

臺北市立教育大學資訊科學系

數位電路實習報告

實習題目：行人號誌控制

學生姓名： 賴柔均、施尹涵、王若芸、張芸綺

班級：資科二

學號：u10016002、04、19、45

中 華 民 國 1 0 2 年 6 月 2 5 日

一、實習目的：

使用 VHDL、元件設計出紅綠燈實作

二、實習設備及使用零件：

1. 硬體設備
2. Quartus II
3. PC

三、工作原理及使用原件：

1. 工作原理

1. 兩位數七段顯示器:運用 clock 做觸發,當 clock=0 時顯示左邊的數字(十位數),當 clock=1 時顯示右邊的數字(個位數),將頻率調高到肉眼無法辨識切換間隔,就會看起來是亮兩位數
2. 下數計時器:運用許多正反器所組成,如果計數值依序遞減,則稱為下數計數器 (Down-counter)

狀態：

1. 一運作時，行人號誌狀態會閃紅燈；車向紅綠燈狀態會閃綠燈。
2. 按下 SW1，行人號誌狀態會從原先的閃紅燈變成紅燈；車向紅綠燈狀態會從原先的閃黃燈變成綠燈。
3. 行人號誌的狀態(亮紅燈)與車向紅綠燈的狀態(亮綠燈)接會持續 15 秒。
4. 默數完 15 秒後，行人號誌的狀態會從紅燈變成閃綠燈；車向紅綠燈的狀態會從原先的綠燈轉成亮黃燈 1 秒，再變成紅燈。除此之外，七段顯示器也會從 15 秒開始倒數，直到歸零。

	行人號誌	車向紅綠燈
原始狀態	閃紅燈	閃黃燈
按下按鈕	紅燈	綠燈(倒數 15 s)
按下過 15 s 後	綠燈(倒數 15 s)	黃燈維持 1s 轉紅燈

圖 1 紅綠燈狀態表格

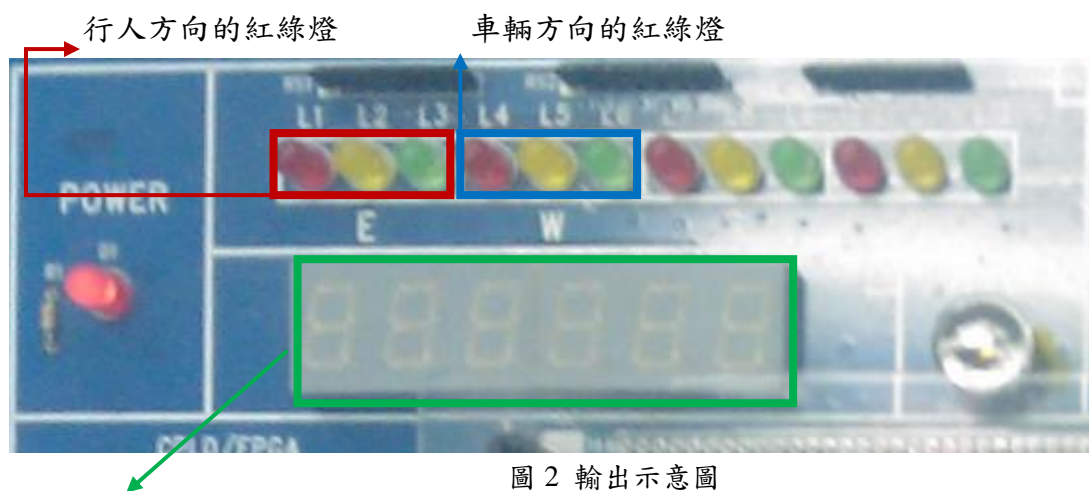


圖 2 輸出示意圖

紅路燈倒數秒數

2. 使用原件

1. 硬體：

a. LED 燈：L1 至 L6(pin 7 至 13)



Code	L1	L2	L3	L4	L5	L6	L7	L8
Device	Red	Yellow	Green	Red	Yellow	Green	Red	Yellow
Pin	Pin 7	Pin 8	Pin 9	Pin 10	Pin 11	Pin 12	Pin 13	Pin 14

圖 3 LED

b. 7-segment display : (pin 23 、 26 、 27 、 28 、 29 、 30 、 31 、 33)



Code	A	B	C	D	E	F	G	DP
Device	7 Segment Display							
Pin	Pin 23	Pin 26	Pin 27	Pin 28	Pin 29	Pin 30	Pin 31	Pin 32

Code	DE1	DE2	DE3	—	—	—	—	—
Device	74138			—	—	—	—	—
Pin	Pin33	Pin36	Pin37	—	—	—	—	—

† DE1, DE2 and DE3 are connected to 74138 which outputs Y0 ~ Y5 as C1 ~ C6.

