

Quartus II 使用流程說明

1. 實驗軟體與環境設定

軟體下載：請至 altera 網站 <http://www.altera.com> 下載檔案 Quartus II Web Edition V9.0。

2. 實驗內容

圖 1-1 為 4-bit adder，此加法器有輸入 $A[3:0]$ 、 $B[3:0]$ 及 C_0 ，輸出為 $S[3:0]$ 、 C_4 ，請利用 Quartus II 及 Verilog HDL 設計出一個 4-bit adder，並模擬波型加以驗證。圖 1-2 為 4-bit adder 中 Full Adder 的電路圖。

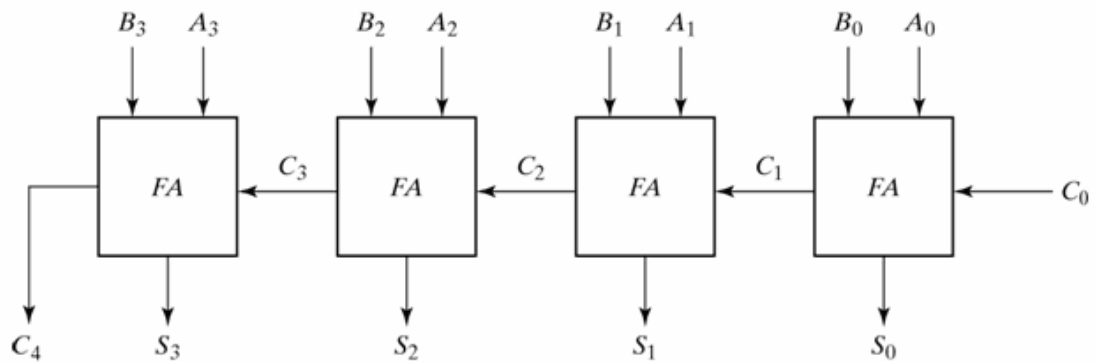


圖 1-1：4-Bit Adder

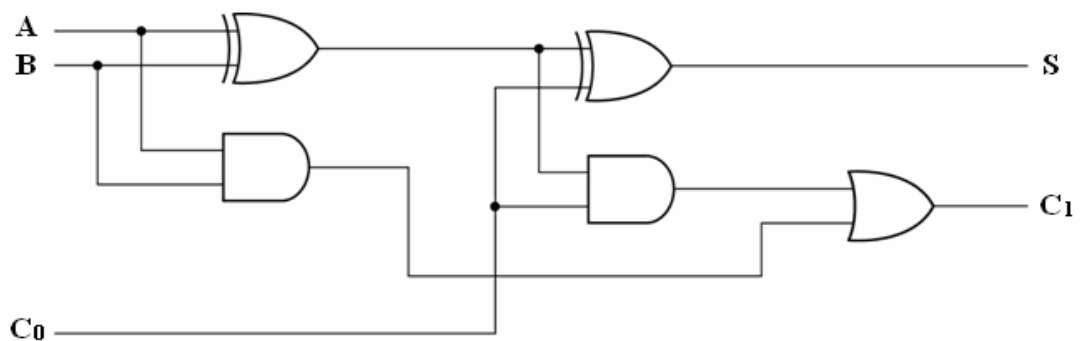
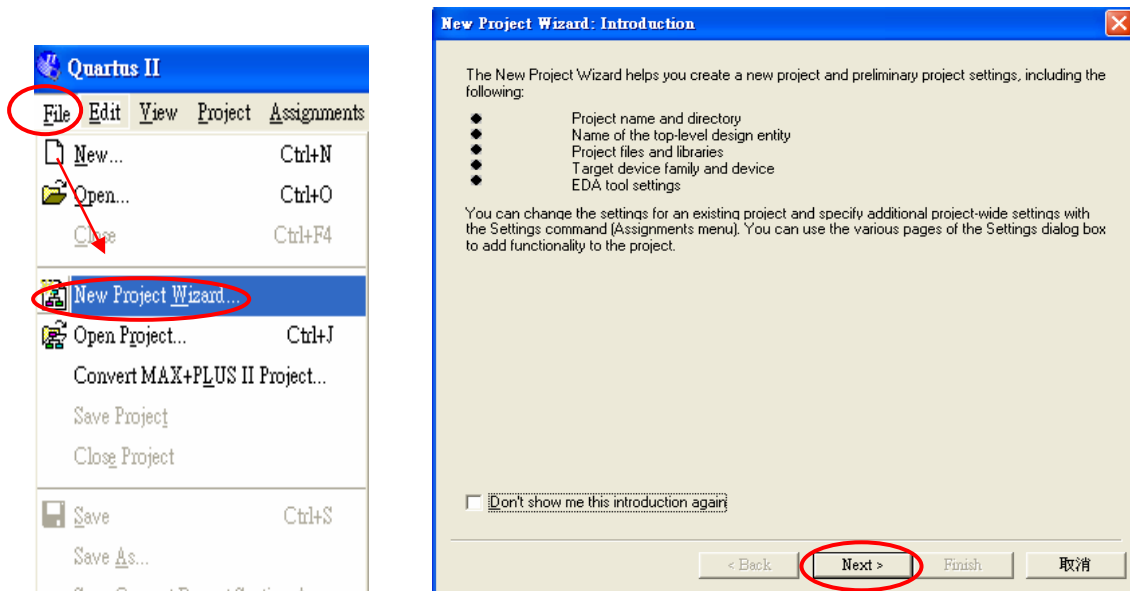


圖 1-2：Full Adder

3. 實驗步驟

Step 1: 首先新建一個 Project，開啟 Quartus II 後，點選 **File** → **New Project Wizard** → **Next**，出現如圖 1-3 之圖示。



如圖 1-3，選擇要儲存 project 的資料夾，並鍵入 project 的名稱按 **Next**，之後出現圖 1-4 之圖示 include file to the project，但我們並沒有 include 其他 file，所以直接按 **Next**。

