系統晶片設計
SOC Design

LAB6

組員:郭銘宸、郭紘碩

學號:311511082、3115120656

系所:電機所碩二、電控所碩二

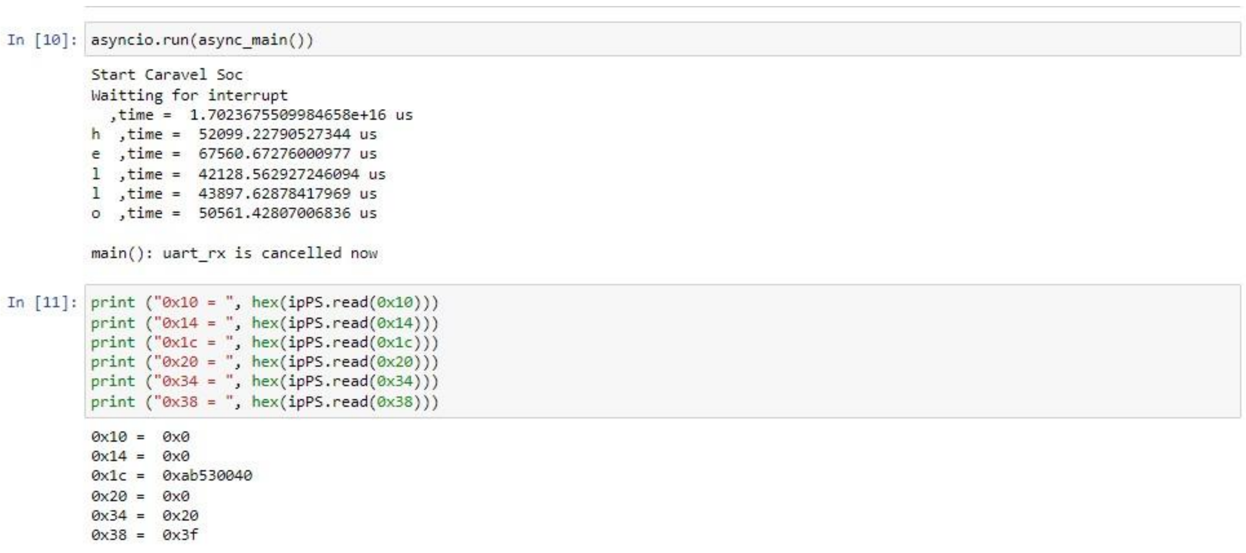
一、 How do you verify your answer from notebook

- 方法一：降低 firmware 運行速度觀察 mprj

　　在本次實驗中需要驗證 4 個功能合一後的 firmware 正確性，而在一般合成的 firmware 中由於運行速度太快，因此需要使用 counter 來減慢每個步驟的間格以方便觀測 mprj data 的變化，下圖(一)是 jupyter notebook 的運行結果，而 firmware 的運行方式如下圖(二)。

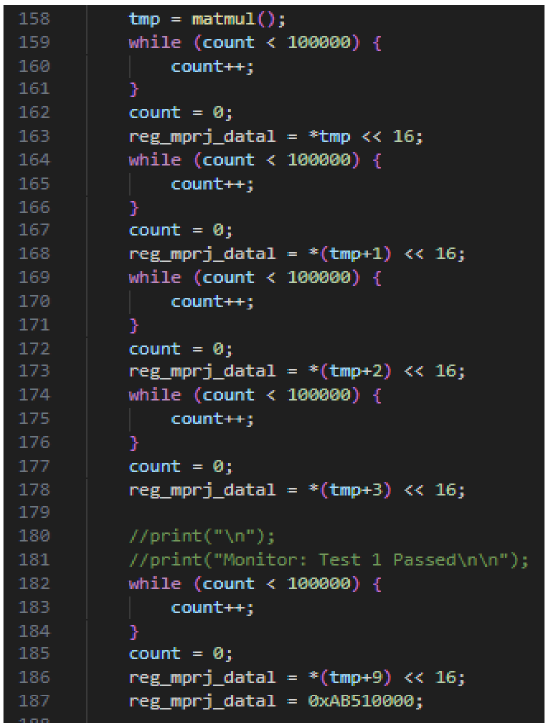
　　Firmware 中利用 counter 計數到 10000，再將下個計算結果放入 mprj，如此一來便能藉由加入圖(三)中所示在 jupyter notebook 的 display 方式，在 notebook 中看到運行中 mprj data 變化被印出，而圖(四)中是在 notebook 中所顯示被印出的 mprj data 的部分變化。

　　在我們設定的 firmware 中，checkbit 為 ab51 代表著 matmul 的結束訊號，也就表示功能可以正常運作，而 ab52、ab53 分別代表著 qsort 和 fir 的結束，這些都能在 mprj 變化被印出所觀察到，在運行結果圖(四)的最後，也能看到 isr 被呼叫並執行，並且利用 uart 逐字印出 hello 的每個 char，並且記錄開始時間與每個 char 的間格時間。



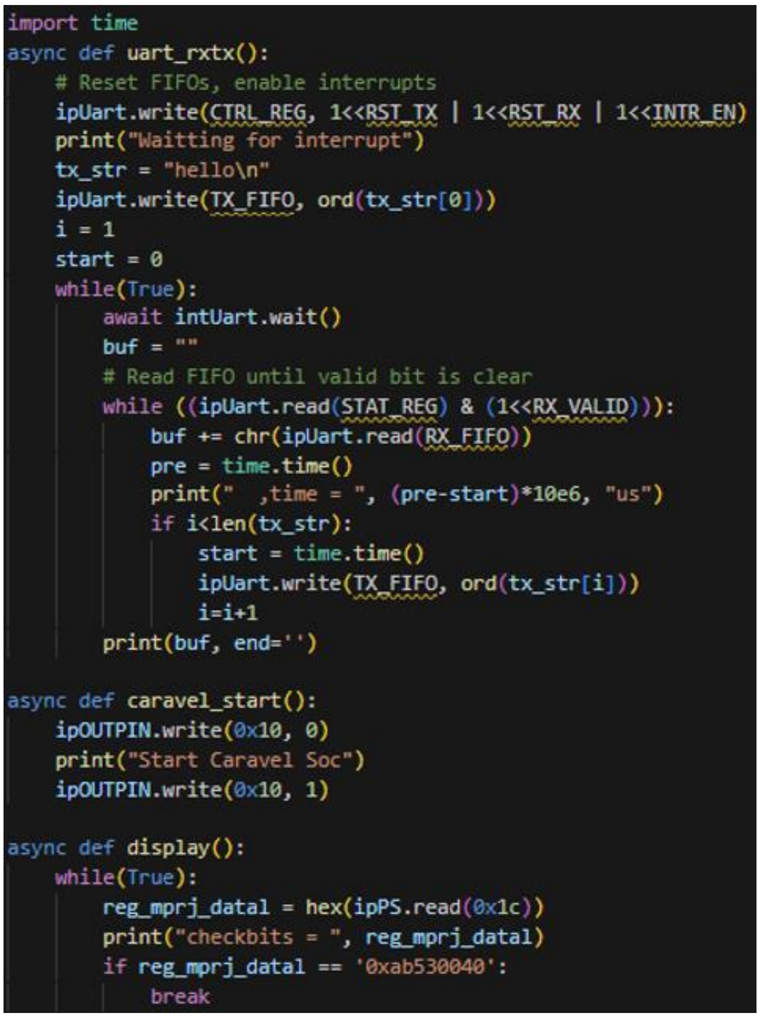
```
In [10]: asyncio.run(async_main())

Start Caravel Soc
Waitting for interrupt
  ,time =  1.7023675509984658e+16 us
h  ,time =  52099.22790527344 us
e  ,time =  67560.67276000977 us
l  ,time =  42128.562927246094 us
l  ,time =  43897.62878417969 us
o  ,time =  50561.42807006836 us

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 =  0xab530040
0x20 =  0x0
0x34 =  0x20
0x38 =  0x3f
```

圖(一)



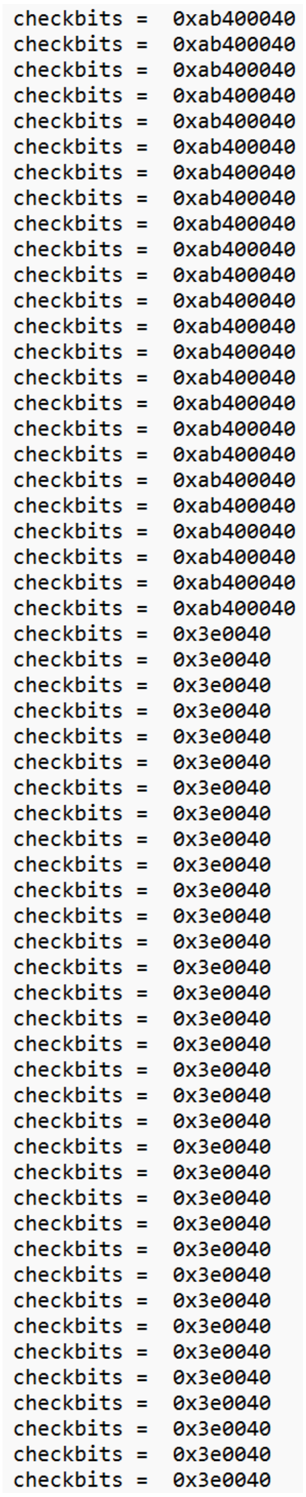
```c
    tmp = matmul();
    while (count < 100000) {
        count++;
    }
    count = 0;
    reg_mprj_datal = *tmp << 16;
    while (count < 100000) {
        count++;
    }
    count = 0;
    reg_mprj_datal = *(tmp+1) << 16;
    while (count < 100000) {
        count++;
    }
    count = 0;
    reg_mprj_datal = *(tmp+2) << 16;
    while (count < 100000) {
        count++;
    }
    count = 0;
    reg_mprj_datal = *(tmp+3) << 16;

    //print("\n");
    //print("Monitor: Test 1 Passed\n\n");
    while (count < 100000) {
        count++;
    }
    count = 0;
    reg_mprj_datal = *(tmp+9) << 16;
    reg_mprj_datal = 0xAB510000;
```

圖（二）

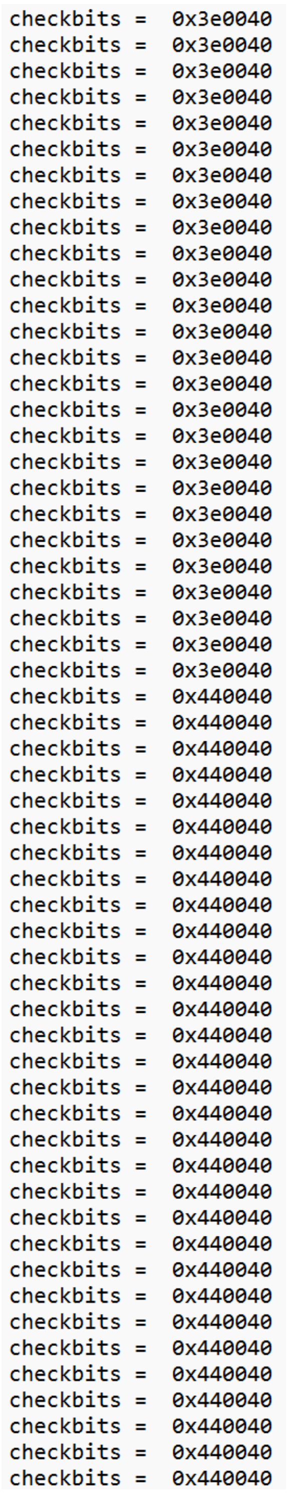


```python
import time
async def uart_rxtx():
    # Reset FIFOs, enable interrupts
    ipUart.write(CTRL_REG, 1<<RST_TX | 1<<RST_RX | 1<<INTR_EN)
    print("Waitting for interrupt")
    tx_str = "hello\n"
    ipUart.write(TX_FIFO, ord(tx_str[0]))
    i = 1
    start = 0
    while(True):
        await intUart.wait()
        buf = ""
        # Read FIFO until valid bit is clear
        while ((ipUart.read(STAT_REG) & (1<<RX_VALID))):
            buf += chr(ipUart.read(RX_FIFO))
            pre = time.time()
            print("   ,time = ", (pre-start)*10e6, "us")
            if i<len(tx_str):
                start = time.time()
                ipUart.write(TX_FIFO, ord(tx_str[i]))
                i=i+1
        print(buf, end='')

async def caravel_start():
    ipOUTPIN.write(0x10, 0)
    print("Start Caravel Soc")
    ipOUTPIN.write(0x10, 1)

async def display():
    while(True):
        reg_mprj_datal = hex(ipPS.read(0x1c))
        print("checkbits = ", reg_mprj_datal)
        if reg_mprj_datal == '0xab530040':
            break
```

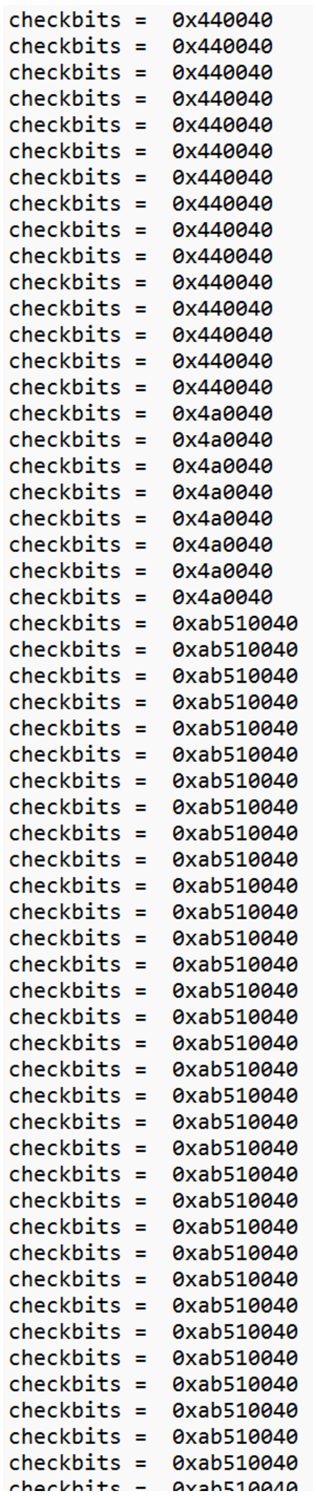
圖（三）



```
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0xab400040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
```



