

Laboratory 7

虛擬裝置 與 VeriInstrument



Department of Electrical Engineering
National Cheng Kung University

實驗目的

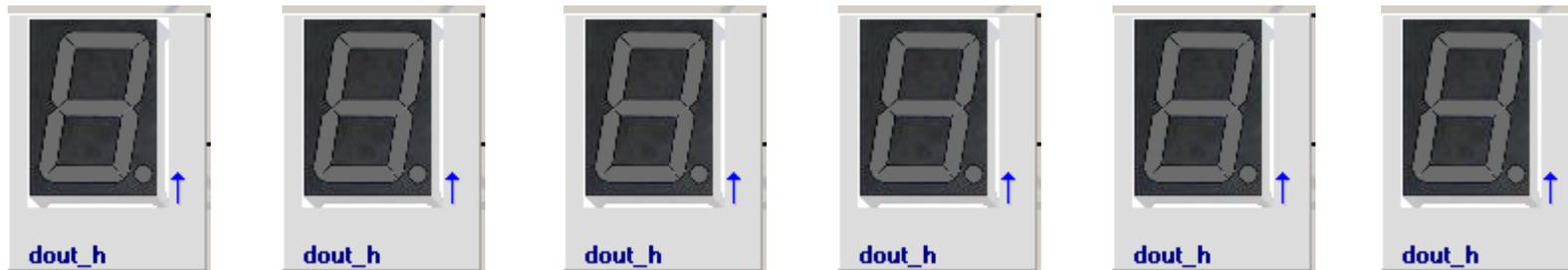
- 了解虛擬裝置的概念
- 學習使用 VerilInstrument 來設計與實作虛擬裝置

使用器材

- 桌上型電腦
- Xilinx FPGA 板

虛擬裝置

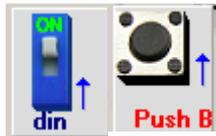
- 許多廠商提供虛擬裝置的功能讓使用者可以在無實體元件的情況下模擬特定元件的行為
 - 例如: LED, 七段顯示器, Switch, Button, LCD, 8x8 LED Matrix 等等
- 使用虛擬裝置可以任意新增元件，同時也省下了許多接線的時間



VeriInstrument (1/4)

- VeriInstrument 是SMIMS提供的虛擬裝置設計軟體套件
- 主要會用到的有兩種虛擬裝置

- 輸入: Switch, Button



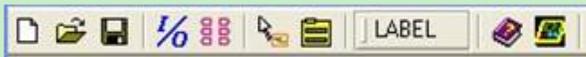
- 輸出: LED, 七段顯示器, LCD等等



VeriInstrument (2/4)

