AAHLS (Lab 2)

1. 簡介

本報告探討 AAHLS（Lab2），內容分為兩個部分。第一部分使用 AXI-M 傳輸數據來實現 FIR，第二部分則改為 AXI-Stream 傳輸數據來實現 FIR。此外，其他介面皆會透過 AXI-Lite 來實現。本次 LAB 使用 KV260(xck26-sfvc784-2LV-c)進行實作，詳細內容會在後面呈現。

2. 報告內容

#1:

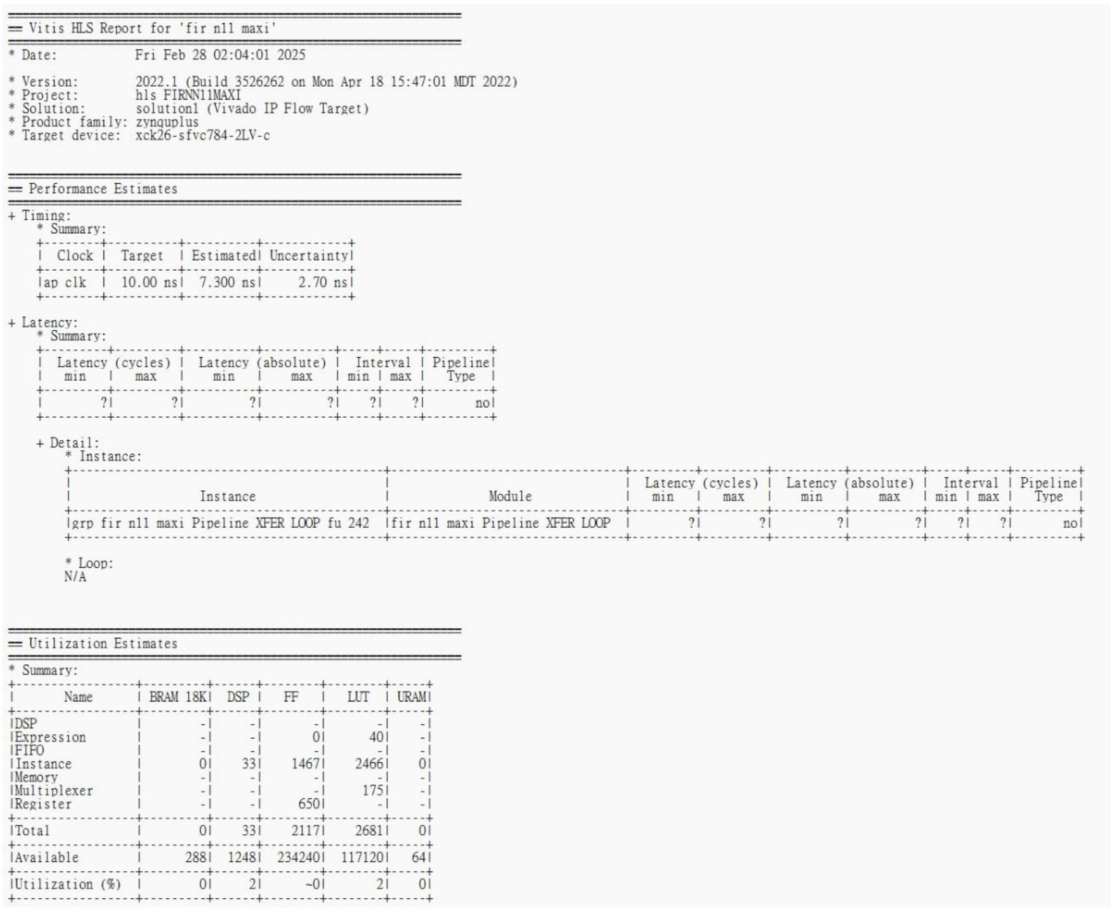
(1) 內容

本次 Lab2 是實作 11taps 的 FIR，其中#1 使用 AXI-Master 作為資料讀寫的 interface，具體硬體架構是使用一個 shift register 將 input data 接到 shift register 中，透過 tap parameters 和 shift register 內部的 data 做 convolution 以實現 FIR。由於此 Lab 著重在 IO 的設定，硬體優化部分，包括 PIPELINE、UNROLL、ARRAY PARTITION 放在 QUESTION 進行回答。這次 HLS 主要有兩個 LOOP，分別是 SHIFT_ACC_LOOP 以及 XFER_LOOP。SHIFT_ACC_LOOP 主要是進行輸出，每完成一次 LOOP 就會輸出一個對應的值，而 XFER_LOOP 只要控制所有輸入 data，完成 XFER_LOOP 代表對所有輸入 data 完成運算。此外，和 Lab1 不同的是要另外定義一個 port=return，並透過 AXI-Lite 控制，此處設定是為了讓 host 能透過 AXI-Lite 控制 kernel。