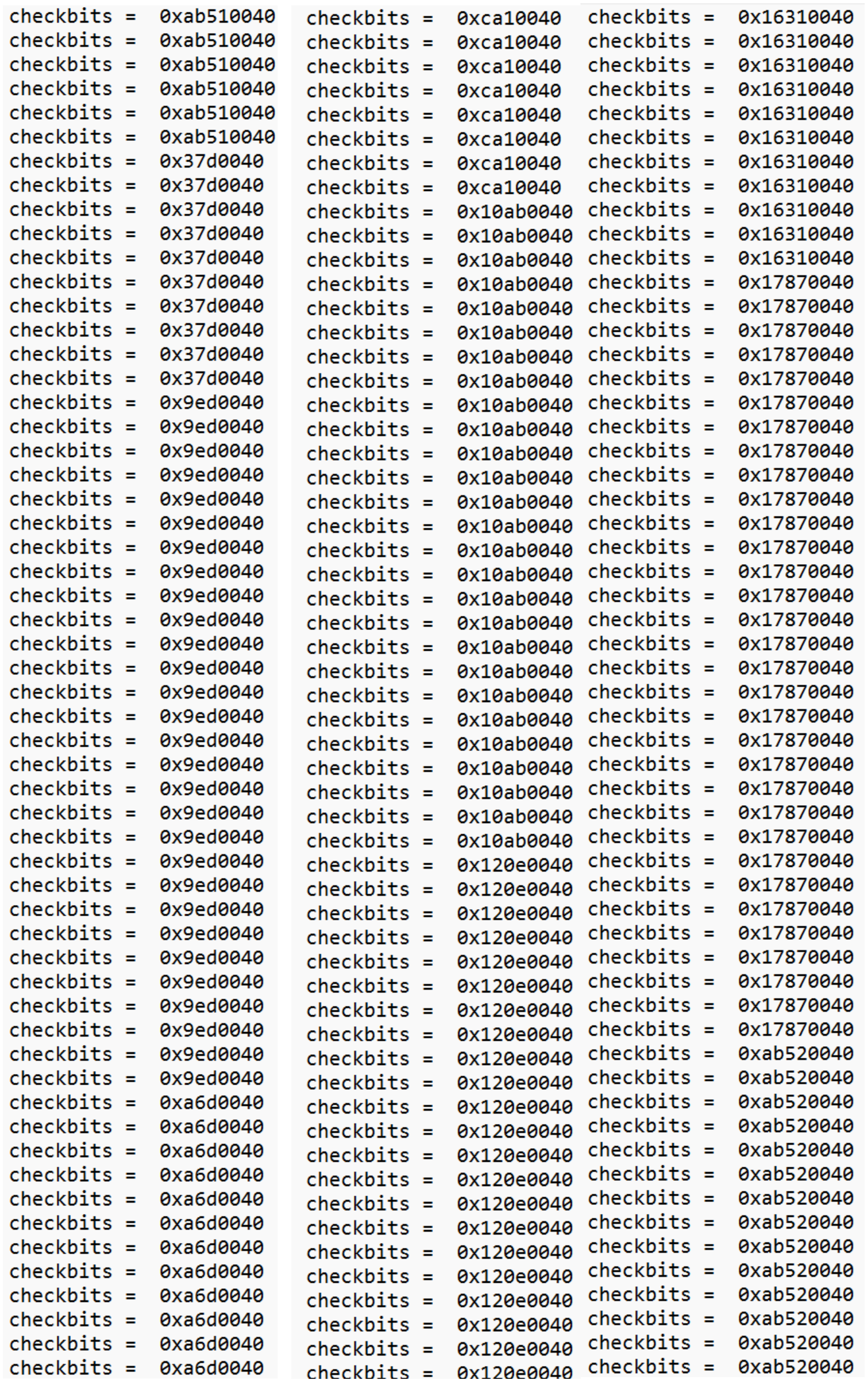
```
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x3e0040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
```



```
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x440040
checkbits =  0x4a0040
checkbits =  0x4a0040
checkbits =  0x4a0040
checkbits =  0x4a0040
checkbits =  0x4a0040
checkbits =  0x4a0040
checkbits =  0x4a0040
checkbits =  0x4a0040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
checkbits =  0xab510040
```

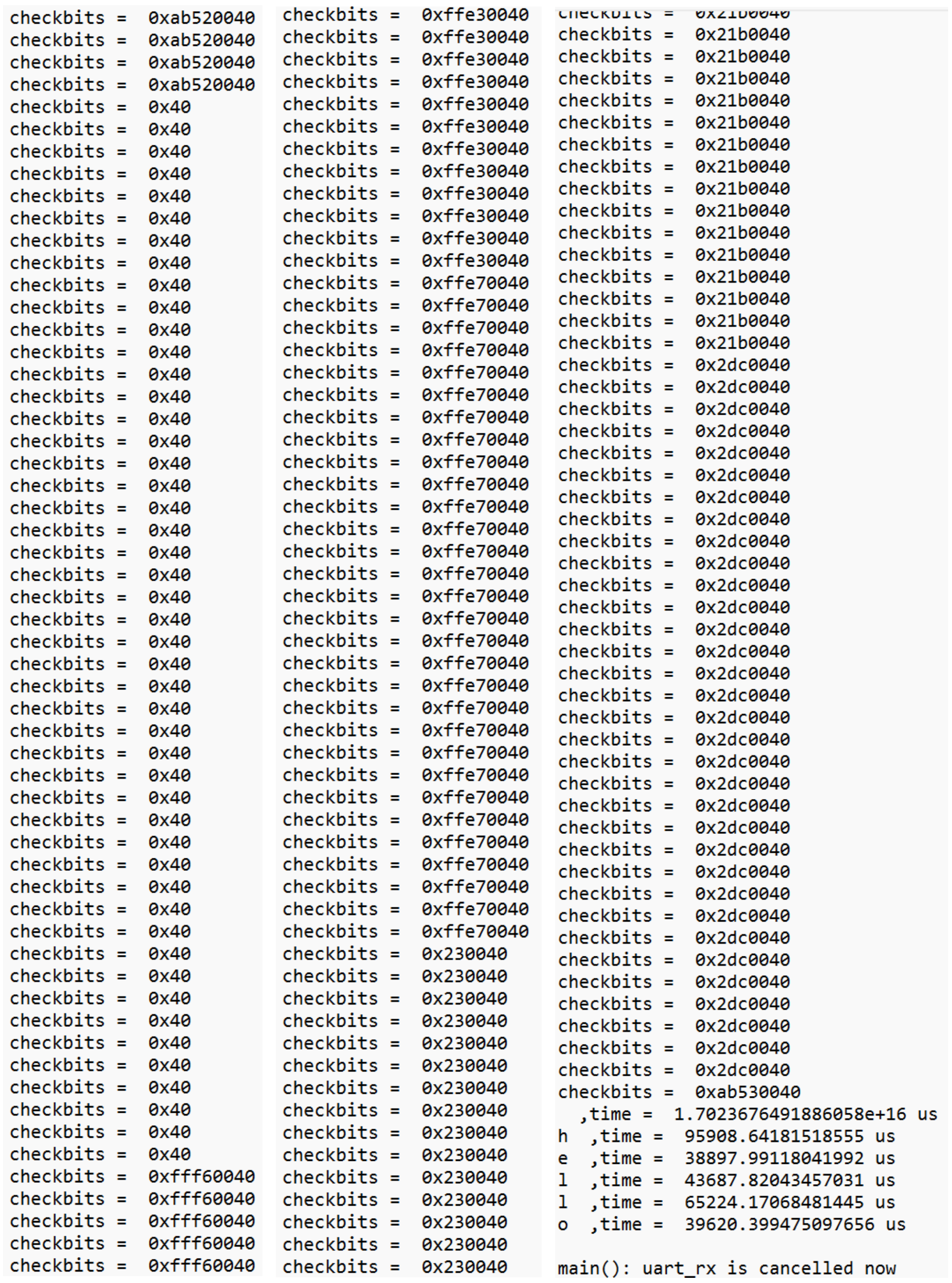
圖（四）

圖（四）



```
checkbits =  0xab510040   checkbits =  0xca10040    checkbits =  0x16310040
checkbits =  0xab510040   checkbits =  0xca10040    checkbits =  0x16310040
checkbits =  0xab510040   checkbits =  0xca10040    checkbits =  0x16310040
checkbits =  0xab510040   checkbits =  0xca10040    checkbits =  0x16310040
checkbits =  0xab510040   checkbits =  0xca10040    checkbits =  0x16310040
checkbits =  0xab510040   checkbits =  0xca10040    checkbits =  0x16310040
checkbits =  0x37d0040    checkbits =  0xca10040    checkbits =  0x16310040
checkbits =  0x37d0040    checkbits =  0xca10040    checkbits =  0x16310040
checkbits =  0x37d0040    checkbits =  0x10ab0040   checkbits =  0x16310040
checkbits =  0x37d0040    checkbits =  0x10ab0040   checkbits =  0x16310040
checkbits =  0x37d0040    checkbits =  0x10ab0040   checkbits =  0x16310040
checkbits =  0x37d0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x37d0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x37d0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x37d0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x37d0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x10ab0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0x17870040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0x9ed0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
checkbits =  0xa6d0040    checkbits =  0x120e0040   checkbits =  0xab520040
```

圖（四）

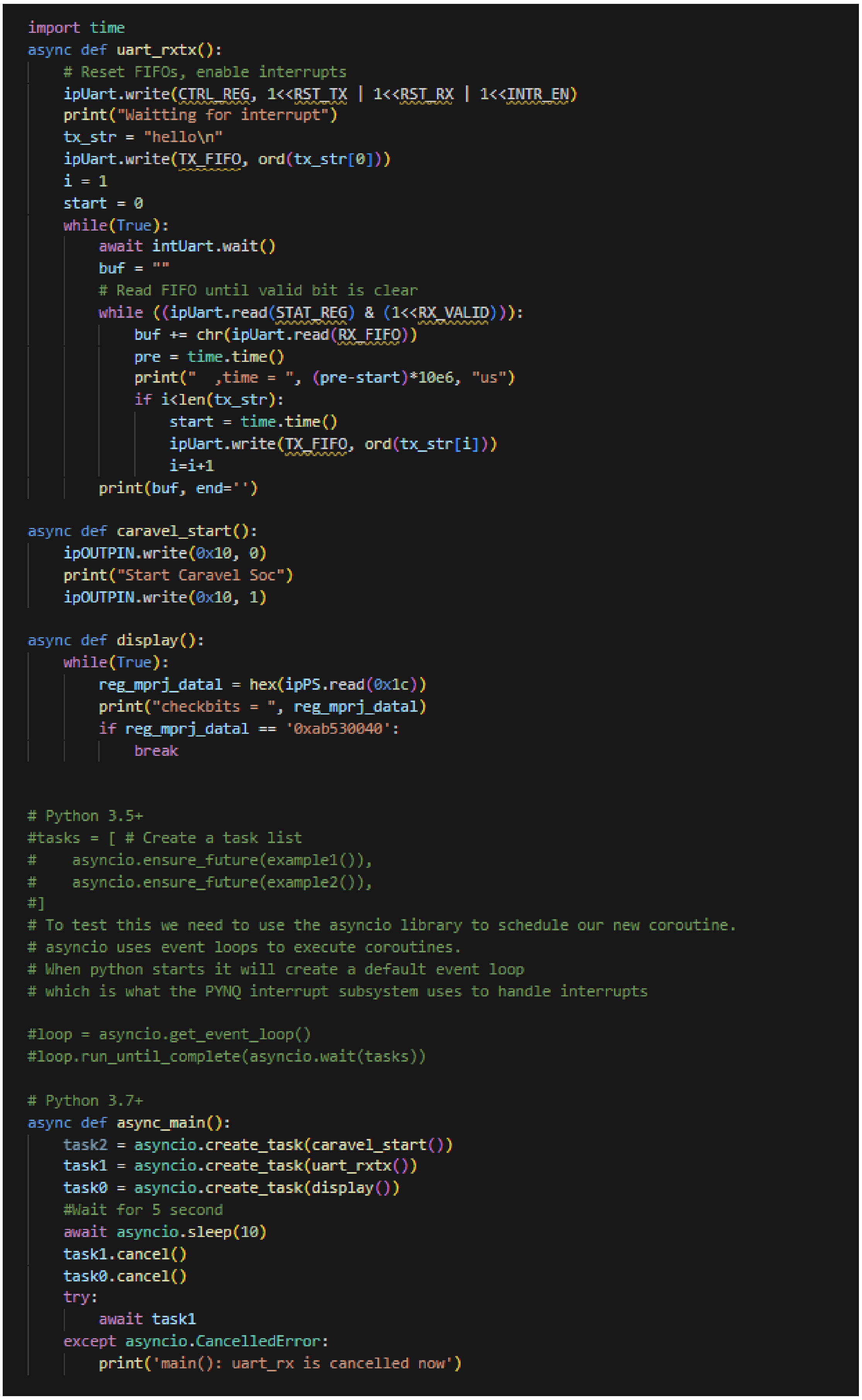


```
checkbits =  0xab520040
checkbits =  0xab520040
checkbits =  0xab520040
checkbits =  0xab520040
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0x40
checkbits =  0xfff60040
checkbits =  0xfff60040
checkbits =  0xfff60040
checkbits =  0xfff60040
checkbits =  0xfff60040
```

```
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe30040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0xffe70040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
checkbits =  0x230040
```

```
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x21b0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0xab530040
  ,time =  1.7023676491886058e+16 us
h  ,time =  95908.64181518555 us
e  ,time =  38897.99118041992 us
l  ,time =  43687.82043457031 us
l  ,time =  65224.17068481445 us
o  ,time =  39620.399475097656 us

main(): uart_rx is cancelled now
```

