

數位電路實驗 HW6: 24-hour Digital Clock

第十四組：A1055547 王公志、A1073136 黃柏盛

Abstract

透過 VHDL 在 FPGA 上做出顯示 00:00:00 ~ 23:59:59 的二十四小時時鐘。

Content

1. Principles:

由於 FPGA 上的七段顯示器一次只能顯示一個，為了讓顯示器實際上看起來是真的時鐘，因此只要讓指定的六個顯示器不斷地快速來回顯示，就能達到目的。

2. Method:

為了讓顯示器不會因為切得太快而漏掉某些數字，因而採用類似除頻器的作法。且考慮到有六個七段顯示器，因此每隔 60 個 cycle 將秒數個位數 (cnt_sec_digs) 加一，並依照時鐘的進位規則累進至時鐘各位數：cnt_sec_tens, cnt_min_digs, cnt_min_tens, cnt_hr_digs, cnt_hr_tens，如此一來每個位數的數字由於重複顯示多次，因此在高頻率下都能盡量顯示明確。

在每個 cycle 將時鐘各位數轉換為 8-bit 長的七段顯示器相對應 output，並在不同 cycle 輪流顯示不同的七段顯示器，使得人眼在高頻率 clock 的狀況下，能夠看起來像是真的時鐘。

3. VHDL 程式碼：

```
1  module clock_24hr(clk, rst,  
2      SEL, out,  
3      cnt_hr_tens, cnt_hr_digs,  
4      cnt_min_tens, cnt_min_digs,  
5      cnt_sec_tens, cnt_sec_digs,  
6      mul_out);  
7    
8      input clk, rst;  
  
10     output reg [2:0] SEL; // 七段顯示器的COM選擇腳位  
11     output reg [7:0] out; // 七段顯示器的輸出腳位  
12   
13     // 各個位數的counter，紀錄實際上的數字。  
14     output reg [3:0] cnt_hr_tens, cnt_hr_digs,  
15                     cnt_min_tens, cnt_min_digs,  
16                     cnt_sec_tens, cnt_sec_digs,  
17                     mul_out;  
18     // mul_out 只是方便在模擬時方便觀察output是否正確。  
  
20     // 用來存放decoder的output  
21     // (decoder將時鐘各個位數轉換為七段顯示器binary code)  
22     reg [7:0] hr_tens, hr_digs,  
23             min_tens, min_digs,  
24             sec_tens, sec_digs;  
25     reg [2:0] sel;  
26     // cnt為除頻器所用，每過十個cnt(clock)，則second的個位數加一。  
27     reg [5:0] cnt;  
  
29     // 將input轉換為七段顯示器輸出腳位形式(a, b, ..., g & dp)  
30     function [7:0] decoder;  
31         input [3:0] count;  
32         begin  
33             case (count)  
34                 0: decoder = 8'b11111100;  
35                 1: decoder = 8'b01100000;  
36                 2: decoder = 8'b11011010;  
37                 3: decoder = 8'b11110010;  
38                 4: decoder = 8'b01100110;  
39                 5: decoder = 8'b10110110;  
40                 6: decoder = 8'b10111110;  
41                 7: decoder = 8'b11100000;  
