

親愛的學員們好

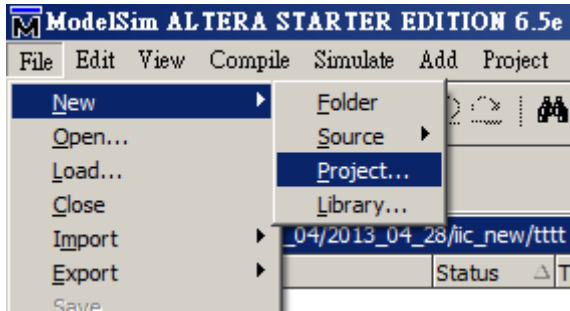
因為我手邊沒有上課用的版子，所以，我就以手邊有的板子來做範例，我手邊的板子是艾睿電子(Arrow) 所提供的，以 Altera 最新的 28 奈米 FPGA Cyclone-V 的 BeMicro CV 的板子，FPGA 編號: 5CEFA7F23C8ES，這個 FPGA 容量是 150K 的 LEs; 而上課用的板子則是 Cyclone4 : EP4CE30F23C8, 容量為 30K LEs. 所以，在建立 QuartusII 的 Project 時，在畫面上的 Device 選擇會是不同的。

首先，第一個範例是 SPI Master 的 ModelSim 操作以及模擬流程。

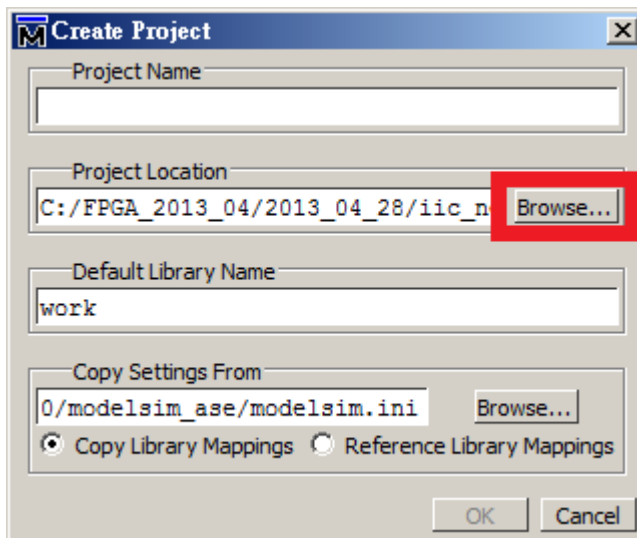
1. 建立 project file folder, 在建立 project 時的注意事項: folder 名稱不可以特殊字元, 空白, 或者中文字 (此範例程式的 folder 在 C:\FPGA_2013_04\SPI_MASTER)
2. 將所需要的檔案 copy 至 project 目錄底下
以這個範例，需要的檔案有
 - a) SPIMasterFSM.v : SPI Master 的程式
 - b) spi_master_top.v : SPI Master 的 test bench



3. 開啟 ModelSim
4. 建立 ModelSim 的模擬 Project
“File” → “New” → “Project”



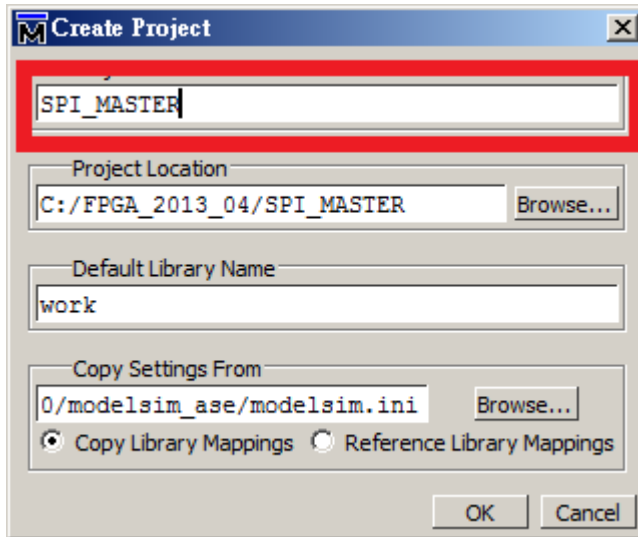
5. 出現“Create Project”的視窗，並且在“Project Location”的欄位，按下“Browse...”



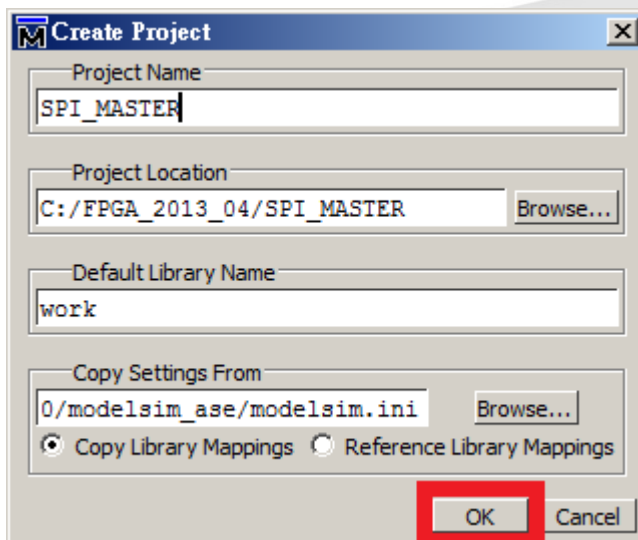
6. 選擇正確的路徑後，按“確定”。(此範例的路是 C:\FPGA_2013_04\SPI_MASTER)



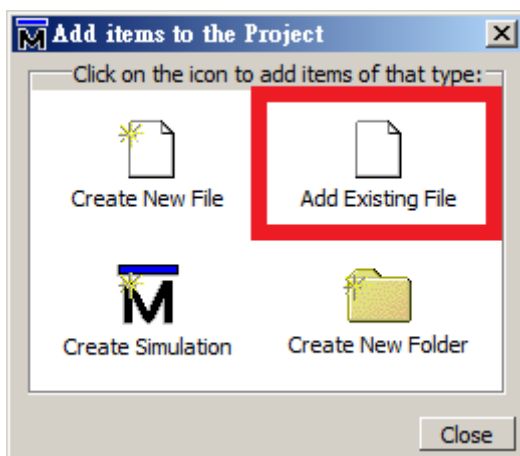
7. 回到 Create Project 後，在“Project Name”的欄位填入 Project Name，命名可以任意，在此範例中我將這個 Project 命名為“SPI_MASTER”。



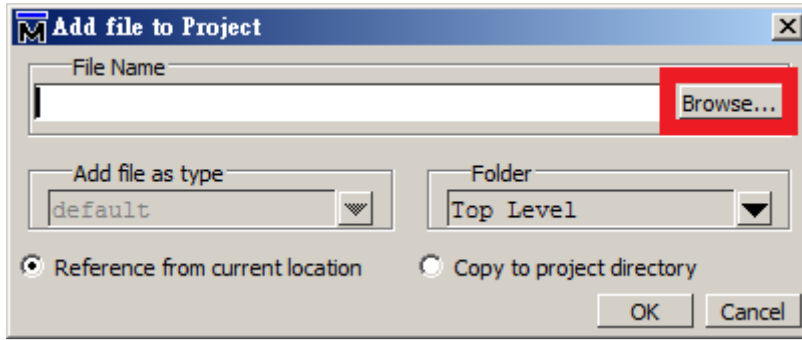
8. 然後, 按下 “OK”



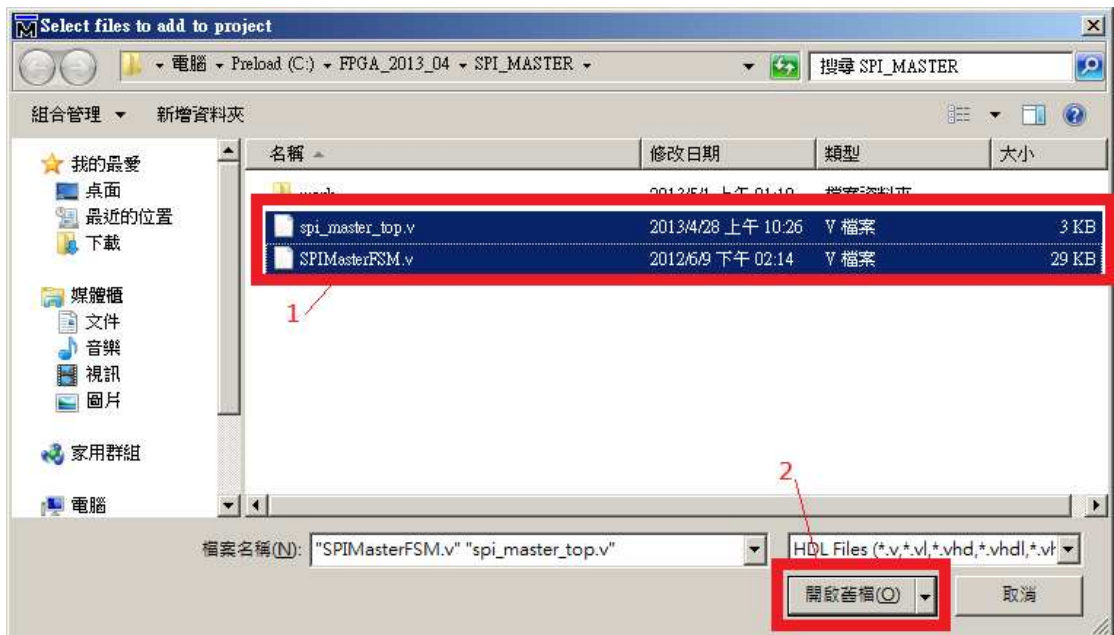
9. 接著, 便是將檔案加入 Project 的動作, 請在新出的視窗 “Add items to the Project” 中按下 “Add Existing File” 的 icon



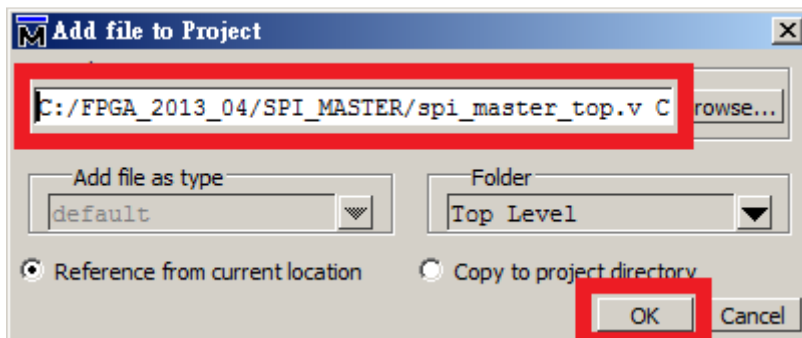
10. 按下 “Browse...”



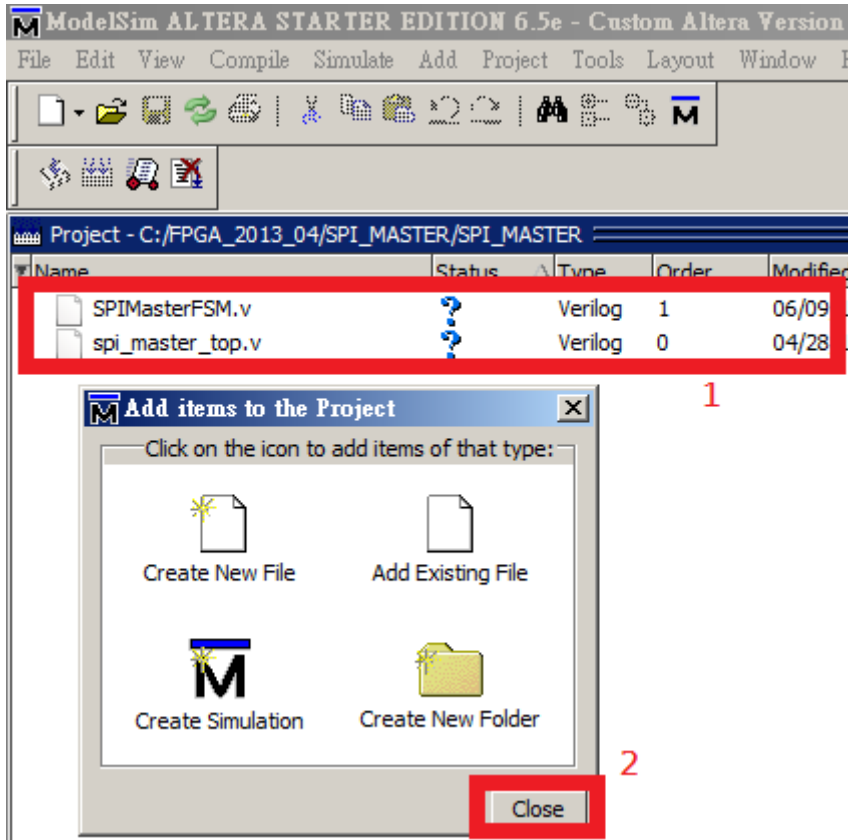
11. 選擇 spi_master_top.v 以及 SPIMasterFSM.v (確認在 “檔案名稱” 的欄位內有出現這兩個檔案), 後按下 “開啟舊檔”



12. 回到 “Add file to Project” 的視窗時, 確認在 “File Name” 的欄位不是空的, 然後按下 “OK”

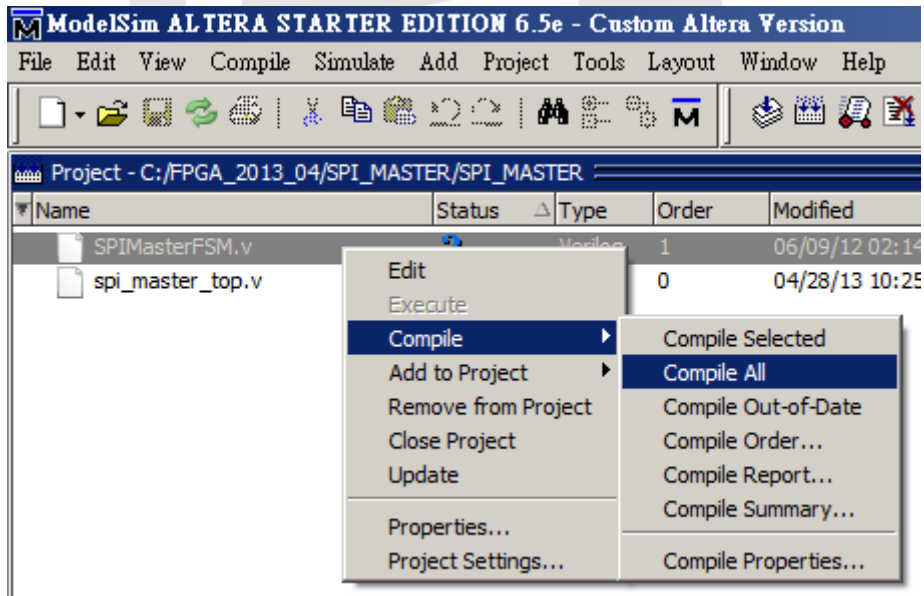


13. 首先, 確認 Project 中有兩個檔案 spi_master_top.v 以及 SPIMasterFSM.v, 然後再按下 “Add items to the Project” 視窗的 “Close”

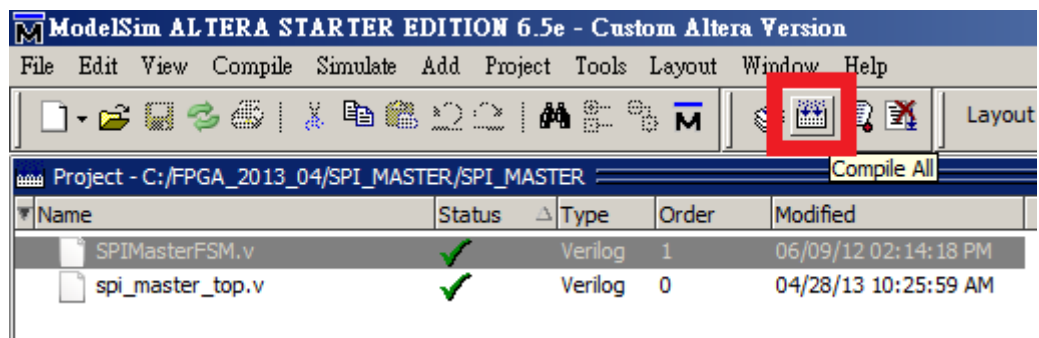


14. 接下來就是做 Compiler 的動作, 方有下列三種方式

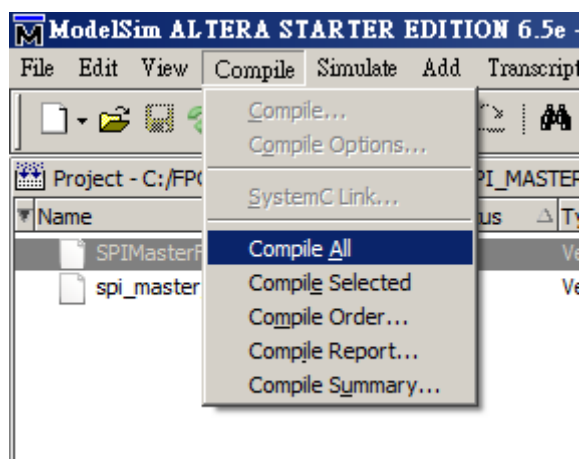
- a) 在滑鼠移至檔案的地方按右鍵, 後選擇 “Compile” → “Compile All”



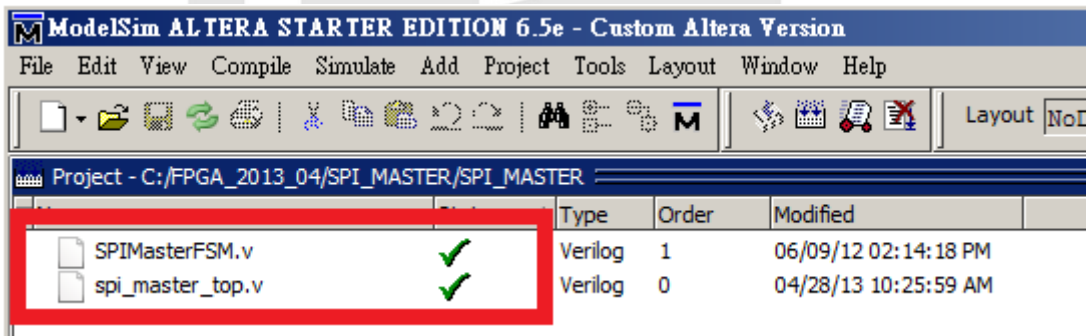
b) 在上方的 icon 中找到紅色框住的圖樣



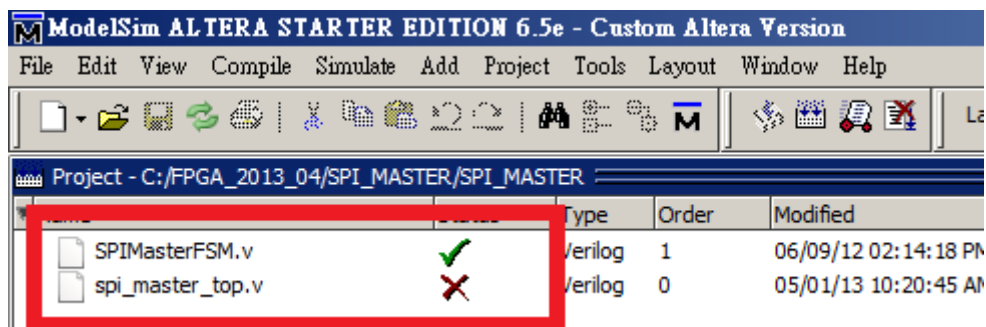
c) “Compiler” → “Compile All”



