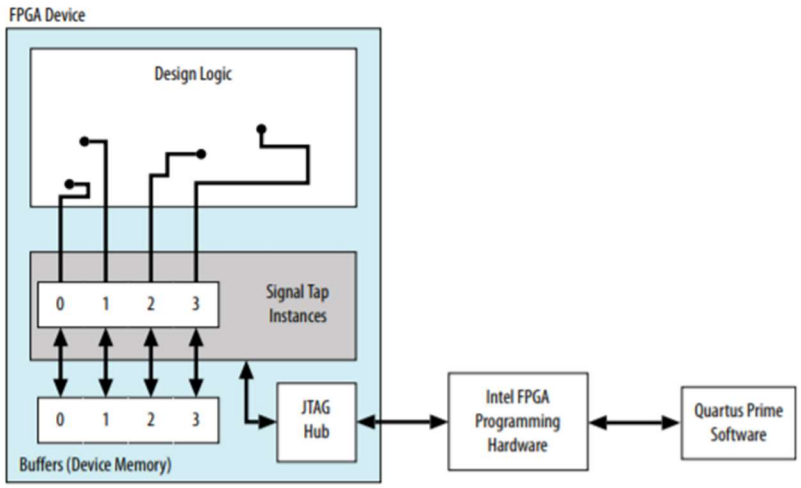
使用 Quartus Signtal Tap Logic Analyzer Debug 電路

FPGA 內部的訊號是否正確

下圖為 Intel® Quartus® Prime Standard Edition Handbook Volume 3 [2]中節錄出來的 Signal Tap Logic Analyzer Block Diagram。

Figure 161. Signal Tap Logic Analyzer Block Diagram

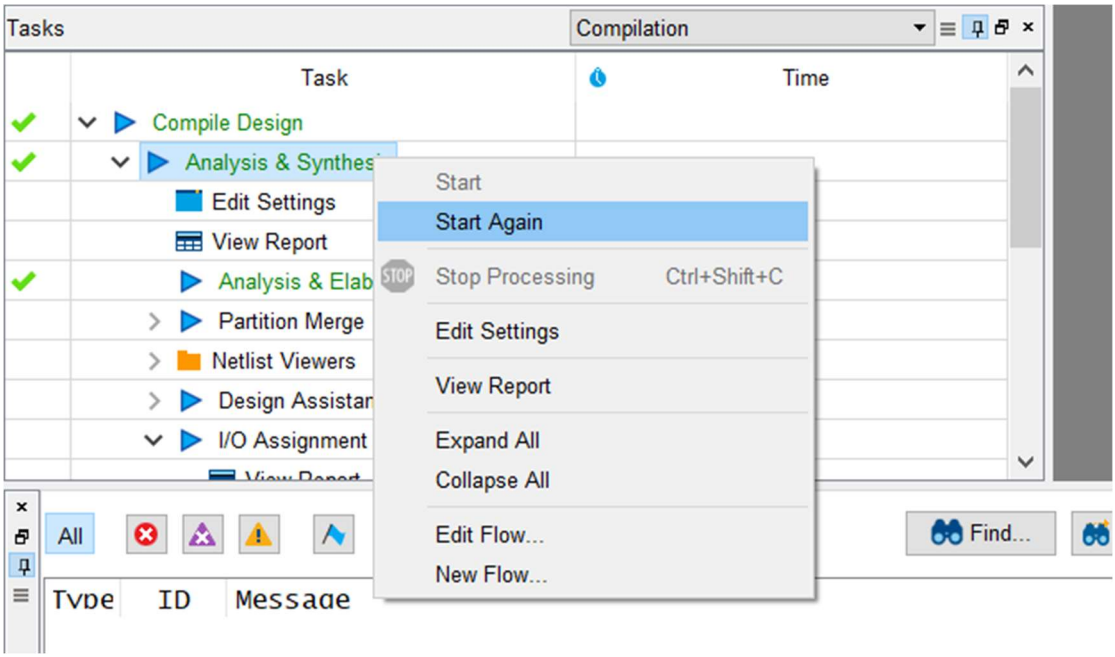


由上圖可知 Signal Tap Logic Analyzer 是一個內嵌在 FPGA 裏面執行 Logic Analyzer(邏輯分析儀)功能的電路。透過 USB Download Cable(USB Blaster II)和電腦中的 Quartus Prime Software 做連線來觀察目前 FPGA 內的 Design Logic 即時訊號狀態。

下列以跑馬燈的電路為例，來看一下如何使用 SignalTap 觀察跑馬燈電路的實際運作情況。

(Quartus Project : led_ctrl)

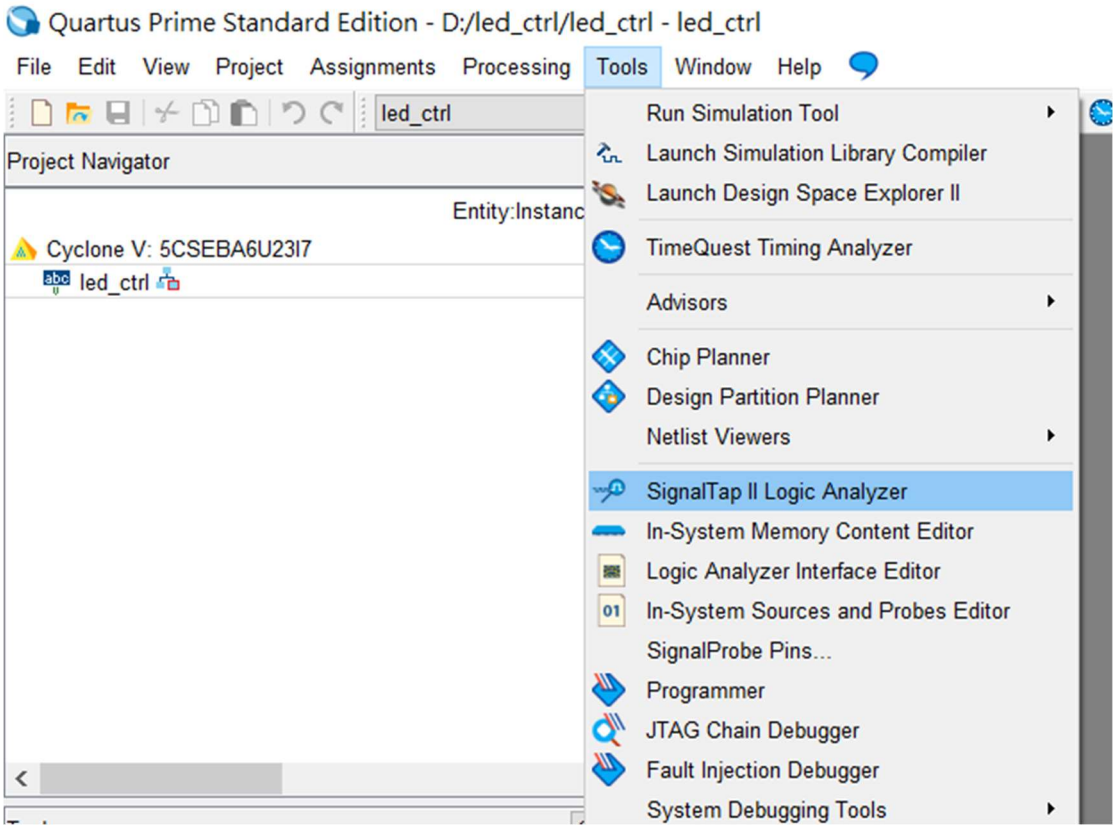
[Step 1]如果有修改過 RTL Code，則要重新執行”Analysis & Synthesis”如下圖一所示。



