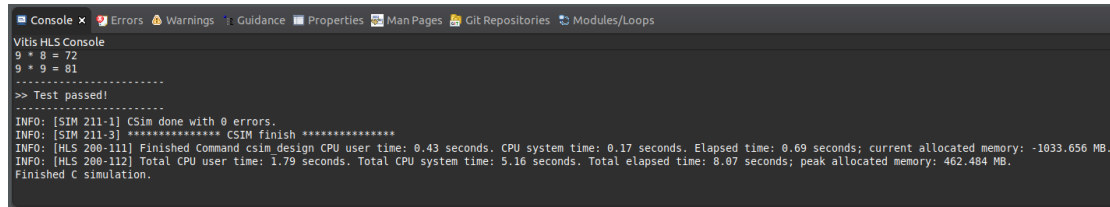


SOC Design Lab01

112061611 陳伯丞

1.

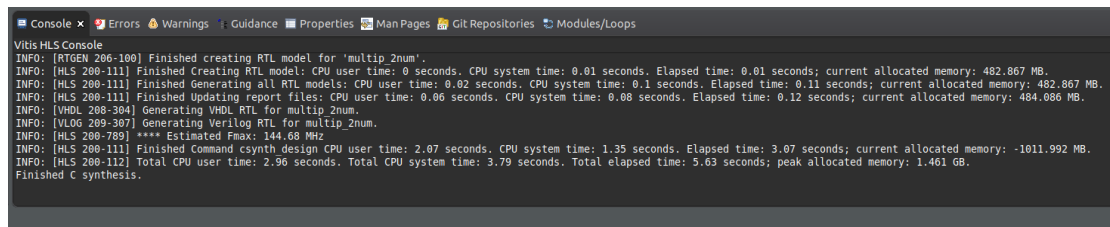
Lab1 要將 c 語言的程式轉換成 RTL code，再合成到 FPGA 板上執行，所以第一步驟要先確認初始的 c program 的正確性。



```
Console x Errors Warnings Guidance Properties Man Pages Git Repositories Modules/Loops
Vitis HLS Console
9 * 8 = 72
9 * 9 = 81
-----
>> Test passed!
-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
INFO: [HLS 200-111] Finished Command csim design CPU user time: 0.43 seconds. CPU system time: 0.17 seconds. Elapsed time: 0.69 seconds; current allocated memory: -1033.656 MB.
INFO: [HLS 200-112] Total CPU user time: 1.79 seconds. Total CPU system time: 5.16 seconds. Total elapsed time: 8.07 seconds; peak allocated memory: 462.484 MB.
Finished C simulation.
```

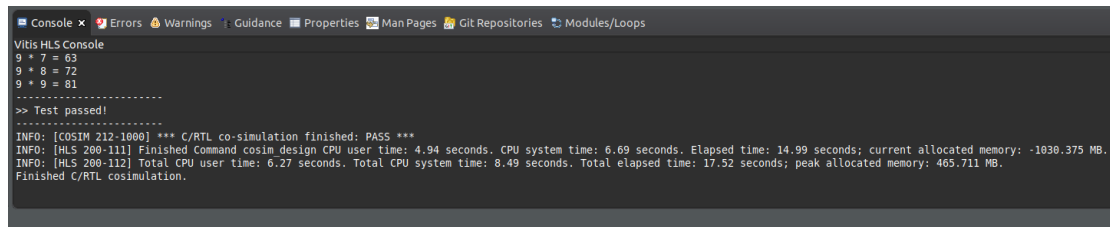
Figure1. C simulation

其次，要將 c program 轉換成 RTL code，並且確認 RTL code 的正確性。



```
Console x Errors Warnings Guidance Properties Man Pages Git Repositories Modules/Loops
Vitis HLS Console
INFO: [RTGEN 206-100] Finished creating RTL model for 'multip_2num'.
INFO: [HLS 200-111] Finished Creating RTL model: CPU user time: 0 seconds. CPU system time: 0.01 seconds. Elapsed time: 0.01 seconds; current allocated memory: 482.867 MB.
INFO: [HLS 200-111] Finished Generating all RTL models: CPU user time: 0.02 seconds. CPU system time: 0.1 seconds. Elapsed time: 0.11 seconds; current allocated memory: 482.867 MB.
INFO: [HLS 200-111] Finished Updating report files: CPU user time: 0.06 seconds. CPU system time: 0.06 seconds. Elapsed time: 0.12 seconds; current allocated memory: 484.086 MB.
INFO: [VHDL 200-304] Generating VHDL RTL for multip_2num.
INFO: [VLOG 200-307] Generating Verilog RTL for multip_2num.
INFO: [HLS 200-789] **** Estimated Fmax: 144.68 MHz
INFO: [HLS 200-111] Finished Command csynth design CPU user time: 2.07 seconds. CPU system time: 1.35 seconds. Elapsed time: 3.07 seconds; current allocated memory: -1011.992 MB.
INFO: [HLS 200-112] Total CPU user time: 2.96 seconds. Total CPU system time: 3.79 seconds. Total elapsed time: 5.63 seconds; peak allocated memory: 1.461 GB.
Finished C synthesis.
```