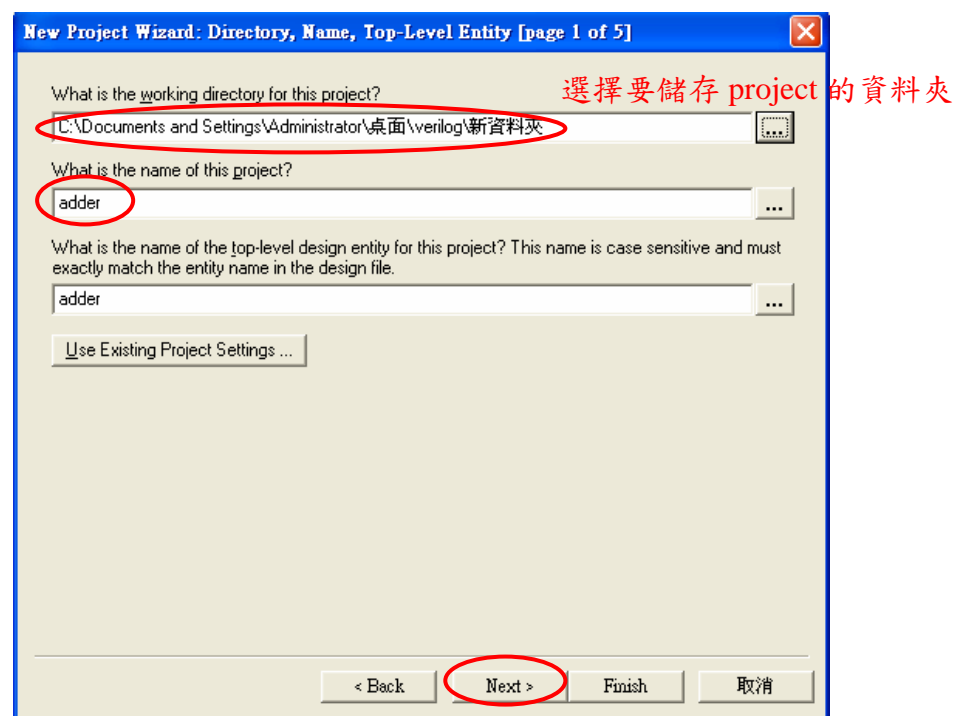


圖 1-3

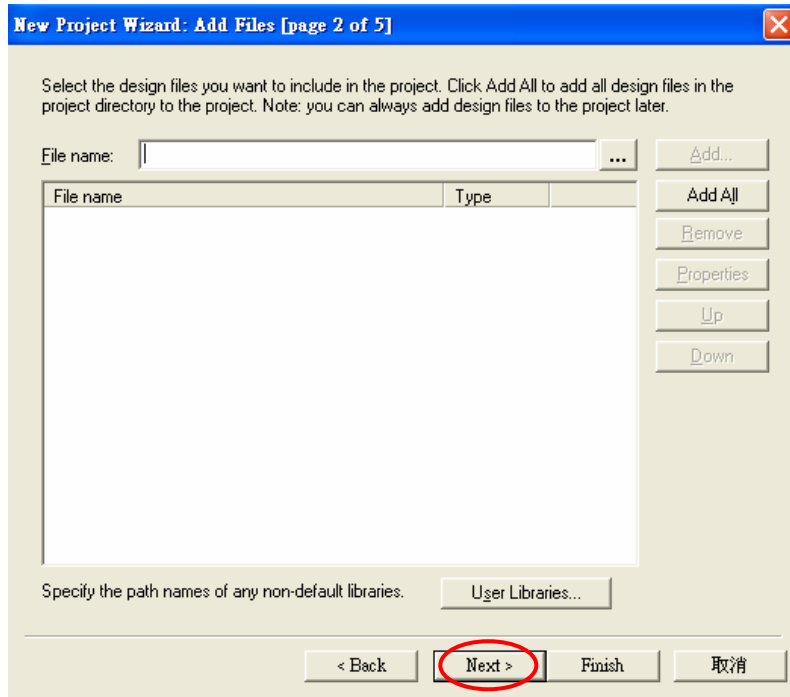


圖 1-4

接下來圖 1-5 為選擇我們所要使用 implement 的 device，因為我們還沒有使用此 device 所以先隨意選擇一個 device 按 *Next*。之後如圖 5-6，選擇其他 EDA tools 的選項也直接按 *Next*。

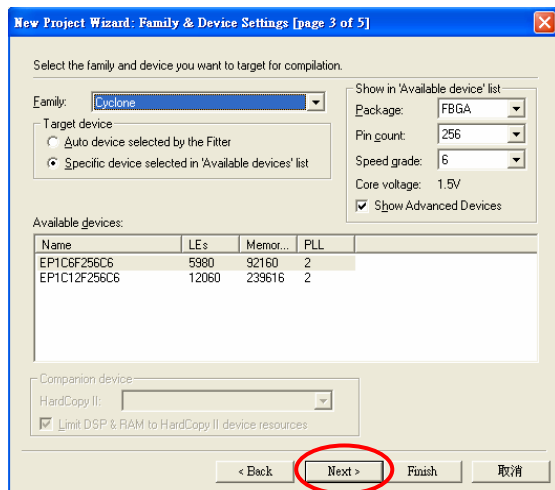


圖 1-5

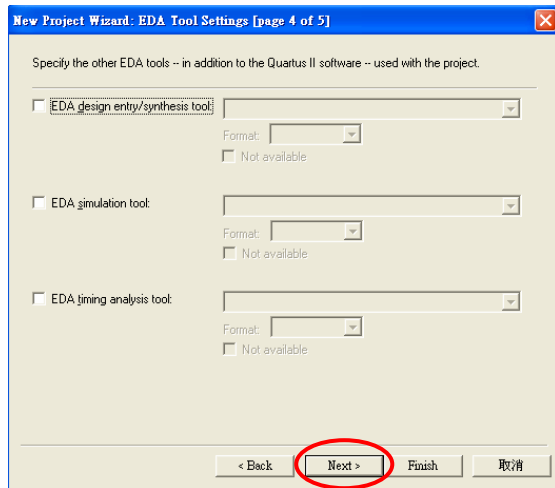


圖 1-6

最後圖 1-7 按 *finish* 後我們便將新的 project 建立完成。

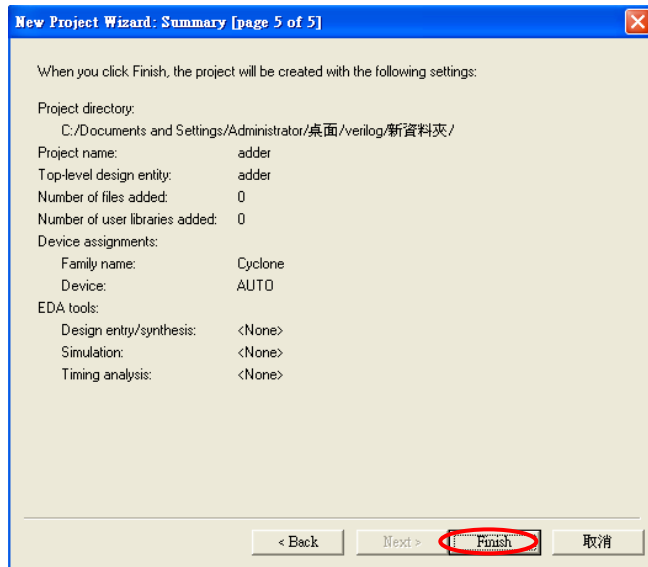


圖 1-7

Step 2: 選擇 *File* → *New* → *Verilog HDL File* → *OK*，之後就可以開始寫 Verilog code 來實現我們所要 4-bit adder。結果如圖 1-8 所示。

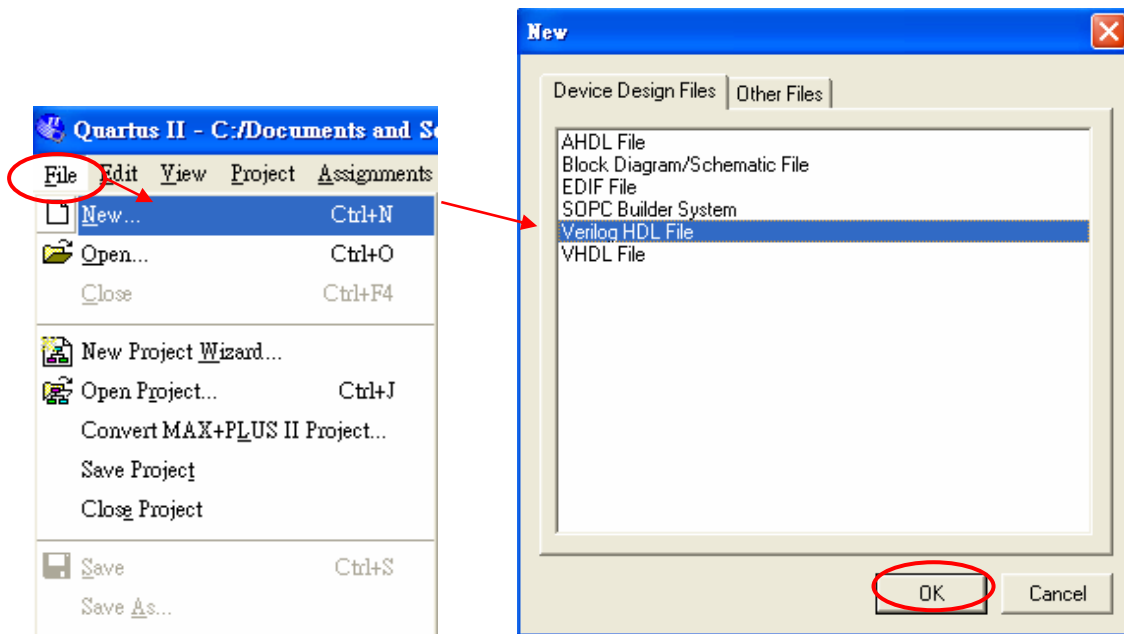


圖 1-8

下列為 Verilog code for 4-bit adder :

