東海大學資訊工程與科學系

專題報告

FPGA-紅綠燈訊號控制

指導教授 : 廖啟賢

學生 :952864 徐睿霆

中華民國 一百 年 一 月 五 日

目次

一、摘要

在上學期接觸了 FPGA 的課程後，不僅認識了 FPGA 也了解了更多關於這方面的知識，課中經過幾個簡單的實作練習後，想到其實在生活中許多周遭的電子產品，都可以利用發展工具來模擬、設計。

在 FPGA 的多媒體開發平台上，有 LED 燈，有七段顯示器，所以想到利用這些來模擬紅綠燈的運作，並加入控制的功能，來增加用路的便利性，例如，可延長綠燈時間的按鈕，讓身障者過馬路的時間更充裕，或者某些路口可切換正常紅綠燈功能，或改閃黃燈、閃紅燈，以及行人穿越用的燈號，使一些路口的用路更有彈性，所以想到了用 FPGA 的發展工具模擬紅綠燈，再使用按鈕或 SWITCH 開關來切換不同的運作模式。

二、前言

剛開始的構想是利用多媒體開發平台上不同顏色的的 LED 燈來表示紅、黃、綠三種燈號，再利用七段顯示器表現紅燈及綠燈時間的倒數，按鈕 1 可以縮短另一組紅燈時間，按鈕 2 可延長該組綠燈時間，SWITCH 開關可切換紅綠燈的運作模式，可改為閃黃燈，則另一組閃紅燈，但因為這學期時間上有許多狀況耽誤，所以只先進行模擬紅綠燈運作，秒數倒數，及 SWITCH 開關切換模式的部分。

三、背景知識介紹

以下是專題內所用到的開發工具、軟體，以及相關的專有名詞的介紹。

1. FPGA (Field Programmable Gate Array)

現場可程式邏輯閘陣列，是一個含有可編輯元件的半導體設備，可供使用者現場程式化的邏輯閘陣列元件。目前以硬體描述語言（Verilog 或 VHDL）所完成的電路設計，可以經過簡單的綜合與佈局，快速的燒錄至 FPGA 上進行測試，是現代 IC 設計驗證的技術主流。這些可編輯元件可以被用來實現一些基本的邏輯閘電路（比如 AND、OR、XOR、NOT）或者更複雜一些的組合功能比如解碼器或數學方程式。在大多數的 FPGA 裡面，這些可編輯的元件里也包含記憶元件例如觸發器（Flip－flop）或者其他更加完整的記憶塊。

系統設計師可以根據需要通過可編輯的連接把 FPGA 內部的邏輯塊連接起來，就好像一個電路試驗板被放在了一個晶片裡。一個出廠後的成品 FPGA 的邏輯塊和連接可以按照設計者而改變，所以 FPGA 可以完成所需要的邏輯功能。

2. DE2-70

Altera DE2-70 多媒體開發平台配備了數量高達 70,000 個邏輯單元的 Altera Cyclone® II 2C70 和更大容量的記憶體元件，並完全承襲了 Altera DE2 多媒體平台豐富的多媒體、儲存及網路等應用介面的優點。