圖一 “Analysis & Synthesis”於 Quartus Tools 中的位置

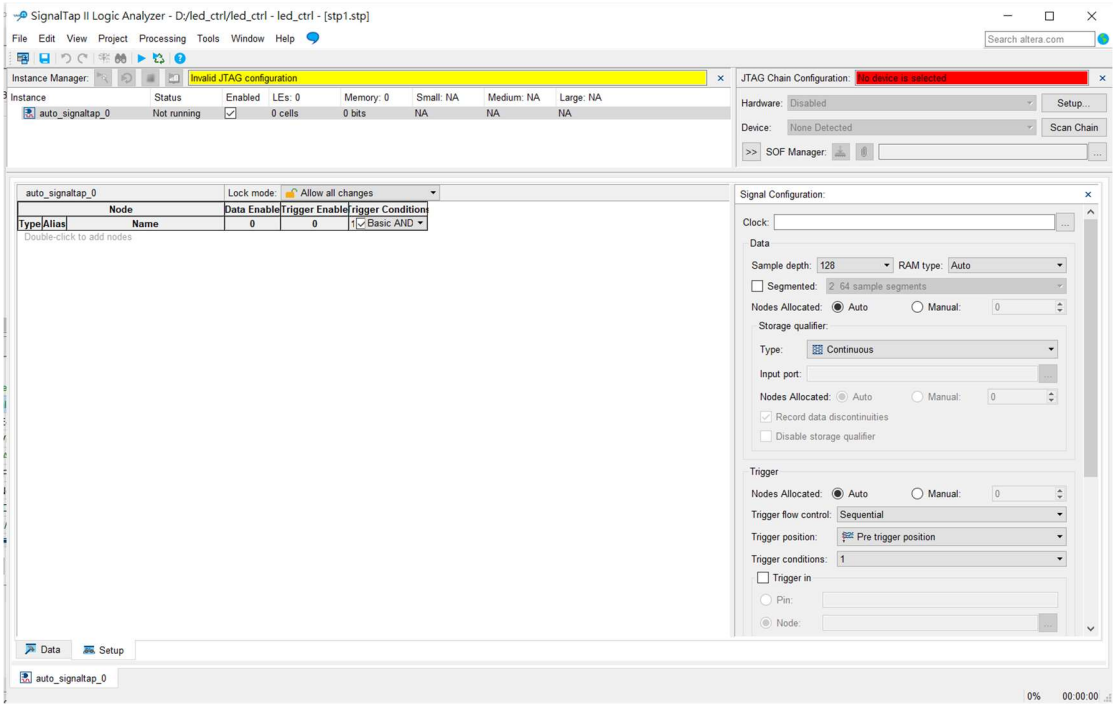
[step 2]開啟 SigntalTap II 圖形化介面(GUI)

利用滑鼠左鍵點選 Tools -> SignalTap II Logic Analyzer(如下圖二所示)



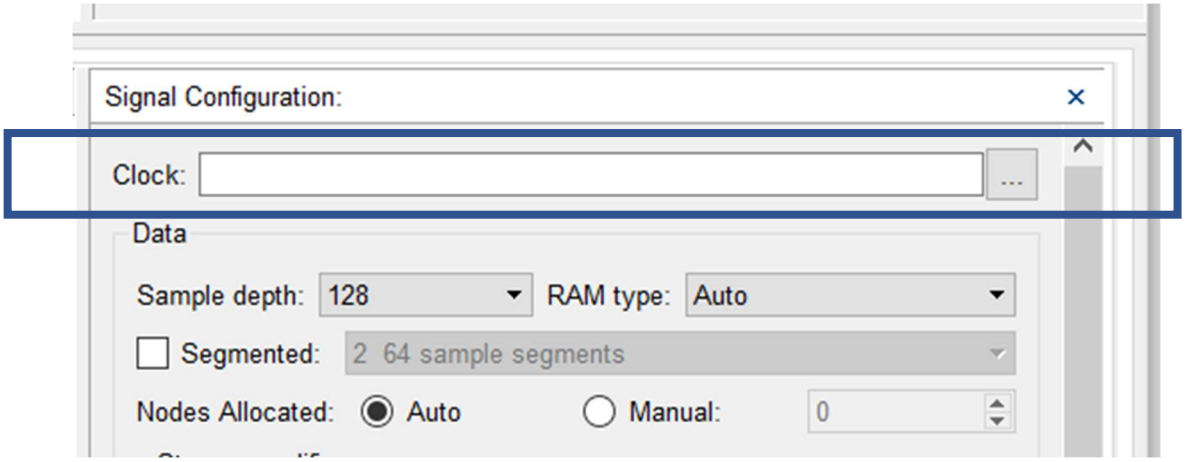
圖二 SignalTap II Logic Analyzer 於 Quartus 軟體中的位置

點選完後會出現如下的視窗



[step3]選擇 SignalTap II Logic Analyzer Clock Source

於 SignalTap II Logic Analyzer 視窗中的 Signal Configuration 中的 Clock 選擇 SignalTap II 的 Clock，如下圖三中方框所示。



圖三: 設定 SignalTap II Logic Analyzer 中的 Clock 訊號

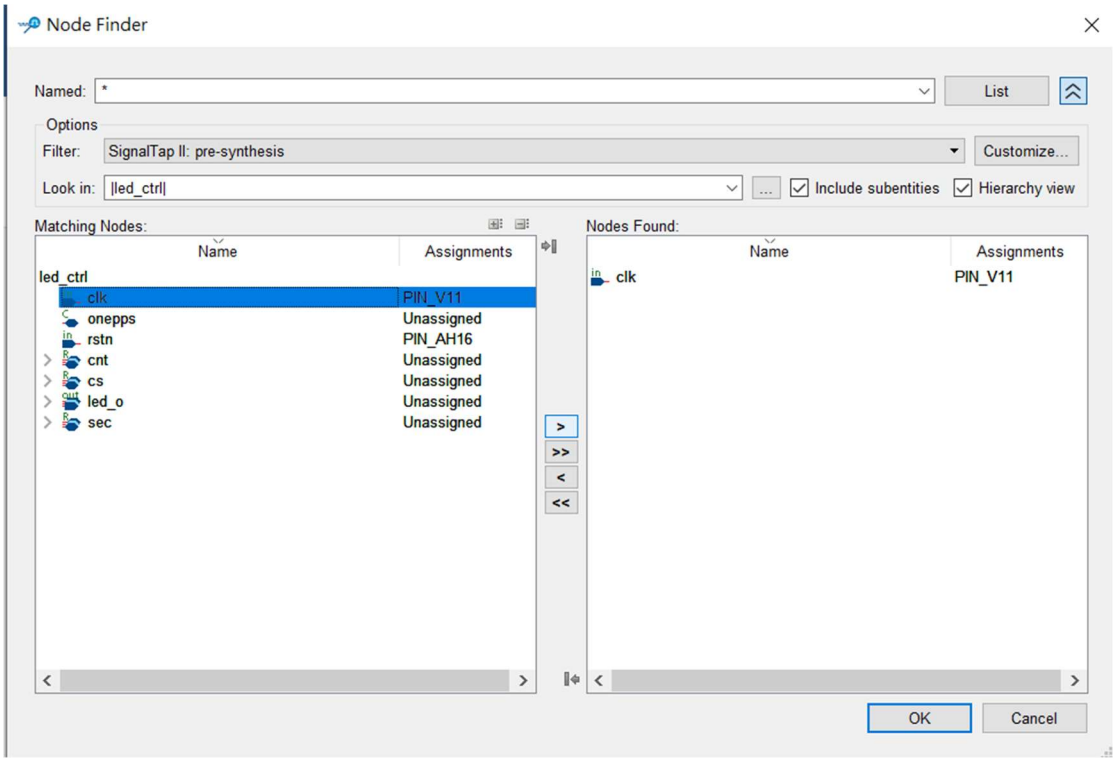
如下節錄說明文件[2]為選用 Acquisition Clock 的原則。

14.3.1 Assigning an Acquisition Clock

To control how the Signal Tap Logic Analyzer acquires data you must assign a clock signal. The logic analyzer samples data on every positive (rising) edge of the acquisition clock. The logic analyzer does not support sampling on the negative (falling) edge of the acquisition clock.

