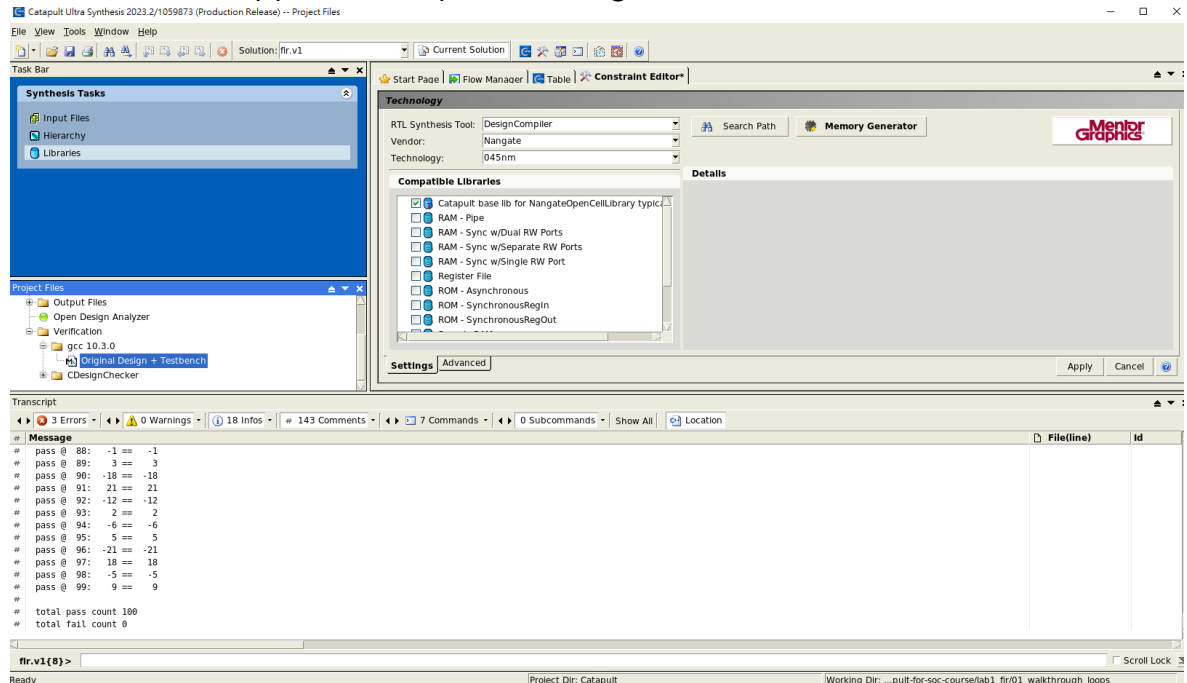


ASoC Lab-catapult-fir Report

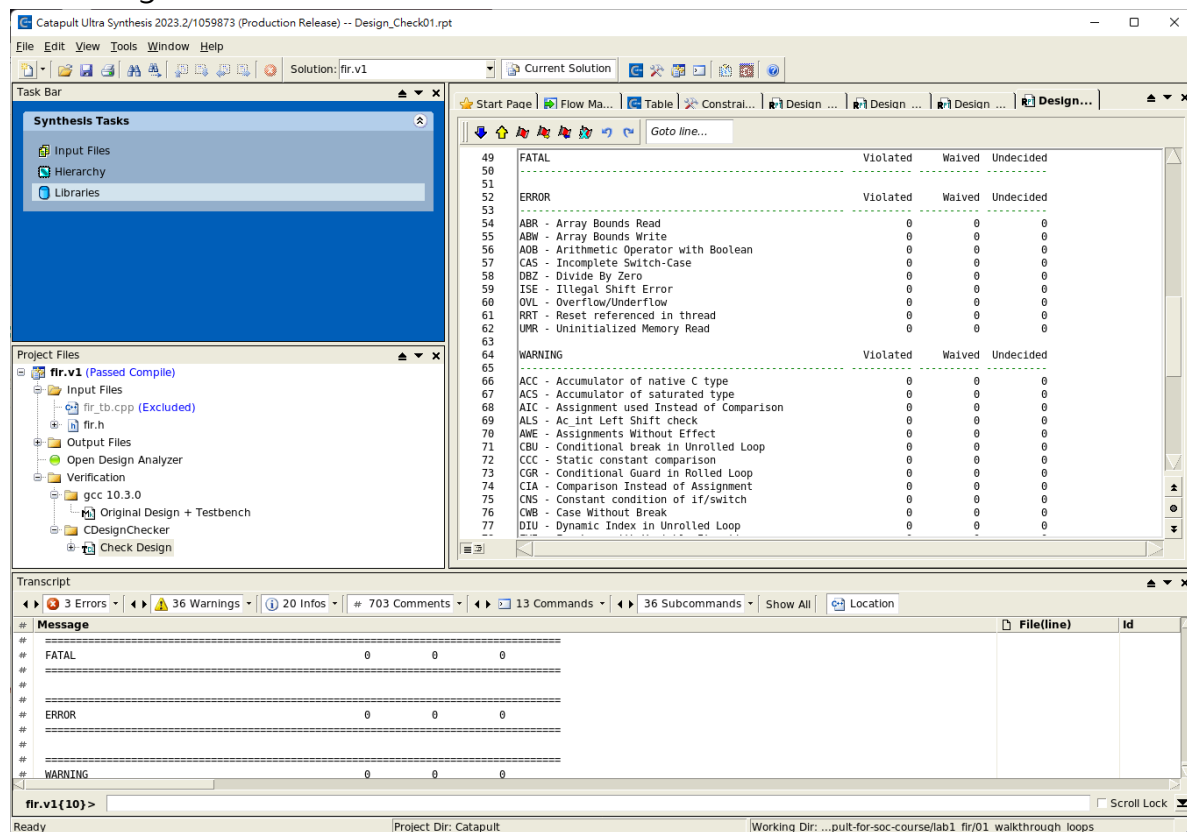
R12943012 蔡東翰

Part I : Walkthrough of Catapult C++ flow

1. 將 fir.h 與 fir_tb.cpp 匯入 catapult 並使用 gcc 編譯執行結果如下圖:



2. 跑 CDesignChecker 結果如下圖:



3. Scheduling 後截圖如下:

Task Bar: Synthesis Tasks (Input Files, Hierarchy, Libraries, Mapping, Architecture, Resources, Schedule, RTL, Power Report (Pre Power Opt), Power RTL)

Project Files: fir.v1 (Passed Allocate) (Input Files, Output Files, Open Design Analyzer, Verification, gcc 10.3.0, Original Design + Testbench, CDesignChecker, Check Design)

Loop Hierarchy:

- run:rlp (run)
 - main
 - SHIFT
 - SHIFT:if:io_read(h
 - SHIFT:else:mux
 - SHIFT:mux
 - SHIFT:mux#1
 - SHIFT:mux#2
 - SHIFT:mux#3
 - SHIFT:mux#4
 - SHIFT:mux#5
 - SHIFT:mux#6
 - SHIFT:mux#7
 - MAC:io_read(coef
 - MAC:mux
 - MAC:mux#1
 - MAC:mul
 - MAC:acc#1
 - SHIFT:acc#1
 - MAC:acc#2
 - io_write(output)
 - coeffs:io_sync(coeff

Scheduled Operations: C0, C1, C2

Runtime Profile: [Bar Chart]

Transcript:

```