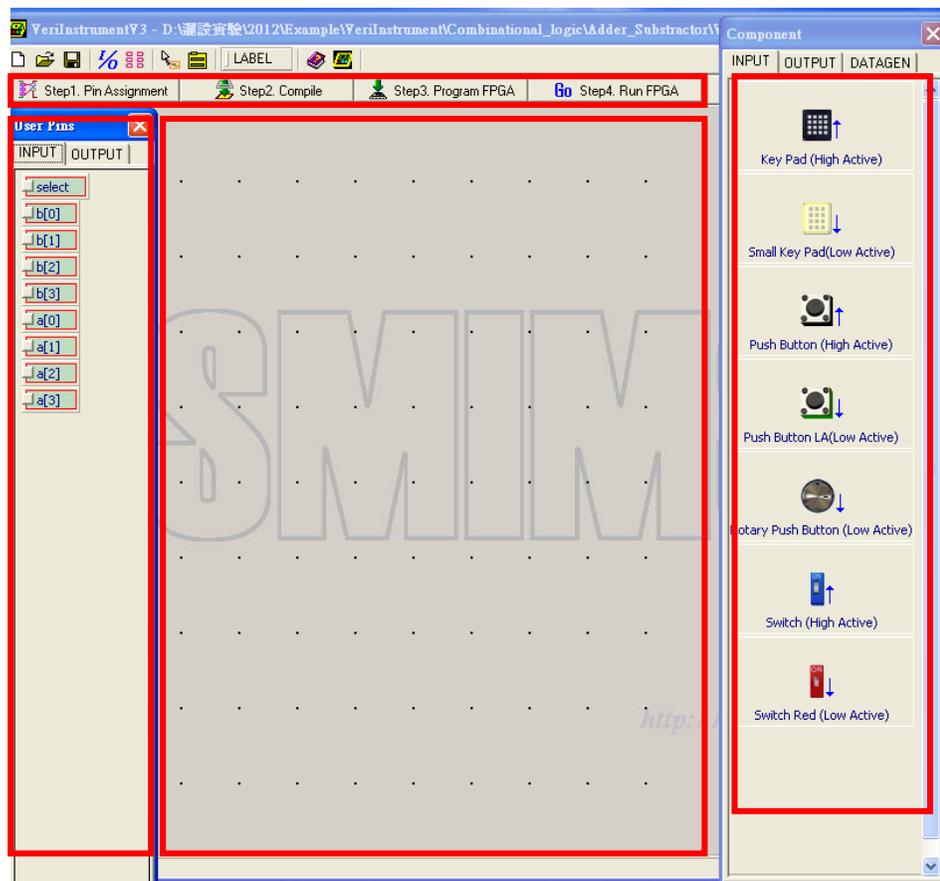
➤ 虛擬儀器工具列

Toolbar functions



1.  Create new project
2.  Open existed project
3.  Save now project
4.  Open or Close **Component** window
5.  Open or Close **User Pins** window
6.  Enable/Disable component information hint
7.  Show project's information
8.  Create a text label on working area
9.  Open Help document
10.  About VeriInstrumentV3

VeriInstrument (3/4)



➤ 操作面板分成四個區塊

- 虛擬板面(中間): 在此安裝虛擬裝置
- 虛擬裝置(右側): 可供使用者新增的虛擬裝置，如: 指撥開關、LED燈
- **User Pin**(左側): 將使用者所設計的邏輯電路接上虛擬裝置
- 操作步驟列(上方): 提示使用者燒入FPGA的四個步驟

VeriInstrument (4/4)

➤ 注意事項

- Veriinstrument軟體簡化了操作，但實際上仍然是呼叫Xilinx ISE來編譯與合成。
- Parse failed可能代表code編譯不過，但在此處不會顯示錯誤訊息。可參照上次實驗，自行使用Xilinx ISE編譯檢查錯誤訊息。

實作題

本次實作分為兩個基本實作與一個挑戰實作

實作題(一): 指撥開關與LED之使用

- 利用虛擬裝置來實作與LED 與指撥開關相關的邏輯電路
- 使用虛擬裝置實作邏輯電路的主要四個步驟為
 - Step1. 設計邏輯電路
 - Step2. 設計虛擬元件配置與連接
 - Step3. 編譯並且產生bit檔
 - Step4. 將bit檔燒入FPGA中執行
- 這個實作將會使用到下面個工具
 - VeriInstrumentV3: 提供虛擬裝置的配置介面與FPGA開發版的燒入執行功能

實作題(一) Step 1 (1/2)

Step1. 設計邏輯電路

1. 請在 “c:\logiclab\<自己的學號>” 的路徑下新增一資料夾 lab7_1
2. 開啟 ISE ，並開啟一新 project
3. 在 working directory 的地方選擇 “c:\logiclab\<自己的學號>\lab7_1”
4. 將此 project 命名為 showLED
5. 其他設定如上次實驗

實作題(一) Step 1 (2/2)

6. 在 File → New 下，選擇 Verilog HDL File，開啟一新 verilog 檔案。
7. 將以下內容鍵入此 Verilog 文件中，並儲存之(可隨意命名)