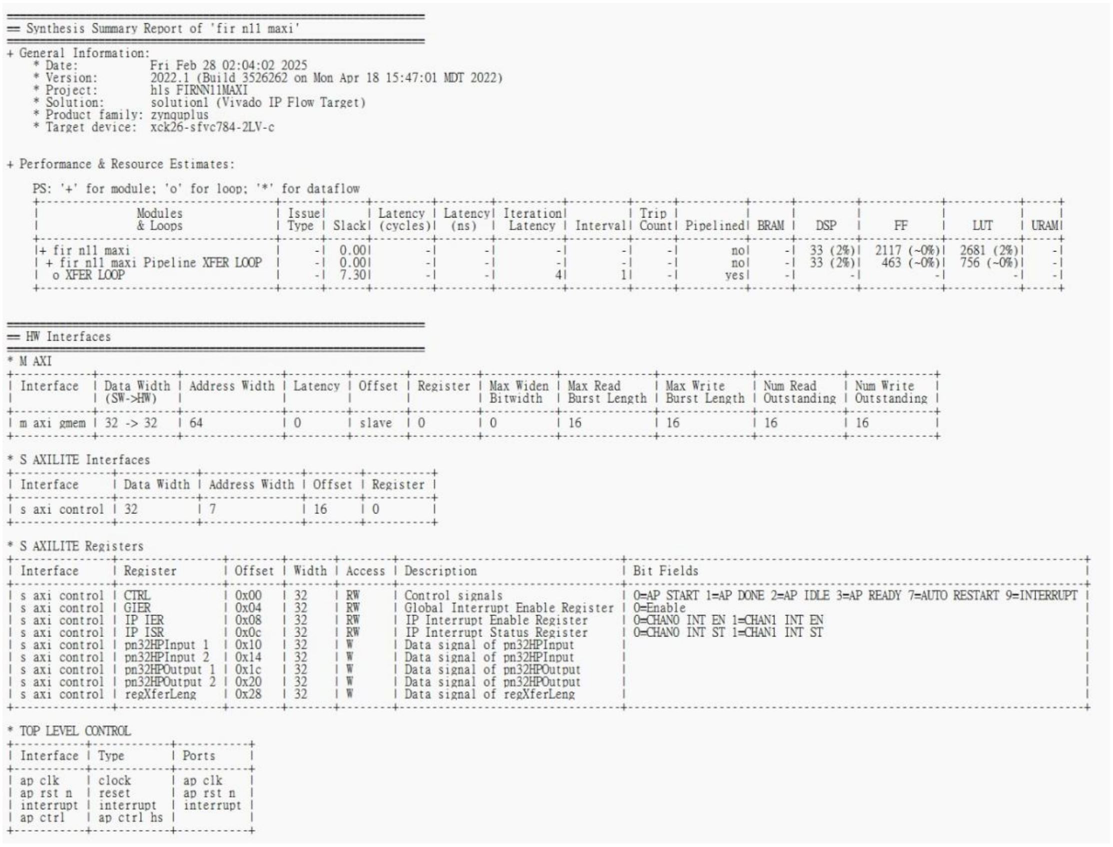
#1 使用的是 AXI-Master 進行資料讀寫，因此需要啟用 Zynq MPSoC 的 Slave HP (High-Performance) Port，使 Kernel 能夠存取 DDR 進行讀寫。這需要在 Block Diagram 中手動啟用 Slave HP Port，完成相關設定後，即可產生 .hwh 和 .bit 檔案，供 Host 端操作。其中，.bit 是 bitstream 檔案，負責透過 fpga_manager 配置 fpga，而 hwh 則包含 mmio、ip 等訊息，方便 python 進行調用 kernel。

在 python code 的部分，和 Lab1 不同的是，此次 Lab 會先將 data 透過 allocate 存入 main memory 中，接著再進行 kernel 的 configuration，在配置過程中也會計算 DC gain 以方便後續進行 normalized。再來會將 main memory 的 data 透過 AXI-Master 傳輸到 kernel，然後開始計算後再將結果一樣透過 AXI-Master 傳回 main memory 並畫出結果。

