

NYCU 電子研究所 系統晶片設計 LAB6

呂紹愷 311510187 江尹凡 311510207 廖智偉 311510216

Simulation

1. Matrix multiplication result

```
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$ source run_clean
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$ source run_sim
Reading counter_la_mm.hex
counter_la_mm.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_mm.vcd opened for output.
LA Test 1 started
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x003e
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0044
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x004a
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0050
LA Test 2 passed
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$
```

原始矩陣乘法的結果為

	1	2	3	4
1	62	68	74	80
2	62	68	74	80
3	62	68	74	80
4	62	68	74	80

在範例 code 中去看了第一列的結果

而結果為 3e 44 4a 50，與正確答案吻合。

而我們將 Headfile 中的兩個矩陣做更改，並去檢查每個數值是否正確。

```
#define SIZE 4
int A[SIZE*SIZE] = {1, 2, 0, 1,
                    2, 0, 3, 2,
                    0, 1, 2, 3,
                    1, 0, 5, 1,
};
int B[SIZE*SIZE] = {1, 2, 3, 4,
                    5, 6, 7, 8,
                    9, 10, 11, 12,
                    13, 14, 15, 16,
};
int result[SIZE*SIZE];
```

	1	2	3	4
1	24	28	32	36
2	55	62	69	76
3	62	68	74	80
4	59	66	73	80

更改 counter_la_mm.c 的 code，檢查所有的結果

```
int *tmp = matmul();
reg_mprj_data1 = *tmp << 16;
reg_mprj_data1 = *(tmp+1) << 16;
reg_mprj_data1 = *(tmp+2) << 16;
reg_mprj_data1 = *(tmp+3) << 16;

reg_mprj_data1 = *(tmp+4) << 16;
reg_mprj_data1 = *(tmp+5) << 16;
reg_mprj_data1 = *(tmp+6) << 16;
reg_mprj_data1 = *(tmp+7) << 16;
reg_mprj_data1 = *(tmp+8) << 16;
reg_mprj_data1 = *(tmp+9) << 16;
reg_mprj_data1 = *(tmp+10) << 16;
reg_mprj_data1 = *(tmp+11) << 16;

reg_mprj_data1 = *(tmp+12) << 16;
reg_mprj_data1 = *(tmp+13) << 16;
reg_mprj_data1 = *(tmp+14) << 16;
reg_mprj_data1 = *(tmp+15) << 16;

//print("\n");
//print("Monitor: Test 1 Passed\n\n"); // Makes simulation very long!
reg_mprj_data1 = 0xAB510000;
```

更改 Testbench

```
$display("LA Test 1 started");
//wait(checkbits == 16'hAB41);

wait(checkbits == 16'h0018); //24
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h001C); //28
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h0020); //32
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h0024); //36
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);

wait(checkbits == 16'h0037); //55
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h003E); //62
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h0045); //69
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h004C); //76
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);

wait(checkbits == 16'h003E); //62
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h0044); //68
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h004A); //74
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h0050); //80
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);

wait(checkbits == 16'h003B); //59
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h0042); //66
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h0049); //73
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
wait(checkbits == 16'h0050); //80
$display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);

wait(checkbits == 16'hAB51);
$display("LA Test 2 passed");
#10000;
$finish;
```

計算結果正確。

```
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$ source run_clean
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$ source run_sim
Reading counter_la_mm.hex
counter_la_mm.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_mm.vcd opened for output.
LA Test 1 started
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0018
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x001c
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0020
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0024
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0037
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x003e
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0045
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x004c
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x003e
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0044
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x004a
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0050
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x003b
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0042
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0049
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0050
LA Test 2 passed
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$
```

2. FIR result

結果應該為{0,-10,-29,-25,35,158,337,539,732,915,1098}

挑後面幾個答案檢查

```
initial begin
    wait(checkbits == 16'hAB40);
    $display("LA Test 1 started");
    //wait(checkbits == 16'hAB41);

    wait(checkbits == 16'd539);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd732);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd915);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd1098);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);

    wait(checkbits == 16'hAB51);
    $display("LA Test 2 passed");
    #10000;
    $finish;
end
```

計算結果正確。

```
ubuntu@ubuntu2004:~/Lab06/testbench/counter_la_fir$ source run_sim
Reading counter_la_fir.hex
counter_la_fir.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_fir.vcd opened for output.
LA Test 1 started
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 539
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 732
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 915
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 1098
LA Test 2 passed
```