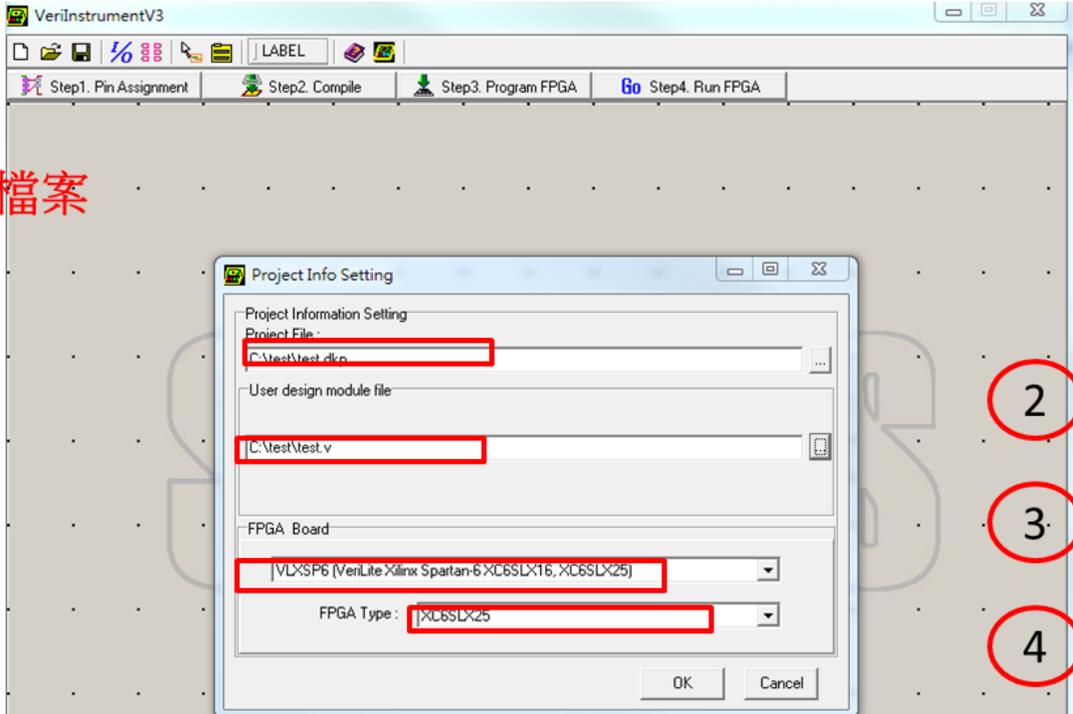
注意: 使用該平台的虛擬裝置時，所有模組都必須要配有clk訊號線，否則之後的編譯會失敗

```
module showLED (led0,led1,led2,led3, in0, in1,clk);  
  
    input    clk,in0,in1;  
    output   led0,led1,led2,led3;  
  
    reg led0,led1,led2,led3;  
  
    always@( in0 or in1 )  
    begin  
        case( {in1,in0} )  
            2'b00:    {led3,led2,led1,led0}=4'b0001;  
            2'b01:    {led3,led2,led1,led0}=4'b0010;  
            2'b10:    {led3,led2,led1,led0}=4'b0100;  
            2'b11:    {led3,led2,led1,led0}=4'b1000;  
        endcase  
    end  
endmodule
```

實作題(一) Step 2 (1/4)

Step2. 設計虛擬元件配置與連接

1. 選取指定邏輯電路的Top Module 與 FPGA型號



The screenshot shows the VeriInstrumentV3 software interface. A 'Project Info Setting' dialog box is open, with several fields highlighted by red boxes and numbered annotations:

- 1** (circled in red): A red circle next to the 'New File' icon in the toolbar, with the text '開新檔案' (Open New File) written in red below it.
- 2** (circled in red): A red circle next to the 'Project File' text box, with the text '輸入專案檔名稱' (Enter project file name) written in red to its right.
- 3** (circled in red): A red circle next to the 'User design module file' text box, with the text '選擇.v檔' (Select .v file) written in red to its right.
- 4** (circled in red): A red circle next to the 'FPGA Board' dropdown menu, with the text '選擇板子型號與FPGA型號' (Select board type and FPGA type) written in red to its right.