15. 在 compiler 之後，確認狀態，若沒有語法上的錯誤的話，則在”Status” 欄位應為綠色的勾勾，如下圖所示



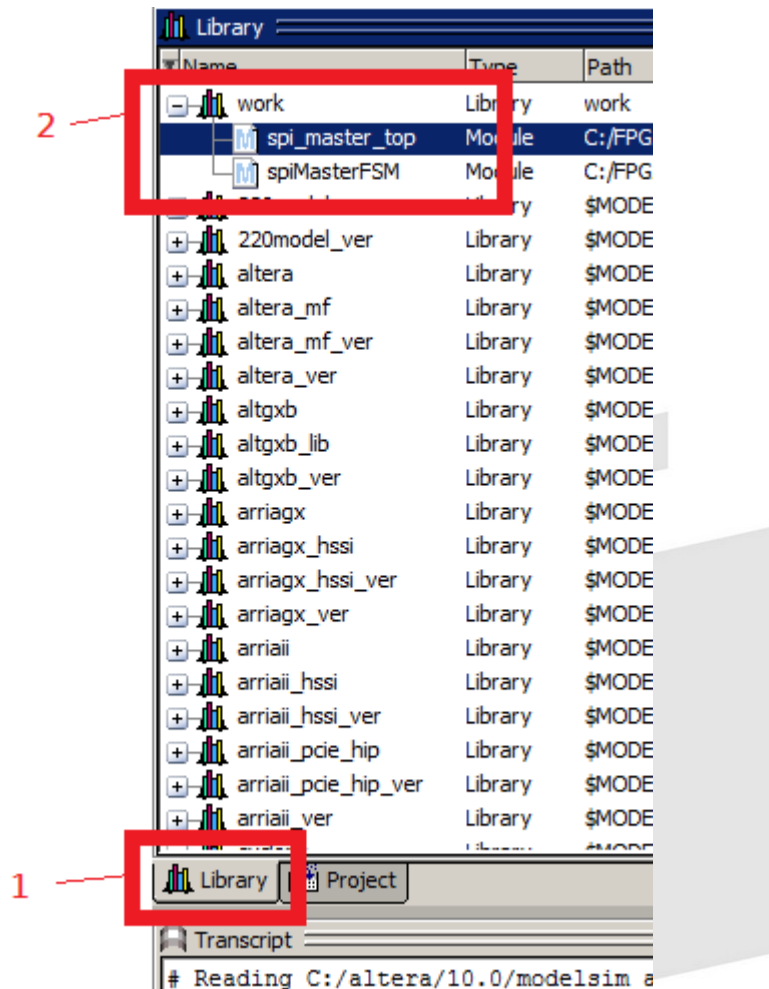
若有語法上的錯誤，則會出現紅色的叉叉，如下圖所示



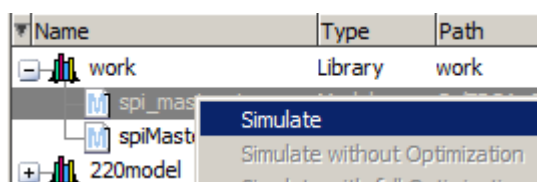
語法錯誤在這裡就不再多做解釋。

16. 在 compiler 之後，確認無任何語法錯誤之後，接下來就是 load library 的動作。

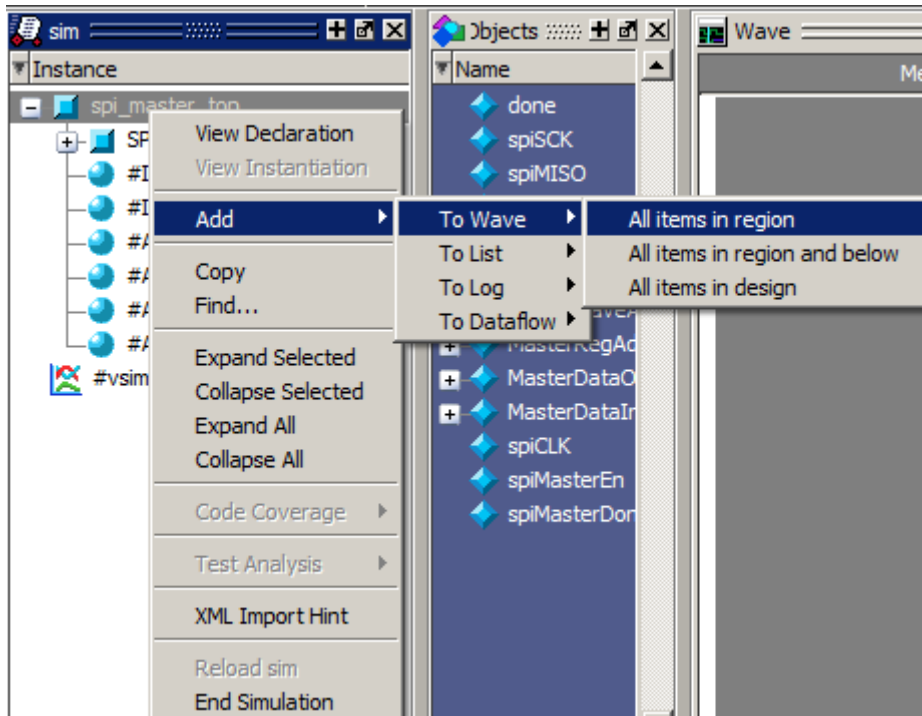
首先，在把工作區切至 “Library”，然後將 work 展開，在這裡可以看到 spi_master_top 跟 spiMasterFSM 的 Module



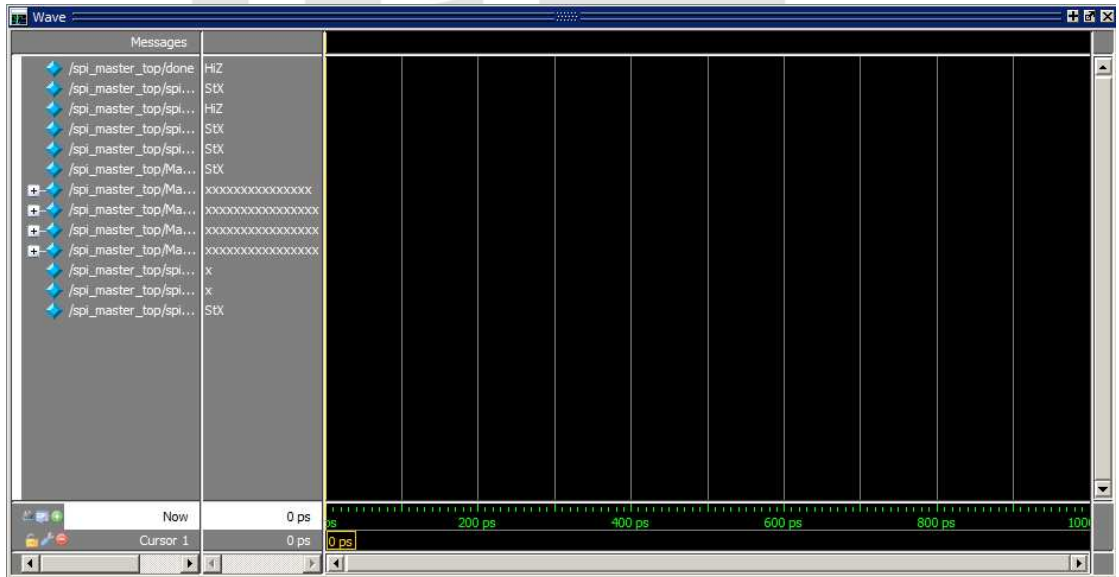
17. 選取 spi_master_top 的 Module，並且將滑鼠移至這個 Module 上按右鍵，後選擇 “Simulate”，或者直接在這個 Module 上按滑鼠左鍵兩下。



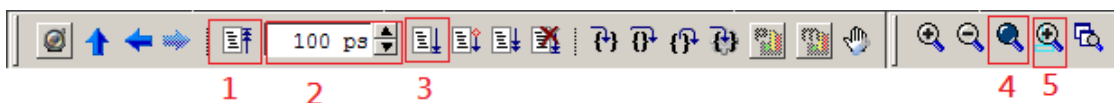
18. 在 load simulate 成功後，會出現“Sim”的視窗，在 sim_master_top 上按滑鼠右鍵，選擇 “Add” → “To Wave” → “All items in region”，如下圖所示



19. 在 load 完要看的信號至 wave 視窗後，在 wave 視窗將會出現所要觀察的信號如下圖所示:

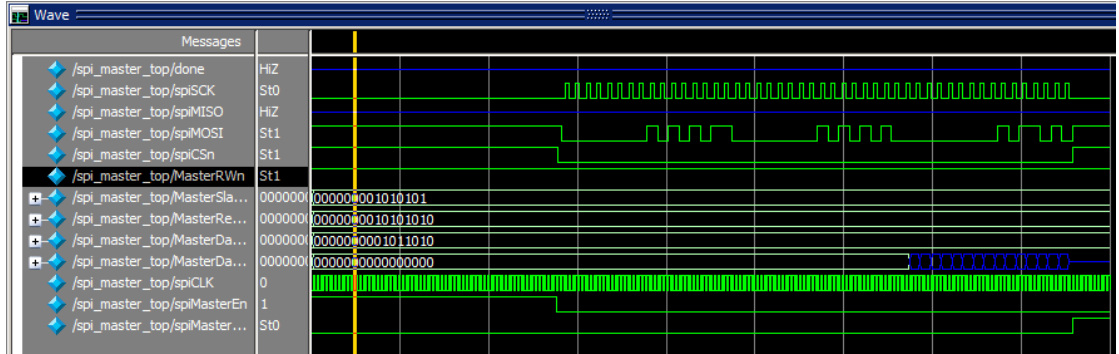


20. 在視窗上方的 tool bar 上可以看到如下圖所示的 icon 按鍵

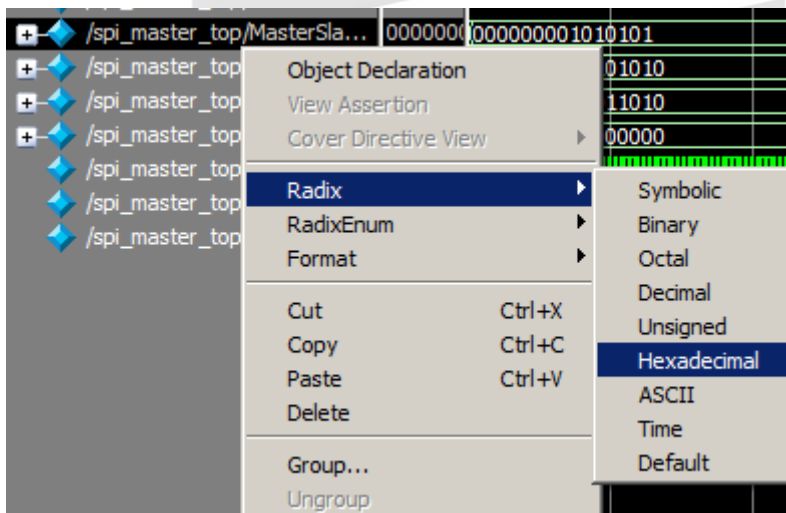


1. reset：清除所有結果，並重置所有狀態至時間點 0 的狀態
2. 每執行一次的時間單位

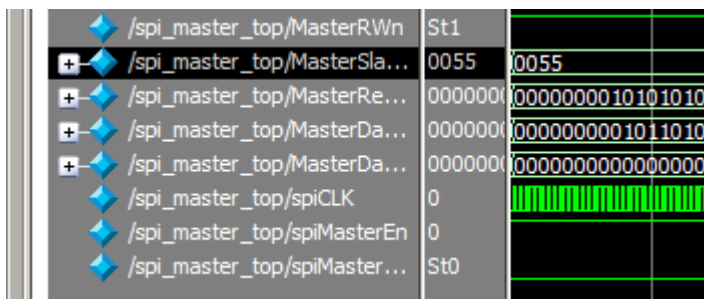
3. 執行
 4. 將所有波形縮放至視窗大小
 5. 滑鼠指定之位置為中心點放大波形
21. 執行完結果如下:



22. 在圖中，信號的內容值顯示在 ModelSim 的 default 是二進制，若是看不習慣可以更改顯示方式。在欲更改的信號上按滑鼠右鍵，選擇 Radix 後自行選擇想要的規格即可。如下圖所示:

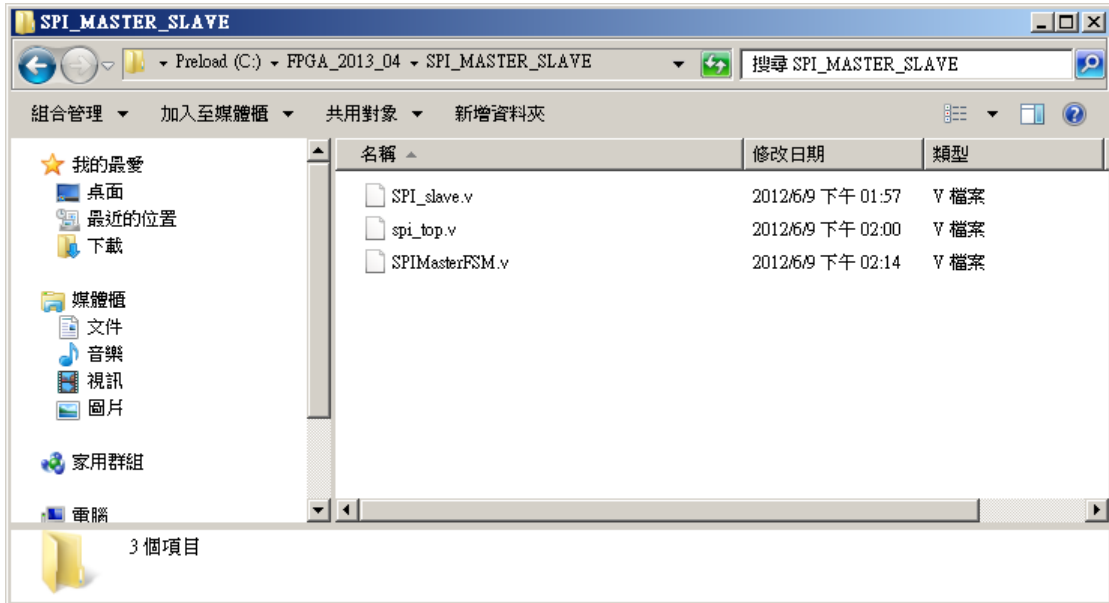


更改為 hexadecimal 後，即改成 十六進 來顯示信號的內容

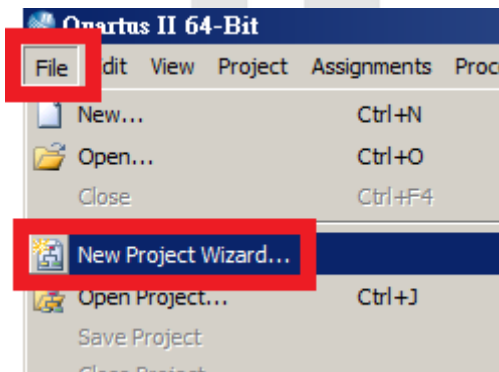


第二個範例是以 16 bits SPI Master 跟 16bits SPI Slave 的 Quartus II project 建立以及實際上板子 debug 的操作流程。

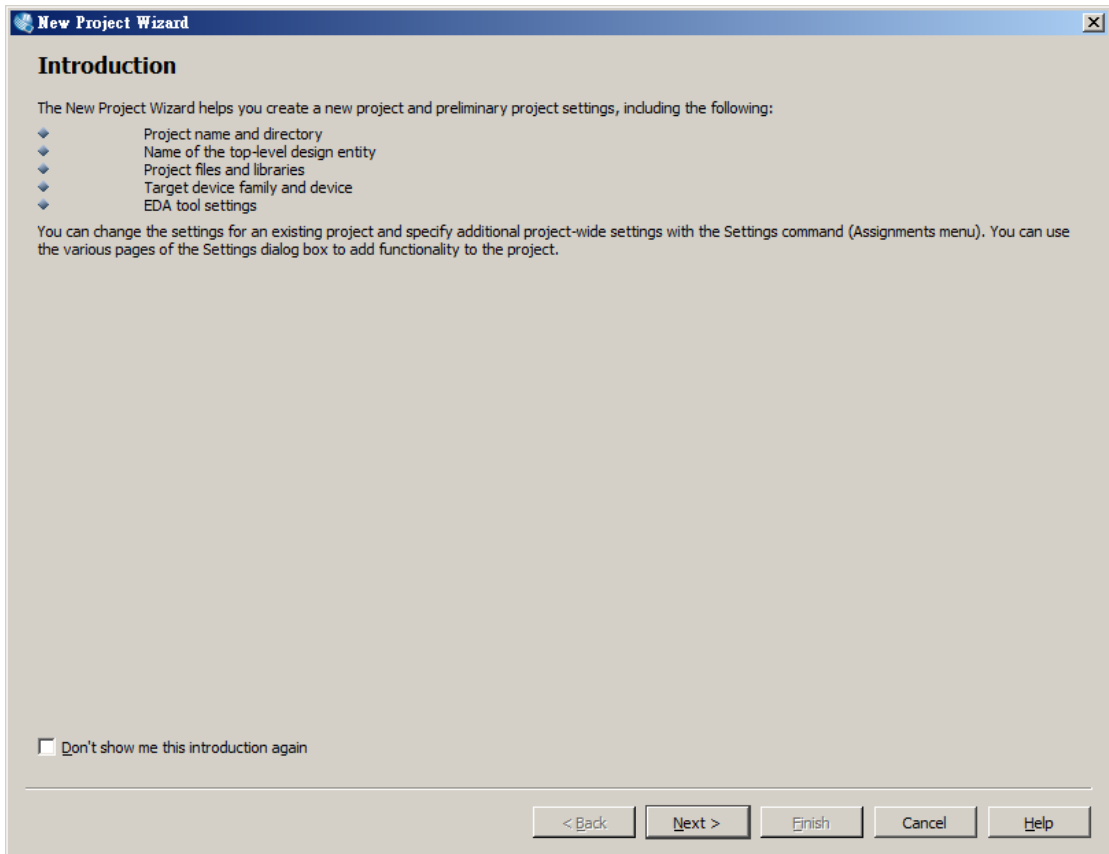
1. 建立 project file folder, 在建立 project 時的注意事項: folder 名稱不可以特殊字元, 空白, 或者中文字 (此範例程式的 folder 在 C:\FPGA_2013_04\SPI_MASTER_SLAVE)
2. 將所需要的檔案 copy 至 project 目錄底下
以這個範例, 需要的檔案有
 - a) SPIMasterFSM.v : SPI Master 的程式
 - a) spi_slave.v : SPI Slave 的程式
 - b) spi_top.v : SPI Master & SPI Slave 的 top module



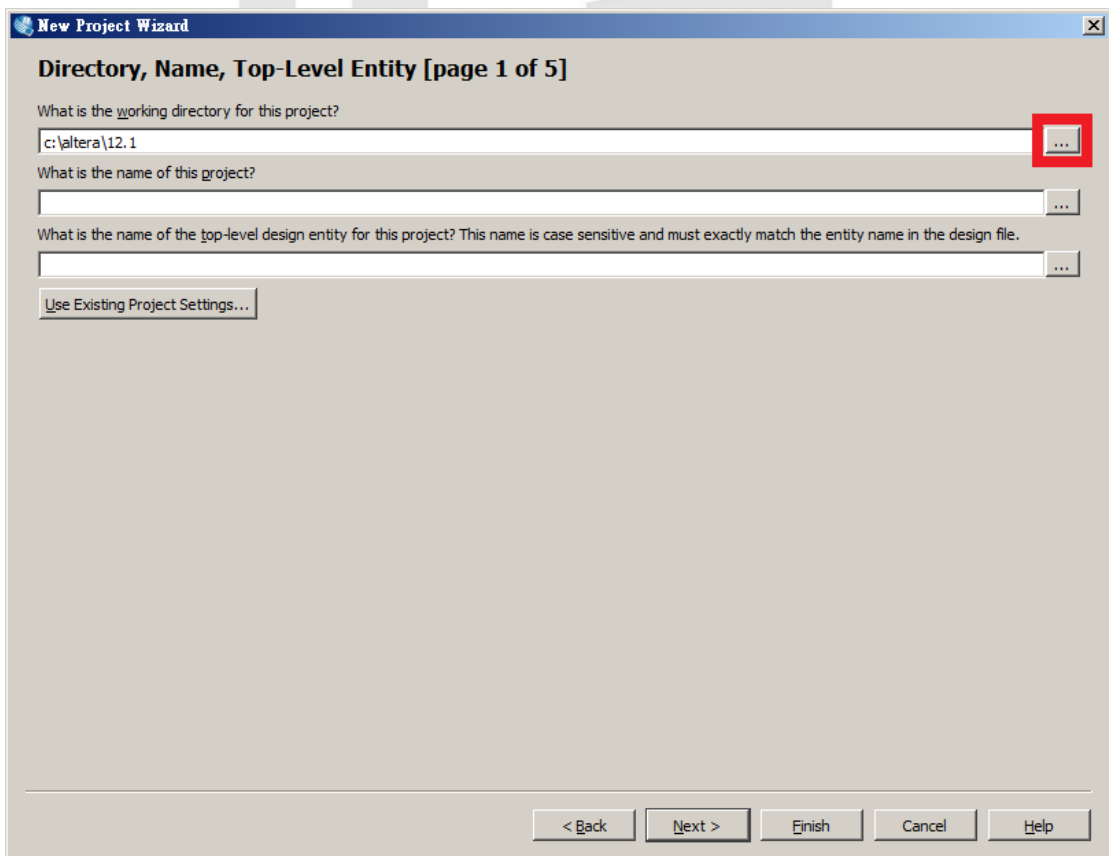
3. 開啟 Quartus II
4. 建立新 project, “File” → “New Project Wizard...”