# Message
# Prescheduled LOOP '/fir/run/run:rlp' (0 c-steps)
# Prescheduled SEQUENTIAL '/fir/run' (total length 10 c-steps)
# Initial schedule of SEQUENTIAL '/fir/run': Latency = 8, Area (Datapath, Register, Total) = 1496.75, 989.52, 2486.27
# At least one feasible schedule exists.
# Final schedule of SEQUENTIAL '/fir/run': Latency = 8, Area (Datapath, Register, Total) = 798.49, 989.52, 1788.01
# Resource allocation and scheduling done.
# Netlist written to file 'schedule.gnt'
# Completed transformation 'allocate' on solution 'fir.v1': elapsed time 0.52 seconds, memory usage 1393636kB, peak memory usage 1450916kB
# Design complexity at end of 'allocate': Total ops = 167, Real ops = 34, Vars = 43
```

4. RTL 後截圖如下:

Task Bar: Synthesis Tasks (Input Files, Hierarchy, Libraries, Mapping, Architecture, Resources, Schedule, RTL, Power Report (Pre Power Opt), Power RTL)

Project Files: fir.v1 (Passed Allocate) (Input Files, Output Files, Open Design Analyzer, Verification, gcc 10.3.0, Original Design + Testbench, CDesignChecker, Check Design)