//Top level module

module adder(S, C4, A, B, C0);

input [3:0] A, B;

input C0;

output [3:0] S;

output C4;

wire C1, C2, C3; **//Intermediate carries**

//Instantiate the fulladder

fulladder FA0(S[0], C1, A[0], B[0], C0);

fulladder FA1(S[1], C2, A[1], B[1], C1);

fulladder FA2(S[2], C3, A[2], B[2], C2);

fulladder FA3(S[3], C4, A[3], B[3], C3);

endmodule

//Description of full adder

module fulladder(S, C, x, y, z);

input x, y, z;

output S, C;

//Define combinational logic circuit

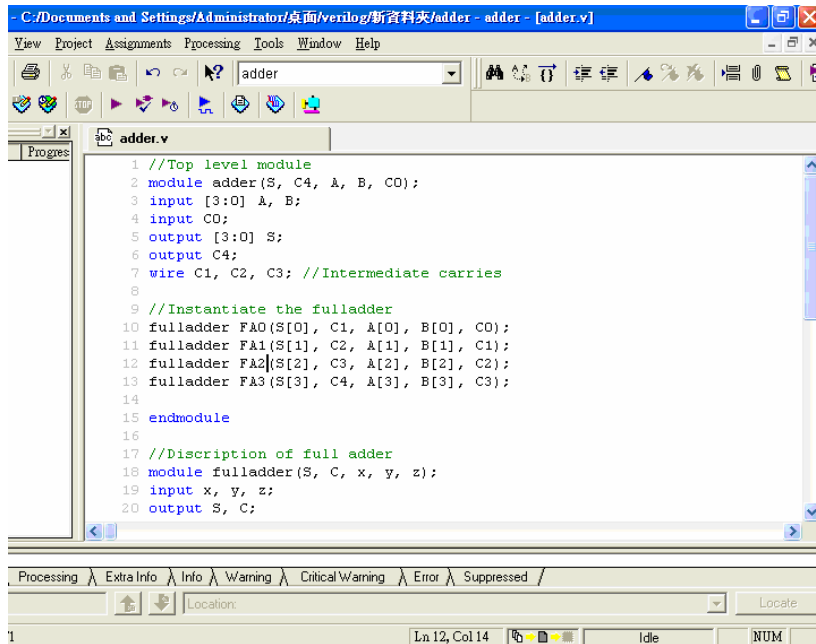
assign C = { (x ^ y) & z } | (x & y);

檔名需和 project 的名稱及存檔的檔名相同

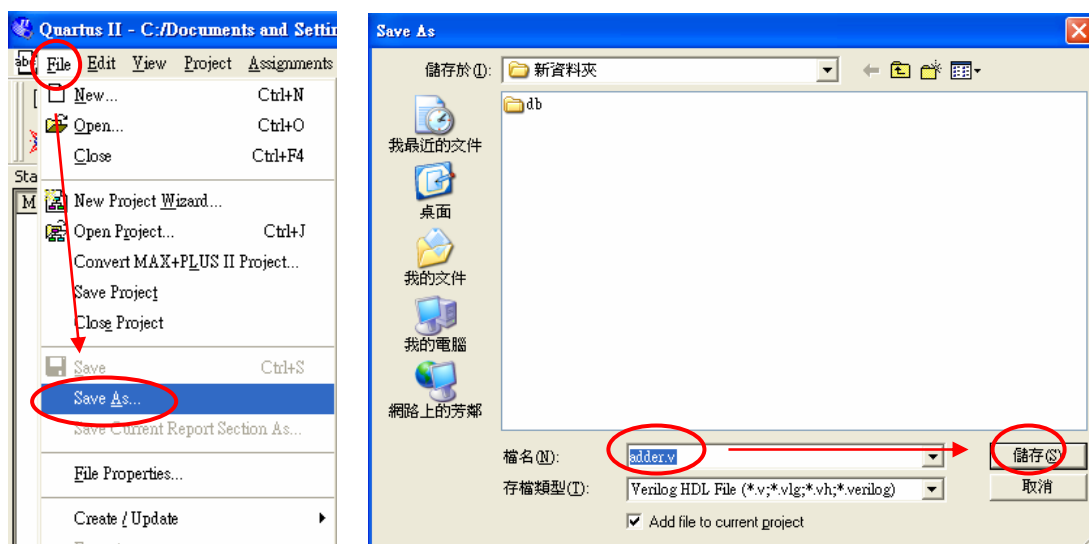
```


assign S = x ^ y ^ z;
endmodule

```



接著我們將寫好的檔案儲存，點擊 Quartus II 的 **File** → **Save As** 後，將檔案選擇儲存於和所建立 project 同一資料夾內，且儲存之檔名必須和所寫 code 的 **Top module** 名稱及 **project** 的名稱一樣。存檔的類型為 Verilog HDL file，副檔名為 .v 檔。



Step 3: 當我們所寫的 code 檔完成之後，接下來要將 code 做 compile 的動作，如圖 1-9 點擊 **Processing** → **Start Compilation**，或者直接按 toolbar 的 icon 。

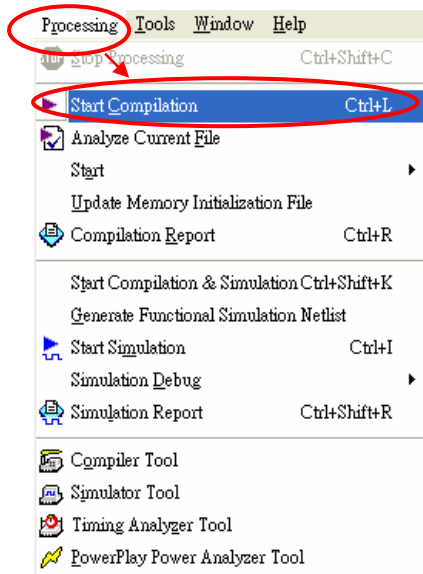
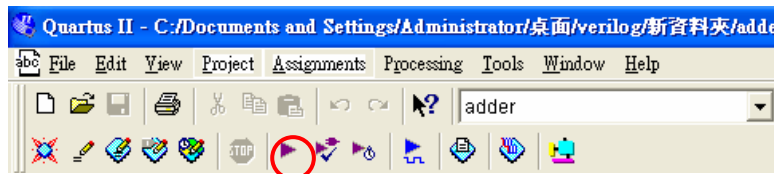
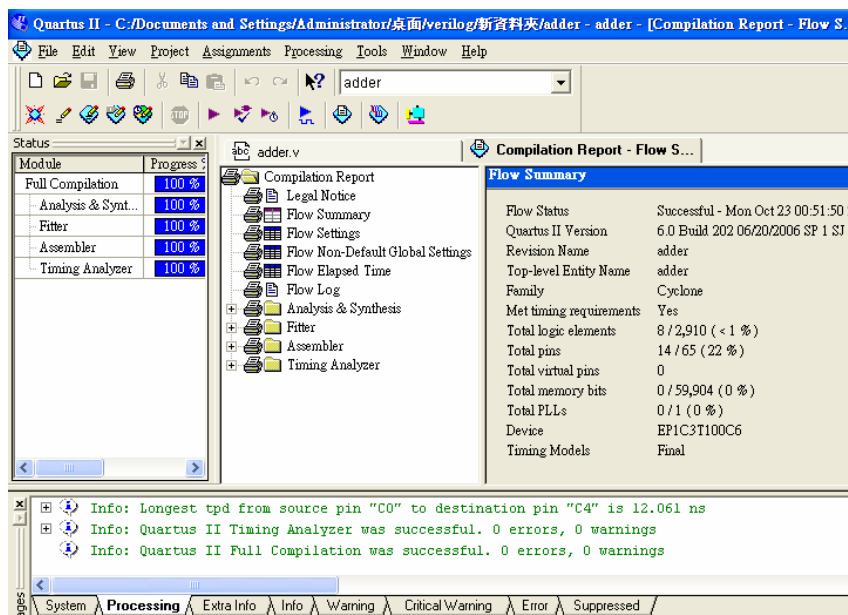
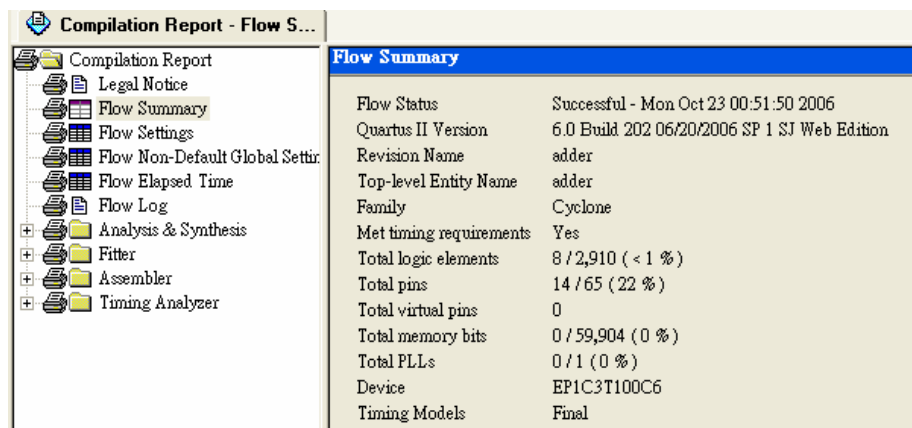