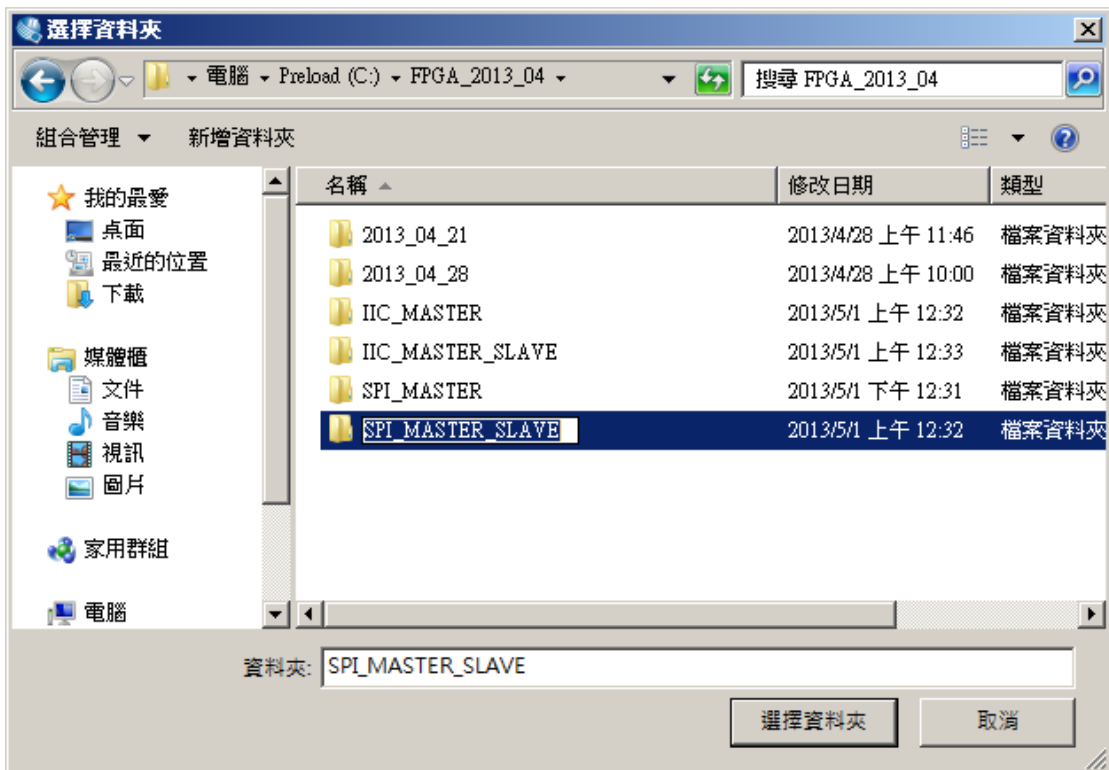
5. New Project Wizard 第一頁是簡介, 請按”Next”



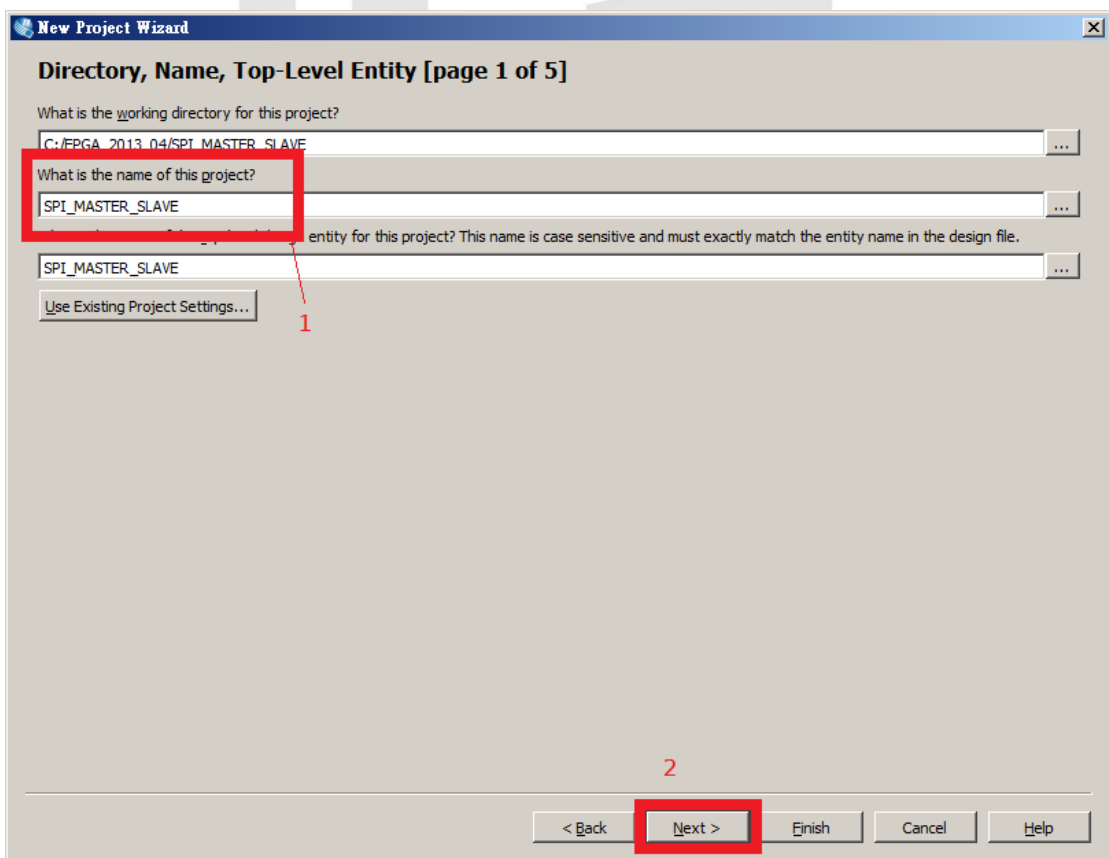
6. 選擇 project 路徑, 按下圖的紅色框住的 "...” 按鈕



7. 選到 C:\FPGA_2013_04\SPI_MASTER_SLAVE 路徑後按 “選擇資料夾”

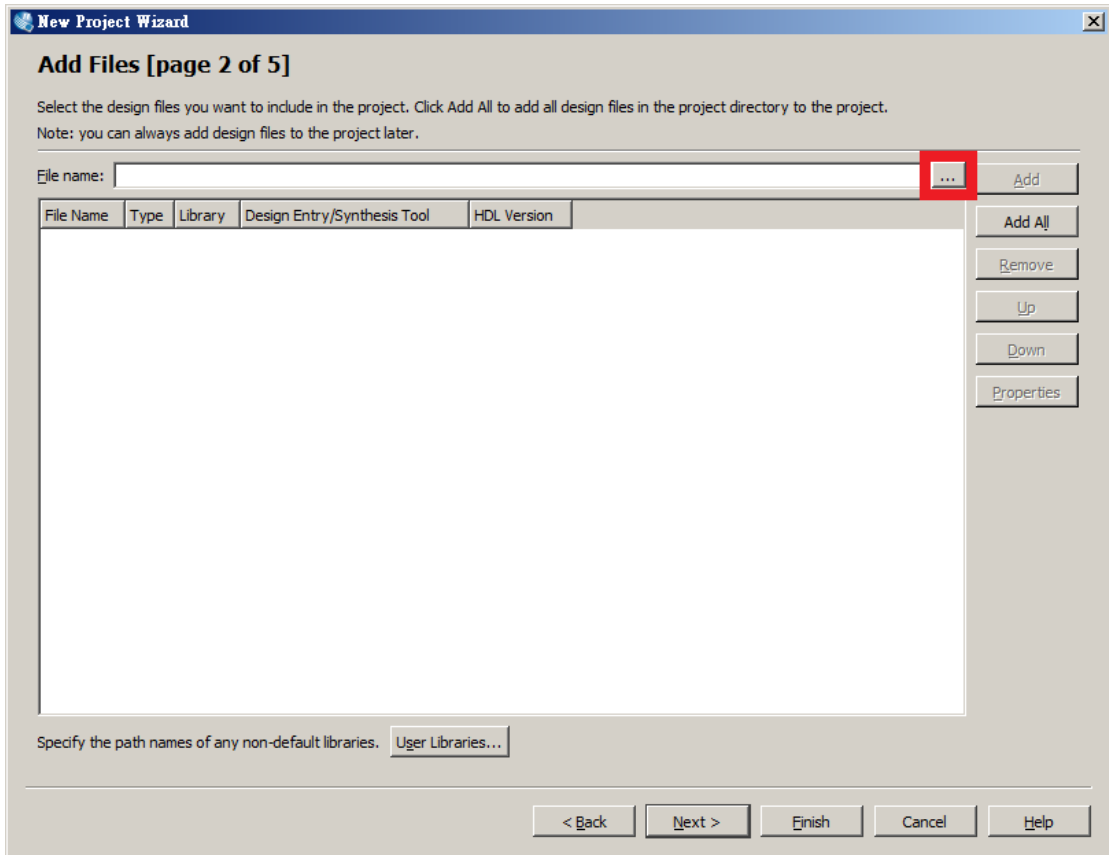


8. 在 “What is the name of this project?” 的欄位內 Key in project name, 在此範例中我將這個 Project 命名為 “SPI_MASTER_SALVE”, 然後按 “Next” 的按鍵

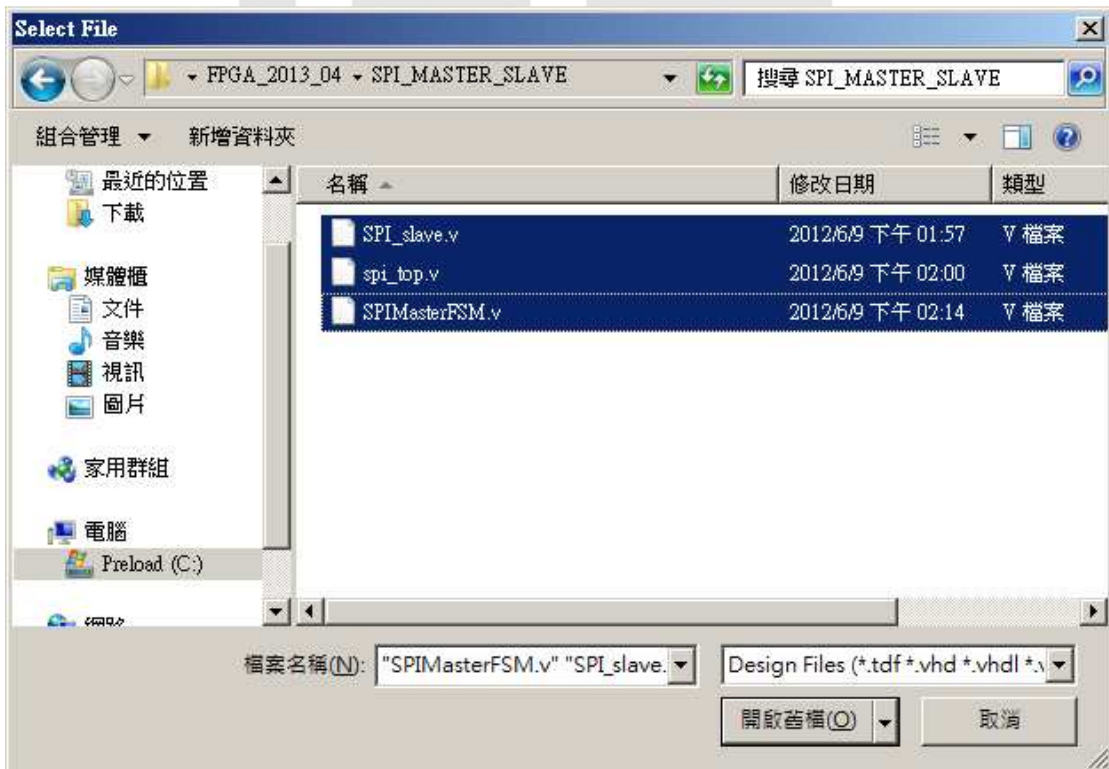


9. 接下來將檔案加入 project, 在 New Project Wizard 的 Add Files 的頁面

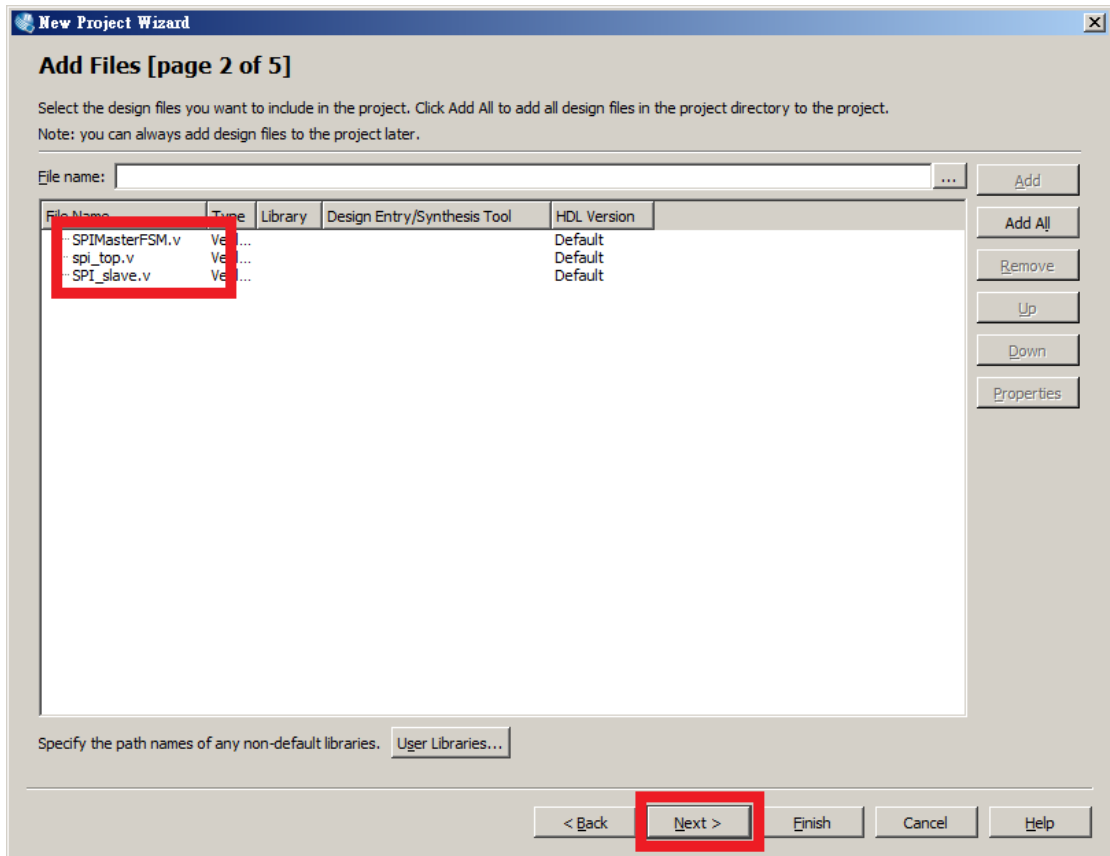
中，在“File name:”的欄位後面找到“...”按一下滑鼠



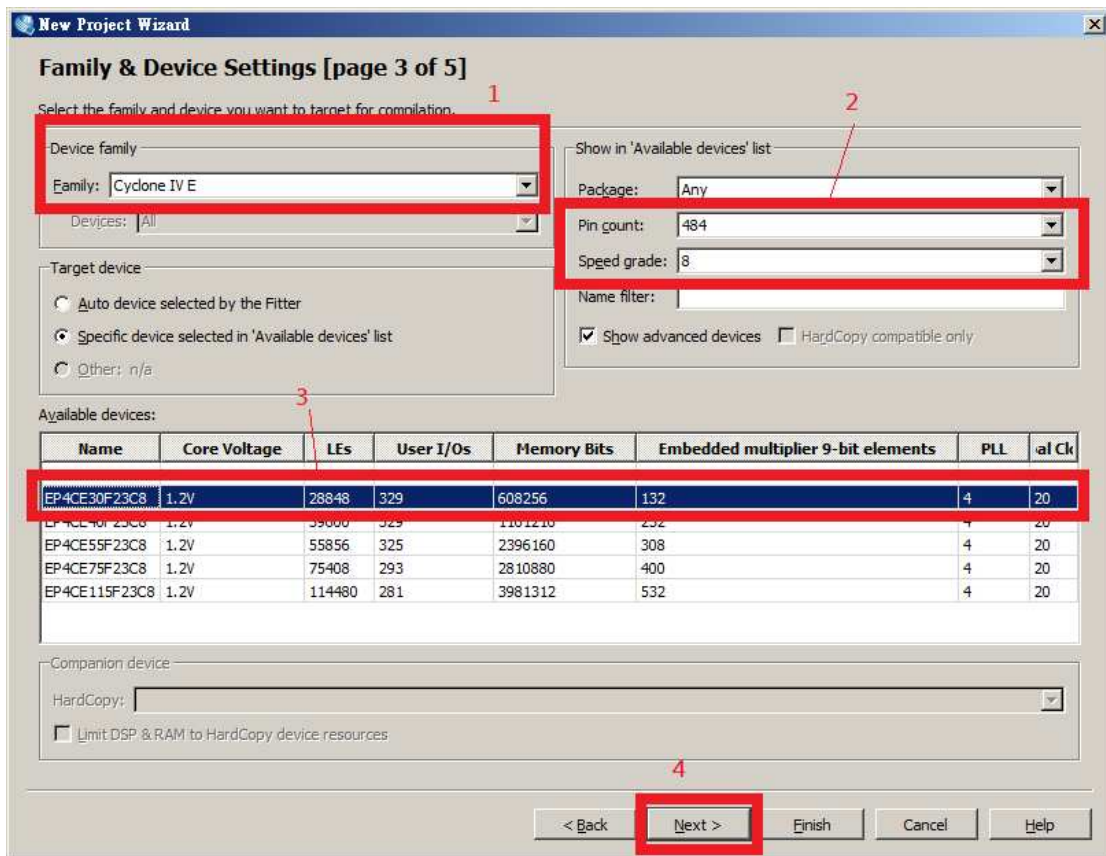
10. 選擇 spi_slave.v, spi_top.v 以及 SPIMasterFSM.v (確認在“檔案名稱”的欄位內有出現這三個檔案)，後按下“開啟舊檔”



