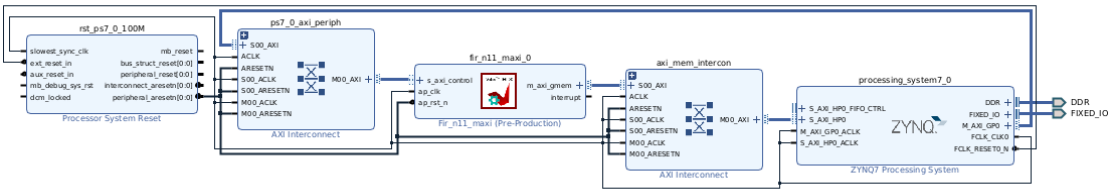
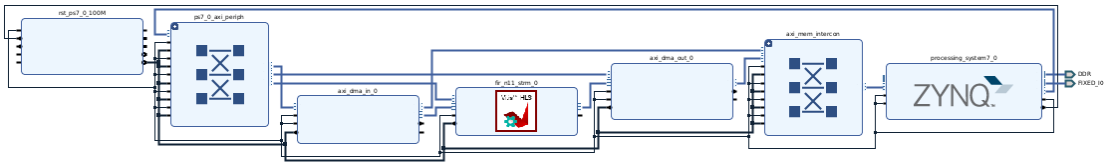
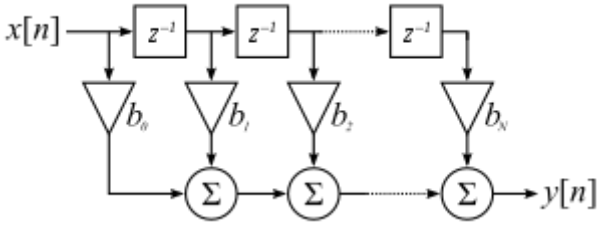
Lab2 Report

1. Introduction about the overall system







Lab2 為 FIR(Finite Impulse Response)在兩個不同的 port level protocol(m_axi、axis)的硬體(PYNQ Z2)實作。

透過 Vitis_hls 撰寫 pragma 或設定 directive 來定義輸入與輸出透過何種 program I/O 來傳輸，並使用先前寫好的 testbench 對於主程式進行 C simulation。在 Simulation 後，進行 Synthesis，觀察在主視窗跳出的 synthesis report 並分析其 timing 和 area 的 information。接著進行 C 和 RTL 的 Co-simulation 觀察波型，並輸出 RTL code 供 Vivado 進行下一步動作。

由 Vitis_hls 所匯出的 RTL code 匯入 Vivado，並對其進行 block

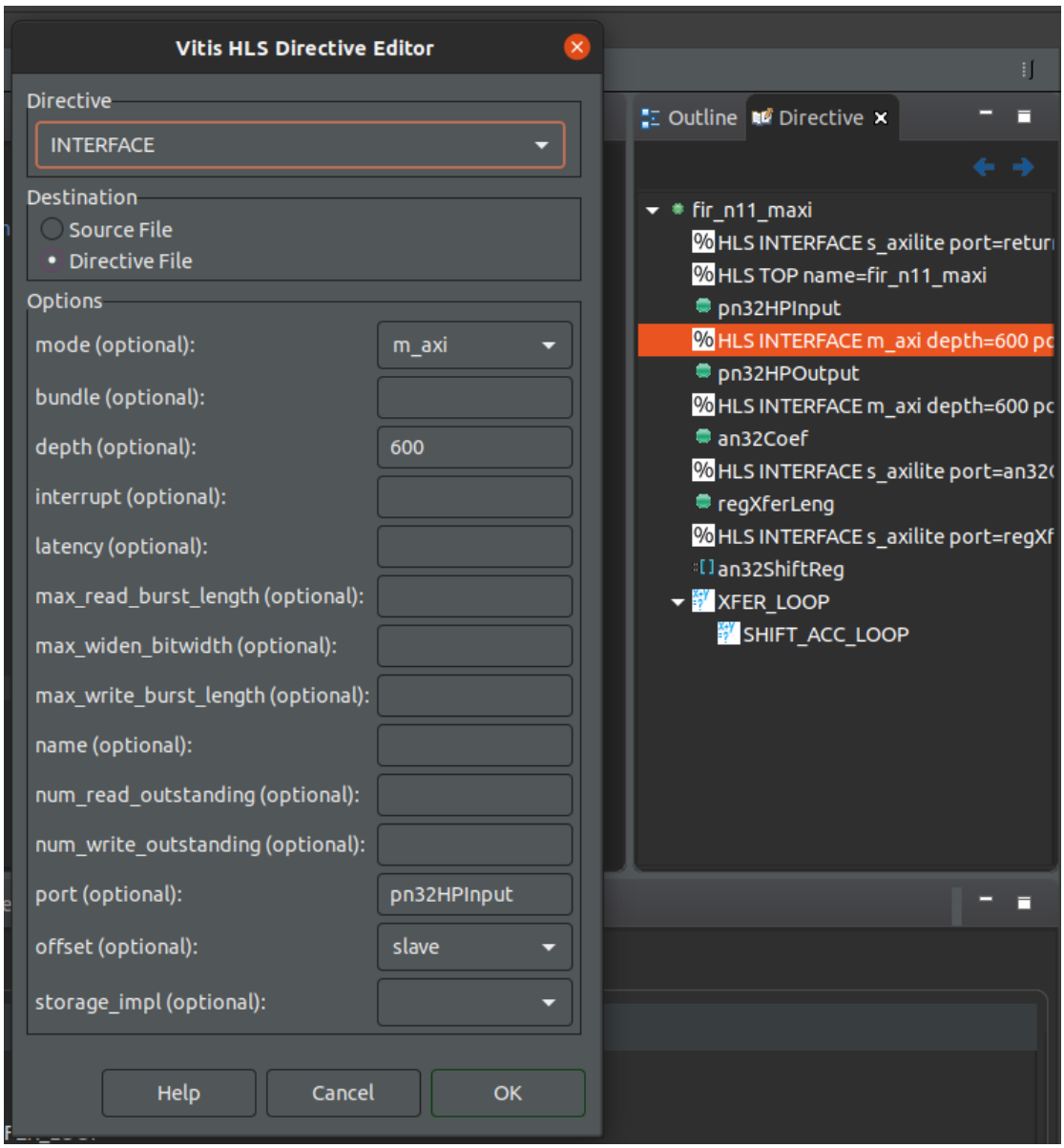
design。加入在 Vitis_hls 設定好的做完 Kernel 端和 ZYNQ7 Processing System 作為 host 端，在不同的 port level protocol 中適當加入需要的硬體資源(如在 axis 部分加入 AXI Directed Memory Address)並透過 auto connection 完成 block design。進行完 block design 後，對其進行 wrapper 以利後續 Bitstream 的生成。最後將 Bitstream 輸出的.bit 檔、.hwh 檔和 host program 上傳至 OnlineFPGA 的 Jupyter Notebook 中，在 PYNQ 上進行運算，並與先前 Vitis_hls 的 C Simulation 結果進行比較。

2. Observation & Learning

(1) Vitis_hls

1. 程式碼理解

在 kernel code 部分，兩個不同的 port level protocol 的行為皆相同，皆為在第幾個的輸出為當下時序前每一個輸入乘上每一個輸入對應到的權重的總和。而在 port level protocol 部分，兩個定義不同。在 m_axi 部分，其透過 directive 方式對於 top module 和 I/O 的 interface 進行定義，而在 axis 部分，則是使用 pragma 如 Lab1 的方式進行定義。