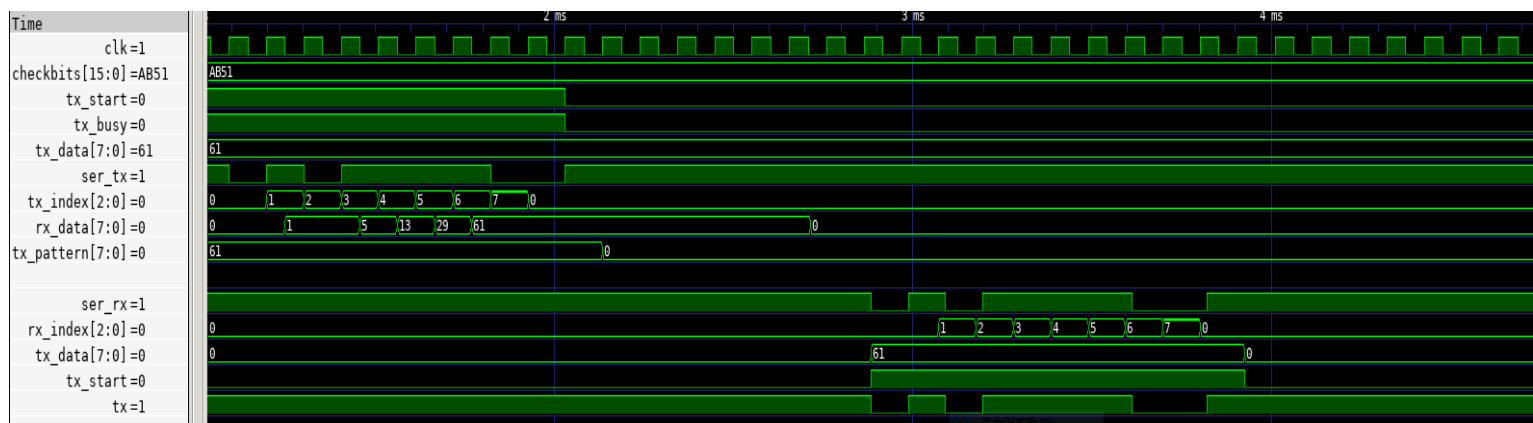
3. Quick sort

```
#define SIZE 10
int A[SIZE] = {893, 40, 3233, 4267, 2669, 2541, 9073, 6023, 5681, 4622};
#endif
```

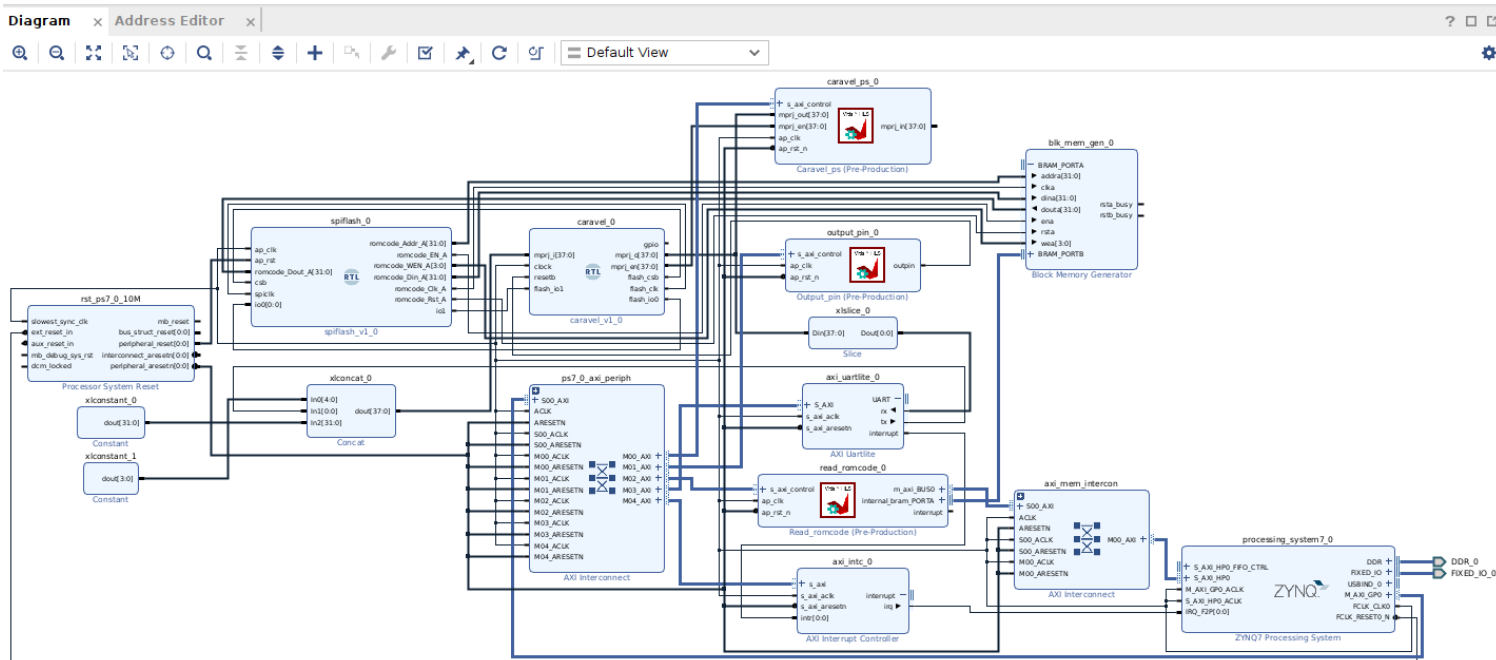
```
ubuntu@ubuntu2004:~/Lab06/testbench/counter_la_qs$ source run_sim
Reading counter_la_qs.hex
counter_la_qs.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_qs.vcd opened for output.
LA Test 1 started
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 40
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 893
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 2541
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 2669
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 3233
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 4267
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 4622
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 5681
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 6023
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x 9073
LA Test 2 passed
```