The 'Project Info Setting' dialog box contains the following fields:

- Project File: C:\test\test.dcp
- User design module file: C:\test\test.v
- FPGA Board: VLXSP6 (VeriLite Xilinx Spartan-6 XC6SLX16, XC6SLX25)
- FPGA Type: XC6SLX25

實作題(一) Step 2 (2/4)

2. 新增需要使用到的虛擬裝置，選擇High Active的元件

The screenshot displays the VeriInstrument V3 software interface. On the left, a 'Component' panel lists various I/O components, with 'Seven Segment (High Active)' and 'Beep (High Active)' highlighted. The main workspace shows a grid with several components placed: four switches and two LEDs. A 'User Pins' panel on the right lists pins like 'select', 'b[0]', 'b[1]', 'b[2]', 'b[3]', 'a[0]', 'a[1]', 'a[2]', and 'a[3]'. Three red text boxes with arrows point to specific actions: '1. 點開 I/O 與 pins 工具列' points to the I/O and pins toolbars; '2. 拖曳 I/O 元件至虛擬面板' points to the component list; '3. 配置虛擬面板上的元件' points to the placed components.

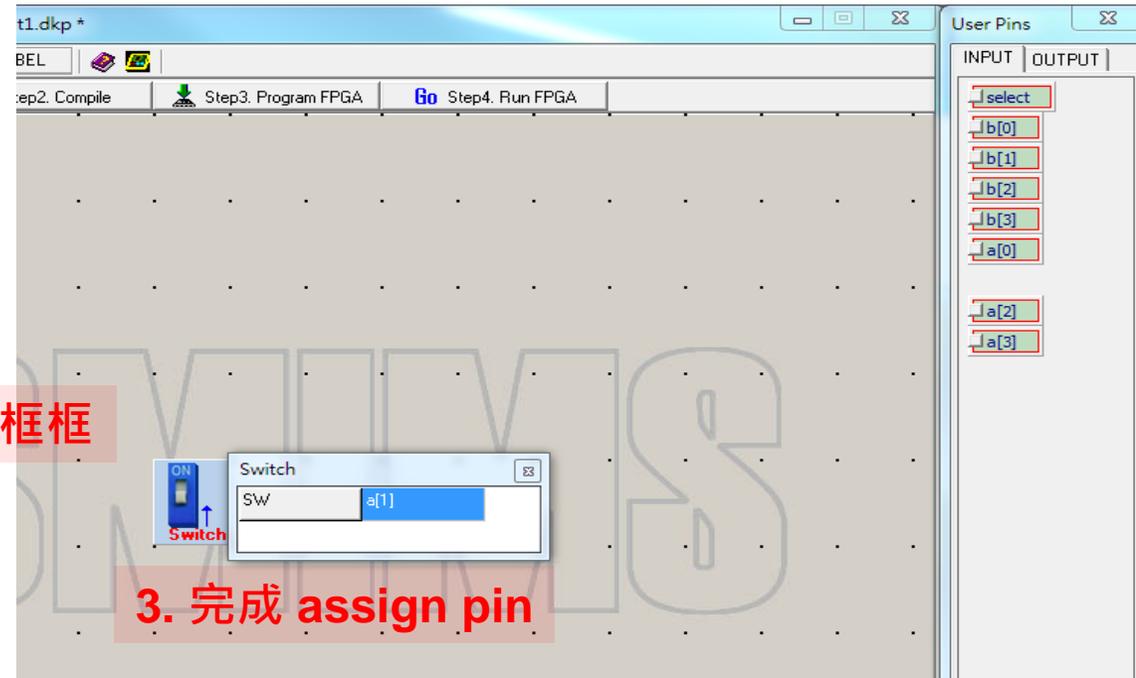
1. 點開 I/O 與 pins 工具列

2. 拖曳 I/O 元件至虛擬面板

3. 配置虛擬面板上的元件

實作題(一) Step 2 (3/4)