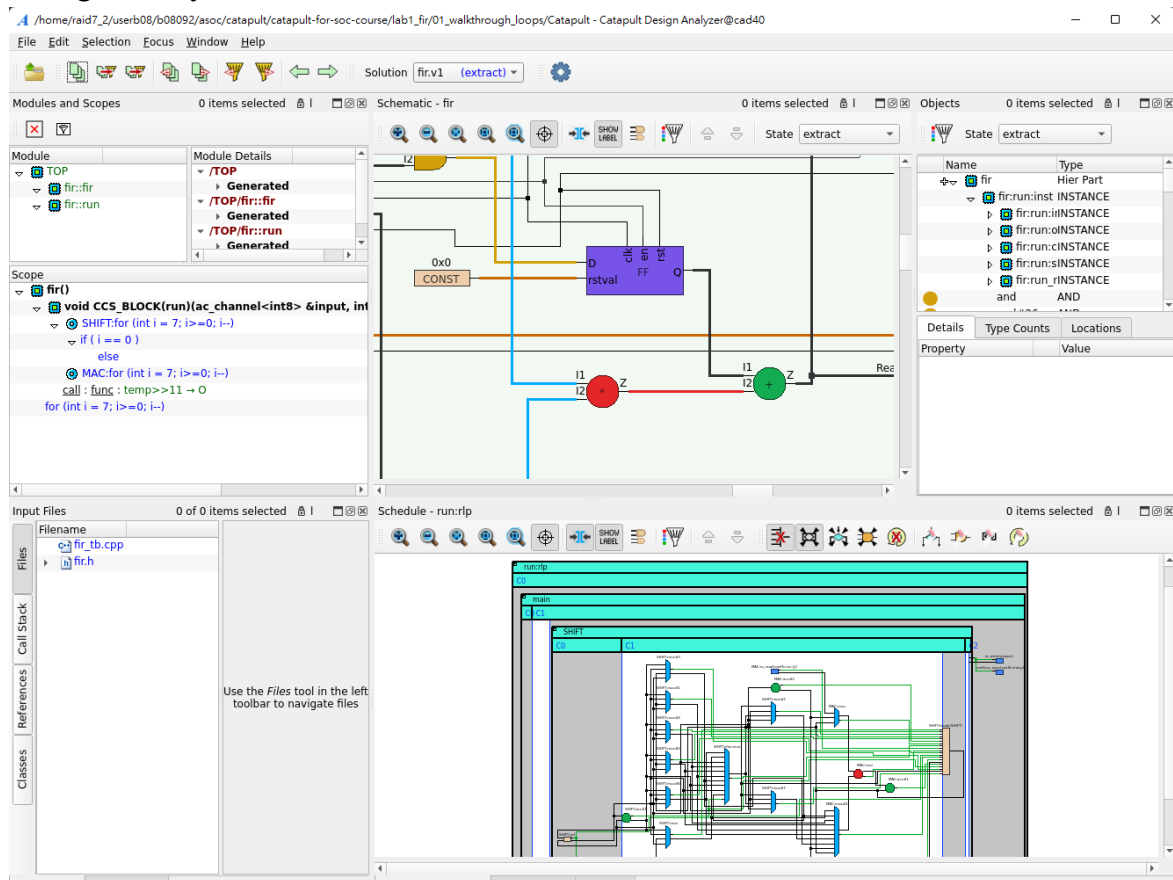
Design Unit: fir

```
module fir (
    clk, rst, input_rsc_dat, input_rsc_vld, input_rsc_rdy, coeffs_rsc_dat, coeffs_triosy_lz,
    output_rsc_dat, output_rsc_vld, output_rsc_rdy
);
    input clk;
    input rst;
    input [7:0] input_rsc_dat;
    input input_rsc_vld;
    output input_rsc_rdy;
    input [63:0] coeffs_rsc_dat;
    output coeffs_triosy_lz;
    output [7:0] output_rsc_dat;
    output output_rsc_vld;
    input output_rsc_rdy;
endmodule
```

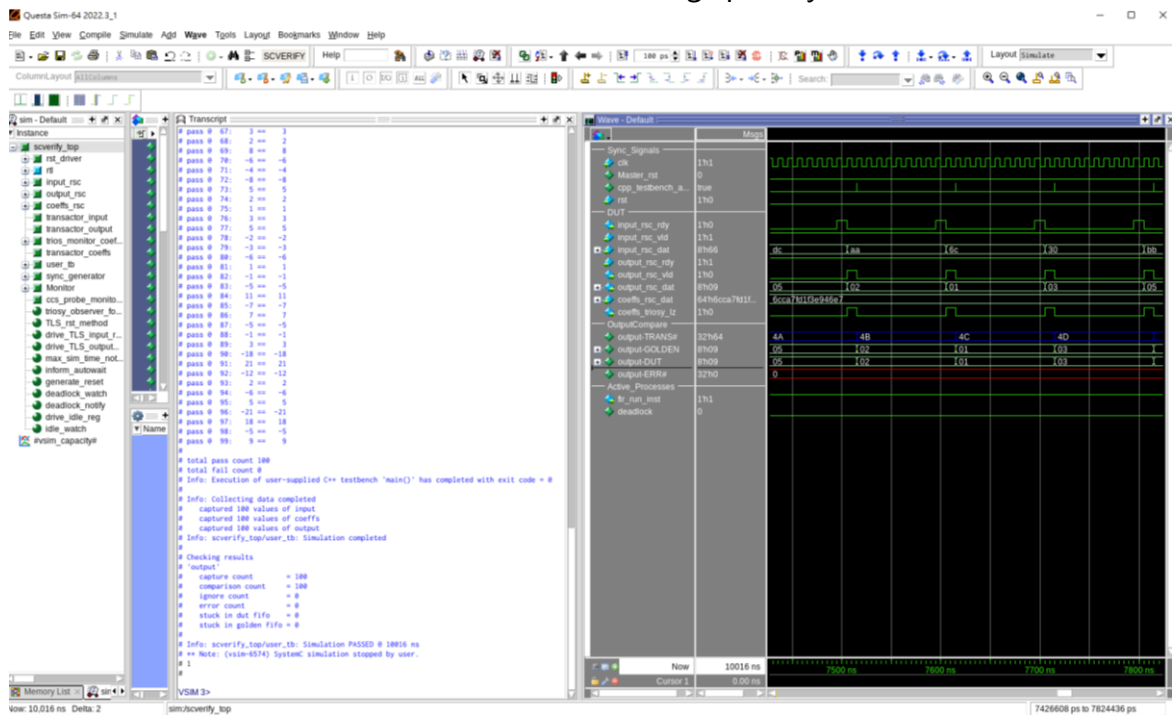
Transcript:

```
# Message
# Add dependent file: /home/raid7_4/raid1_1/linux/mentor/Catapult/2023.2/Mgc_home/pkgs/siflibs/ccs_out_wait_v1.v
# Add dependent file: /home/raid7_4/raid1_1/linux/mentor/Catapult/2023.2/Mgc_home/pkgs/siflibs/mgc_io_sync_v2.v
# Add dependent file: /home/raid7_4/raid1_1/linux/mentor/Catapult/2023.2/Mgc_home/pkgs/siflibs/ccs_in_v1.v
# Add dependent file: ./rtl.v
# Finished writing concatenated simulation file: /home/raid7_2/userb08/b08092/asoc/catapult/catapult-for-soc-course/lab1_fir/01_walkthrough_loops/Catapult/fi
# Makefile for RTL Verilog output 'rtl.v' vs Untimed C++ written to file './sverify/Verify_rtl_v_msim.mk'
# Makefile for Concat RTL Verilog output 'concat_sim_rtl.v' vs Untimed C++ written to file './sverify/Verify_concat_sim_rtl_v_msim.mk'
# Completed transformation 'extract' on solution 'fir.v1': elapsed time 10.58 seconds, memory usage 1401816kB, peak memory usage 1450916kB
# Design complexity at end of 'extract': Total ops = 275, Real ops = 126, Vars = 107
```