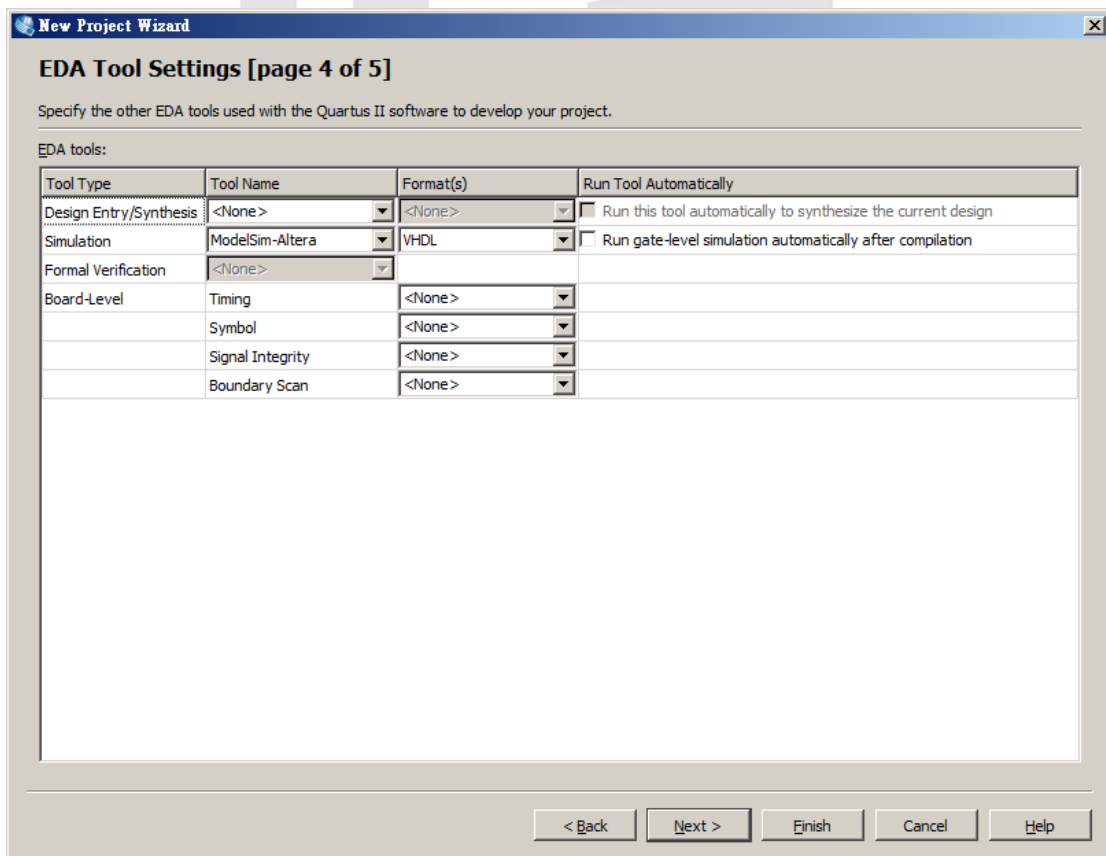
11. 回到 “Add files” 的視窗時，確認在 “File Name” 的欄位不是空的，然後按下 “Next”



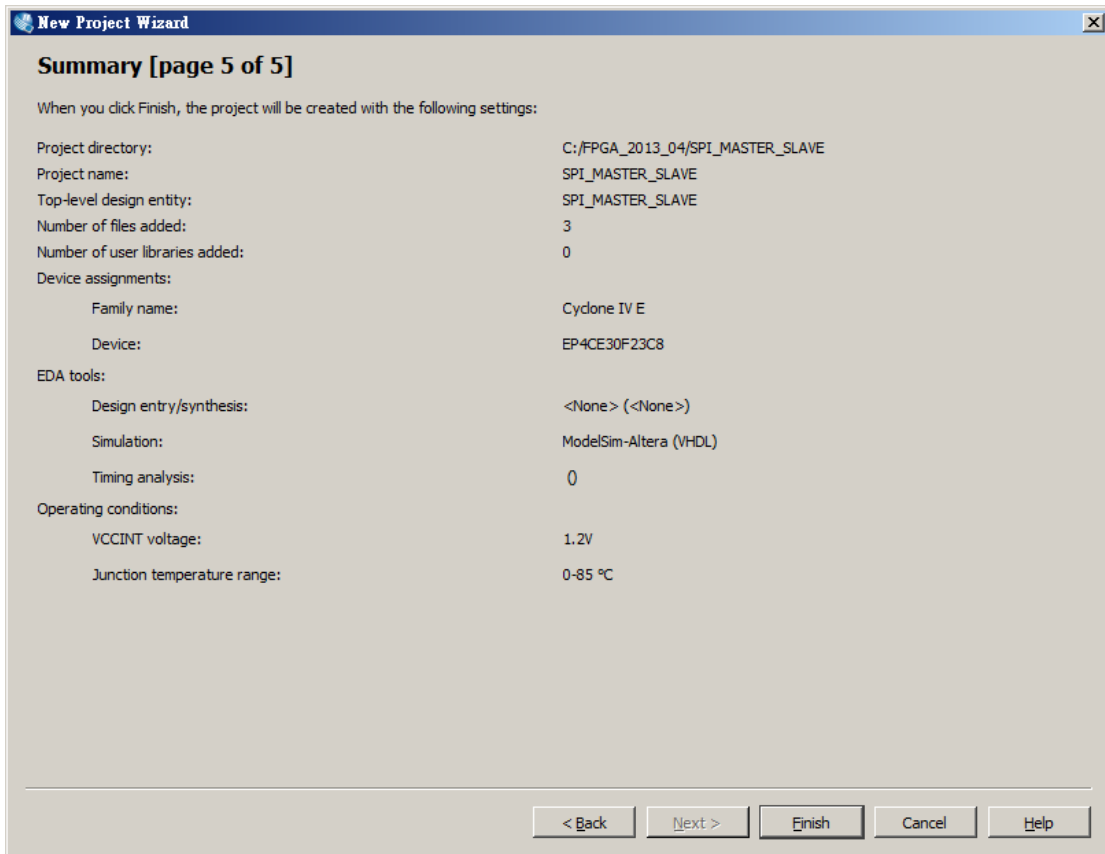
12. 選擇 FPGA 型號，上課用的板子是 EP4CE30F23C8，所以在篩選條件設定如下：
- a) Device Family : Cyclone IV E
 - b) Pin count: 484
 - c) Speed grade : 8
- 然後在 Available devices 的欄位會將符合的 FPGA 型號都列出來，這個時候所列出的項目會少很多，在這個欄位裡選擇 EP4CE30F23C8 後，按 “Next”



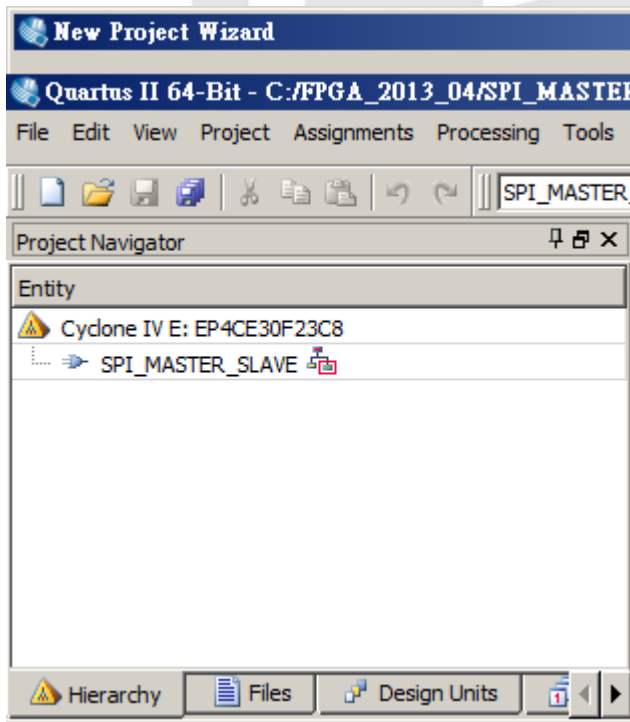
13. 接下來是“EDA Tool Settings”，按下“Next”



14. 接下來是“Summary”，按下“Finish”

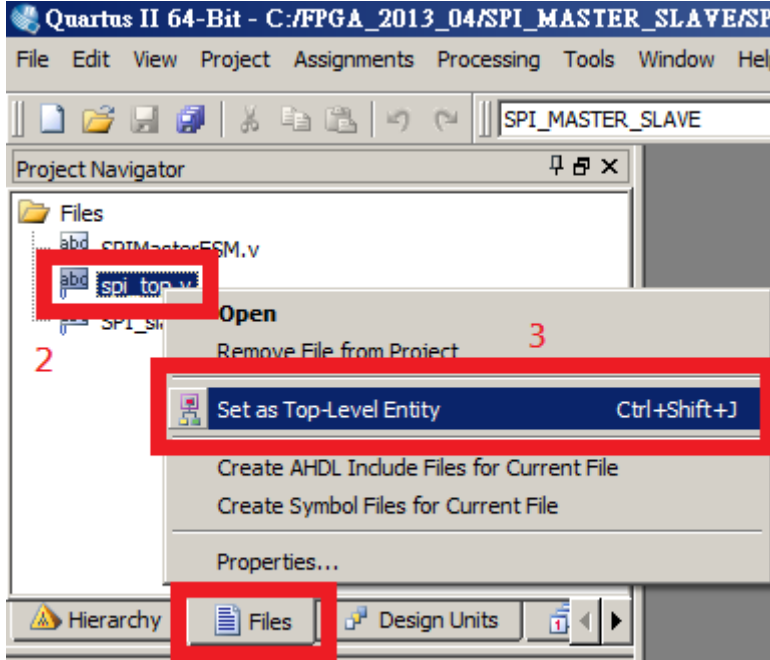


15. 建立好 project 後在 “Project Navigator” 中的 “Entity” → “Hierarchy”, 如下圖所示, 其中 “SPI_MASTER_SLAVE” 是 project name, 但不是真正的 top module.



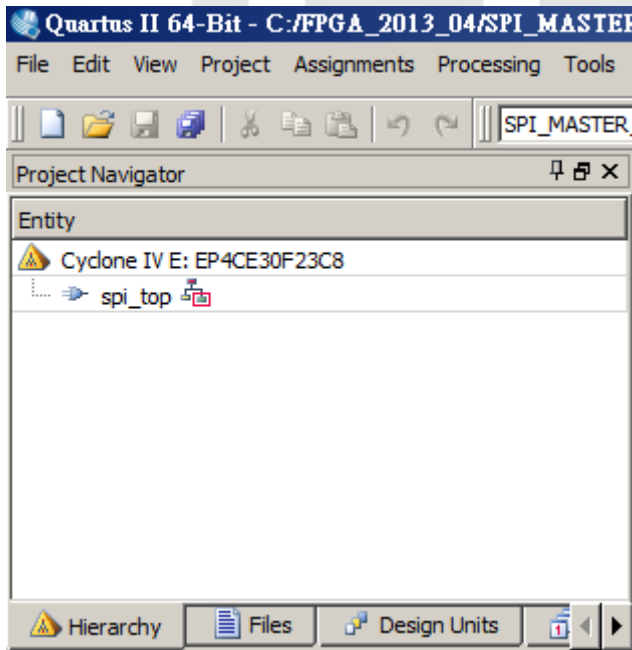
16. 接下來指定 top module.

- a) 首先將 tab 選至 “Files”
- b) 選擇 spi_top.v (使其呈藍底白字)
- c) 在 spi_top.v 上按滑鼠右鍵
- d) 選擇 “Set as Top-Level Entity”



1

17. 再切回 “Hierarchy”, 就可以看到如下圖所示的 spi_top 為 top module.



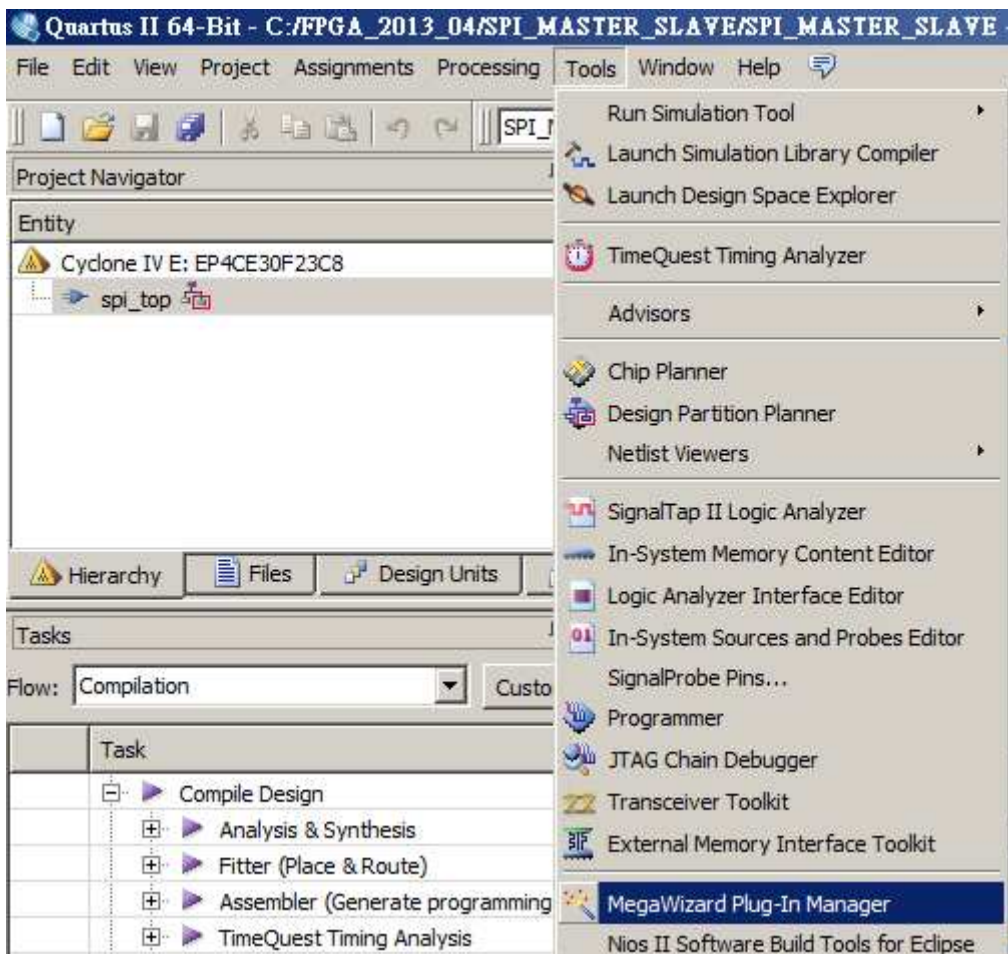
18. 打開 spi_top.v, 將第一行的 `define __SIM 註記成註解

```

1  //`define __SIM
2
3  module spi_top(
4
5      `ifdef __SIM
6      clk,
7      rst,
8      `endif
9      done,
10     spiSCK,
11     spiMISO,
12     spiMOSI,
13     spiCSn

```

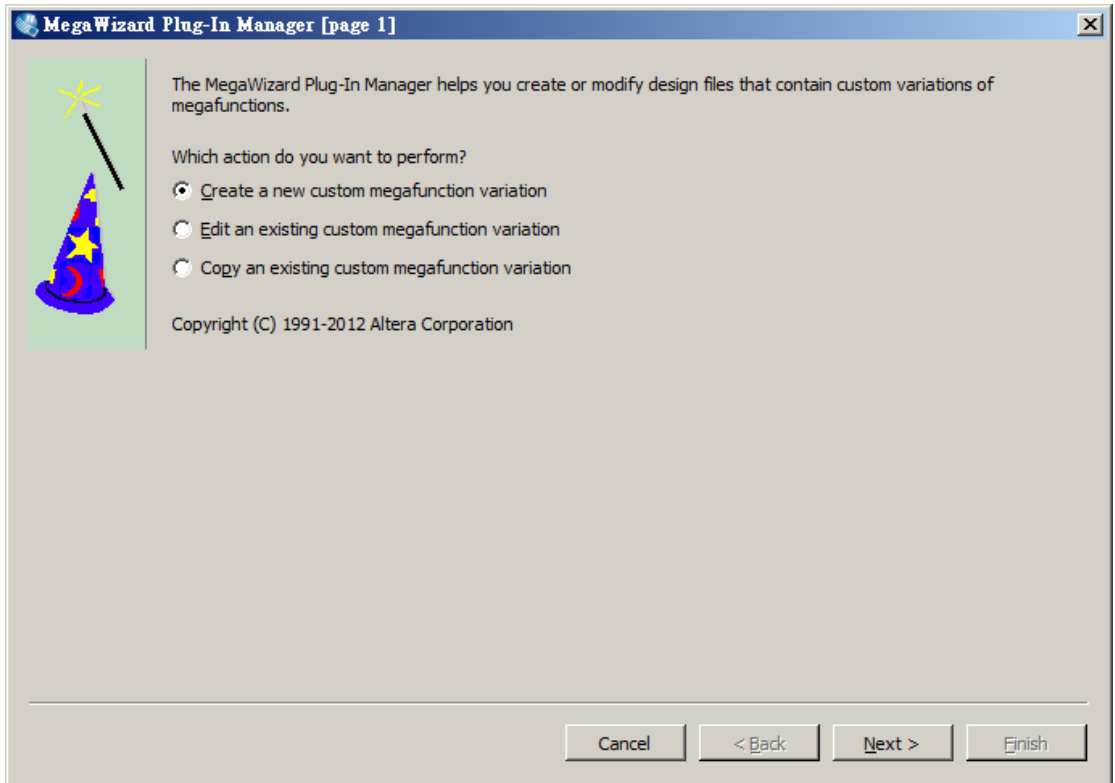
19. 接下來, 要做的是建立 PLL 的 IP
„Tool“ → „MegaWizard Plug-In Manager“