圖(1) m_axi 定義 port level protocol 方式(directive)

```
6  #pragma HLS INTERFACE s_axilite port=regXferLeng
7  #pragma HLS INTERFACE s_axilite port=an32Coef
8  #pragma HLS INTERFACE axis register both port=pstrmOutput
9  #pragma HLS INTERFACE axis register both port=pstrmInput
10 #pragma HLS INTERFACE s_axilite port=return
```

圖(2) axis 定義 port level protocol 方式(pragma)

	m_axi	**axis**
pn32HPInput/pstrmInput	m_axi depth = 600 offset:slave	axis register mode : both
pn32HPOutput/pstrmOutput	m_axi depth = 600 offset:slave	axis register mode : both
an32Coef	s_axilite	s_axilite
regXferLeng	s_axilite	s_axilite

表(1) 兩個不同 program I/O 所使用所有的 Interface

以上為本次實驗中兩個不同 program I/O 所使用所有的 Interface 設定。在 testbench 部分匯入標頭檔後，將輸入信號寫入 input array 當中供代入 top module 產生資料所用。

在將輸入寫入 input array 部分，透過兩者的 testbench 可以分別觀察到兩個不同的 protocol 儲存資料方式的不同。同樣都是將輸入信號寫入 array 中，axis 部分需多紀錄 valDataCtrl 的訊號，當 index > 75 時，valDataCtrl 為 1。

有了 input array 後，可將 input、tabs 和要傳送的資料長度引入 top module，運算後的結果將寫入 out.dat，最後將 out.dat 與已知的 out_gold.dat 內所有資料比對得到結果，若完全相同，則印出”Test Passed!”，否則印出”Test Failed!”。

2. C Simulation

在了解程式碼的運作方式後對於其進行 C Simulation，可透過 testbench 對於 C code 進行 behavior 的驗證，如下兩張圖所示。

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
>> Start test!
>> Comparing against output data...
>> Test passed!
------------------------
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

圖(3) m_axi 的 C Simulation

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
>> Start test!
>> Comparing against output data...
>> Test passed!
------------------------
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

圖(4) axis 的 C Simulation

3. C Synthesis

在進行完成 C simulation 後，將對於 C code 進行 Synthesis。

Detail:
* Instance:

Instance	Module	BRAM_18K	DSP	FF	LUT	URAM
control_s_axi_U	control_s_axi	0	0	288	436	0
grp_fir_n11_maxi_Pipeline_XFER_LOOP_fu_242	fir_n11_maxi_Pipeline_XFER_LOOP	0	33	2794	1084	0
gmem_m_axi_U	gmem_m_axi	0	0	718	1318	0
Total		0	33	3800	2838	0

圖(5) m_axi 的 Utilization

Detail:
* Instance:

Instance	Module	BRAM_18K	DSP	FF	LUT	URAM
control_s_axi_U	control_s_axi	0	0	154	180	0
grp_fir_n11_strm_Pipeline_XFER_LOOP_fu_112	fir_n11_strm_Pipeline_XFER_LOOP	0	33	2834	1153	0
Total		0	33	2988	1333	0

圖(6)axis 的 Utilization

在 Utilization 部分，m_axi 部分共使用了 33 個 DSP(Digital System Processing)、3800 個 FF(Flip-Flop)和 2838 個 LUT(Look Up Table)來建構此運算需要的 resource。在 axis 部分， 共使用了 33 個 DSP(Digital System Processing)、2988 個 FF(Flip-Flop)和 1333 個 LUT(Look Up Table)來建構此運算需要的 resource。由結果可推論，在運算部分(DSP)，兩者使用的硬體資源相同，而在記憶體存儲部分，axis 在硬體資源的使用上相較 m_axi 可以更加節省。

```
* S_AXILITE Registers
+----------------+-----------------+---------+-------+---------+------------------------------+
| Interface      | Register        | Offset  | Width | Access  | Description                  |
+----------------+-----------------+---------+-------+---------+------------------------------+
| s_axi_control  | pn32HPInput_1   | 0x10    | 32    | W       | Data signal of pn32HPInput   |
| s_axi_control  | pn32HPInput_2   | 0x14    | 32    | W       | Data signal of pn32HPInput   |
| s_axi_control  | pn32HPOutput_1  | 0x1c    | 32    | W       | Data signal of pn32HPOutput  |
| s_axi_control  | pn32HPOutput_2  | 0x20    | 32    | W       | Data signal of pn32HPOutput  |
| s_axi_control  | regXferLeng     | 0x28    | 32    | W       | Data signal of regXferLeng   |
+----------------+-----------------+---------+-------+---------+------------------------------+
```

圖(7) m_axi 的 s_axilite configuration register

```
* S_AXILITE Registers
+---------------+------------+--------+-------+--------+-------------------------------+-----------------------------------------------------------------------------+
| Interface     | Register   | Offset | Width | Access | Description                   | Bit Fields                                                                  |
+---------------+------------+--------+-------+--------+-------------------------------+-----------------------------------------------------------------------------+
| s_axi_control | CTRL       | 0x00   | 32    | RW     | Control signals               | 0=AP_START 1=AP_DONE 2=AP_IDLE 3=AP_READY 7=AUTO_RESTART 9=INTERRUPT        |
| s_axi_control | GIER       | 0x04   | 32    | RW     | Global Interrupt Enable Register | 0=Enable                                                                    |
| s_axi_control | IP_IER     | 0x08   | 32    | RW     | IP Interrupt Enable Register  | 0=CHAN0_INT_EN 1=CHAN1_INT_EN                                               |
| s_axi_control | IP_ISR     | 0x0c   | 32    | RW     | IP Interrupt Status Register  | 0=CHAN0_INT_ST 1=CHAN1_INT_ST                                               |
| s_axi_control | regXferLeng| 0x10   | 32    | W      | Data signal of regXferLeng    |                                                                             |
+---------------+------------+--------+-------+--------+-------------------------------+-----------------------------------------------------------------------------+
```

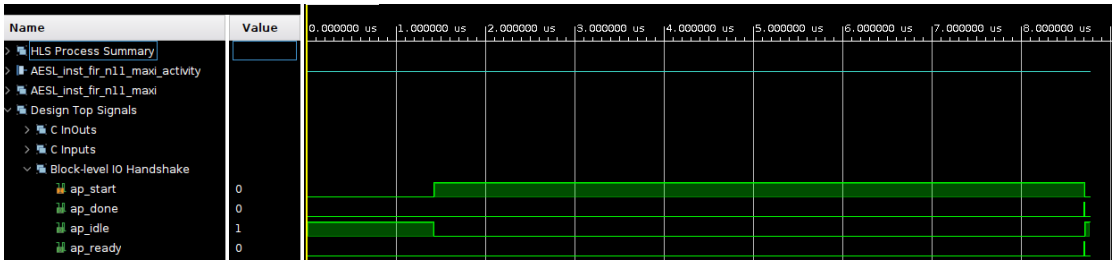
圖(8) m_axi 的 s_axilite configuration register

在 C synthesis 的部分，也可以觀察兩個不同 program I/O 所使用 Interface 在合成出的 s_axilite configuration register 差異，如圖(7)和圖(8)所示。