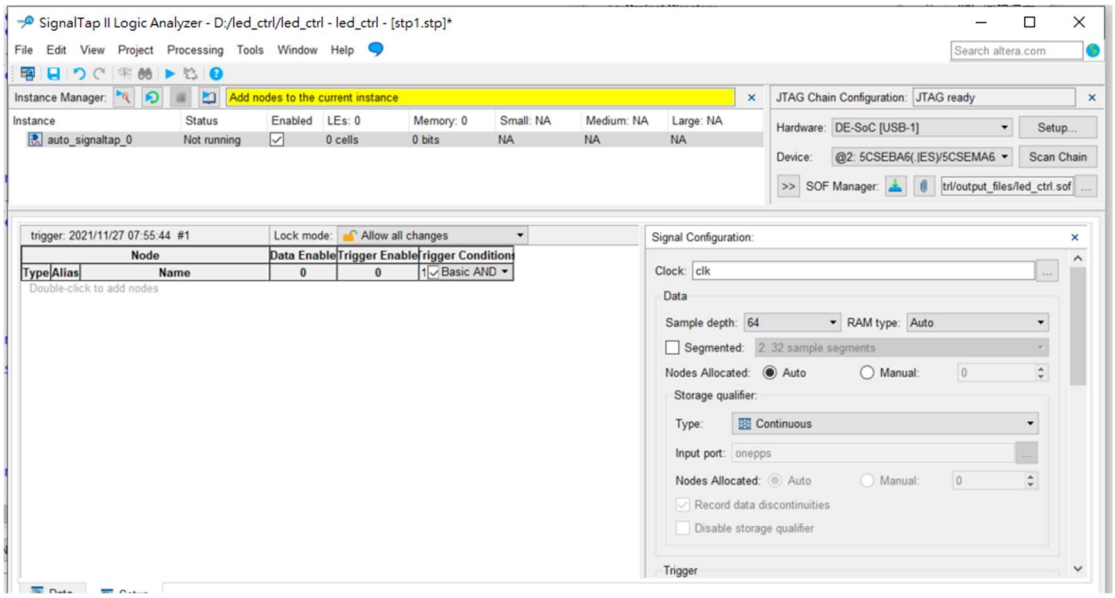
You can use any signal in your design as the acquisition clock. However, for best results in data acquisition, use a global, non-gated clock that is synchronous to the signals under test. Using a gated clock as your acquisition clock can result in unexpected data that does not accurately reflect the behavior of your design. The Intel

所以我們選用系統時脈訊號，在本線路中是訊號名為 clk，如下圖四所示:



圖四 Signal Tap 選取 clk

點選 OK 之後，整個 Signal Tap 視窗應如下圖五所示。



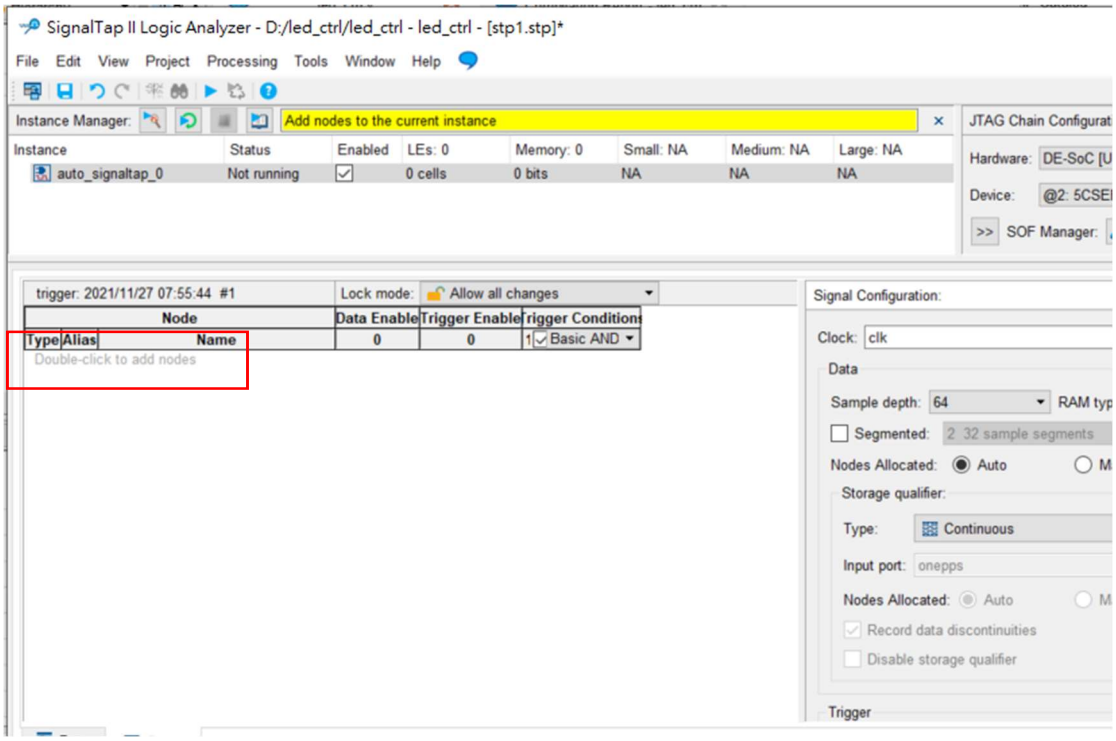
圖五

[step3]選取預觀察的訊號

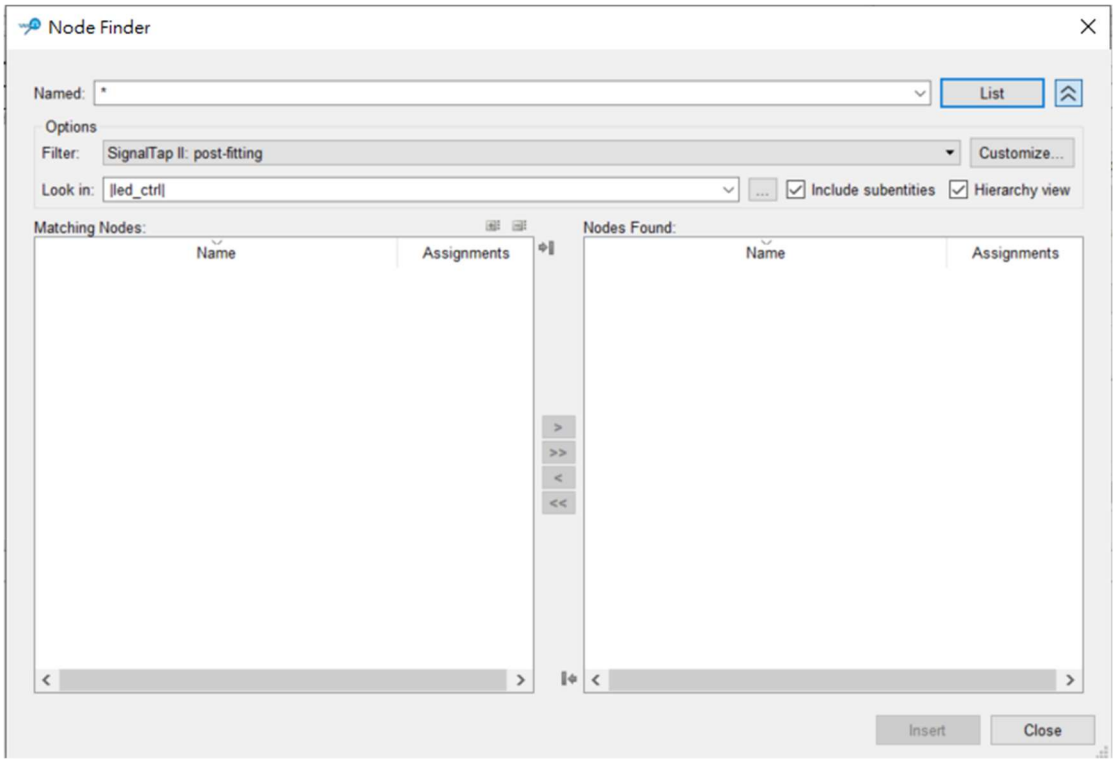
假設我們要觀察四顆 LED 燈是否真會在每隔一秒依序亮起，則需要觀察 1 pps(wire name : onepps)及 LED 燈控制訊號(wire name : led_o[3:0])。

將滑鼠左鍵於下圖圖六中標示”Double-click to add nodes”處點選二下(如紅框標

示處)