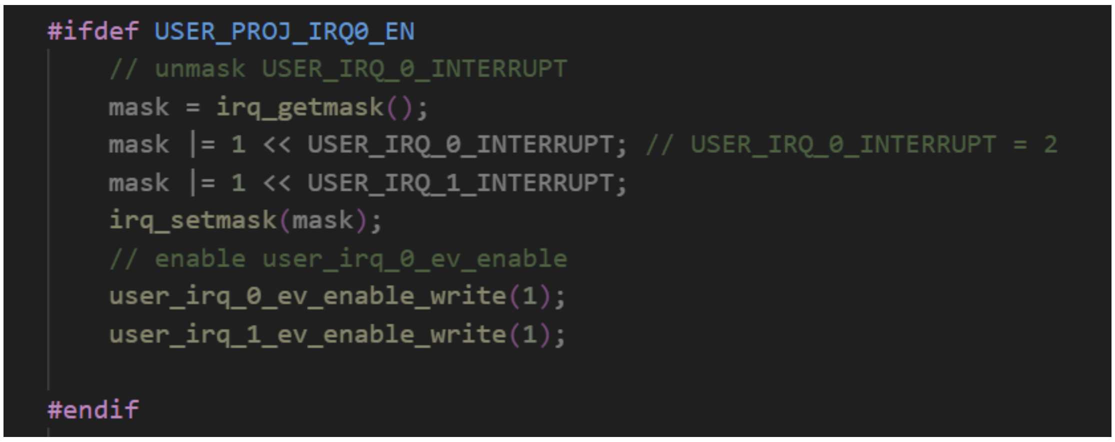
圖(四)



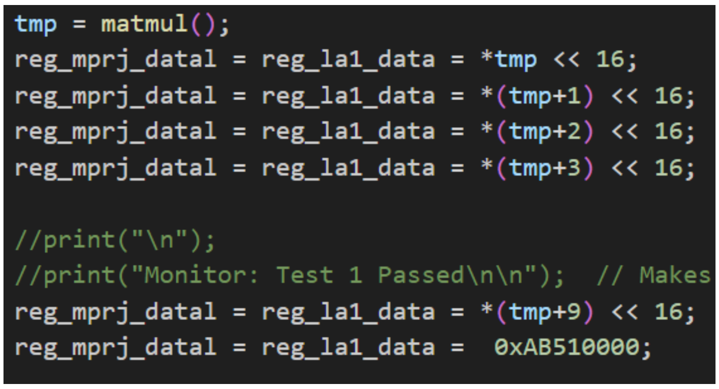
```python
import time
async def uart_rxtx():
    # Reset FIFOs, enable interrupts
    ipUart.write(CTRL_REG, 1<<RST_TX | 1<<RST_RX | 1<<INTR_EN)
    print("Waitting for interrupt")
    tx_str = "hello\n"
    ipUart.write(TX_FIFO, ord(tx_str[0]))
    i = 1
    start = 0
    while(True):
        await intUart.wait()
        buf = ""
        # Read FIFO until valid bit is clear
        while ((ipUart.read(STAT_REG) & (1<<RX_VALID))):
            buf += chr(ipUart.read(RX_FIFO))
            pre = time.time()
            print("   ,time = ", (pre-start)*10e6, "us")
            if i<len(tx_str):
                start = time.time()
                ipUart.write(TX_FIFO, ord(tx_str[i]))
                i=i+1
        print(buf, end='')

async def caravel_start():
    ipOUTPIN.write(0x10, 0)
    print("Start Caravel Soc")
    ipOUTPIN.write(0x10, 1)

async def display():
    while(True):
        reg_mprj_datal = hex(ipPS.read(0x1c))
        print("checkbits = ", reg_mprj_datal)
        if reg_mprj_datal == '0xab530040':
            break


# Python 3.5+
#tasks = [ # Create a task list
#    asyncio.ensure_future(example1()),
#    asyncio.ensure_future(example2()),
#]
# To test this we need to use the asyncio library to schedule our new coroutine.
# asyncio uses event loops to execute coroutines.
# When python starts it will create a default event loop
# which is what the PYNQ interrupt subsystem uses to handle interrupts

#loop = asyncio.get_event_loop()
#loop.run_until_complete(asyncio.wait(tasks))

# Python 3.7+
async def async_main():
    task2 = asyncio.create_task(caravel_start())
    task1 = asyncio.create_task(uart_rxtx())
    task0 = asyncio.create_task(display())
    #Wait for 5 second
    await asyncio.sleep(10)
    task1.cancel()
    task0.cancel()
    try:
        await task1
    except asyncio.CancelledError:
        print('main(): uart_rx is cancelled now')
```

➢ 方法二：利用 ISR 的 uart 傳輸功能印出 checkbit 驗證

在此驗證方法中，會利用到改變 isr 的功能，使其 uart 可以回傳 checkbit 做中斷時的驗證，而方法首先是如圖(五)所示，再加入一組 irq_1，形成兩組 uart 分別傳送 checkbit 的 upper 8 bit 與 lower 8 bit，因此需要在 firmware 裡額外加上 irq_1_enable，並且將其加入 mask 定義，而在每次呼叫函式時，也都會額外將結果放到 reg_la1_data 上，方便後面使用 uart 傳輸時的資料拿取。

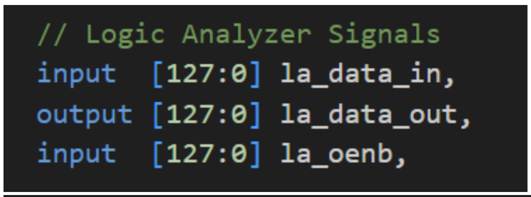


```
#ifdef USER_PROJ_IRQ0_EN
    // unmask USER_IRQ_0_INTERRUPT
    mask = irq_getmask();
    mask |= 1 << USER_IRQ_0_INTERRUPT; // USER_IRQ_0_INTERRUPT = 2
    mask |= 1 << USER_IRQ_1_INTERRUPT;
    irq_setmask(mask);
    // enable user_irq_0_ev_enable
    user_irq_0_ev_enable_write(1);
    user_irq_1_ev_enable_write(1);

#endif
```

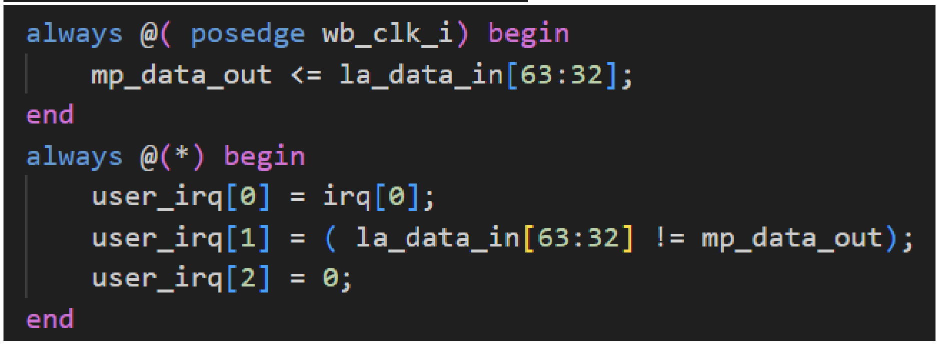


```
tmp = matmul();
reg_mprj_datal = reg_la1_data = *tmp << 16;
reg_mprj_datal = reg_la1_data = *(tmp+1) << 16;
reg_mprj_datal = reg_la1_data = *(tmp+2) << 16;
reg_mprj_datal = reg_la1_data = *(tmp+3) << 16;

//print("\n");
//print("Monitor: Test 1 Passed\n\n");  // Makes
reg_mprj_datal = reg_la1_data = *(tmp+9) << 16;
reg_mprj_datal = reg_la1_data =  0xAB510000;
```
圖(五) Firmware code

有了兩組 interrupt quest 後，再來就是設定 irq 觸發的條件，在圖(六)中使用到的是 la 的 63~32 位元來完成 uart 的輸入輸出，而觸發 irq 的條件便是每當 la_data_in 的值出現變更，就會開啟中斷請求，來讓 uart 印出當前 checkbit 的數值。



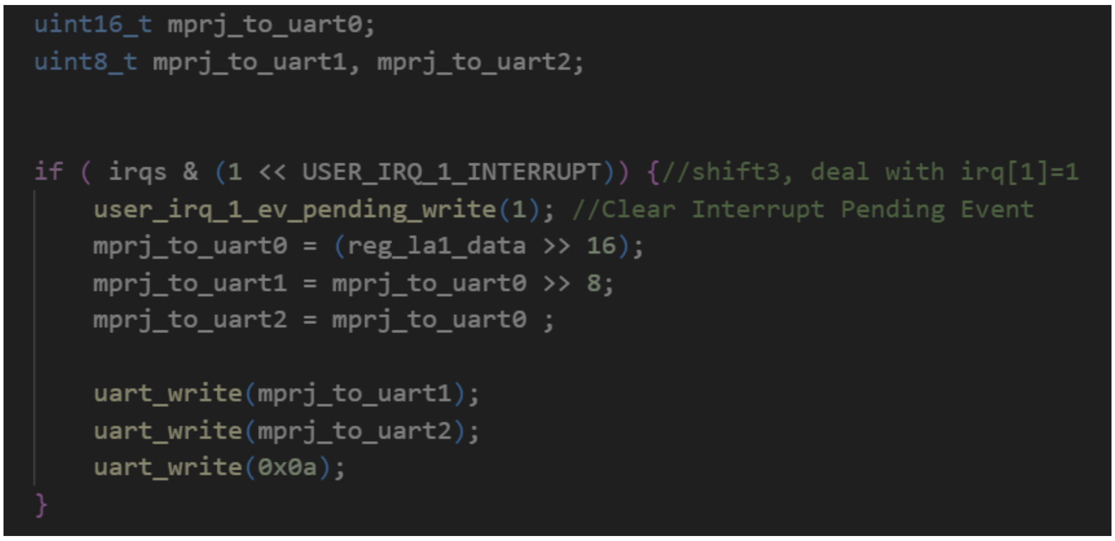
```
// Logic Analyzer Signals
input  [127:0] la_data_in,
output [127:0] la_data_out,
input  [127:0] la_oenb,
```



```
always @( posedge wb_clk_i) begin
    mp_data_out <= la_data_in[63:32];
end
always @(*) begin
    user_irq[0] = irq[0];
    user_irq[1] = ( la_data_in[63:32] != mp_data_out);
    user_irq[2] = 0;
end
```

圖(六) User_project_wrapper.v

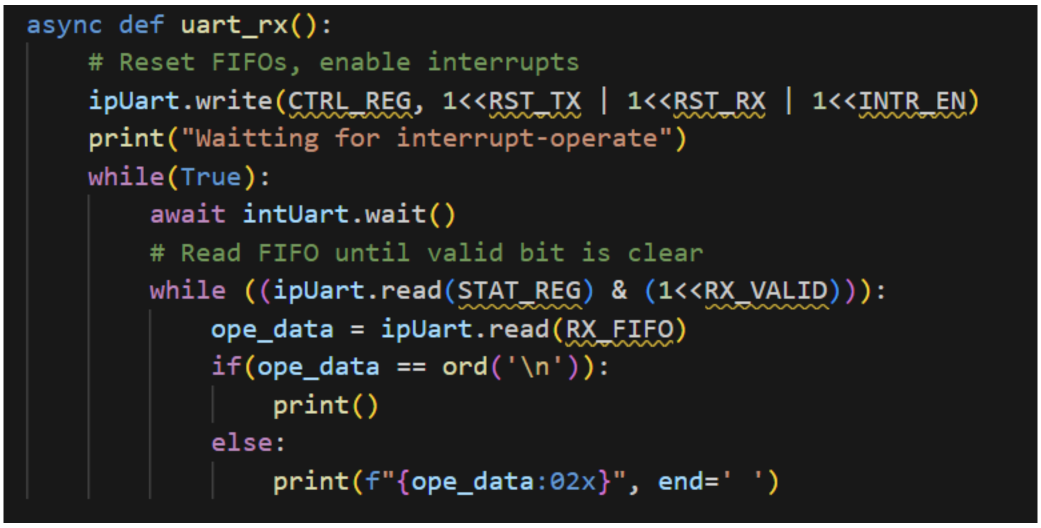
在 isr 內部也需要如圖(七)增加兩組 uart 來完成 checkbit 前後 8bit 的傳輸，其中會將 la 上的資料位移 16bit，之後再將前後 8bit 分配給兩組 uart 進行傳輸以印出完整的 checkbit，由於在 isr 內部加入新的工作理論上會使得中斷效率降低，因此在後面的 uart loop back latency 中會將回傳速度進行比較與討論。