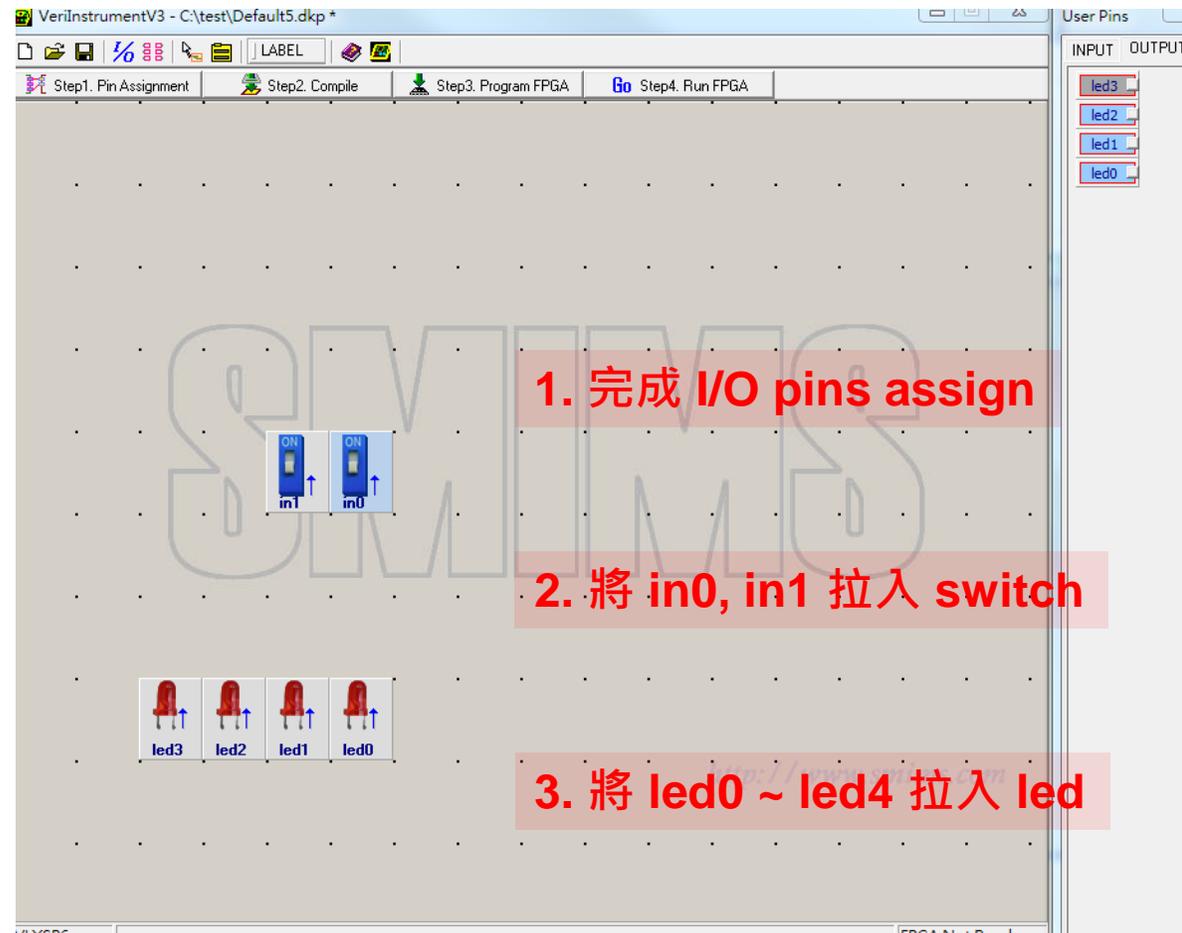
3. 連接虛擬裝置與邏輯電路



也可以直接將PIN腳拖曳到元件上來完成連接的工作

實作題(一) Step 2 (4/4)

4. 依照前面的步驟完成所有虛擬元件的配置與連接



實作題(一) Step 3

Step3. 編譯並且產生bit檔

The screenshot shows the 'CompileWindow' interface. It features a 'User design modules' section with a 'Top module' field containing 'C:\test\test.v' and a 'Sub-modules' table. A 'Compile' button with a download icon is highlighted. The status bar at the bottom shows 'PAR done!' and 'Started: "Generate Programming File". Process "Generate Programming File" completed successfully. Compile stage (4) pass.'

1. 按下上面 "Step2. Compile"

2. 選擇 design module

3. 按下 Compile

4. 完成 Compile 動作

實作題(一) Step 4 (1/2)

Step4. 將bit檔燒入FPGA中執行

1. 按下”Step3. Program FPGA” 將bit檔燒入FPGA中準備



實作題(一) Step 4 (2/2)

2. 按下”Step4. Run FPGA” 就會開始運作

燒入的電路為解碼器

注意: 進行下一個實作前請先再次按下

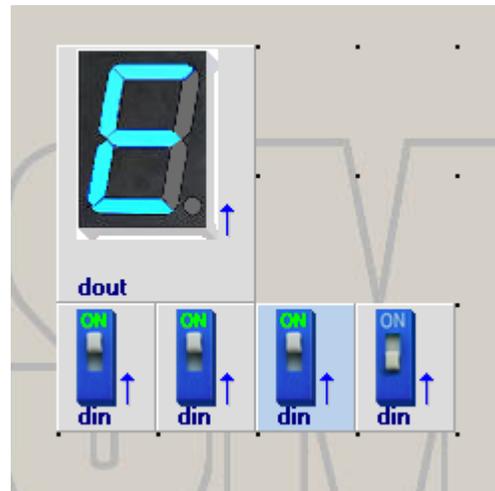
該按鈕來結束該次的運作，否則無法燒

入新的bit檔



實作題(二) 七段顯示器

- 利用虛擬裝置來實作與驗證七段顯示器的解碼電路



實作題(二) Step 1 (1/2)

Step1. 設計邏輯電路

1. 請在 “c:\logiclab\<自己的學號>” 的路徑下新增一資料夾 lab7_2
2. 開啟 ISE ，並開啟一新 project
3. 在 working directory 的地方選擇 “c:\logiclab\<自己的學號>\lab7_2”
4. 將此 project 命名為 sevenSeg
5. 其他設定如上次實驗

實作題(二) Step 1 (2/2)

6. 在 File → New 下，選擇 Verilog HDL File，開啟一新 verilog 檔案。
7. 將以下內容鍵入此 Verilog 文件中，並儲存之(可隨意命名)

注意: 使用該平台的虛擬裝置時，所有模組都必須要配有clk訊號線，否則之後的編譯會失敗

```
module SevenSeg ( dout, din,clk);
input   [3:0]   din;
input   clk;
output  [6:0]   dout;
reg     [6:0]   dout;

always@(din)
begin
    case(din)
        4'b0000 : dout = 7'b0111111 ;
        4'b0001 : dout = 7'b0000110 ;
        4'b0010 : dout = 7'b1011011 ;
        4'b0011 : dout = 7'b1001111 ;
        4'b0100 : dout = 7'b1100110 ;
        4'b0101 : dout = 7'b1101101 ;
        4'b0110 : dout = 7'b1111101 ;
        4'b0111 : dout = 7'b0000111 ;
        4'b1000 : dout = 7'b1111111 ;
        4'b1001 : dout = 7'b1101111 ;
        4'b1010 : dout = 7'b1110111 ;
        4'b1011 : dout = 7'b1111100 ;
        4'b1100 : dout = 7'b0111001 ;
        4'b1101 : dout = 7'b1011110 ;
        4'b1110 : dout = 7'b1111001 ;
        4'b1111 : dout = 7'b1110001 ;
        default : dout = 7'b0000000 ;
    endcase
end
endmodule
```

實作題(二) Step 2 (1/4)

Step2. 設計虛擬元件配置與連接

1. 選取指定邏輯電路的Top Module 與 FPGA型號

1 開新檔案

2 輸入專案檔名稱

3 選擇.v檔

4 選擇板子型號與FPGA型號

The screenshot shows the VeriInstrumentV3 interface with the Project Info Setting dialog box open. The dialog box contains the following fields:

- Project File: C:\test\test.dcp
- User design module file: C:\test\test.v
- FPGA Board: VLXSP6 (VeriLite Xilinx Spartan-6 XC6SLX16, XC6SLX25)
- FPGA Type: XC6SLX25

實作題(二) Step 2 (2/4)

2. 新增需要使用到的虛擬裝置，選擇High Active的元件

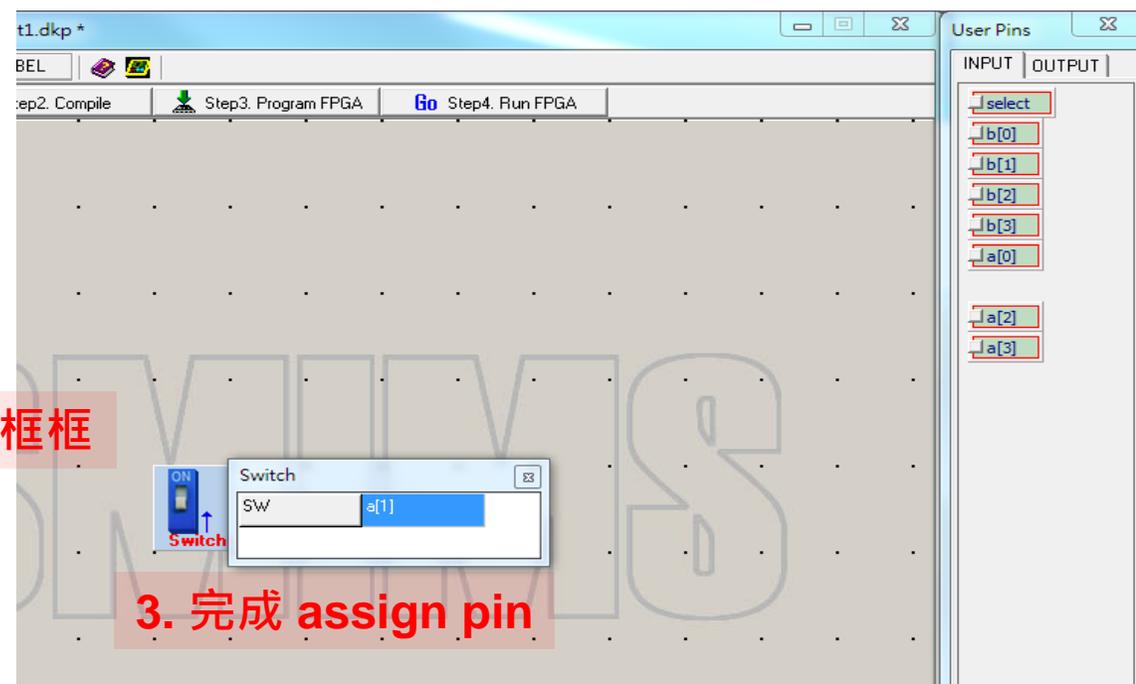
The screenshot displays the VeriInstrument V3 software interface. The main window shows a grid for configuring a virtual device. On the left, a 'Component' panel lists various I/O components, including 'Seven Segment (High Active)', 'Seven Segment LA(Low Active)', 'Beep (High Active)', 'Gray Image Player', 'LED (High Active)', 'Green LED (High Active)', 'Yellow LED (High Active)', and 'Matrix LED 8x8 (High Active)'. A red box highlights the 'LED (High Active)' component. In the center, a red box contains the text '1. 點開 I/O 與 pins 工具列', pointing to the toolbar. Below the grid, a red box contains the text '3. 配置虛擬面板上的元件', pointing to a configuration area where four 'Switch' components and two 'LED' components are placed. On the right, a 'User Pins' panel shows a list of pins: 'select', 'b[0]', 'b[1]', 'b[2]', 'b[3]', 'a[0]', 'a[1]', 'a[2]', and 'a[3]'. The status bar at the bottom indicates 'V LXSP6' and 'FPGA:Not Ready'.