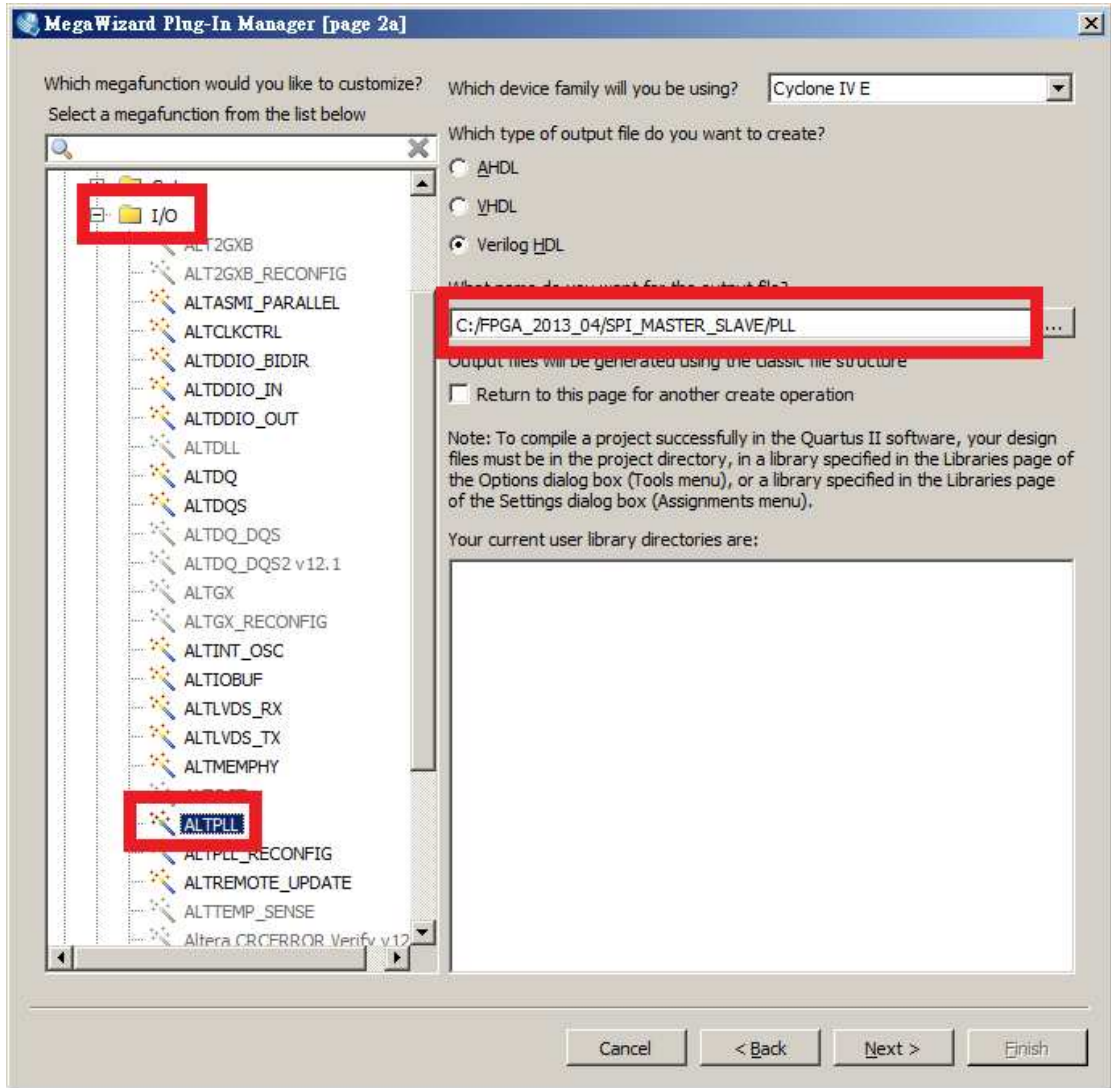
20. MegaWizard Plug-In Manager, 有三個選項
- a) Create a new custom megafunction variation
建立新的 megafunction
 - b) Edit an existing meagafunction variation
修改已有的 megafunction 的參數
 - c) Copy an existing custom megafunction variation
從另一個 project 裡 copy 一個已有的 megafunction 至目前的

project.

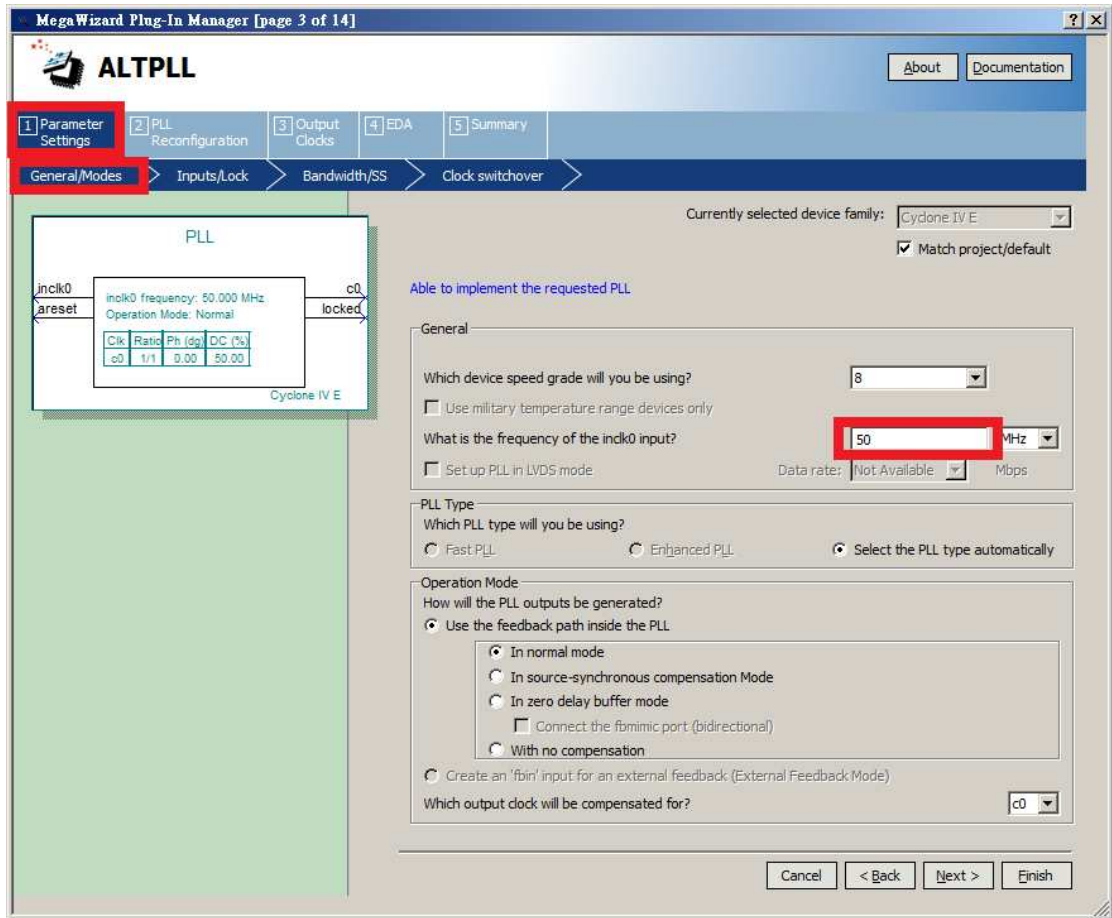
在這裡，因為是要建立新的 IP，所以要選擇的是 “Create a new custom megafunction variation”，然後，按 “Next”



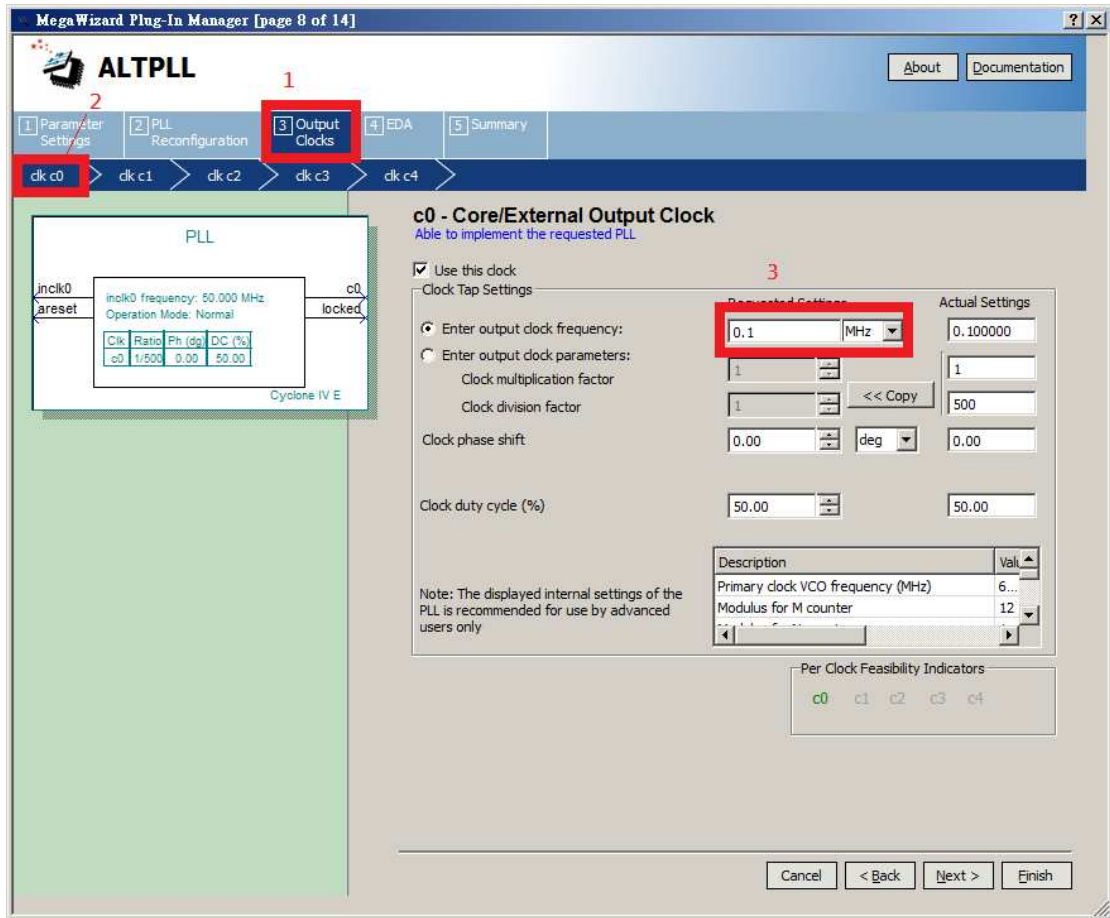
21. 在欄位 „What name do you want for the output file?“ 裡輸入 megafunction 的名字 → “C:/FPGA_2013_04/SPI_MASTER_SLAVE/PLL”
左邊的 IP 分類 „IO“ 選擇 „ALTPLL“，然後，按 „Next“



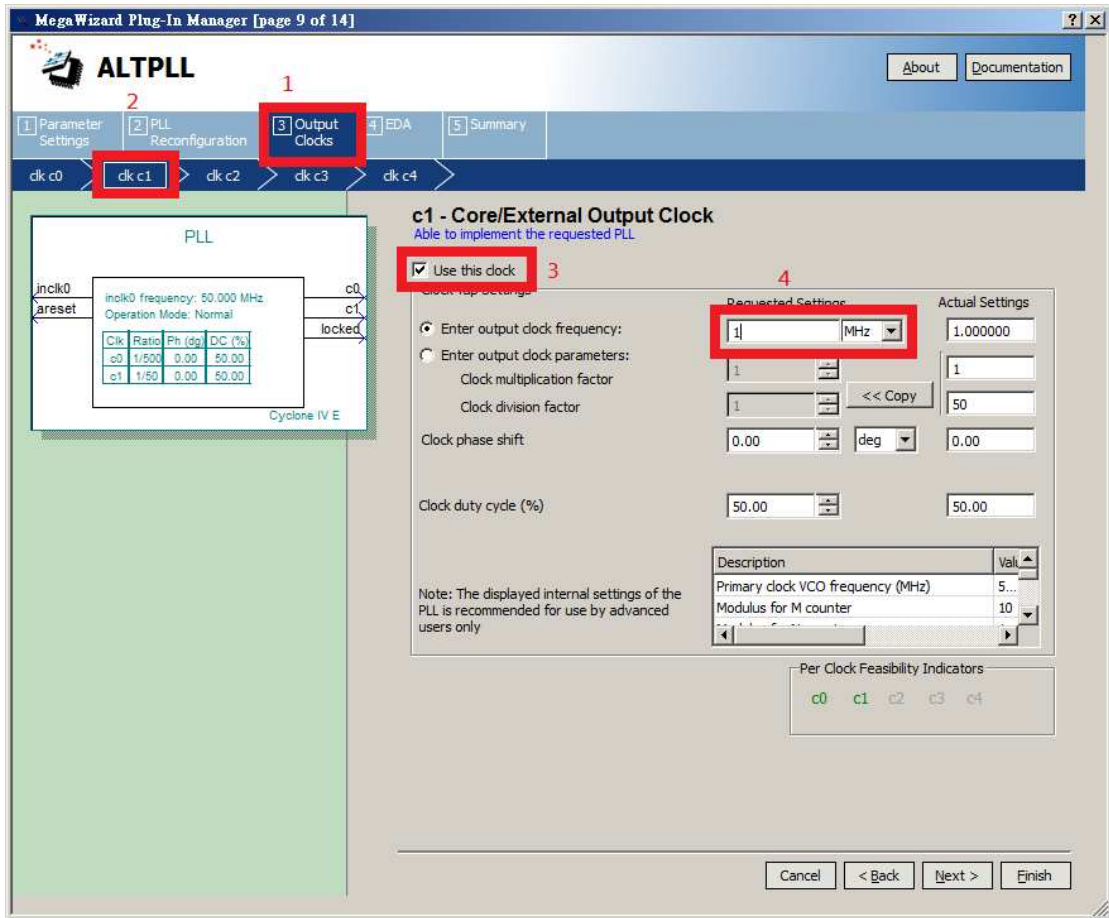
22. 板子上的 osc 頻率是 50MHz, 所以在 ALTPLL 的設定頁 „Parameter Settings“ → „General/Modes“ 的欄位 „What is the frequency of the inclk0 input? „ 原本 default 值為 100 MHz 改成 50MHz.



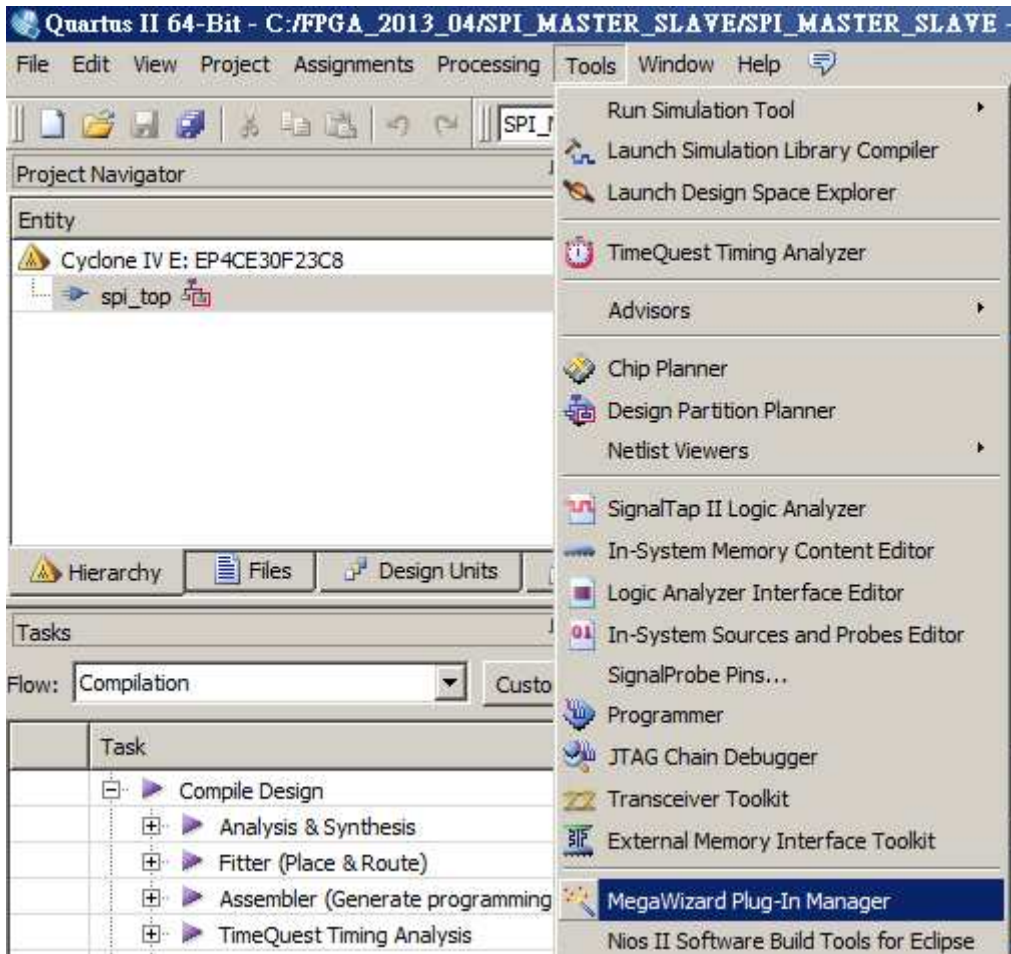
23. 在 spi_top.v 裡, 安排 c0 的 clock 為 spi_master 的 operation clock, 所以, 在 ALTPLL 的設頁 „Output Clocks“ → „clk c0“ 的 „Enter output clock frequency“ 設定成 100KHz.



