AAHLS (Lab 1)

1. 簡介

本報告探討 AAHLS（Lab1），主要目的是透過 AXI-Lite 來實現簡單的乘法器，透過此 Lab 讓同學能夠認識使用 VIVADO HLS 在 PYNQ-Z2 board 上的基本開發流程，詳細內容會在後面呈現。

2. 報告內容

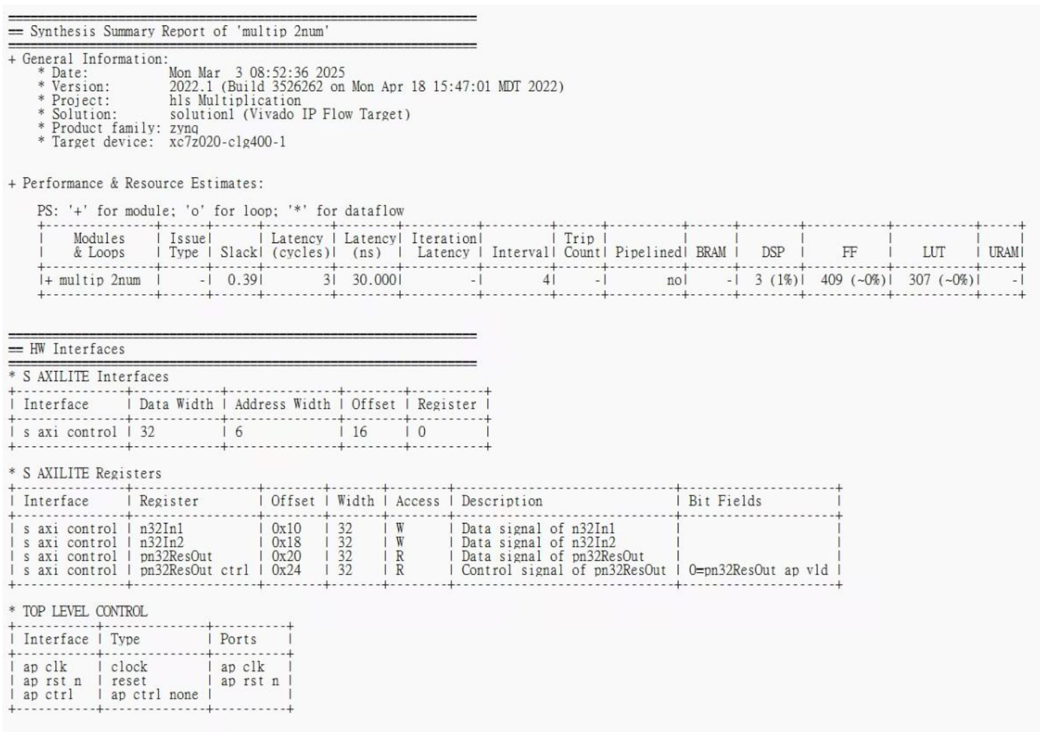
(1) 內容

本次 Lab 實作一個 32bits fix-point 的乘法器，其中 input 以及 output 皆使用 AXI-Lite 進行傳輸。由於此 module 是 always active 因此 control signal 選擇為 ap_ctrl_none。在本次 Lab 有特別提到關於 interface 設定主要有兩種方法，分別是 inline 以及 directive，directive 和 inline 不同的是它沒有寫在 code 裡面，因此在開發階段可以更方便調適，當開發完成就應該改成 inline 方便其他專案使用，不需要再手動調整。

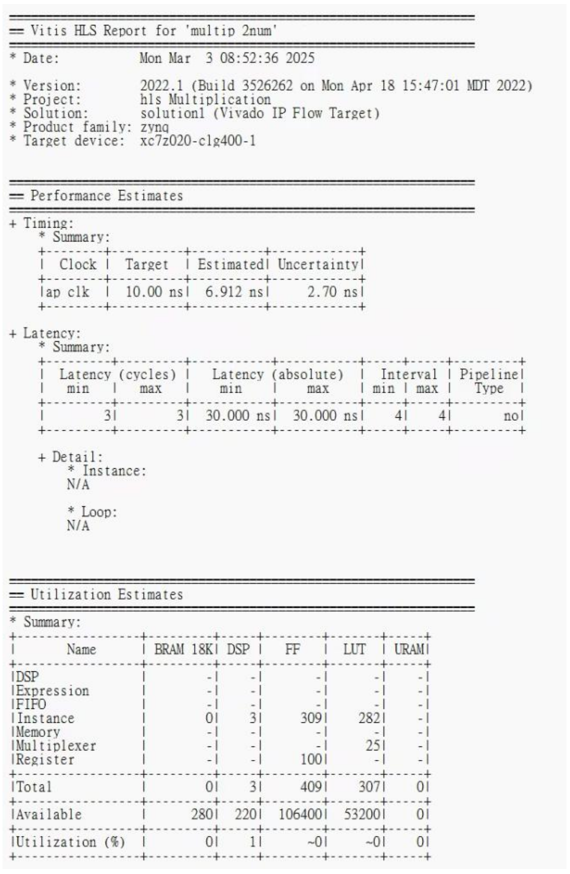
此外，此次 Lab 也有使用 vivado 進行 block 的 connect，由於 AXI-Lite 是透過 master GP port 連接，zynq master port 預設是開啟的因此不用特別設定。原本 PL clock 助教是設定 5ns，但因為觀察到 slack 是負的 (time violation)，因此把它調回 10ns。