42                 8: decoder = 8'b11111110;  
43                 9: decoder = 8'b11110110;  
44                 default: decoder = 8'b10011110; // set default to 'E'  
45             endcase  
46         end  
47     endfunction
```

```

49 initial begin
50     cnt_hr_tens = 0;
51     cnt_hr_digs = 0;
52     cnt_min_tens = 0;
53     cnt_min_digs = 0;
54     cnt_sec_tens = 0;
55     cnt_sec_digs = 0;
56     sel = 0;
57     cnt = 0;
58 end
59
60 always @ (posedge clk, negedge rst) begin
61     if (rst == 0) begin
62         cnt_hr_tens = 0;
63         cnt_hr_digs = 0;
64         cnt_min_tens = 0;
65         cnt_min_digs = 0;
66         cnt_sec_tens = 0;
67         cnt_sec_digs = 0;
68         sel = 0;
69         cnt = 0;
70     end
71
72     else begin
73         cnt <= cnt + 1;
74         // 60_divider, cnt_sec_digs(個位數) + 1 every 60 cycle.
75         if (cnt == 59) begin // 由於有六個七段顯示器，
76             cnt <= 0; // 因此每60個cycle再將秒數的個位數+1，
77             cnt_sec_digs <= cnt_sec_digs + 1; // 可以使各個位數的顯示較明確一些。
78             if (cnt_sec_digs == 9) begin
79                 cnt_sec_tens <= cnt_sec_tens + 1;
80                 cnt_sec_digs <= 0;
81                 if (cnt_sec_tens == 5) begin
82                     cnt_min_digs <= cnt_min_digs + 1;
83                     cnt_sec_tens <= 0;
84                     if (cnt_min_digs == 9) begin
85                         cnt_min_tens <= cnt_min_tens + 1;
86                         cnt_min_digs <= 0;
87                         if (cnt_min_tens == 5) begin
88                             cnt_hr_digs <= cnt_hr_digs + 1;
89                             cnt_min_tens <= 0;
90                             if (cnt_hr_tens == 2 && cnt_hr_digs == 3) begin
91                                 cnt_hr_tens <= 0;
92                                 cnt_hr_digs <= 0;
93                             end
94                             else if (cnt_hr_tens != 2 && cnt_hr_digs == 9) begin