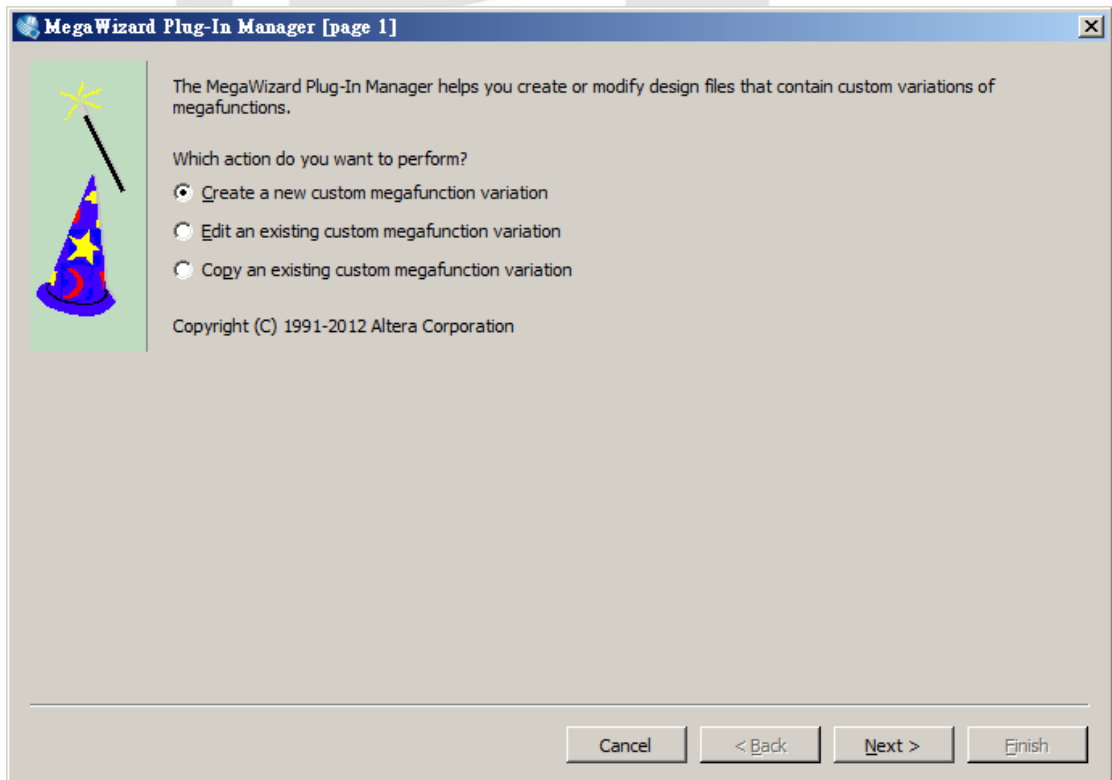
24. 在 spi_top.v 裡, 安排 c1 的 clock 為 spi_slave 的 operation clock, 所以, 在 ALTPLL 的設頁 „Output Clocks“ → „clk c1“ 的 „Enter output clock frequency“ 設定成 1MHz. 然後按 „Finish“ 到此, PLL 的 megafunction 就建立好了



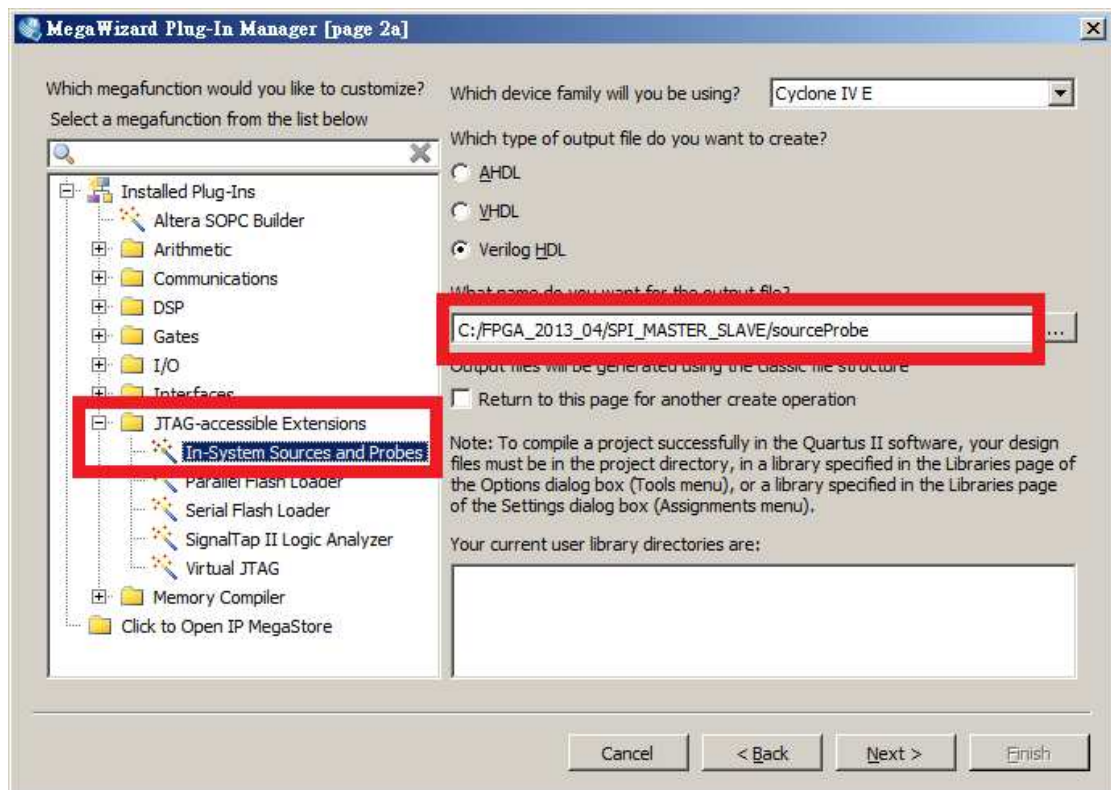
25. 接下來, 要做的是建立是用來做為虛擬 IO 的 debug tool „SourceProbe“ „Tool“ → „MegaWizard Plug-In Manager“



26. MegaWizard Plug-In Manager,選擇的是 “Create a new custom megafunction variation”, 然後, 按 “Next”



27. 在欄位 „What name do you want for the output file?“ 裡輸入 megafunction 的名字 → “C:/FPGA_2013_04/SPI_MASTER_SLAVE/sourceProbe”
 左邊的 IP 分類 „JTAG-accessible Extensions“ 選擇 „In-System Sources and Probes“, 然後, 按 „Next“

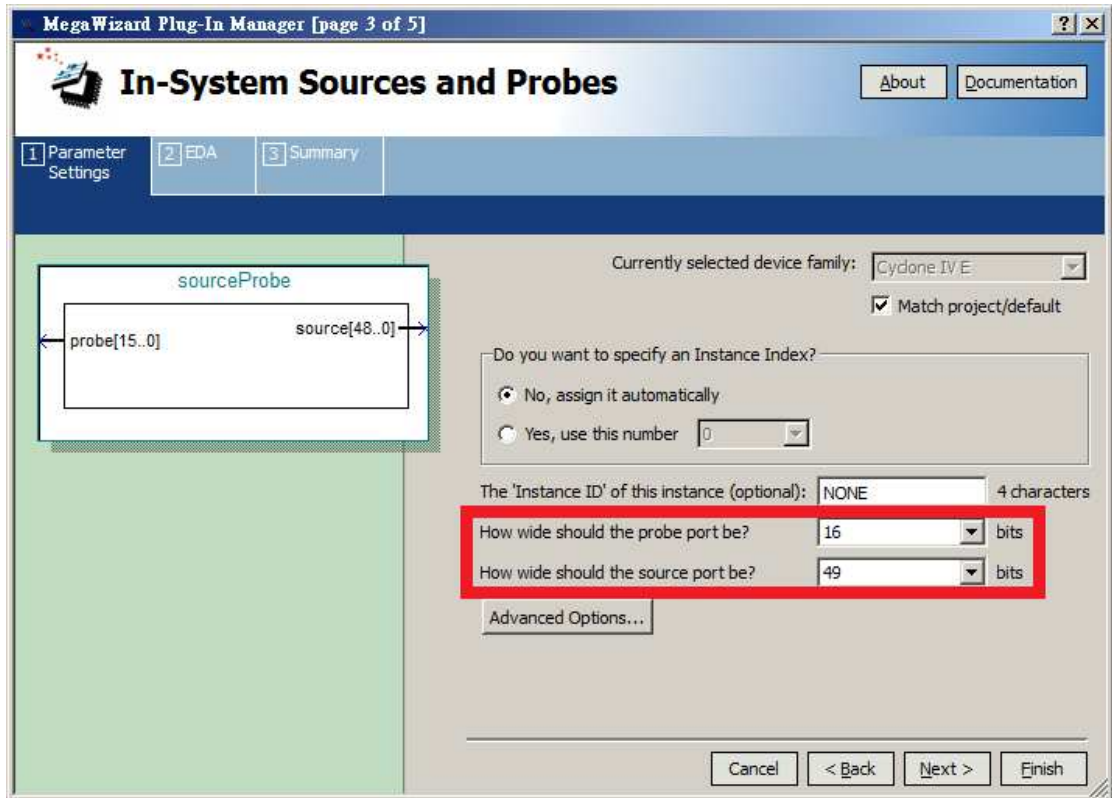


28. 在 spi_top.v 中, 設定的 source 是 49 位元, probe 是 16 位元

```

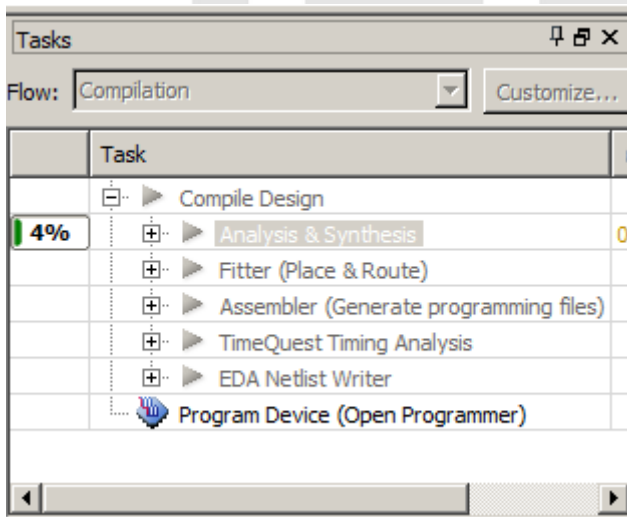
wire    [15:0]    probe;
wire    [48:0]    source;
sourceProbe SP( .probe    (probe[15:0]),
                .source    (source[48:0])
                );
assign  spiMasterEn = source[48];
assign  MasterRwN   = source[47];
assign  MasterSlaveAddr[14:0] = source[14:0];
assign  MasterRegAddr[15:0] = source[30:15];
assign  MasterDataOut[15:0] = source[46:31];
assign  probe[15:0] = MasterDataIn[15:0];
    
```

所以在 „Parameter Settings“ 的欄位 “How wide should the probe port be?” 選擇 16 bits. 另一個欄位 “How wide should the source port be?” 選擇 49 bits, 然後按 “Finish“



29. 接下來，要做的是 pin assignment，但是，因為到目前為止，都還未做 synthesis，因此，QuartusII 不會認出到底有哪些 IO，所以，現在先做一次 synthesis。執行 synthesis 的方式有以下四種。

a) 在“Task”裡直接在“Analysis & Synthesis”按兩下



b) 在 QuartusII 上方的 Tool bar 按如下的圖紅色線框住的 icon

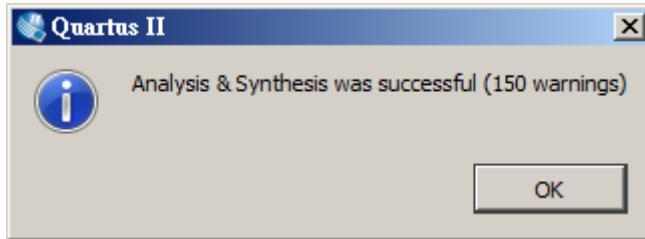


c) 在“Processing” → “Start” → “Start Analysis & Synthesis”



d) 直接按快速鍵 Ctrl+K

30. synthesis 完成後會出彈出如下圖的視窗，按 “OK” 關閉視窗



31. 接著就可以做 pin assignment, 選擇 “Assignments” → “Pin Planner”



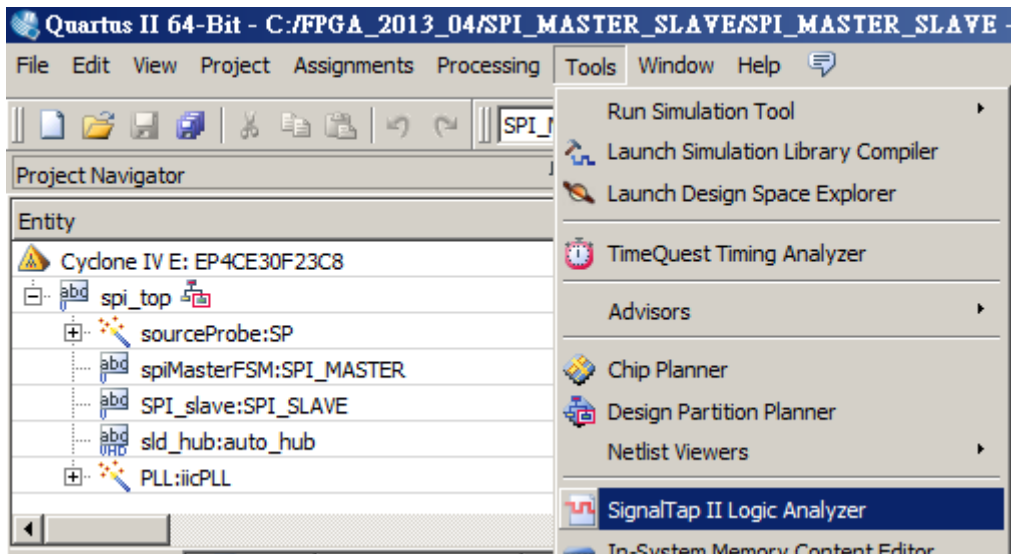
32. clock : PIN_G1, rst : PIN_T21, I/O standard : 3.3V LVCMOS.

QuartusII 的 pin assignment 一經設定即儲存，所以，設定完後，就可以關閉 pin assignment 的視窗。

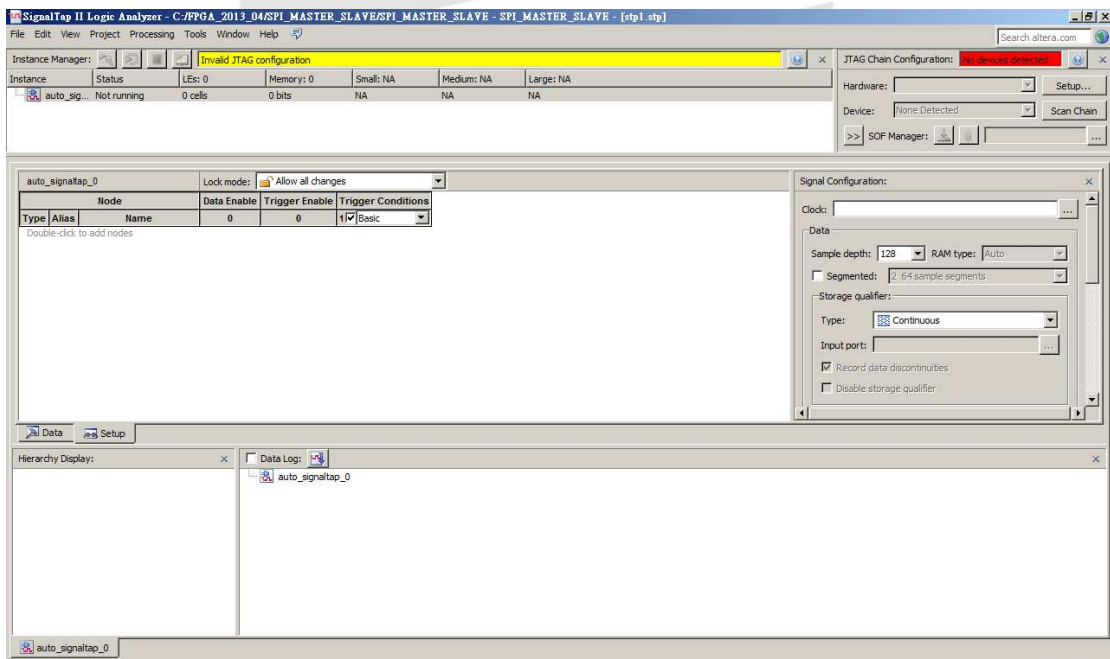
| Node Name | Direction | Location | I/O Bank | VREF Group | I/O Standard | Reserved | Current Strength | Slew Rate | Differential Pair |
|---------------------|-----------|----------|----------|------------|-----------------|----------|------------------|-------------|-------------------|
| altera_reserved_tck | Input | | | | 2.5 V (default) | | 8mA (default) | | |
| altera_reserved_tdi | Input | | | | 2.5 V (default) | | 8mA (default) | | |
| altera_reserved_tdo | Output | | | | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| altera_reserved_tms | Input | | | | 2.5 V (default) | | 8mA (default) | | |
| clk | Input | PIN_G1 | 1 | B1_N3 | 3.3-V LVCMOS | | 2mA (default) | | |
| done | Output | | | | 3.3-V LVCMOS | | 2mA (default) | 2 (default) | |
| rst | Input | PIN_T21 | 5 | B5_N0 | 3.3-V LVCMOS | | 2mA (default) | | |
| spiCSn | Output | | | | 3.3-V LVCMOS | | 2mA (default) | 2 (default) | |
| spiMISO | Output | | | | 3.3-V LVCMOS | | 2mA (default) | 2 (default) | |
| spiMOSI | Output | | | | 3.3-V LVCMOS | | 2mA (default) | 2 (default) | |
| spiSCK | Output | | | | 3.3-V LVCMOS | | 2mA (default) | 2 (default) | |
| <<new node>> | | | | | | | | | |

33. 接下來，就是設定 Signal Tab, 將要觀察的 SPI interface 的信號利用 Signal Tab 觀察是否動作正常。

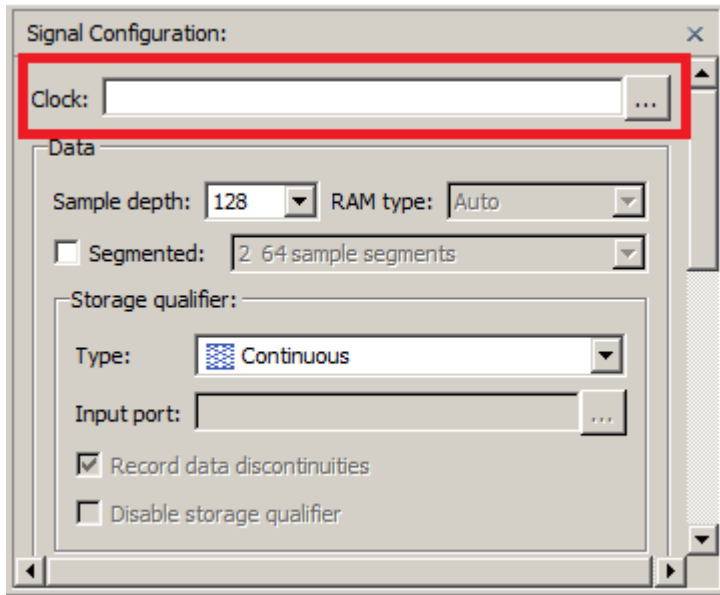
“Tools” → “SignalTab II Logic Analyzer”



34. SignalTab 是一套由 Altera 提供的 tool, 其功能類似於邏輯分析儀, 但是, 是利用 FPGA 的 LE 以及 BlockRAM 來做條件以及資料的儲存. 而存在 BlockRAM 的資料只需要透過 Altera USB-Blaster (download cable) 即可。



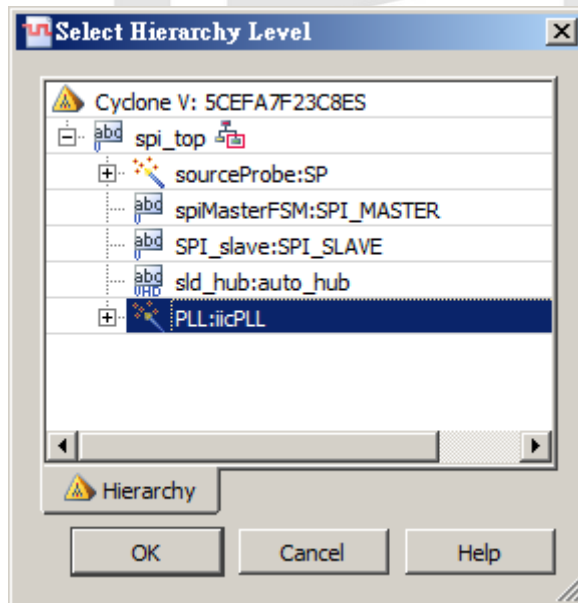
35. 首先, 要指定 SignalTab 的工作頻率. 按下在 “Signal Configuration” 的 “Clock” 欄位後的按鍵 “...”