4. C & RTL Co-simulation

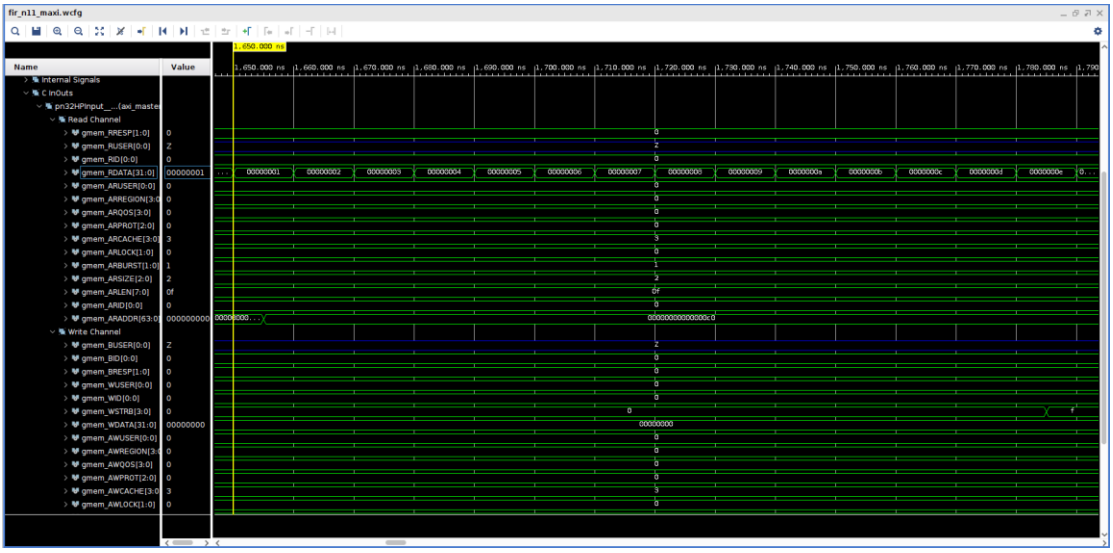
(1) m_axi

進行 C 和 RTL 的 Co-simulation 時，可得整個系統的 Waveform，如下圖所示。

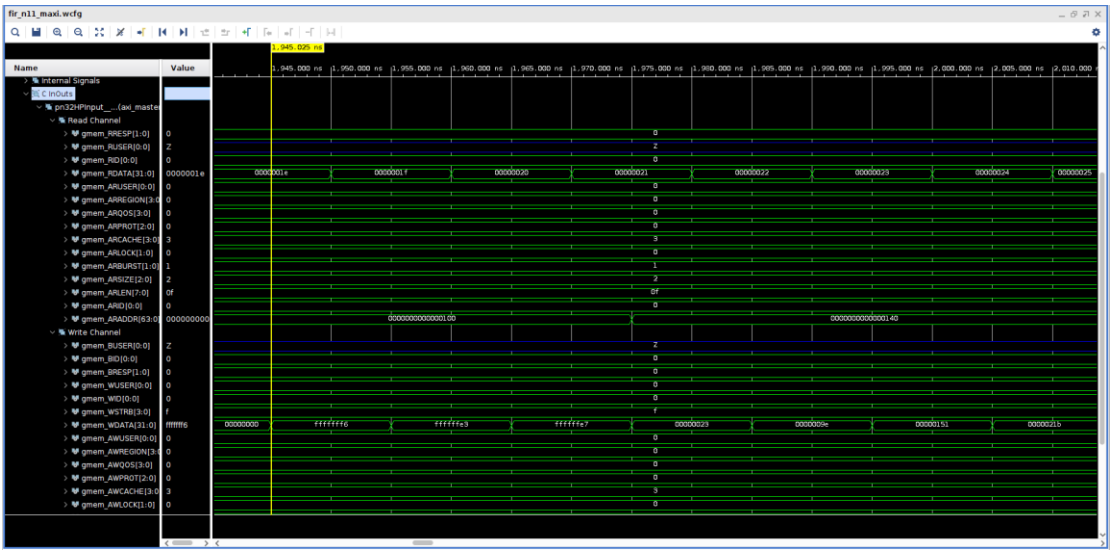


圖(10) block level protocol of m_axi

圖(10)為 m_axi 的 block level protocol 的 Waveform，綠色波型依序為 ap_start、ap_done、ap_idle、ap_ready。當 ap_start 由 0 抬升至 1，ap_idle 離開 ap_idle 狀態由 1 轉為 0。當 ap_ready 由 0 抬升至 1，ap_done 也隨之升起，此時 Master 端與 Slave 端開始 handshake。而當 ap_ready 下降至 0 時，ap_start 與 ap_done 隨之下降至 0，ap_idle 則抬升至 1 等待下一次 handshake，並且整個測試結束。

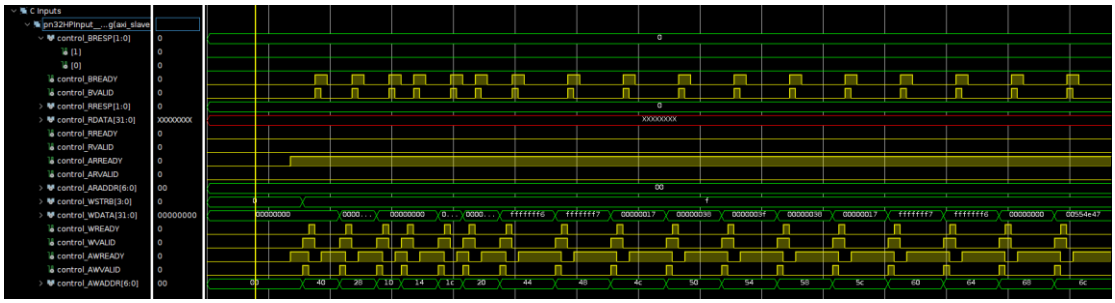


圖(11) axi_master of m_axi (1)



圖(12) axi_master of m_axi (1)

圖(11)為 axi_master 讀寫資料的波型圖。可觀察到在 1650ns 時，axi_master 每 10ns 讀取一筆資料，並資料由 000000 讀取至 ffffff，而在寫入部分則是在 1945ns 開始寫入，axi_master 每 10ns 寫入一筆資料，然而寫入無特定順序。



圖(13) port level protocol of m_axi

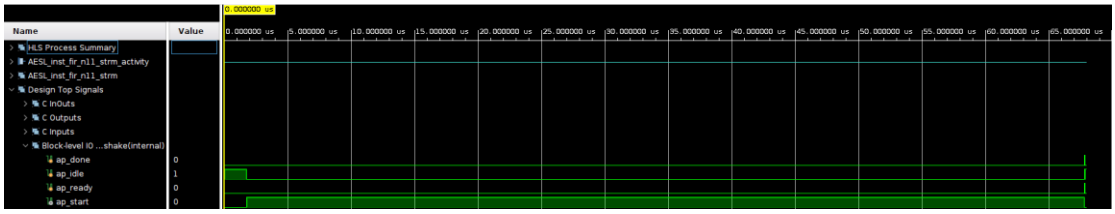
圖(13)為 port level protocol 的 Waveform，可觀察到在 ARVALID 部分都為 0，故都沒有讀取任何的資料且可觀察到 RDATA 都為高阻抗。在寫入部分，可觀察到 AWREADY 先由 0 抬升至 1 等待 AWVALID 為 1 準備 handshake，並在 AWVALID 由 0 抬升至 1 開始 handshake 得到要寫入的 address。WVALID 部分，則是先由 0 抬升至 1 等待得到要寫入的 address。當 AWVALID 和 AWREADY 都變成 negedge 時，WREADY 由 0 抬升至 1 開始寫入資料。完成資料寫入後，WVALID 和 WREADY 降至 0。