最後是透過 jupyter lab server 連線 pynq-z2 board，透過 python code 調用 API 來進行 host 以及 kernel 的 communication。Kernel 的 address 可以從*_hw.h file 中查看，裡面顯示的 address 都是相較於 base address 的位移，這些設定可以協助 python code 進行調用 kernel。Python code 首先透過 o1.multip_2num_0 得到 base address，再調用 API 的 READ/WRITE function，透過輸入相對 address 來進行 communication。

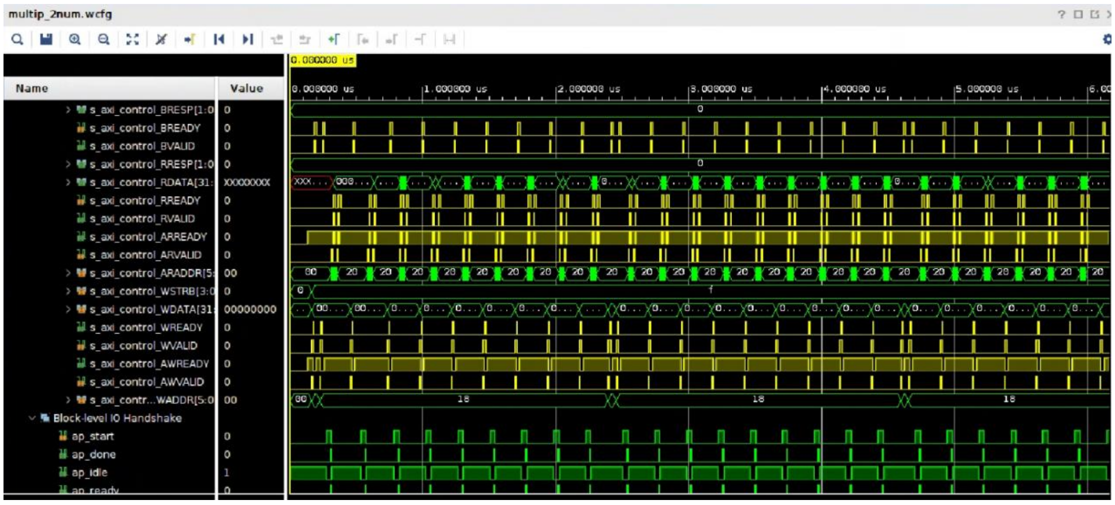
(2) 相關截圖



```
================================================================
== Synthesis Summary Report of 'multip 2num'
================================================================
+ General Information:
    * Date:           Mon Mar  3 08:52:36 2025
    * Version:        2022.1 (Build 3526262 on Mon Apr 18 15:47:01 MDT 2022)
    * Project:        hls Multiplication
    * Solution:       solution1 (Vivado IP Flow Target)
    * Product family: zynq
    * Target device:  xc7z020-clg400-1


+ 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|
    +---------------+------+------+---------+--------+----------+---------+------+----------+------+--------+-----------+-----------+-----+
    |+ multip 2num  |     -|  0.39|        3|  30.000|         -|        4|     -|        no|     -|  3 (1%)|  409 (~0%)|  307 (~0%)|    -|
    +---------------+------+------+---------+--------+----------+---------+------+----------+------+--------+-----------+-----------+-----+


================================================================
== HW Interfaces
================================================================
* S AXILITE Interfaces
+---------------+------------+---------------+--------+----------+
| Interface     | Data Width | Address Width | Offset | Register |
+---------------+------------+---------------+--------+----------+
| s axi control | 32         | 6             | 16     | 0        |
+---------------+------------+---------------+--------+----------+

* S AXILITE Registers
+---------------+------------------+--------+-------+--------+-----------------------------+----------------------+
| Interface     | Register         | Offset | Width | Access | Description                 | Bit Fields           |
+---------------+------------------+--------+-------+--------+-----------------------------+----------------------+
| s axi control | n32In1           | 0x10   | 32    | W      | Data signal of n32In1       |                      |
| s axi control | n32In2           | 0x18   | 32    | W      | Data signal of n32In2       |                      |
| s axi control | pn32ResOut       | 0x20   | 32    | R      | Data signal of pn32ResOut   |                      |
| s axi control | pn32ResOut ctrl  | 0x24   | 32    | R      | Control signal of pn32ResOut | 0=pn32ResOut ap vld |
+---------------+------------------+--------+-------+--------+-----------------------------+----------------------+

* TOP LEVEL CONTROL
+-----------+--------------+----------+
| Interface | Type         | Ports    |
+-----------+--------------+----------+
| ap clk    | clock        | ap clk   |
| ap rst n  | reset        | ap rst n |
| ap ctrl   | ap ctrl none |          |
+-----------+--------------+----------+
```