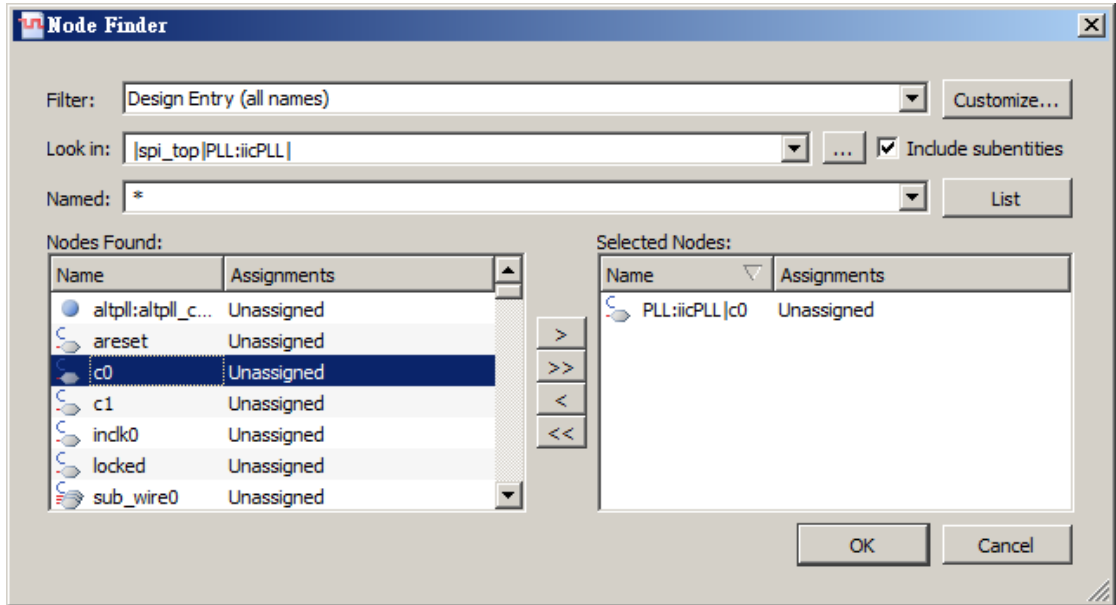
36. 在“Node Finder”的視窗裡

- a) “Filter” 欄位中選擇 Design Entry (all names)
- b) “Lock in” 欄位中選擇 spi_top|PLL

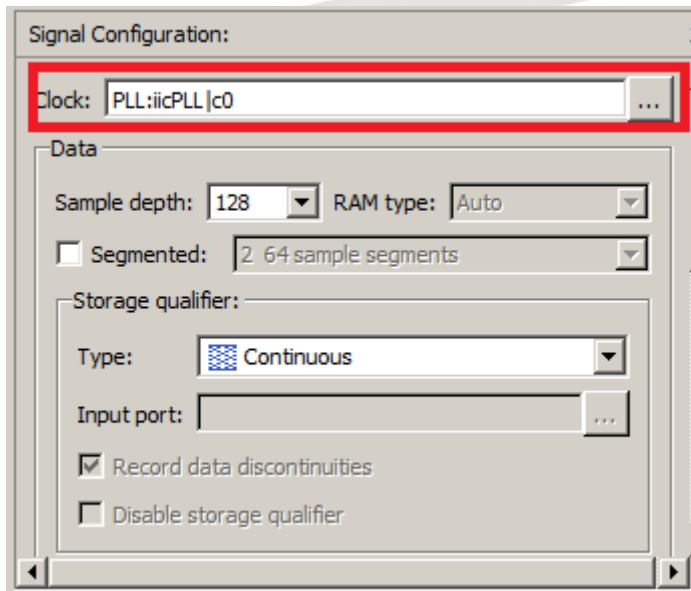
按下“Lock in”欄位後的“...”按鈕，會有“Select Hierarchy Level”中選擇 PLL。如下圖所示。然後按“OK”按鈕。



- c) 按“List” 按鈕
- d) 在“Nodes Found” 按兩下“c0”
- e) 確認 c0 有出現在“Selected Nodes”
- f) 按“OK”

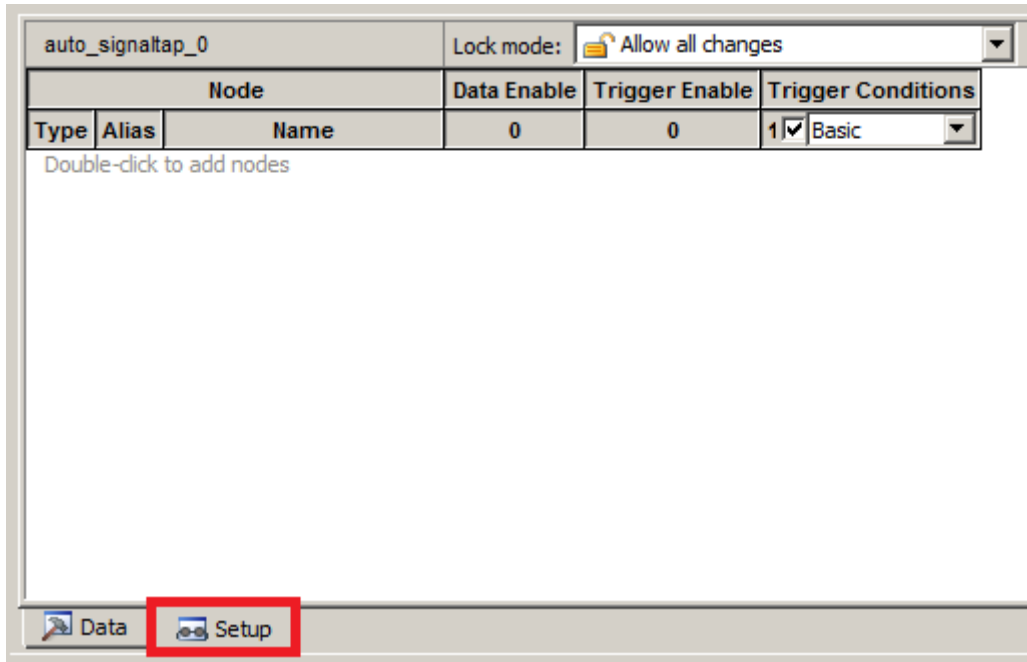


37. 確認 “Signal Configuration” 的 “Clock” 是有指至 “PLL|c0”



38. 接下來就是加入要觀察的 SPI 信號

在 auto_signaltab_0, 選擇 “Setup” 的頁面, 並且在空白處按滑鼠兩下.
(如下圖所示, 有一條灰色的訊息提示 : Double-click to add nodes.)

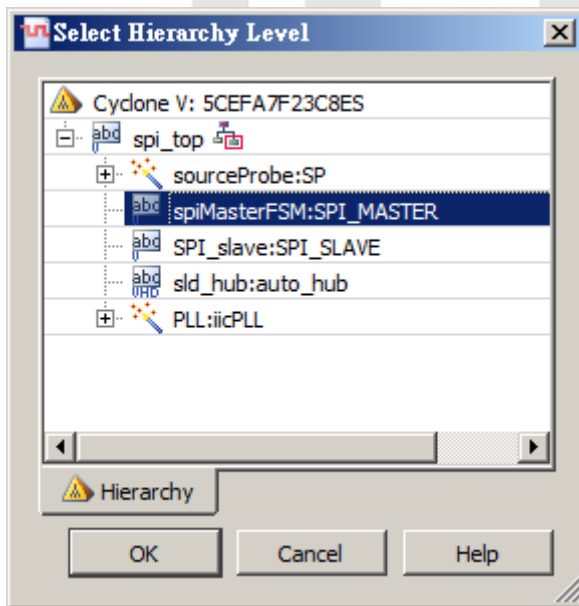


39. 先加入 SPI MASTER 的信號，在 “Node Finder” 的視窗裡

a) “Filter” 欄位中選擇 Design Entry (all names)

b) “Lock in” 欄位中選擇 spi_top|spiMasterFSM

按下 “Lock in” 欄位後的 “...” 按鈕，會有 “Select Hierarchy Level” 中選擇 spiMasterFSM. 如下圖所示. 然後按 “OK” 按鈕.



c) “Named” key in “spi*” 做為篩選條件

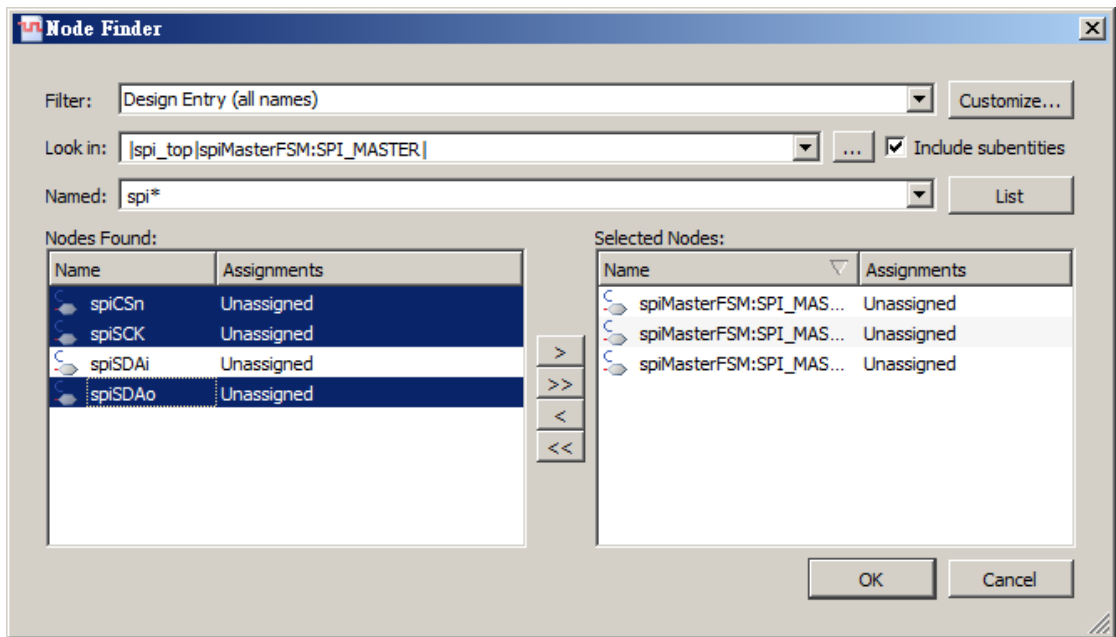
d) 按 “List” 按鈕

e) 在 “Nodes Found” 選 “spiCSn”, “spiSCK”, 以及 “spiSDAo”

f) 按 “>” 按鈕

g) 確認 “spiCSn”, “spiSCK”, 以及 “spiSDAo” 有出現在 “Selected Nodes”

h) 按 “OK”

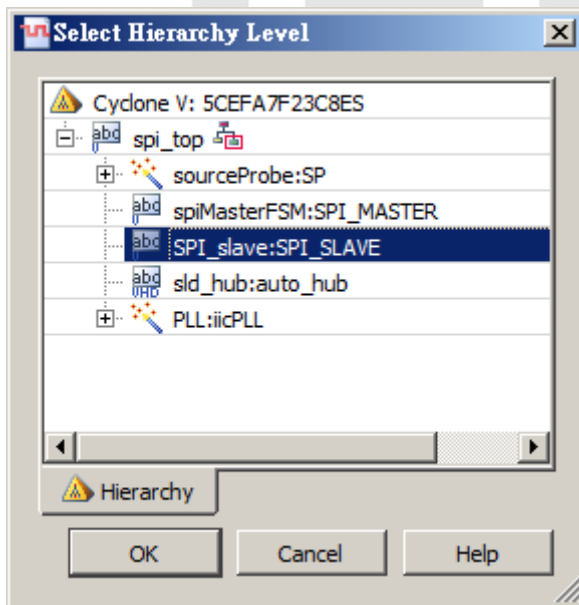


40. 加入 SPI SLAVE 的信號，在 “Node Finder” 的視窗裡

a) “Filter” 欄位中選擇 Design Entry (all names)

b) “Lock in” 欄位中選擇 spi_top|SPI_slave

按下 “Lock in” 欄位後的 “...” 按鍵，會有 “Select Hierarchy Level” 中選擇 SPI_slave. 如下圖所示. 然後按 “OK” 按鍵.

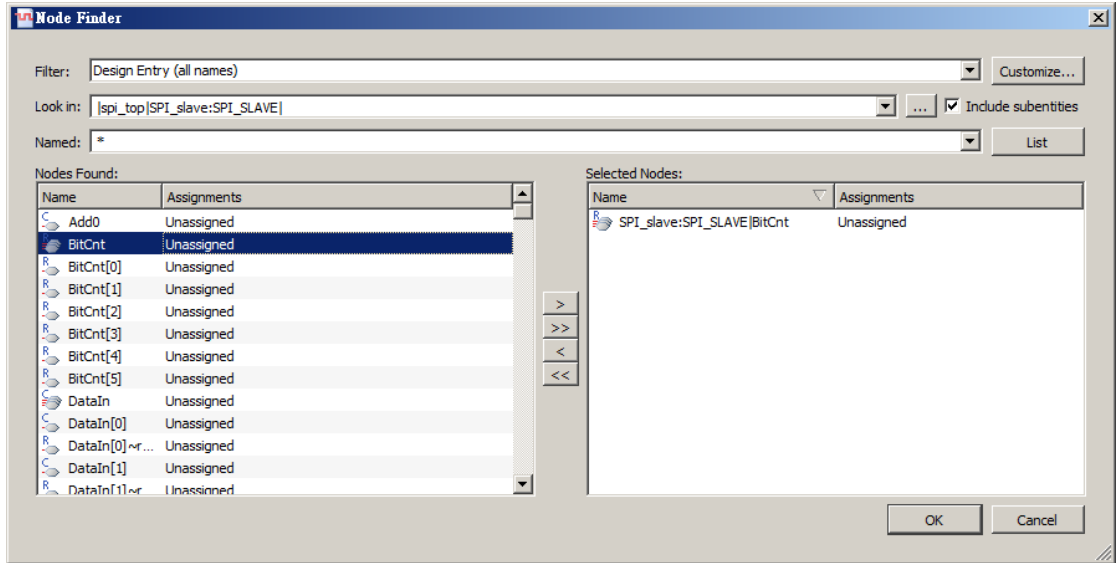


c) 按 “List” 按鍵

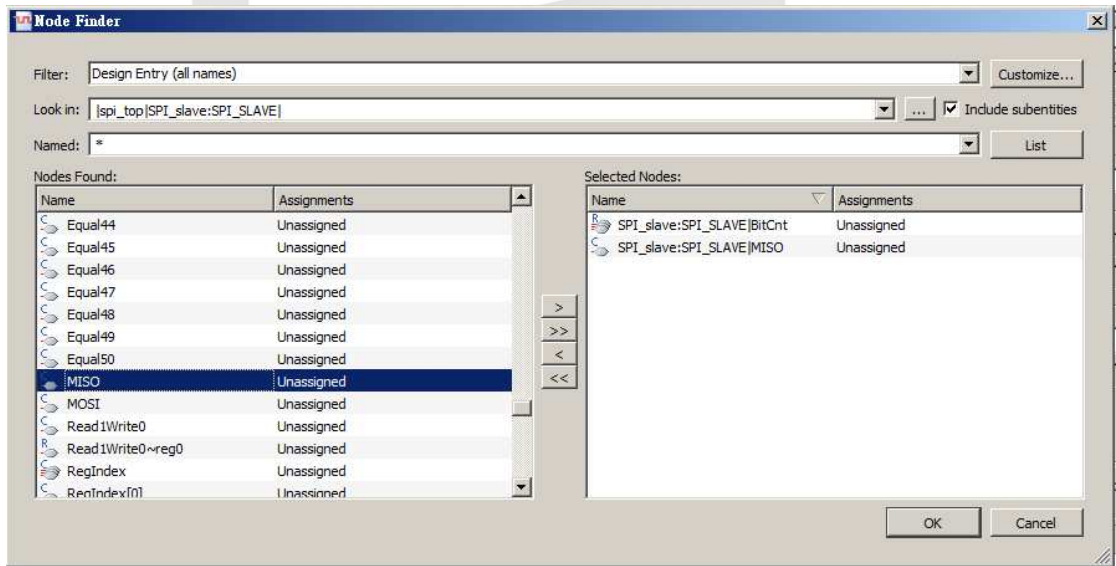
d) 在 “Nodes Found” 選 “BitCnt”

e) 按 “>” 按鍵

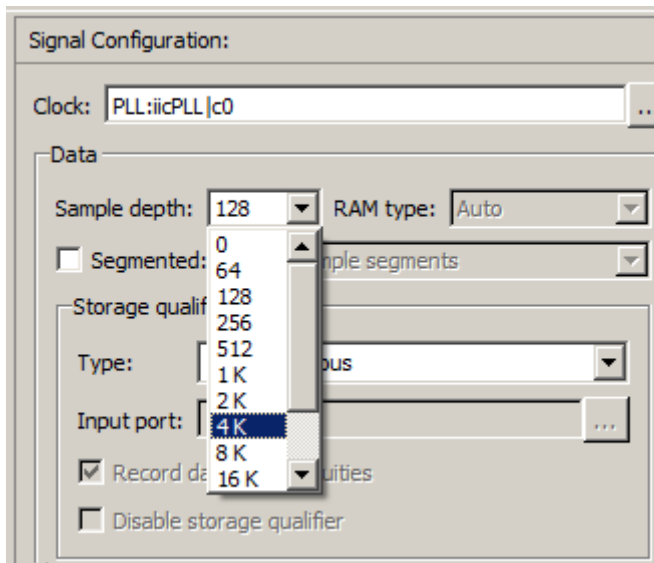
f) 確認 “BitCnt” 有出現在 “Selected Nodes”



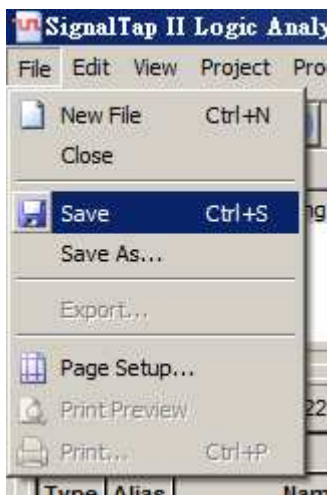
- g) 在 “Nodes Found” 選 “MISO”
- h) 按 “>” 按鍵
- i) 確認 “BitCnt” 有出現在 “Selected Nodes”
- j) 按 “OK”