Modules & Loops	Avg II	Max II	Min II	Avg Latency	Max Latency	Min Latency
fir_n11_maxi				728	728	728
fir_n11_maxi_Pipeline_XFER_LOOP				715	715	715

圖(14) Latency of m_axi

上圖為 m_axi 的 Co-sim 結果，得到平均延遲、最小延遲和最大延遲如圖(14)。

(2) axis

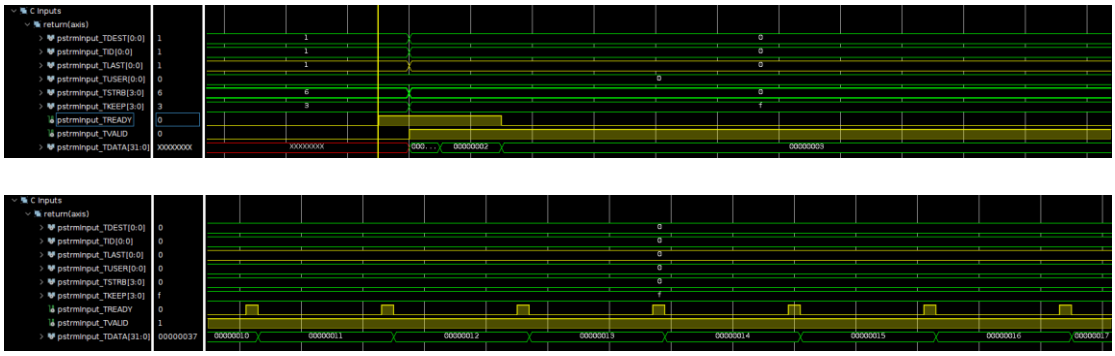


圖(15) block level protocol of axis

圖(15)為 axis 的 block level protocol 的 Waveform，綠色波型依序為 ap_start、ap_done、ap_idle、ap_ready。當 ap_start 由 0 抬升至 1，ap_idle 離開 ap_idle 狀態由 1 轉為 0。當 ap_ready 由 0 抬升至 1，ap_done 也隨之升起，此時 Master 端與 Slave 端開始 handshake。而當 ap_ready 下降至 0 時，ap_start 與 ap_done 隨之下降至 0，ap_idle 則抬升至 1 等待下一次 handshake，並且整個測試結束，與 m_axi 的波型近乎完全相同，差別在於完成整體測試時間。

圖(16)　output of axis

圖(16)為 axis output 的相關資訊，可觀察到當 t = 1.925us 時，資料開始輸出，並且每 0.1us 輸出一筆資料直至資料完整輸出。



圖(17) input of axis

圖(17)為 axis input 的相關資訊，可觀察到起始時 TREADY 先由 0 抬升至 1，等待 TVALID 為 1 做資料的 handshake，如同老師上課所述

的前面有著 FIFO Buffer 存放資料，使得 TREADY 可以叫 TVALID 先抬升至 1。由圖(17)下半部分可觀察出當完成輸入一筆資料時，TREADY 就會有一個 pluse，並且每 0.11us 輸入一筆資料。

Modules & Loops	Avg II	Max II	Min II	Avg Latency	Max Latency	Min Latency
fir_n11_strm				6606	6606	6606
fir_n11_strm_Pipeline_XFER_LOOP				6603	6603	6603

圖(18) Latency of axis

上圖為 axis 的 Co-sim 結果，得到平均延遲、最小延遲和最大延遲如圖(18)。

5. Export RTL

在完成 C 和 RTL 的 Co-simulation 後，匯出 RTL code 供 Vivado 進一步分析使用。

(2) Vivado

由 Vitis_hls 匯出的 RTL code，在 Vivado 中可建立其 block design。在 block design 中，加入 Vitis_hls 設定好的做完 Kernel 端和 ZYNQ7 Processing System 作為 host 端，在不同的 port level protocol 中適當加入需要的硬體資源(如在 axis 部分加入 AXI Directed Memory Address)並透過 auto connection 完成 block design。進行完 block design 後，對其進行 wrapper 以利後續 Bitstream 的生成。

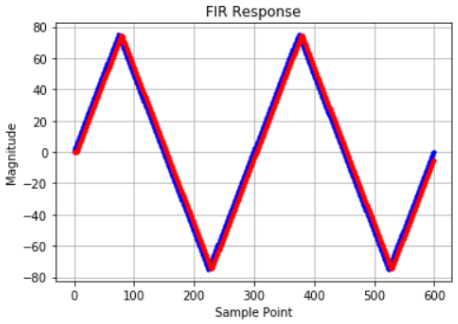
(3) Jupyter Notebook

由 Vivado 輸出的 BitStream 中的.bit 檔、.hwh 檔與寫好的.ipynb 檔，將其上傳至 OnlineFPGA 的 Jupyter Notebook 中並在 PYZQ Z2 上進行運算得到結果。

在.ipynb 檔中，須將 axi_dma_out_1 改為在 Vivado block design 階段我們所設計的 AXI Direct Memory Address(axi_dma_out_0)，否則會編譯失敗。

(a) m_axi

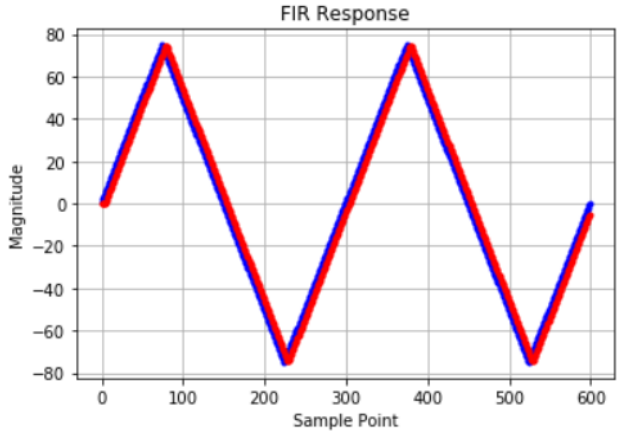
```
Entry: /usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py
System argument(s): 3
Start of "/usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py"
Kernel execution time: 0.0006015300750732422 s
```




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

(b) axis

```
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: 0.0016248226165771484 s
```




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

3. Screen dump

(1) Performance

(a) m_axi

```
+ Timing:
    * Summary:
    +--------+----------+----------+------------+
    |  Clock |  Target  | Estimated| Uncertainty|
    +--------+----------+----------+------------+
    |ap_clk  |  10.00 ns|  7.300 ns|     2.70 ns|
    +--------+----------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+---------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
    |   min   |   max   |    min   |    max   | min | max |   Type  |
    +---------+---------+----------+----------+-----+-----+---------+
    |        ?|        ?|         ?|         ?|    ?|    ?|       no|
    +---------+---------+----------+----------+-----+-----+---------+

    + Detail:
        * Instance:
        +-----------------------------------------------+---------------------------------+---------+---------+---------+---------+-----+-----+---------+
        |                                               |                                 |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
        |                   Instance                    |              Module             |   min   |   max   |    min  |    max  | min | max |   Type  |
        +-----------------------------------------------+---------------------------------+---------+---------+---------+---------+-----+-----+---------+
        |grp_fir_n11_maxi_Pipeline_XFER_LOOP_fu_242  |fir_n11_maxi_Pipeline_XFER_LOOP  |        ?|        ?|         ?|         ?|    ?|    ?|       no|
        +-----------------------------------------------+---------------------------------+---------+---------+---------+---------+-----+-----+---------+
```