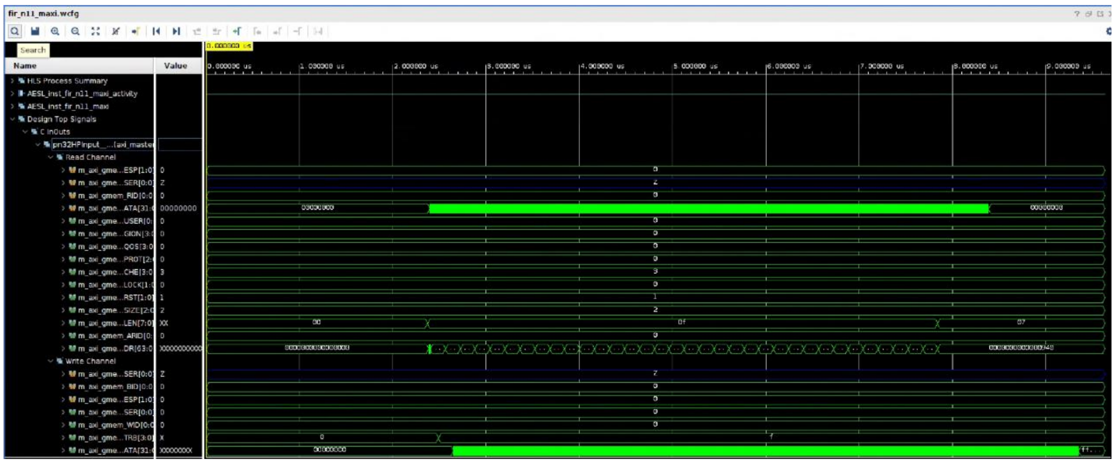
(2) 相關截圖

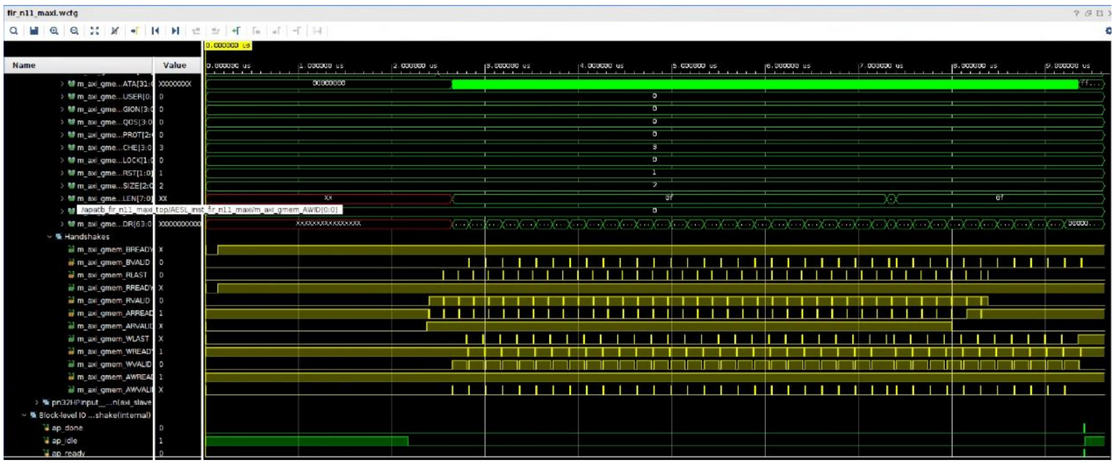


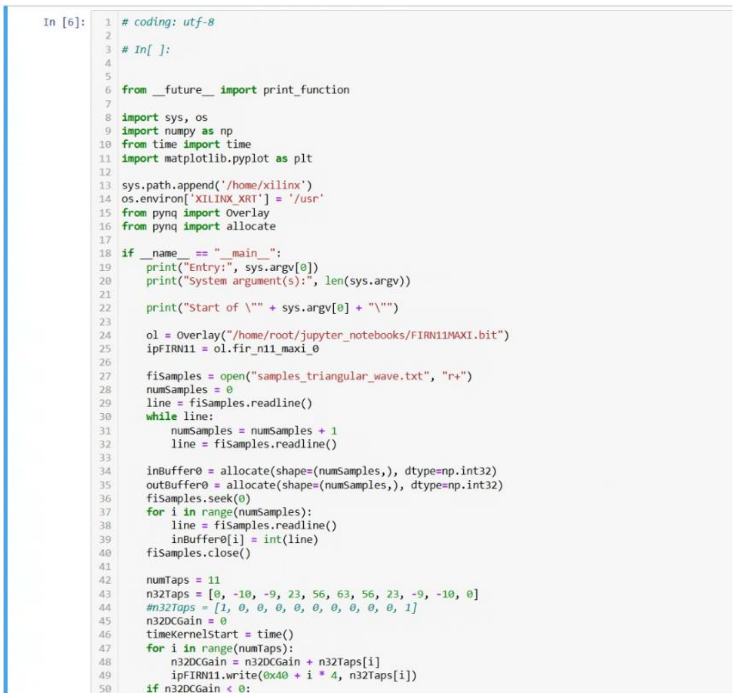
```
================================================================
== Vitis HLS Report for 'fir n11 maxi'
================================================================
* Date:           Fri Feb 28 02:04:01 2025

* Version:        2022.1 (Build 3526262 on Mon Apr 18 15:47:01 MDT 2022)
* Project:        hls FIRNN11MAXI
* Solution:       solution1 (Vivado IP Flow Target)
* Product family: zynquplus
* Target device:  xck26-sfvc784-2LV-c


================================================================
== Performance Estimates
================================================================
+ 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|
        +-----------------------------------------+----------------------------------+---------+---------+----------+----------+-----+-----+---------+

        * Loop:
        N/A


================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+------+--------+--------+-----+
|       Name      | BRAM 18K|  DSP |   FF   |   LUT  | URAM|
+-----------------+---------+------+--------+--------+-----+
|DSP              |        -|     -|       -|       -|    -|
|Expression       |        -|     -|       0|      40|    -|
|FIFO             |        -|     -|       -|       -|    -|
|Instance         |        0|    33|    1467|    2466|    0|
|Memory           |        -|     -|       -|       -|    -|
|Multiplexer      |        -|     -|       -|     175|    -|
|Register         |        -|     -|     650|       -|    -|
+-----------------+---------+------+--------+--------+-----+
|Total            |        0|    33|    2117|    2681|    0|
+-----------------+---------+------+--------+--------+-----+
|Available        |      288|  1248|  234240|  117120|   64|
+-----------------+---------+------+--------+--------+-----+
|Utilization (%)  |        0|     2|     ~0|       2|    0|
+-----------------+---------+------+--------+--------+-----+
```