† C1 ~ C6 are the common cathodes of 6 7-segment display.

圖 4 7-segment display

c. Data Switches: SW1(pin 47)

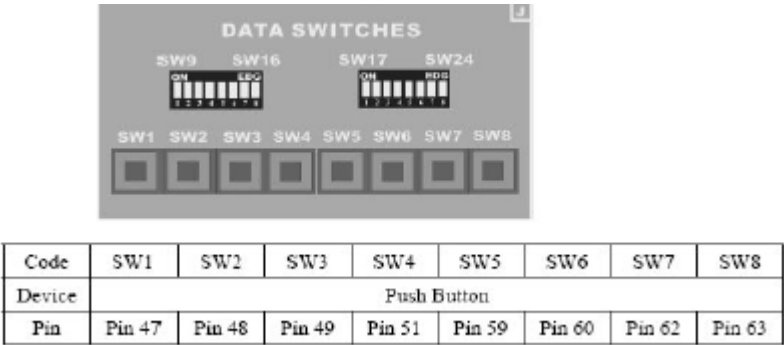


圖 5 Data Switches

2. 元件：

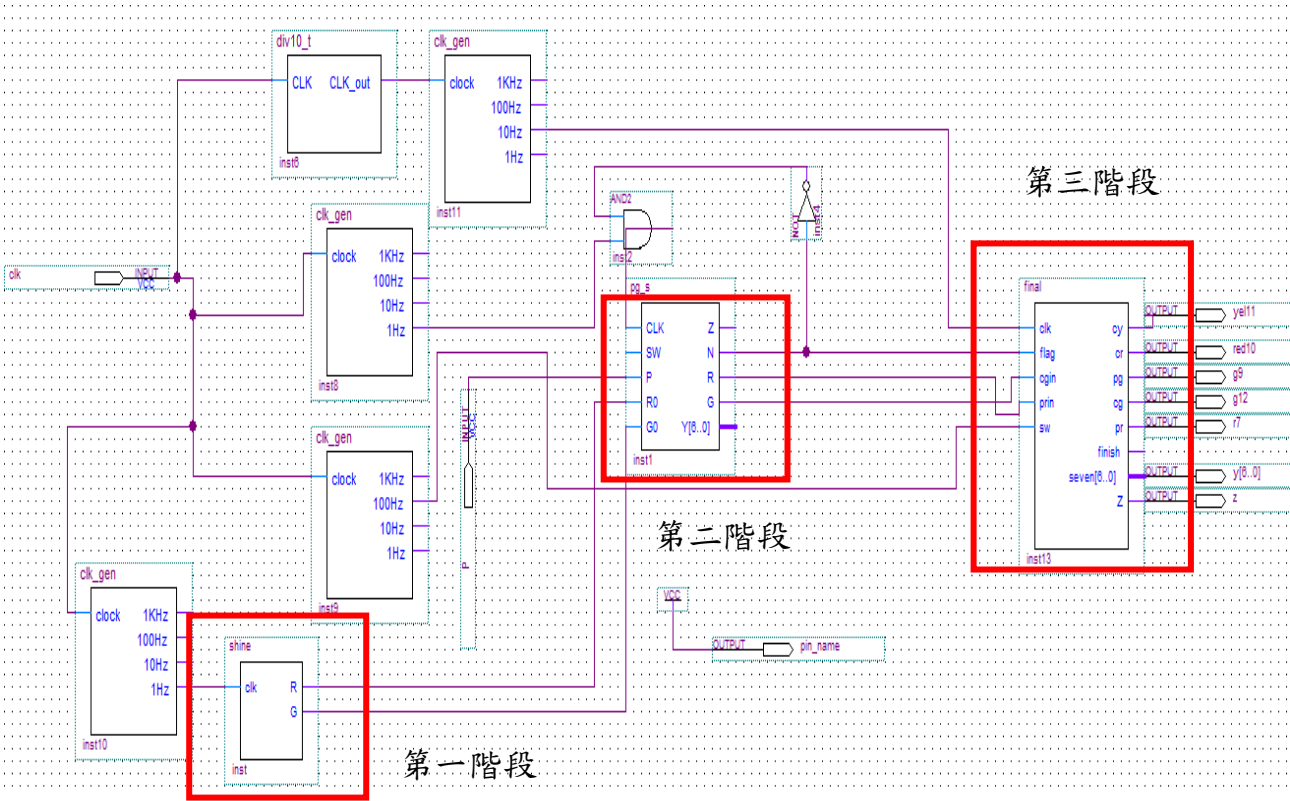


圖 6 電路圖

第一階段：一運作時，行人號誌狀態會閃紅燈；車向紅綠燈狀態會閃綠燈

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.all;
3
4  ==entity shine is
5  ==port( clk : IN STD_LOGIC;
6         R : OUT STD_LOGIC;
7         G : OUT STD_LOGIC);
8
9  end shine;
10
11 ==architecture a of shine is
12 SIGNAL Y,Z :STD_LOGIC;
13 ==begin
14 ==    process(clk,Y,Z)
15 ==    BEGIN
16 ==        if clk='1'then
17 ==            Y<='1';
18 ==            Z<='1';
19 ==        elsif clk='0' then
20 ==            Y<='0';
21 ==            Z<='0';
22 ==        end if;
23 ==        R<=Y;
24 ==        G<=Z;
25 ==    end process;
26
27 end a;

```

clk :計數器

R :行人狀態的紅燈

G :車向狀態的綠燈

若 clk=1

則行人狀態的紅燈與車向狀態的綠燈皆會閃

若 clk=0

行人狀態的紅燈與車向狀態的綠燈皆不運行

圖 7 第一階段 VHDL 程式碼

第二階段：按下 SW1，行人號誌狀態會從原先的閃紅燈變成紅燈；車向紅綠燈狀態會從原先的閃黃燈變成綠燈(行人號誌的狀態(亮紅燈)與車向紅綠燈的狀態(亮綠燈)接會持續 15 秒)

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  USE IEEE.STD_LOGIC_UNSIGNED.ALL;
4  ==entity pg_s is