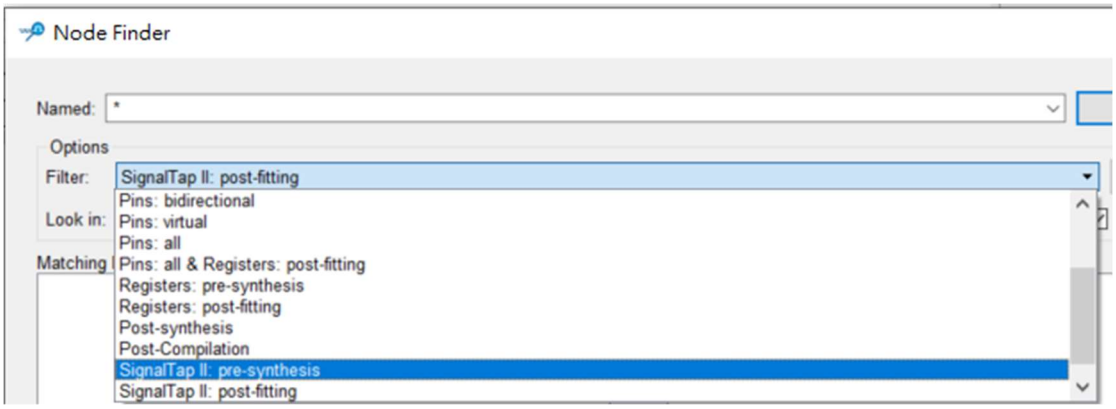


完成後應會出現如下圖七所示的視窗



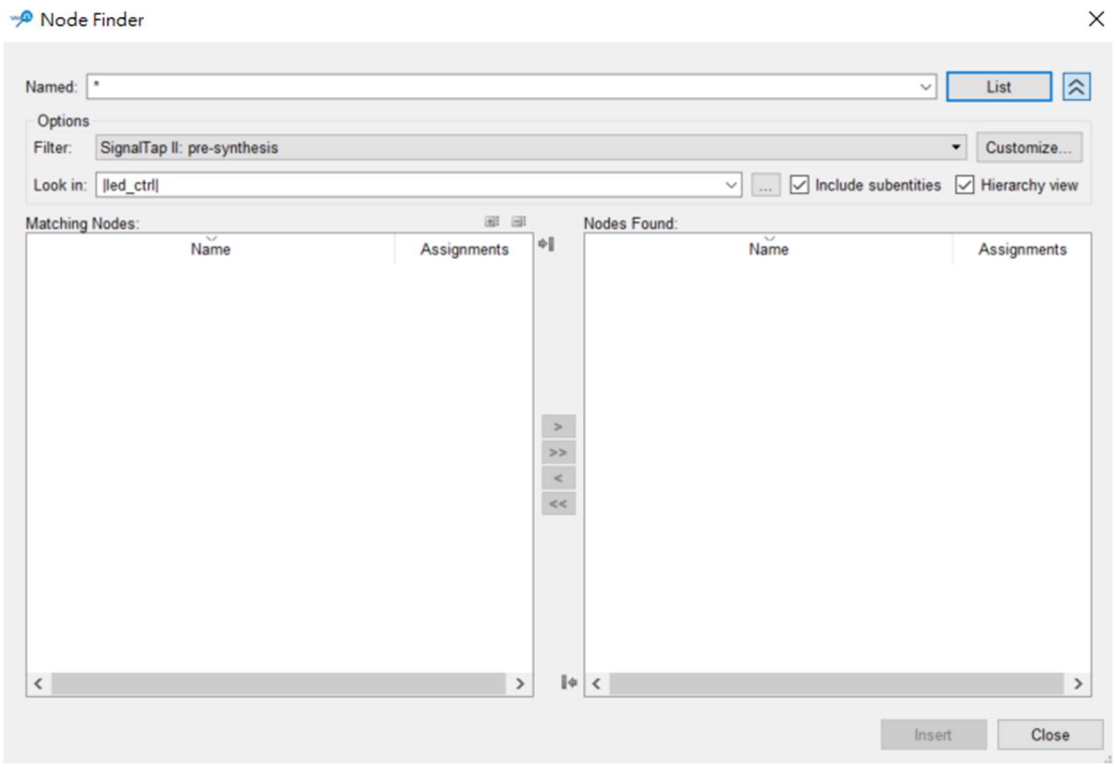
圖七

於再點選 Filter 欄位後，選取”SignalTap II:pre-synthesis”選項如下圖圖八所示:



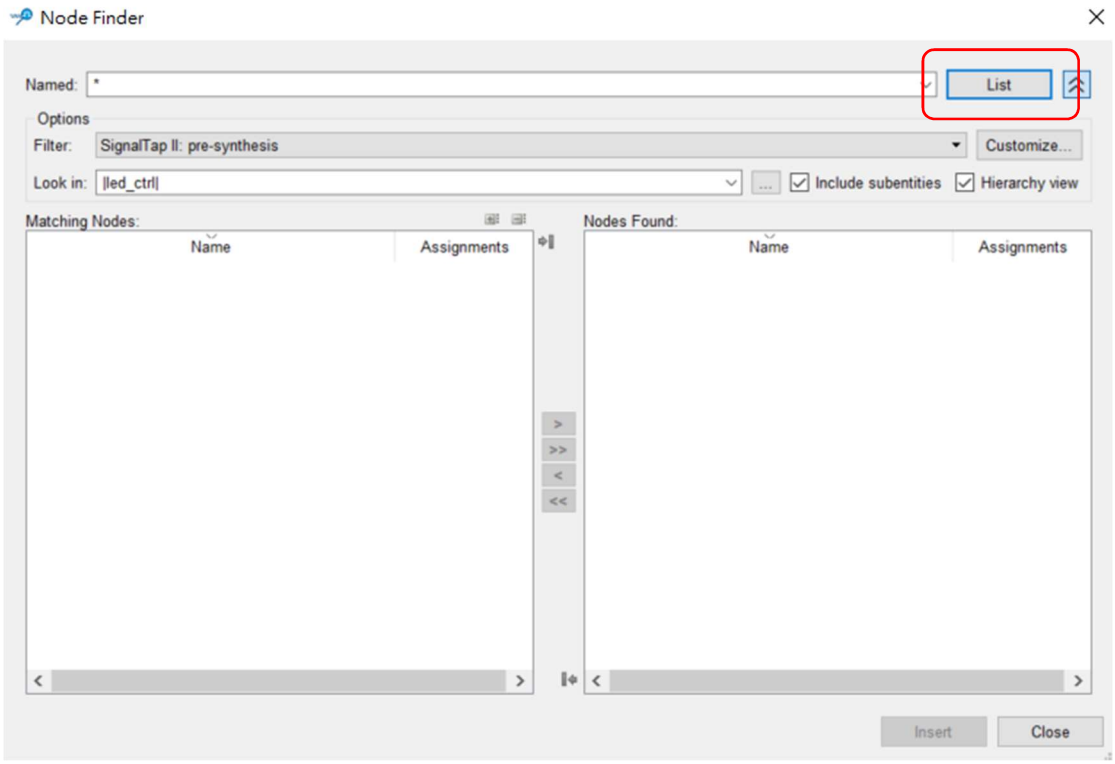
圖八

完成後會如下圖圖九所示:



圖九

接下來點選”List”如下圖圖十紅框標示處:

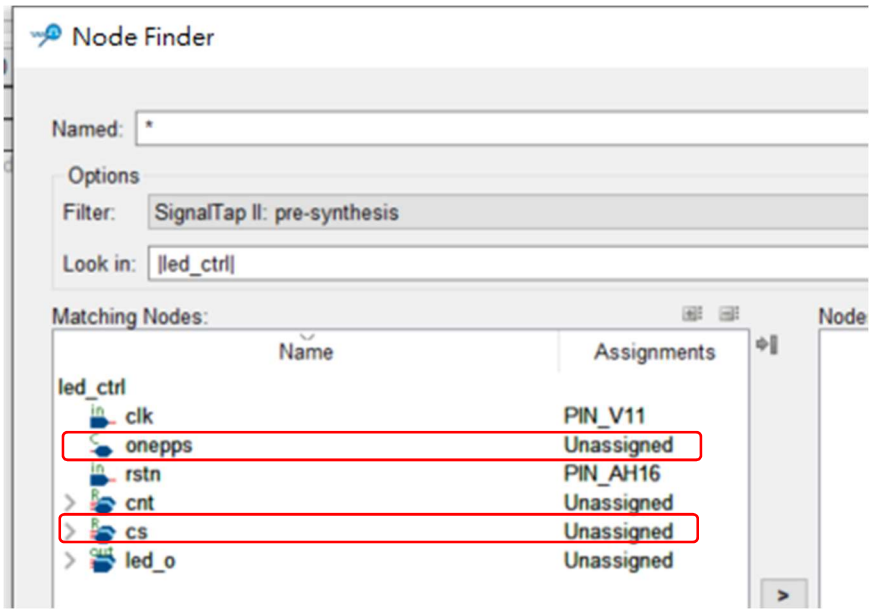


圖十

完成後於 Matching Nodes 欄位處會出現可以觀察的訊號。

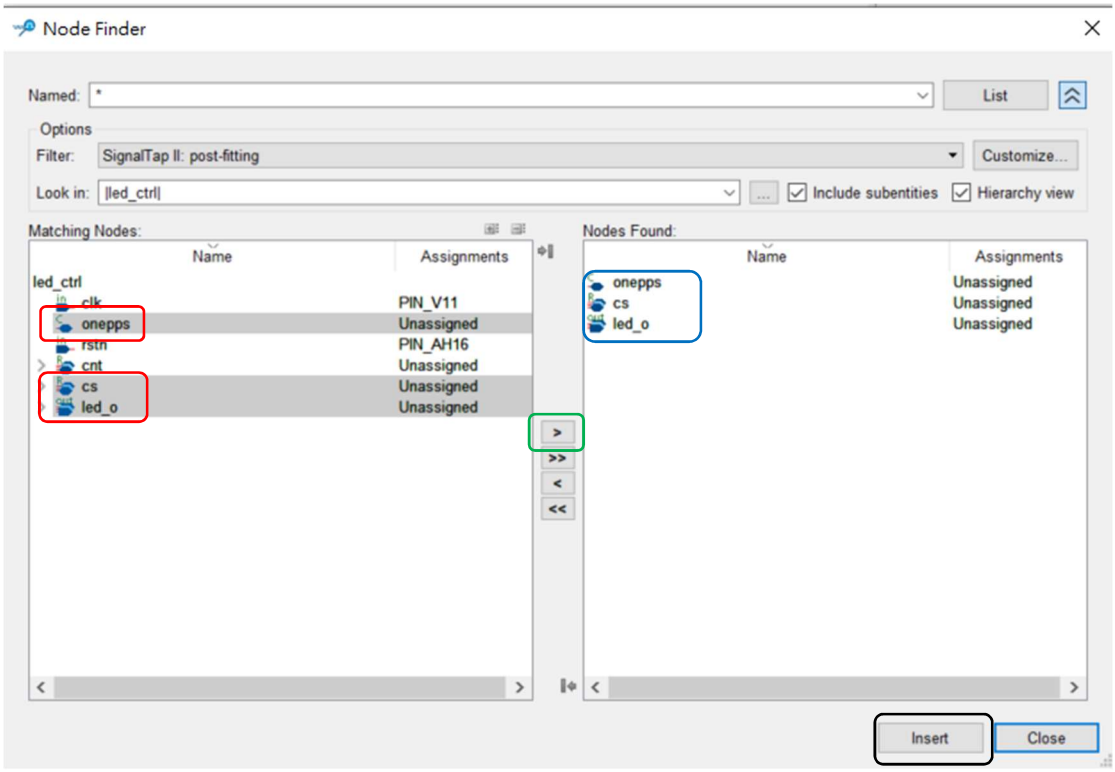
因為我們在 onepps 的宣告中加入如下的宣告，所以 onepps 的訊號線線名被保留下來，而沒有被軟體化簡掉。由 cs 訊號也因為加入如下的宣告，而沒有被軟體化簡。所以皆會以完整的名稱出現在 Matching Nodes 的欄位。如圖十一紅框所示。

```
wire onepps/*synthesis keep*/;
reg   [1:0] cs/*synthesis preserve*/;
```



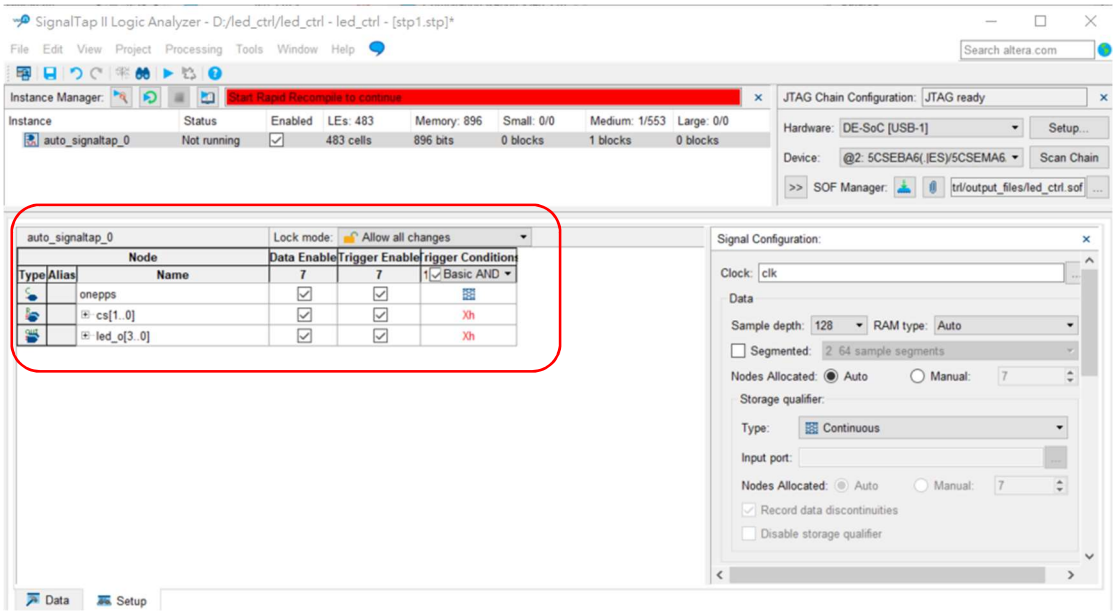
圖十一

利用滑鼠左鍵點選 onepps,led_o,及 cs 三個訊號(如圖十二紅框所示)，再點選  (如圖十二綠框所示)即會在 Nodes Found:的視窗中看到三個訊號(如下圖十二藍框)所示。



圖十二

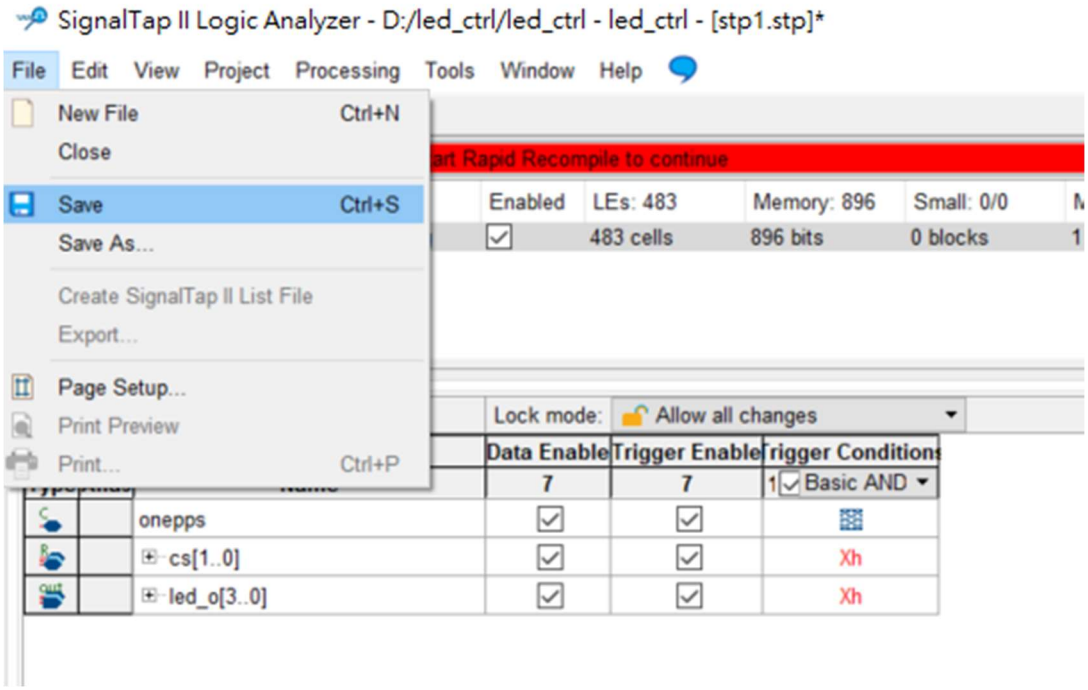
再點選 insert 按鈕(如圖十二黑框)，再按 close 關閉”Node Finder”視窗。即會出現圖十三視窗，到此我們就完成 signal tap 觀察訊號的選取。