圖 1-9

成功 compile 之後的畫面如下：

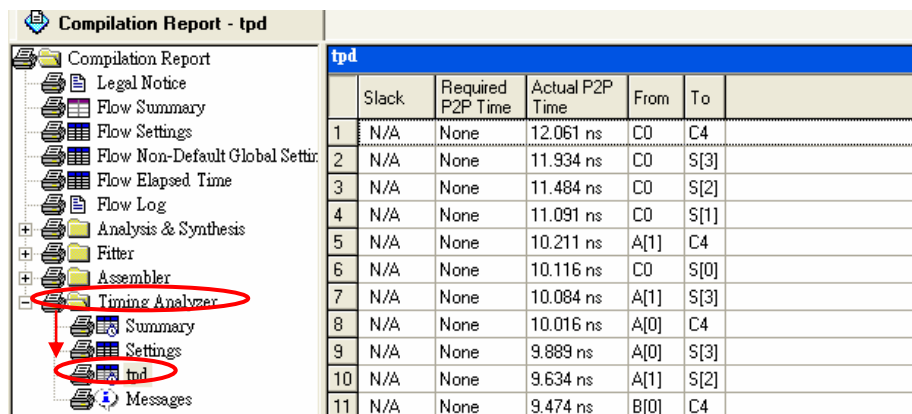


此時我們可以在 Compilation Report 的地方看到 compile 後的資料。



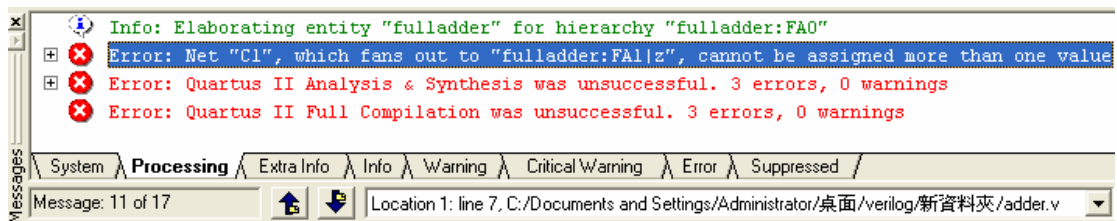
Flow Summary	
Flow Status	Successful - Mon Oct 23 00:51:50 2006
Quartus II Version	6.0 Build 202 06/20/2006 SP 1 SJ Web Edition
Revision Name	adder
Top-level Entity Name	adder
Family	Cyclone
Met timing requirements	Yes
Total logic elements	8 / 2,910 (< 1 %)
Total pins	14 / 65 (22 %)
Total virtual pins	0
Total memory bits	0 / 59,904 (0 %)
Total PLLs	0 / 1 (0 %)
Device	EP1C3T100C6
Timing Models	Final

我們也可以在 Compilation Report 的地方點擊 **Timing Analyzer** → *tpd*，此處可以得知 Critical path 為何。

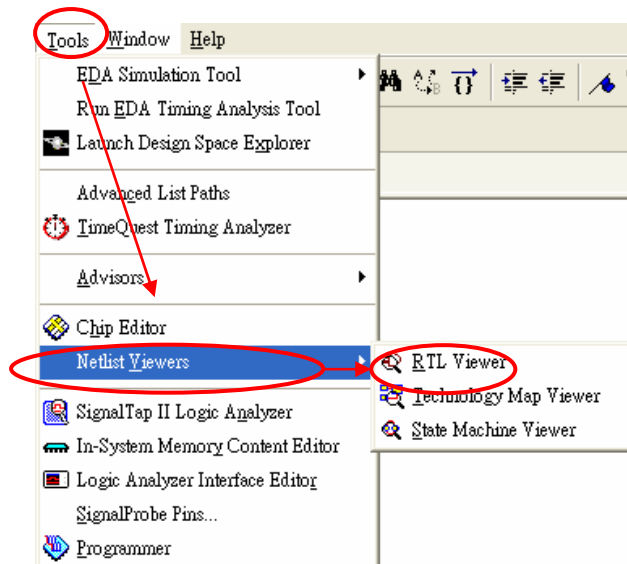


tpd						
	Slack	Required P2P Time	Actual P2P Time	From	To	
1	N/A	None	12.061 ns	C0	C4	
2	N/A	None	11.934 ns	C0	S[3]	
3	N/A	None	11.484 ns	C0	S[2]	
4	N/A	None	11.091 ns	C0	S[1]	
5	N/A	None	10.211 ns	A[1]	C4	
6	N/A	None	10.116 ns	C0	S[0]	
7	N/A	None	10.084 ns	A[1]	S[3]	
8	N/A	None	10.016 ns	A[0]	C4	
9	N/A	None	9.889 ns	A[0]	S[3]	
10	N/A	None	9.634 ns	A[1]	S[2]	
11	N/A	None	9.474 ns	B[0]	C4	

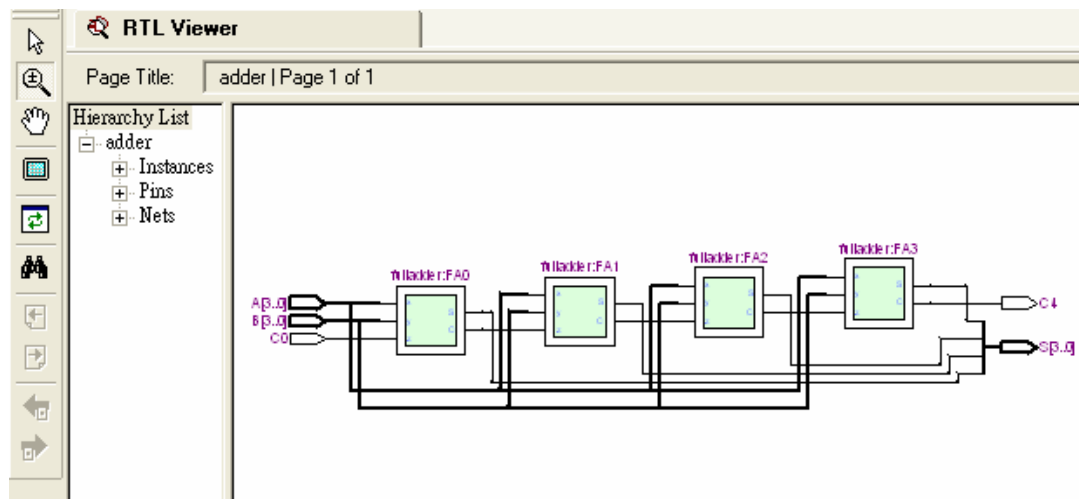
若 compile 之後發生錯誤，我們可以在底下的視窗得知發生錯誤的原因，且在 Error 處點擊 2 下，可以得知我們所寫 code 發生錯誤的地方。