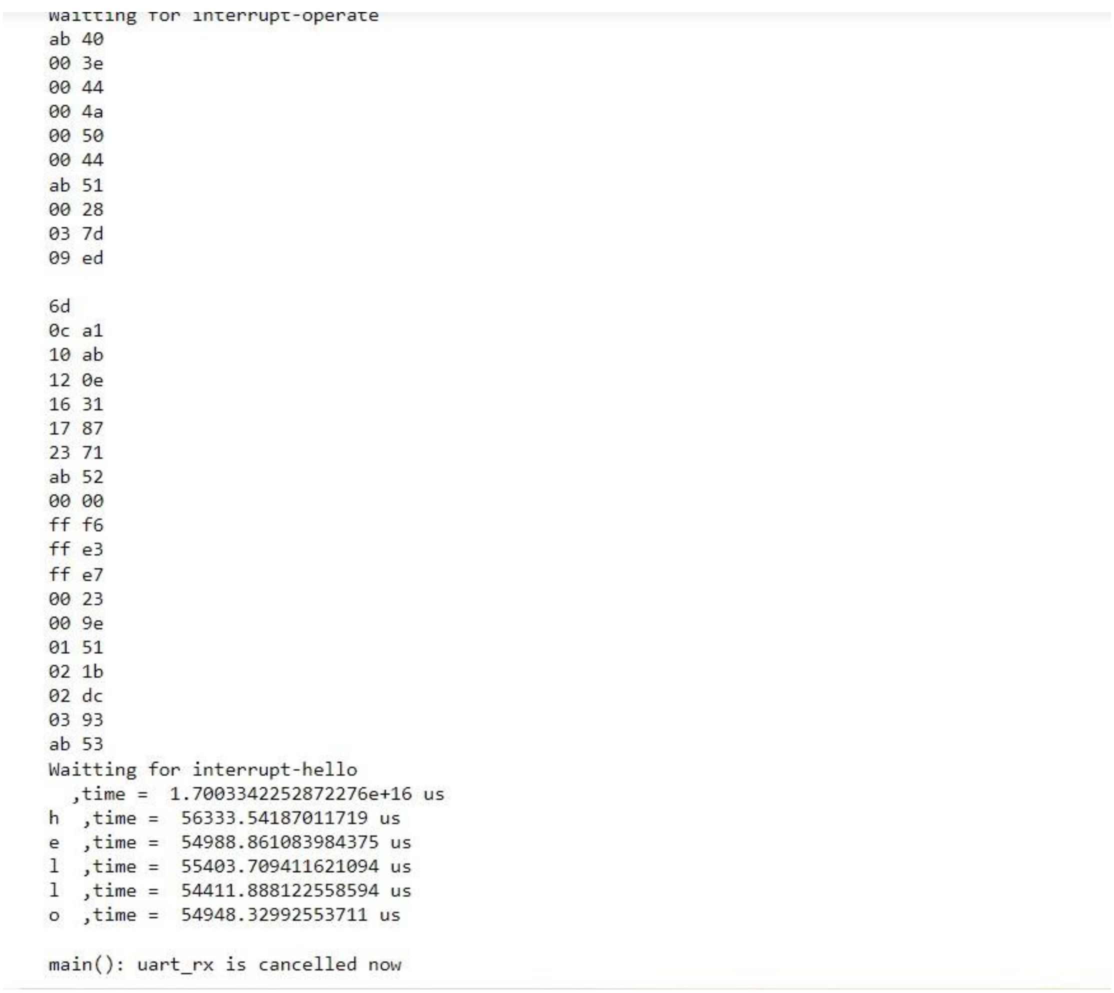
```
uint16_t mprj_to_uart0;
uint8_t mprj_to_uart1, mprj_to_uart2;


if ( irqs & (1 << USER_IRQ_1_INTERRUPT)) {//shift3, deal with irq[1]=1
    user_irq_1_ev_pending_write(1); //Clear Interrupt Pending Event
    mprj_to_uart0 = (reg_la1_data >> 16);
    mprj_to_uart1 = mprj_to_uart0 >> 8;
    mprj_to_uart2 = mprj_to_uart0 ;

    uart_write(mprj_to_uart1);
    uart_write(mprj_to_uart2);
    uart_write(0x0a);
}
```

圖(七) isr.c

圖(八)為 jupyter notebook 上的執行結果，能發現上半部有一對一對的資料被印出，也就是兩組 uart 分別傳送前後的 checkbit 到 ps side 被接收，通過觀察 ab40、ab51、ab52、ab53 等 checkbit 訊號，我們能知道 firmware 運行的狀況以及其 mprj 上的值，藉此在 notebook 上驗證 firmware 的正確性。



```
async def uart_rx():
    # Reset FIFOs, enable interrupts
    ipUart.write(CTRL_REG, 1<<RST_TX | 1<<RST_RX | 1<<INTR_EN)
    print("Waitting for interrupt-operate")
    while(True):
        await intUart.wait()
        # Read FIFO until valid bit is clear
        while ((ipUart.read(STAT_REG) & (1<<RX_VALID))):
            ope_data = ipUart.read(RX_FIFO)
            if(ope_data == ord('\n')):
                print()
            else:
                print(f"{ope_data:02x}", end=' ')
```



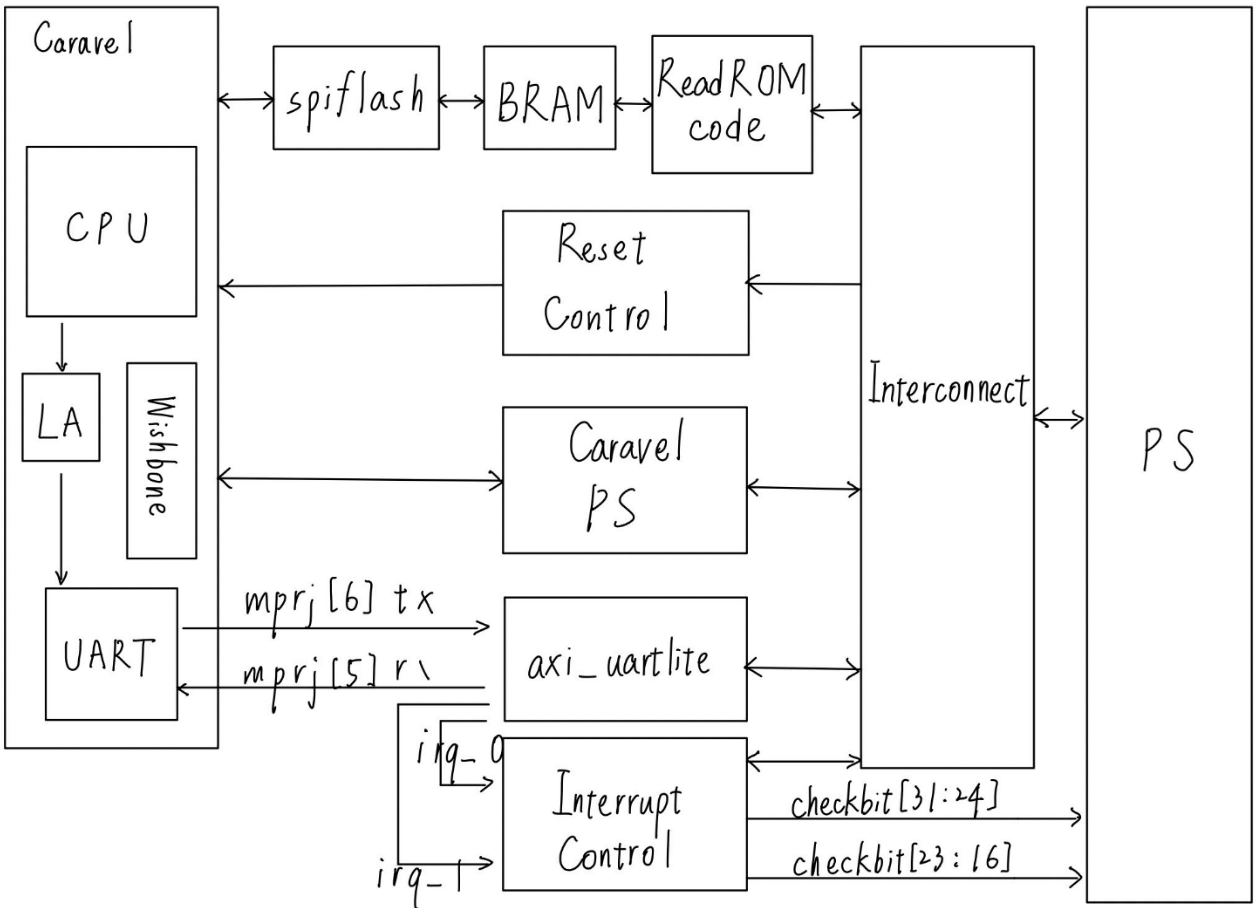
```
Waitting for interrupt-operate
ab 40
00 3e
00 44
00 4a
00 50
00 44
ab 51
00 28
03 7d
09 ed

6d
0c a1
10 ab
12 0e
16 31
17 87
23 71
ab 52
00 00
ff f6
ff e3
ff e7
00 23
00 9e
01 51
02 1b
02 dc
03 93
ab 53
Waitting for interrupt-hello
  ,time =  1.7003342252872276e+16 us
h  ,time =  56333.54187011719 us
e  ,time =  54988.861083984375 us
l  ,time =  55403.709411621094 us
l  ,time =  54411.888122558594 us
o  ,time =  54948.32992553711 us

main(): uart_rx is cancelled now
```

圖(八) caravel_fpga_uart.ipynb

二、Block Diagram

➢ 硬體模擬架構

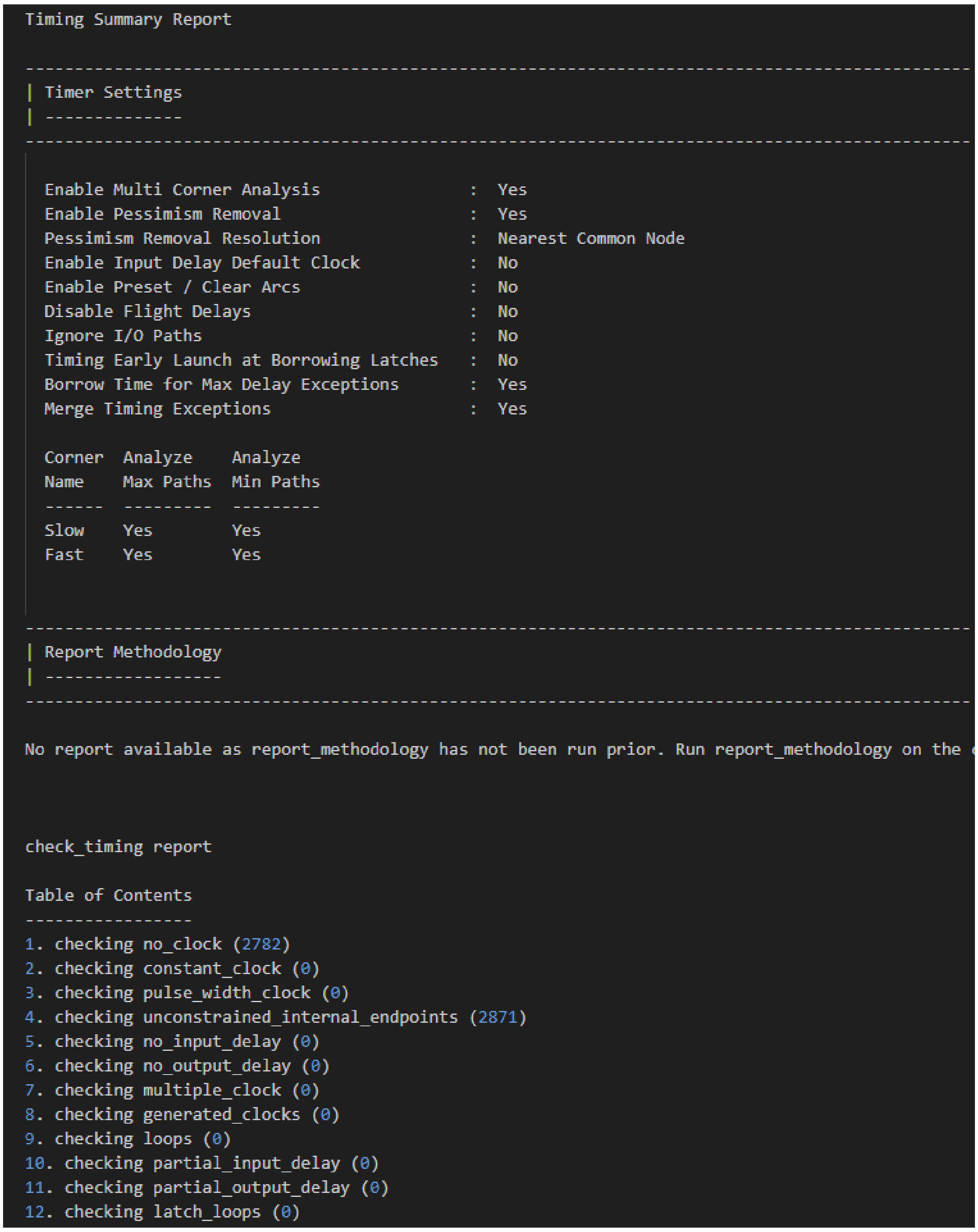
在這個架構中首先會由 cpu 將 firmware 中所放到 mprj 的資料也放到 la 上，而我們使用了兩個 irq 來完成 checkbit 的驗證，除傳送 hello 的 irq_0，每當 mprj 出現變化都會觸發 irq_1，並且他會在 isr 中使用去讀取 la 上的資料，接著利用 uart 兩次的傳輸，來完成將 checkbit 分成兩個 8bit 來傳遞的任務，最後再將兩組 8bit 的 checkbit 由 PS side 進行接收來完成整個驗證程序。