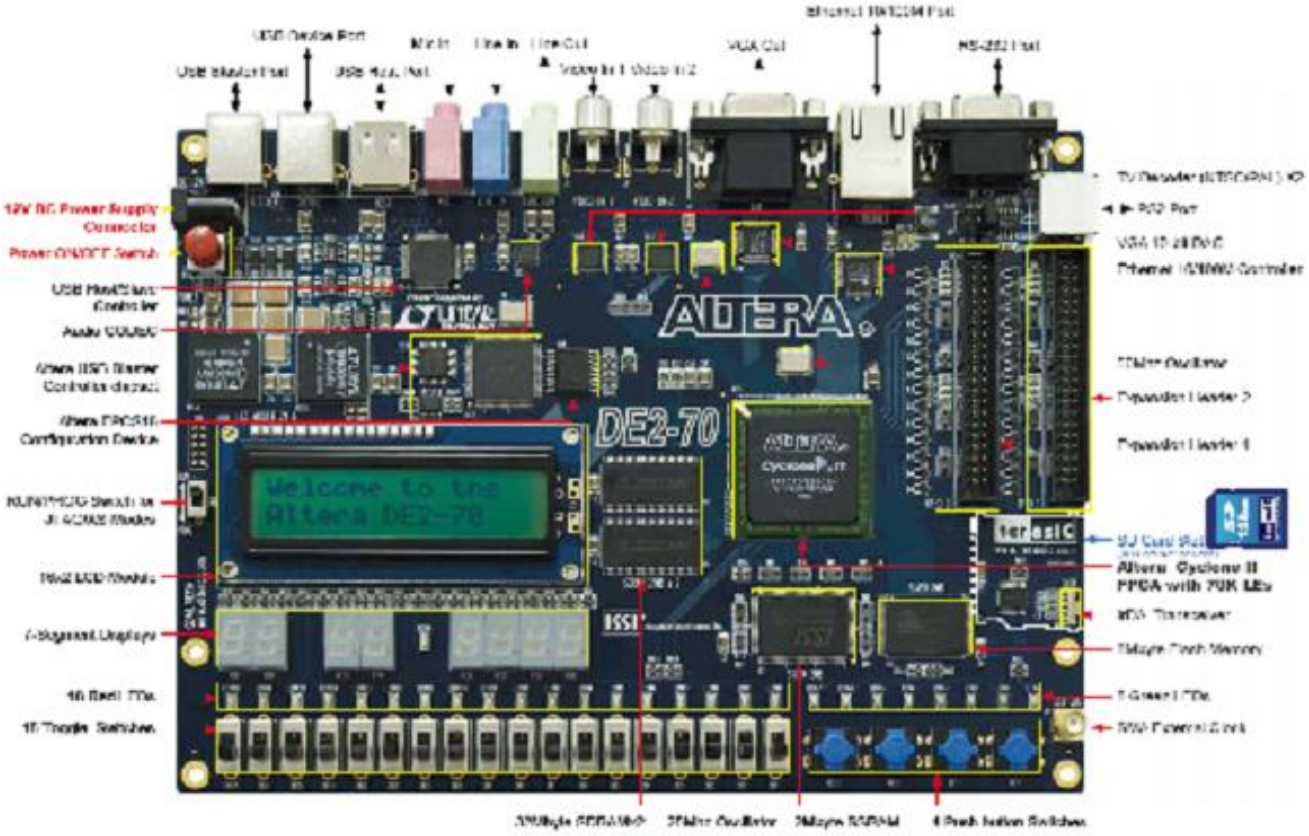


圖 3-1 ： Altera DE2-70 多媒體開發平台。

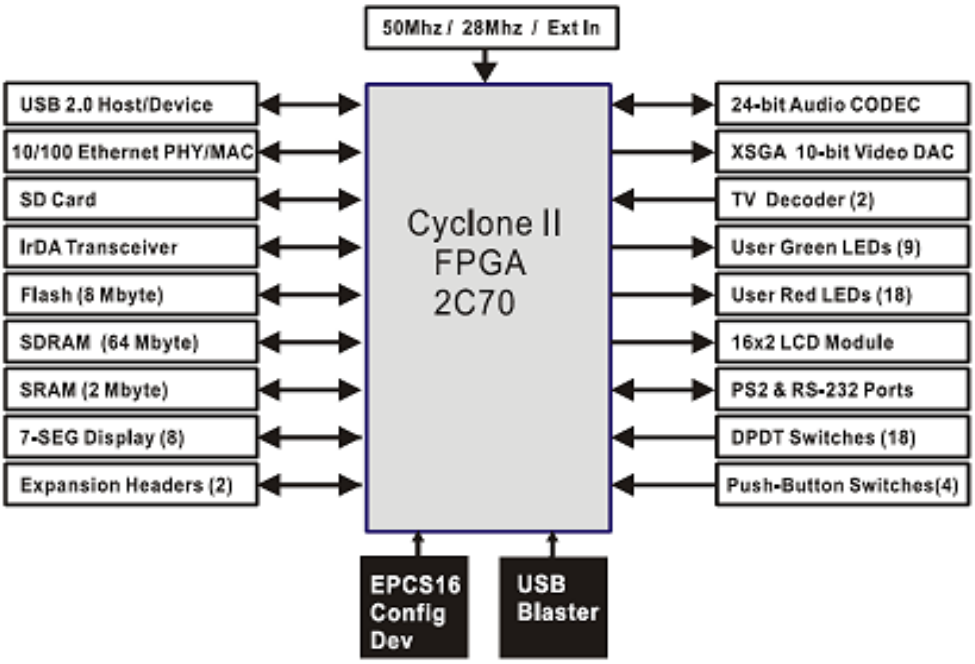


圖 3-2: Block diagram of the DE2-70 board.

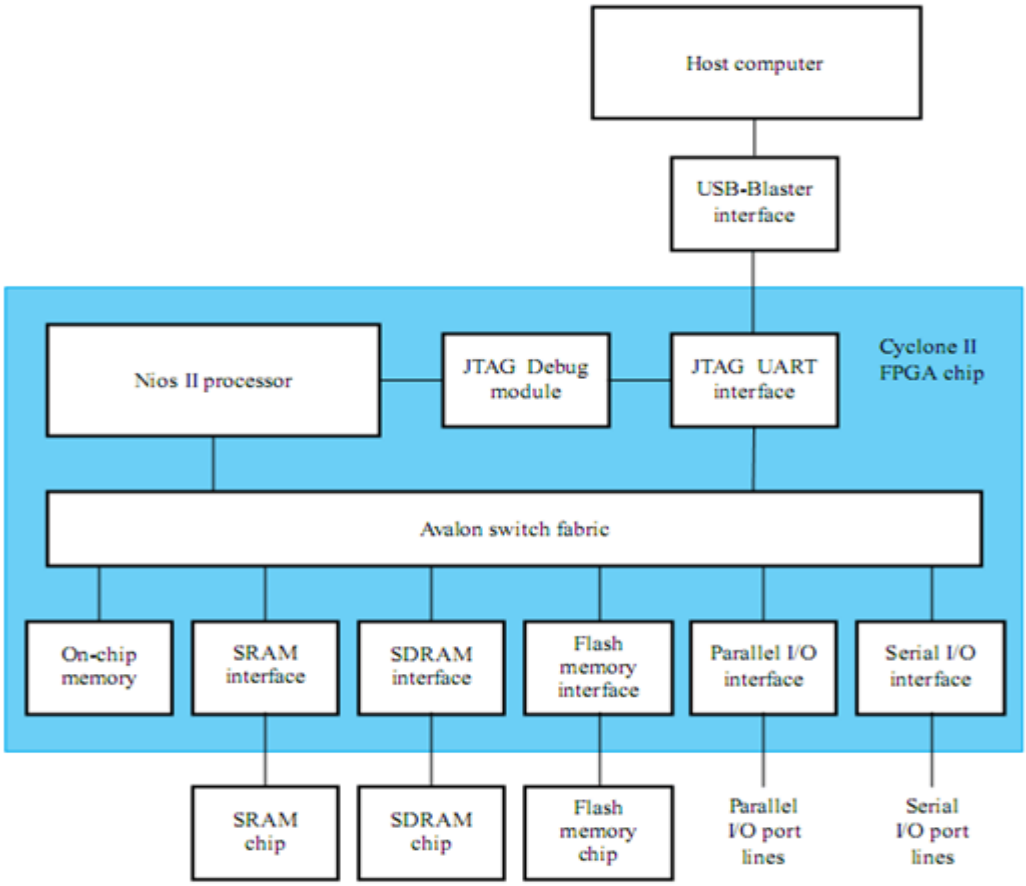


圖 3-3 ： DE2-70 系統架構圖

3. Quartus II

Altera 的 QuartusII 設計軟體提供一個非常容易適應特定設計所需要的完整的多平臺設計環境。它是一個可程化晶片系統(SOPC)設計的綜合性環境。QuartusII 軟體包括 FPGA 和 CPLD 設計所有階段的解決方案。QuartusII 軟體包含 QuartusII 圖形用戶介面、EDA 工具介面或指令行介面，可在整個流程中僅使用其中一種介面，也可以在設計流程中的不同階段，依各人喜好使用不同的選項。