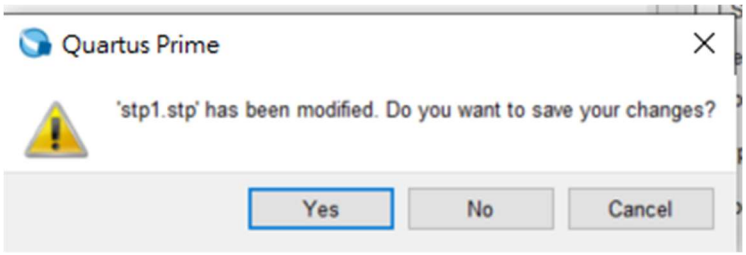
圖十三

[step4]Signal Tap II Logic Analyzer 設定檔存檔

點選 File -> Save 來將 Signal Tap II Logic Analyzer 設定檔存檔，如下圖十四所示。



圖十四

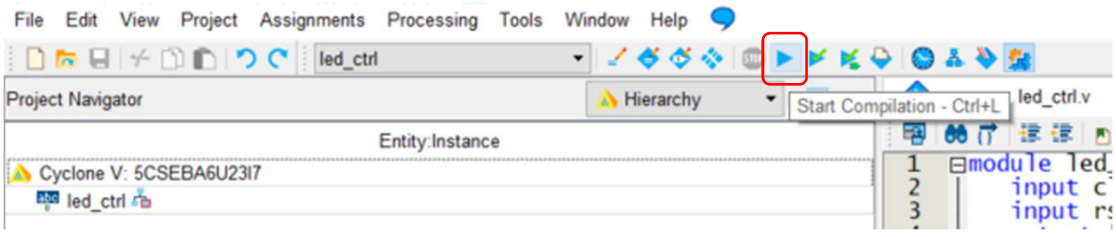


選取 Yes，即完成存檔。

Note: Each `.stp` file is associated with a programming (`.sof`) file. To function correctly, the settings in the `.stp` file you use at runtime must match Signal Tap settings in the `.sof` file you use to program the device.