4. UART

```
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/uart$ source run_clean
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/uart$ source run_sim
Reading uart.hex
uart.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile uart.vcd opened for output.
LA Test 1 started
tx data bit index 0: 1
tx data bit index 1: 0
tx data bit index 2: 1
tx data bit index 3: 1
tx data bit index 4: 1
tx data bit index 5: 1
tx data bit index 6: 0
tx data bit index 7: 0
tx complete 2
LA Test 1 passed
rx data bit index 0: 1
rx data bit index 1: 0
rx data bit index 2: 1
rx data bit index 3: 1
rx data bit index 4: 1
rx data bit index 5: 1
rx data bit index 6: 0
rx data bit index 7: 0
received word 61
```



- Block diagram



Timing report

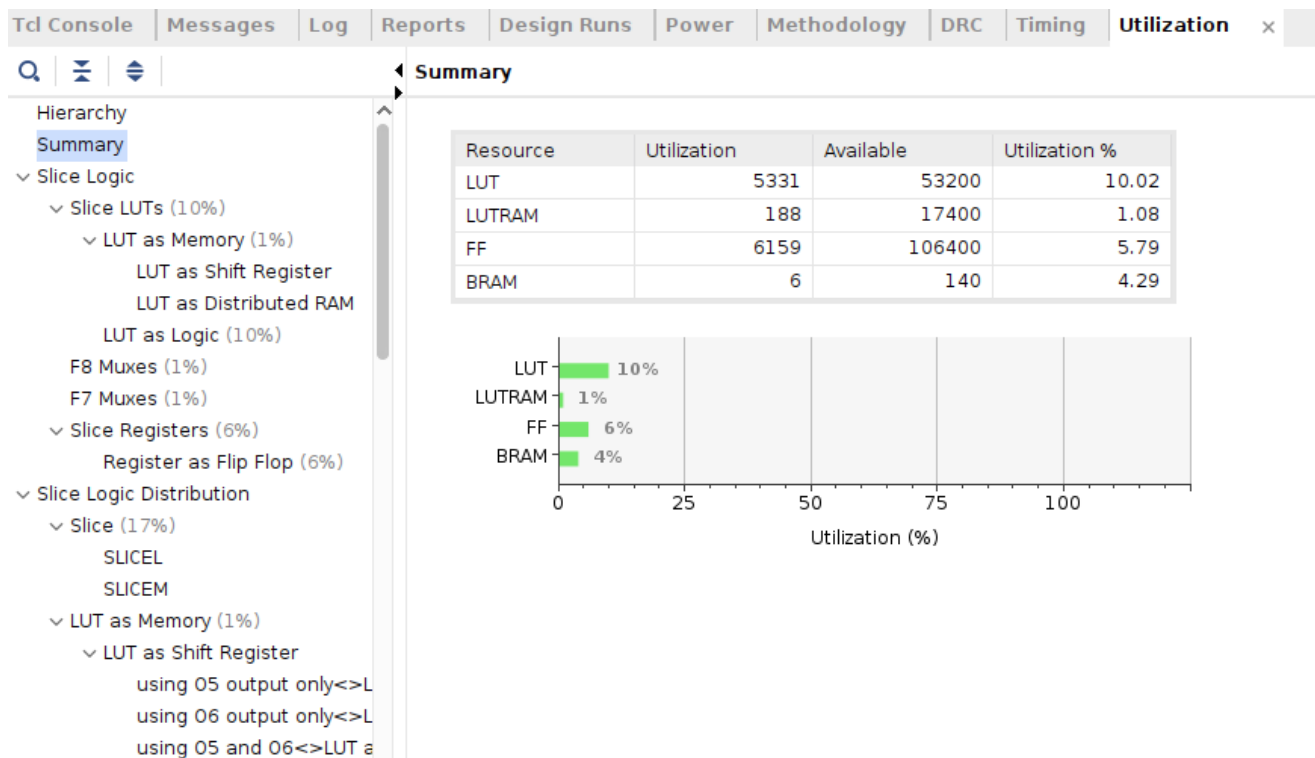
Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 9.001 ns	Worst Hold Slack (WHS): 0.043 ns	Worst Pulse Width Slack (WPWS): 11.250 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 12669	Total Number of Endpoints: 12669	Total Number of Endpoints: 5261

All user specified timing constraints are met.

Utilization

[illegible]

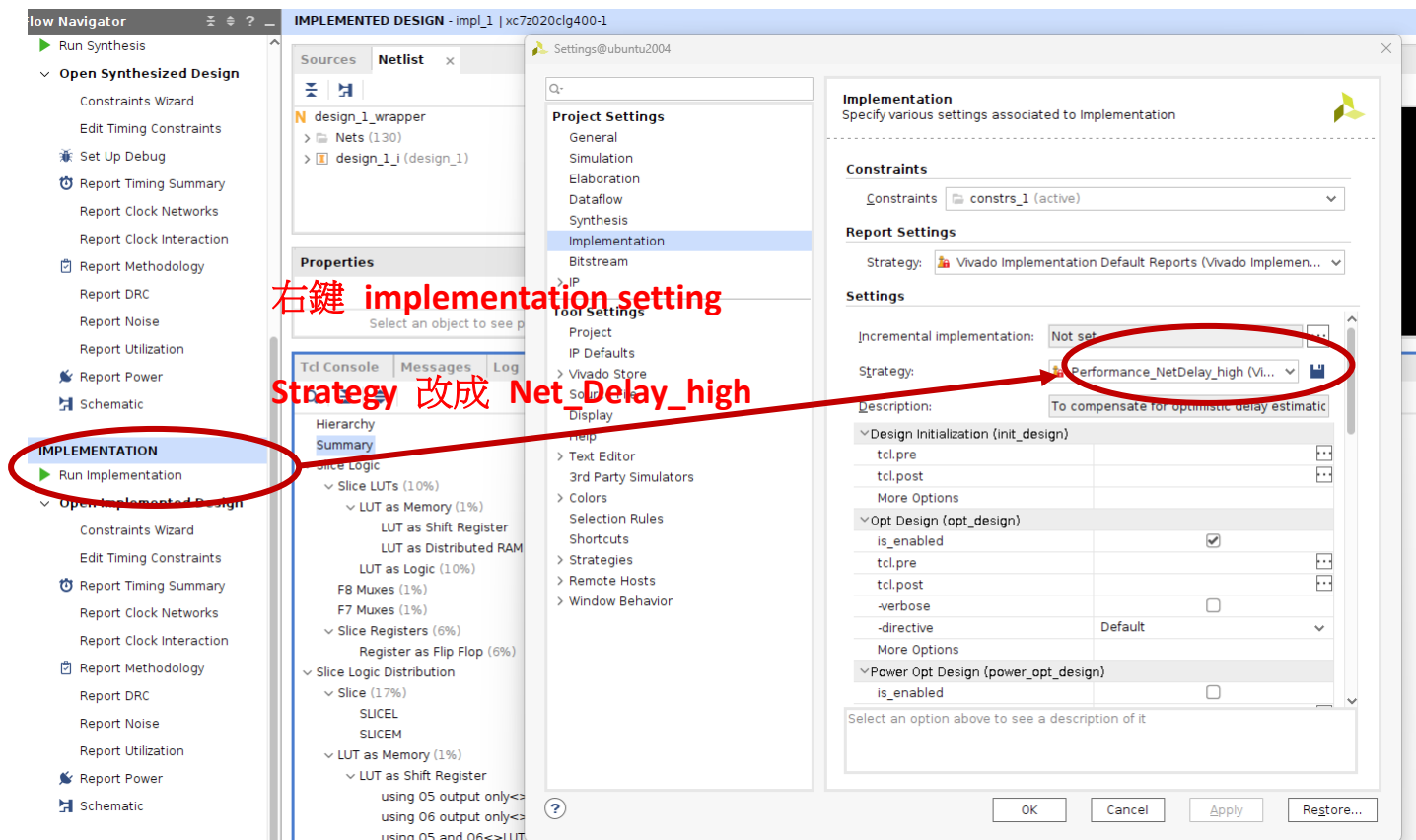


原本我們也有遇到 timeing-violation，

參考討論區給的方法得以解決

```
Reading XDEF routing.
Read XDEF Files: Time (s): cpu = 00:00:00.6 ; elapsed = 00:00:00.6 . Memory (MB): peak = 3402.273 ; gain = 6.938 ; free p
Restored from archive | CPU: 0.600000 secs | Memory: 14.330307 MB |
Finished XDEF File Restore: Time (s): cpu = 00:00:00.6 ; elapsed = 00:00:00.6 . Memory (MB): peak = 3402.273 ; gain = 6.9
7385
Netlist sorting complete. Time (s): cpu = 00:00:00.01 ; elapsed = 00:00:00 . Memory (MB): peak = 3402.293 ; gain = 0.000
9
INFO: [Project 1-111] Unisim Transformation Summary:
A total of 5 instances were transformed.
RAM32M => RAM32M (RAMD32(x6), RAMS32(x2)): 4 instances
RAM32X1D => RAM32X1D (RAMD32(x2)): 1 instance

open_run: Time (s): cpu = 00:00:08 ; elapsed = 00:00:09 . Memory (MB): peak = 3402.293 ; gain = 761.949 ; free physical =
# report_timing_summary -file timing_report.log
INFO: [Timing 38-91] UpdateTimingParams: Speed grade: -1, Delay Type: min_max.
INFO: [Timing 38-191] Multithreading enabled for timing update using a maximum of 2 CPUs
# exit
INFO: [Common 17-206] Exiting Vivado at Thu Dec 14 09:00:56 2023...
