

3 Verilog & iVerilog

3.1 實驗目的

- 練習 Verilog 語法以及學習 Verilog 架構
- 使用 Icarus Verilog Simulator 完成設計驗證

3.2 實驗器材與元件

名稱	說明
 <p>Icarus Verilog</p>	<p>Icarus Verilog 是一套可進行 Verilog 編譯與模擬的免費軟體。本次實驗課,會使用 Icarus Verilog 中 iverilog、vvp來模擬16-bit 乘法器的執行 pattern的結果。</p> <p>同學請自行從下方網址其一來下載 Icarus Verilog。</p> <p>下載網址：</p> <p>Windows : http://bleyer.org/icarus/</p> <p>Mac : http://easonchang.logdown.com/posts/649863</p>

3.3 實驗內容

本次實驗課是要讓同學學習 Verilog 語法, 使用 Icarus Verilog 模擬驗證。本次實驗分為

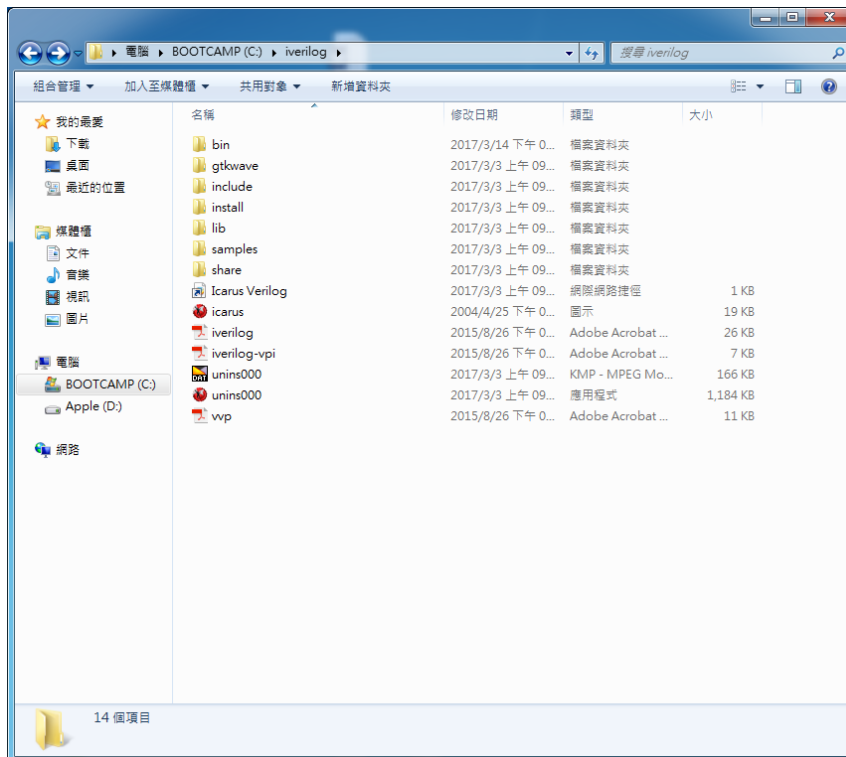
1. 開啟範例程式，先學會編譯 Verilog 的指令。
2. 修改範例程式，程式行為是顯示 28825252 的 Karnaugh Map 轉碼結果。了解 Verilog 架構以及語法。

實驗一

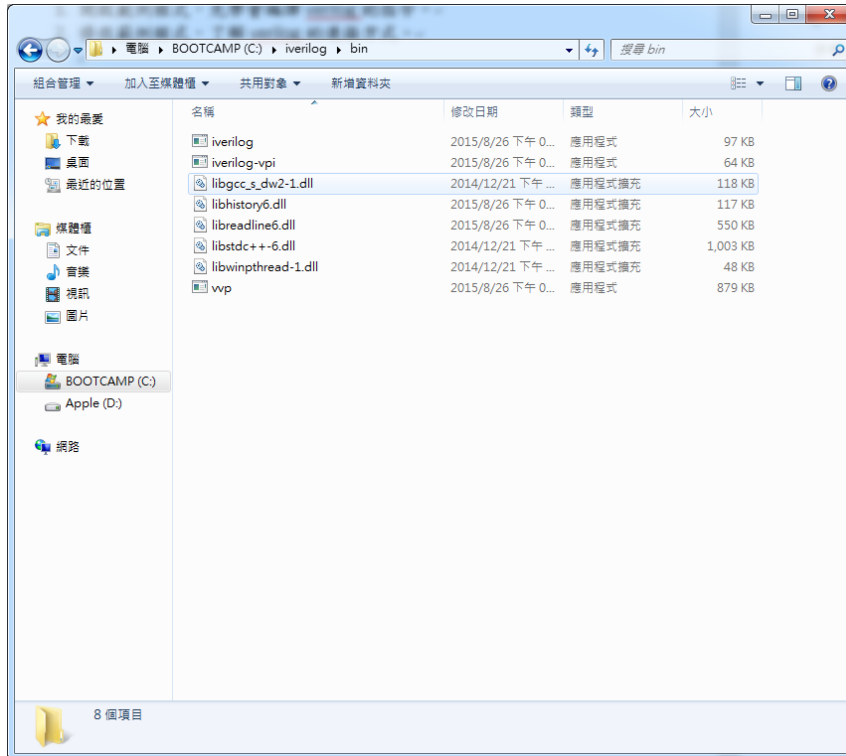
此實驗分為兩個部分，第一個部分的主要目的是希望學習如何利用 iverilog 模擬驗證 Verilog，並觀察結果。助教已經提供完整範例程式，並已上傳到 E-course 上，所以第一個實驗只要下載範例程式即可。

