

Jini Intelligent Computing Workbook of Lab. #2 For KV260

Preamble

在 Lab. #2 範例分為兩個主題，第一類 AXI-Master Interface 實作範例；第二類 Stream Interface 實作範例。

第一類實作範例的檔案系統中主要有下列專案目錄：

- hls_FIRN11MAXI
Vitis HLS 之以 AXI-Master Interface 為設計 FIR 原始碼檔案
- vvd_FIRN11MAXI
範例乘法器 Vivado Design Suite 參考檔案
 - design_1.tcl
範例 FIR 之 Block Design 完成 Generate Bitstream 後匯出之 TCL Script 檔
 - MakeBit.bat
範例 FIR 完成 Generate Bitstream 後，將.bit/.hwh 拷貝至專案根目錄之批次檔
- ipy_Multip2Num
範例 FIR 系統程式 Python 原始碼檔及 Jupyter Notebook 原始碼編輯檔

第二類實作範例的檔案系統中主要有下列專案目錄：

- hls_FIRN11Stream
Vitis HLS 之以 Stream Interface 為設計 FIR 原始碼檔案
- vvd_FIRN11Stream
範例乘法器 Vivado Design Suite 參考檔案
 - design_1.tcl
範例 FIR 之 Block Design 完成 Generate Bitstream 後匯出之 TCL Script 檔
 - MakeBit.bat
範例 FIR 完成 Generate Bitstream 後，將.bit/.hwh 拷貝至專案根目錄之批次檔
- ipy_Multip2Num

與 PYNQ-Z2 不同的步驟會以紅底 HIGHLIGHT

1. FIR with Interface AXI-Master

【施作環境為在使用者 PC/laptop/notebook (Windows Base) 。】

1.1. HLS/IP Design

本 Lab.#2 實作，HLS 開發及驗證同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。（如果你在使用 Linux，Tester 中 Linux 下不能用 fc，需要用 diff, 且需要注意檔案的 path 是否正確，以及由於 out_golden.dat 檔案來自 Windows，所採用的換行符和 Linux 系統不一樣，需用 dos2unix 命令轉換。）

1.2. Vivado Implementation

1.2.1. Create Design Project

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

period 設為 10ns 避免 Simulation 時出現 time violation。

Parts 使用 **xck26-sfvc784-2LV-c** → add source*2, add testbench*1, add top

function → directives → C simulation → C synthesis → Export RTL

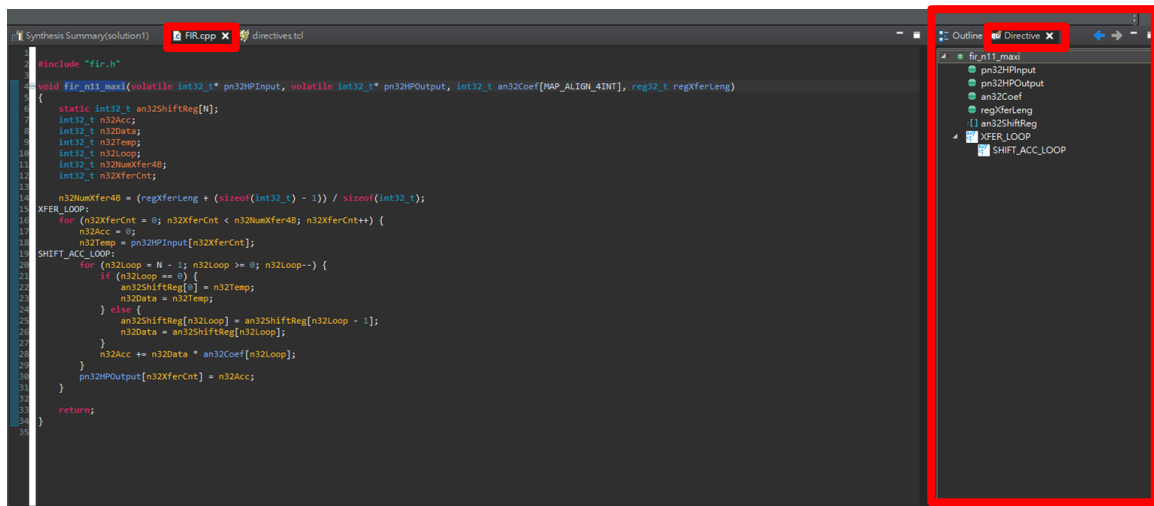
(雖然 VITIS HLS 有 KV260 的 Board file，但跑 C sim 及 C syn 時會出現 error)