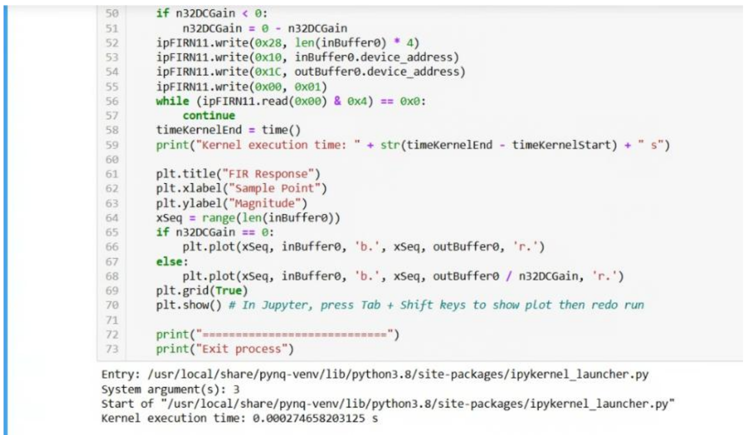
```
================================================================
== Synthesis Summary Report of 'fir n11 maxi'
================================================================
+ General Information:
    * Date:           Fri Feb 28 02:04:02 2025
    * Version:        2022.1 (Build 3526262 on Mon Apr 18 15:47:01 MDT 2022)
    * Project:        hls FIRNN11MAXI
    * Solution:       solution1 (Vivado IP Flow Target)
    * Product family: zynquplus
    * Target device:  xck26-sfvc784-2LV-c


+ Performance & Resource Estimates:

    PS: '+' for module; 'o' for loop; '*' for dataflow
    +--------------------------------+------+------+---------+--------+----------+---------+------+----------+------+---------+-----------+-----------+-----+
    |             Modules            | Issue|      | Latency | Latency| Iteration|         | Trip |          |      |         |           |           |     |
    |             & Loops            | Type | Slack| (cycles)|  (ns)  |  Latency | Interval| Count| Pipelined| BRAM |   DSP   |    FF     |    LUT    | URAM|
    +--------------------------------+------+------+---------+--------+----------+---------+------+----------+------+---------+-----------+-----------+-----+
    |+ fir n11 maxi                  |     -|  0.00|        -|       -|         -|        -|     -|        no|     -|  33 (2%)| 2117 (~0%)| 2681 (2%)|    -|
    | + fir n11 maxi Pipeline XFER LOOP |  -|  0.00|        -|       -|         -|        -|     -|        no|     -|  33 (2%)|  463 (~0%)|  756 (~0%)|    -|
    |  o XFER LOOP                   |     -|  7.30|        -|       -|         4|        1|     -|       yes|     -|        -|          -|          -|    -|
    +--------------------------------+------+------+---------+--------+----------+---------+------+----------+------+---------+-----------+-----------+-----+


================================================================
== HW Interfaces
================================================================
* M AXI
+------------+------------+---------------+---------+--------+----------+-----------+--------------+--------------+-------------+-------------+
| Interface  | Data Width | Address Width | Latency | Offset | Register | Max Widen | Max Read     | Max Write    | Num Read    | Num Write   |
|            | (SW->HW)   |               |         |        |          | Bitwidth  | Burst Length | Burst Length | Outstanding | Outstanding |
+------------+------------+---------------+---------+--------+----------+-----------+--------------+--------------+-------------+-------------+
| m axi gmem | 32 -> 32   | 64            | 0       | slave  | 0        | 0         | 16           | 16           | 16          | 16          |
+------------+------------+---------------+---------+--------+----------+-----------+--------------+--------------+-------------+-------------+

* S AXILITE Interfaces
+---------------+------------+---------------+--------+----------+
| Interface     | Data Width | Address Width | Offset | Register |
+---------------+------------+---------------+--------+----------+
| s axi control | 32         | 7             | 16     | 0        |
+---------------+------------+---------------+--------+----------+

* 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 | 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       |                                                                      |
+---------------+----------------+--------+-------+--------+----------------------------------+----------------------------------------------------------------------+

* TOP LEVEL CONTROL
+-----------+------------+-----------+
| Interface | Type       | Ports     |
+-----------+------------+-----------+
| ap clk    | clock      | ap clk    |
| ap rst n  | reset      | ap rst n  |
| interrupt | interrupt  | interrupt |
| ap ctrl   | ap ctrl hs |           |
+-----------+------------+-----------+
```