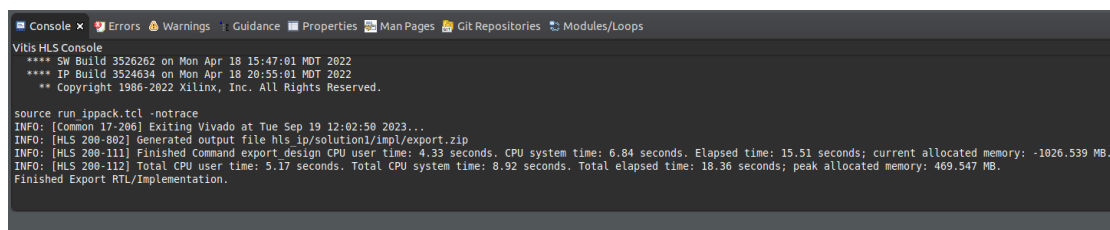
Figure2. C Synthesis



```
Console x Errors Warnings Guidance Properties Man Pages Git Repositories Modules/Loops
Vitis HLS Console
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
-----
>> Test passed!
-----
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [HLS 200-111] Finished Command cosim design CPU user time: 4.94 seconds. CPU system time: 6.09 seconds. Elapsed time: 14.99 seconds; current allocated memory: -1030.375 MB.
INFO: [HLS 200-112] Total CPU user time: 6.27 seconds. Total CPU system time: 8.49 seconds. Total elapsed time: 17.52 seconds; peak allocated memory: 465.711 MB.
Finished C/RTL cosimulation.
```

Figure3. RTL Cosimulation

接著將 RTL code 匯出至 Vivado。



```
Console x Errors Warnings Guidance Properties Man Pages Git Repositories Modules/Loops
Vitis HLS Console
**** SW Build 3526262 on Mon Apr 18 15:47:01 MDT 2022
**** IP Build 3524634 on Mon Apr 18 20:55:01 MDT 2022
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

source run ippack.tcl -notrace
INFO: [Common 17-206] Exiting Vivado at Tue Sep 19 12:02:50 2023...
INFO: [HLS 200-882] Generated output file hls_ip/solution1/impl/export.zip
INFO: [HLS 200-111] Finished Command export_design CPU user time: 4.33 seconds. CPU system time: 6.84 seconds. Elapsed time: 15.51 seconds; current allocated memory: -1026.539 MB.
INFO: [HLS 200-112] Total CPU user time: 5.17 seconds. Total CPU system time: 8.92 seconds. Total elapsed time: 18.36 seconds; peak allocated memory: 469.547 MB.
Finished Export RTL/Implementation.
```

Figure4. Export RTL

在 Vivado 中將先前得到的 RTL code 與 processor 等區塊相接，並且輸出 FPGA 能夠執行的 bit 檔。

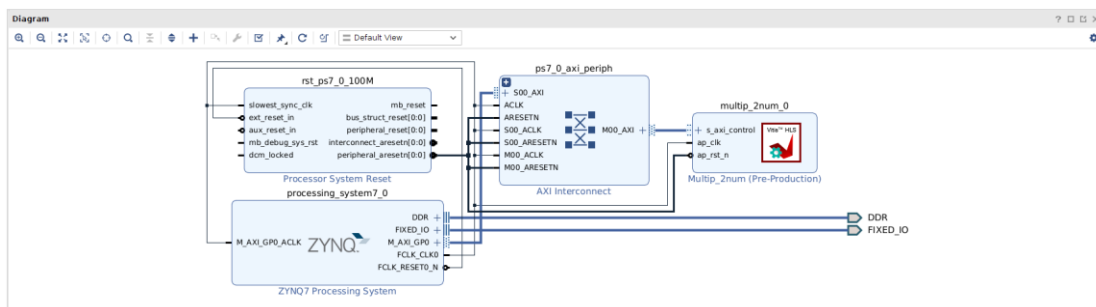


Figure5. Block Diagram

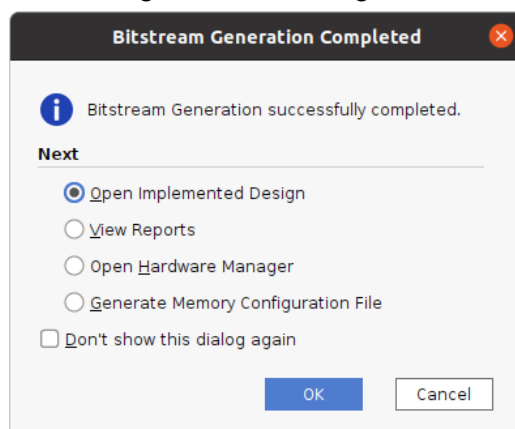


Figure6. Bitstream Generation

最後在 onlineFPGA 上執行，確認執行結果正確。

```
Jupyter Multip2Num Last Checkpoint: 1 分鐘前 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
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
In [ ]:
```