1.2.2. Vitis Directive controls

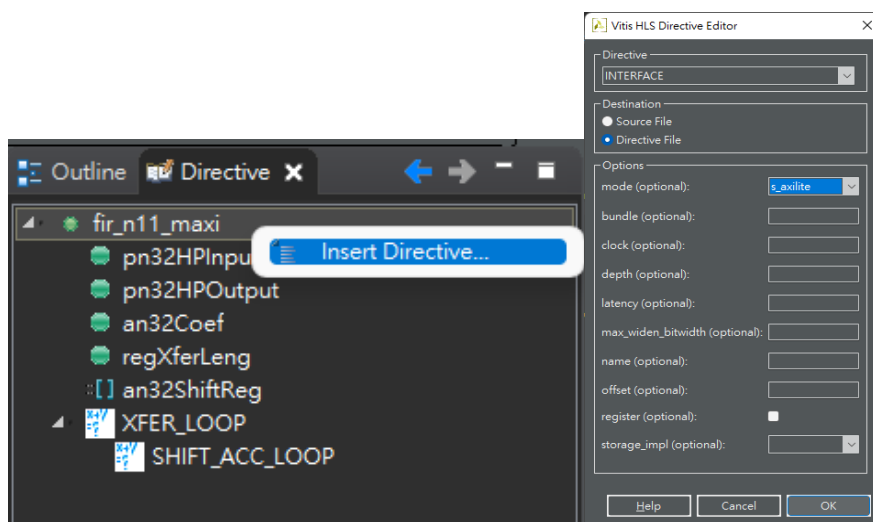
可以從 FIR.cpp 及 solution1 裡看出這次是使用 directives.tcl 檔來設定 directives。