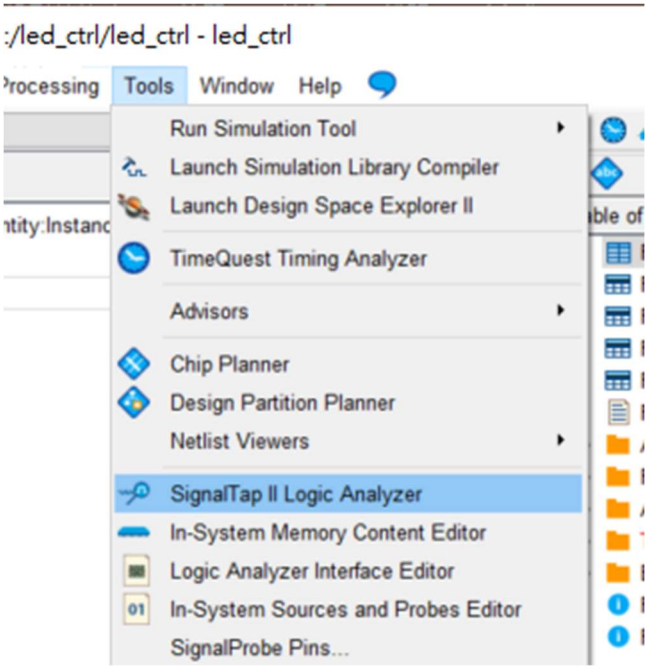
[step5]重新 Compile Design

點選”Start Compilation”按鈕(如下圖十五中的紅框所示)來重新 compile 整個 Design。

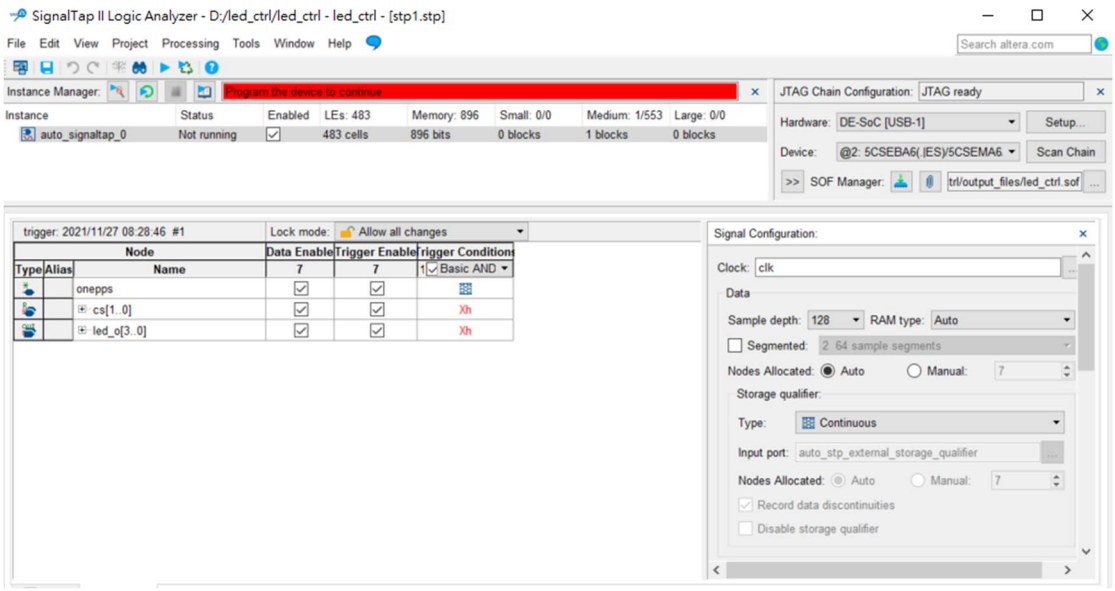


[step6]設定 SignalTap II

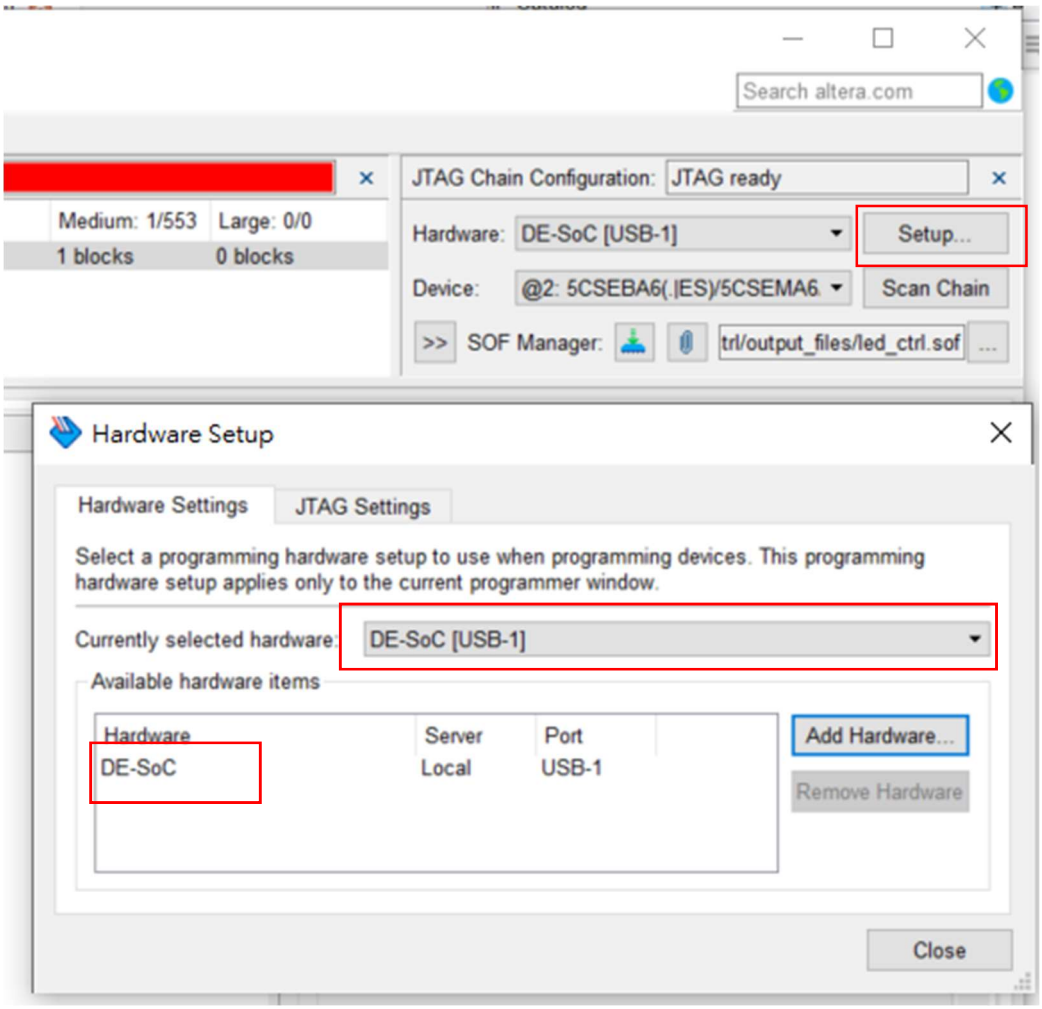
待整個 Quartus Project Compile 完後，點選 Tools



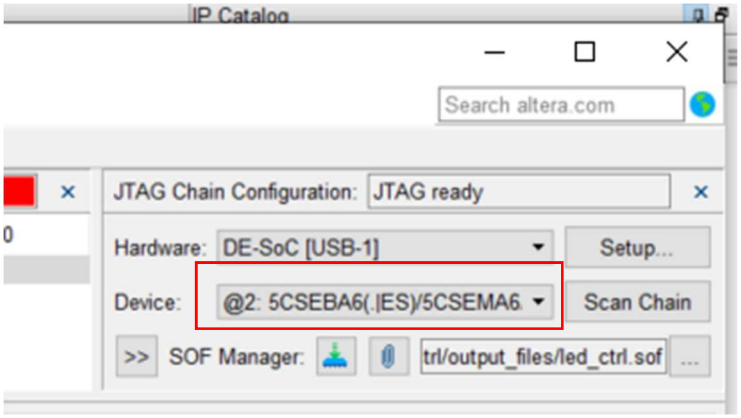
即會出現如下的視窗



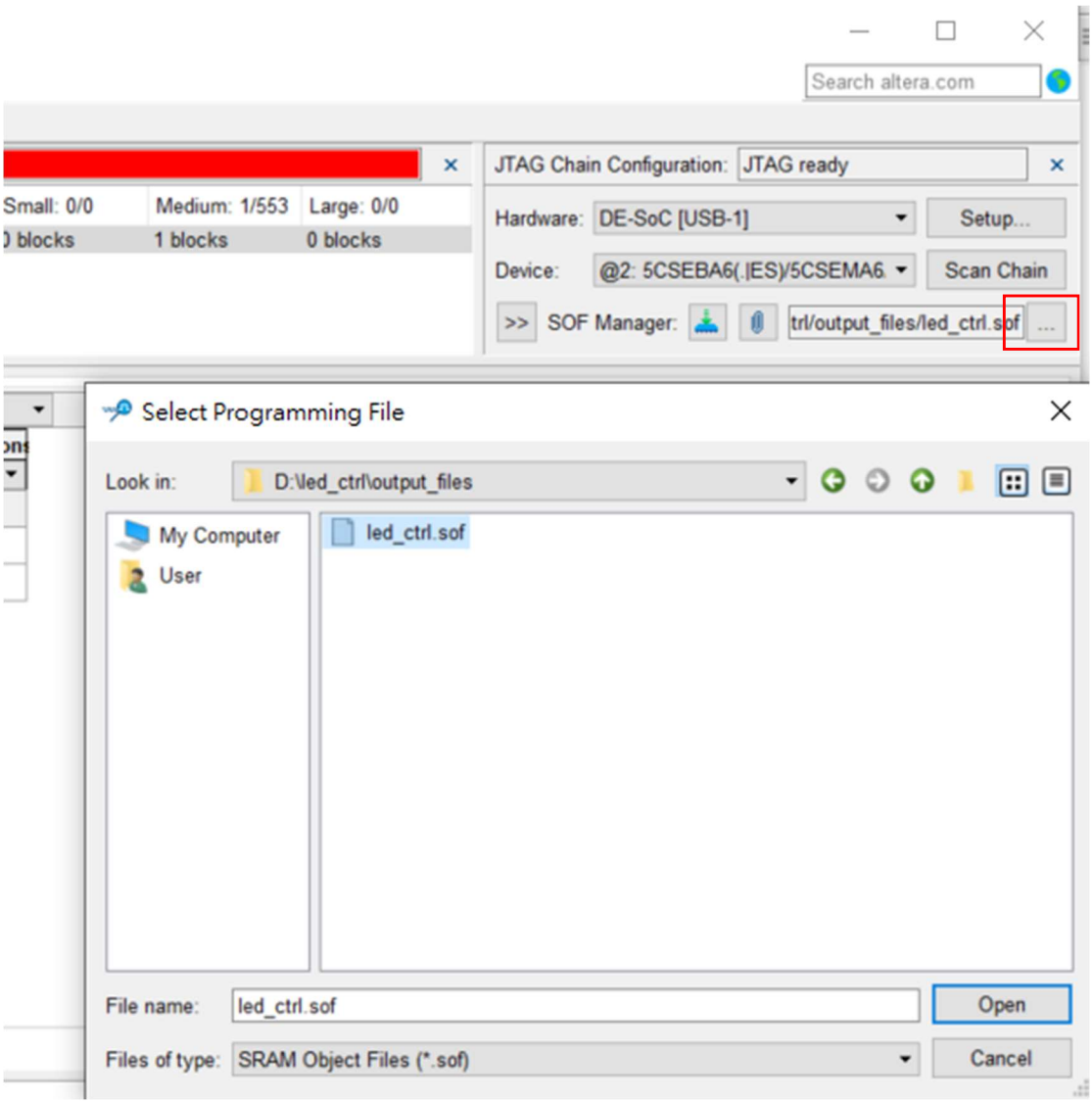
點選”Setup”選項，並選取 DE-SoC[USB-1]



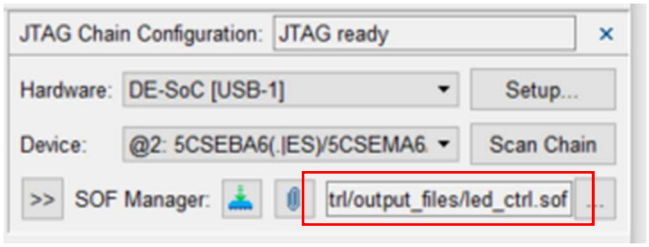
點選”Scan Chain”選項，並選擇”@2:5CSEBA6(.IES)/5CSEMA6”



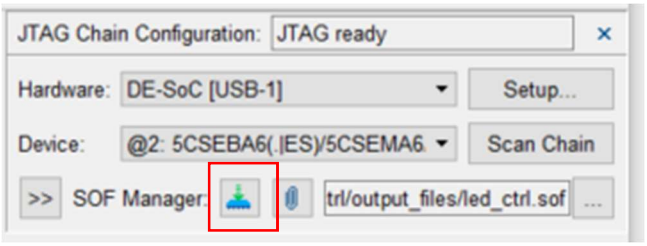
點選如下圖紅框標示處來選擇 FPGA 的燒錄檔。此 Project 的檔名為 led_ctrl.sof