步驟一：安裝 Icarus Verilog（軟體已安裝進電腦中可忽略）。

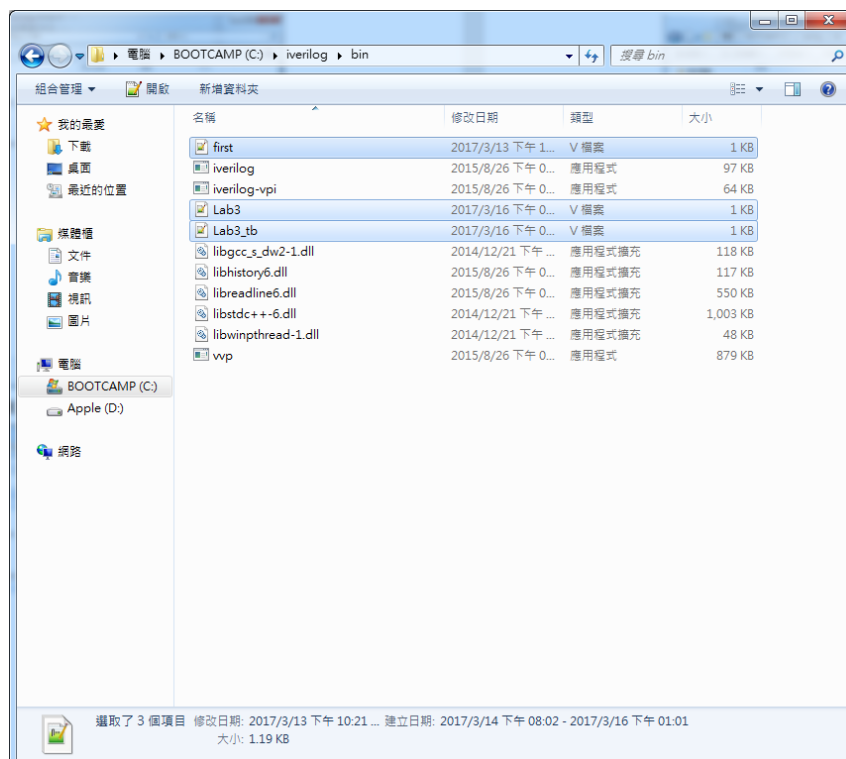
步驟二：找到 iverilog 安裝目錄。



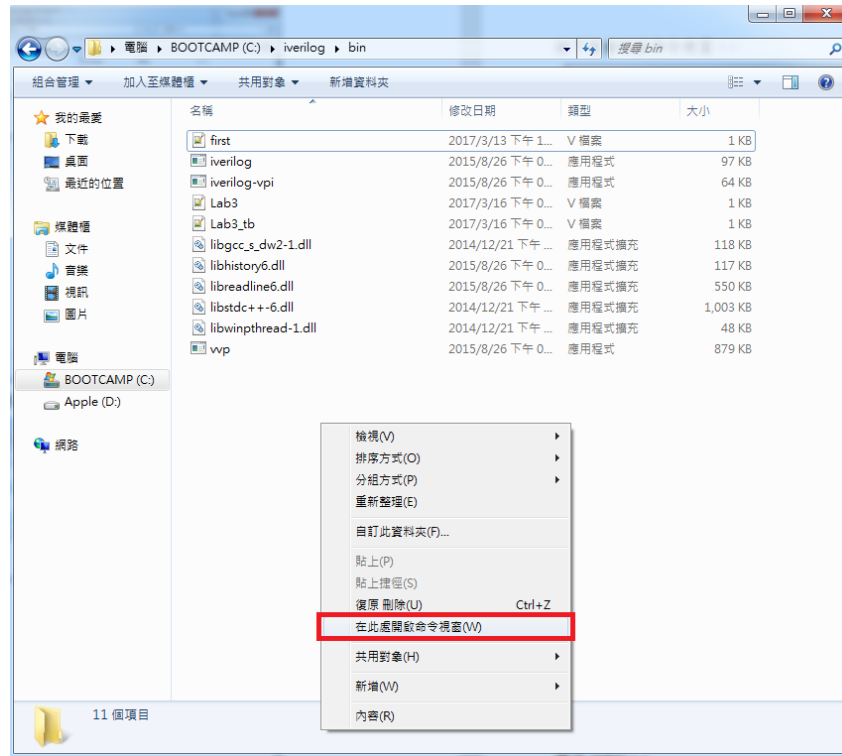
進入 bin 資料夾中。



步驟三：將範例程式中的三個檔案複製到 bin 資料夾內。



步驟四：在資料夾中，同時按下 shift+右鍵，即可選擇”在此處開起命令視窗”。

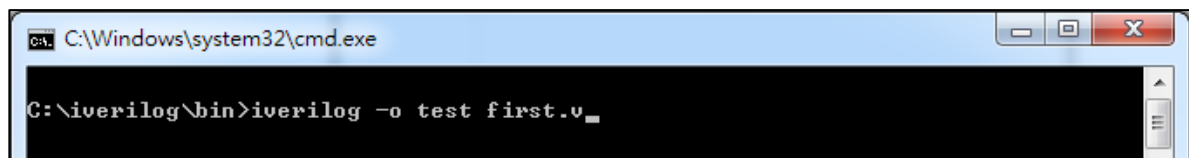


按下後會跳出命令視窗。