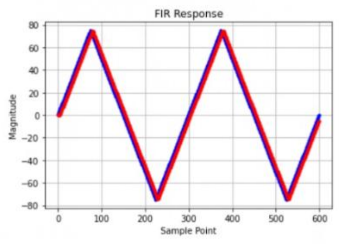


```
In [6]:
# coding: utf-8

# In[ ]:


from __future__ import print_function

import sys, os
import numpy as np
from time import time
import matplotlib.pyplot as plt

sys.path.append('/home/xilinx')
os.environ['XILINX_XRT'] = '/usr'
from pynq import Overlay
from pynq import allocate

if __name__ == "__main__":
    print("Entry:", sys.argv[0])
    print("System argument(s):", len(sys.argv))

    print("Start of \"" + sys.argv[0] + "\"")

    ol = Overlay("/home/root/jupyter_notebooks/FIRN11MAXI.bit")
    ipFIRN11 = ol.fir_n11_maxi_0

    fiSamples = open("samples_triangular_wave.txt", "r+")
    numSamples = 0
    line = fiSamples.readline()
    while line:
        numSamples = numSamples + 1
        line = fiSamples.readline()

    inBuffer0 = allocate(shape=(numSamples,), dtype=np.int32)
    outBuffer0 = allocate(shape=(numSamples,), dtype=np.int32)
    fiSamples.seek(0)
    for i in range(numSamples):
        line = fiSamples.readline()
        inBuffer0[i] = int(line)
    fiSamples.close()

    numTaps = 11
    n32Taps = [0, -10, -9, 23, 56, 63, 56, 23, -9, -10, 0]
    #n32Taps = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    n32DCGain = 0
    timeKernelStart = time()
    for i in range(numTaps):
        n32DCGain = n32DCGain + n32Taps[i]
        ipFIRN11.write(0x40 + i * 4, n32Taps[i])
    if n32DCGain < 0:
```



```
    if n32DCGain < 0:
        n32DCGain = 0 - n32DCGain
    ipFIRN11.write(0x28, len(inBuffer0) * 4)
    ipFIRN11.write(0x10, inBuffer0.device_address)
    ipFIRN11.write(0x1C, outBuffer0.device_address)
    ipFIRN11.write(0x00, 0x01)
    while (ipFIRN11.read(0x00) & 0x4) == 0x0:
        continue
    timeKernelEnd = time()
    print("Kernel execution time: " + str(timeKernelEnd - timeKernelStart) + " s")

    plt.title("FIR Response")
    plt.xlabel("Sample Point")
    plt.ylabel("Magnitude")
    xSeq = range(len(inBuffer0))
    if n32DCGain == 0:
        plt.plot(xSeq, inBuffer0, 'b.', xSeq, outBuffer0, 'r.')
    else:
        plt.plot(xSeq, inBuffer0, 'b.', xSeq, outBuffer0 / n32DCGain, 'r.')
    plt.grid(True)
    plt.show() # In Jupyter, press Tab + Shift keys to show plot then redo run

    print("===========================")
    print("Exit process")
```

```
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: 0.000274658203125 s
```



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

#2 :

