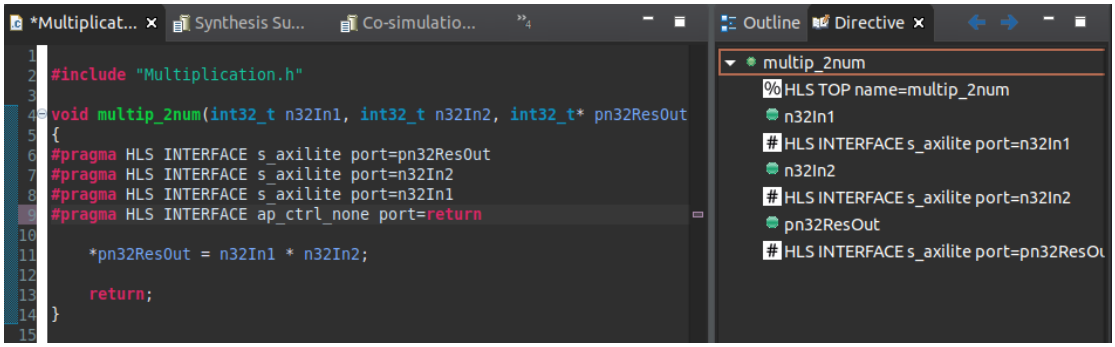
SOC_Lab1　　311510207　江尹凡

Brief introduction about the overall system

1. 撰寫成法器
2. Directive control
3. 寫 tesbench 用 vitis_hls 做 C sim 模擬 C 語言寫的乘法器是否正確。
4. C synthesis。
5. Cosimulation，cosimulation 這邊要把圖片中第九行註解調重新合成才能跑。Cosimulation 後就可以打開波型觀察。
6. Export RTL ，輸出 RTL 前，要把前一步註解掉的那行弄回來，重新合成在輸出完整的 RTL code。

What is observed & learned

學習安裝 HLS、vivado 環境，並跑一次流程，大概了解設計時需要跑甚麼步驟。並且用遠端連線的方式，操作 fpga 相當新穎。



設定 in、output 的 interface 有 2 種方式

1. 用右邊 Directive 創建 ，通常建議，如果要視不同的 solution 可以直接改右邊去模擬，這樣就可以用同一份 source code 測試。
2. (inline)用 #pragma 這行指令直接撰寫，如果 code 已經確定形式，建議是打成#pragma，這樣 code 傳上 github 或分享給別人，可以比較完整知道設計。

Screen dump

C_sim

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../Multiplication.cpp in debug mode
   Generating csim.exe
>> Start test!
------------------------
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
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
------------------------
>> Test passed!
------------------------
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Synthesis Summary Report

```
================================================================
== Vitis HLS Report for 'multip_2num'
================================================================
* Date:           Mon Sep 25 11:08:29 2023

* Version:        2023.1 (Build 3854077 on May  4 2023)
* Project:        Lab1
* 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|
    +---------+---------+-----------+-----------+-----+-----+---------+
```

Performence

```
================================================================
== Vitis HLS Report for 'multip_2num'
================================================================
* Date:           Mon Sep 25 11:08:29 2023

* Version:        2023.1 (Build 3854077 on May  4 2023)
* Project:        Lab1
* 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

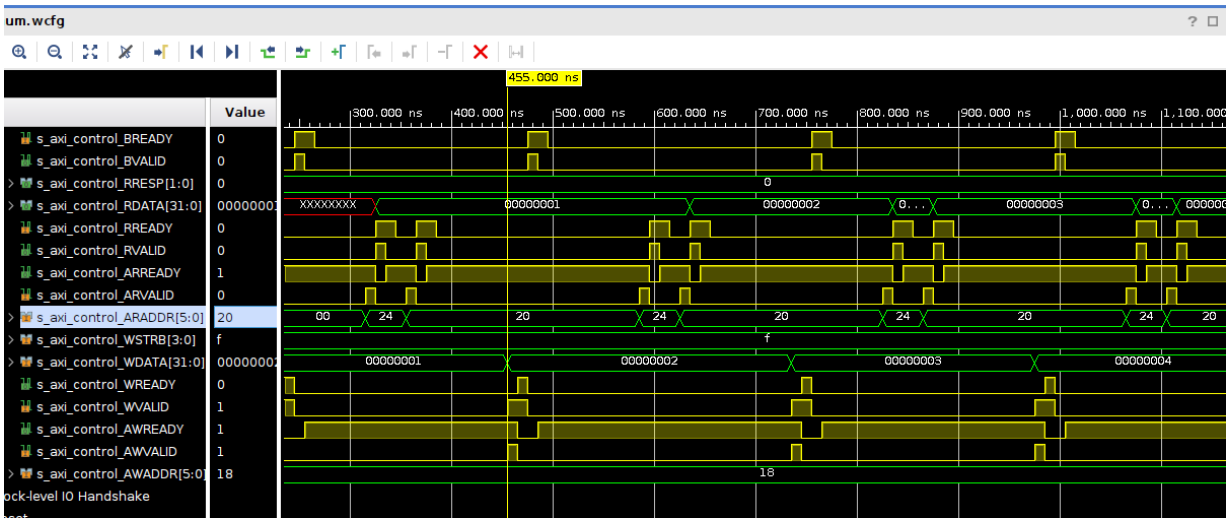
```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-----+--------+-------+-----+
|       Name      | BRAM_18K| DSP |   FF   |  LUT  | URAM|
+-----------------+---------+-----+--------+-------+-----+
|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|
+-----------------+---------+-----+--------+-------+-----+