步驟五：編譯，iverilog -o 輸出檔案名稱 輸入的原始碼

鍵入指令，iverilog -o test first.v

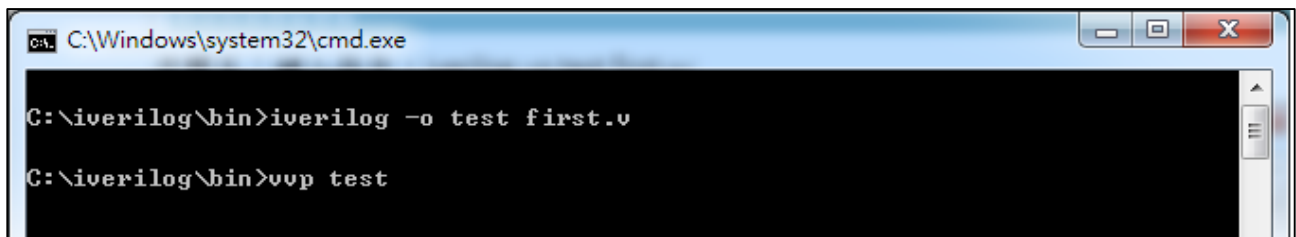


按下 enter。



步驟六：模擬執行

vvp 輸出檔案名稱
鍵入指令，vvp test

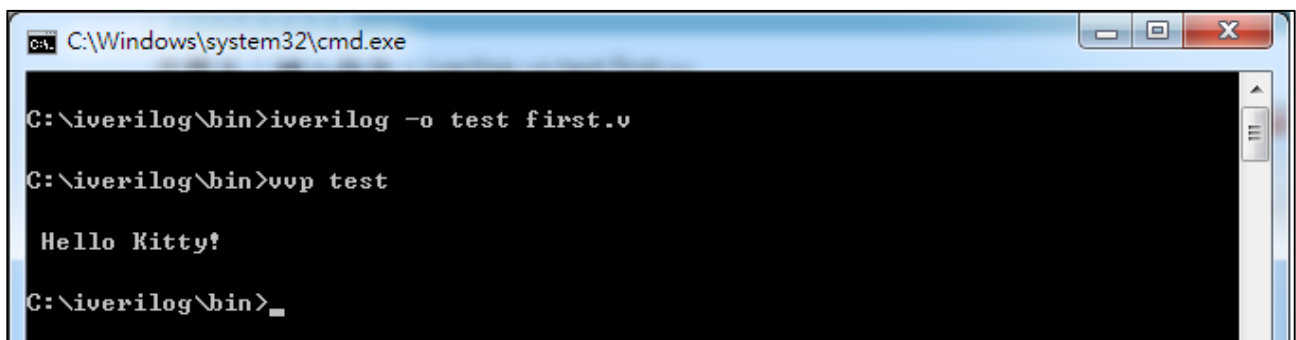


```
C:\Windows\system32\cmd.exe

C:\iverilog\bin>iverilog -o test first.v

C:\iverilog\bin>vvp test
```