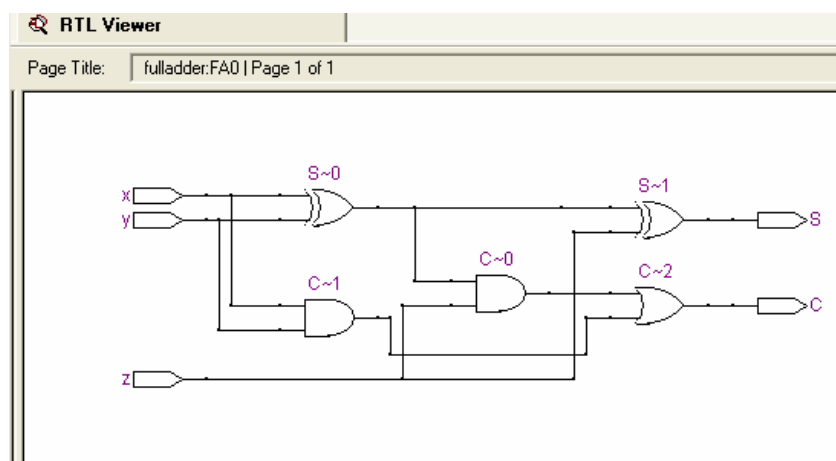
此外，點擊 **Tools** → **Netlist Viewers** → **RTL Viewer**，可以看見 RTL Level 之電路。



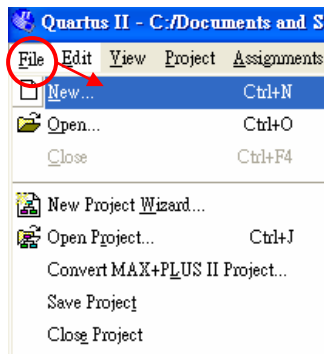
RTL Viewer



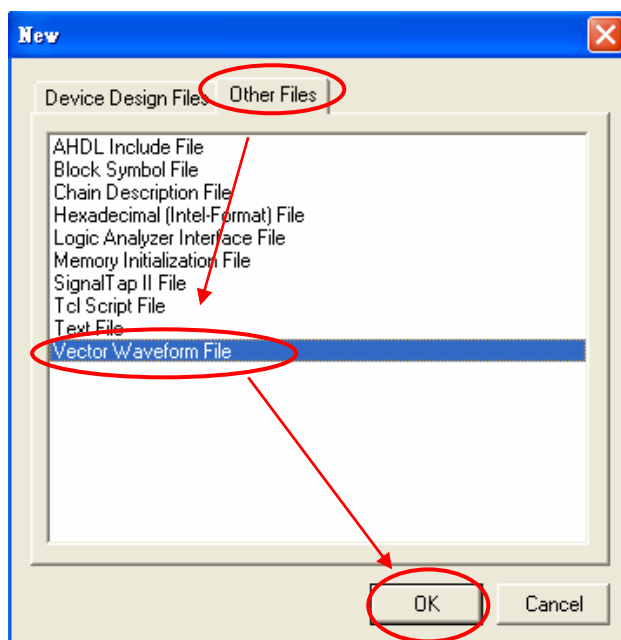
直接在圖示的任一 fulladder 點擊 2 下，便可看見 fulladder 架構如圖 1-2 所示相同。



Step 4: 最後我們要做 simulation 的動作，來驗證我們所寫的程式和題目所要求的是否相同。首先開啟一個 Vector Waveform file，點擊 **File** → **New**



選擇 **Other Files** → **Vector Waveform file**，點擊 **OK**。



開啟後的圖示如圖 1-10

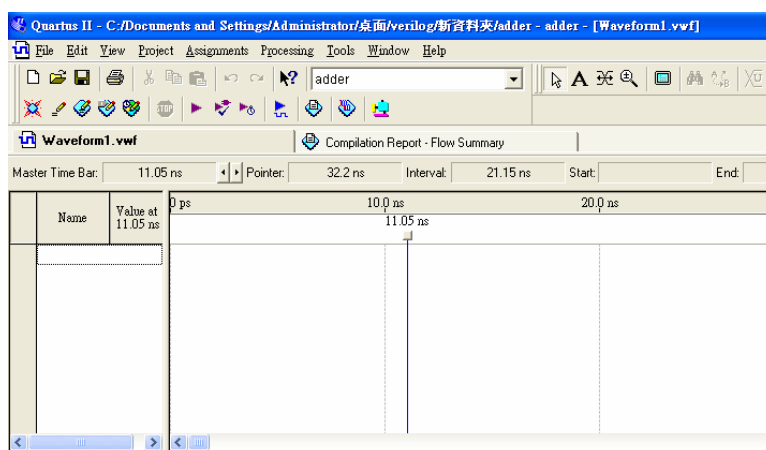
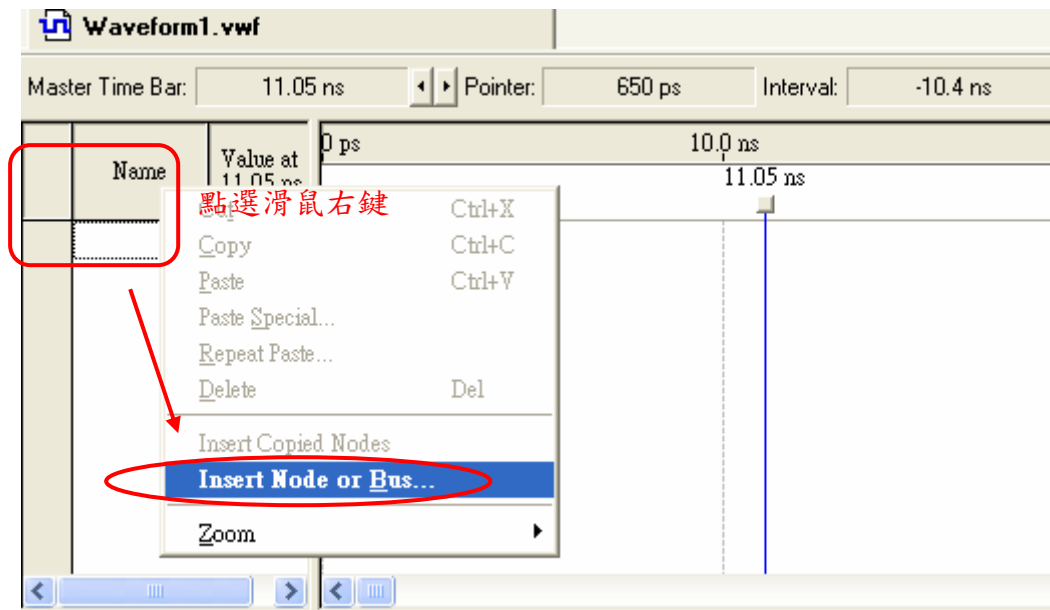
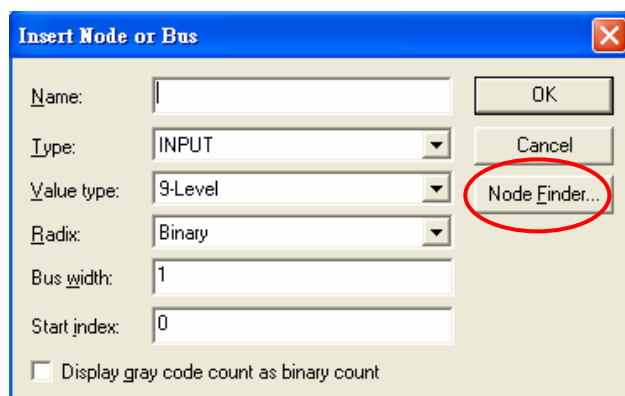