(b) axis

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+----------+----------+------------+
    |  Clock |  Target  | Estimated| Uncertainty|
    +--------+----------+----------+------------+
    |ap_clk  |  10.00 ns|  6.923 ns|     2.70 ns|
    +--------+----------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+---------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
    |   min   |   max   |    min   |    max   | min | max |   Type  |
    +---------+---------+----------+----------+-----+-----+---------+
    |        ?|        ?|         ?|         ?|    ?|    ?|       no|
    +---------+---------+----------+----------+-----+-----+---------+

    + Detail:
        * Instance:
        +-----------------------------------------+---------------------------------+---------+---------+----------+----------+-----+-----+---------+
        |                                         |                                 |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
        |                 Instance                |              Module             |   min   |   max   |    min   |    max   | min | max |   Type  |
        +-----------------------------------------+---------------------------------+---------+---------+----------+----------+-----+-----+---------+
        |grp_fir_n11_strm_Pipeline_XFER_LOOP_fu_112  |fir_n11_strm_Pipeline_XFER_LOOP  |        ?|        ?|         ?|         ?|    ?|    ?|       no|
        +-----------------------------------------+---------------------------------+---------+---------+----------+----------+-----+-----+---------+

        * Loop:
        N/A
```

(2) Utilization

(a) m_axi

== Utilization Estimates

* Summary:

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	40	-
FIFO	-	-	-	-	-
Instance	0	33	3800	2838	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	175	-
Register	-	-	650	-	-
Total	0	33	4450	3053	0
Available	280	220	106400	53200	0
Utilization (%)	0	15	4	5	0

+ Detail:

    * Instance:

Instance	Module	BRAM_18K	DSP	FF	LUT	URAM
control_s_axi_U	control_s_axi	0	0	288	436	0
grp_fir_n11_maxi_Pipeline_XFER_LOOP_fu_242	fir_n11_maxi_Pipeline_XFER_LOOP	0	33	2794	1084	0
gmem_m_axi_U	gmem_m_axi	0	0	718	1318	0
Total		0	33	3800	2838	0

    * DSP:
    N/A

    * Memory:
    N/A

    * FIFO:
    N/A

    * Expression:

Variable Name	Operation	DSP	FF	LUT	Bitwidth P0	Bitwidth P1
add_ln16_fu_289_p2	+	0	0	40	33	2
Total		0	0	40	33	2

    * Multiplexer:

Name	LUT	Input Size	Bits	Total Bits
an32Coef_address0	65	12	4	48
ap_NS_fsm	65	15	1	15
gmem_ARVALID	9	2	1	2
gmem_AWVALID	9	2	1	2
gmem_BREADY	9	2	1	2
gmem_RREADY	9	2	1	2
gmem_WVALID	9	2	1	2
Total	175	37	10	73

    * Register:

Name	FF	LUT	Bits	Const Bits
an32Coef_load_10_reg_446	32	0	32	0
an32Coef_load_1_reg_340	32	0	32	0
an32Coef_load_2_reg_350	32	0	32	0
an32Coef_load_3_reg_360	32	0	32	0
an32Coef_load_4_reg_370	32	0	32	0
an32Coef_load_5_reg_380	32	0	32	0
an32Coef_load_6_reg_390	32	0	32	0
an32Coef_load_7_reg_400	32	0	32	0
an32Coef_load_8_reg_410	32	0	32	0
an32Coef_load_9_reg_420	32	0	32	0
an32Coef_load_reg_330	32	0	32	0
ap_CS_fsm	14	0	14	0
grp_fir_n11_maxi_Pipeline_XFER_LOOP_fu_242_ap_start_reg	1	0	1	0
lshr_ln16_cast_reg_440	31	0	31	0
pn32HPInput_read_reg_435	64	0	64	0
pn32HPOutput_read_reg_430	64	0	64	0
trunc_ln18_1_reg_451	62	0	62	0
trunc_ln30_1_reg_456	62	0	62	0
Total	650	0	650	0

(b) axis

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-----+--------+-------+-----+
|       Name      | BRAM_18K| DSP |   FF   |  LUT  | URAM|
+-----------------+---------+-----+--------+-------+-----+
|DSP              |        -|    -|       -|      -|    -|
|Expression       |        -|    -|       0|     42|    -|
|FIFO             |        -|    -|       -|      -|    -|
|Instance         |        0|   33|    2988|   1333|    -|
|Memory           |        -|    -|       -|      -|    -|
|Multiplexer      |        -|    -|       -|     34|    -|
|Register         |        -|    -|      36|      -|    -|
+-----------------+---------+-----+--------+-------+-----+
|Total            |        0|   33|    3024|   1409|    0|
+-----------------+---------+-----+--------+-------+-----+
|Available        |      280|  220|  106400|  53200|    0|
+-----------------+---------+-----+--------+-------+-----+
|Utilization (%)  |        0|   15|       2|      2|    0|
+-----------------+---------+-----+--------+-------+-----+

+ Detail:
    * Instance:
    +-----------------------------------------+--------------------------------+---------+----+------+------+-----+
    |                 Instance                |             Module             | BRAM_18K| DSP|  FF  |  LUT | URAM|
    +-----------------------------------------+--------------------------------+---------+----+------+------+-----+
    |control_s_axi_U                          |control_s_axi                   |        0|   0|   154|   180|    0|
    |grp_fir_n11_strm_Pipeline_XFER_LOOP_fu_112  |fir_n11_strm_Pipeline_XFER_LOOP  |        0|  33|  2834|  1153|    0|
    +-----------------------------------------+--------------------------------+---------+----+------+------+-----+
    |Total                                    |                                |        0|  33|  2988|  1333|    0|
    +-----------------------------------------+--------------------------------+---------+----+------+------+-----+

    * DSP:
    N/A

    * Memory:
    N/A

    * FIFO:
    N/A

    * Expression:
    +-----------------------------------------------------------+----------+---+---+----+------------+------------+
    |                      Variable Name                        | Operation| DSP| FF| LUT| Bitwidth P0| Bitwidth P1|
    +-----------------------------------------------------------+----------+---+---+----+------------+------------+
    |ret_V_fu_171_p2                                            |         +|   0|  0|  40|          33|           2|
    |grp_fir_n11_strm_Pipeline_XFER_LOOP_fu_112_pstrmOutput_TREADY  |       and|   0|  0|   2|           1|           1|
    +-----------------------------------------------------------+----------+---+---+----+------------+------------+
    |Total                                                      |          |   0|  0|  42|          34|           3|
    +-----------------------------------------------------------+----------+---+---+----+------------+------------+

    * Multiplexer:
    +--------------------------------+----+-----------+-----+-----------+
    |              Name              | LUT| Input Size| Bits| Total Bits|
    +--------------------------------+----+-----------+-----+-----------+
    |ap_NS_fsm                       |  25|          5|    1|          5|
    |pstrmInput_TREADY_int_regslice  |   9|          2|    1|          2|
    +--------------------------------+----+-----------+-----+-----------+
    |Total                           |  34|          7|    2|          7|
    +--------------------------------+----+-----------+-----+-----------+

    * Register:
    +-----------------------------------------------------+----+----+-----+-----------+
    |                         Name                        | FF | LUT| Bits| Const Bits|
    +-----------------------------------------------------+----+----+-----+-----------+
    |ap_CS_fsm                                            |   4|   0|    4|          0|
    |grp_fir_n11_strm_Pipeline_XFER_LOOP_fu_112_ap_start_reg  |   1|   0|    1|          0|
    |tmp_reg_187                                          |  31|   0|   31|          0|
    +-----------------------------------------------------+----+----+-----+-----------+
    |Total                                                |  36|   0|   36|          0|
    +-----------------------------------------------------+----+----+-----+-----------+
```