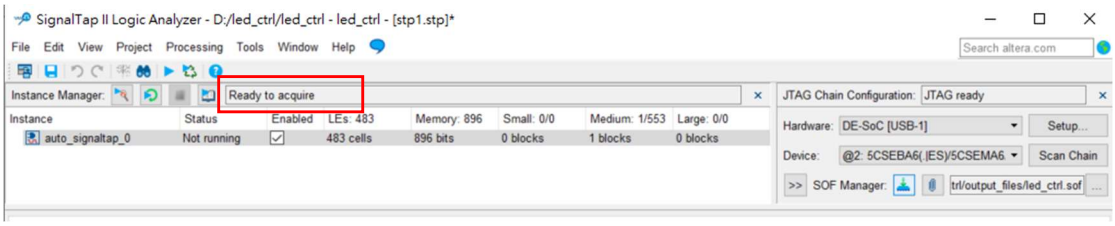
完成後，SOF 檔應會出現如下圖紅框所示之處。



點選下圖紅框所示的按鈕  來燒錄 FPGA 。



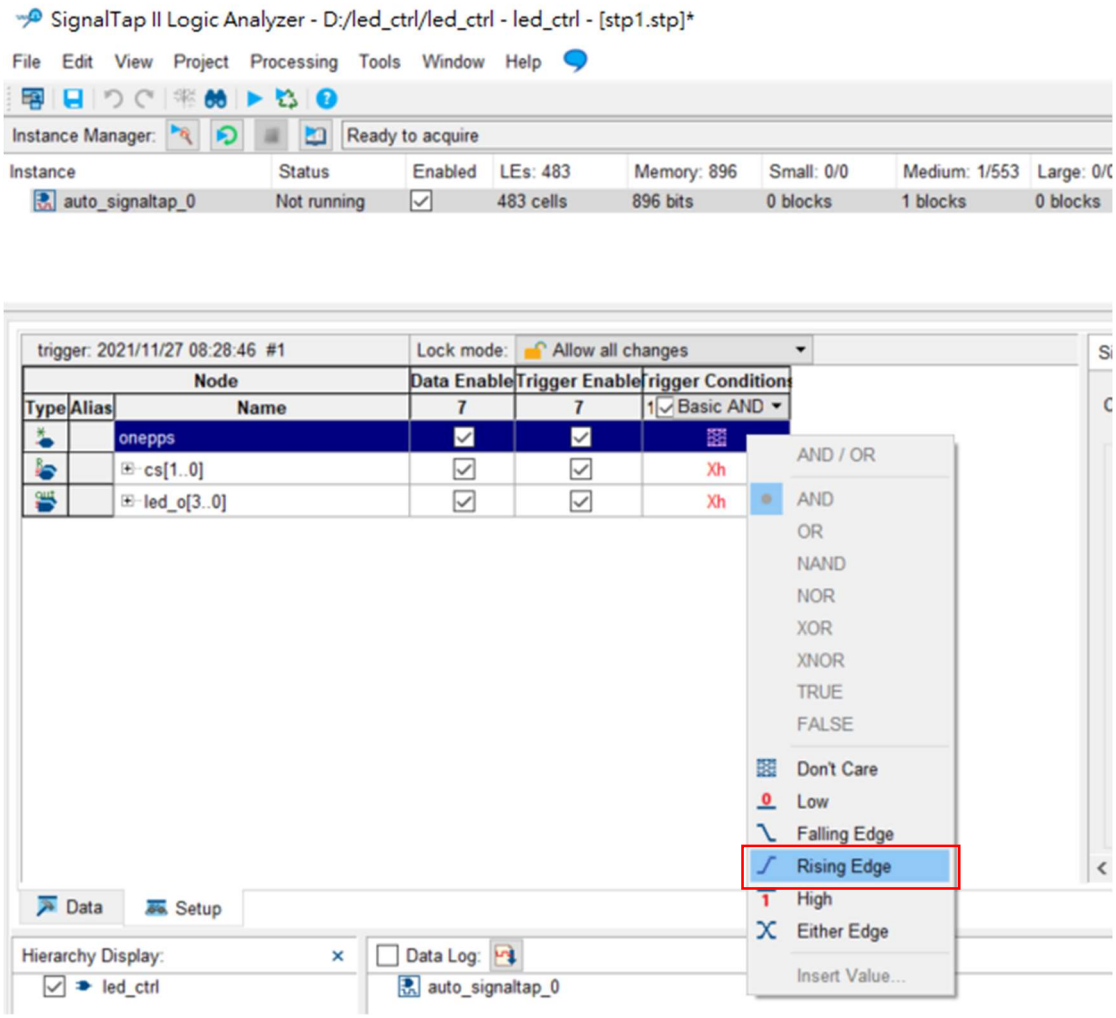
完成後應會出現”Ready to acquire”字樣如下紅框所示:



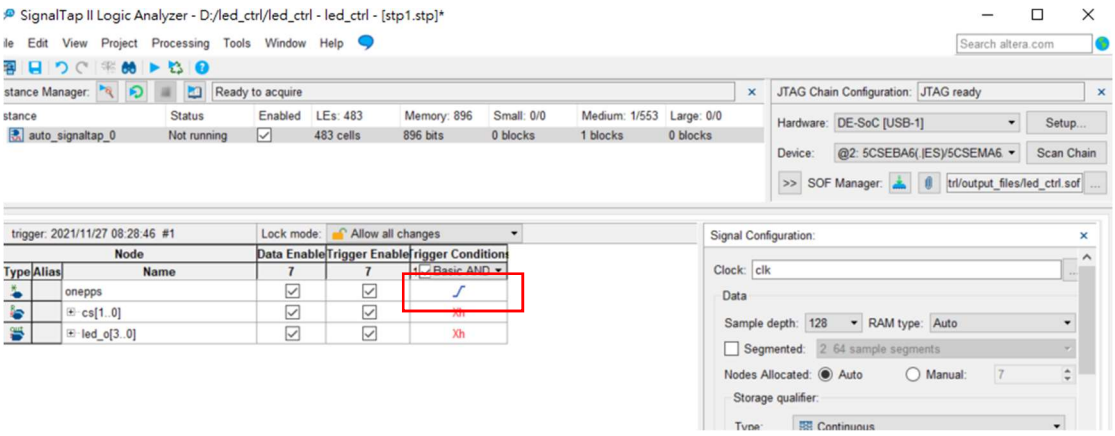
以上 signaltap 即可完成抓取訊號的準備。

[step]設定 SignalTap II trigger 條件

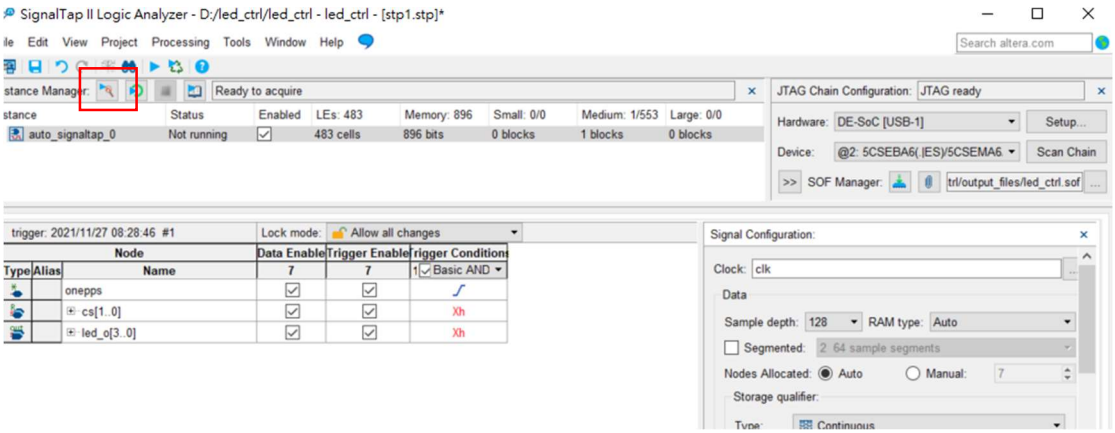
設定 signaltap trigger 條件為 onepps 由 low 到 high 變化時,即抓取訊號。設定如下圖紅框所示:



完成後會如下圖所示:



點選  按鈕(如下圖紅框)，即會開始抓取訊號。



抓取訊號如下圖所示。

Reference:

[1] https://faculty-web.msoe.edu/johnsontimoj/Common/FILES/qts-qps-5v3_signaltap_17.1.pdf

[2]https://www.intel.la/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qpp-compiler.pdf