按下 enter。



```
C:\Windows\system32\cmd.exe

C:\iverilog\bin>iverilog -o test first.v

C:\iverilog\bin>vvp test

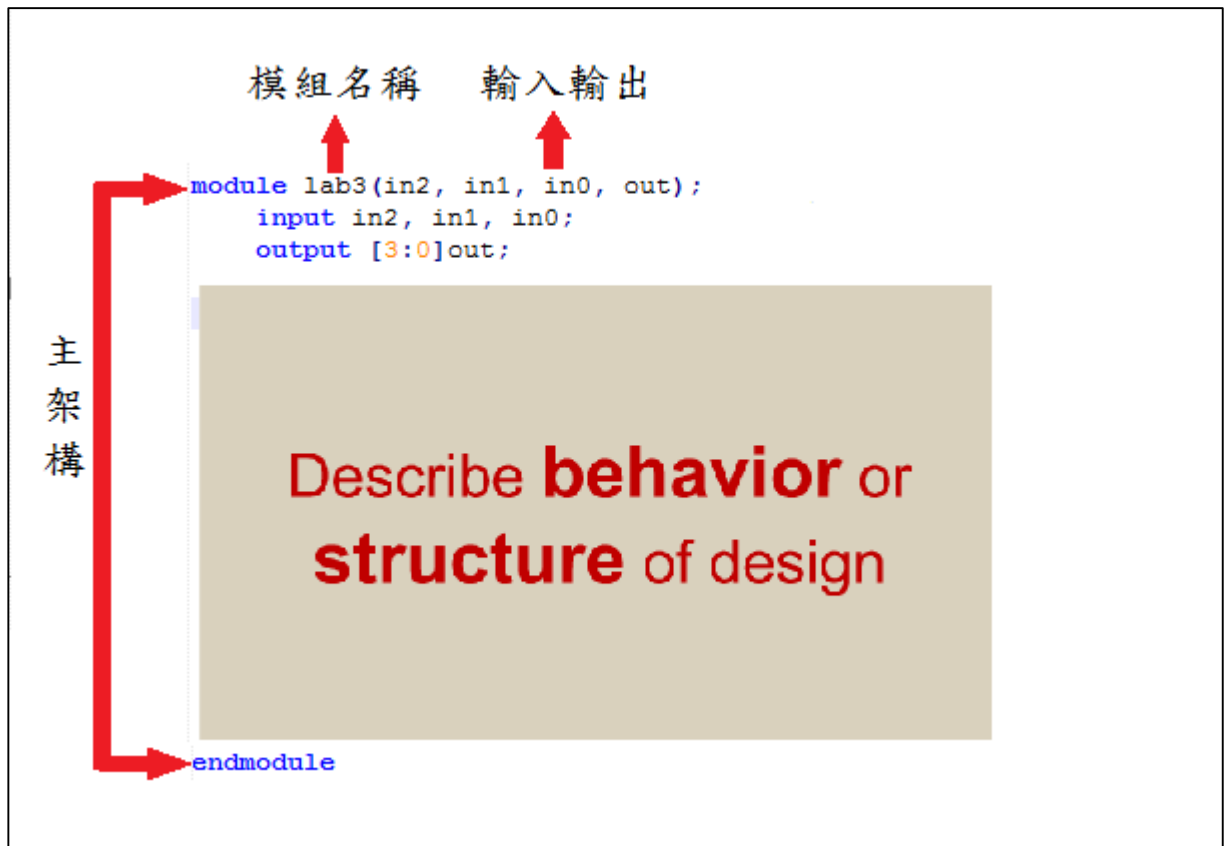
Hello Kitty!

C:\iverilog\bin>_
```

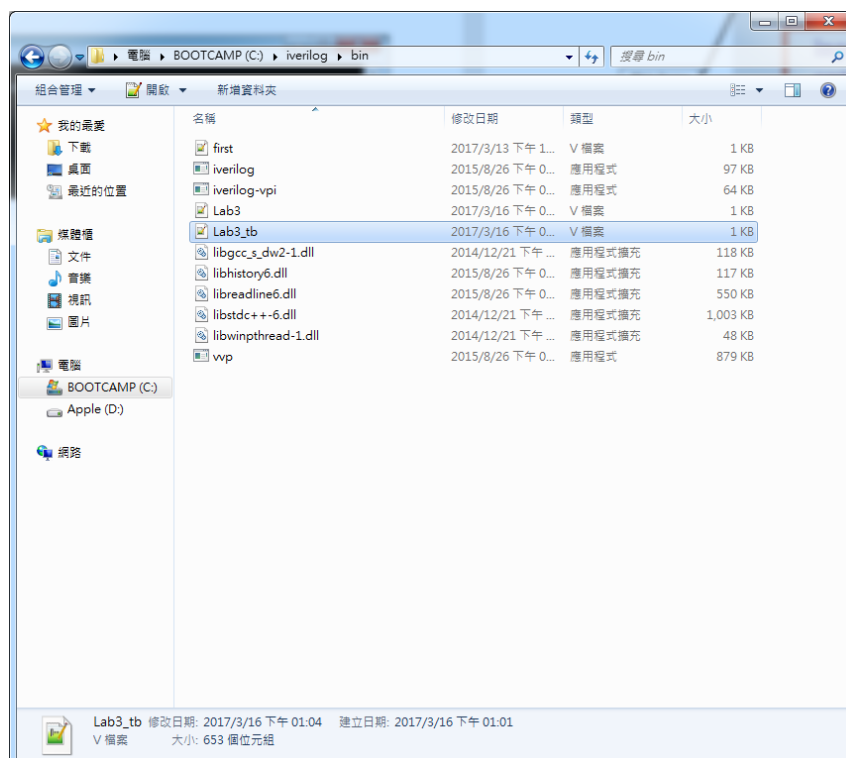
實驗二

第二個部分的主要目的是希望學習 Verilog 架構並完成 SOP 轉碼並顯示結果。

Verilog 的架構：



步驟一：開啟 iverilog\bin 中的 Lab3_tb.v。Lab3_tb.v 的功能為驗證 Lab3.v 的 SOP 轉碼功能，程式行為是定時將計數器加一後傳給 Lab3.v，並接收回傳結果後顯示。



