```
`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;
```

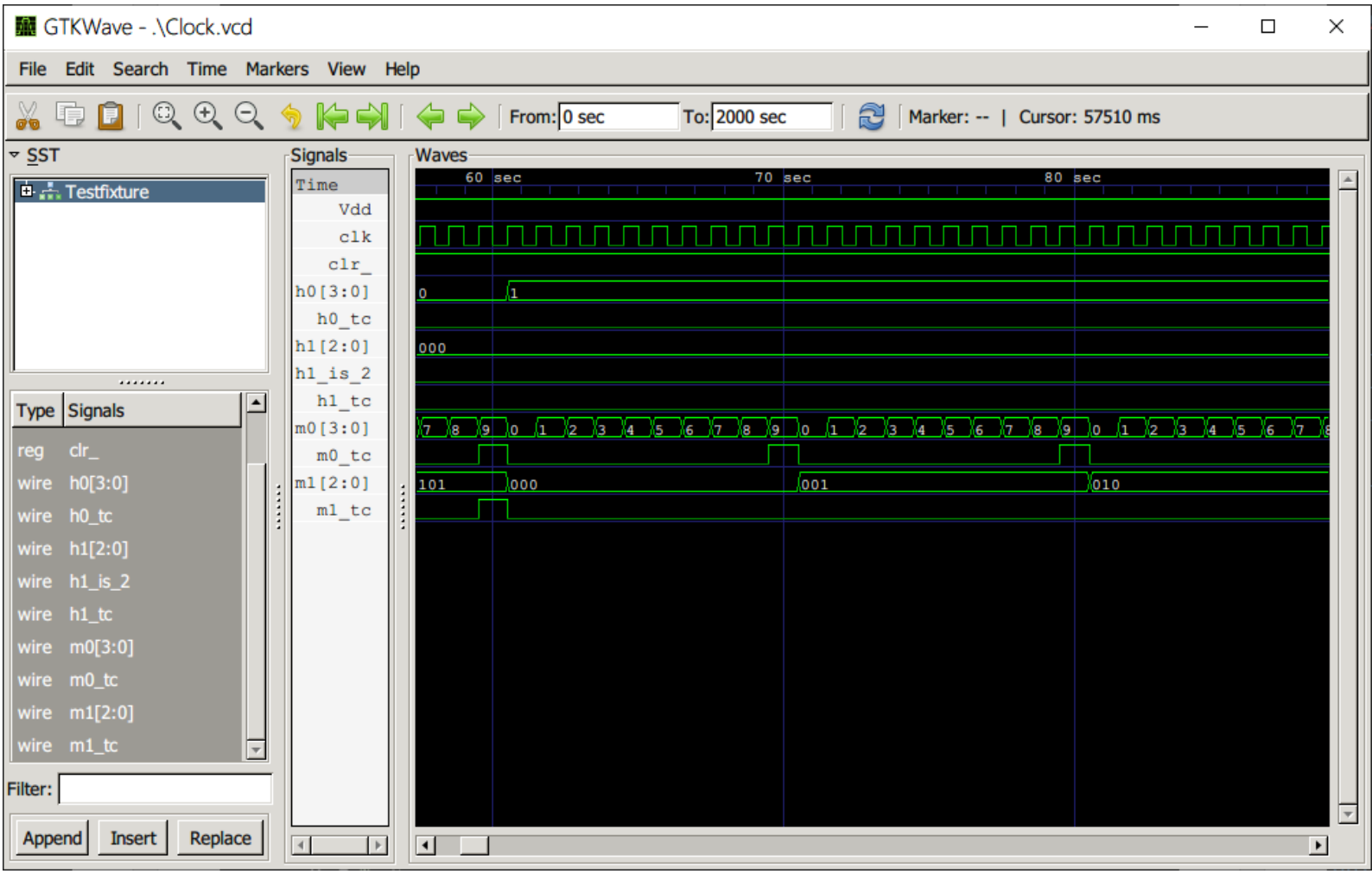
```
        else
            count = count + 1 ;
endmodule

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_, e
nb , h1_is_2, clk)