95                                 cnt_hr_digs <= 0;
96                             end
97                         end
98                     end
99                 end
100             end

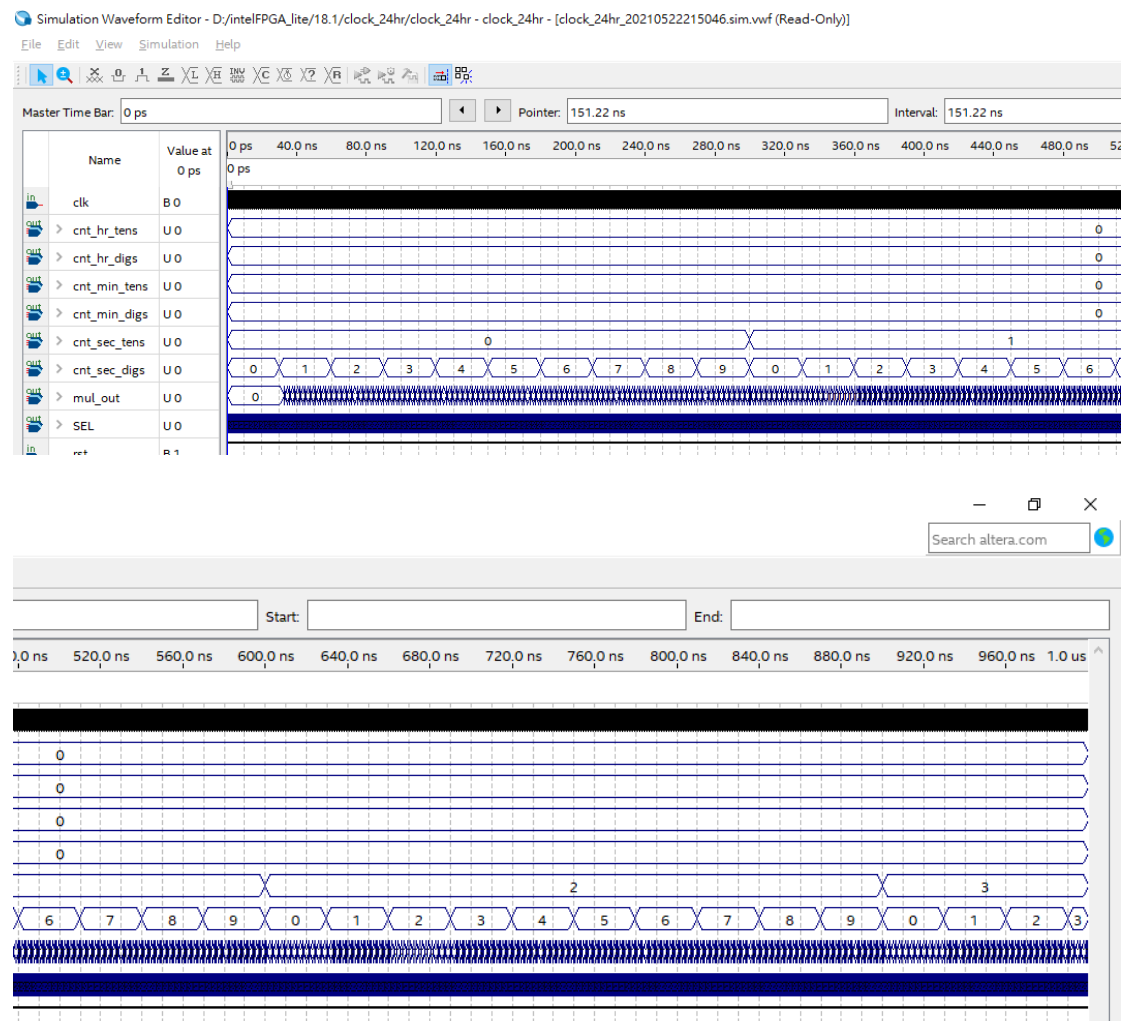
```

```

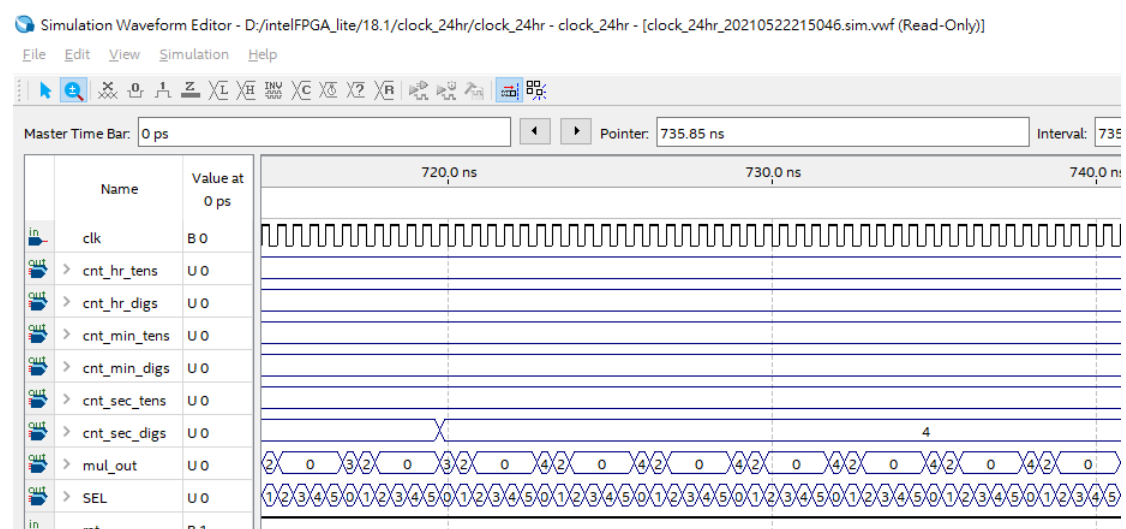
102     sel = sel + 1; // 每個cycle都切換到下一位數的七段顯示器
103     if(sel == 6) begin
104         sel = 0;
105     end
106     case (sel)
107         0: SEL = 3'b000; // 七段顯示器中second的個位數
108         1: SEL = 3'b001; // second的十位數
109         2: SEL = 3'b010; // minute的個位數
110         3: SEL = 3'b011; // minute的十位數
111         4: SEL = 3'b100; // hour的個位數
112         5: SEL = 3'b101; // hour的十位數
113         default: SEL = 3'b000;
114     endcase
115
116     // call function "decoder",
117     // decode binary number to 7-segment-display bit.
118     sec_digs = decoder(cnt_sec_digs);
119     sec_tens = decoder(cnt_sec_tens);
120     min_digs = decoder(cnt_min_digs);
121     min_tens = decoder(cnt_min_tens);
122     hr_digs = decoder(cnt_hr_digs);
123     hr_tens = decoder(cnt_hr_tens);
124     case (SEL) // 每個cycle都切換一次七段顯示器
125         3'b000: begin
126             out = sec_digs;
127             mul_out = cnt_sec_digs;
128         end
129         3'b001: begin
130             out = sec_tens;
131             mul_out = cnt_sec_tens;
132         end
133         3'b010: begin
134             out = min_digs;
135             mul_out = cnt_min_digs;
136         end
137         3'b011: begin
138             out = min_tens;
139             mul_out = cnt_min_tens;
140         end
141         3'b100: begin
142             out = hr_digs;
143             mul_out = cnt_hr_digs;
144         end
145         3'b101: begin
146             out = hr_tens;
147             mul_out = cnt_hr_tens;
148         end
149         default: begin
150             out = 8'b10011110; // default顯示'E'
151             mul_out = 0;
152         end
153     endcase
154 end
155 end
156 endmodule

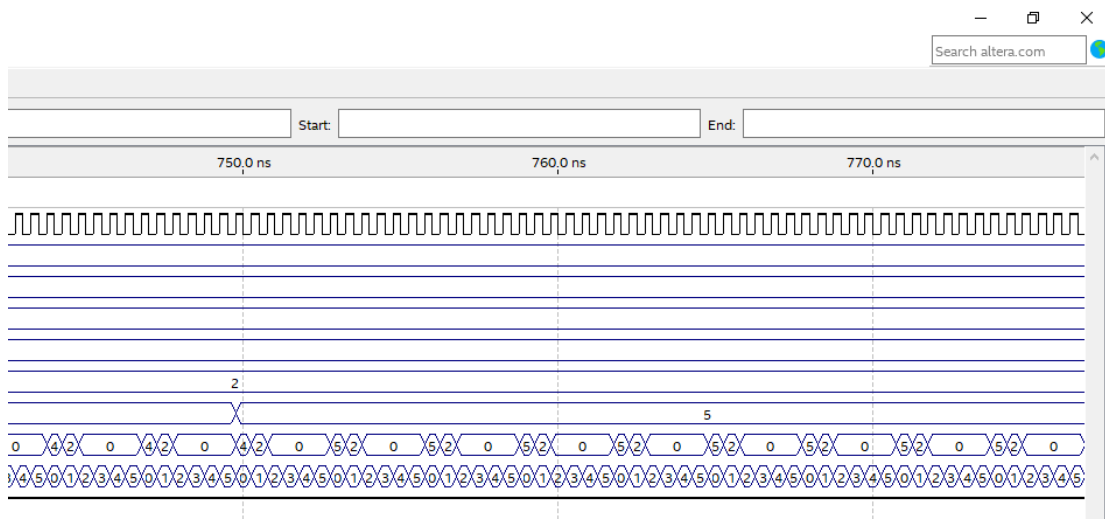
```