(3) Interface

(a) m_axi

```
================================================================
== Interface
================================================================
* Summary:
+-----------------------+-----+-----+------------+--------------+--------------+
|       RTL Ports       | Dir | Bits|  Protocol  | Source Object|    C Type    |
+-----------------------+-----+-----+------------+--------------+--------------+
|s_axi_control_AWVALID  |   in|    1|       s_axi|       control|         array|
|s_axi_control_AWREADY  |  out|    1|       s_axi|       control|         array|
|s_axi_control_AWADDR   |   in|    7|       s_axi|       control|         array|
|s_axi_control_WVALID   |   in|    1|       s_axi|       control|         array|
|s_axi_control_WREADY   |  out|    1|       s_axi|       control|         array|
|s_axi_control_WDATA    |   in|   32|       s_axi|       control|         array|
|s_axi_control_WSTRB    |   in|    4|       s_axi|       control|         array|
|s_axi_control_ARVALID  |   in|    1|       s_axi|       control|         array|
|s_axi_control_ARREADY  |  out|    1|       s_axi|       control|         array|
|s_axi_control_ARADDR   |   in|    7|       s_axi|       control|         array|
|s_axi_control_RVALID   |  out|    1|       s_axi|       control|         array|
|s_axi_control_RREADY   |   in|    1|       s_axi|       control|         array|
|s_axi_control_RDATA    |  out|   32|       s_axi|       control|         array|
|s_axi_control_RRESP    |  out|    2|       s_axi|       control|         array|
|s_axi_control_BVALID   |  out|    1|       s_axi|       control|         array|
|s_axi_control_BREADY   |   in|    1|       s_axi|       control|         array|
|s_axi_control_BRESP    |  out|    2|       s_axi|       control|         array|
|ap_clk                 |   in|    1|  ap_ctrl_hs|  fir_n11_maxi|  return value|
|ap_rst_n               |   in|    1|  ap_ctrl_hs|  fir_n11_maxi|  return value|
|ap_start               |   in|    1|  ap_ctrl_hs|  fir_n11_maxi|  return value|
|ap_done                |  out|    1|  ap_ctrl_hs|  fir_n11_maxi|  return value|
|ap_idle                |  out|    1|  ap_ctrl_hs|  fir_n11_maxi|  return value|
|ap_ready               |  out|    1|  ap_ctrl_hs|  fir_n11_maxi|  return value|
|m_axi_gmem_AWVALID     |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWREADY     |   in|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWADDR      |  out|   64|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWID        |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWLEN       |  out|    8|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWSIZE      |  out|    3|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWBURST     |  out|    2|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWLOCK      |  out|    2|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWCACHE     |  out|    4|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWPROT      |  out|    3|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWQOS       |  out|    4|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWREGION    |  out|    4|       m_axi|          gmem|       pointer|
|m_axi_gmem_AWUSER      |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_WVALID      |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_WREADY      |   in|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_WDATA       |  out|   32|       m_axi|          gmem|       pointer|
|m_axi_gmem_WSTRB       |  out|    4|       m_axi|          gmem|       pointer|
|m_axi_gmem_WLAST       |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_WID         |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_WUSER       |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARVALID     |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARREADY     |   in|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARADDR      |  out|   64|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARID        |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARLEN       |  out|    8|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARSIZE      |  out|    3|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARBURST     |  out|    2|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARLOCK      |  out|    2|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARCACHE     |  out|    4|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARPROT      |  out|    3|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARQOS       |  out|    4|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARREGION    |  out|    4|       m_axi|          gmem|       pointer|
|m_axi_gmem_ARUSER      |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_RVALID      |   in|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_RREADY      |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_RDATA       |   in|   32|       m_axi|          gmem|       pointer|
|m_axi_gmem_RLAST       |   in|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_RID         |   in|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_RUSER       |   in|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_RRESP       |   in|    2|       m_axi|          gmem|       pointer|
|m_axi_gmem_BVALID      |   in|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_BREADY      |  out|    1|       m_axi|          gmem|       pointer|
|m_axi_gmem_BRESP       |   in|    2|       m_axi|          gmem|       pointer|
|m_axi_gmem_BID         |   in|    1|       m_axi|          gmem|       pointer|
```