參照 solution1 檔案夾裡的 directives.tcl 檔案路徑設定。

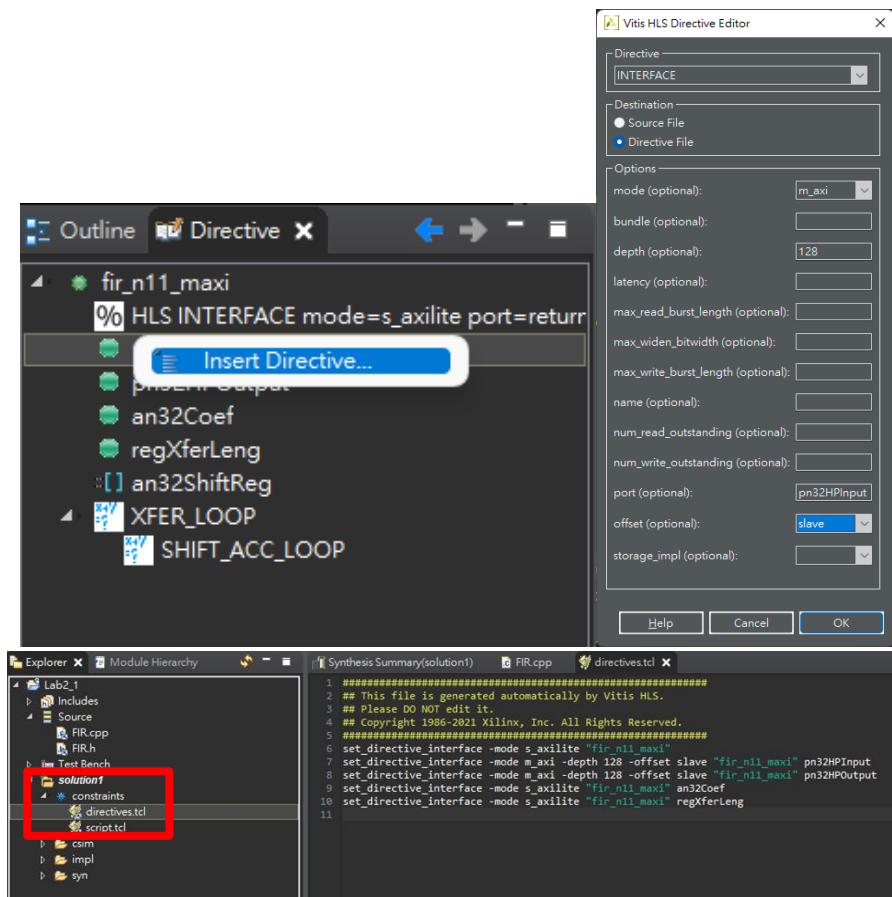
打開 FIR.cpp 後右側點擊 Directive。



首先在 directives tab 頁面上編輯 directives。在 top function 按滑鼠右鍵插入 interface 的 directive 的組態。



相同地，為其他 top function 引入的引數插入 interface 的 directive 的組態。(有的設置要把視窗拉大才看的到) (depth 請設定超過 600)



完成後可以在這裡看 directives 。

1.2.3. Import IP

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

(Board 選 KV260 → settings → ip → repository → add ip (vitis project folder) → ok)

1.2.4. Block Design

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

Zynq UltraScale+ MPSoC > run block automation > set port > add function block > run connection automation *2

以下僅針對 AXI-Master 在 processing system block 的設定補充。（Run block automation 會重置 ZYNQ 的 configuration，一定要先 run block automation 然後再完成 ZYNQ 的 configure）

AXI-Lite 與 AXI-Master 在 MPSoC 使用的 port 並不相同，所以在 Run Block Automation 後用滑鼠左鍵雙擊 MPSoC，開啟 HP port 設定。

Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

Documentation Presets IP Location

Page Navigator

Switch To Advanced Mode

PS UltraScale+ Block Design

I/O Configuration

Clock Configuration

DDR Configuration

PS-PL Configuration

PS-PL Configuration

Search: Q-

Name	Select
> General	
✓ PS-PL Interfaces	
> Master Interface	
✓ Slave Interface	
✓ AXI HP	
> AXI HPC0 FPD	<input type="checkbox"/>
> AXI HPC1 FPD	<input type="checkbox"/>
> AXI HP0 FPD	<input checked="" type="checkbox"/>
> AXI HP1 FPD	<input type="checkbox"/>
> AXI HP2 FPD	<input type="checkbox"/>
> AXI HP3 FPD	<input type="checkbox"/>
> AXI LPD	<input type="checkbox"/>
> S AXI ACP	
> S AXI ACE	
> Debug	

OK

Cancel

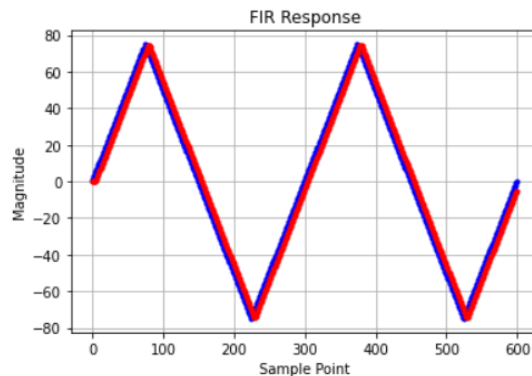
1.3. Python Code Validation via Jupyter Notebook