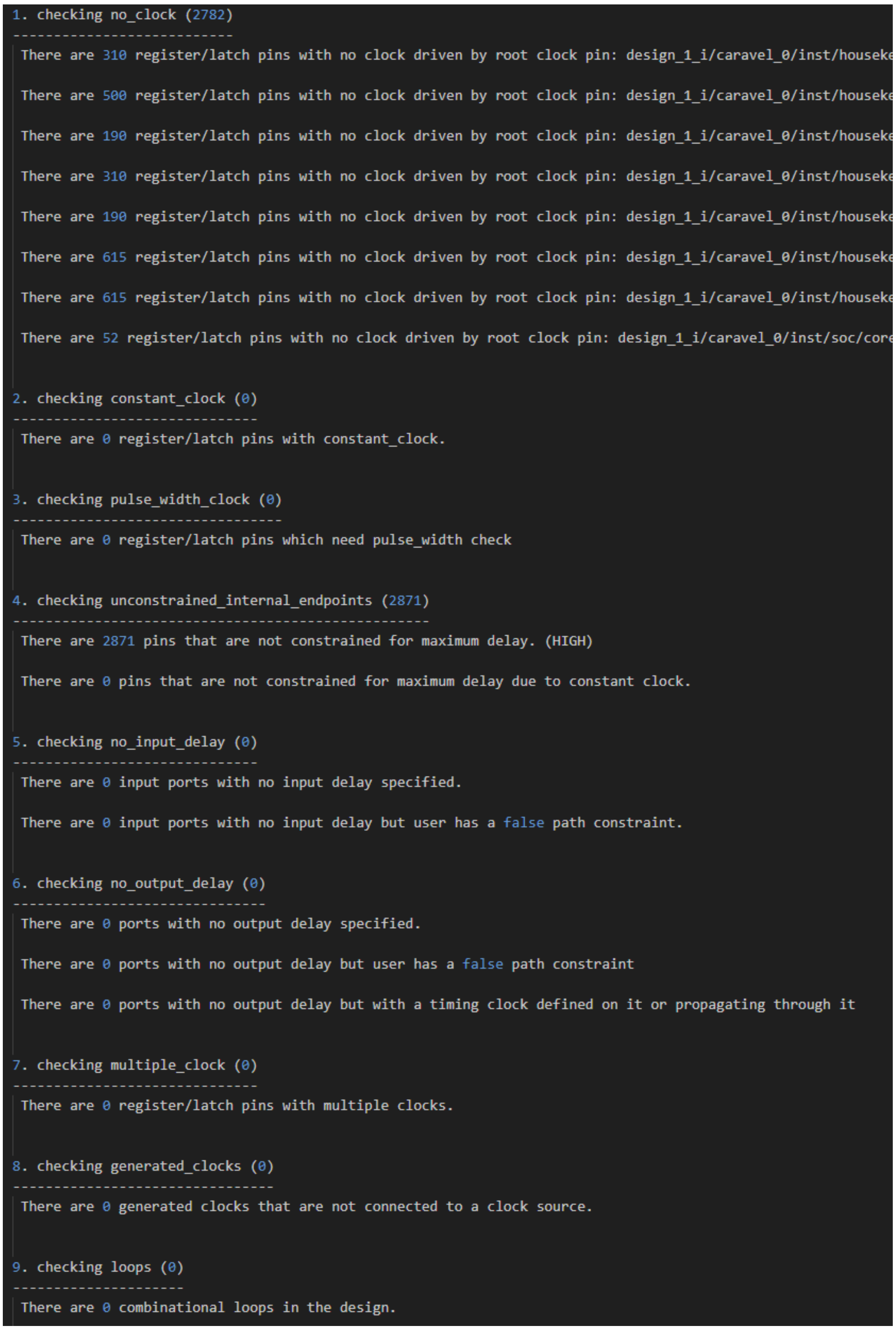
圖(九) Caravel SOC UART 架構圖

三、 Timing report/ resource report after synthesis

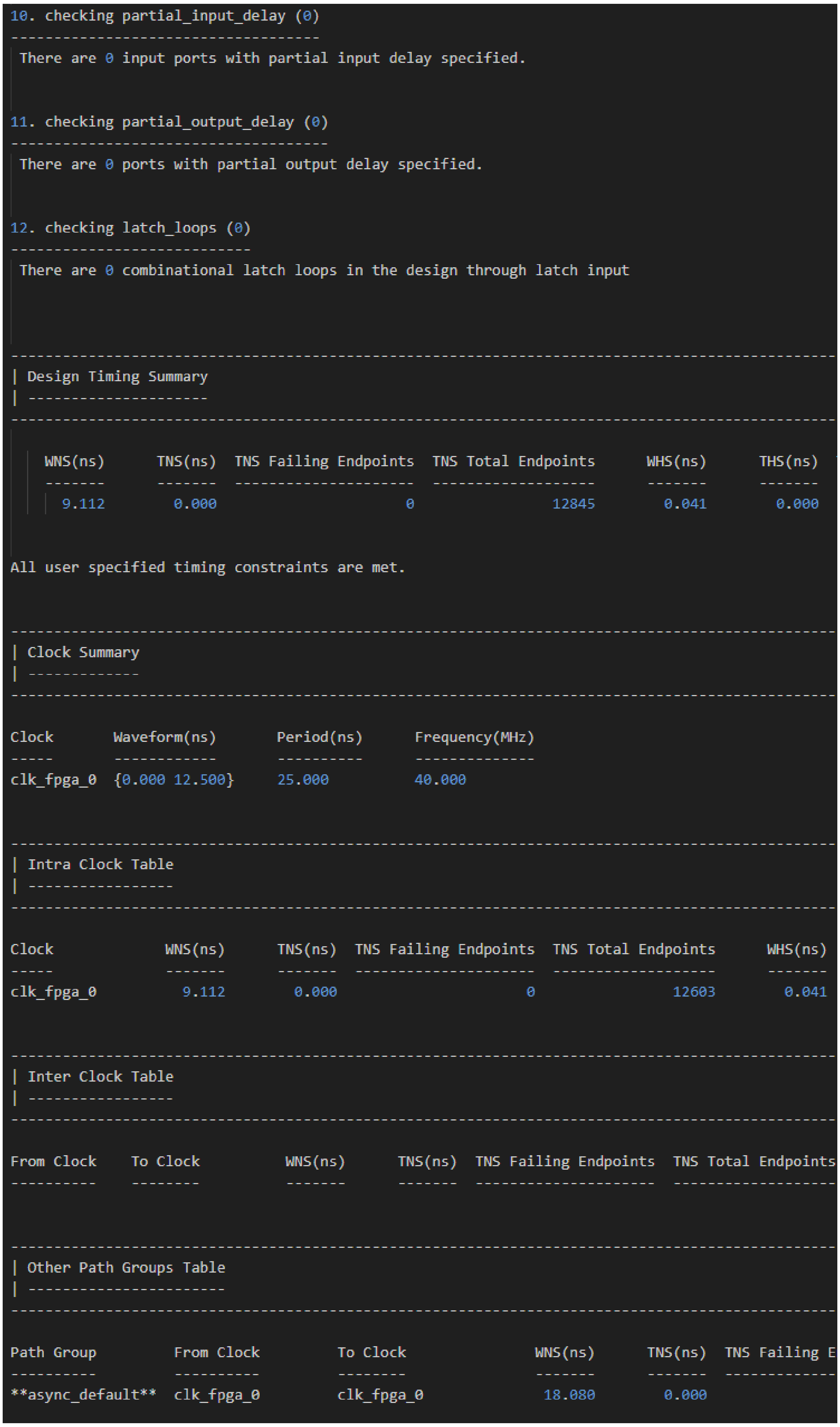
➢ Timing report



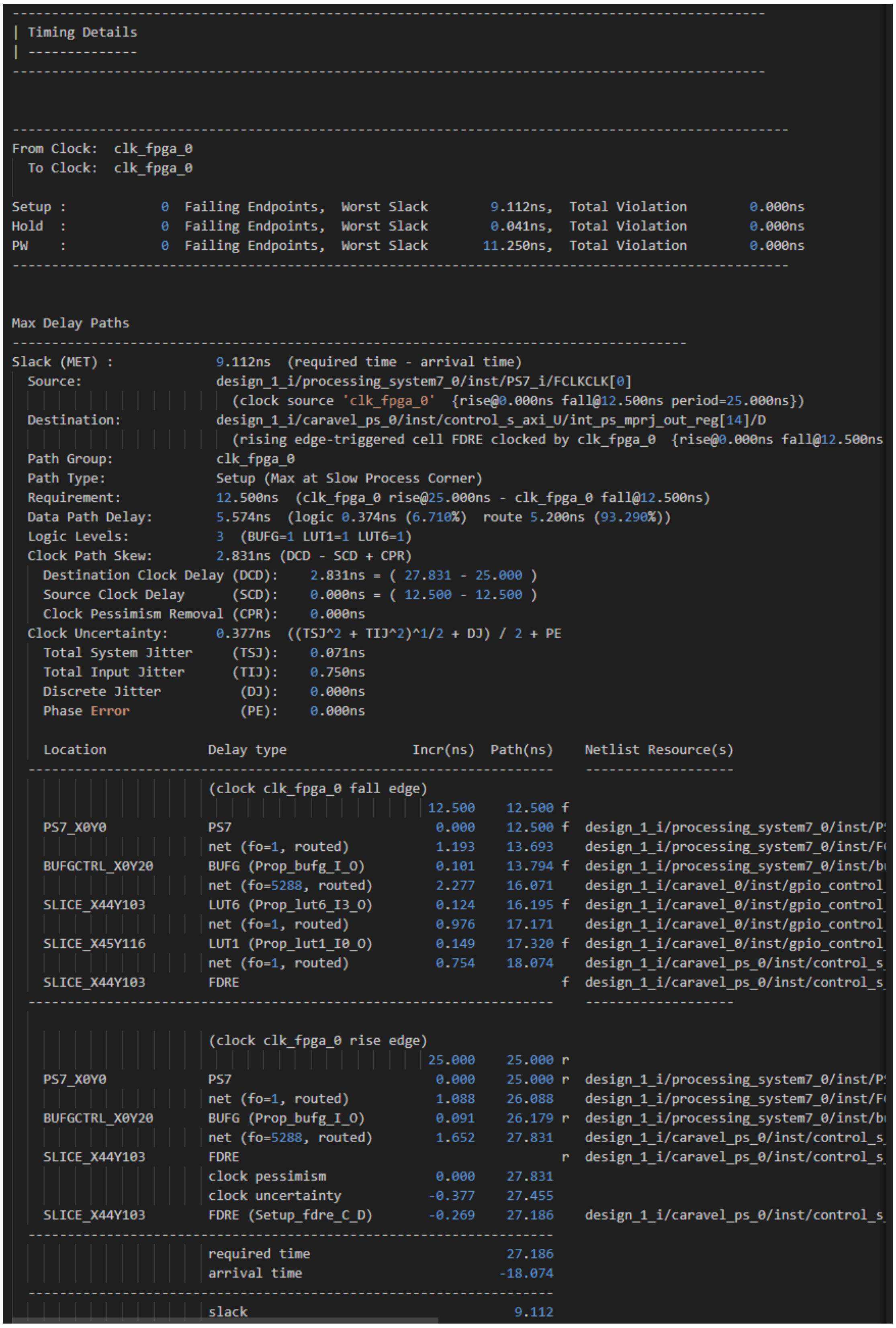
```
Timing Summary Report

------------------------------------------------------------------------------------------------
| Timer Settings
| --------------
------------------------------------------------------------------------------------------------

  Enable Multi Corner Analysis               :  Yes
  Enable Pessimism Removal                   :  Yes
  Pessimism Removal Resolution               :  Nearest Common Node
  Enable Input Delay Default Clock           :  No
  Enable Preset / Clear Arcs                 :  No
  Disable Flight Delays                      :  No
  Ignore I/O Paths                           :  No
  Timing Early Launch at Borrowing Latches   :  No
  Borrow Time for Max Delay Exceptions       :  Yes
  Merge Timing Exceptions                    :  Yes

  Corner  Analyze    Analyze
  Name    Max Paths  Min Paths
  ------  ---------  ---------
  Slow    Yes        Yes
  Fast    Yes        Yes


------------------------------------------------------------------------------------------------
| Report Methodology
| ------------------
------------------------------------------------------------------------------------------------

No report available as report_methodology has not been run prior. Run report_methodology on the 



check_timing report

Table of Contents
-----------------
1. checking no_clock (2782)
2. checking constant_clock (0)
3. checking pulse_width_clock (0)
4. checking unconstrained_internal_endpoints (2871)
5. checking no_input_delay (0)
6. checking no_output_delay (0)
7. checking multiple_clock (0)
8. checking generated_clocks (0)
9. checking loops (0)
10. checking partial_input_delay (0)
11. checking partial_output_delay (0)
12. checking latch_loops (0)
```