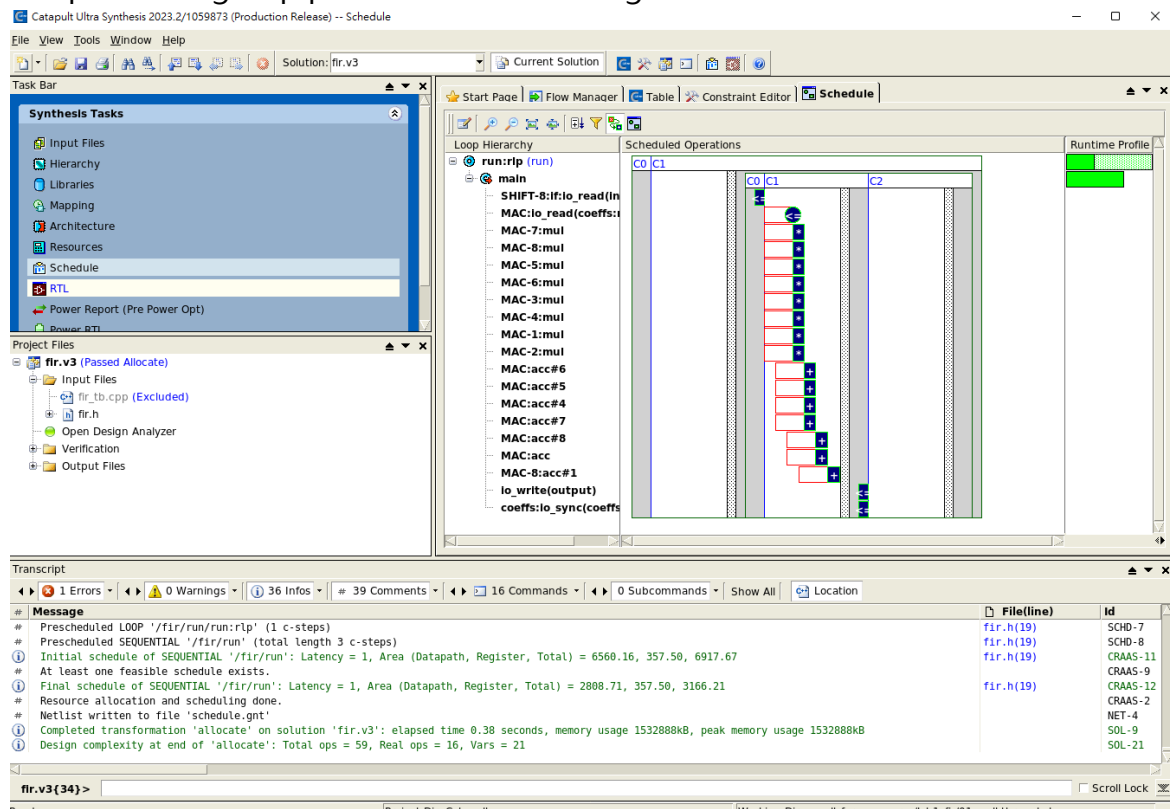
5. Design Analyzer 截圖: (可以看到一個加法器與一個乘法器構造)