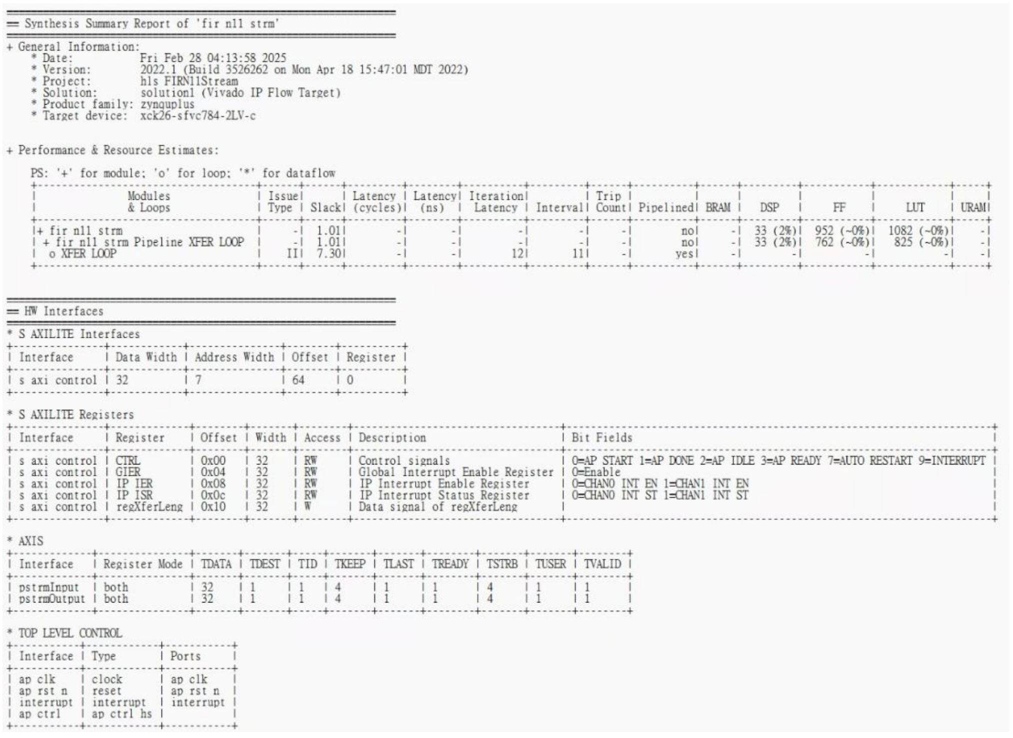
(1) 内容

#2 和#1 幾乎一樣，只是在讀寫 data 的介面從 AXI-Master 換成 AXI-Stream。接下來只會提到不同的部分。

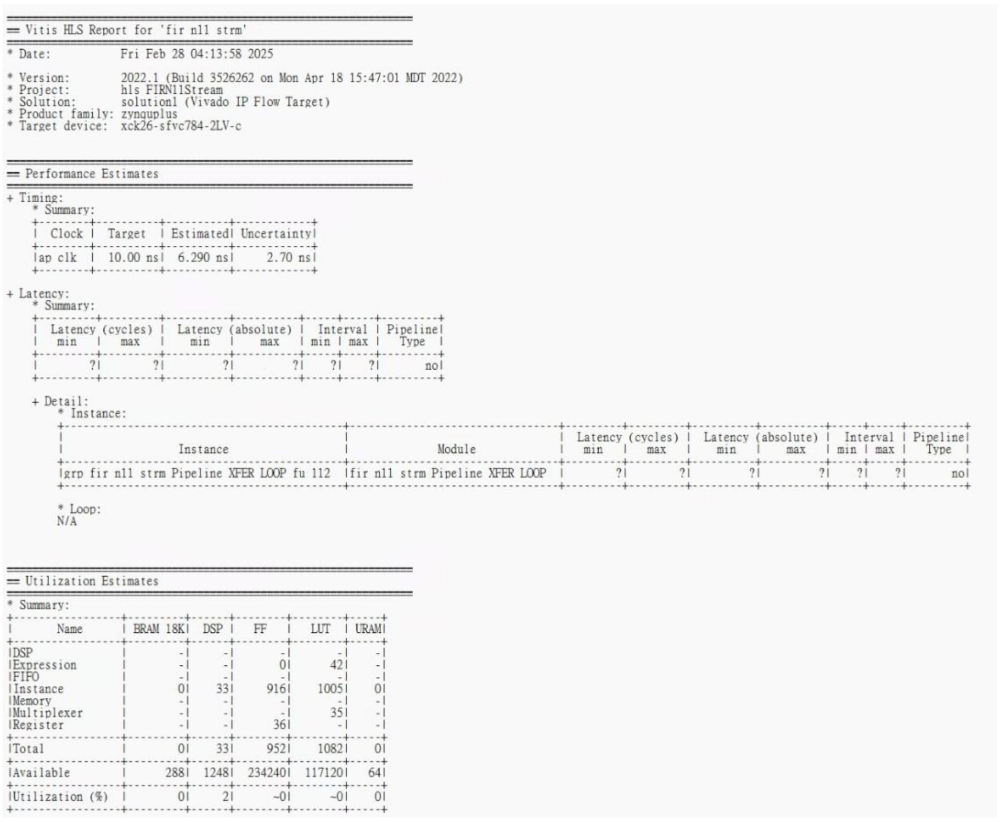
在 vivado 進行 block diagram 的 connection 需要注意的是，要將 dma ip 呼叫出來，並且將其中一個設為 read 另一個設為 write 並提前手動連接 s_ss2m 以及 s_m2ss，將 s_ss2m 連接到 axi_dma_1 而 s_m2ss 連接到 axi_dma_0，接著再讓它進行自動連接，此時一樣要開啟 slave hp port。

在 python code 的部分，由於其採用 DMA 連結 PS 的 AXI-Master 和 kernel 的 AXI-Stream，因此要透過 ipDMAIn 以及 ipDMAOut 進行 data 讀寫，其他部分和#1 相同。