5  ==port (CLK,SW,P:IN STD_LOGIC;--SW PIN 55 prestatement P is pause
6         Z:OUT STD_LOGIC;--Z PIN 33
7         N:OUT STD_LOGIC;--N next statement
8         R0,G0:IN STD_LOGIC;--light input
9         R,G:OUT STD_LOGIC;--light output
10        Y:OUT STD_LOGIC_VECTOR(6 downto 0)--ten,bit
11        );
12 end pg_s;
13
14 ==architecture a of pg_s is
15 SIGNAL QN : STD_LOGIC_VECTOR(3 DOWNT0 0);
16 SIGNAL op :STD_LOGIC_VECTOR(6 downto 0);
17 SIGNAL n1 :STD_LOGIC;
18 ==BEGIN
19 Z<=SW;
20 ==    PROCESS (CLK,P) --down
21 ==    BEGIN
22 n1<='0';
23 ==    if P='0'then
24 ==        R<=R0;
25 ==        G<=G0;
26 ==    elsif P='1' then
27 ==        R<='1';
28 ==        G<='1';
29 ==    IF CLK'event AND CLK='1' THEN
30 ==        QN <= QN-1;
31 ==    end if;
32 ==    if QN="0001" then
33 ==        n1<='1';
34 ==    END IF;
35 END IF;
36 end process;
37 N<=n1;
38 end a;

```

Clk:接給下數計數器

N:將此階段狀態傳給下一階段

R0:上一階段人行狀態之紅燈

G0:上一階段車向狀態之綠燈

R:行人狀態的紅燈

G:車向狀態的綠燈

圖 8 第二階段 VHDL 程式碼

當按下 p=1 時開始這個階段,否則繼續跑上一階段

下數計數器跑過一輪後,輸出 N=1 給下一階段,為了終止 clock 繼續跑影響之後的狀態,解決辦法是將 N 取 not 和 clock 取 and

第三階段：默數完 15 秒後，行人號誌的狀態會從紅燈變成閃綠燈；車向紅綠燈的狀態會從原先的綠燈轉成亮黃燈 1 秒，再變成紅燈。除此之外，七段顯示器也會從 15 秒開始倒數，直到歸零。