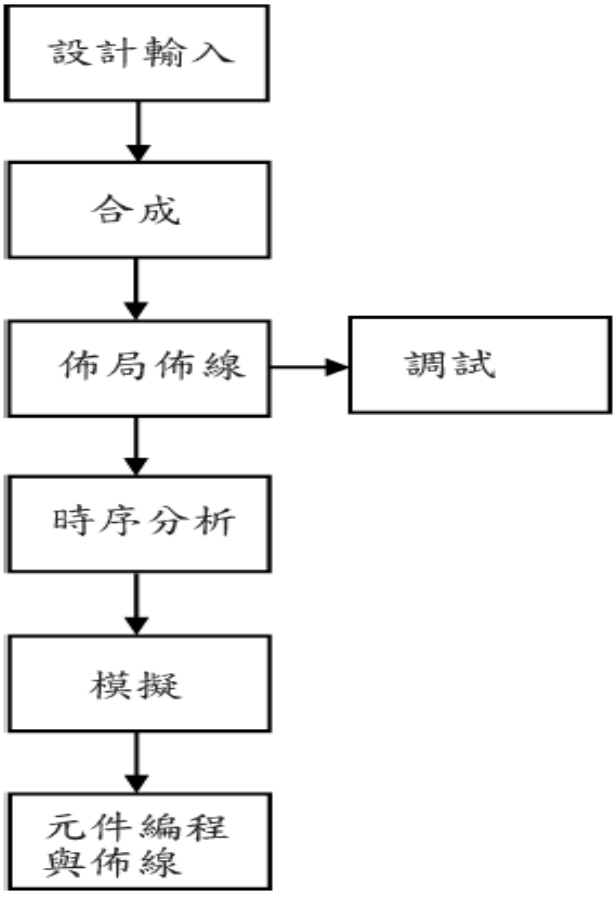


圖 3-4: Quartus II 設計流程

4. VHDL

VHDL 是 Very High Speed Integrated Circuit (VHSIC)Description Language 的縮寫，即超高速集成電路硬體描述語言。 簡單的說，它就是一種專門用來設計硬體電路的語言。在基於 CPLD／FPGA 和 ASIC 的數位系統設計中有著廣泛的應用。VHDL 語言誕生於 1983 年，1987 年被美國國防部和 IEEE 確定為標準的硬體描述語言。使用 VHDL 作數位電路設計，只需依據標準的 VHDL 語言規範，描述複雜的電路系統，再用一般軟體的模組化觀念，描述數位系統的規格及功能,接著利用軟體(可在 PC 或工作站上執行)將所寫的 VHDL 編譯合成電子電路，可以節省人工將傳統電子元件拼湊成電路的時間。VHDL 最基本的電路設計可分為二大部份，一是 entity declaration，另一個部分是 architecture body。

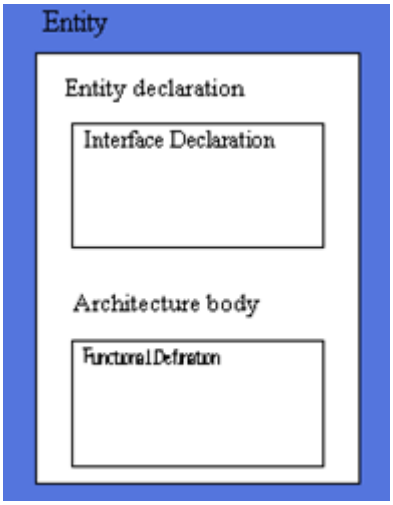


圖 3-5

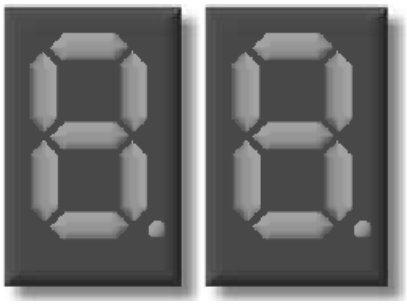
四、設計原理分析

以六個 LED 燈代表兩組紅綠燈，四個七段顯示器，每兩個為一組顯示倒數的秒數，再以 SWITCH 開關來切換，DIP 為 0 時以代表白天使用正常紅綠燈模式， DIP 為 1，表示晚上改成閃黃燈模式。