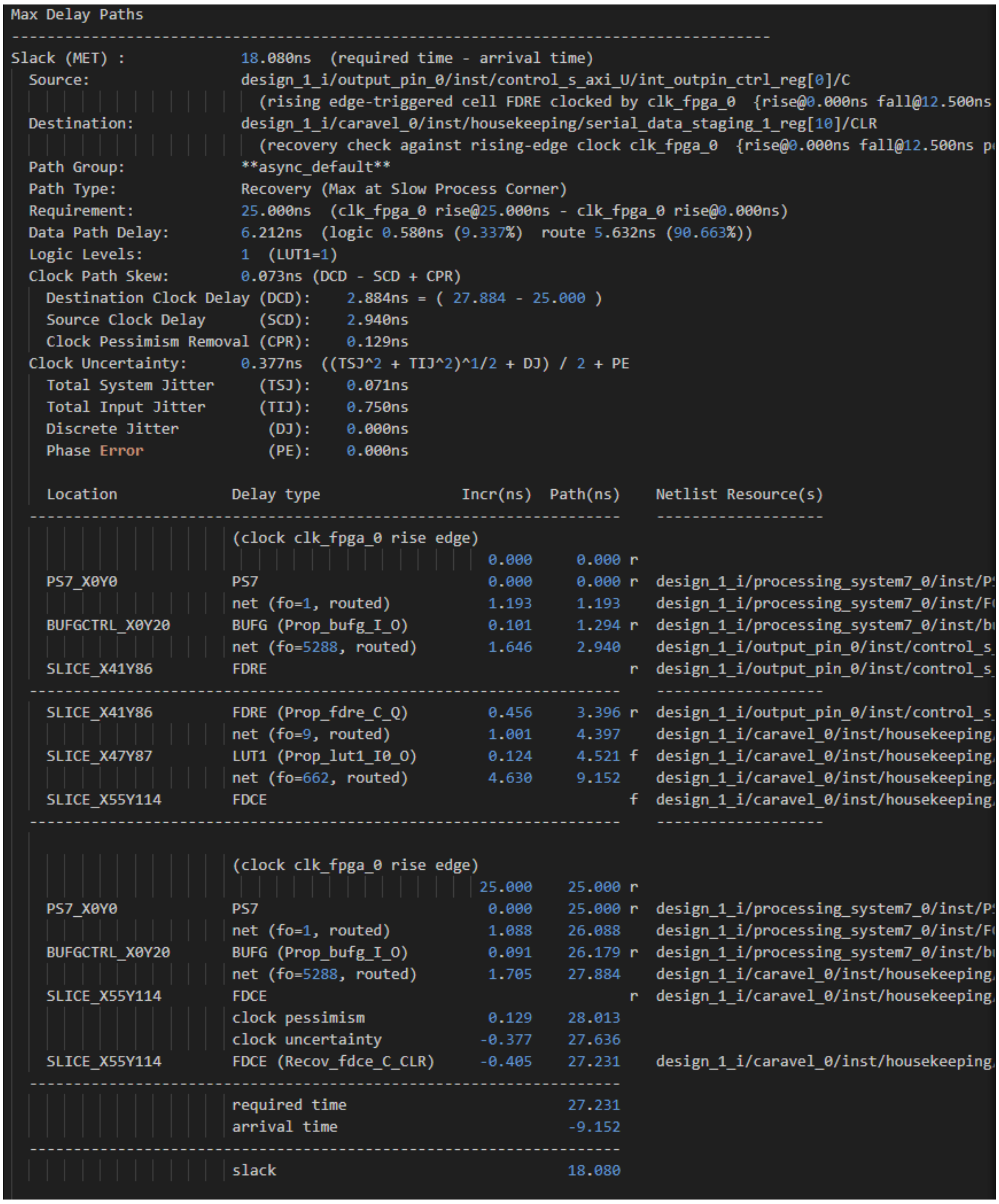
```
1. checking no_clock (2782)
---------------------------
 There are 310 register/latch pins with no clock driven by root clock pin: design_1_i/caravel_0/inst/houseke

 There are 500 register/latch pins with no clock driven by root clock pin: design_1_i/caravel_0/inst/houseke

 There are 190 register/latch pins with no clock driven by root clock pin: design_1_i/caravel_0/inst/houseke

 There are 310 register/latch pins with no clock driven by root clock pin: design_1_i/caravel_0/inst/houseke

 There are 190 register/latch pins with no clock driven by root clock pin: design_1_i/caravel_0/inst/houseke

 There are 615 register/latch pins with no clock driven by root clock pin: design_1_i/caravel_0/inst/houseke

 There are 615 register/latch pins with no clock driven by root clock pin: design_1_i/caravel_0/inst/houseke

 There are 52 register/latch pins with no clock driven by root clock pin: design_1_i/caravel_0/inst/soc/core


2. checking constant_clock (0)
------------------------------
 There are 0 register/latch pins with constant_clock.


3. checking pulse_width_clock (0)
---------------------------------
 There are 0 register/latch pins which need pulse_width check


4. checking unconstrained_internal_endpoints (2871)
---------------------------------------------------
 There are 2871 pins that are not constrained for maximum delay. (HIGH)

 There are 0 pins that are not constrained for maximum delay due to constant clock.


5. checking no_input_delay (0)
------------------------------
 There are 0 input ports with no input delay specified.

 There are 0 input ports with no input delay but user has a false path constraint.


6. checking no_output_delay (0)
-------------------------------
 There are 0 ports with no output delay specified.

 There are 0 ports with no output delay but user has a false path constraint

 There are 0 ports with no output delay but with a timing clock defined on it or propagating through it


7. checking multiple_clock (0)
------------------------------
 There are 0 register/latch pins with multiple clocks.


8. checking generated_clocks (0)
--------------------------------
 There are 0 generated clocks that are not connected to a clock source.


9. checking loops (0)
---------------------
 There are 0 combinational loops in the design.
```



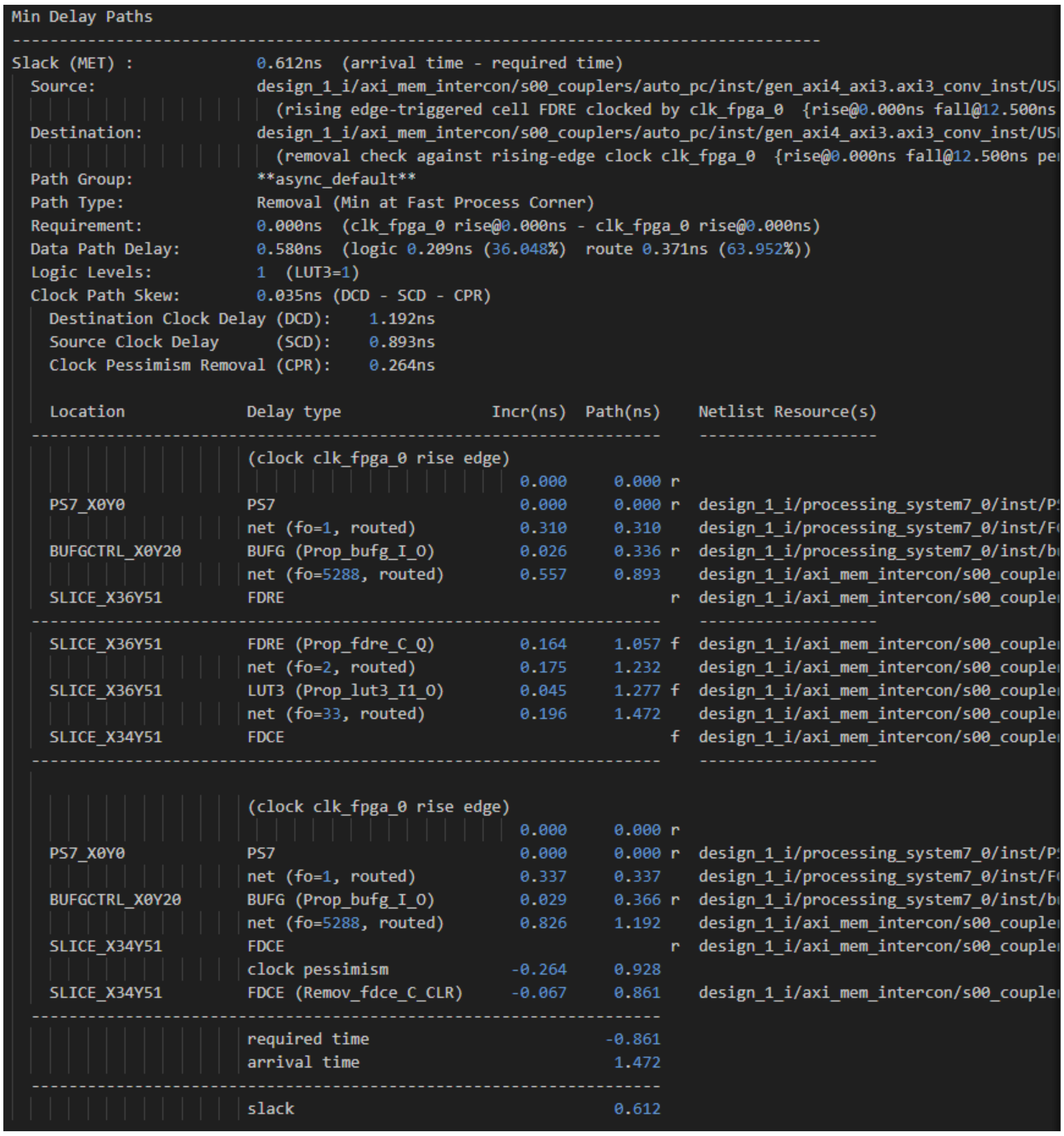
```
10. checking partial_input_delay (0)
------------------------------------
 There are 0 input ports with partial input delay specified.


11. checking partial_output_delay (0)
-------------------------------------
 There are 0 ports with partial output delay specified.


12. checking latch_loops (0)
----------------------------
 There are 0 combinational latch loops in the design through latch input



------------------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)
    -------      -------  ---------------------  -------------------      -------      -------
      9.112        0.000                      0                12845        0.041        0.000


All user specified timing constraints are met.


------------------------------------------------------------------------------------------------
| Clock Summary
| -------------
------------------------------------------------------------------------------------------------

Clock       Waveform(ns)       Period(ns)      Frequency(MHz)
-----       ------------       ----------      --------------
clk_fpga_0  {0.000 12.500}     25.000          40.000


------------------------------------------------------------------------------------------------
| Intra Clock Table
| -----------------
------------------------------------------------------------------------------------------------

Clock             WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)
-----             -------      -------  ---------------------  -------------------      -------
clk_fpga_0          9.112        0.000                      0                12603        0.041


------------------------------------------------------------------------------------------------
| Inter Clock Table
| -----------------
------------------------------------------------------------------------------------------------

From Clock    To Clock          WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints
----------    --------          -------      -------  ---------------------  -------------------


------------------------------------------------------------------------------------------------
| Other Path Groups Table
| -----------------------
------------------------------------------------------------------------------------------------

Path Group          From Clock          To Clock                WNS(ns)      TNS(ns)  TNS Failing E
----------          ----------          --------                -------      -------  ------------
**async_default**  clk_fpga_0          clk_fpga_0               18.080        0.000
```



```
------------------------------------------------------------------------------------------------
| Timing Details
| --------------
------------------------------------------------------------------------------------------------


---------------------------------------------------------------------------------------------------
From Clock:  clk_fpga_0
  To Clock:  clk_fpga_0

Setup :            0  Failing Endpoints,  Worst Slack        9.112ns,  Total Violation        0.000ns
Hold  :            0  Failing Endpoints,  Worst Slack        0.041ns,  Total Violation        0.000ns
PW    :            0  Failing Endpoints,  Worst Slack       11.250ns,  Total Violation        0.000ns
---------------------------------------------------------------------------------------------------


Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             9.112ns  (required time - arrival time)
  Source:                 design_1_i/processing_system7_0/inst/PS7_i/FCLKCLK[0]
                            (clock source 'clk_fpga_0'  {rise@0.000ns fall@12.500ns period=25.000ns})
  Destination:            design_1_i/caravel_ps_0/inst/control_s_axi_U/int_ps_mprj_out_reg[14]/D
                            (rising edge-triggered cell FDRE clocked by clk_fpga_0  {rise@0.000ns fall@12.500ns
  Path Group:             clk_fpga_0
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            12.500ns  (clk_fpga_0 rise@25.000ns - clk_fpga_0 fall@12.500ns)
  Data Path Delay:        5.574ns  (logic 0.374ns (6.710%)  route 5.200ns (93.290%))
  Logic Levels:           3  (BUFG=1 LUT1=1 LUT6=1)
  Clock Path Skew:        2.831ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    2.831ns = ( 27.831 - 25.000 )
    Source Clock Delay      (SCD):    0.000ns = ( 12.500 - 12.500 )
    Clock Pessimism Removal (CPR):    0.000ns
  Clock Uncertainty:      0.377ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Total Input Jitter      (TIJ):    0.750ns
    Discrete Jitter          (DJ):    0.000ns
    Phase Error              (PE):    0.000ns

    Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
  -------------------------------------------------------------------    -------------------
                         (clock clk_fpga_0 fall edge)
                                                     12.500    12.500 f
    PS7_X0Y0             PS7                          0.000    12.500 f  design_1_i/processing_system7_0/inst/P
                         net (fo=1, routed)           1.193    13.693    design_1_i/processing_system7_0/inst/F
    BUFGCTRL_X0Y20       BUFG (Prop_bufg_I_O)         0.101    13.794 f  design_1_i/processing_system7_0/inst/b
                         net (fo=5288, routed)        2.277    16.071    design_1_i/caravel_0/inst/gpio_control_
    SLICE_X44Y103        LUT6 (Prop_lut6_I3_O)        0.124    16.195 f  design_1_i/caravel_0/inst/gpio_control_
                         net (fo=1, routed)           0.976    17.171    design_1_i/caravel_0/inst/gpio_control_
    SLICE_X45Y116        LUT1 (Prop_lut1_I0_O)        0.149    17.320 f  design_1_i/caravel_0/inst/gpio_control_
                         net (fo=1, routed)           0.754    18.074    design_1_i/caravel_ps_0/inst/control_s_
    SLICE_X44Y103        FDRE                                         f  design_1_i/caravel_ps_0/inst/control_s_
  -------------------------------------------------------------------    -------------------

                         (clock clk_fpga_0 rise edge)
                                                     25.000    25.000 r
    PS7_X0Y0             PS7                          0.000    25.000 r  design_1_i/processing_system7_0/inst/P
                         net (fo=1, routed)           1.088    26.088    design_1_i/processing_system7_0/inst/F
    BUFGCTRL_X0Y20       BUFG (Prop_bufg_I_O)         0.091    26.179 r  design_1_i/processing_system7_0/inst/b
                         net (fo=5288, routed)        1.652    27.831    design_1_i/caravel_ps_0/inst/control_s_
    SLICE_X44Y103        FDRE                                         r  design_1_i/caravel_ps_0/inst/control_s_
                         clock pessimism              0.000    27.831
                         clock uncertainty           -0.377    27.455
    SLICE_X44Y103        FDRE (Setup_fdre_C_D)       -0.269    27.186    design_1_i/caravel_ps_0/inst/control_s_
  -------------------------------------------------------------------
                         required time                         27.186
                         arrival time                         -18.074
  -------------------------------------------------------------------
                         slack                                  9.112
```