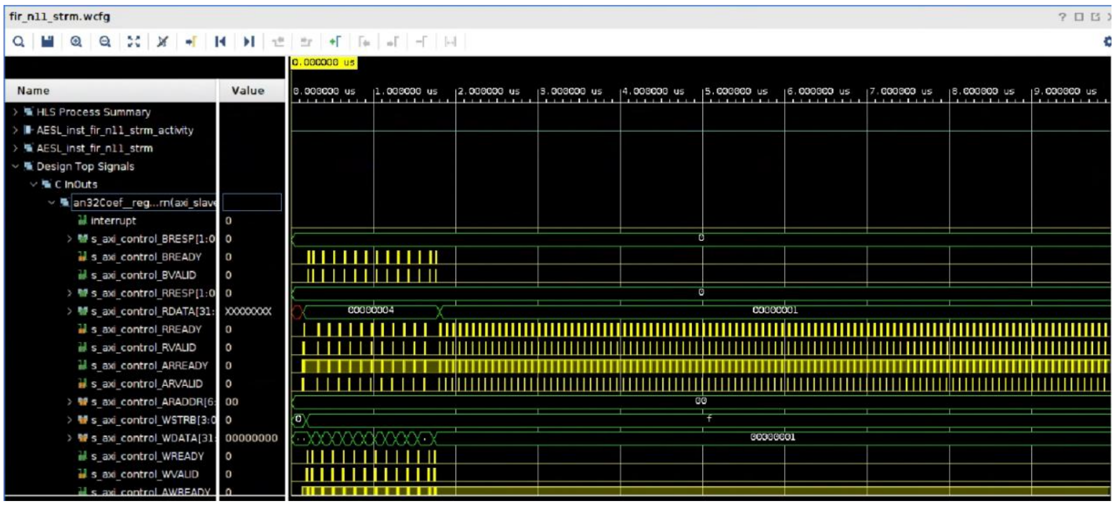
(2) 相關截圖

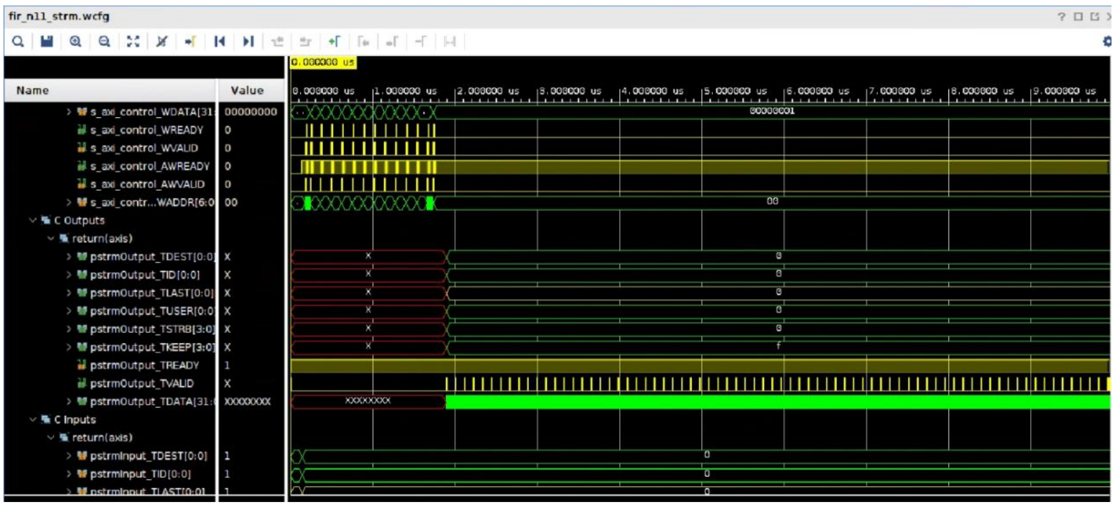


```
================================================================
== Synthesis Summary Report of 'fir n11 strm'
================================================================
+ General Information:
    * Date:           Fri Feb 28 04:13:58 2025
    * Version:        2022.1 (Build 3526262 on Mon Apr 18 15:47:01 MDT 2022)
    * Project:        hls FIRN11Stream
    * Solution:       solution1 (Vivado IP Flow Target)
    * Product family: zynquplus
    * Target device:  xck26-sfvc784-2LV-c


+ Performance & Resource Estimates:

    PS: '+' for module; 'o' for loop; '*' for dataflow
    +---------------------------------+------+------+---------+--------+----------+---------+------+----------+------+---------+-----------+------------+-----+
    |              Modules            | Issue|      | Latency | Latency| Iteration|         | Trip |          |      |         |           |            |     |
    |              & Loops            | Type | Slack| (cycles)|  (ns)  |  Latency | Interval| Count| Pipelined| BRAM |   DSP   |    FF     |    LUT     | URAM|
    +---------------------------------+------+------+---------+--------+----------+---------+------+----------+------+---------+-----------+------------+-----+
    |+ fir n11 strm                   |     -|  1.01|        -|       -|         -|        -|     -|        no|     -| 33 (2%)| 952 (~0%)| 1082 (~0%)|    -|
    | + fir n11 strm Pipeline XFER LOOP|    -|  1.01|        -|       -|         -|        -|     -|        no|     -| 33 (2%)| 762 (~0%)|  825 (~0%)|    -|
    |  o XFER LOOP                    |    II|  7.30|        -|       -|        12|       11|     -|       yes|     -|        -|          -|           -|    -|
    +---------------------------------+------+------+---------+--------+----------+---------+------+----------+------+---------+-----------+------------+-----+


================================================================
== HW Interfaces
================================================================
* S AXILITE Interfaces
+---------------+------------+---------------+--------+----------+
| Interface     | Data Width | Address Width | Offset | Register |
+---------------+------------+---------------+--------+----------+
| s axi control | 32         | 7             | 64     | 0        |
+---------------+------------+---------------+--------+----------+

* 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       |                                                                      |
+---------------+--------------+--------+-------+--------+----------------------------------+----------------------------------------------------------------------+

* AXIS
+-------------+---------------+-------+-------+-----+-------+-------+--------+-------+-------+--------+
| Interface   | Register Mode | TDATA | TDEST | TID | TKEEP | TLAST | TREADY | TSTRB | TUSER | TVALID |
+-------------+---------------+-------+-------+-----+-------+-------+--------+-------+-------+--------+
| pstrmInput  | both          | 32    | 1     | 1   | 4     | 1     | 1      | 4     | 1     | 1      |
| pstrmOutput | both          | 32    | 1     | 1   | 4     | 1     | 1      | 4     | 1     | 1      |
+-------------+---------------+-------+-------+-----+-------+-------+--------+-------+-------+--------+

* TOP LEVEL CONTROL
+-----------+------------+-----------+
| Interface | Type       | Ports     |
+-----------+------------+-----------+
| ap clk    | clock      | ap clk    |
| ap rst n  | reset      | ap rst n  |
| interrupt | interrupt  | interrupt |
| ap ctrl   | ap ctrl hs |           |
+-----------+------------+-----------+
```