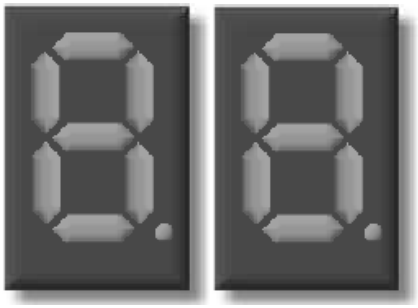
其顯示狀況為 ：

綠燈 1　黃燈 1　紅燈 1

綠燈 2　黃燈 2　紅燈 2



第一組紅綠燈的倒數秒數



第二組紅綠燈的倒數秒數

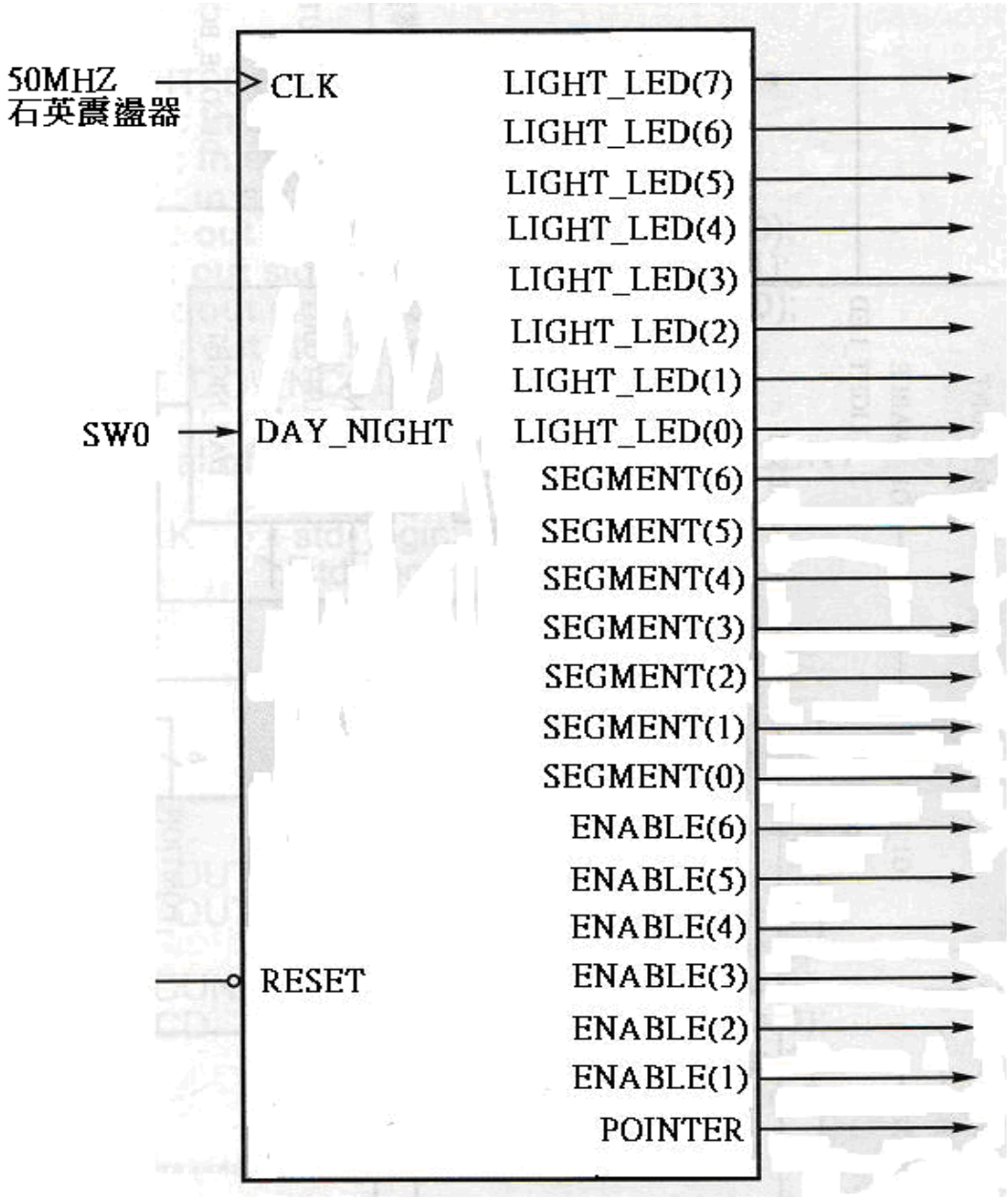


圖 4-1 ： 控制電路方塊總圖

五、程式說明

1. Time Base Generator

```
42    --************************
43    --* Time Base Generator *
44    --************************
45    begin
46
47       process(CLK,RESET)
48          begin
49             if RESET = '0' then
50                DIVIDER <= "00000000000000000000000000";
51             elsif CLK'event and CLK = '1' then
52                DIVIDER <= DIVIDER + 1;
53             end if;
54       end process;
55    SCAN_CLK    <= DIVIDER(15);
56    YELLOW_CLK  <= DIVIDER(23);
57    CLOCK       <= DIVIDER(25);
```

此為一除頻電路，它是負責將 50MHZ 石英振盪頻率低，產生紅綠燈計秒時序 CLOCK、黃燈閃爍的時序 YELLOW_CLK 與七段顯示器掃描時序 SCAN_CLK。方塊電路如圖 5-1。

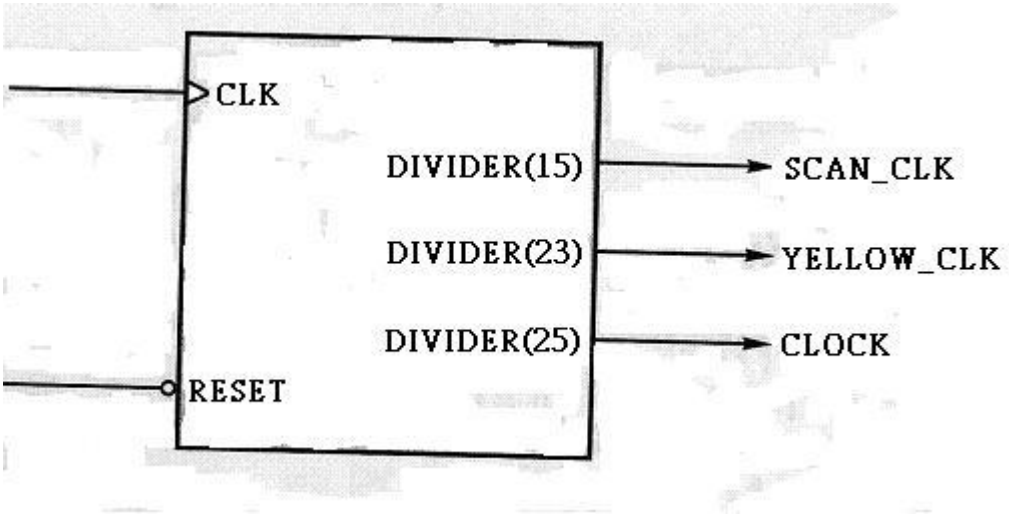
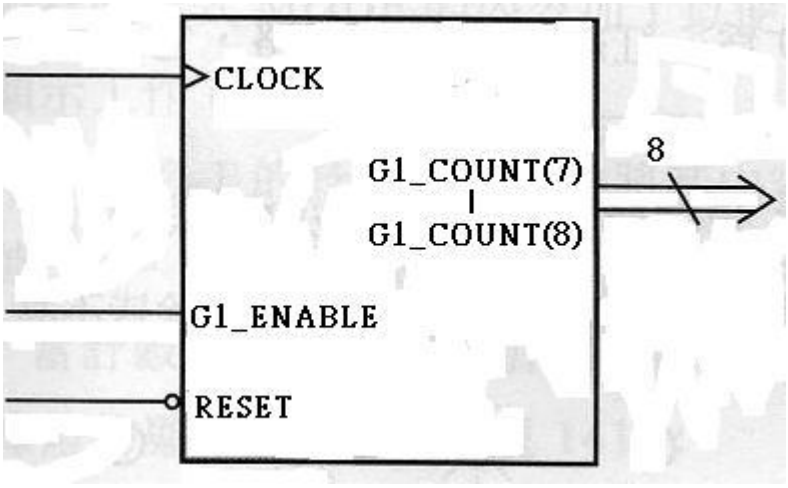


