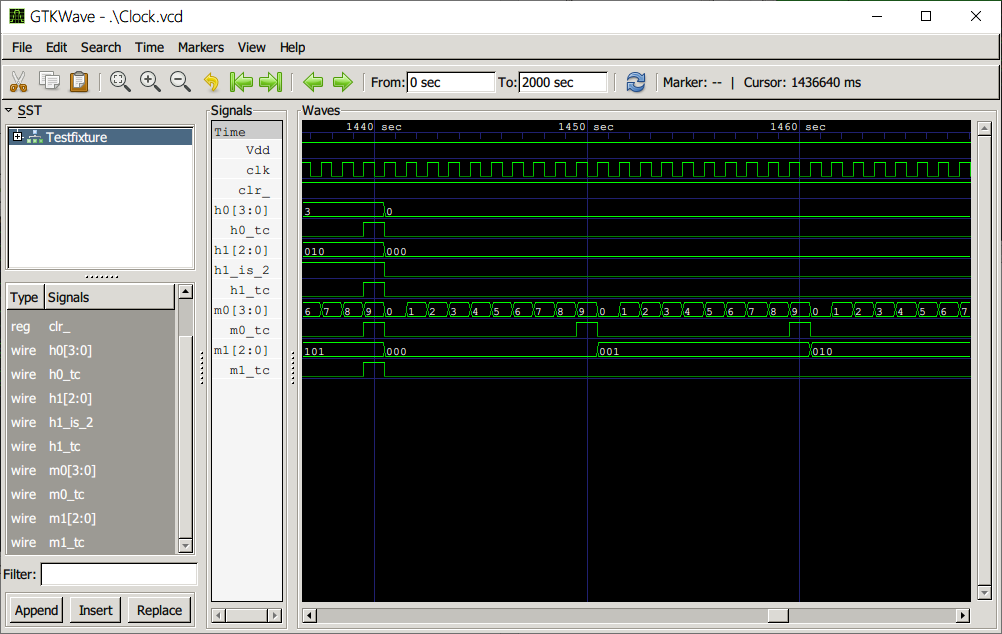
end

endmodule

simulation:(因為總共有2000分鐘的圖形，我只擷取我認為比較有代表性的部分)

00:59到01:00

09:59 到 10:00

23:59 到 00:00