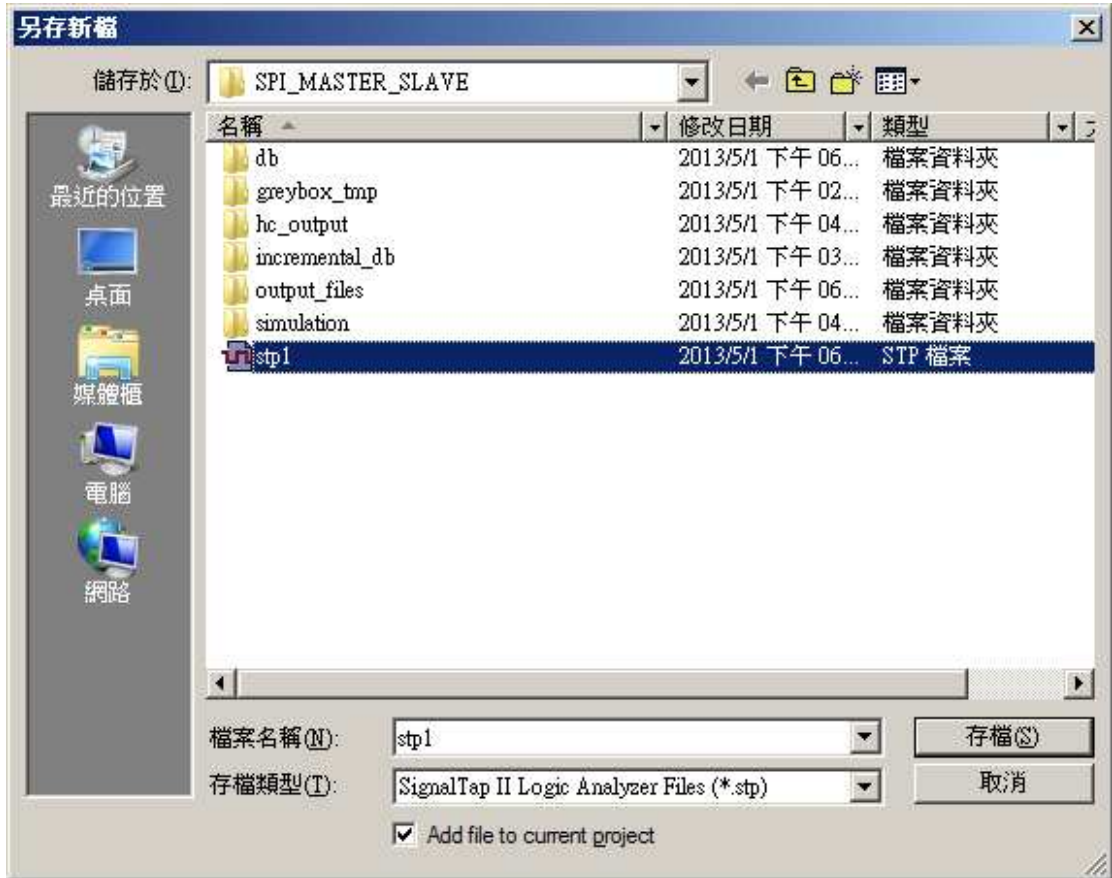
41. 設定想要看的資料數量. “Signal Configuration” → “Sample depth”



42. 最後，將設定好的 SignalTab 存檔。“File” → “Save”

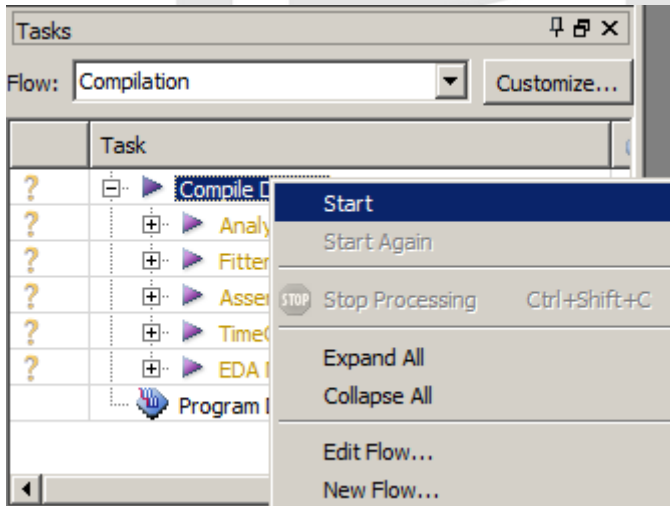


43. 存檔通常都會使用 default 的檔名，且路徑與 project file 存在同一個 folder 中

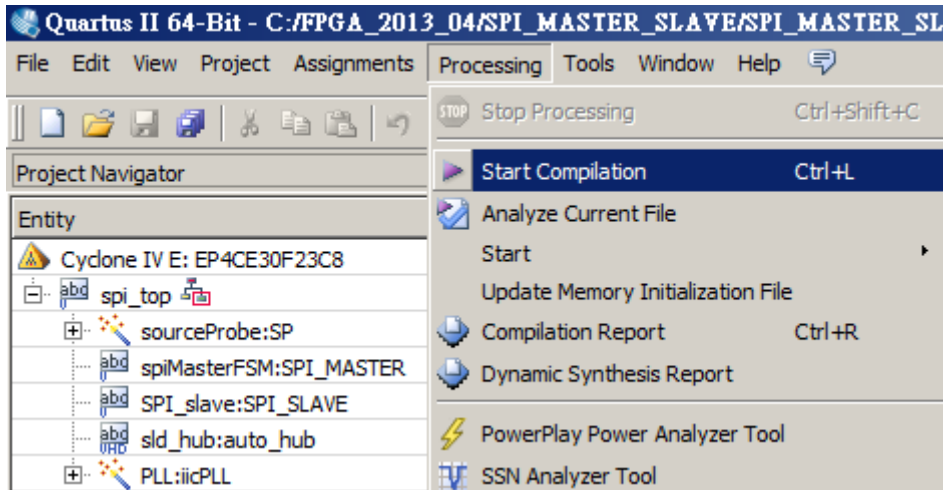


44. 在設定完 SignalTab 後，就可以開始做 all compiler 的動作，而做 all compiler 的方式有下列四種方式

a) 在 “Tasks” 裡在 “Compile Design” 按右鍵，選 “Start”



b) 在 “Processing” → “Start Compilation”

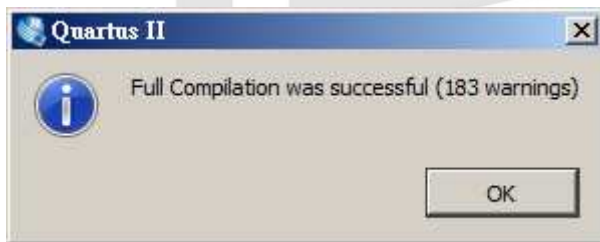


c) 在 QuartusII 上方的 Tool bar 按如下的圖紅色線框住的 icon

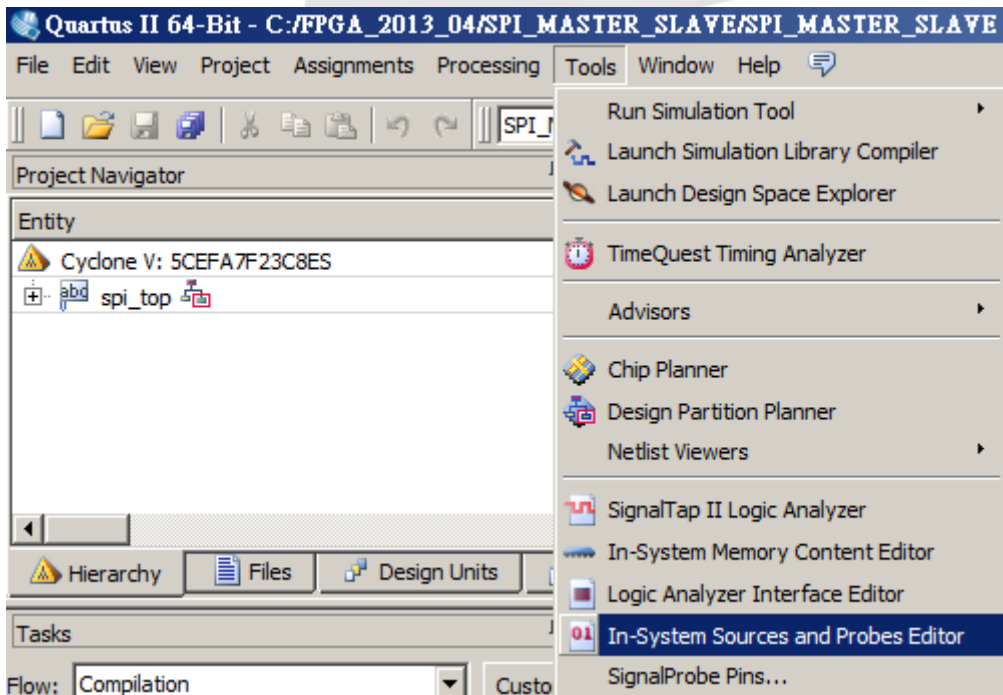


d) 快速鍵 Ctrl + L

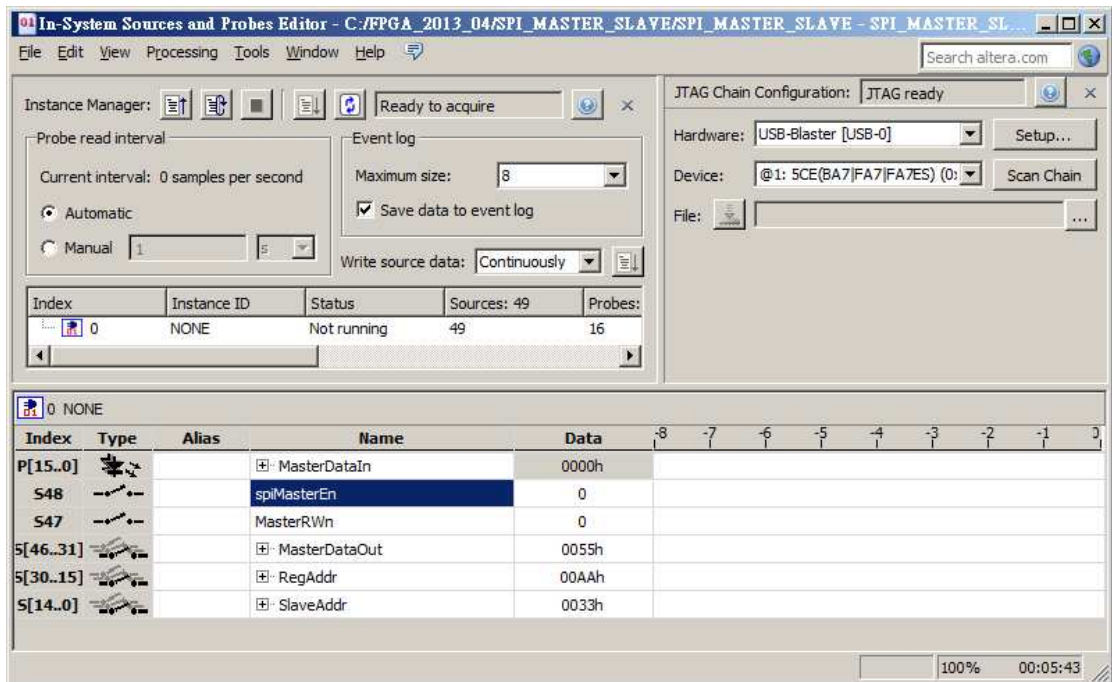
45. compiler all 完成後會出彈出如下圖的視窗，按“OK”關閉視窗



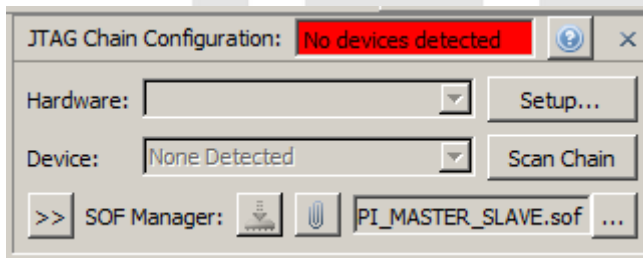
46. 開啟 SourceProbe 視窗. “Tools” → “In-System Sources and Probes Editor”



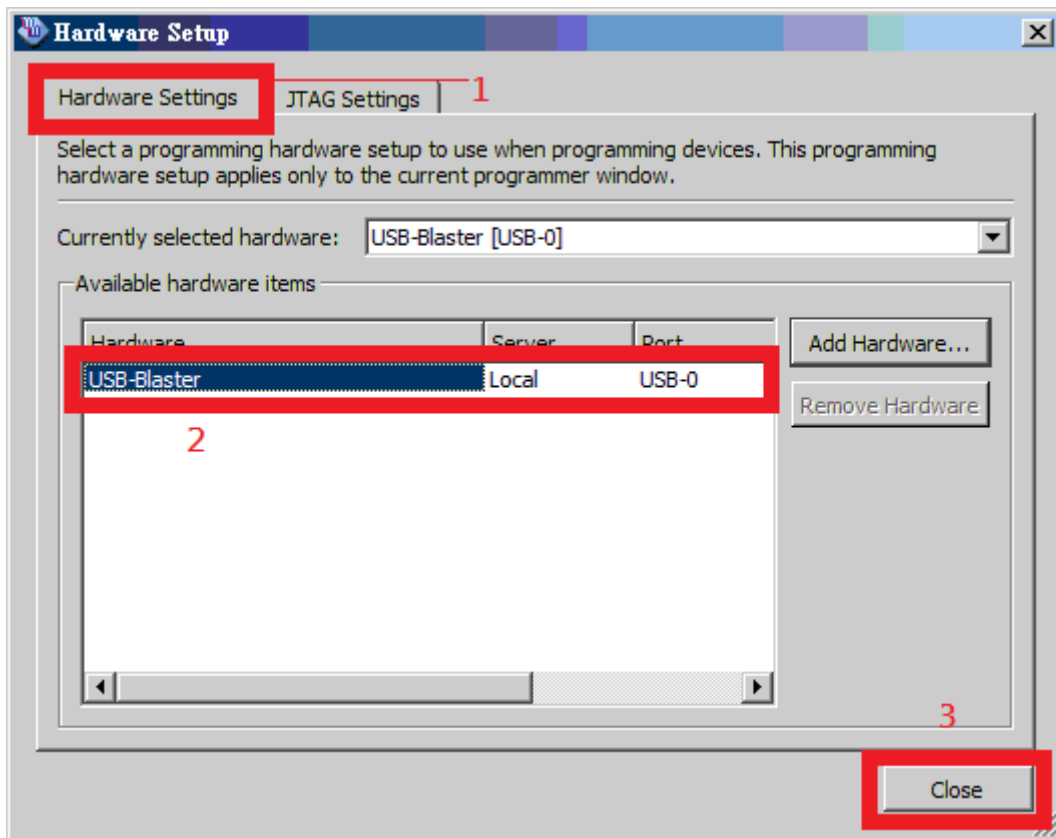
47. 依照程式的 assignment, 將每組相對應的信號 group 起來, 並且, 重新命名



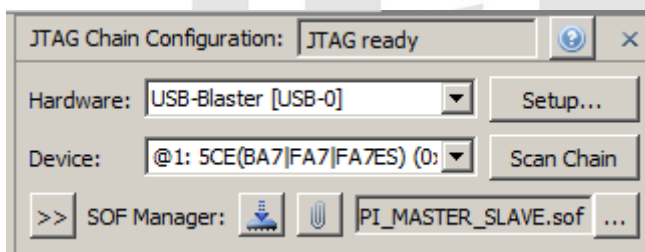
48. 若在 JTAG Chain Configuration 的 Download Cable 沒有 detect 到, 如下圖所示, 但是板子是有插電源的, 且 Download Cable 有接至電腦的話, 就重新設定即可, 按 “Setup...” 按鍵



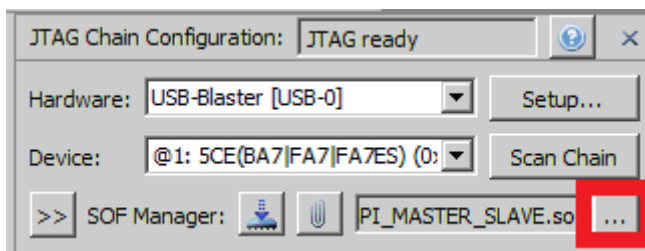
49. 會跳出 “Hardware Setup” 的視窗, 在 “Hardware Settings” 頁面, 在 “Available hardware items” 中, USB-Blaster 按兩下, 然後按 “Close” 關閉畫面.



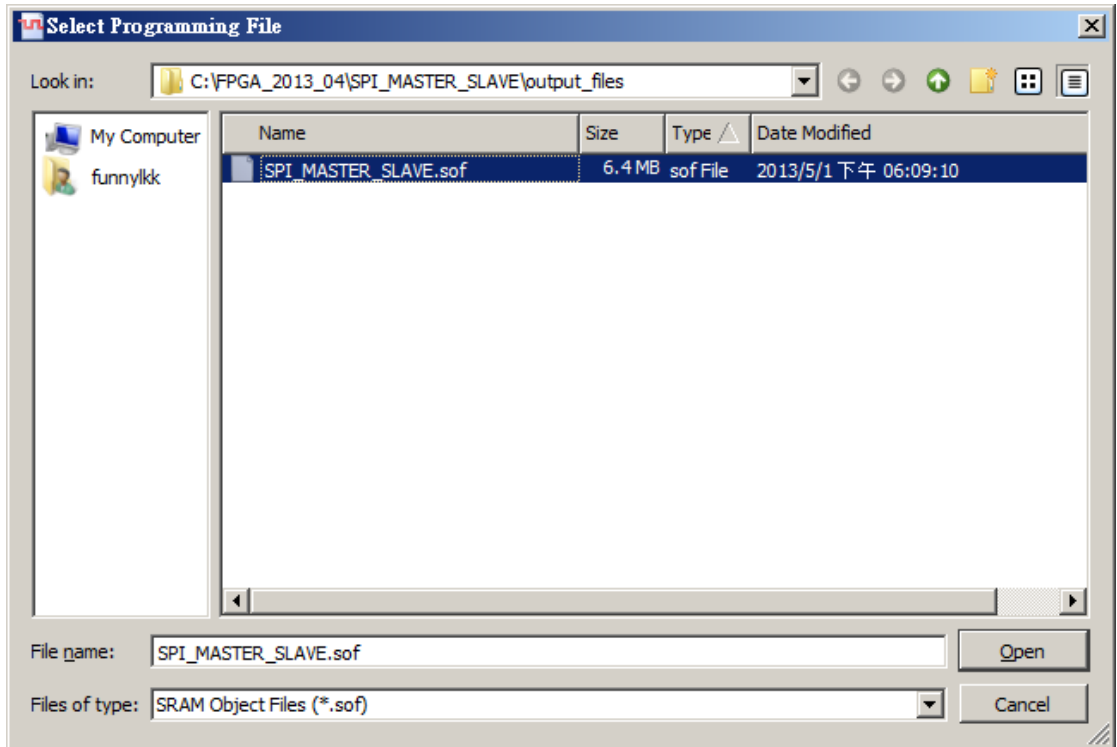
50. 回到 JTAG Chain Configuration 時，就會如下圖所示。



51. 若檔案沒有 link 進來的話，則按下“SOF Manager”欄位的“...”按鍵去 link SOF 檔案



52. 在 QuartusII 11 版之前, SOF 檔案 default 會在 project 同一個路徑底下; 但 QuartusII 12.1 版, 則改成 default 在 project 路徑底下的 output_files 的目錄。



53. 執行的結果....