圖 5-1

2. Green Light1 Down Counter And Auto Reload

```
59    --*********************************************************
60    --* Green Light1 Down Counter And Auto Reload(30 Second) *
61    --*********************************************************
62
63    process(CLOCK,RESET,G1_ENABLE)
64       begin
65          if RESET = '0' then
66             G1_COUNT <= "00000000";
67          elsif G1_ENABLE = '1' then
68             if CLOCK'event and CLOCK = '1' then
69                if G1_COUNT = "00000000" or G1_COUNT_OUT = "11111111" then
70                   G1_COUNT <= "00000000";
71                else
72                   if G1_COUNT(3 downto 0) = "0000" then
73                      G1_COUNT(3 downto 0) <= "1001";
74                      G1_COUNT(7 downto 4) <= G1_COUNT(7 downto 4)- 1;
75                   else
76                      G1_COUNT(3 downto 0) <= G1_COUNT(3 downto 0) -1;
77                   end if;
78                   if G1_COUNT(7 downto 4) = "1001" then
79                      G1_COUNT(7 downto 4) <= "0011";
80                   end if;
81                end if;
82             end if;
83          end if;
84    end process;
```

為一個可自動載入的綠燈 1 倒數計時器，其電路方塊如圖 5-2



3. Green Light2 Down Counter And Auto Reload

```
86  --*********************************************************
87  --* Green Light2 Down Counter And Auto Reload(30 Second)*
88  --*********************************************************
89
90  process(CLOCK,RESET,G2_ENABLE)
91     begin
92        if RESET = '0' then
93           G2_COUNT <= "00000000";
94        elsif G2_ENABLE = '1' then
95           if CLOCK'event and CLOCK = '1' then
96              if G2_COUNT = "00000000" or G2_COUNT_OUT = "11111111" then
97                 G2_COUNT <= "00110000";
98              else
99                 if G2_COUNT(3 downto 0) = "0000" then
100                   G2_COUNT(3 downto 0) <= "1001";
101                   G2_COUNT(7 downto 4) <= G2_COUNT(7 downto 4)- 1;
102                else
103                   G2_COUNT(3 downto 0) <= G2_COUNT(3 downto 0) -1;
104                end if;
105                if G2_COUNT(7 downto 4) = "1001" then
106                   G2_COUNT(7 downto 4) <= "0011";
107                end if;
108             end if;
109          end if;
110       end if;
111 end process;
```