開啟後可看到如下圖。testbench 為模組名稱；reg 表示宣告為 reg 變數，做為儲存空間；reg[3:0] count 意思為宣告 count 為 4-bit-reg；initial begin 到 end 的程式，為初始化區塊。

```

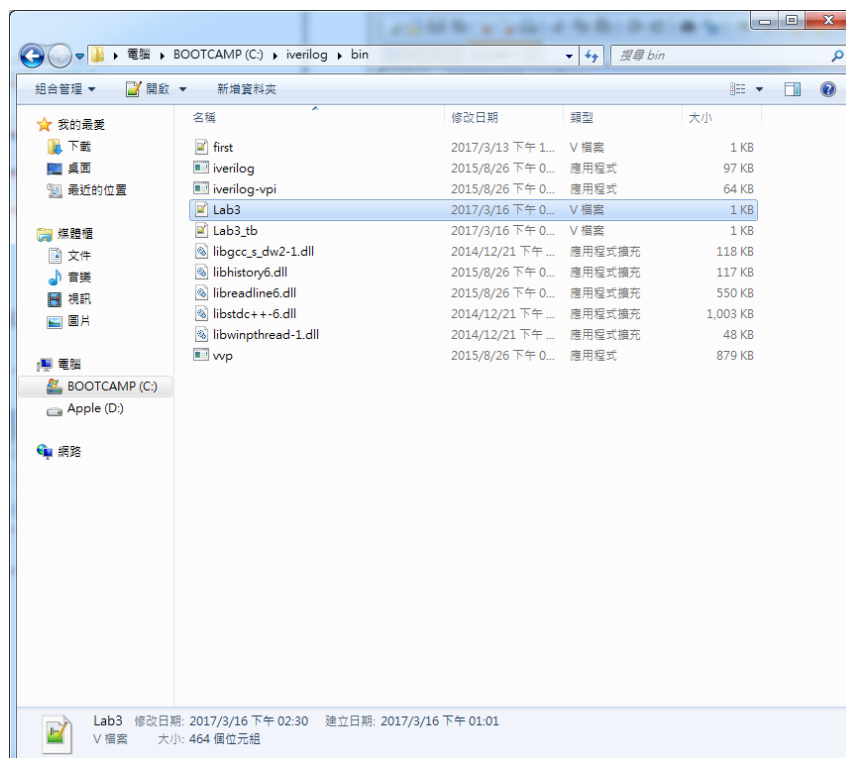
1 timescale 1ns/1ns //告訴iverilog等模擬軟體，以前者(1ns)為單位，以後者(10ps)的時間，查看一次電路的行為。
2 include "Lab3.v"
3
4 module testbench; // 模組名稱
5     reg CLK; // 暫存器宣告
6     reg [3:0] count;
7     wire [3:0] out;
8
9     lab3 l3(.in2(count[2]), .in1(count[1]), .in0(count[0]), .out);
10
11     always #10 CLK=~CLK; //每10個單位時間，CLK的值反相一次
12
13     always @ (posedge CLK)
14     begin
15         count=count+1;
16         if(count[3]==1'b1) count<=4'b0;
17     end
18
19
20     always @ (posedge CLK)
21     begin
22         #5 $display ("count = %d, out = %d",count, out);
23     end
24
25     initial begin //僅執行一次
26         CLK=0;
27         count=4'b111;
28         #160 $finish;
29     end
30 endmodule
  
```

計數器程式

每5個單位時間
顯示計數器的值

初始化區塊

步驟二：開啟 iverilog\bin 中的 Lab3.v。Lab3.v 的程式行為是運算 SOP 轉碼後，將結果回傳給 Lab3_tb.v。



開啟後可看到如下圖。

```

1 `timescale 1ns/1ns
2
3 module lab3(in2, in1, in0, out);
4     input in2, in1, in0; //宣告各個input 與output
5     reg out3, out2, out1, out0;
6     output [3:0]out;
7
8     reg [3:0]out;
9
10    always @ (*)
11    begin
12        /* 修改以下部分的程式 */
13        out0=(in0&in2);
14        out1=(in0&in1)|(in0&in2)|(!in0&in1&in2);
15        out2=(!in0&in2);
16        out3=(in0&!in1&in2)|(!in0&in1&in2);
17        /* 修改以上部分的程式 */
18        out={out3, out2, out1, out0};
19    end
20 endmodule
21