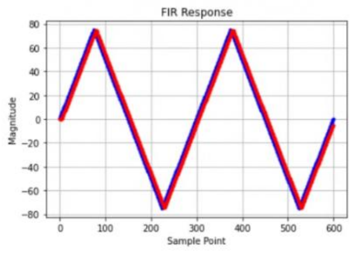
```
================================================================
== Vitis HLS Report for 'fir n11 strm'
================================================================
* Date:           Fri Feb 28 04:13:58 2025

* Version:        2022.1 (Build 3526262 on Mon Apr 18 15:47:01 MDT 2022)
* Project:        hls FIRN11Stream
* Solution:       solution1 (Vivado IP Flow Target)
* Product family: zynquplus
* Target device:  xck26-sfvc784-2LV-c


================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+----------+----------+------------+
    |  Clock |  Target  | Estimated| Uncertainty|
    +--------+----------+----------+------------+
    |ap clk  |  10.00 ns|  6.290 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


================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-----+--------+--------+-----+
|       Name      | BRAM 18K| DSP |   FF   |   LUT  | URAM|
+-----------------+---------+-----+--------+--------+-----+
|DSP              |        -|    -|       -|       -|    -|
|Expression       |        -|    -|       0|      42|    -|
|FIFO             |        -|    -|       -|       -|    -|
|Instance         |        0|   33|     916|    1005|    0|
|Memory           |        -|    -|       -|       -|    -|
|Multiplexer      |        -|    -|       -|      35|    -|
|Register         |        -|    -|      36|       -|    -|
+-----------------+---------+-----+--------+--------+-----+
|Total            |        0|   33|     952|    1082|    0|
+-----------------+---------+-----+--------+--------+-----+
|Available        |      288| 1248|  234240|  117120|   64|
+-----------------+---------+-----+--------+--------+-----+
|Utilization (%)  |        0|    2|     ~0|      ~0|    0|
+-----------------+---------+-----+--------+--------+-----+
```





In [10]:
```python
# coding: utf-8

# In[3]:


from __future__ import print_function

import sys, os
import numpy as np
from time import time
import matplotlib.pyplot as plt

sys.path.append('/home/xilinx')
os.environ['XILINX_XRT'] = '/usr'
from pynq import Overlay
from pynq import allocate

if __name__ == "__main__":
    print("Entry:", sys.argv[0])
    print("System argument(s):", len(sys.argv))

    print("Start of \"" + sys.argv[0] + "\"")

    ol = Overlay("/home/root/jupyter_notebooks/FIRN11Stream.bit")
    ipFIRN11 = ol.fir_n11_strm_0
    ipDMAIn = ol.axi_dma_0
    ipDMAOut = ol.axi_dma_1

    fiSamples = open("samples_triangular_wave.txt", "r+")
    numSamples = 0
    line = fiSamples.readline()
    while line:
        numSamples = numSamples + 1
        line = fiSamples.readline()

    inBuffer0 = allocate(shape=(numSamples,), dtype=np.int32)
    outBuffer0 = allocate(shape=(numSamples,), dtype=np.int32)
    fiSamples.seek(0)
    for i in range(numSamples):
        line = fiSamples.readline()
        inBuffer0[i] = int(line)
    fiSamples.close()

    numTaps = 11
    n32Taps = [0, -10, -9, 23, 56, 63, 56, 23, -9, -10, 0]
    #n32Taps = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    n32DCGain = 0
    timeKernelStart = time()
    for i in range(numTaps):
        n32DCGain = n32DCGain + n32Taps[i]
        ipFIRN11.write(0x40 + i * 4, n32Taps[i])
    if n32DCGain < 0:
        n32DCGain = 0 - n32DCGain
    ipFIRN11.write(0x10, len(inBuffer0) * 4)
    ipFIRN11.write(0x00, 0x01)
    ipDMAIn.sendchannel.transfer(inBuffer0)
    ipDMAOut.recvchannel.transfer(outBuffer0)
    ipDMAIn.sendchannel.wait()
    ipDMAOut.recvchannel.wait()
    timeKernelEnd = time()
    print("Kernel execution time: " + str(timeKernelEnd - timeKernelStart) + " s")

    plt.title("FIR Response")
    plt.xlabel("Sample Point")
    plt.ylabel("Magnitude")
    xSeq = range(len(inBuffer0))
    if n32DCGain == 0:
        plt.plot(xSeq, inBuffer0, 'b.', xSeq, outBuffer0, 'r.')
    else:
        plt.plot(xSeq, inBuffer0, 'b.', xSeq, outBuffer0 / n32DCGain, 'r.')
    plt.grid(True)
    plt.show() # In Jupyter, press Tab + Shift keys to show plot then redo run

    print("============================")
    print("Exit process")
```

```
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: 0.0008769035339355469 s
```



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