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

上傳.bit、.hwh、smiple 檔到 KV260 上，新建 python 3，記得更改 code 裡使用到的檔案路徑。

[KV260 檔案位置為 ol = Overlay("/home/root/jupyter_notebook/FIRN11Stream.bit")]

```
Entry: /usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py
System argument(s): 3
Start of "/usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py"
Kernel execution time: 0.0002677440643310547 s
```



```
=====
Exit process
```

2. FIR with Interface Streaming

【施作環境為在使用者 PC/laptop/notebook (Windows Base) 。】

2.1. HLS/IP Design

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

(Review：Parts：**xck26-sfvc784-2LV-c** → Add Source*2, Add testbench*1, Add Top Function → Set Directives → C Simulation → C Synthesis → Export RTL)

2.2. Vivado Implementation

2.2.1. Create Design Project

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

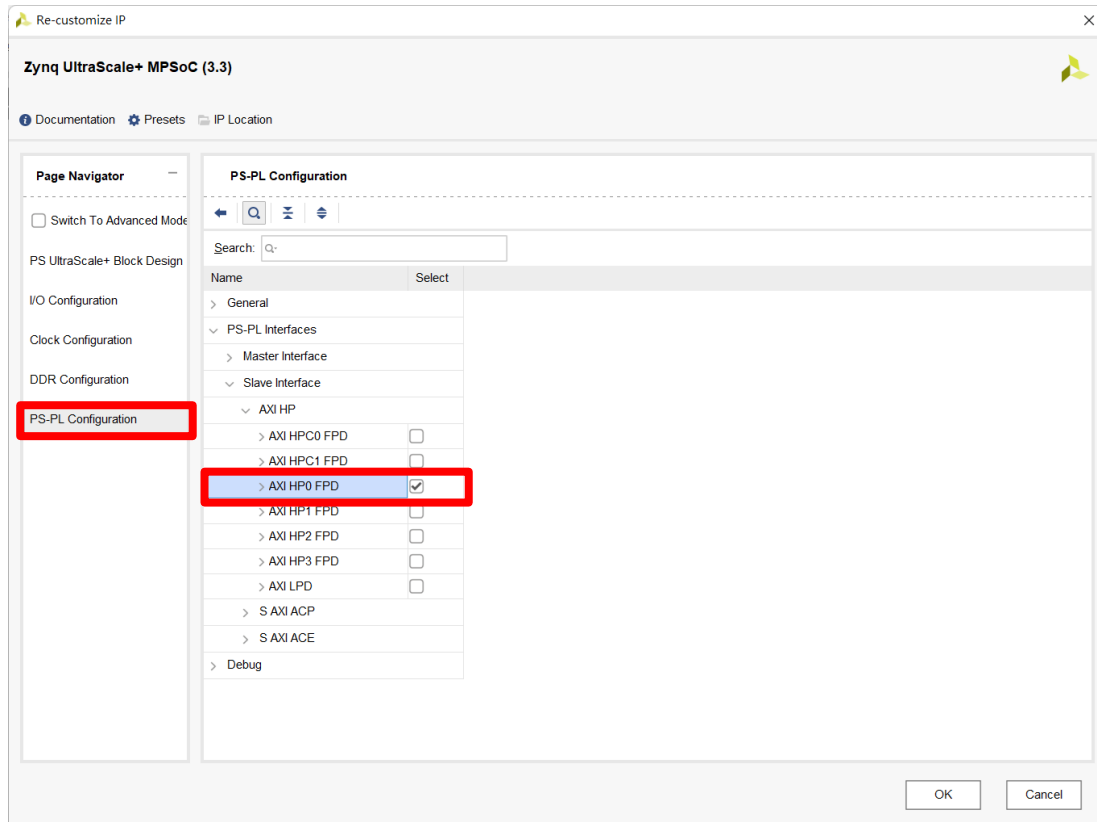
2.2.2. Import IP

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

2.2.3. Block Design

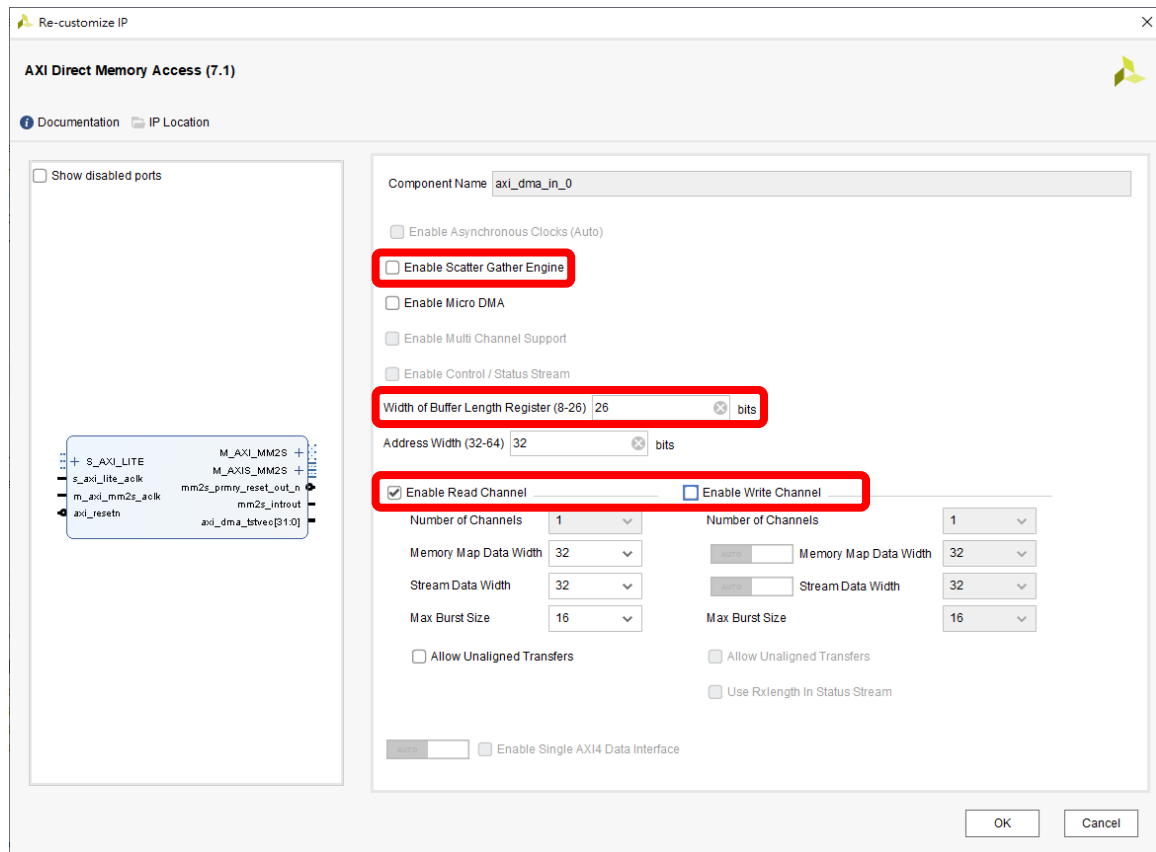
同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。以下僅針對 AXI-Stream 在 processing system block 的設定補充。

AXI-Lite 與 AXI-Stream 在 MPSOC 使用的 port 並不相同，所以在 Run Block Automation 後用滑鼠左鍵雙擊 MPSOC，必須開啟 HP port 設定。