Figure7. OnlineFPGA

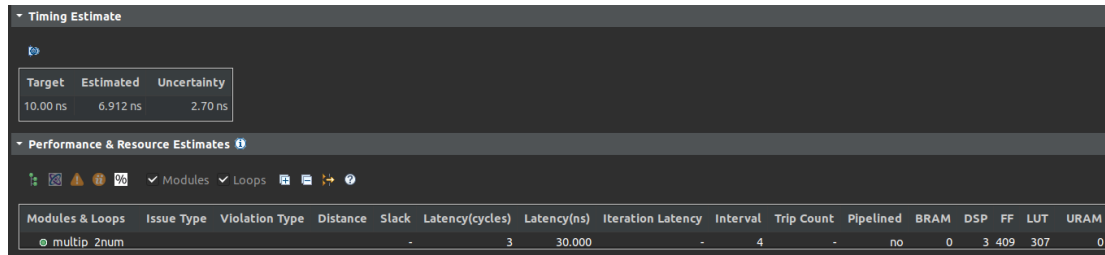


Figure8. Performance

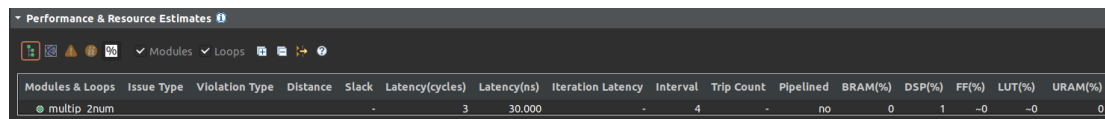


Figure9. Utilization

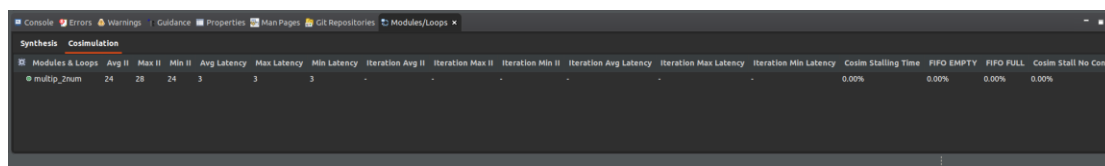


Figure10. Co-simulation Script

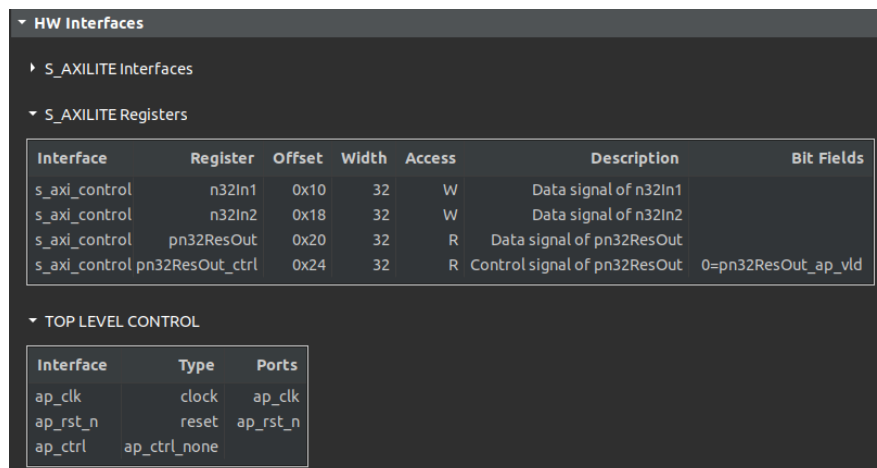


Figure11. Interfaces

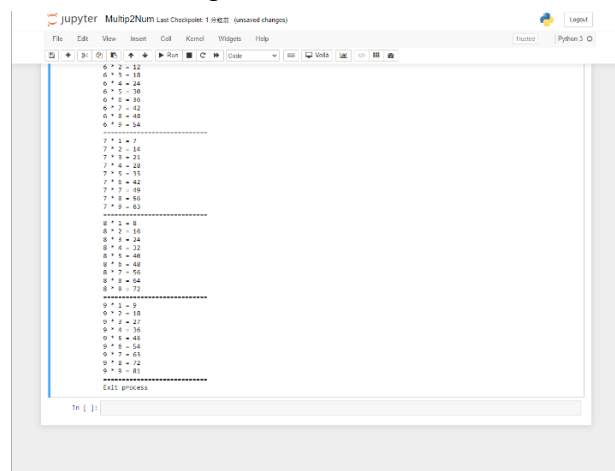


Figure12. Jupyter