2. 拖曳 I/O 元件至虛擬面板

3. 配置虛擬面板上的元件

實作題(二) Step 2 (3/4)

3. 連接虛擬裝置與邏輯電路



也可以直接將PIN腳拖曳到元件上來完成連接的工作

實作題(二) Step 2 (4/4)

4. 依照前面的步驟完成所有虛擬元件的配置與連接

The screenshot shows the VeriInstrumentV3 software interface. The main workspace displays a grid with a 'Seven Segment' component and a 'User Pins' panel on the right. The 'User Pins' panel has an 'INPUT' tab and an 'OUTPUT' tab. The 'OUTPUT' tab shows a list of pins: dout[0], dout[1], dout[2], dout[3], dout[4], dout[5], and dout[6]. A red box highlights a 'Seven Segment' component in the workspace, and a red box highlights a 'Seven Segment' dialog box. The dialog box has a table with the following data:

SEG	Pin
SEG[0]	dout[0]
SEG[1]	dout[1]
SEG[2]	dout[2]
SEG[3]	dout[3]
SEG[4]	dout[4]
SEG[5]	dout[5]
SEG[6]	dout[6]
SEG[7]	

1. 完成 I/O pins assign

2. 將dout[0]~dout[6]分別拉到SEG[0]~SEG[6]

<http://www.smitms.com>

實作題(二) Step 3

Step3. 編譯並且產生bit檔

1. 按下上面
“Step2. Compile”

2. 選擇 design module

3. 按下 Compile

4. 完成Compile動作

CompileWindow

User design modules
Top module
C:\test\test.v

Sub-modules

File name	Type
-----------	------

Add Files

Compile

Peak Memory Usage: 251 MB

Placer: Placement generated during map.
Routing: Completed - No errors found.
Timing: Completed - No errors found.

Number of error messages: 0
Number of warning messages: 0
Number of info messages: 0

Writing design to filetop.ncd

PAR done!

Started : "Generate Programming File".
Process "Generate Programming File" completed successfully
Compile stage (4) pass.

INPUT OUTPUT

dout[0]	
dout[1]	
dout[2]	
dout[3]	
dout[4]	
dout[5]	
dout[6]	

Open Directory

Close

實作題(二) Step 4 (1/2)

Step4. 將bit檔燒入FPGA中執行

1. 按下”Step3. Program FPGA” 將bit檔燒入FPGA中準備



實作題(二) Step 4 (2/2)

2. 按下”Step4. Run FPGA” 就會開始運作

燒入的電路為七段顯示器的解碼器

注意: 進行下一個實作前請先再次按下

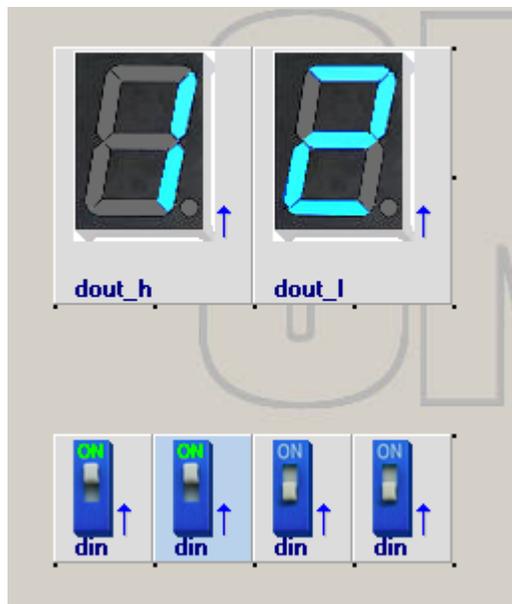
該按鈕來結束該次的運作，否則無法燒

入新的bit檔



挑戰題(一) 十進位顯示器 (1/2)

- 利用虛擬裝置來實作簡易的十進位顯示器
- 請參考前面實驗的步驟與後面的提示完成本實作



挑戰題(一) 十進位顯示器 (2/2)

Hint1:十進位顯示器的解碼器電路

```
module dec4bits(dout_hi,dout_lo,din,clk);
  input  [3:0]  din;
  input                clk;
  output [6:0]  dout_hi, dout_lo;

  wire [6:0] din_hi, din_lo;

  SevenSeg u1(dout_hi,din_hi,clk);
  SevenSeg u2(dout_lo,din_lo,clk);

  assign din_hi = din / 10;
  assign din_lo = din % 10;

endmodule
```