圖 1-10

接著在 Name 的地方點選滑鼠右鍵，點選 *Insert Node or Bus* 來選取輸入的 input 及我們所要觀察的 output。



點選 *Node Finder*



在 Filter 的地方選擇 *Pins:all* 後，點擊 *List*，如圖 1-11 所示。

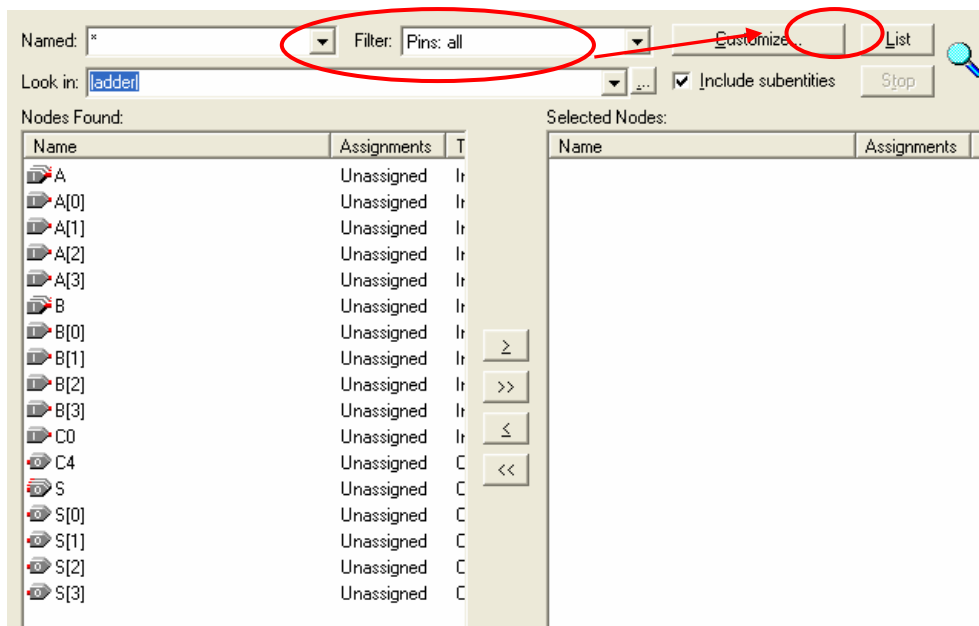
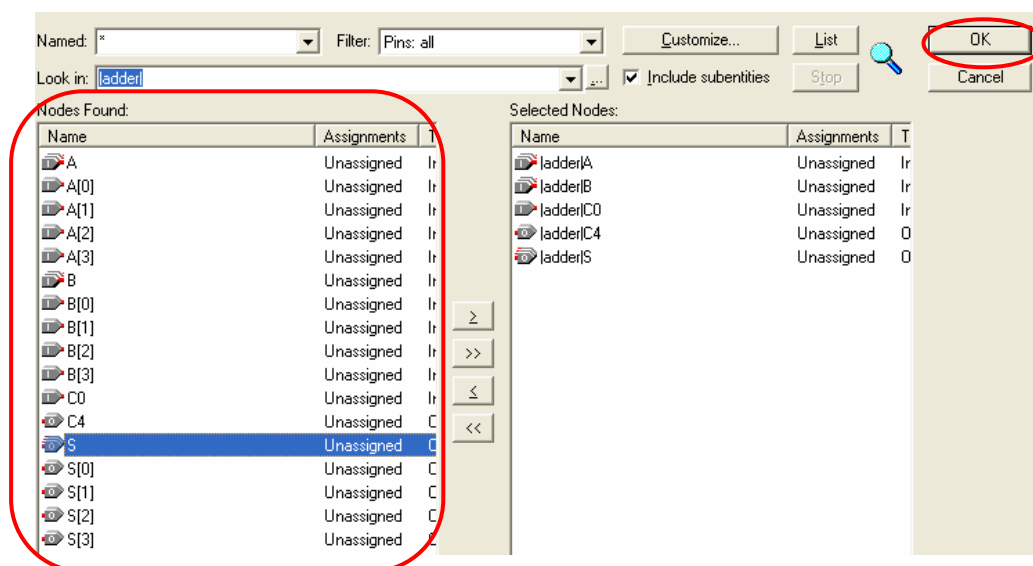
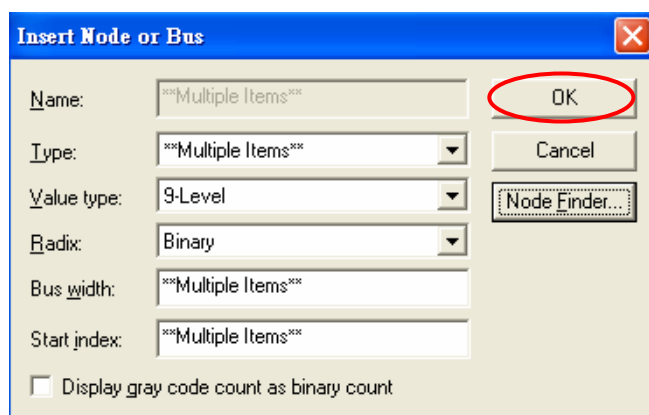



圖 5-11

點擊 2 下滑鼠左鍵，在 Node Found 處選擇輸入的 input 及我們所要觀察的 output。



點選 OK



之後的畫面如圖 1-12 所示，點擊圖示可以調整波型的視窗。

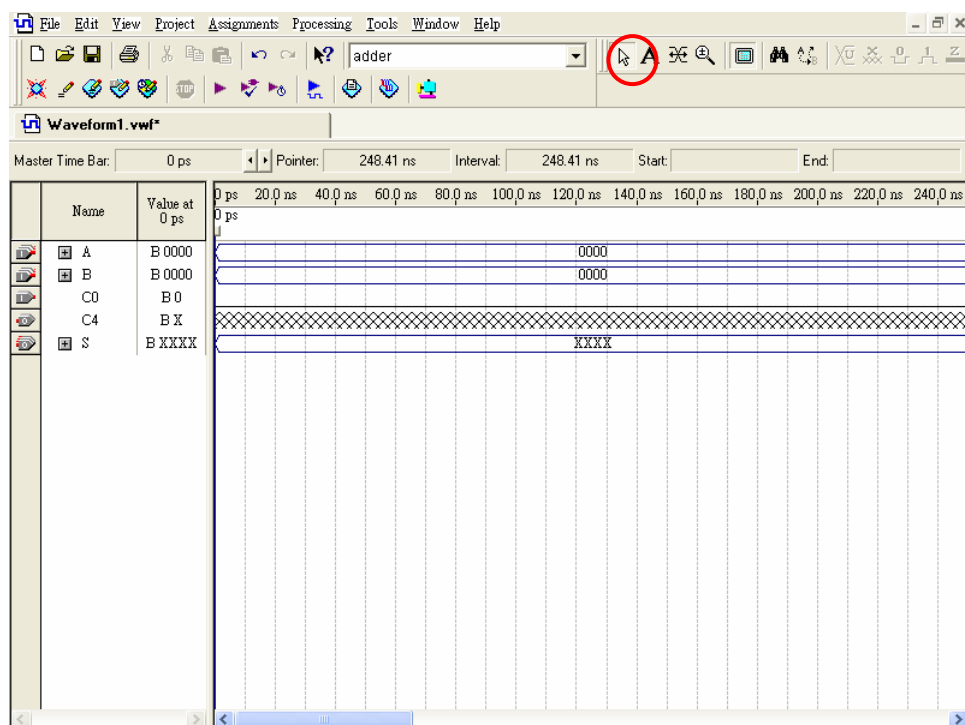
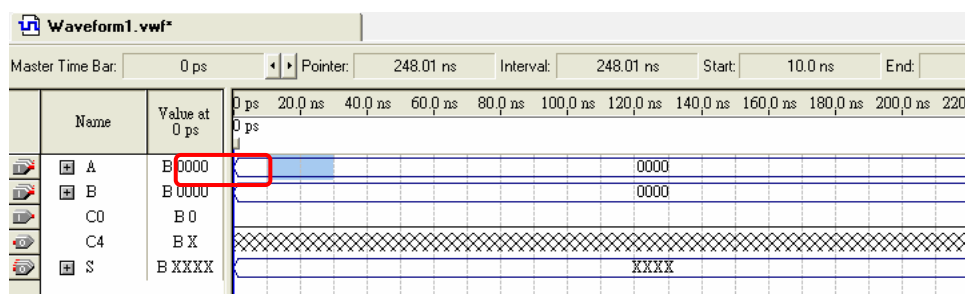