4. 模擬結果：



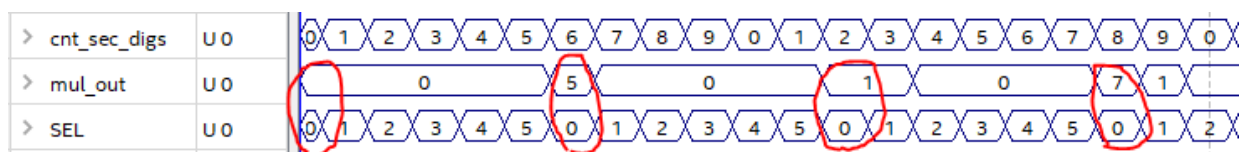
放大顯示↓



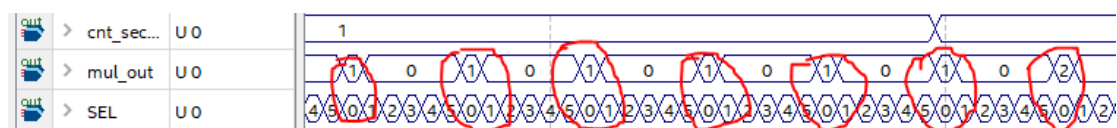


Discussions

由於最初沒有考慮到七段顯示器不能同時顯示的問題，所以最初是採用沒有除頻器且每個 cycle 切換顯示器的作法，因此輸出結果會有數字跳號的問題，像是以下的情形：



在圖中可以發現，每當 SEL 為 0(選擇到 second 個位數)時，數字已經跳了好幾號(0→5→1→7)，雖然時鐘的進位是正確的，且在 clock 頻率很高時，人眼較難注意到時鐘是否正確顯示，但實際上顯示出來的數字是漏掉許多號的，仍不符合理想的情形。因此必須採用除頻器的作法，讓同一個位數的數字顯示多次，且搭配高頻率 clock，才能使七段顯示器時鐘的顯示看起來更合理。



如上圖所示，個位數的 1 在重複出現數次後，才往前加到 2。