```

以下說明邏輯運算子:

&	AND
	OR
!	NOT

以下說明：28825252 的 Karnaugh Map 轉碼過程。

count	out3	out2	out1	out0
000	0	0	1	0
001	1	0	0	0
010	1	0	0	0
011	0	0	1	0
100	0	1	0	1
101	0	0	1	0
110	0	1	0	1
111	0	0	1	0

out0 : (!in0&in2)

in1 \ in0 \ in2		0	1
0	0	0	1
0	1	0	0
1	1	0	0
1	0	0	1

out1 : (in0&in1)+(in0&in2)+(!in0&!in1&in2)

in1 \ in0 \ in2		0	1
0	0	1	0
0	1	0	1
1	1	1	1
1	0	0	0

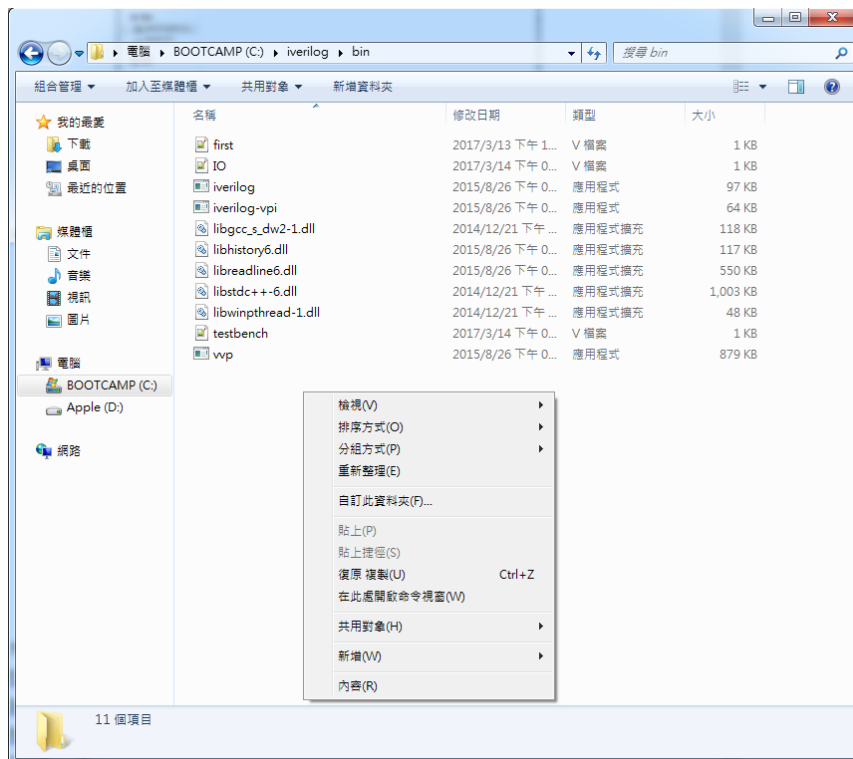
out2 : (!in0&in2)

in1 \ in0 \ in2		0	1
0	0	0	1
0	1	0	0
1	1	0	0
1	0	0	1

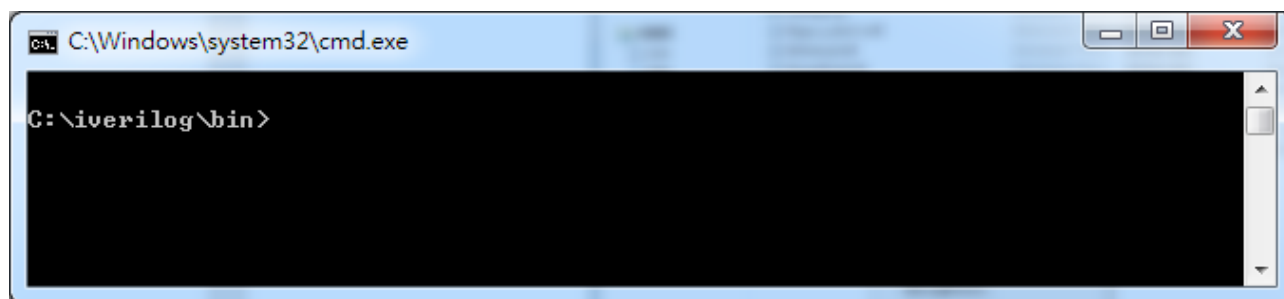
out3 : (in0&!in1&!in2)+(!in0&in1&!in2)

<div style="display: inline-block; width: 100px; height: 100px; position: relative;"> <div style="position: absolute; top: 0; right: 0;">in2</div> <div style="position: absolute; bottom: 0; left: 0;">in1</div> <div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);">in0</div> </div>		0	1
		0	1
0	0	0	0
0	1	1	0
1	1	0	0
1	0	1	0