圖 5-12

接下來我們設定波型給 input A、B 及 C0，首先選取所給波型的區域。



雙擊滑鼠左鍵後，在 *Numeric or named value* 處可以輸入所給 input 的值，輸入之後點擊 **OK**。

Arbitrary Value

Time range

Start time: 10.0 ns

End time: 30.0 ns

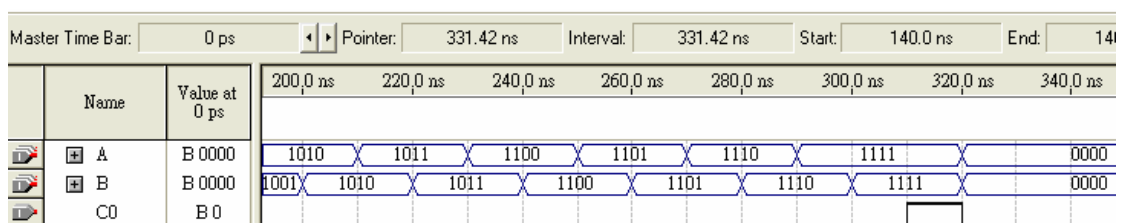
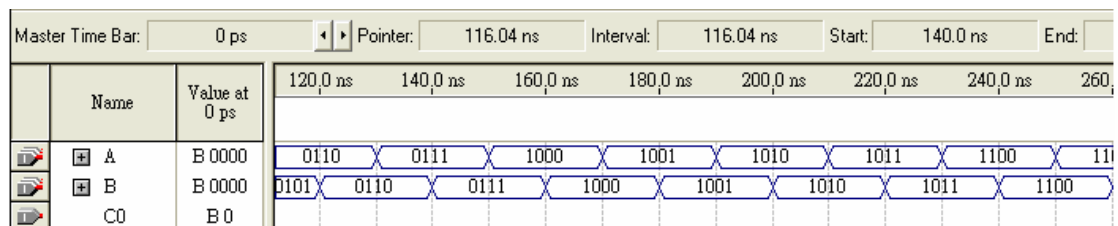
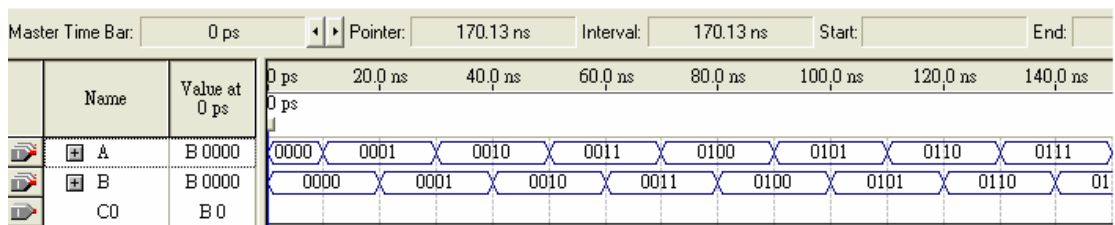
Arbitrary value

Radix: Binary

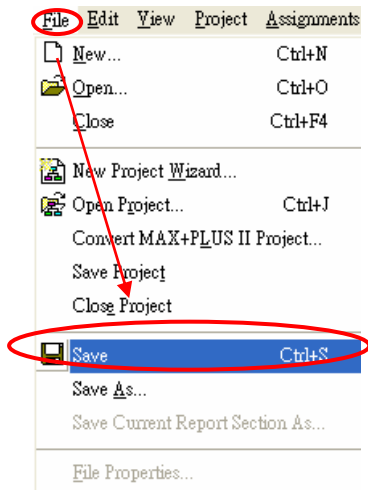
Numeric or named value: 0001

OK Cancel

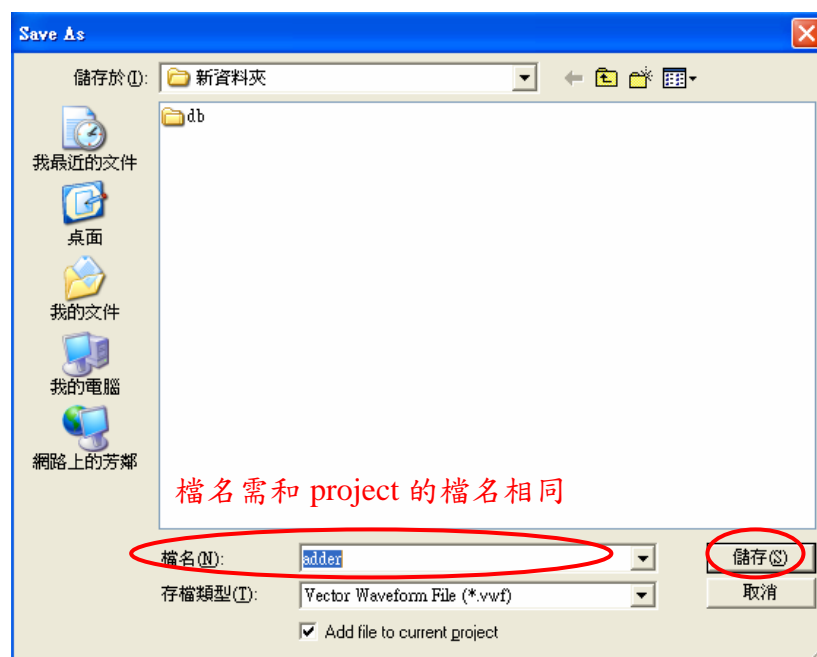
編輯好的波型如下列圖所示。



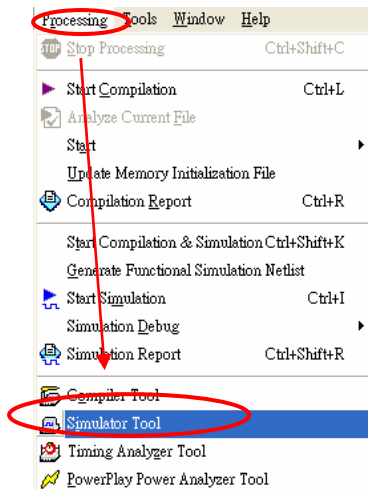
當編輯完波型之後，點擊 **File** → **Save**，或直接點擊  圖示，儲存檔案。



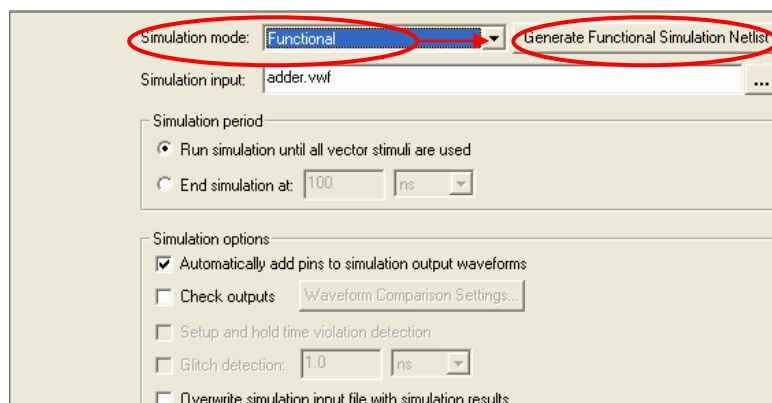
儲存之檔名需和 project 檔名相同，之後點擊**儲存**，存檔類型為 Vector Waveform file，副檔名為 .vwf 檔。