為一個可自動載入的綠燈 2 倒數計時器，其電路方塊如圖 5-2

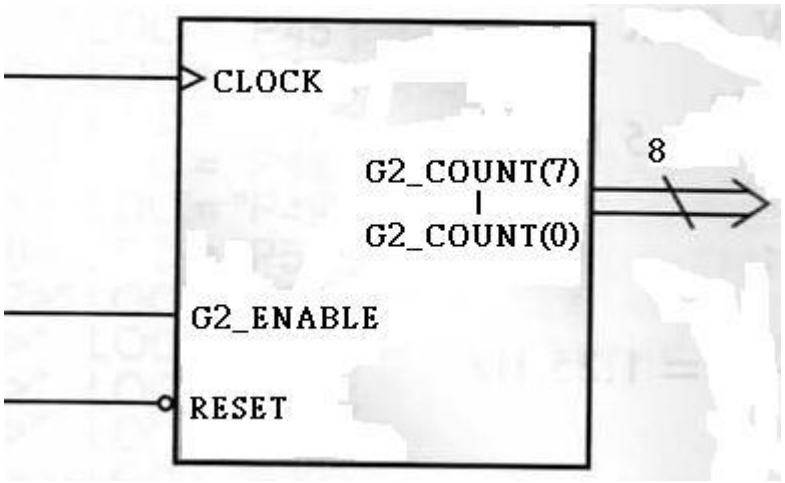


圖 5-2

4. Traffic Light Control

```
113     --*************************
114     --* Traffic Light Control *
115     --*************************
116
117     process(YELLOW_CLK,RESET,DAY_NIGHT)
118
119        begin
120           if RESET = '0' then
121              LIGHT <= "11011011";
122              MODE <= "000";
123              YELLOW_COUNT <= "1111";
124              G1_ENABLE <= '0';
125              G2_ENABLE <= '0';
126           elsif DAY_NIGHT = '1' then
127              if YELLOW_CLK'event and YELLOW_CLK = '1' then
128                 case MODE is
129                    when "000" =>
130                       LIGHT <= "11011011";
131                       G1_ENABLE <= '1';
132                       if G1_COUNT = "00000101" then
133                          MODE <= MODE + 1;
134                       end if;
135                    when "001" =>
136                       if G1_COUNT = "00000000" then
137                          MODE <= MODE + 1;
138                          G1_ENABLE <= '0';
139                       end if;
140                    when "010" =>
141                       LIGHT <= "10111011";
142                       if YELLOW_COUNT = "0000" then
143                          YELLOW_COUNT <= "1111";
144                          MODE <= MODE + 1;
145                       else
146                          YELLOW_COUNT <= YELLOW_COUNT - 1;
147                       end if;
148                    when "011" =>
149                       LIGHT <= "01111110";
150                       G2_ENABLE <= '1';
151                       if G2_COUNT = "00000101" then
```

```
152                          MODE <= MODE + 1;
153                       end if;
154                    when "100" =>
155                       if G2_COUNT = "00000000" then
156                          MODE <= MODE + 1;
157                          G2_ENABLE <= '0';
158                       end if;
159                    when others =>
160                       LIGHT <= "01111101";
161                       if YELLOW_COUNT = "0000" then
162                          YELLOW_COUNT <= "1111";
163                          MODE <= "000";
164                       else
165                          YELLOW_COUNT <= YELLOW_COUNT - 1;
166                       end if;
167                 end case;
168              end if;
169           else
170              LIGHT <= '1' & CLOCK & "1111" & CLOCK & '1';
171        end if;
172     end process;
173     G1_LIGHT       <= YELLOW_CLK when MODE = "001" else LIGHT(5);
174     G2_LIGHT       <= YELLOW_CLK when MODE = "100" else LIGHT(0);
175     G1_COUNT_OUT   <= G1_COUNT when DAY_NIGHT = '1' else "11111111";
176     G2_COUNT_OUT   <= G2_COUNT when DAY_NIGHT = '1' else "11111111";
177     LIGHT_LED      <= LIGHT(7 downto 6) & G1_LIGHT & LIGHT(4 downto 1)& G2_LIGHT;
```

此部分為一紅綠燈控制電路，它是依照計數模式 MODE 的內容來決定紅綠燈的顯示動作。

120~125 行 ： 當硬體執行重置 RESET，則

(1) 設定紅綠燈為綠燈 1 及綠燈 2 亮，其餘不亮。

(2) 設定紅綠燈的計數模式為”000”,以便重頭開始執行。

(3) 設定黃燈顯示倒數計數值為”1111”以便往後倒數計數

(4) 設定綠燈 1 及綠燈 2 的 G1_ENABLE、G2_ENABLE 為”0”，以便禁止兩個計時器計時。

126 行判斷若 DAY_NIGHT=”1”代表白天，因此往下執行。

128~169 行依 MODE 的內容執行紅綠燈動作，整個紅綠燈控制電路的主軸為計數模式 MODE，對應顯示狀況如下圖，

MODE 計數值	紅綠燈的顯示狀況
0 0 0	綠燈 1 亮，紅燈 2 亮
0 0 1	綠燈 1 閃爍，紅燈 2 亮
0 1 0	黃燈 1 亮，紅燈 2 亮
0 1 1	紅燈 1 亮，綠燈 2 亮
1 0 0	紅燈 1 亮，綠燈 2 閃爍
1 0 0	紅燈 1 亮，黃燈 2 亮

圖 5-3

5. 多工器

```
173     G1_LIGHT        <= YELLOW_CLK when MODE = "001" else LIGHT(5);
174     G2_LIGHT        <= YELLOW_CLK when MODE = "100" else LIGHT(0);
175     G1_COUNT_OUT    <= G1_COUNT when DAY_NIGHT = '1' else "11111111";
176     G2_COUNT_OUT    <= G2_COUNT when DAY_NIGHT = '1' else "11111111";
177     LIGHT_LED       <= LIGHT(7 downto 6) & G1_LIGHT & LIGHT(4 downto 1)& G2_LIGHT;
```

第 173 行~第 176 行為多工器。

(1) 173 行的電路方塊如圖 5-4

圖 5-4

當 MODE=”001”時，綠燈 1 的輸出 G1_LIGHT 訊號為 YELLOW_CLK，因此綠燈 1 執行閃爍動作，當 MODE 內容不為”001”時，綠燈 1 的輸出 G1_LIGHT 訊號為正常的綠燈 1 LIGHT(5)顯示訊號。

(2) 174 行執行動作原理類似 173 行，電路方塊如圖 5-5

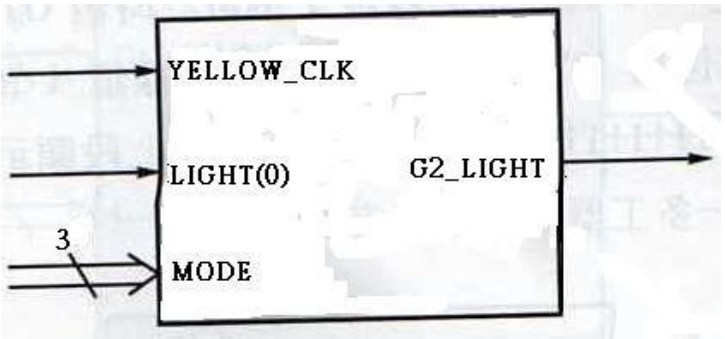


圖 5-5

(3) 第 175 行，當 DAY_NIGHT 電位為”1”代表白天，此綠燈 1 的倒數計時輸出 G1_COUNT_OUT 為正常綠燈 1 倒數計時器 G1_COUNT，當 DAY_NIGHT 電位為”0”代表晚上，因此綠燈 1 的 G1_COUNT 為”1111111”（七段顯示器皆不亮），其電路方塊如圖 5-6