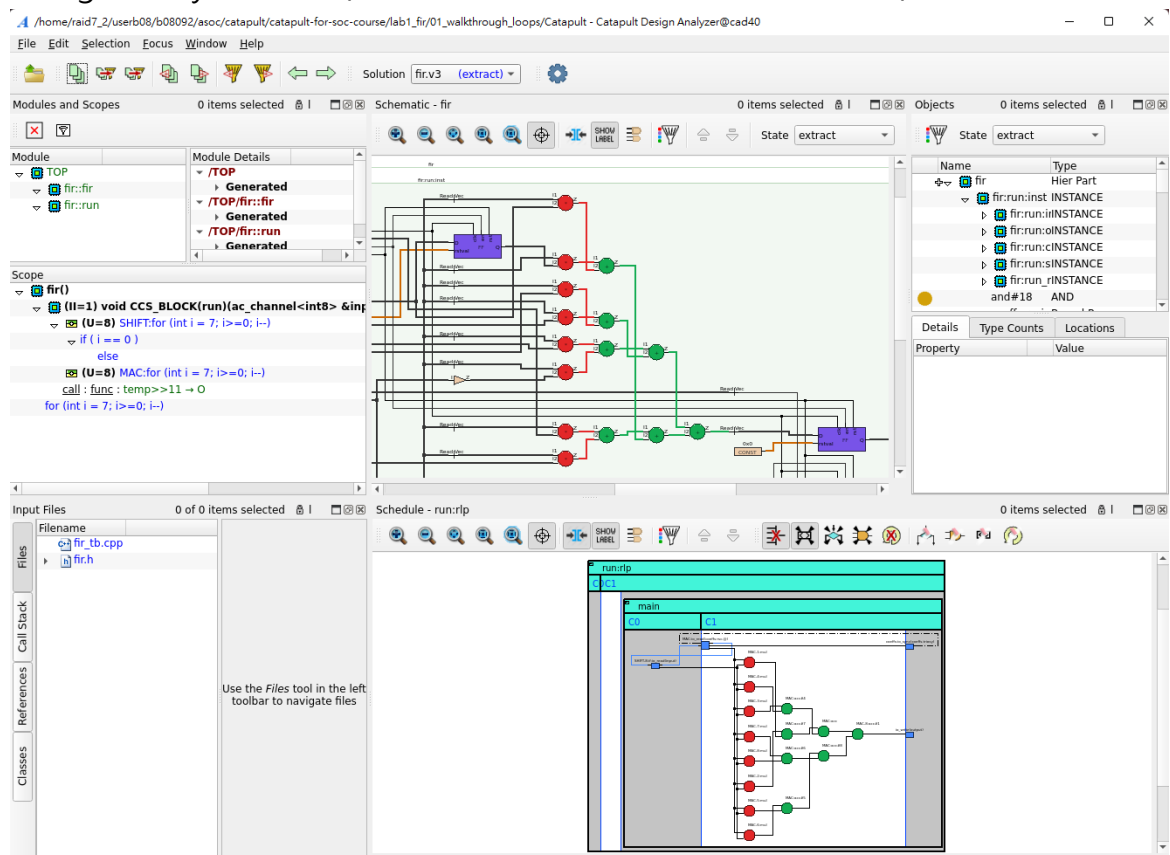
6. RTL Verification 截圖: (以及波形圖可看出 throughput_cycle)



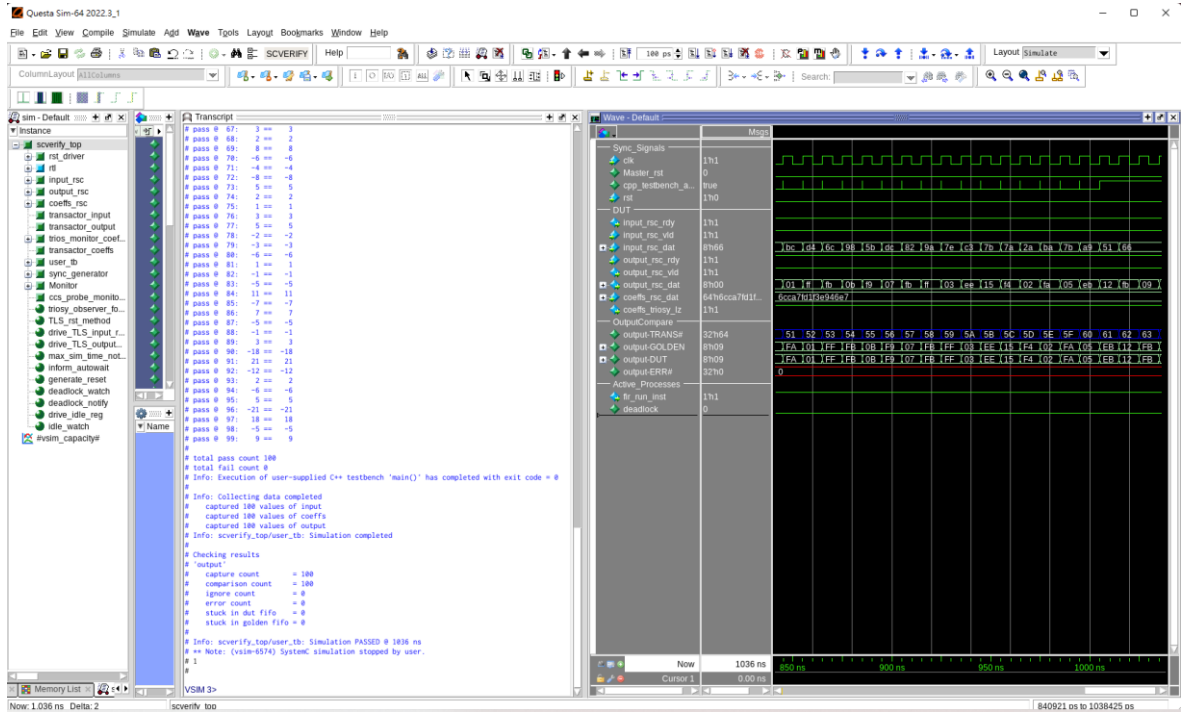
7. Loop unrolling 與 pipeline 後的 scheduling 如下圖:



8. Design Analyzer 截圖: (可以看到 7 個加法器與 8 個乘法器構造)

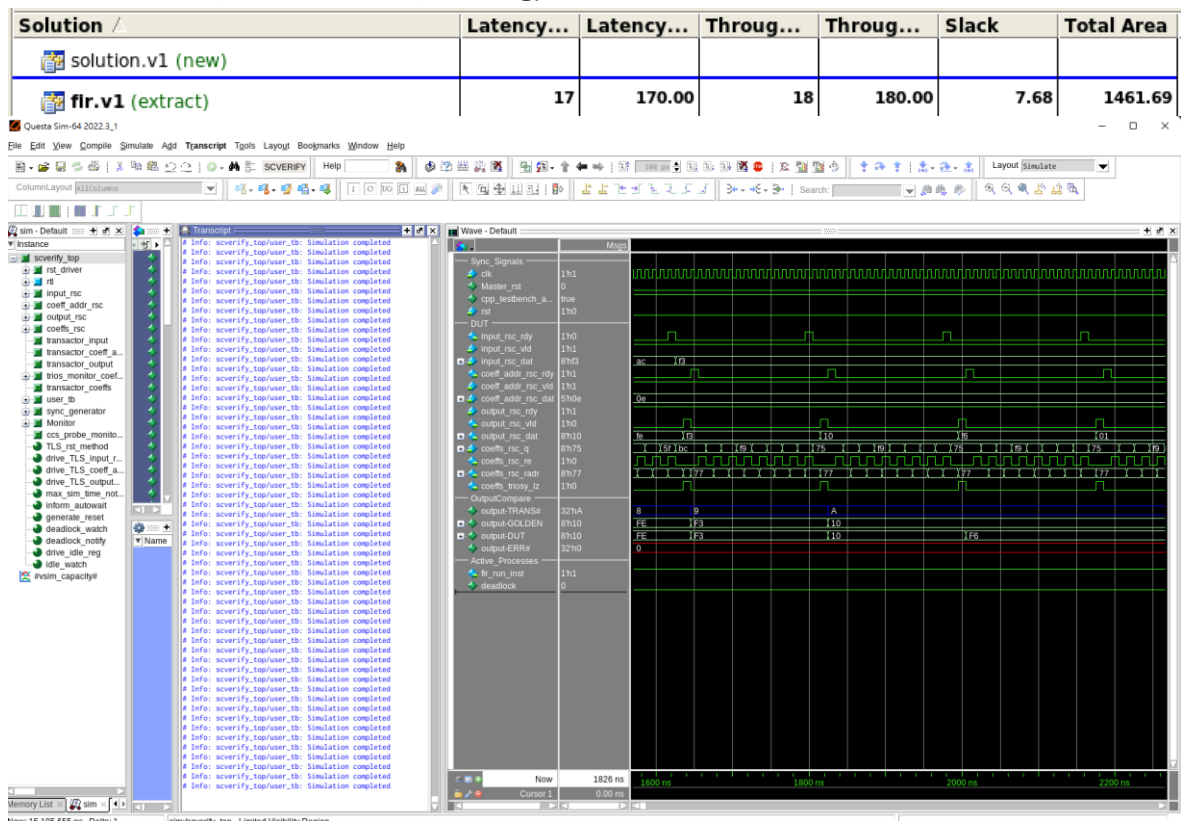


9. RTL Verification 截圖: (以及波形圖可看出 throughput_cycle)

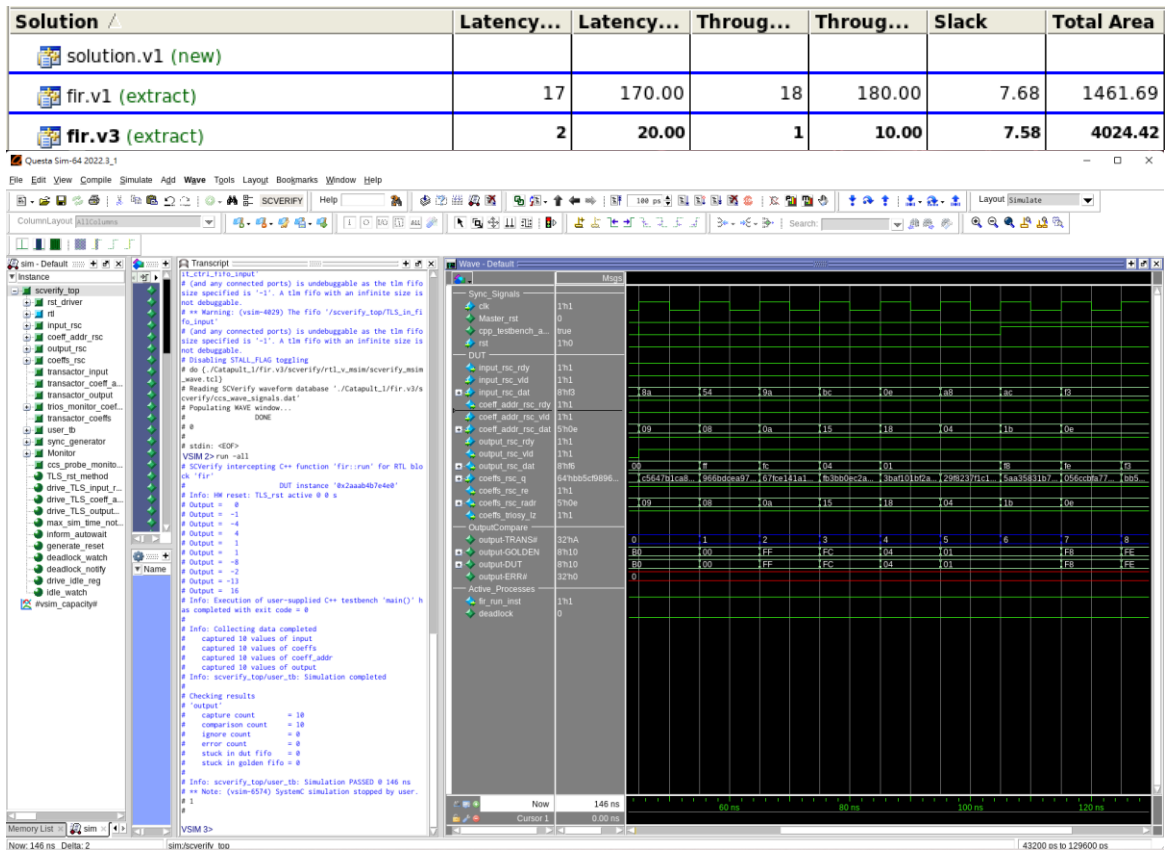


Part II : Memory Interface

- 跑 directive.tcl 檔後結果如下: (要將 options set Input/TargetPlatform x86_64 這行指令複製到 tcl 檔內一起跑才不會有 bug)