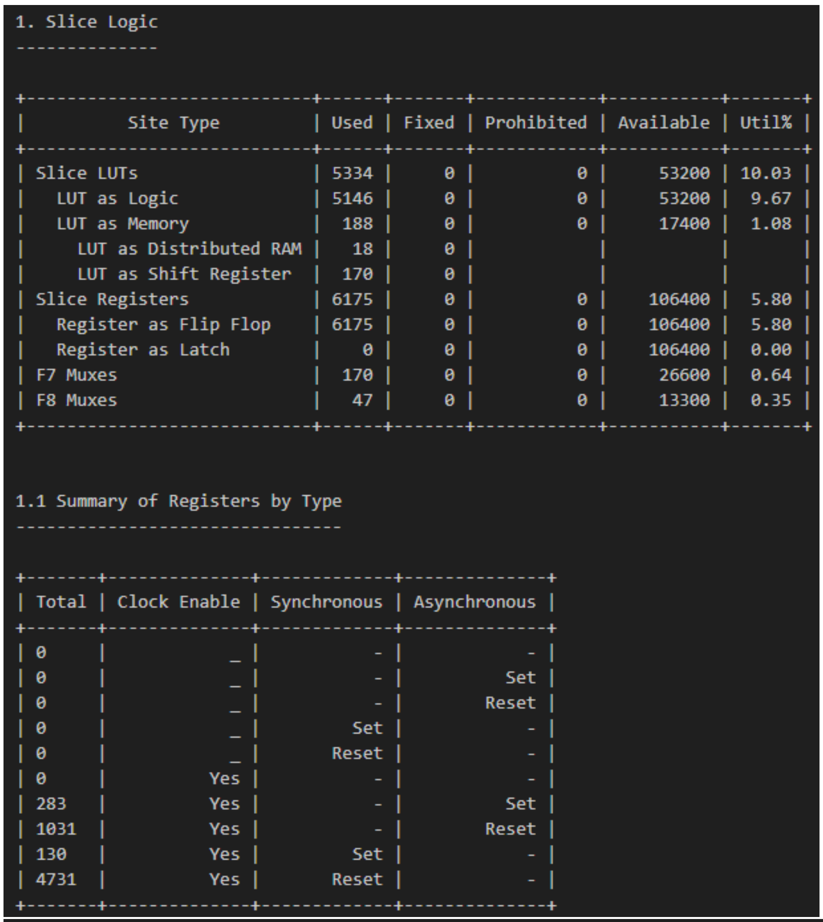


```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             18.080ns  (required time - arrival time)
  Source:                 design_1_i/output_pin_0/inst/control_s_axi_U/int_outpin_ctrl_reg[0]/C
                            (rising edge-triggered cell FDRE clocked by clk_fpga_0  {rise@0.000ns fall@12.500ns
  Destination:            design_1_i/caravel_0/inst/housekeeping/serial_data_staging_1_reg[10]/CLR
                            (recovery check against rising-edge clock clk_fpga_0  {rise@0.000ns fall@12.500ns p
  Path Group:             **async_default**
  Path Type:              Recovery (Max at Slow Process Corner)
  Requirement:            25.000ns  (clk_fpga_0 rise@25.000ns - clk_fpga_0 rise@0.000ns)
  Data Path Delay:        6.212ns  (logic 0.580ns (9.337%)  route 5.632ns (90.663%))
  Logic Levels:           1  (LUT1=1)
  Clock Path Skew:        0.073ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    2.884ns = ( 27.884 - 25.000 )
    Source Clock Delay      (SCD):    2.940ns
    Clock Pessimism Removal (CPR):    0.129ns
  Clock Uncertainty:      0.377ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Total Input Jitter      (TIJ):    0.750ns
    Discrete Jitter          (DJ):    0.000ns
    Phase Error              (PE):    0.000ns

    Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
  -------------------------------------------------------------------    -------------------
                         (clock clk_fpga_0 rise edge)
                                                      0.000     0.000 r
    PS7_X0Y0             PS7                          0.000     0.000 r  design_1_i/processing_system7_0/inst/P
                         net (fo=1, routed)           1.193     1.193    design_1_i/processing_system7_0/inst/F
    BUFGCTRL_X0Y20       BUFG (Prop_bufg_I_O)         0.101     1.294 r  design_1_i/processing_system7_0/inst/b
                         net (fo=5288, routed)        1.646     2.940    design_1_i/output_pin_0/inst/control_s
    SLICE_X41Y86         FDRE                                         r  design_1_i/output_pin_0/inst/control_s
  -------------------------------------------------------------------    -------------------
    SLICE_X41Y86         FDRE (Prop_fdre_C_Q)         0.456     3.396 r  design_1_i/output_pin_0/inst/control_s
                         net (fo=9, routed)           1.001     4.397    design_1_i/caravel_0/inst/housekeeping
    SLICE_X47Y87         LUT1 (Prop_lut1_I0_O)        0.124     4.521 f  design_1_i/caravel_0/inst/housekeeping
                         net (fo=662, routed)         4.630     9.152    design_1_i/caravel_0/inst/housekeeping
    SLICE_X55Y114        FDCE                                         f  design_1_i/caravel_0/inst/housekeeping
  -------------------------------------------------------------------    -------------------

                         (clock clk_fpga_0 rise edge)
                                                     25.000    25.000 r
    PS7_X0Y0             PS7                          0.000    25.000 r  design_1_i/processing_system7_0/inst/P
                         net (fo=1, routed)           1.088    26.088    design_1_i/processing_system7_0/inst/F
    BUFGCTRL_X0Y20       BUFG (Prop_bufg_I_O)         0.091    26.179 r  design_1_i/processing_system7_0/inst/b
                         net (fo=5288, routed)        1.705    27.884    design_1_i/caravel_0/inst/housekeeping
    SLICE_X55Y114        FDCE                                         r  design_1_i/caravel_0/inst/housekeeping
                         clock pessimism              0.129    28.013
                         clock uncertainty           -0.377    27.636
    SLICE_X55Y114        FDCE (Recov_fdce_C_CLR)     -0.405    27.231    design_1_i/caravel_0/inst/housekeeping
  -------------------------------------------------------------------
                         required time                         27.231
                         arrival time                          -9.152
  -------------------------------------------------------------------
                         slack                                 18.080
```

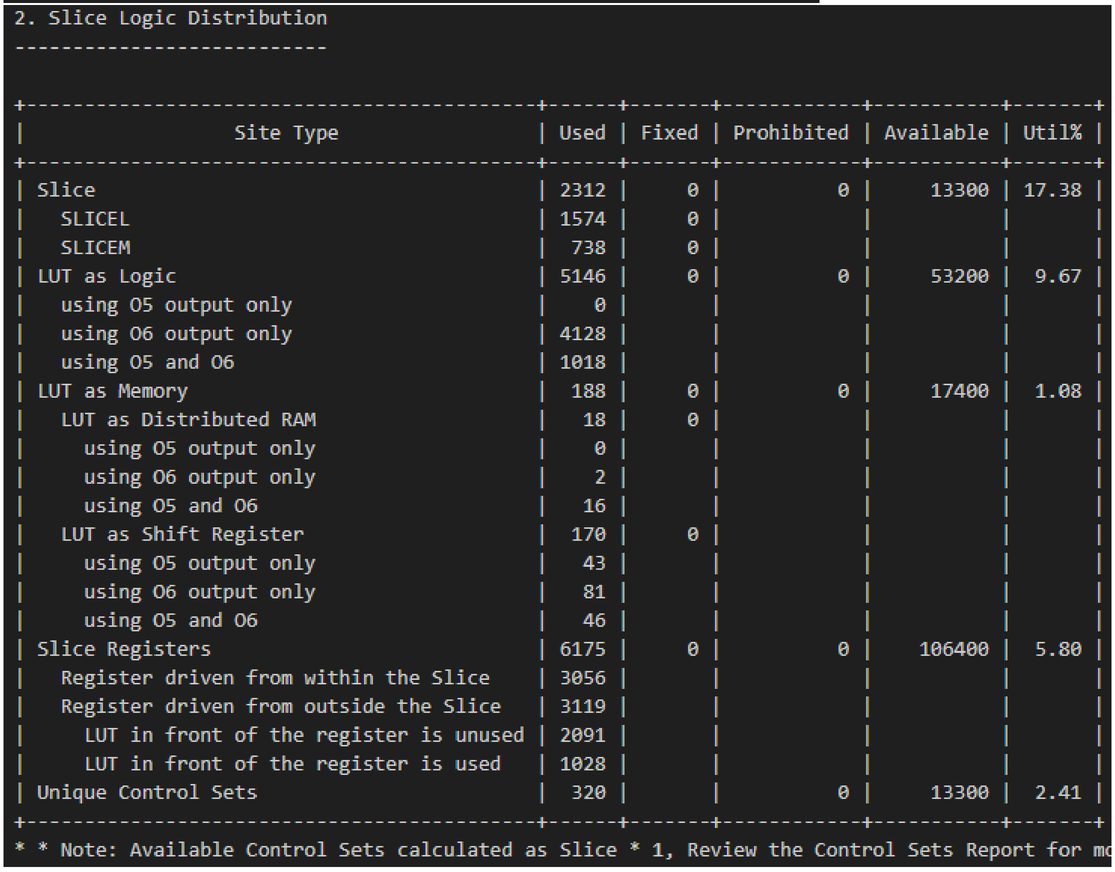


```
Min Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.612ns  (arrival time - required time)
  Source:                 design_1_i/axi_mem_intercon/s00_couplers/auto_pc/inst/gen_axi4_axi3.axi3_conv_inst/US
                            (rising edge-triggered cell FDRE clocked by clk_fpga_0  {rise@0.000ns fall@12.500ns
  Destination:            design_1_i/axi_mem_intercon/s00_couplers/auto_pc/inst/gen_axi4_axi3.axi3_conv_inst/US
                            (removal check against rising-edge clock clk_fpga_0  {rise@0.000ns fall@12.500ns pe
  Path Group:             **async_default**
  Path Type:              Removal (Min at Fast Process Corner)
  Requirement:            0.000ns  (clk_fpga_0 rise@0.000ns - clk_fpga_0 rise@0.000ns)
  Data Path Delay:        0.580ns  (logic 0.209ns (36.048%)  route 0.371ns (63.952%))
  Logic Levels:           1  (LUT3=1)
  Clock Path Skew:        0.035ns (DCD - SCD - CPR)
    Destination Clock Delay (DCD):    1.192ns
    Source Clock Delay      (SCD):    0.893ns
    Clock Pessimism Removal (CPR):    0.264ns

    Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
  -------------------------------------------------------------------    -------------------
                         (clock clk_fpga_0 rise edge)
                                                      0.000     0.000 r
    PS7_X0Y0             PS7                          0.000     0.000 r  design_1_i/processing_system7_0/inst/P
                         net (fo=1, routed)           0.310     0.310    design_1_i/processing_system7_0/inst/F
    BUFGCTRL_X0Y20       BUFG (Prop_bufg_I_O)         0.026     0.336 r  design_1_i/processing_system7_0/inst/b
                         net (fo=5288, routed)        0.557     0.893    design_1_i/axi_mem_intercon/s00_couple
    SLICE_X36Y51         FDRE                                         r  design_1_i/axi_mem_intercon/s00_couple
  -------------------------------------------------------------------    -------------------
    SLICE_X36Y51         FDRE (Prop_fdre_C_Q)         0.164     1.057 f  design_1_i/axi_mem_intercon/s00_couple
                         net (fo=2, routed)           0.175     1.232    design_1_i/axi_mem_intercon/s00_couple
    SLICE_X36Y51         LUT3 (Prop_lut3_I1_O)        0.045     1.277 f  design_1_i/axi_mem_intercon/s00_couple
                         net (fo=33, routed)          0.196     1.472    design_1_i/axi_mem_intercon/s00_couple
    SLICE_X34Y51         FDCE                                         f  design_1_i/axi_mem_intercon/s00_couple
  -------------------------------------------------------------------    -------------------

                         (clock clk_fpga_0 rise edge)
                                                      0.000     0.000 r
    PS7_X0Y0             PS7                          0.000     0.000 r  design_1_i/processing_system7_0/inst/P
                         net (fo=1, routed)           0.337     0.337    design_1_i/processing_system7_0/inst/F
    BUFGCTRL_X0Y20       BUFG (Prop_bufg_I_O)         0.029     0.366 r  design_1_i/processing_system7_0/inst/b
                         net (fo=5288, routed)        0.826     1.192    design_1_i/axi_mem_intercon/s00_couple
    SLICE_X34Y51         FDCE                                         r  design_1_i/axi_mem_intercon/s00_couple
                         clock pessimism             -0.264     0.928
    SLICE_X34Y51         FDCE (Remov_fdce_C_CLR)     -0.067     0.861    design_1_i/axi_mem_intercon/s00_couple
  -------------------------------------------------------------------
                         required time                         -0.861
                         arrival time                           1.472
  -------------------------------------------------------------------
                         slack                                  0.612
```