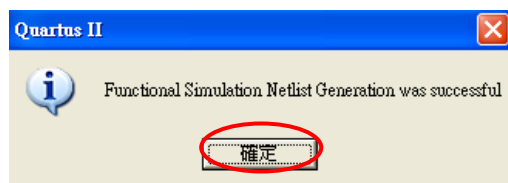
當 input 的波型設好，也將檔案儲存完畢之後，我們開始做模擬的動作。點擊 **Processing → Simulator Tool**。



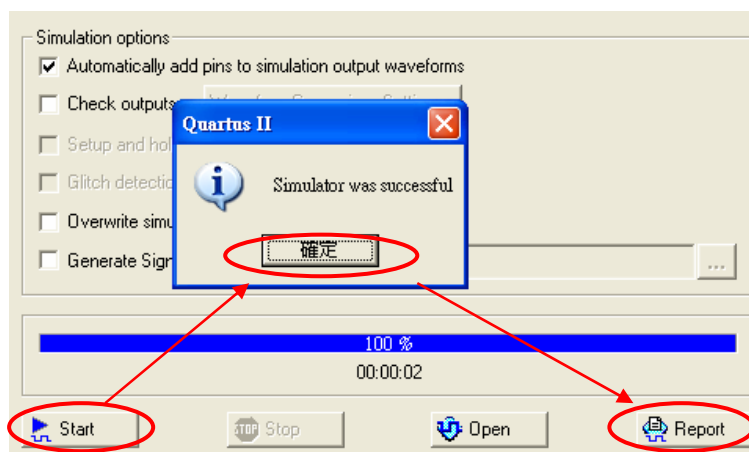
在 Simulation mode 的地方，我們可以選擇只做 Functional simulation 或 Timing 的 simulation。此處我們先選擇 **Functional simulation**，之後點擊 **Generate Functional Simulation Netlist**。



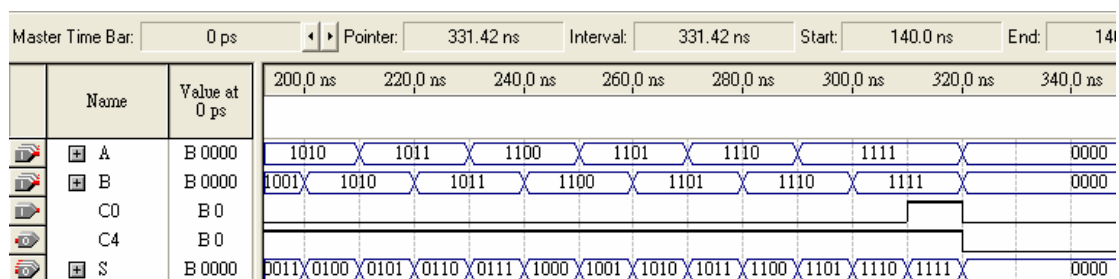
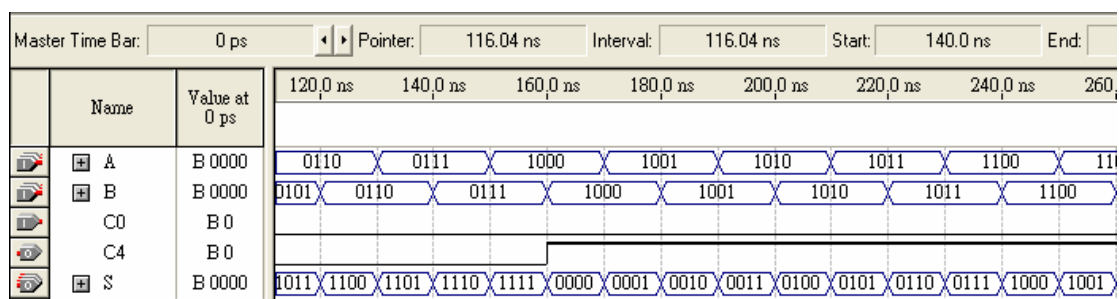
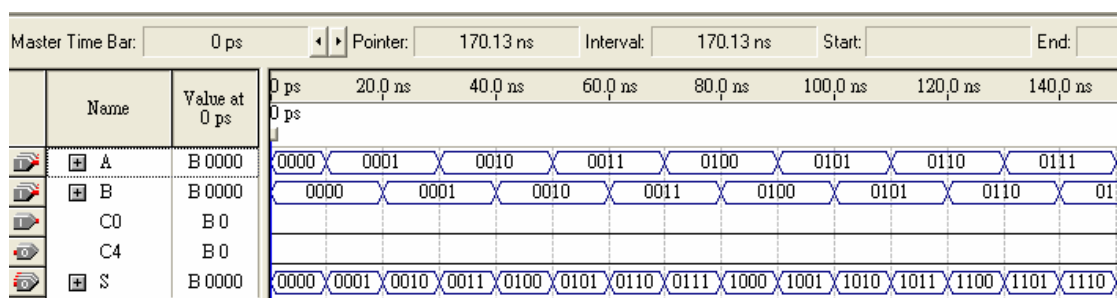
完成後按**確定**



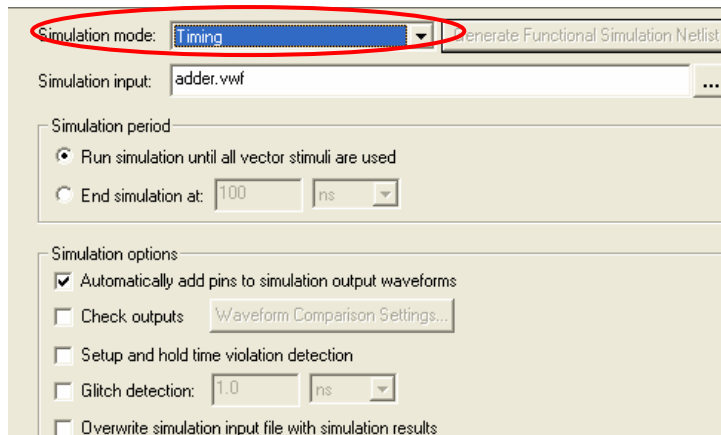
之後在 Simulation Tool 的地方點擊 **Start**，點擊**確定** → **Report**。



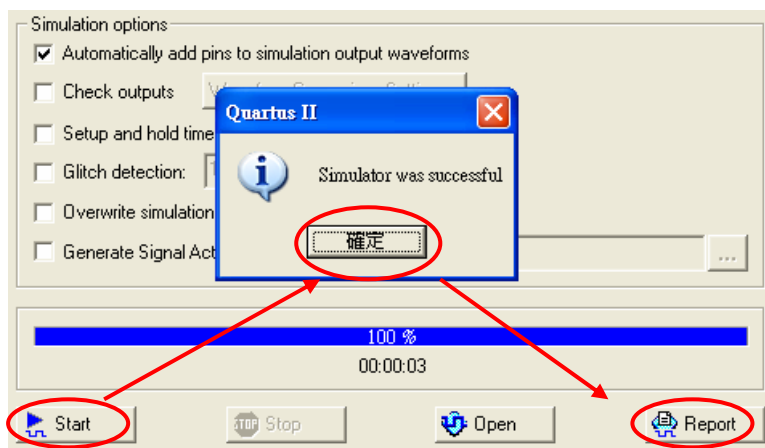
我們可以得到模擬後的波型圖如下列的圖示，所有輸出的組合都必須模擬測試。



之後點選回剛才的 Simulation Tool，我們改做 Timing 的 simulation，在 Simulation mode 處點選 Timing。



點擊 **Start** → **確定** → **Report**。



加入的 Timing 模擬後的波形如下，驗證輸出的結果是否為題目所要求。