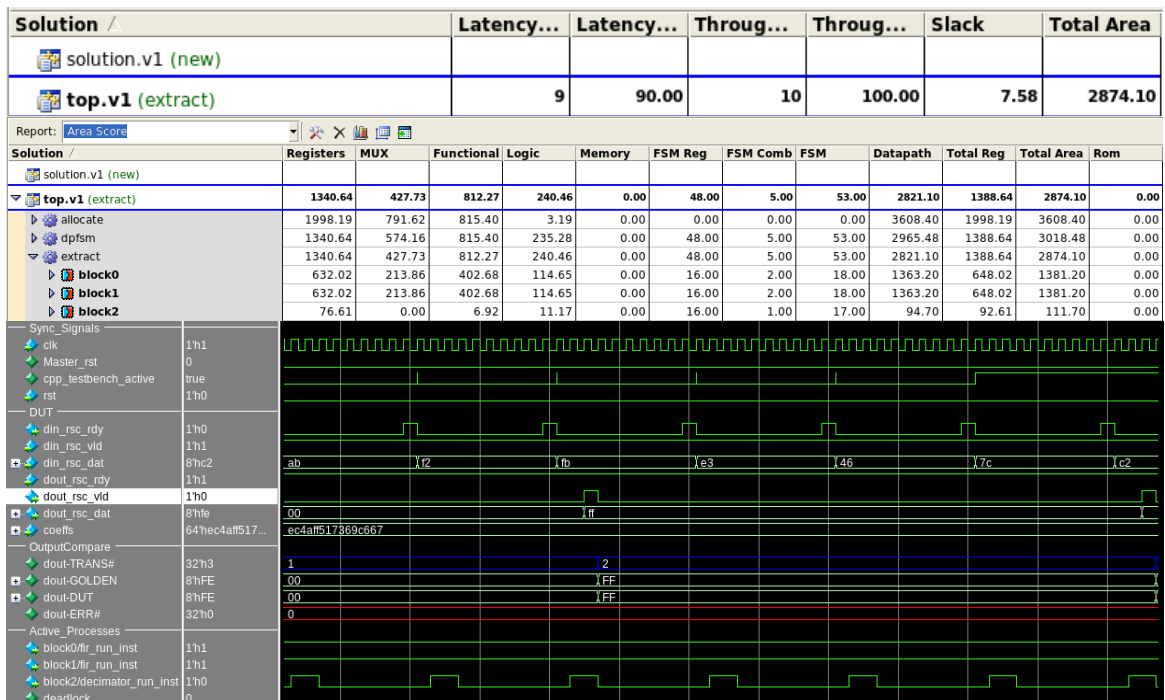


2. 再進行 unroll 與 pipeline，並且增加了記憶體資料寬度後的結果如下圖：

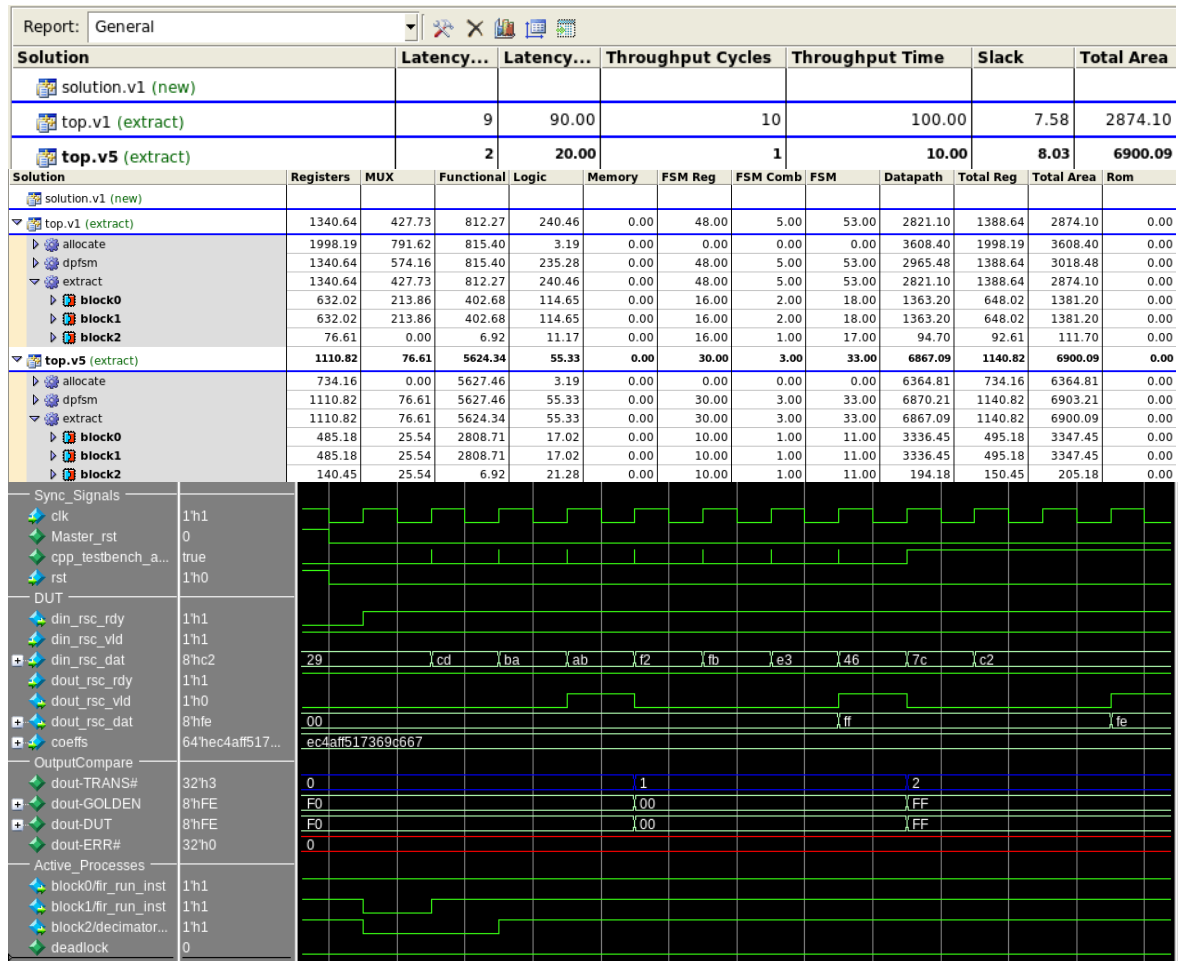


Part III : Multiple blocks

1. 跑 directive.tcl 檔後結果如下: (圖中可見 latency=9, throughput_cycle=10, block0_area=1381.2, block1_area=1381.2, block2_area=111.7)



2. 為將 throughput_cycle 降至 1，我們模仿前面的方法將所有 main 進行 pipeline，所有 SHIFT 與 MAC 函式進行 unroll，即發現成功達成目標，如下圖:



然而，增加 throughput 的代價便是面積變得比原來大上許多，block0 與 block1 從 1381.2 上升至 3347.45，且 block2 從 111.70 上升至 205.18。