Hex_counter_2 H1 (h1,h1_tc,clr_, h0_tc, h1_is_2 , clk) ; //Hex_counter_2(count,TC,clr_, en
b , 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;
```

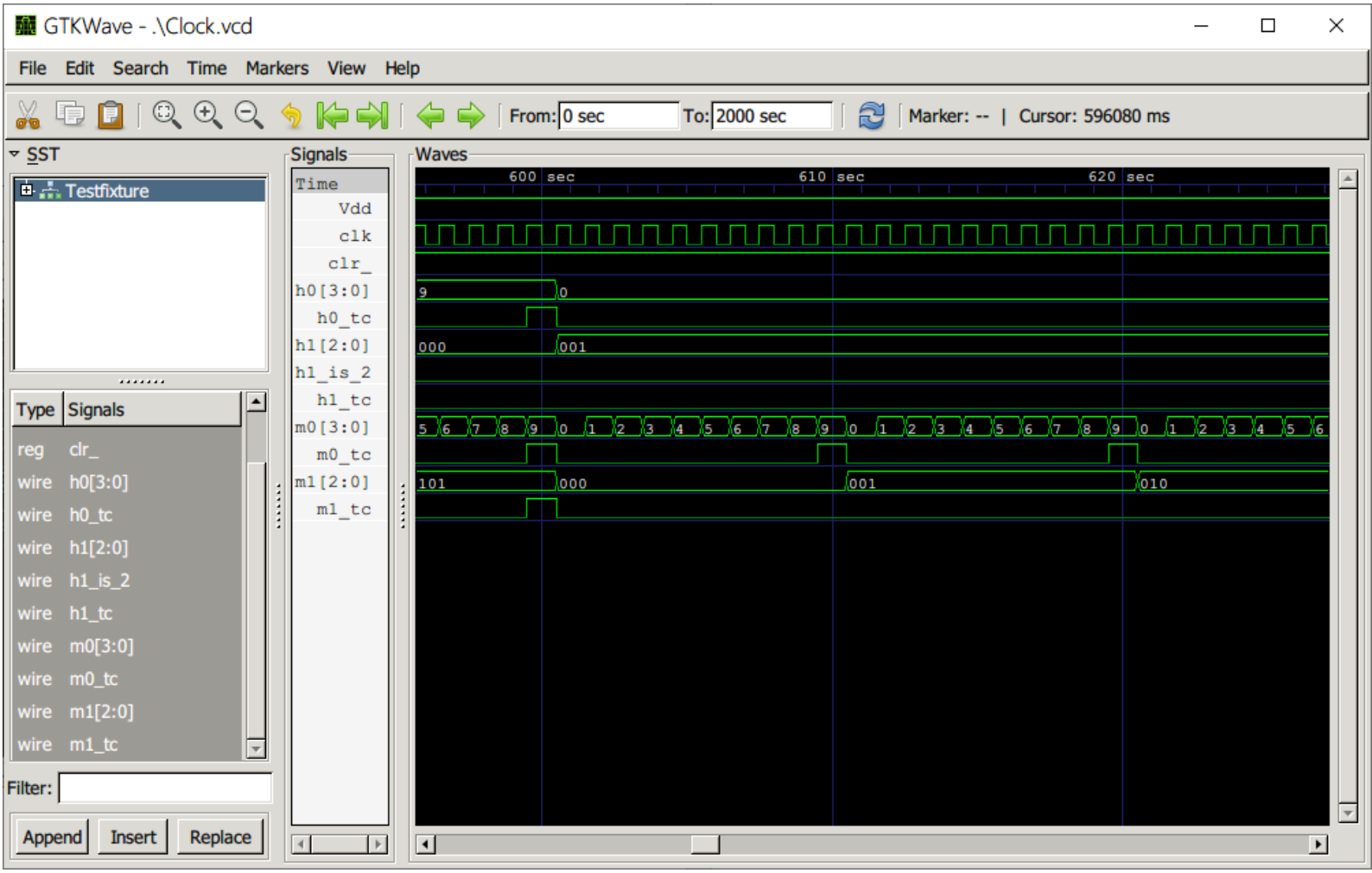
```
end

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