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_UNSIGNED.all;
4
5  entity final is
6  port (clk, flag, cgin, prin, sw: in std_logic;
7        cy, cr, pg, cg, pr, finish: out std_logic;
8        seven: OUT STD_LOGIC_VECTOR(6 downto 0);
9        Z: OUT STD_LOGIC);
10 end final;
11
12 architecture a of final is
13 signal qn: std_logic_vector(3 downto 0);
14 SIGNAL cred, cyel, pgre, sign, cg1, pr1 : STD_LOGIC;
15 SIGNAL op : STD_LOGIC_VECTOR(6 downto 0);
16 begin
17     process (clk, cyel, cred, pgre, flag, sign)
18     begin
19
20         if (flag='1') then
21             cg1<='0';
22             pr1<='0';
23             if clk'event AND clk='1' then
24                 qn <= qn-1;
25             end if;
26             if (qn="0000") then
27                 cyel <= '1';
28                 cred <= '0';
29                 pgre <= '0';
30             else
31                 cyel <= '0';
32                 cred <= '1';
33
34                 if (clk='1') then
35                     pgre <= '1';
36                 elsif (clk='0') then
37                     pgre<='0';
```

```

37         pgre<='0';
38     end if;
39 end if;
40 Z<=SW;
41 IF SW='1' THEN
42     case QN is
43     when "0000"=> op<="1111110";--0 bit
44     when "0001"=> op<="0110000"; --1
45     when "0010"=> op<="1101101"; --2
46     when "0011"=> op<="1111001";--3
47     when "0100"=> op<="0110011";--4
48     when "0101"=> op<="1011011";--5
49     when "0110"=> op<="1011111";--6
50     when "0111"=> op<="1110000";--7
51     when "1000"=> op<="1111111";--8
52     when "1001"=> op<="1111011";--9
53     when "1010"=> op<="1111110";--10
54     when "1011"=> op<="0110000";--11
55     when "1100"=> op<="1101101";--12
56     when "1101"=> op<="1111001";--13
57     when "1110"=> op<="0110011";--14
58     when "1111"=> op<="1011011";--15
59     end case;
60 ELSIF SW='0' THEN
61     case QN is
62     when "0000"=> op<="1111110";--0 ten
63     when "0001"=> op<="1111110";--1
64     when "0010"=> op<="1111110";--2
65     when "0011"=> op<="1111110";--3
66     when "0100"=> op<="1111110";--4
67     when "0101"=> op<="1111110";--5
68     when "0110"=> op<="1111110";--6
69     when "1001"=> op<="1111110";--9
70     when "1010"=> op<="0110000";--10
71     when "1011"=> op<="0110000";--11
72     when "1100"=> op<="0110000";--12
73     when "1101"=> op<="0110000";--13
74     when "1110"=> op<="0110000";--14
75     when "1111"=> op<="0110000";--15
76     end case;
77 END IF;
78 else
79     qn<="0000";
80     op<="0000000";
81     cyel<='0';
82     cred<='0';
83     pgre<='0';
84     cgl<=cgin;
85     prl<=prin;
86 end if;
87
88 end process;
89
90
91 cy<=cyel;
92 cr<=cred;
93 pg<=pgre;
94 cg<=cgl;
95 pr<=prl;
96 seven<=op;
97
98
99 end a;

```

接收上一個狀態的結束 flag(當上一個狀態結束後 flag=1)，車子綠燈，和行人紅燈。

設一個 clock，當 flag=1 時將上個狀態的燈號歸零，並利用 clock 開始做下數記數，再用交換顯示的方法，顯示兩位數在七段顯示器上。

同時，第一秒時輛車子黃燈，第二秒開始閃行人綠燈。

當 flag=0 時，維持上一個狀態的燈號，並將下數計數歸零，且不顯示七段顯示器。

圖 第三階段 VHDL 程式碼

四、預測結果：

























原始狀態	<div>    </div> <div>    </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>
按下後	<div>    </div> <div>    </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>
按下 過 15 秒後	<div>    </div> <div>    </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>
按下 過 16 秒後	<div>    </div> <div>    </div> <div> <div>1</div> <div>5</div> <div></div> <div></div> <div></div> <div></div> </div>

圖 6 預測結果示意表格