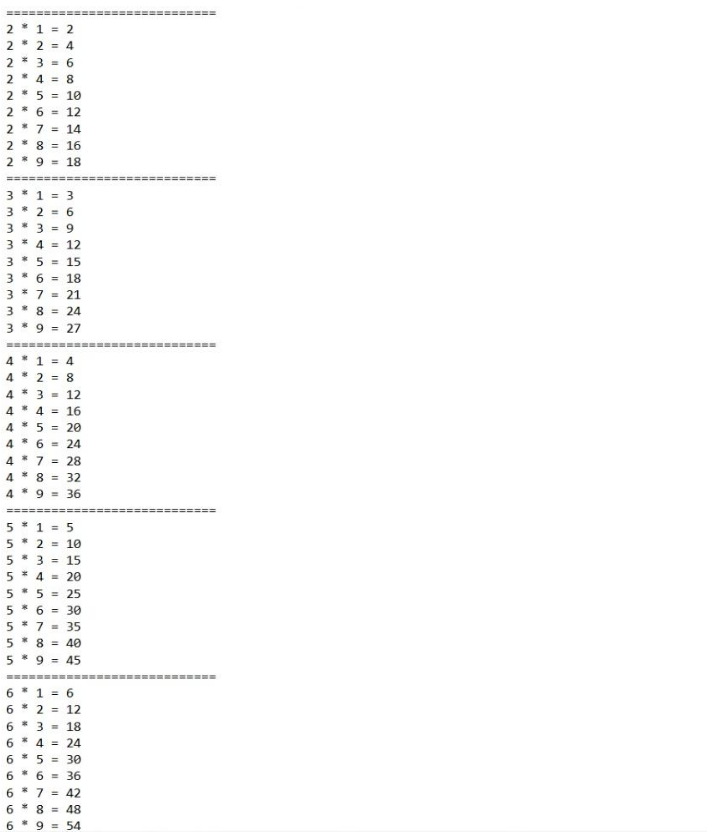


```
================================================================
== Vitis HLS Report for 'multip 2num'
================================================================
* Date:           Mon Mar  3 08:52:36 2025

* Version:        2022.1 (Build 3526262 on Mon Apr 18 15:47:01 MDT 2022)
* Project:        hls Multiplication
* Solution:       solution1 (Vivado IP Flow Target)
* Product family: zynq
* Target device:  xc7z020-clg400-1


================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap clk  | 10.00 ns| 6.912 ns |     2.70 ns|
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+-----------+-----------+-----+-----+---------+
    |  Latency (cycles) |  Latency (absolute)   |  Interval | Pipeline|
    |   min   |   max   |    min    |    max    | min | max |   Type  |
    +---------+---------+-----------+-----------+-----+-----+---------+
    |        3|        3| 30.000 ns | 30.000 ns |    4|    4|       no|
    +---------+---------+-----------+-----------+-----+-----+---------+

    + Detail:
        * Instance:
        N/A

        * Loop:
        N/A


================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-----+--------+-------+-----+
|       Name      | BRAM 18K| DSP |   FF   |  LUT  | URAM|
+-----------------+---------+-----+--------+-------+-----+
|DSP              |        -|    -|       -|      -|    -|
|Expression       |        -|    -|       -|      -|    -|
|FIFO             |        -|    -|       -|      -|    -|
|Instance         |        0|    3|     309|    282|    -|
|Memory           |        -|    -|       -|      -|    -|
|Multiplexer      |        -|    -|       -|     25|    -|
|Register         |        -|    -|     100|      -|    -|
+-----------------+---------+-----+--------+-------+-----+
|Total            |        0|    3|     409|    307|    0|
+-----------------+---------+-----+--------+-------+-----+
|Available        |      280|  220|  106400|  53200|    0|
+-----------------+---------+-----+--------+-------+-----+
|Utilization (%)  |        0|    1|     ~0|     ~0|    0|
+-----------------+---------+-----+--------+-------+-----+
```

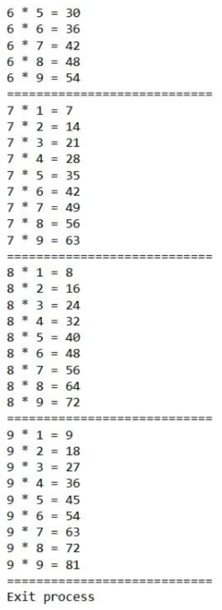


```
# coding: utf-8

# In[ ]:


from __future__ import print_function

import sys, os

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

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

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

    ol = Overlay("/home/xilinx/jupyter_notebooks/Multip2Num.bit")
    regIP = ol.multip_2num_0

    for i in range(9):
        print("===========================")
        for j in range(9):
            regIP.write(0x10, i + 1)
            regIP.write(0x18, j + 1)
            Res = regIP.read(0x20)
            print(str(i + 1) + " * " + str(j + 1) + " = " + str(Res))
    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"
===========================
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
===========================
```



```
============================
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
============================
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
============================
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
============================
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
============================
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
```



```
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
============================
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
============================
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
============================
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
============================
Exit process
```