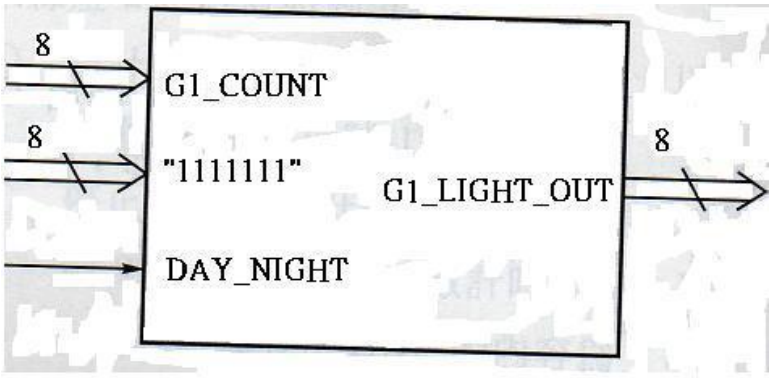


圖 5-6

(4) 176 行執行動作原理同第 175 行，其電路方塊如圖 5-7

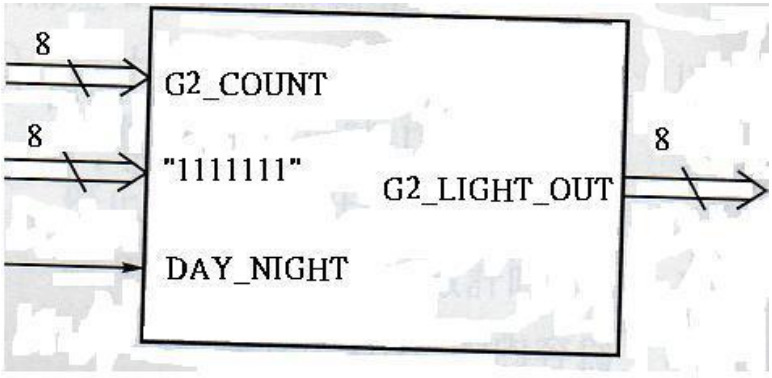


圖 5-7

(5) 第 177 行則將紅綠燈的控制輸出接到 LED 去顯示。

6. Decide Seven Segment Display Position

```
179     --****************************************
180     --* Decide Seven Segment Display Position *
181     --****************************************
182
183     process(SCAN_CLK,RESET)
184
185        begin
186           if RESET = '0' then
187              POSITION <= "111110";
188           elsif SCAN_CLK'event and SCAN_CLK = '1' then
189              case POSITION is
190                 when "111110" => POSITION <= "111101";
191                 when "111101" => POSITION <= "101111";
192                 when "101111" => POSITION <= "011111";
193                 when "011111" => POSITION <= "111110";
194                 when others   => POSITION <= "111110";
195              end case;
196           end if;
197     end process;
198     ENABLE <= POSITION;
```

為一七段顯示器的顯示位置掃描電路，電路方塊如圖 5-8

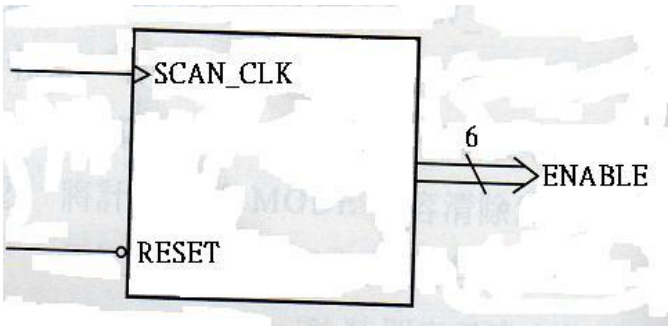


圖 5-8

7. Decide Seven Segment Display Data

```
200     --*************************************
201     --* Decide Seven Segment Display Data *
202     --*************************************
203
204     process(POSITION,G1_COUNT_OUT,G2_COUNT_OUT)
205
206           begin
207              case POSITION is
208                 when "111110" => DECODE_BCD <= G1_COUNT_OUT(3 downto 0);
209                 when "111101" => DECODE_BCD <= G1_COUNT_OUT(7 downto 4);
210                 when "101111" => DECODE_BCD <= G2_COUNT_OUT(3 downto 0);
211                 when "011111" => DECODE_BCD <= G2_COUNT_OUT(7 downto 4);
212                 when others   => null;
213              end case;
214        end process;
```

唯一配合前面七段顯示器的顯示位置掃描電路的顯示資料選

擇器，電路方塊如圖 5-9

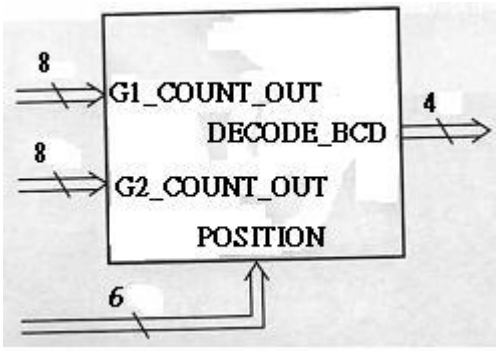


圖 5-9

8. BCD To Seven Segment Decoder

```
216     --*********************************
217     --* BCD To Seven Segment Decoder *
218     --*********************************
219
220     with DECODE_BCD select
221        SEGMENT <= "1000000" when "0000", --0
222                   "1111001" when "0001", --1
223                   "0100100" when "0010", --2
224                   "0110000" when "0011", --3
225                   "0011001" when "0100", --4
226                   "0010010" when "0101", --5
227                   "0000010" when "0110", --6
228                   "1111000" when "0111", --7
229                   "0000000" when "1000", --8
230                   "0010000" when "1001", --9
231                   "1111111" when others;
```