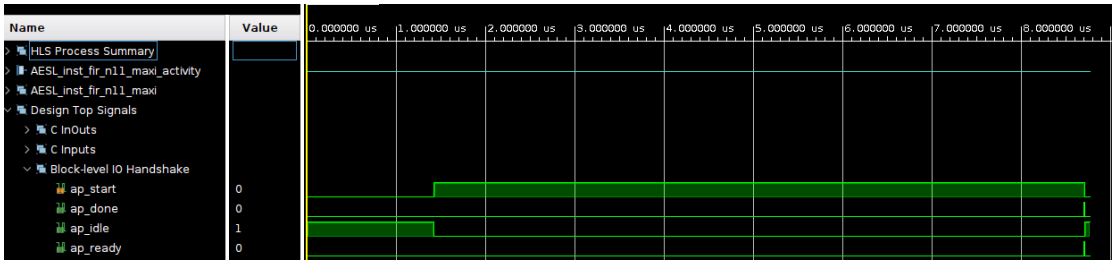
(b) axis

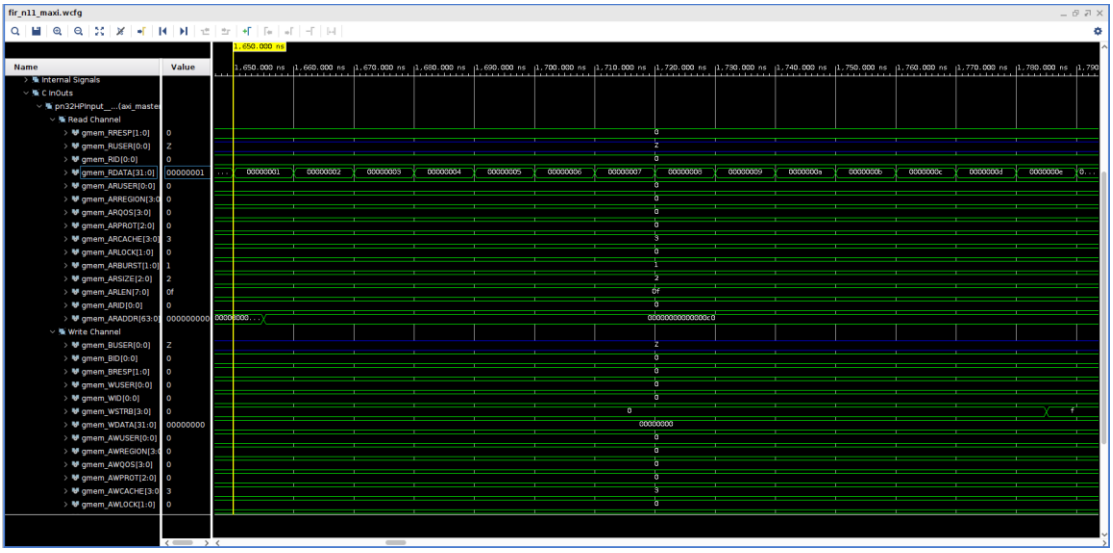
```
================================================================
== Interface
================================================================
* Summary:
+-----------------------+-----+-----+------------+---------------------+--------------+
|       RTL Ports       | Dir | Bits|  Protocol  |     Source Object   |    C Type    |
+-----------------------+-----+-----+------------+---------------------+--------------+
|s_axi_control_AWVALID  |   in|    1|       s_axi|              control|         array|
|s_axi_control_AWREADY  |  out|    1|       s_axi|              control|         array|
|s_axi_control_AWADDR   |   in|    7|       s_axi|              control|         array|
|s_axi_control_WVALID   |   in|    1|       s_axi|              control|         array|
|s_axi_control_WREADY   |  out|    1|       s_axi|              control|         array|
|s_axi_control_WDATA    |   in|   32|       s_axi|              control|         array|
|s_axi_control_WSTRB    |   in|    4|       s_axi|              control|         array|
|s_axi_control_ARVALID  |   in|    1|       s_axi|              control|         array|
|s_axi_control_ARREADY  |  out|    1|       s_axi|              control|         array|
|s_axi_control_ARADDR   |   in|    7|       s_axi|              control|         array|
|s_axi_control_RVALID   |  out|    1|       s_axi|              control|         array|
|s_axi_control_RREADY   |   in|    1|       s_axi|              control|         array|
|s_axi_control_RDATA    |  out|   32|       s_axi|              control|         array|
|s_axi_control_RRESP    |  out|    2|       s_axi|              control|         array|
|s_axi_control_BVALID   |  out|    1|       s_axi|              control|         array|
|s_axi_control_BREADY   |   in|    1|       s_axi|              control|         array|
|s_axi_control_BRESP    |  out|    2|       s_axi|              control|         array|
|ap_clk                 |   in|    1| ap_ctrl_hs|          fir_n11_strm| return value|
|ap_rst_n               |   in|    1| ap_ctrl_hs|          fir_n11_strm| return value|
|interrupt              |  out|    1| ap_ctrl_hs|          fir_n11_strm| return value|
|pstrmInput_TDATA       |   in|   32|       axis|   pstrmInput_V_data_V|       pointer|
|pstrmInput_TVALID      |   in|    1|       axis|   pstrmInput_V_dest_V|       pointer|
|pstrmInput_TREADY      |  out|    1|       axis|   pstrmInput_V_dest_V|       pointer|
|pstrmInput_TDEST       |   in|    1|       axis|   pstrmInput_V_dest_V|       pointer|
|pstrmInput_TKEEP       |   in|    4|       axis|   pstrmInput_V_keep_V|       pointer|
|pstrmInput_TSTRB       |   in|    4|       axis|   pstrmInput_V_strb_V|       pointer|
|pstrmInput_TUSER       |   in|    1|       axis|   pstrmInput_V_user_V|       pointer|
|pstrmInput_TLAST       |   in|    1|       axis|   pstrmInput_V_last_V|       pointer|
|pstrmInput_TID         |   in|    1|       axis|     pstrmInput_V_id_V|       pointer|
|pstrmOutput_TDATA      |  out|   32|       axis|  pstrmOutput_V_data_V|       pointer|
|pstrmOutput_TVALID     |  out|    1|       axis|  pstrmOutput_V_dest_V|       pointer|
|pstrmOutput_TREADY     |   in|    1|       axis|  pstrmOutput_V_dest_V|       pointer|
|pstrmOutput_TDEST      |  out|    1|       axis|  pstrmOutput_V_dest_V|       pointer|
|pstrmOutput_TKEEP      |  out|    4|       axis|  pstrmOutput_V_keep_V|       pointer|
|pstrmOutput_TSTRB      |  out|    4|       axis|  pstrmOutput_V_strb_V|       pointer|
|pstrmOutput_TUSER      |  out|    1|       axis|  pstrmOutput_V_user_V|       pointer|
|pstrmOutput_TLAST      |  out|    1|       axis|  pstrmOutput_V_last_V|       pointer|
|pstrmOutput_TID        |  out|    1|       axis|    pstrmOutput_V_id_V|       pointer|
+-----------------------+-----+-----+------------+---------------------+--------------+
```

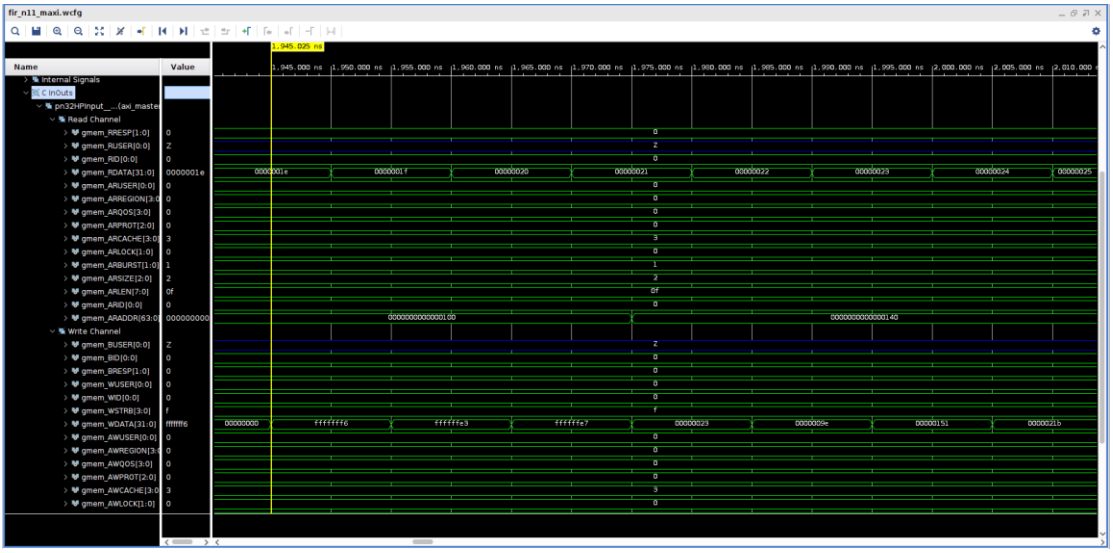
(4) Co-simulation transcript/waveform

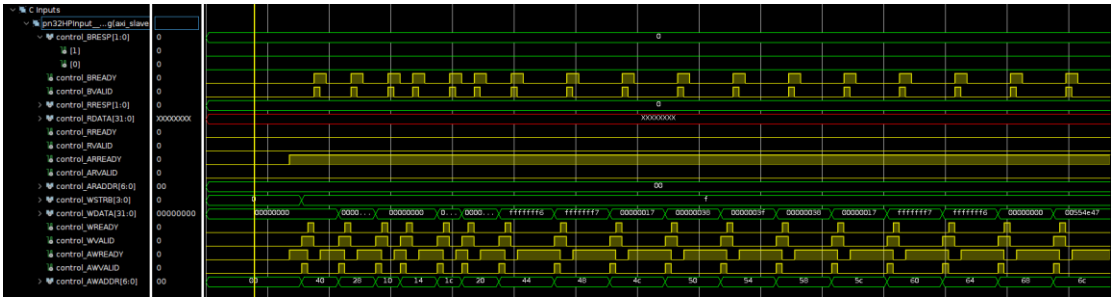
(a) m_axi

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
>> Start test!
>> Comparing against output data...
>> Test passed!
------------------------
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```