+ Detail:
    * Instance:
    +-----------------------+---------------------+---------+----+-----+-----+-----+
    |        Instance       |        Module       | BRAM_18K| DSP|  FF | LUT | URAM|
    +-----------------------+---------------------+---------+----+-----+-----+-----+
    |control_s_axi_U        |control_s_axi        |        0|   0|  144|  232|    0|
    |mul_32s_32s_32_2_1_U1  |mul_32s_32s_32_2_1   |        0|   3|  165|   50|    0|
    +-----------------------+---------------------+---------+----+-----+-----+-----+
    |Total                  |                     |        0|   3|  309|  282|    0|
    +-----------------------+---------------------+---------+----+-----+-----+-----+
```

Interface

```
================================================================
== Interface
================================================================
* Summary:
+-----------------------+-----+-----+--------------+--------------+--------------+
|       RTL Ports       | Dir | Bits|   Protocol   | Source Object|    C Type    |
+-----------------------+-----+-----+--------------+--------------+--------------+
|s_axi_control_AWVALID  |   in|    1|        s_axi|       control|       pointer|
|s_axi_control_AWREADY  |  out|    1|        s_axi|       control|       pointer|
|s_axi_control_AWADDR   |   in|    6|        s_axi|       control|       pointer|
|s_axi_control_WVALID   |   in|    1|        s_axi|       control|       pointer|
|s_axi_control_WREADY   |  out|    1|        s_axi|       control|       pointer|
|s_axi_control_WDATA    |   in|   32|        s_axi|       control|       pointer|
|s_axi_control_WSTRB    |   in|    4|        s_axi|       control|       pointer|
|s_axi_control_ARVALID  |   in|    1|        s_axi|       control|       pointer|
|s_axi_control_ARREADY  |  out|    1|        s_axi|       control|       pointer|
|s_axi_control_ARADDR   |   in|    6|        s_axi|       control|       pointer|
|s_axi_control_RVALID   |  out|    1|        s_axi|       control|       pointer|
|s_axi_control_RREADY   |   in|    1|        s_axi|       control|       pointer|
|s_axi_control_RDATA    |  out|   32|        s_axi|       control|       pointer|
|s_axi_control_RRESP    |  out|    2|        s_axi|       control|       pointer|
|s_axi_control_BVALID   |  out|    1|        s_axi|       control|       pointer|
|s_axi_control_BREADY   |   in|    1|        s_axi|       control|       pointer|
|s_axi_control_BRESP    |  out|    2|        s_axi|       control|       pointer|
|ap_clk                 |   in|    1| ap_ctrl_none|   multip_2num|  return value|
|ap_rst_n               |   in|    1| ap_ctrl_none|   multip_2num|  return value|
+-----------------------+-----+-----+--------------+--------------+--------------+
```

Co-simulation transcript/waveform

```
-----------------------
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 8.14 seconds. CPU system time: 1.74 seconds. Elapsed time: 13.83 seconds; current allocated memory: 7.094 MB.
INFO: [HLS 200-112] Total CPU user time: 10.19 seconds. Total CPU system time: 2.35 seconds. Total elapsed time: 17.73 seconds; peak allocated memory: 1.097 GB.
Finished C/RTL cosimulation.
```



Jupyter Notebook execution results

In [2]:

```
# 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_la
uncher.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
```

```
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
```

In [ ]: