# SOC LAB

## LAB6 Report

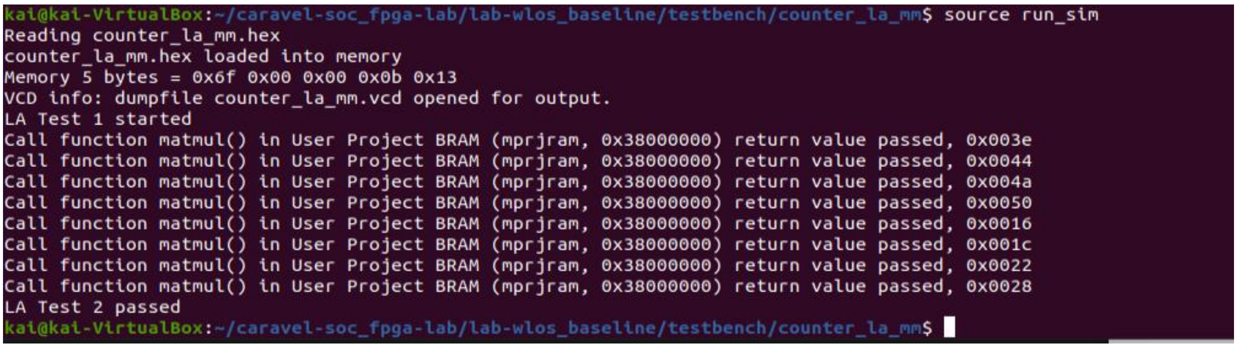
Group 13

2023/12/06

# Run Simulation

## 1. Matrix multiplication

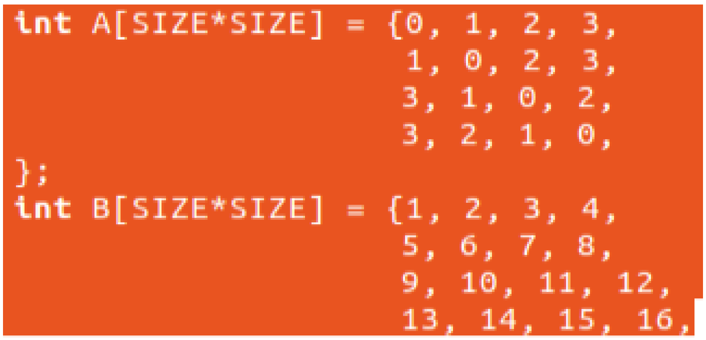
- 執行 simulation 結果:



```
kai@kai-VirtualBox:~/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
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0016
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, 0x0022
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0028
LA Test 2 passed
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$ 
```

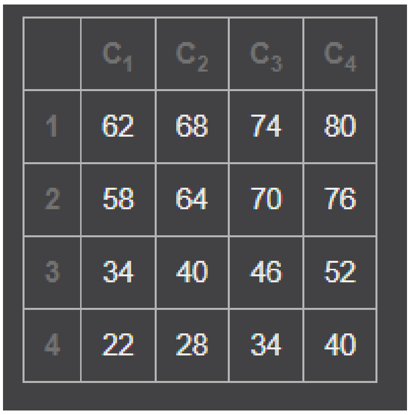
- 驗證方法:

我們在 headfile 預設的矩陣參數如下



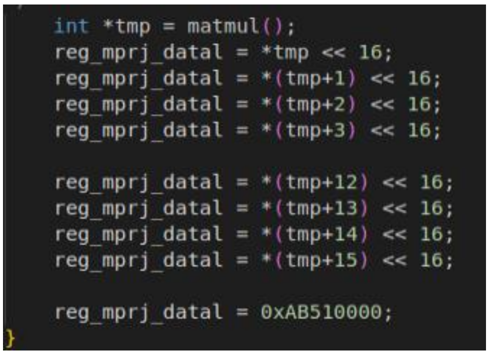
```
int A[SIZE*SIZE] = {0, 1, 2, 3,
                     1, 0, 2, 3,
                     3, 1, 0, 2,
                     3, 2, 1, 0,
};
int B[SIZE*SIZE] = {1, 2, 3, 4,
                     5, 6, 7, 8,
                     9, 10, 11, 12,
                     13, 14, 15, 16,
```

相乘後的結果應為



|   | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| 1 | 62 | 68 | 74 | 80 |
| 2 | 58 | 64 | 70 | 76 |
| 3 | 34 | 40 | 46 | 52 |
| 4 | 22 | 28 | 34 | 40 |

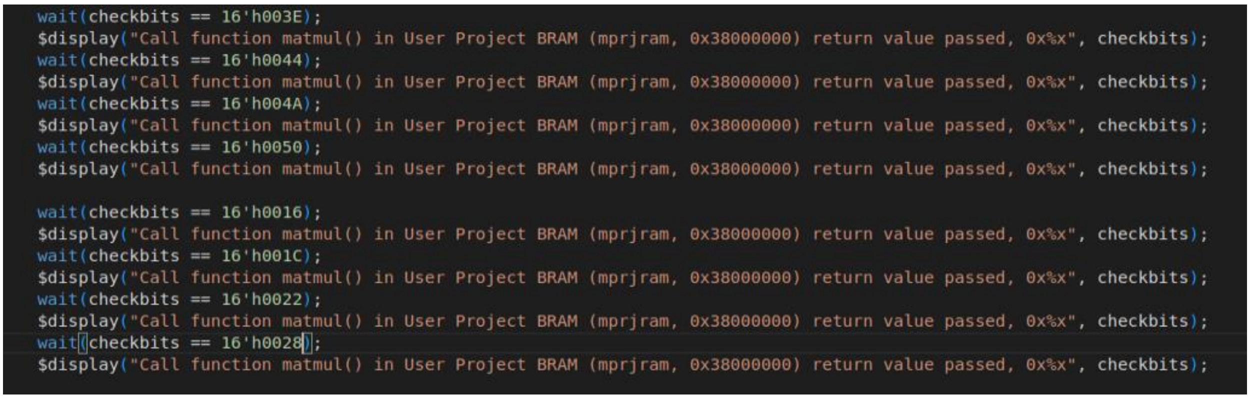
透過 testbench 的 checkbits 檢查運算結果是否正確，在此我們檢查了第一列與最後一列的結果



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

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

    reg_mprj_datal = 0xAB510000;
}
```

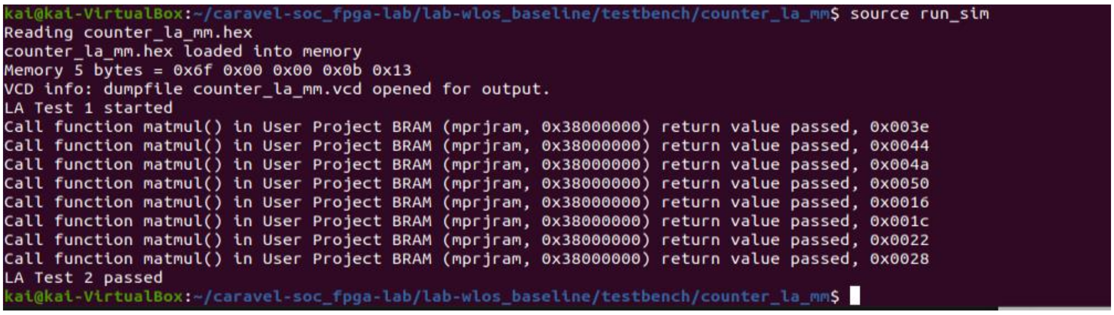
輸出結果依序應為{62, 68, 74, 80, 22, 28, 34, 40}

轉為 16 進制: {3E, 44, 4A, 50, 16, 1C, 22, 28}



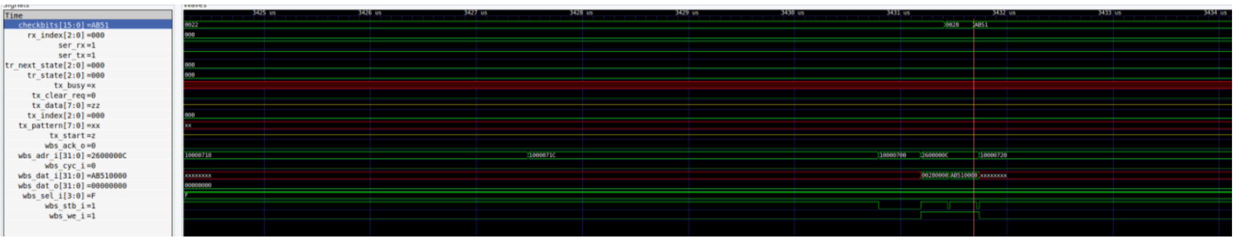
```
    wait(checkbits == 16'h003E);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h0044);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);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

    wait(checkbits == 16'h0016);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h001C);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h0022);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h0028);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
```

最終輸出結果正確



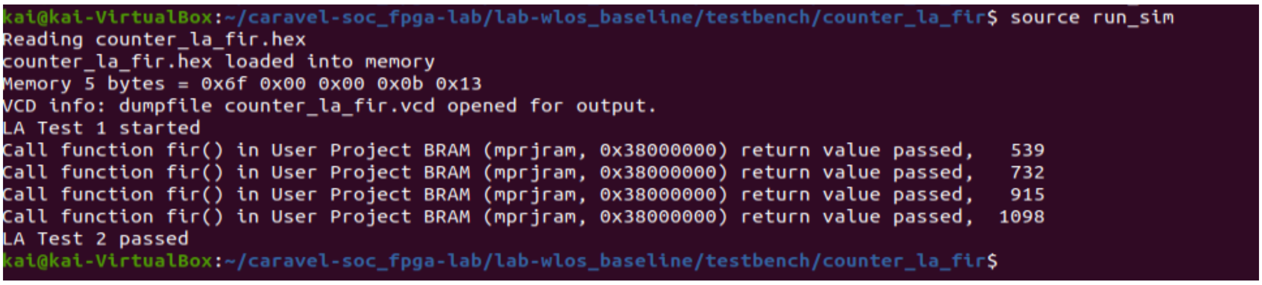
```
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$ source run_sim
Reading counter_la_mm.hex
counter_la_mm.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_mm.vcd opened for output.
LA Test 1 started
Call function matmul() 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
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0016
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, 0x0022
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0028
LA Test 2 passed
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_mm$ 
```

- waveform



## 2. Fir

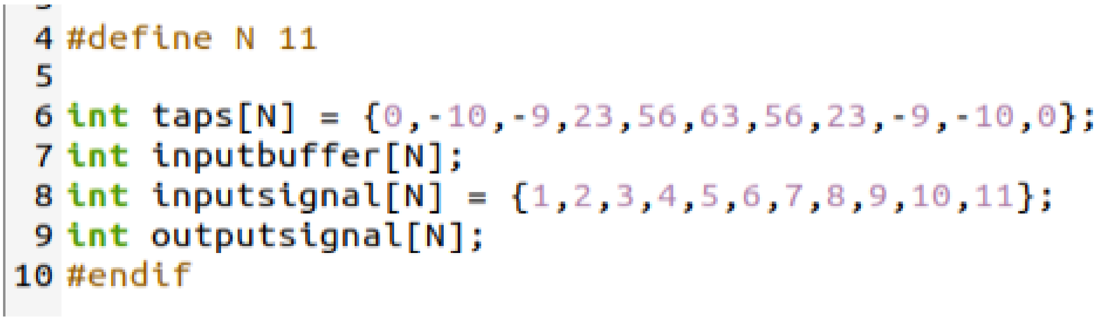
- 執行 simulation 結果:



```
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/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 fir() in User Project BRAM (mprjram, 0x38000000) return value passed,   539
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed,   732
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed,   915
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed,  1098
LA Test 2 passed
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_fir$
```

- 驗證方法:

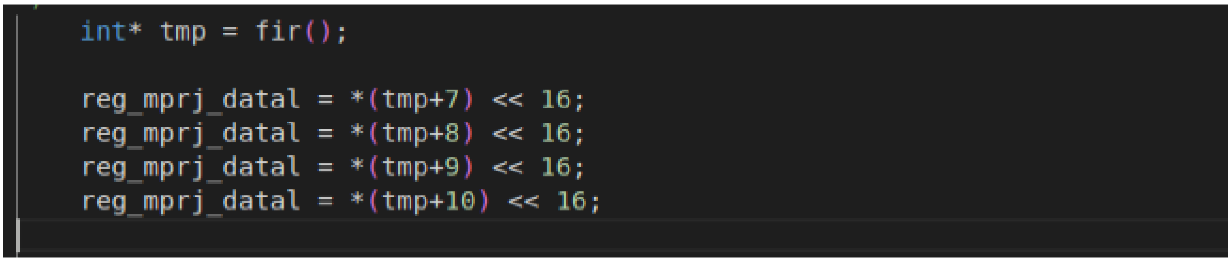
我們在 headfile 定義的 fir 參數如下



```
#define N 11

int taps[N] = {0,-10,-9,23,56,63,56,23,-9,-10,0};
int inputbuffer[N];
int inputsignal[N] = {1,2,3,4,5,6,7,8,9,10,11};
int outputsignal[N];
#endif
```

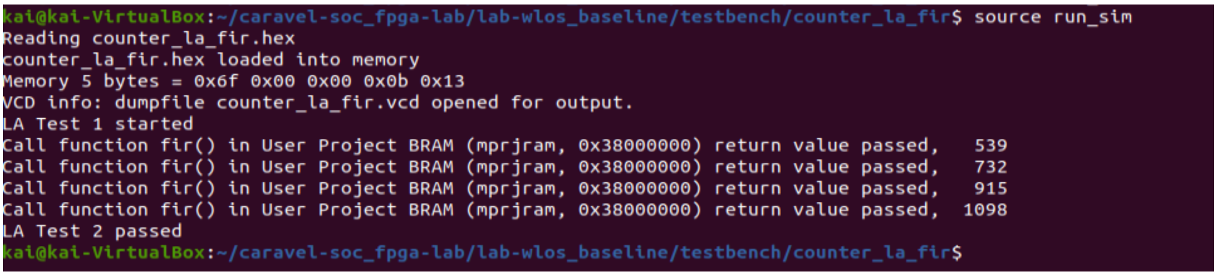
運算後的結果應為{0, -10, -29, -25, 35, 158, 337, 539, 732, 915, 1098}

透過 testbench 的 checkbits 檢查運算結果是否正確，在此我們檢查了最後四筆的結果



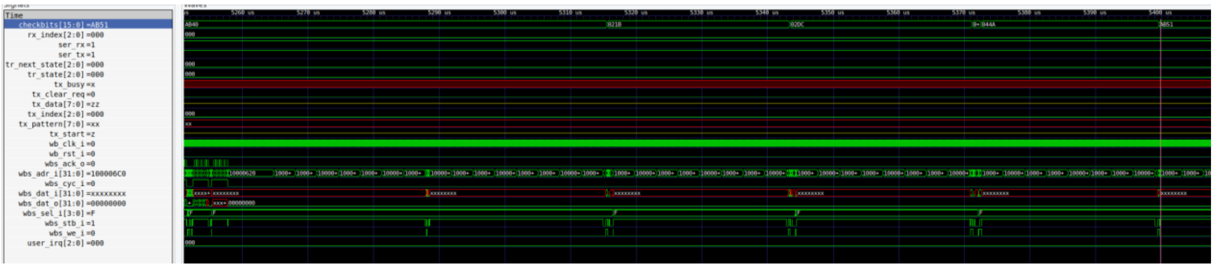
```
    int* tmp = fir();

    reg_mprj_datal = *(tmp+7) << 16;
    reg_mprj_datal = *(tmp+8) << 16;
    reg_mprj_datal = *(tmp+9) << 16;
    reg_mprj_datal = *(tmp+10) << 16;
```

最終輸出結果正確



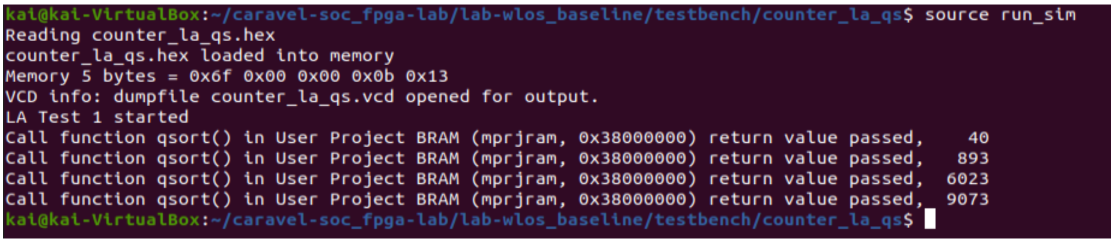
```
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_fir$ source run_sim
Reading counter_la_fir.hex
counter_la_fir.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_fir.vcd opened for output.
LA Test 1 started
Call function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

- waveform



## 3. Quick sort

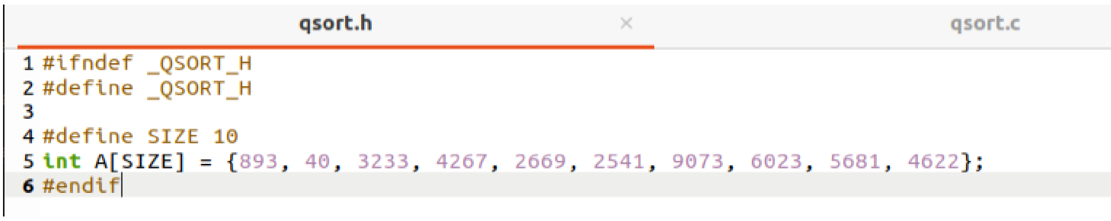
- 執行 simulation 結果:



```
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/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 qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,    40
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,   893
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,  6023
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,  9073
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_qs$ 
```

- 驗證方法:

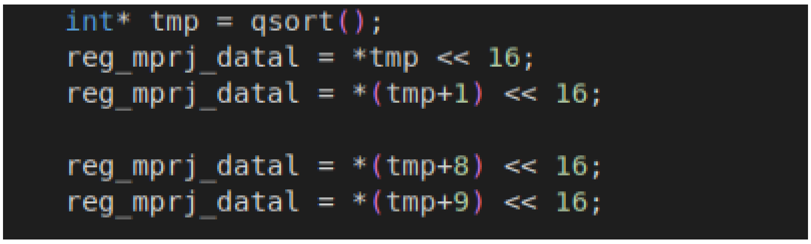
我們在 headfile 定義的參數如下



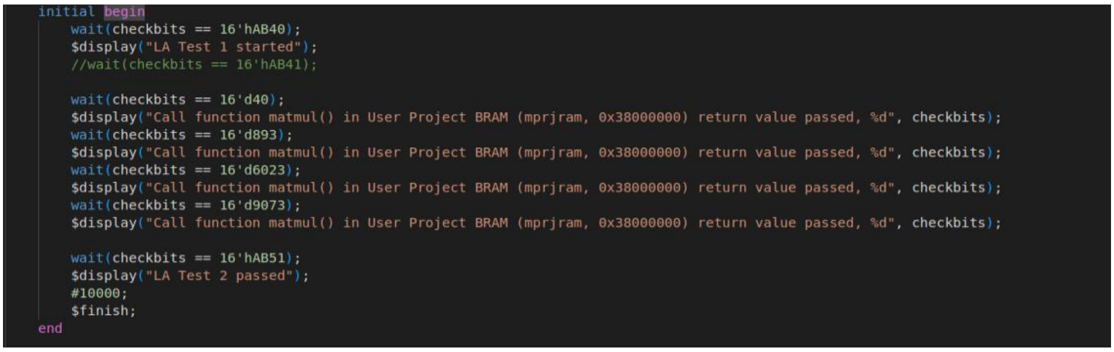
```c
#ifndef _QSORT_H
#define _QSORT_H

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

經排列後結果應為{40, 893, 2541, 2669, 3233, 4267, 4622, 5681, 6023, 9073}

透過 testbench 的 checkbits 檢查運算結果是否正確，在此我們檢查了前兩筆與最後兩筆的結果

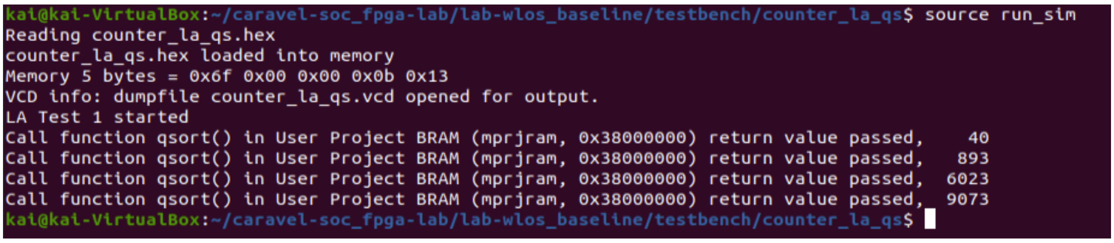


```c
int* tmp = qsort();
reg_mprj_datal = *tmp << 16;
reg_mprj_datal = *(tmp+1) << 16;

reg_mprj_datal = *(tmp+8) << 16;
reg_mprj_datal = *(tmp+9) << 16;
```



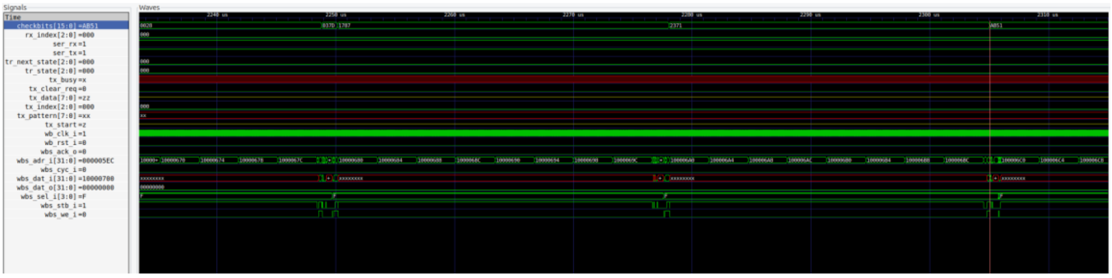
```verilog
initial begin
    wait(checkbits == 16'hAB40);
    $display("LA Test 1 started");
    //wait(checkbits == 16'hAB41);

    wait(checkbits == 16'd40);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd893);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd6023);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd9073);
    $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
```

最終輸出結果正確



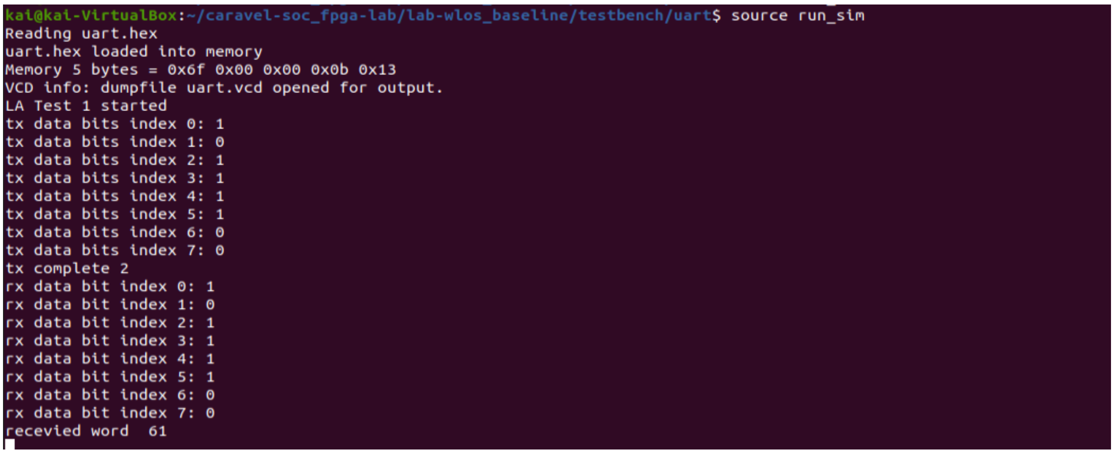
```
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_qs$ source run_sim
Reading counter_la_qs.hex
counter_la_qs.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_qs.vcd opened for output.
LA Test 1 started
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,    40
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,   893
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,  6023
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,  9073
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/counter_la_qs$ 
```

- waveform



# 4. Uart

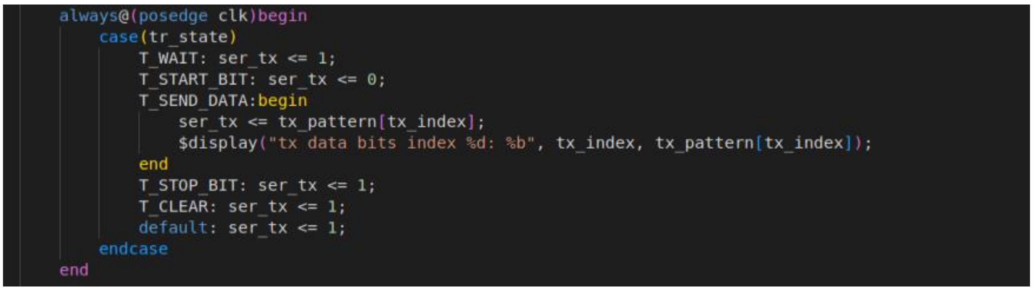
- 執行 simulation 結果:



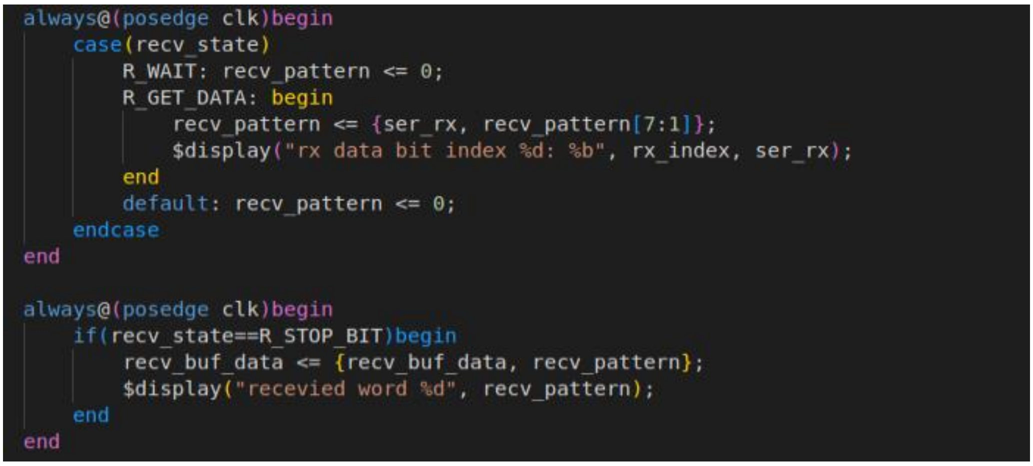
```
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/uart$ source run_sim
Reading uart.hex
uart.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile uart.vcd opened for output.
LA Test 1 started
tx data bits index 0: 1
tx data bits index 1: 0
tx data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

- 驗證方法:

Tbuart.v 會 check 送入及送出的 rx、tx

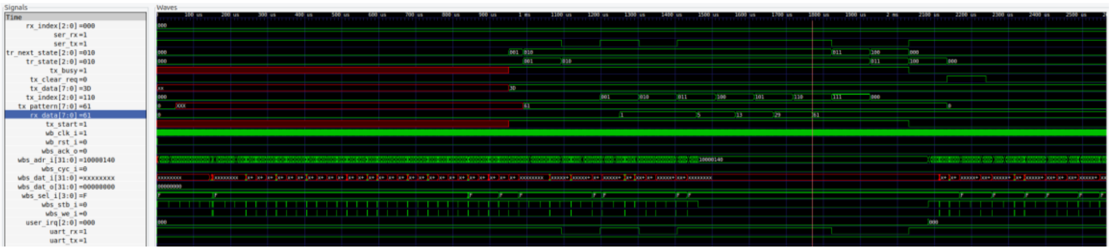


```
always@(posedge clk)begin
    case(tr_state)
        T_WAIT: ser_tx <= 1;
        T_START_BIT: ser_tx <= 0;
        T_SEND_DATA:begin
            ser_tx <= tx_pattern[tx_index];
            $display("tx data bits index %d: %b", tx_index, tx_pattern[tx_index]);
        end
        T_STOP_BIT: ser_tx <= 1;
        T_CLEAR: ser_tx <= 1;
        default: ser_tx <= 1;
    endcase
end
```



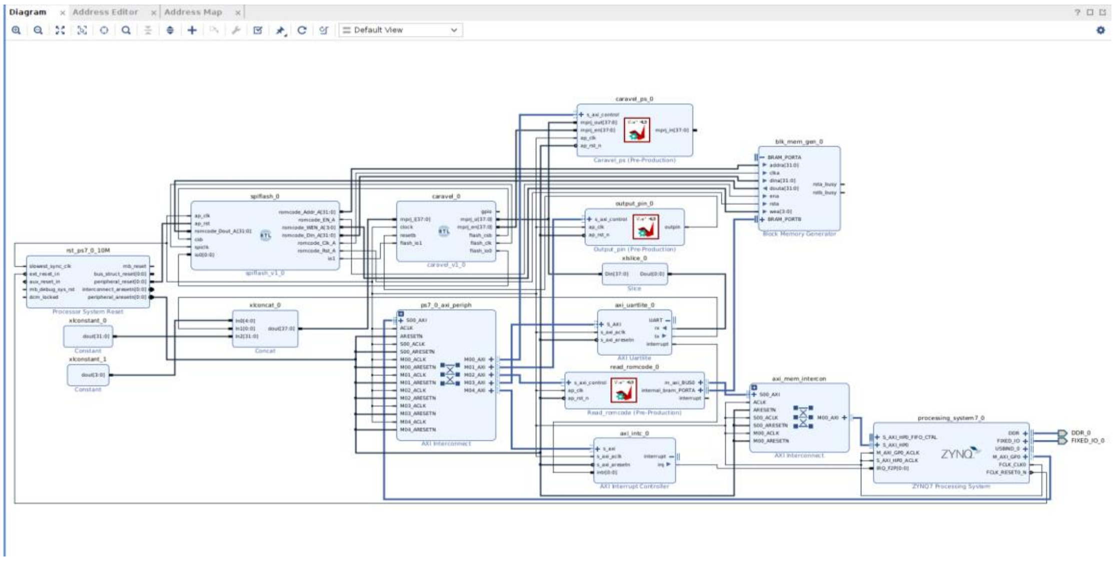
```
always@(posedge clk)begin
    case(recv_state)
        R_WAIT: recv_pattern <= 0;
        R_GET_DATA: begin
            recv_pattern <= {ser_rx, recv_pattern[7:1]};
            $display("rx data bit index %d: %b", rx_index, ser_rx);
        end
        default: recv_pattern <= 0;
    endcase
end

always@(posedge clk)begin
    if(recv_state==R_STOP_BIT)begin
        recv_buf_data <= {recv_buf_data, recv_pattern};
        $display("recevied word %d", recv_pattern);
    end
end
```

- waveform

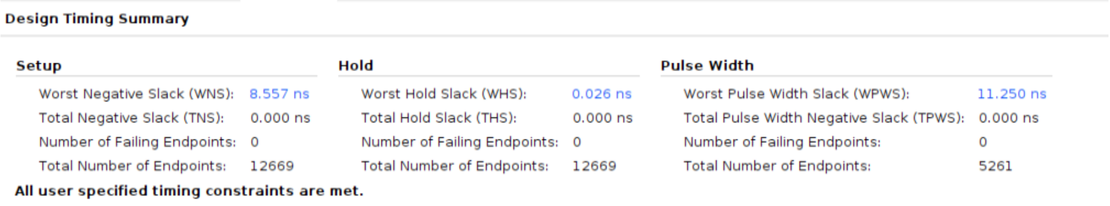


# Run UART on FPGA

- Block diagram



- Time report

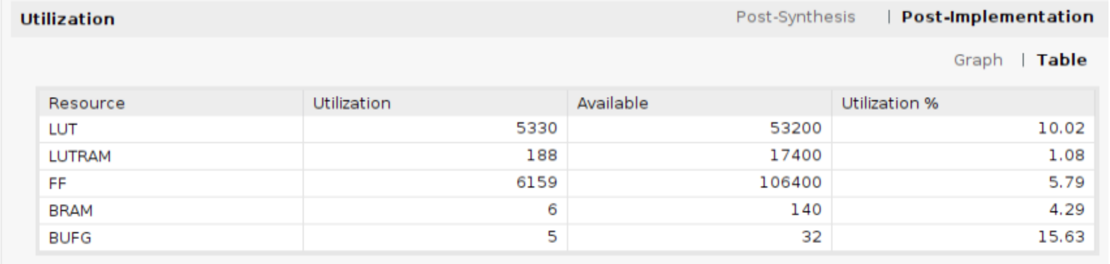


**Design Timing Summary**

| Setup | | Hold | | Pulse Width | |
|---|---|---|---|---|---|
| Worst Negative Slack (WNS): | 8.557 ns | Worst Hold Slack (WHS): | 0.026 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.**

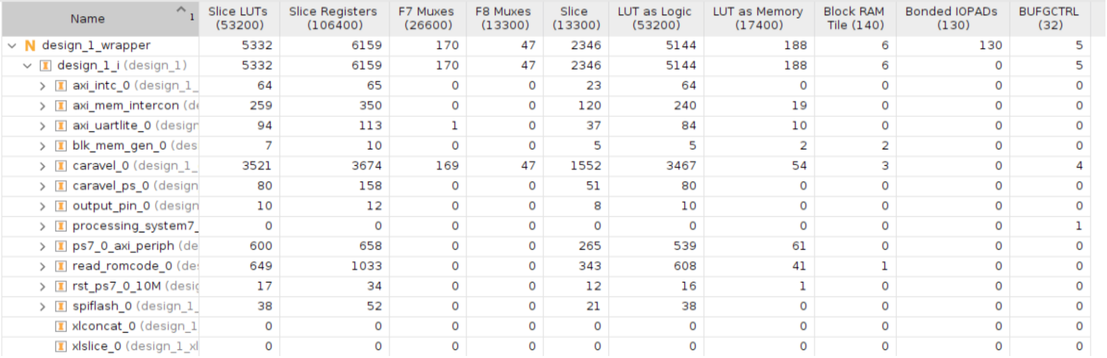
- Resource report



**Utilization** Post-Synthesis | **Post-Implementation**

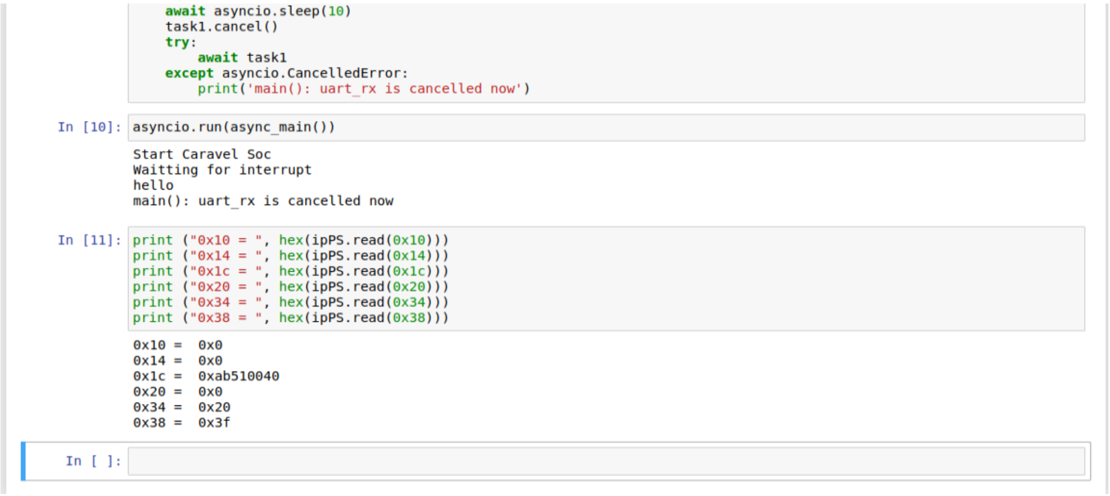
Graph | **Table**

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 5330 | 53200 | 10.02 |
| LUTRAM | 188 | 17400 | 1.08 |
| FF | 6159 | 106400 | 5.79 |
| BRAM | 6 | 140 | 4.29 |
| BUFG | 5 | 32 | 15.63 |



| Name | Slice LUTs (53200) | Slice Registers (106400) | F7 Muxes (26600) | F8 Muxes (13300) | Slice (13300) | LUT as Logic (53200) | LUT as Memory (17400) | Block RAM Tile (140) | Bonded IOPADs (130) | BUFGCTRL (32) |
|---|---|---|---|---|---|---|---|---|---|---|
| design_1_wrapper | 5332 | 6159 | 170 | 47 | 2346 | 5144 | 188 | 6 | 130 | 5 |
| design_1_i (design_1) | 5332 | 6159 | 170 | 47 | 2346 | 5144 | 188 | 6 | 0 | 5 |
| axi_intc_0 (design_1_ | 64 | 65 | 0 | 0 | 23 | 64 | 0 | 0 | 0 | 0 |
| axi_mem_intercon (de | 259 | 350 | 0 | 0 | 120 | 240 | 19 | 0 | 0 | 0 |
| axi_uartlite_0 (design | 94 | 113 | 1 | 0 | 37 | 84 | 10 | 0 | 0 | 0 |
| blk_mem_gen_0 (des | 7 | 10 | 0 | 0 | 5 | 5 | 2 | 2 | 0 | 0 |
| caravel_0 (design_1_ | 3521 | 3674 | 169 | 47 | 1552 | 3467 | 54 | 3 | 0 | 4 |
| caravel_ps_0 (design | 80 | 158 | 0 | 0 | 51 | 80 | 0 | 0 | 0 | 0 |
| output_pin_0 (design | 10 | 12 | 0 | 0 | 8 | 10 | 0 | 0 | 0 | 0 |
| processing_system7_ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ps7_0_axi_periph (de | 600 | 658 | 0 | 0 | 265 | 539 | 61 | 0 | 0 | 0 |
| read_romcode_0 (de | 649 | 1033 | 0 | 0 | 343 | 608 | 41 | 1 | 0 | 0 |
| rst_ps7_0_10M (desig | 17 | 34 | 0 | 0 | 12 | 16 | 1 | 0 | 0 | 0 |
| spiflash_0 (design_1_ | 38 | 52 | 0 | 0 | 21 | 38 | 0 | 0 | 0 | 0 |
| xlconcat_0 (design_1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| xlslice_0 (design_1_xl | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

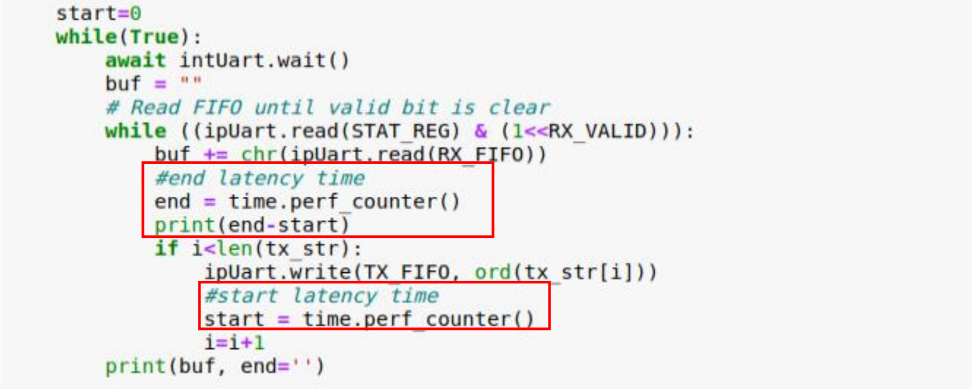
- result



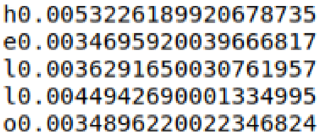
```
        await asyncio.sleep(10)
        task1.cancel()
        try:
            await task1
        except asyncio.CancelledError:
            print('main(): uart_rx is cancelled now')
```

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

         Start Caravel Soc
         Waitting 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

In [ ]:
```

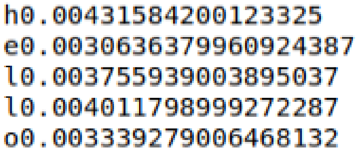
- Latency



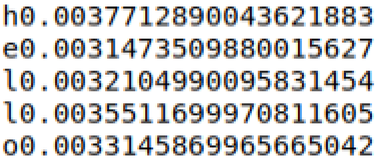
```
start=0
while(True):
    await intUart.wait()
    buf = ""
    # Read FIFO until valid bit is clear
    while ((ipUart.read(STAT_REG) & (1<<RX_VALID))):
        buf += chr(ipUart.read(RX_FIFO))
        #end latency time
        end = time.perf_counter()
        print(end-start)
        if i<len(tx_str):
            ipUart.write(TX_FIFO, ord(tx_str[i]))
            #start latency time
            start = time.perf_counter()
            i=i+1
    print(buf, end='')
```



```
h0.0053226189920678735
e0.0034695920039666817
l0.0036291650030761957
l0.0044942690001334995
o0.0034896220022346824
```



```
h0.00431584200123325
e0.0030636379960924387
l0.0037559390003895037
l0.0040117989992722287
o0.003339279006468132
```



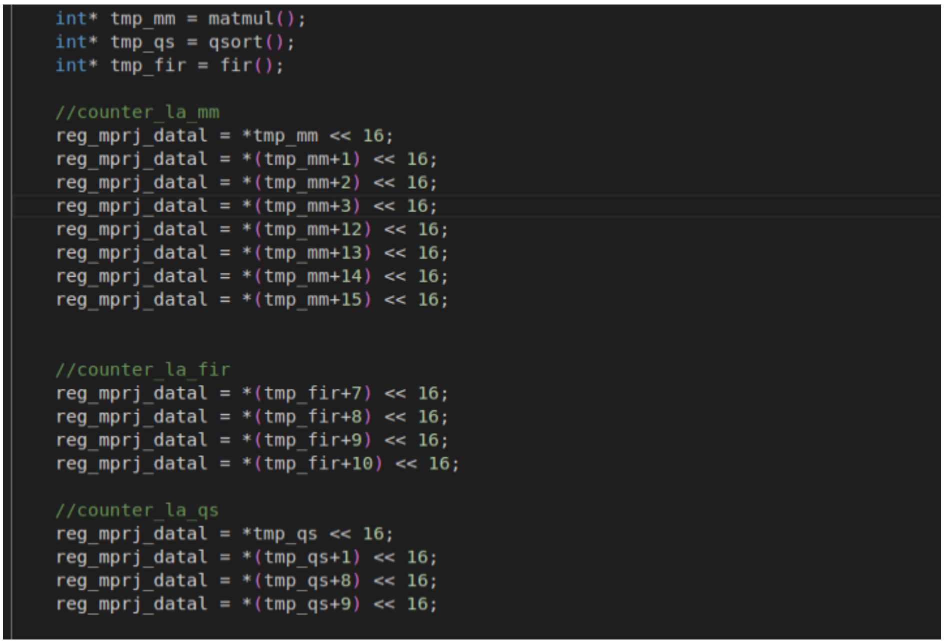
```
h0.0037712890043621883
e0.0031473509880015627
l0.0032104990095831454
l0.0035511699970811605
o0.0033145869965665042
```

實測三輪後得每個 character loop back 的 latency time 約為 0.003~0.004s

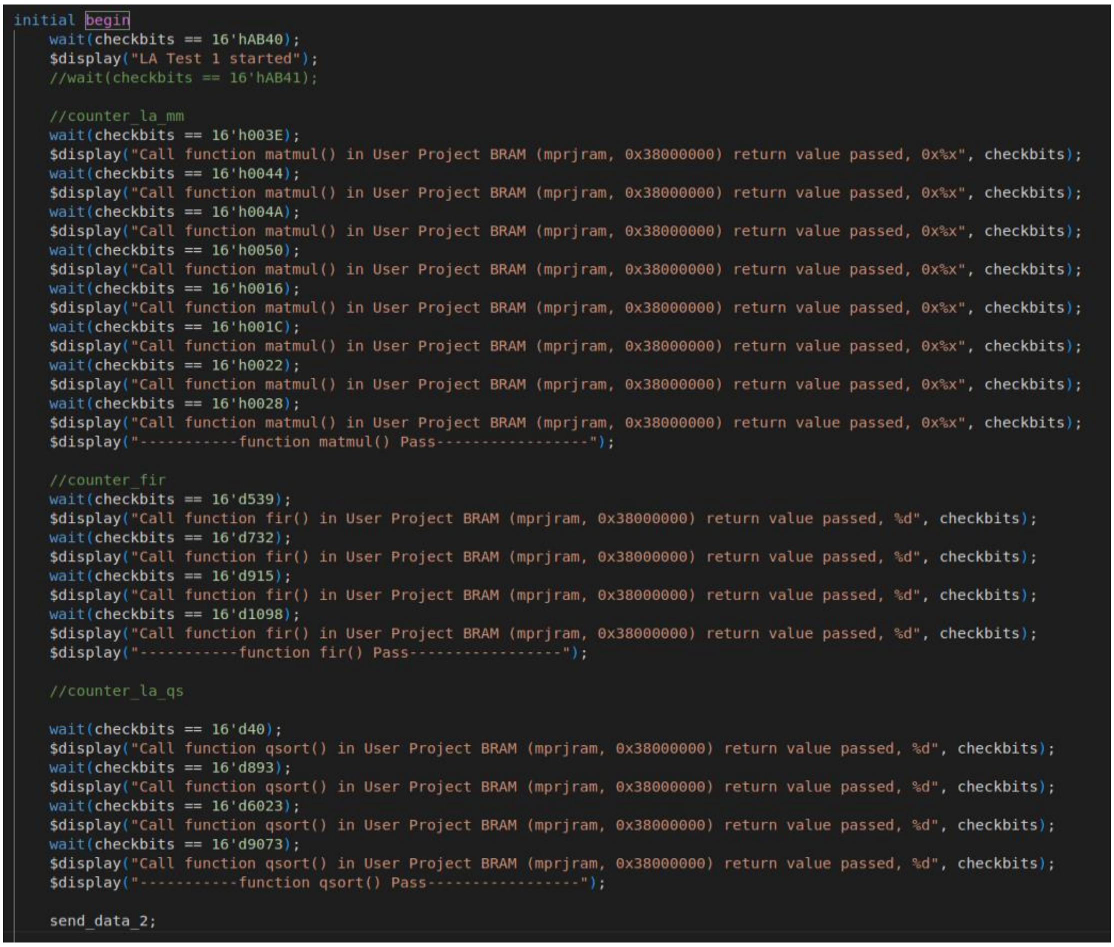
# 合併 Firmware

- 驗證方法:

將先前的驗證方式合併，依序檢查 matmul()、fir()、qsort()、uart 的值

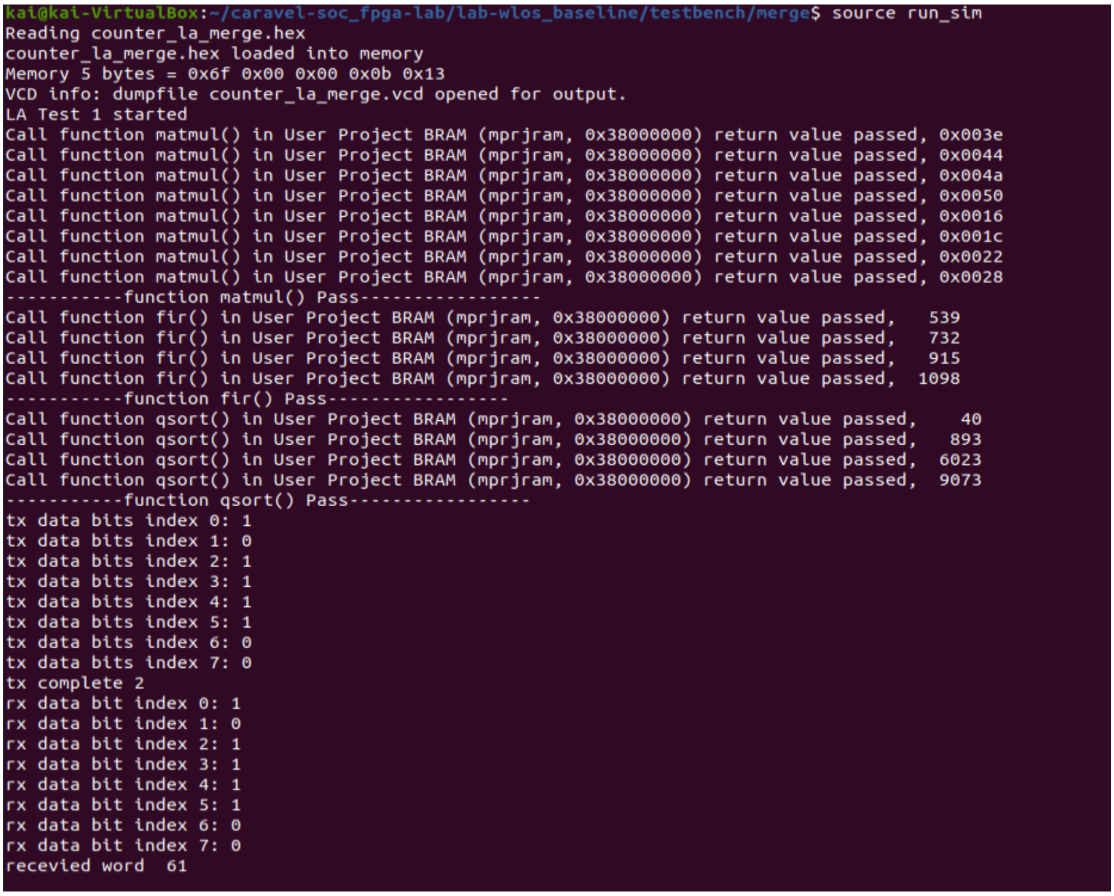


```c
    int* tmp_mm = matmul();
    int* tmp_qs = qsort();
    int* tmp_fir = fir();

    //counter_la_mm
    reg_mprj_datal = *tmp_mm << 16;
    reg_mprj_datal = *(tmp_mm+1) << 16;
    reg_mprj_datal = *(tmp_mm+2) << 16;
    reg_mprj_datal = *(tmp_mm+3) << 16;
    reg_mprj_datal = *(tmp_mm+12) << 16;
    reg_mprj_datal = *(tmp_mm+13) << 16;
    reg_mprj_datal = *(tmp_mm+14) << 16;
    reg_mprj_datal = *(tmp_mm+15) << 16;


    //counter_la_fir
    reg_mprj_datal = *(tmp_fir+7) << 16;
    reg_mprj_datal = *(tmp_fir+8) << 16;
    reg_mprj_datal = *(tmp_fir+9) << 16;
    reg_mprj_datal = *(tmp_fir+10) << 16;

    //counter_la_qs
    reg_mprj_datal = *tmp_qs << 16;
    reg_mprj_datal = *(tmp_qs+1) << 16;
    reg_mprj_datal = *(tmp_qs+8) << 16;
    reg_mprj_datal = *(tmp_qs+9) << 16;
```



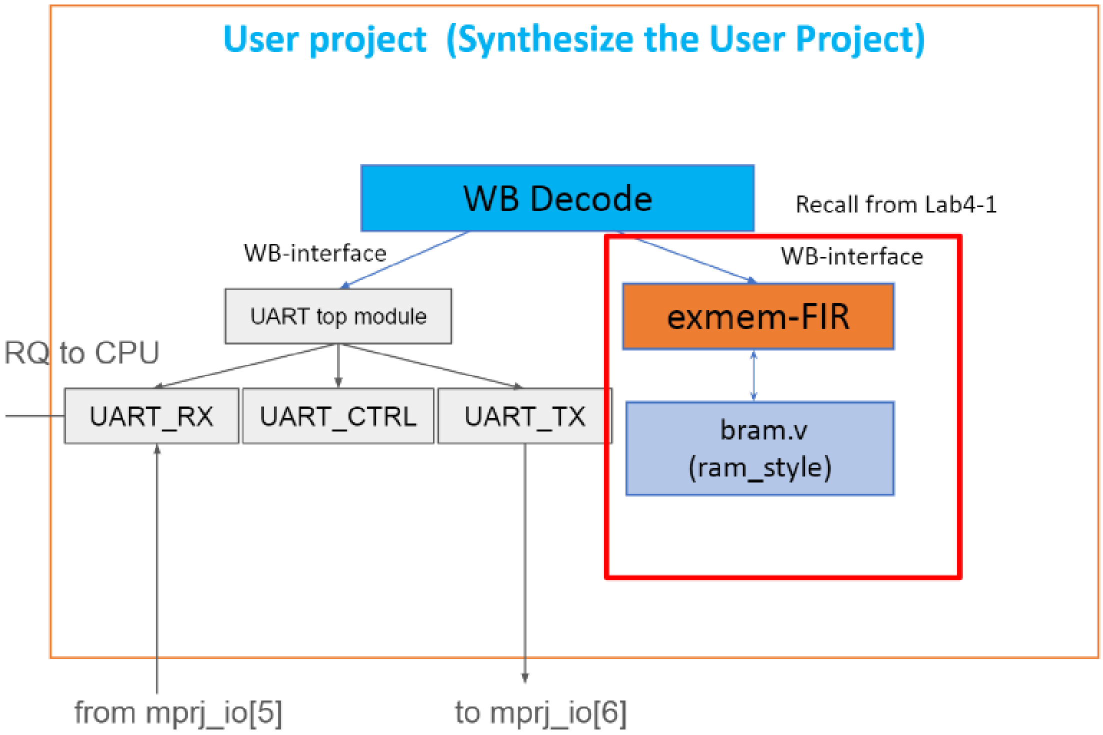
```verilog
initial begin
    wait(checkbits == 16'hAB40);
    $display("LA Test 1 started");
    //wait(checkbits == 16'hAB41);

    //counter_la_mm
    wait(checkbits == 16'h003E);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h0044);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h004A);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h0050);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h0016);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h001C);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h0022);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    wait(checkbits == 16'h0028);
    $display("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x%x", checkbits);
    $display("-----------function matmul() Pass-----------------");

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

    //counter_la_qs

    wait(checkbits == 16'd40);
    $display("Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd893);
    $display("Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd6023);
    $display("Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    wait(checkbits == 16'd9073);
    $display("Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, %d", checkbits);
    $display("-----------function qsort() Pass-----------------");

    send_data_2;
```

模擬結果:

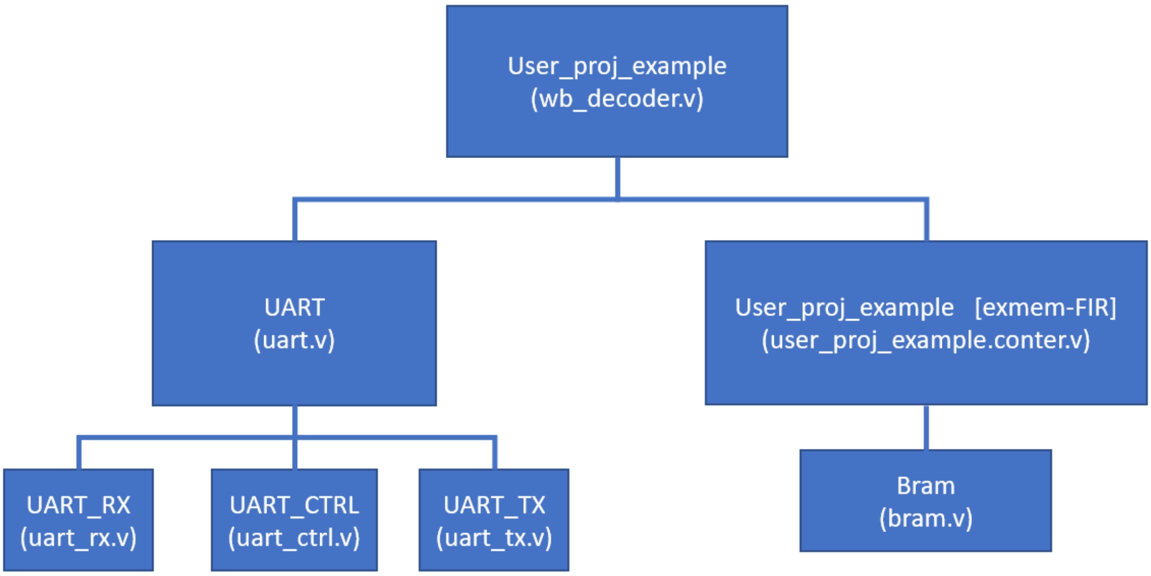


```
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbench/merge$ source run_sim
Reading counter_la_merge.hex
counter_la_merge.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_merge.vcd opened for output.
LA Test 1 started
Call function matmul() 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
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0016
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, 0x0022
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0028
-----------function matmul() Pass-----------------
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed,   539
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed,   732
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed,   915
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed,  1098
-----------function fir() Pass-----------------
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,    40
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,   893
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,  6023
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed,  9073
-----------function qsort() Pass-----------------
tx data bits index 0: 1
tx data bits index 1: 0
tx data bits index 2: 1
tx data bits index 3: 1
tx data bits index 4: 1
tx data bits index 5: 1
tx data bits index 6: 0
tx data bits index 7: 0
tx complete 2
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
recevied word  61
```

## Wishbone decoder



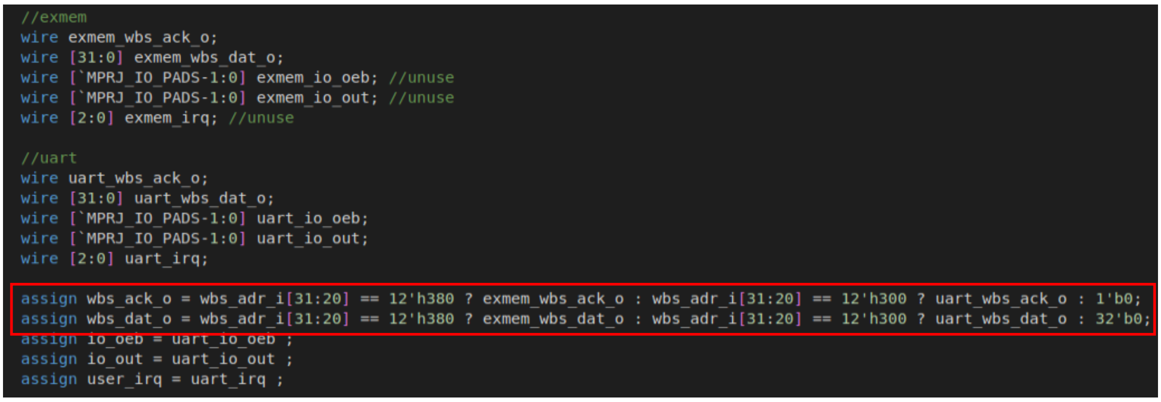
我們實作後的架構如下:



實作方法:
我們修改了 user_proj_example.conter.v 作為 exmem-fir(如 lab4-2)

在 wb_decoder 中我們將 user_proj_wrapper 所有 input wire 接上 uart、user_proj_example 直接對接，只處理 user_proj_wrapper 的 output wire。
此外，user_proj_example 的 io 及 irq 為 unuse 因此我們直接將 uart 的 io 及 irq 輸出至 wrapper。

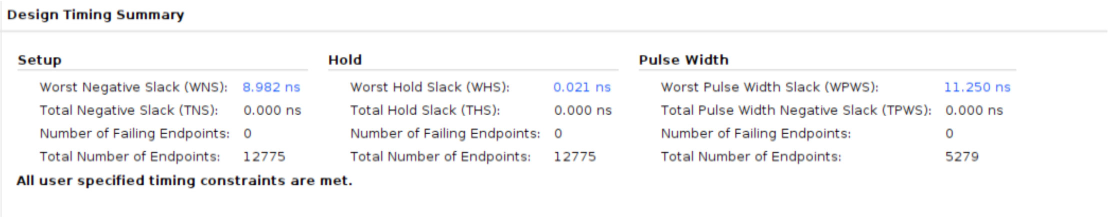
最後根據 address 開頭為 380 或 300 來決定 wbs_ack 與 wbs_dat 是由 uart 或者 user_proj_example 來輸出。



```verilog
//exmem
wire exmem_wbs_ack_o;
wire [31:0] exmem_wbs_dat_o;
wire [`MPRJ_IO_PADS-1:0] exmem_io_oeb; //unuse
wire [`MPRJ_IO_PADS-1:0] exmem_io_out; //unuse
wire [2:0] exmem_irq; //unuse

//uart
wire uart_wbs_ack_o;
wire [31:0] uart_wbs_dat_o;
wire [`MPRJ_IO_PADS-1:0] uart_io_oeb;
wire [`MPRJ_IO_PADS-1:0] uart_io_out;
wire [2:0] uart_irq;

assign wbs_ack_o = wbs_adr_i[31:20] == 12'h380 ? exmem_wbs_ack_o : wbs_adr_i[31:20] == 12'h300 ? uart_wbs_ack_o : 1'b0;
assign wbs_dat_o = wbs_adr_i[31:20] == 12'h380 ? exmem_wbs_dat_o : wbs_adr_i[31:20] == 12'h300 ? uart_wbs_dat_o : 32'b0;
assign io_oeb = uart_io_oeb ;
assign io_out = uart_io_out ;
assign user_irq = uart_irq ;
```

# Run Merged Firmware on FPGA

- Time report

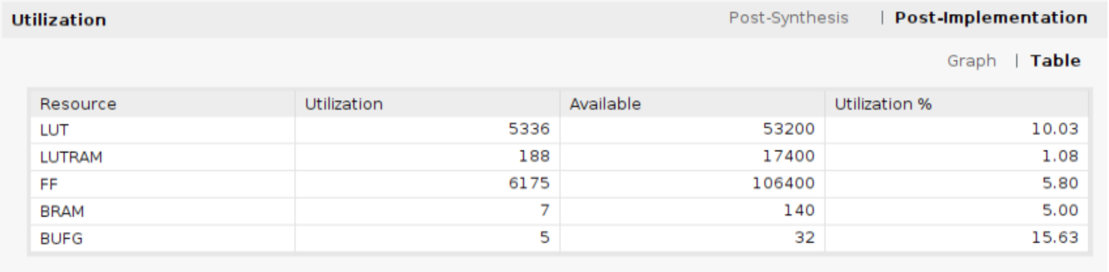


**Design Timing Summary**

| Setup | | Hold | | Pulse Width | |
|---|---|---|---|---|---|
| Worst Negative Slack (WNS): | 8.982 ns | Worst Hold Slack (WHS): | 0.021 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: | 12775 | Total Number of Endpoints: | 12775 | Total Number of Endpoints: | 5279 |

**All user specified timing constraints are met.**

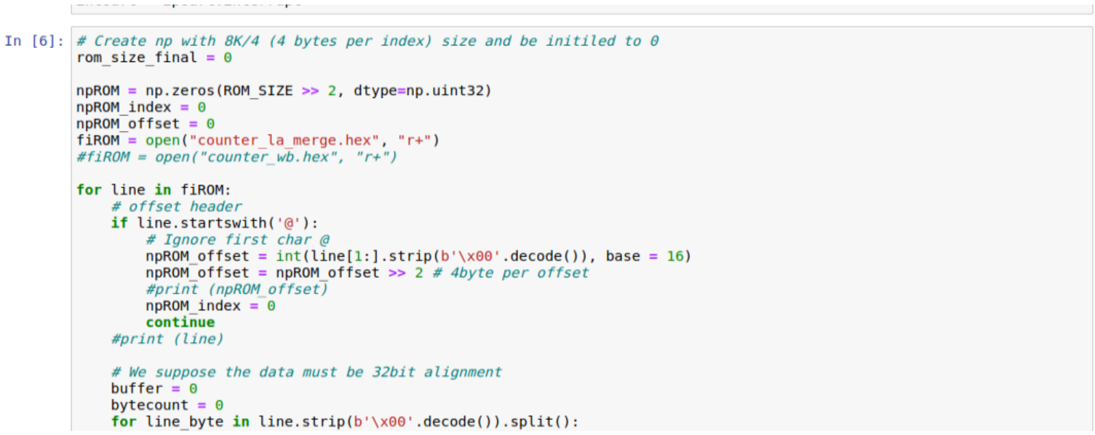
- Resource report



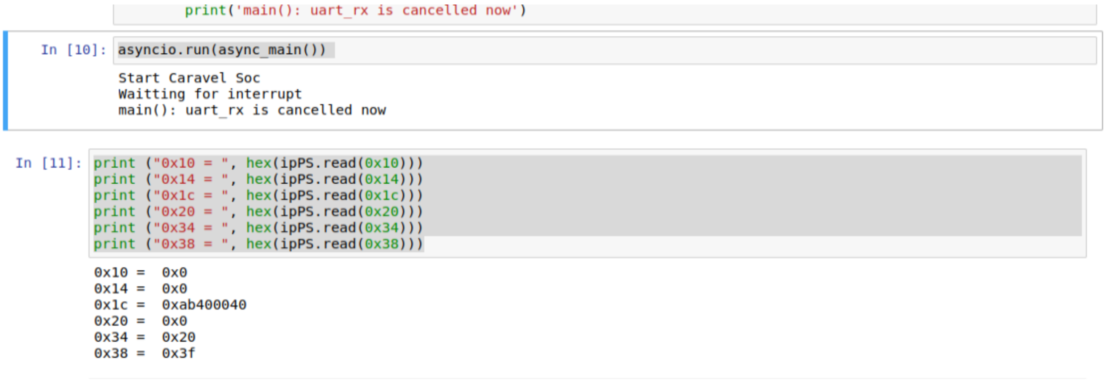
**Utilization**

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 5336 | 53200 | 10.03 |
| LUTRAM | 188 | 17400 | 1.08 |
| FF | 6175 | 106400 | 5.80 |
| BRAM | 7 | 140 | 5.00 |
| BUFG | 5 | 32 | 15.63 |

- Test result



```
In [6]: # Create np with 8K/4 (4 bytes per index) size and be initiled to 0
        rom_size_final = 0

        npROM = np.zeros(ROM_SIZE >> 2, dtype=np.uint32)
        npROM_index = 0
        npROM_offset = 0
        fiROM = open("counter_la_merge.hex", "r+")
        #fiROM = open("counter_wb.hex", "r+")

        for line in fiROM:
            # offset header
            if line.startswith('@'):
                # Ignore first char @
                npROM_offset = int(line[1:].strip(b'\x00'.decode()), base = 16)
                npROM_offset = npROM_offset >> 2 # 4byte per offset
                #print (npROM_offset)
                npROM_index = 0
                continue
            #print (line)

            # We suppose the data must be 32bit alignment
            buffer = 0
            bytecount = 0
            for line_byte in line.strip(b'\x00'.decode()).split():
```

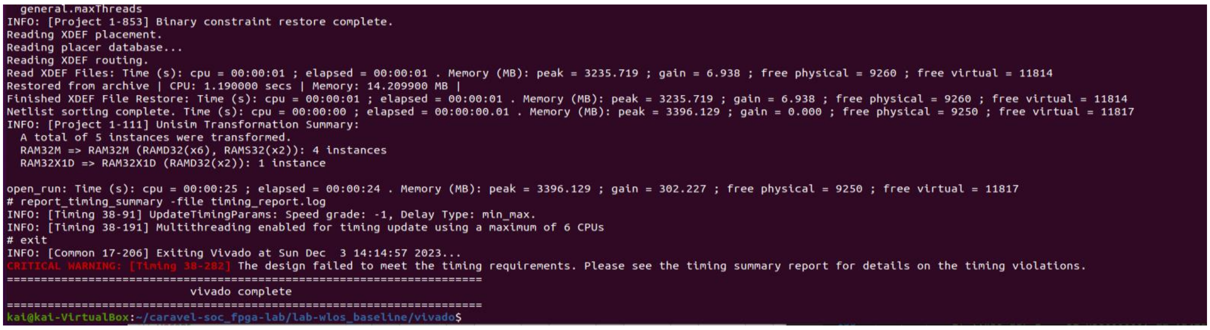


```
                print('main(): uart_rx is cancelled now')

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

         Start Caravel Soc
         Waitting for interrupt
         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 =  0xab400040
         0x20 =  0x0
         0x34 =  0x20
         0x38 =  0x3f
```

- Observe

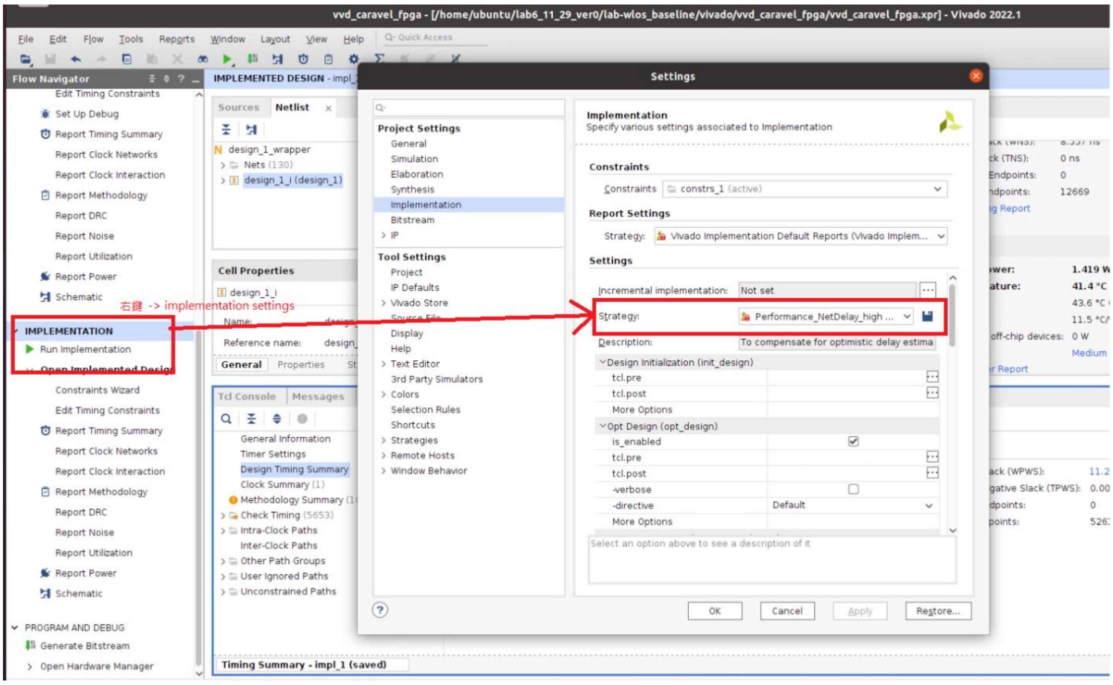
Time report 問題:

我們在模擬測試過結合後的各功能正常，但跑合成時出現 fail to meet the timing requirement 的問題。



```
  general.maxThreads
INFO: [Project 1-853] Binary constraint restore complete.
Reading XDEF placement.
Reading placer database...
Reading XDEF routing.
Read XDEF Files: Time (s): cpu = 00:00:01 ; elapsed = 00:00:01 . Memory (MB): peak = 3235.719 ; gain = 6.938 ; free physical = 9260 ; free virtual = 11814
Restored from archive | CPU: 1.190000 secs | Memory: 14.209900 MB |
Finished XDEF File Restore: Time (s): cpu = 00:00:01 ; elapsed = 00:00:01 . Memory (MB): peak = 3235.719 ; gain = 6.938 ; free physical = 9260 ; free virtual = 11814
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.01 . Memory (MB): peak = 3396.129 ; gain = 0.000 ; free physical = 9250 ; free virtual = 11817
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 = 00:00:25 ; elapsed = 00:00:24 . Memory (MB): peak = 3396.129 ; gain = 302.227 ; free physical = 9250 ; free virtual = 11817
# 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 6 CPUs
# exit
INFO: [Common 17-206] Exiting Vivado at Sun Dec  3 14:14:57 2023...
CRITICAL WARNING: [Timing 38-282] The design failed to meet the timing requirements. Please see the timing summary report for details on the timing violations.
======================================================================
                           vivado complete
======================================================================
kai@kai-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/vivado$
```

後來參考 github 討論群同學的回答，將進 vivado gui 改變 implementation 的 strategy，將 default 改成使用 performance_NetDelay_high 後有成功修復我們的 hold time violation。



vvd_caravel_fpga - [/home/ubuntu/lab6_11_29_ver0/lab-wlos_baseline/vivado/vvd_caravel_fpga/vvd_caravel_fpga.xpr] - Vivado 2022.1

右鍵 -> implementation settings

Settings

Implementation
Specify various settings associated to Implementation

Constraints: constrs_1 (active)

Report Settings — Strategy: Vivado Implementation Default Reports (Vivado Implem...

Settings

Incremental implementation: Not set

Strategy: Performance_NetDelay_high ...

Description: To compensate for optimistic delay estima

Design Initialization (init_design)

Opt Design (opt_design)

is_enabled ☑

-verbose □

-directive: Default

Select an option above to see a description of it