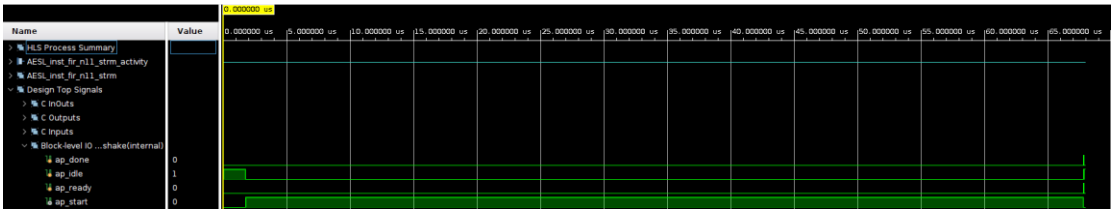


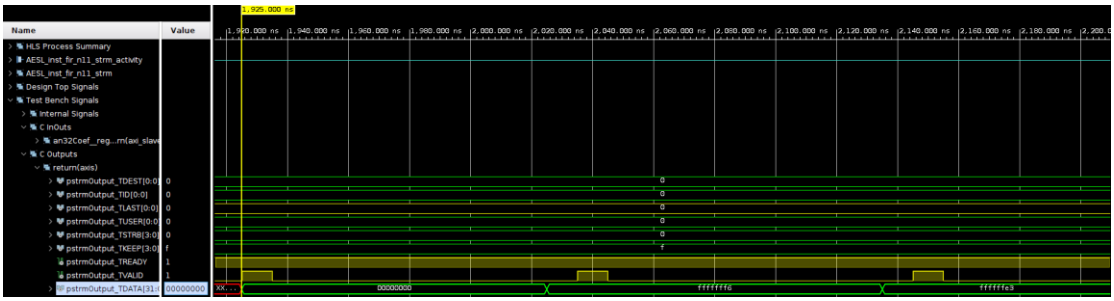


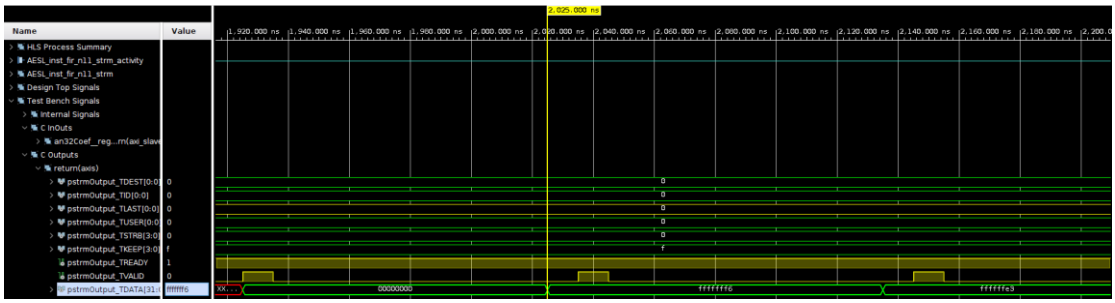


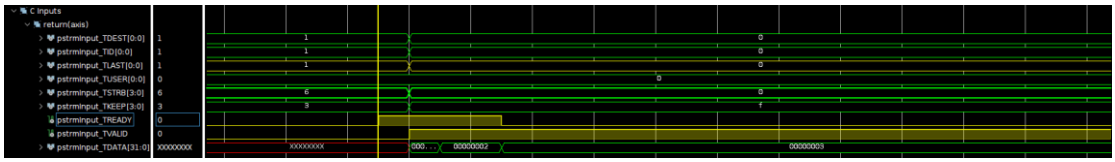
(b) axis

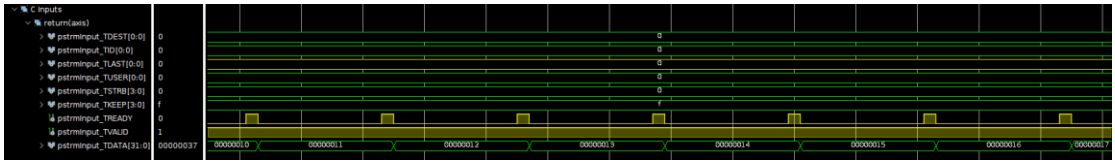
```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
>> Start test!
>> Comparing against output data...
>> Test passed!
------------------------
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```






Name
Value
2,025.000 ns
HLS Process Summary
AESL_inst_fir_n11_strm_activity
AESL_inst_fir_n11_strm
Design Top Signals
Test Bench Signals
Internal Signals
C InOuts
C Outputs
return(axis)
pstrmOutput_TDEST[0:0]
pstrmOutput_TID[0:0]
pstrmOutput_TLAST[0:0]
pstrmOutput_TUSER[0:0]
pstrmOutput_TSTRB[3:0]
pstrmOutput_TKEEP[3:0]
pstrmOutput_TREADY
pstrmOutput_TVALID
pstrmOutput_TDATA[31:0]
ffffffff6
00000000
fffffff6
fffffffa

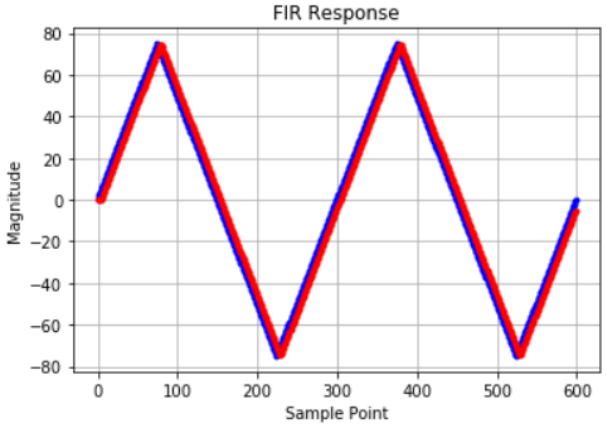

C Inputs
return(axis)
pstrmInput_TDEST[0:0]
pstrmInput_TID[0:0]
pstrmInput_TLAST[0:0]
pstrmInput_TUSER[0:0]
pstrmInput_TSTRB[3:0]
pstrmInput_TKEEP[3:0]
pstrmInput_TREADY
pstrmInput_TVALID
pstrmInput_TDATA[31:0]
XXXXXXXX
00000002
00000003


C Inputs
return(axis)
pstrmInput_TDEST[0:0]
pstrmInput_TID[0:0]
pstrmInput_TLAST[0:0]
pstrmInput_TUSER[0:0]
pstrmInput_TSTRB[3:0]
pstrmInput_TKEEP[3:0]
pstrmInput_TREADY
pstrmInput_TVALID
pstrmInput_TDATA[31:0]
00000037
00000010
00000011
00000012
00000013
00000014
00000015
00000016
00000017

(5) Jupyter Notebook execution results

(c) m_axi

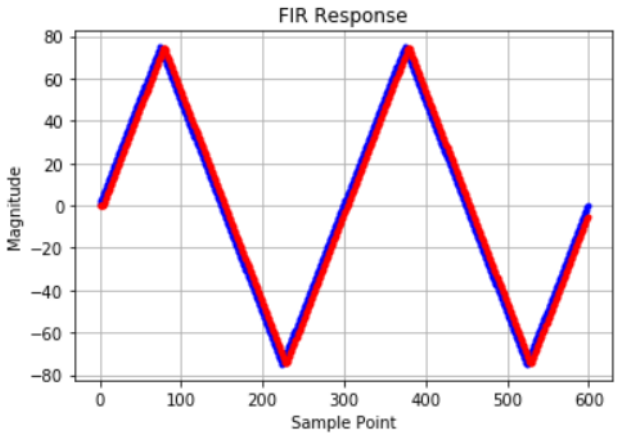
```
Entry: /usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py
System argument(s): 3
Start of "/usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py"
Kernel execution time: 0.0006015300750732422 s
```




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

(d) axis

```
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: 0.0016248226165771484 s
```




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