為一 BCD 對七段顯示器的解碼電路，其電路方塊如圖 5-10

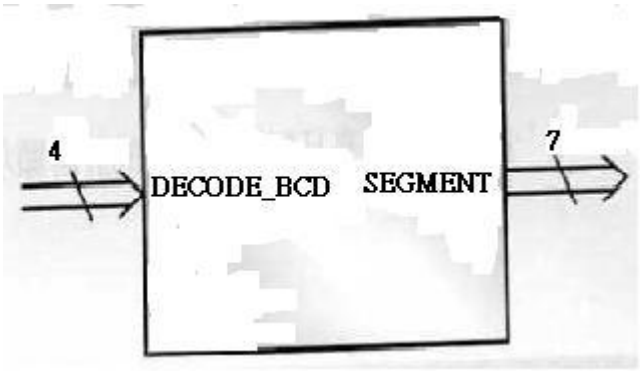


圖 5-10

9. 使用 Quartus II 將此程式編譯的結果，如圖 5-11

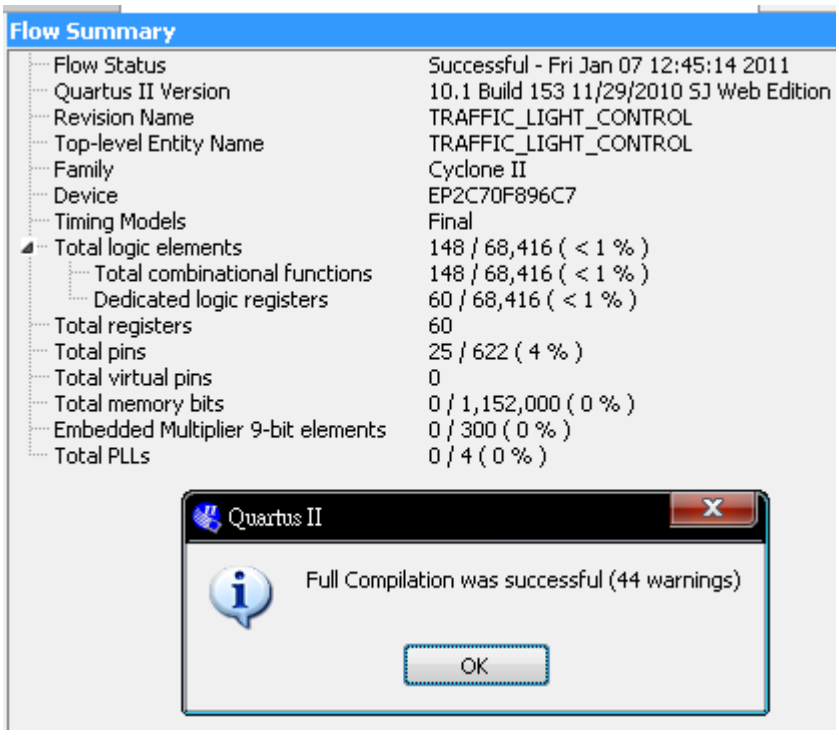


圖 5—11

此電路方塊如圖 5-12

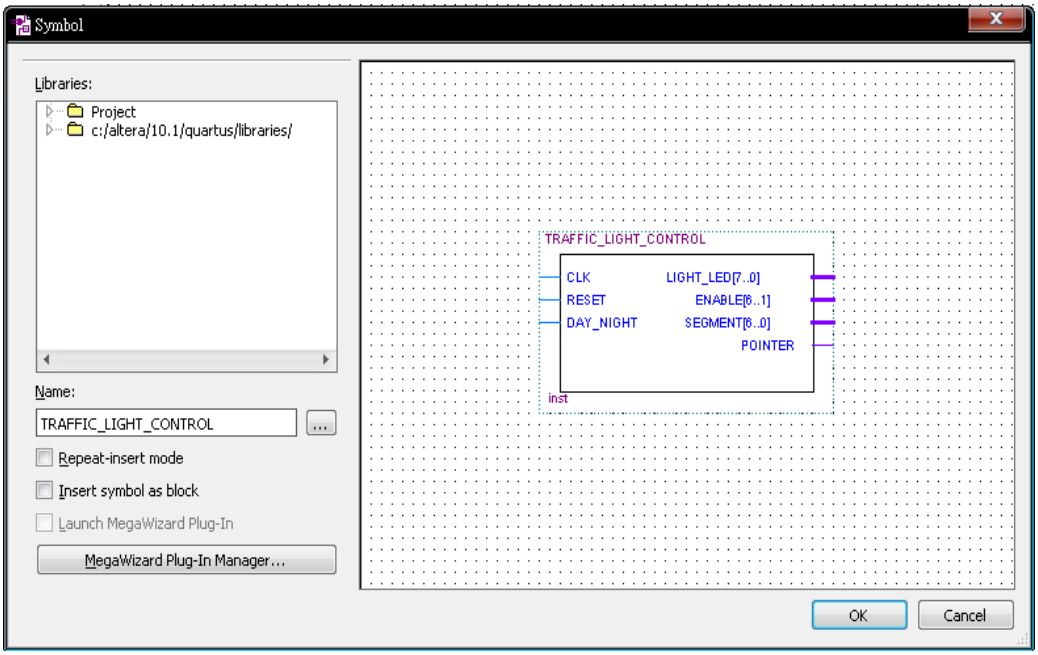


圖 5-12

六、結論

在此次專題中，程式設計的部分有查詢閱讀了許多關於 VHDL 語言的教學書籍及網路上的參考範例，把電路的部分用 VHDL 寫出來，還是有許多不熟悉的部分，常常需要找一些範例來幫助了解，原本是想用 VerilogHDL，但是書籍以及網路上的範例找到的以 VHDL 居多，所以這次專題是使用 VHDL。

原本許多功能的構想在這次專題中沒有實做出來，沒有足夠時間完成，整體規劃上也不夠詳細，不過在做這次專題的過程中，在資料查詢或程式的編寫時有對這部分的技術有更多的接觸，也進行更深入的了解，將來希望能利用 FPGA 在嘗試更多的設計,把一些想法透過 FPGA 來呈現。

七、參考文獻

1. 維基百科
http://zh.wikipedia.org/zh-tw/
2. 網路投影片
http://www.eel.tsint.edu.tw/teacher/wbchang/ES_PT/Ch2.pdf
3. 部落格 - 嵌入式平台實驗室
http://tw.myblog.yahoo.com/mathlab-net/
4. DE2-70 的參考文件 :
DE2_70_User_manual_v105
My First FPGA Design Tutorial
http://www.altera.com/
5. VHDL 簡介
http://access.ee.ntu.edu.tw/course/VLSI_design_89first/homework/project/vhdl_introduction.htm
6. **VHDL 硬體描述語言數位電路設計實務**
作者：鄭信源 出版社：儒林
7. **VHDL 與 FPGA 設計**
作者：胡振華 出版社：全華圖書
8. **FPGA 晶片設計與專題製作**
作者: 劉紹漢 出版社: 全華圖書
9. **VHDL 與 FPGA 設計**
作者：胡振華 出版社: 中國鐵道出版社