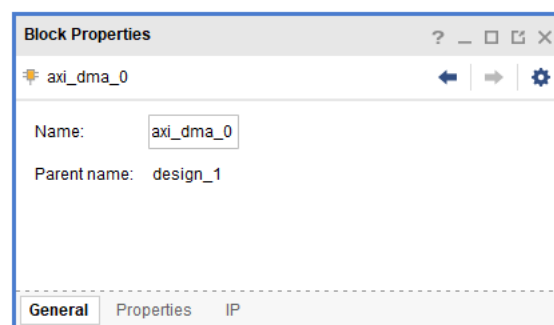
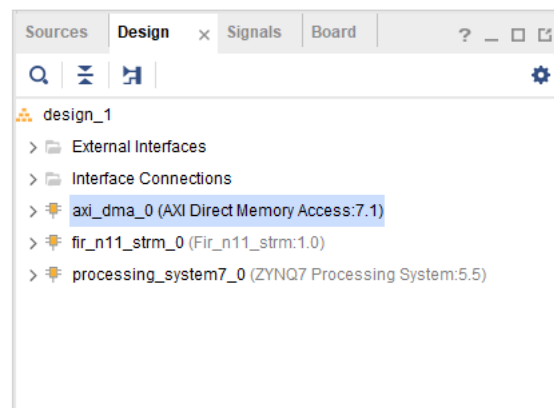


另外，AXI-Master to stream 是用 Xilinx DMA IP 來實現，針對 DMA IP 需要調整及設定。在 Block Design 時，除匯入 HLS fir_n11_strm IP 並將 IP component 加入至 Diagram 中之外，還需要加入 Xilinx DMA IP component，加入後以滑鼠雙擊 DMA block。下圖圈選處：

1. 關閉 Scatter-Gather Mode
2. 調整 Width of Buffer Length Register
3. 選擇單一 Read Channel，單一 Write Channel，或兩者 Read/Write Channel。（由開發者設計需求決定，此 lab 會需要兩個 dma，一個單一 Read 做為 dma in，一個單一 Write 做為 dma out）



dma block 名稱可以在左側更改，python code 裡有用到 dma in 跟 out 的名稱，記得更改成 axi_dma_in_0 及 axi_dma_out_0。



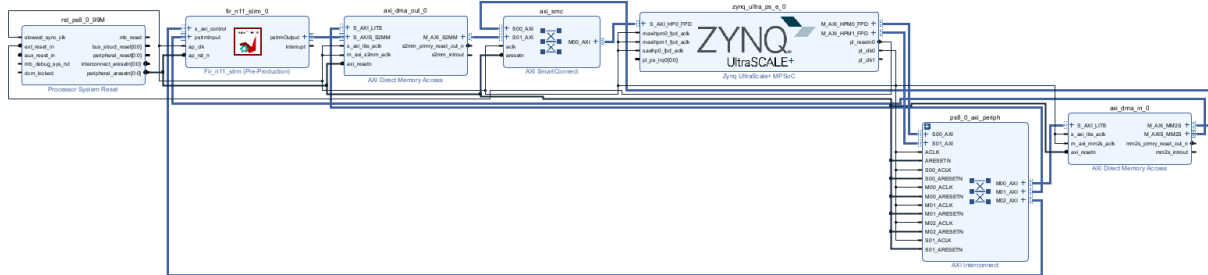


Diagram x Address Editor x Address Map x

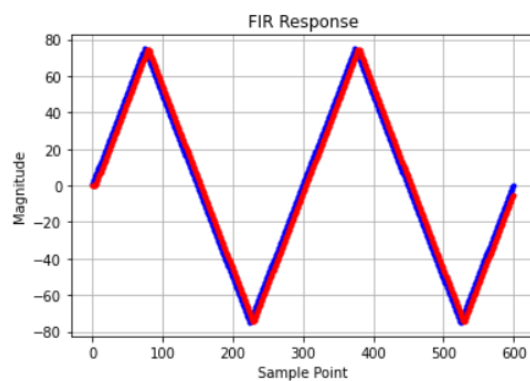
2.2.4. Synthesis/Placement/Routing/Generate Bit-stream

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

2.3. Python Code Validation via Jupyter Notebook

同 Lab.#1，請自行參照 Workbook 1 說明步驟操作。

```
Entry: /usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py
System argument(s): 3
Start of "/usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py"
Kernel execution time: 0.000820159912109375 s
```



```
=====
Exit process
```