步驟三：依照實驗一的步驟執行一次。在資料夾中，同時按下 shift+右鍵，即可選擇”在此處開啟命令視窗”。

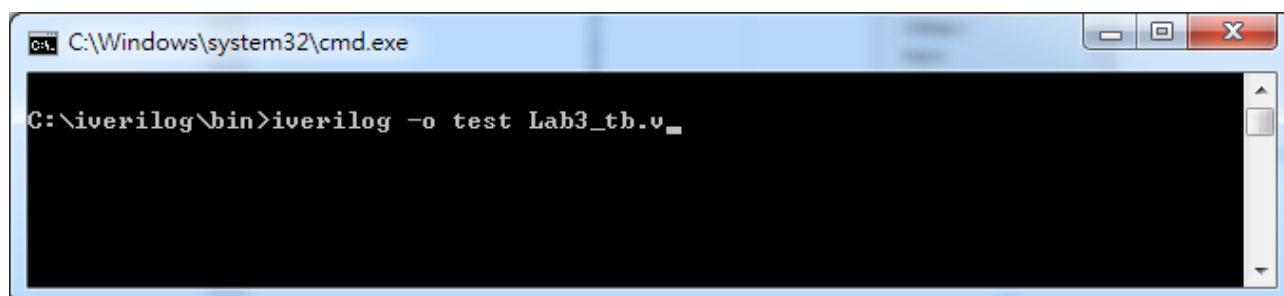


按下後會跳出命令視窗



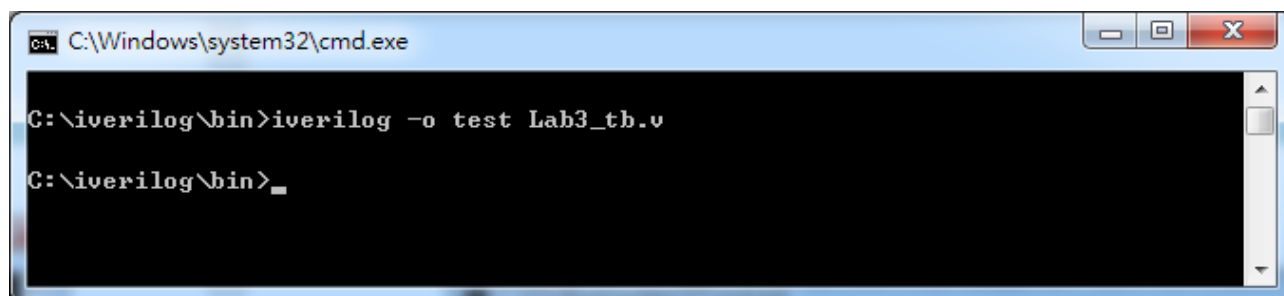
```
C:\Windows\system32\cmd.exe  
C:\iverilog\bin>
```

步驟四：鍵入指令，iverilog -o test Lab3_tb.v



```
C:\Windows\system32\cmd.exe  
C:\iverilog\bin>iverilog -o test Lab3_tb.v_
```

按下 enter。



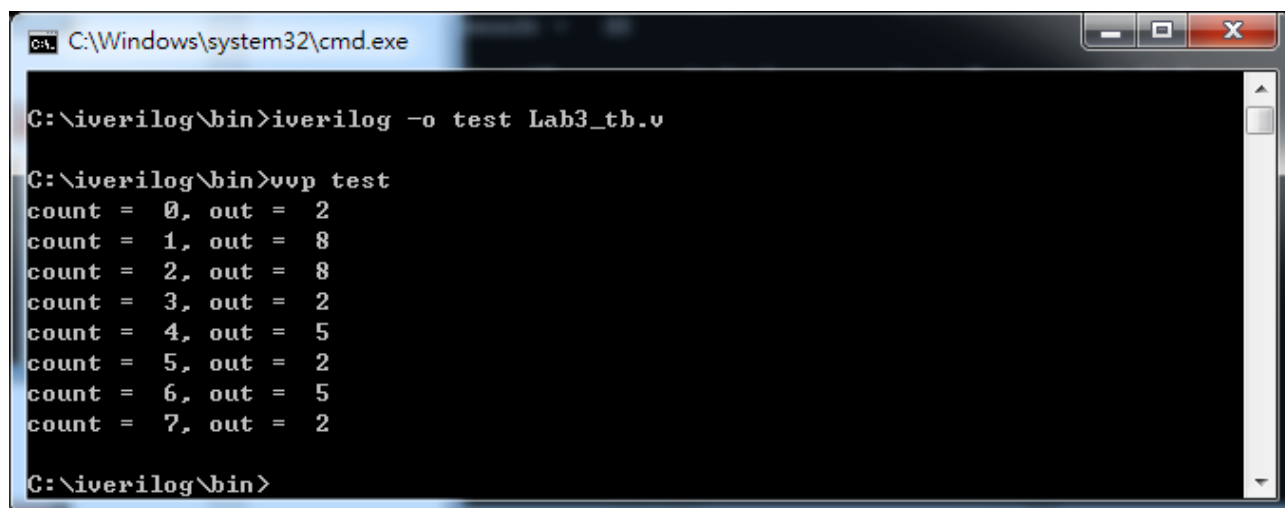
```
C:\Windows\system32\cmd.exe  
C:\iverilog\bin>iverilog -o test Lab3_tb.v  
C:\iverilog\bin>_
```