CRITICAL WARNING: [Timing 38-282] The design failed to meet the timing requirements. Please see the timing summary report
=====
vivado complete
```



一、Jupyter Notebook

```
In [8]: # Initialize AXI UART
        uart = UartAXI(ipUart.mmio.base_addr)

        # Setup AXI UART register
        uart.setupCtrlReg()

        # Get current UART status
        uart.currentStatus()
```

```
Out[8]: {'RX_VALID': 0,
        'RX_FULL': 0,
        'TX_EMPTY': 1,
        'TX_FULL': 0,
        'IS_INTR': 0,
        'OVERRUN_ERR': 0,
        'FRAME_ERR': 0,
        'PARITY_ERR': 0}
```

```
await asyncio.sleep(10)
task1.cancel()
try:
    await task1
except asyncio.CancelledError:
    print('main(): uart_rx is cancelled now')
```

```
In [10]: asyncio.run(async_main())
```

```
Start Caravel Soc
Waiting for interrupt
hello
main(): uart_rx is cancelled now
```

```
In [11]: print ("0x10 = ", hex(ipPS.read(0x10)))
        print ("0x14 = ", hex(ipPS.read(0x14)))
        print ("0x1c = ", hex(ipPS.read(0x1c)))
        print ("0x20 = ", hex(ipPS.read(0x20)))
        print ("0x34 = ", hex(ipPS.read(0x34)))
        print ("0x38 = ", hex(ipPS.read(0x38)))
```

```
0x10 = 0x0
0x14 = 0x0
0x1c = 0xab510040
0x20 = 0x0
0x34 = 0x20
0x38 = 0x3f
```