步驟五：鍵入指令，vvp test



```
C:\Windows\system32\cmd.exe  
C:\iverilog\bin>iverilog -o test Lab3_tb.v  
C:\iverilog\bin>vvp test_
```

按下 enter。



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The prompt is at "C:\iverilog\bin>". The user has entered the command "iverilog -o test Lab3_tb.v", which has been executed. The next prompt is "C:\iverilog\bin>vvp test". The output of the simulation is displayed as follows:

```
count = 0, out = 2  
count = 1, out = 8  
count = 2, out = 8  
count = 3, out = 2  
count = 4, out = 5  
count = 5, out = 2  
count = 6, out = 5  
count = 7, out = 2
```

The prompt is now "C:\iverilog\bin>".

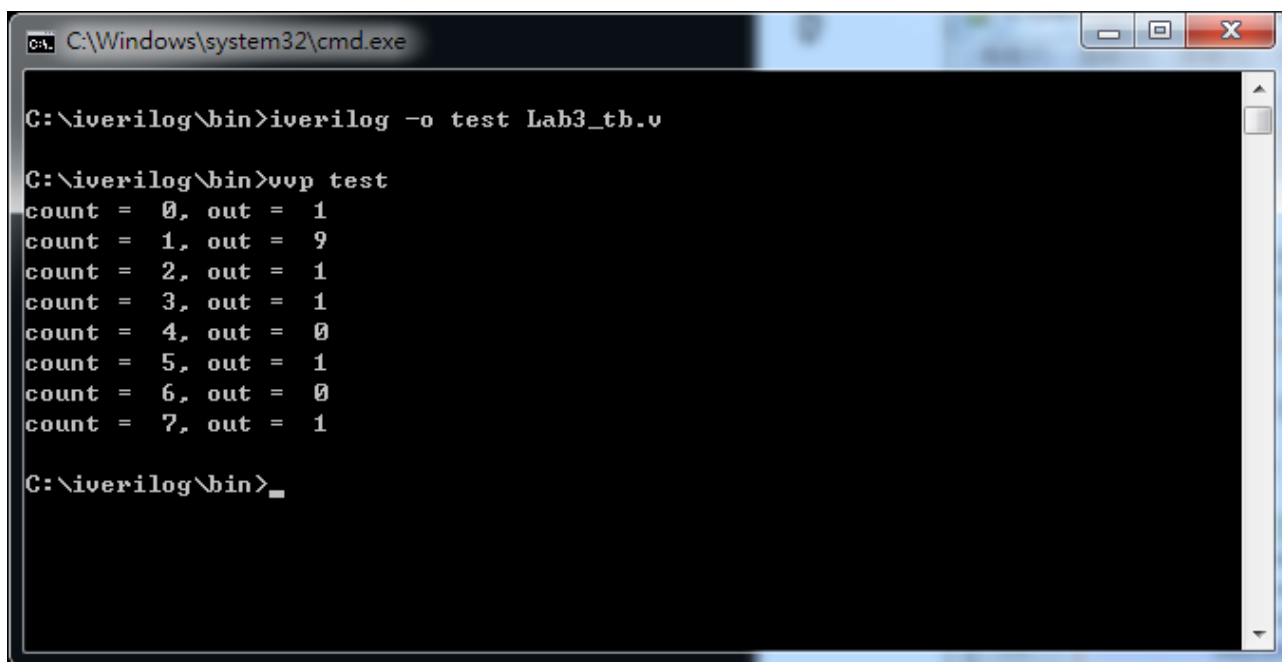
3.4 練習題

練習修改實驗二的 Lab3.v 檔。

修改的範圍為 Lab3.v 檔的 13~16 行程式碼，使執行結果為顯示組員其中一人的生日。

```
12      /* 修改以下部分的程式 */
13      out0=(!in0&in2);
14      out1=(in0&in1) | (in0&in2) | (!in0&!in1&!in2);
15      out2=(!in0&in2);
16      out3=(in0&!in1&!in2) | (!in0&in1&!in2);
17      /* 修改以上部分的程式 */
```

以生日為 1911/01/01 為例，修改後之命令視窗的圖示如下：



```
C:\Windows\system32\cmd.exe

C:\iverilog\bin>iverilog -o test Lab3_tb.v

C:\iverilog\bin>vvp test
count = 0, out = 1
count = 1, out = 9
count = 2, out = 1
count = 3, out = 1
count = 4, out = 0
count = 5, out = 1
count = 6, out = 0
count = 7, out = 1

C:\iverilog\bin>
```