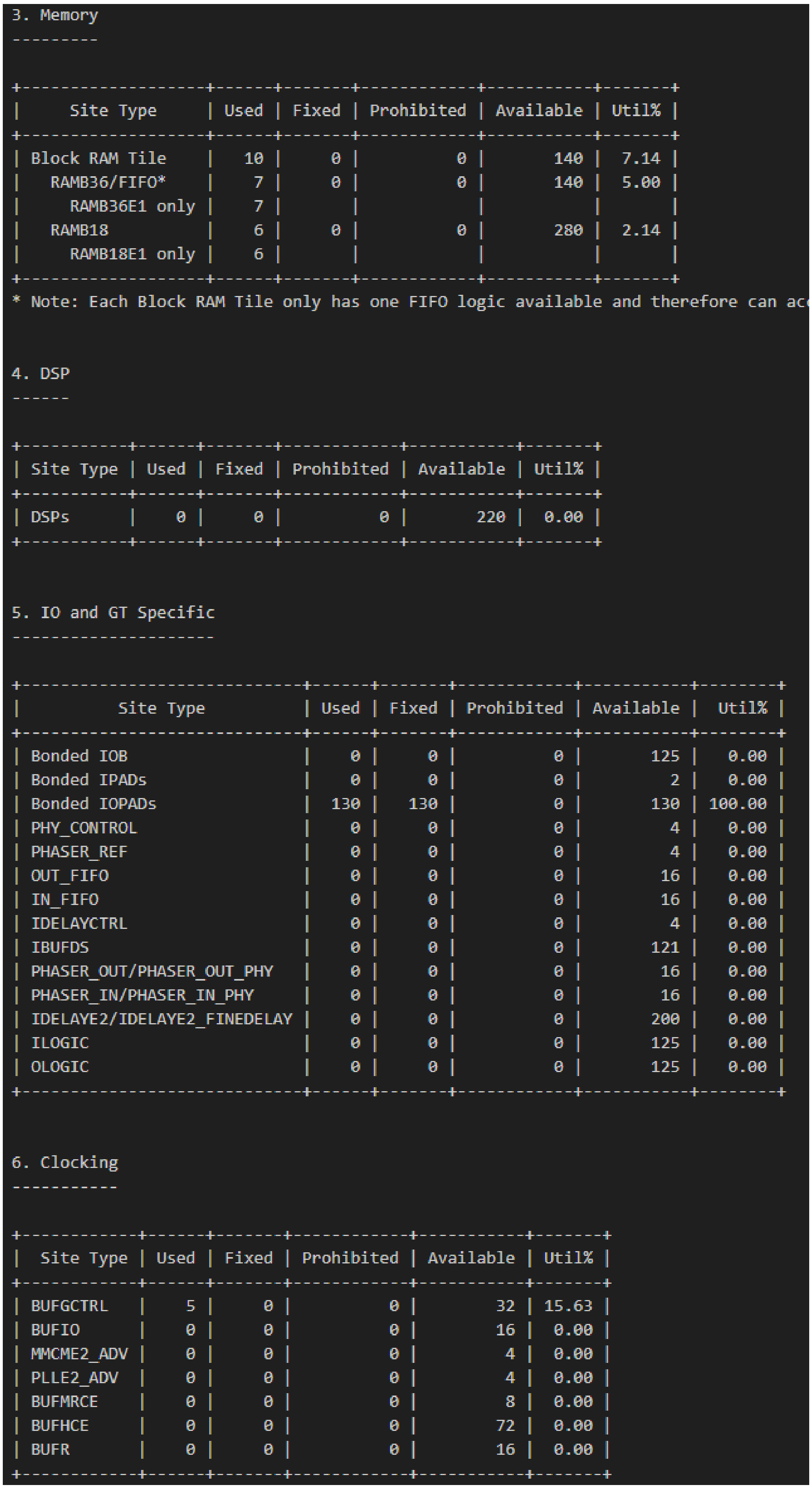
➢ Resource report



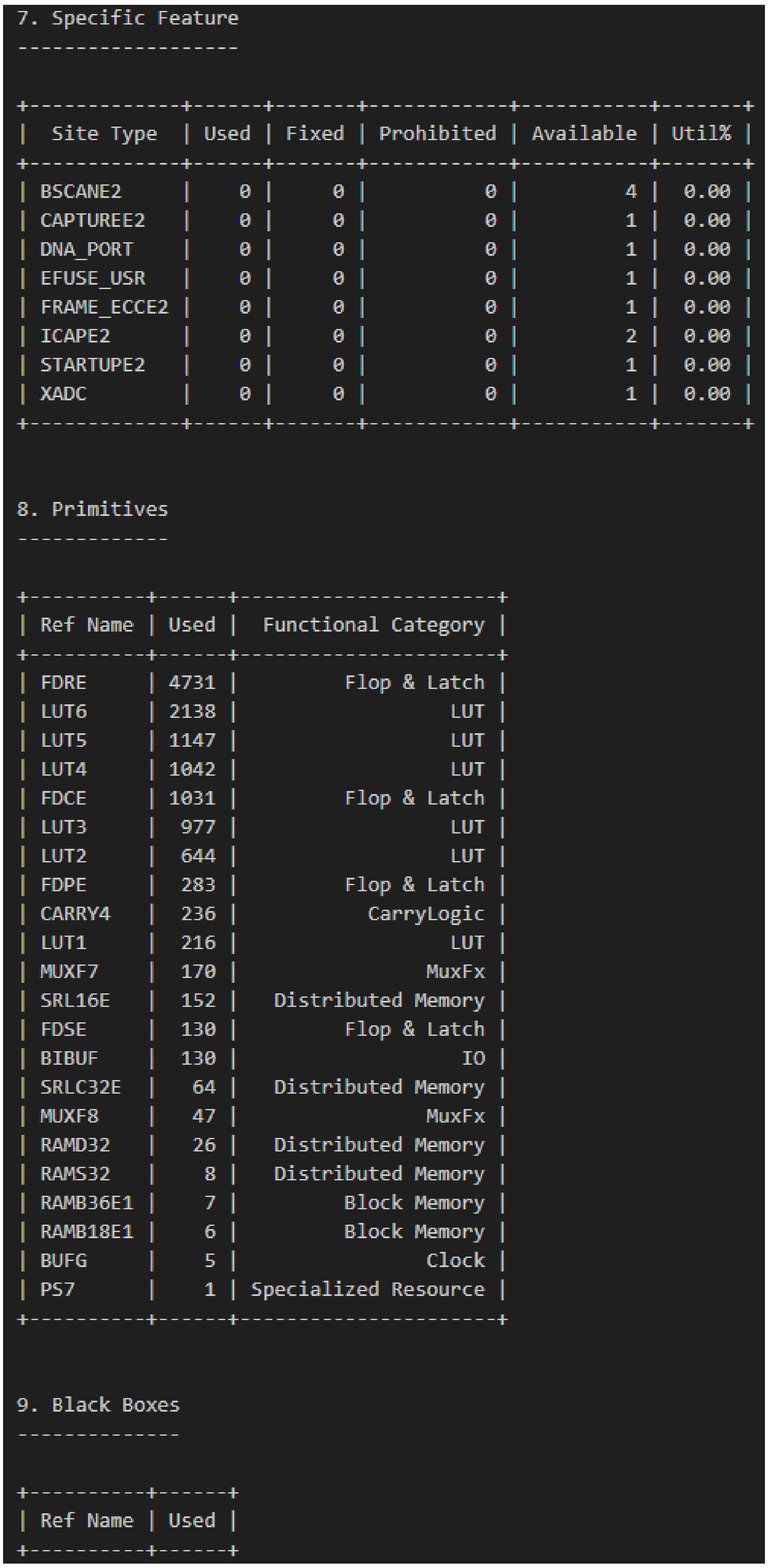
```
1. Slice Logic
-------------

+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs              | 5334 |     0 |          0 |     53200 | 10.03 |
|   LUT as Logic          | 5146 |     0 |          0 |     53200 |  9.67 |
|   LUT as Memory         |  188 |     0 |          0 |     17400 |  1.08 |
|     LUT as Distributed RAM |   18 |     0 |            |           |       |
|     LUT as Shift Register  |  170 |     0 |            |           |       |
| Slice Registers         | 6175 |     0 |          0 |    106400 |  5.80 |
|   Register as Flip Flop | 6175 |     0 |          0 |    106400 |  5.80 |
|   Register as Latch     |    0 |     0 |          0 |    106400 |  0.00 |
| F7 Muxes                |  170 |     0 |          0 |     26600 |  0.64 |
| F8 Muxes                |   47 |     0 |          0 |     13300 |  0.35 |
+-------------------------+------+-------+------------+-----------+-------+


1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |            _ |           - |            - |
| 0     |            _ |           - |          Set |
| 0     |            _ |           - |        Reset |
| 0     |            _ |         Set |            - |
| 0     |            _ |       Reset |            - |
| 0     |          Yes |           - |            - |
| 283   |          Yes |           - |          Set |
| 1031  |          Yes |           - |        Reset |
| 130   |          Yes |         Set |            - |
| 4731  |          Yes |       Reset |            - |
+-------+--------------+-------------+--------------+
```



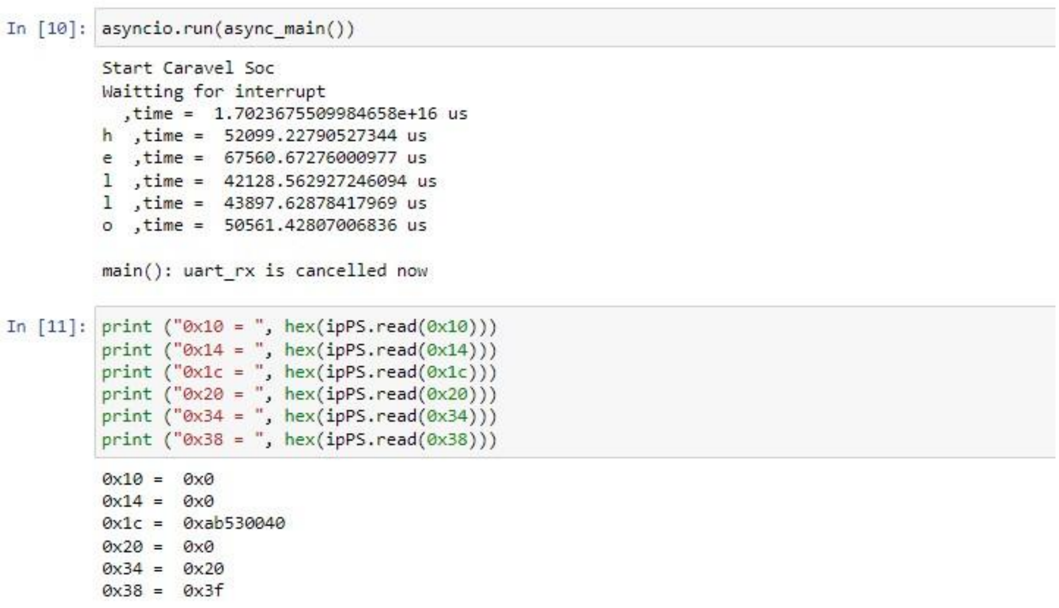
```
2. Slice Logic Distribution
---------------------------

+-------------------------------------------+------+-------+------------+-----------+-------+
|                 Site Type                 | Used | Fixed | Prohibited | Available | Util% |
+-------------------------------------------+------+-------+------------+-----------+-------+
| Slice                                     | 2312 |     0 |          0 |     13300 | 17.38 |
|   SLICEL                                  | 1574 |     0 |            |           |       |
|   SLICEM                                  |  738 |     0 |            |           |       |
| LUT as Logic                              | 5146 |     0 |          0 |     53200 |  9.67 |
|   using O5 output only                    |    0 |       |            |           |       |
|   using O6 output only                    | 4128 |       |            |           |       |
|   using O5 and O6                         | 1018 |       |            |           |       |
| LUT as Memory                             |  188 |     0 |          0 |     17400 |  1.08 |
|   LUT as Distributed RAM                  |   18 |     0 |            |           |       |
|     using O5 output only                  |    0 |       |            |           |       |
|     using O6 output only                  |    2 |       |            |           |       |
|     using O5 and O6                       |   16 |       |            |           |       |
|   LUT as Shift Register                   |  170 |     0 |            |           |       |
|     using O5 output only                  |   43 |       |            |           |       |
|     using O6 output only                  |   81 |       |            |           |       |
|     using O5 and O6                       |   46 |       |            |           |       |
| Slice Registers                           | 6175 |     0 |          0 |    106400 |  5.80 |
|   Register driven from within the Slice   | 3056 |       |            |           |       |
|   Register driven from outside the Slice  | 3119 |       |            |           |       |
|     LUT in front of the register is unused | 2091 |       |            |           |       |
|     LUT in front of the register is used  | 1028 |       |            |           |       |
| Unique Control Sets                       |  320 |       |          0 |     13300 |  2.41 |
+-------------------------------------------+------+-------+------------+-----------+-------+
* * Note: Available Control Sets calculated as Slice * 1, Review the Control Sets Report for mo
```



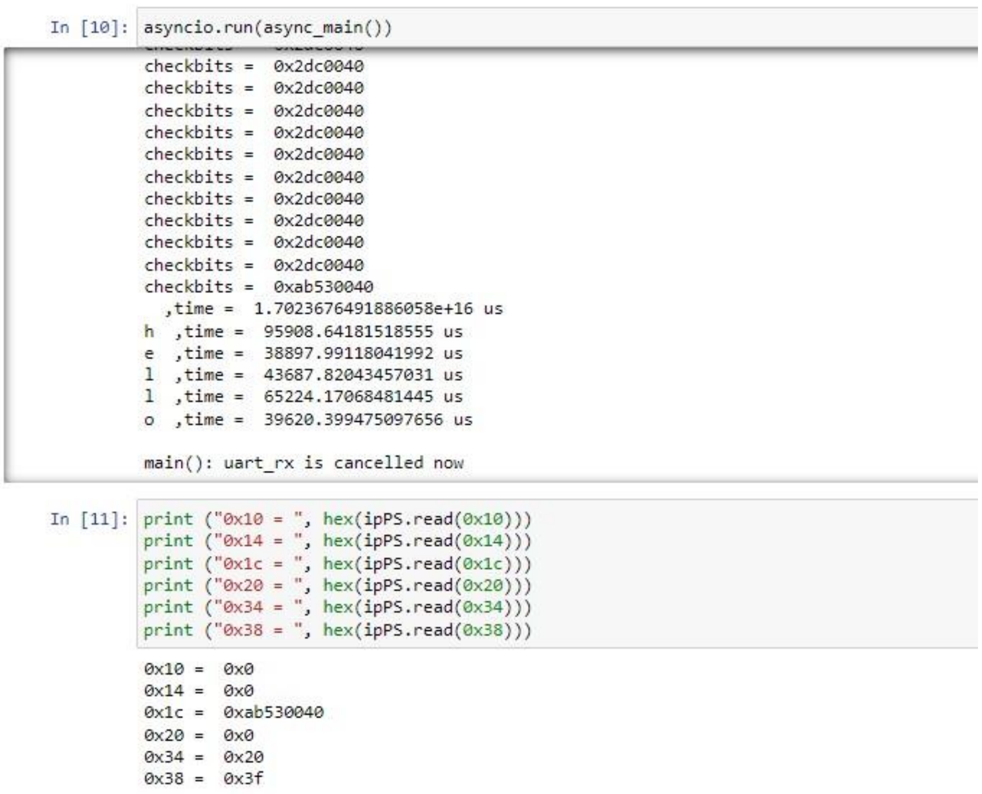
```
3. Memory
---------

+-------------------+------+-------+------------+-----------+-------+
|     Site Type     | Used | Fixed | Prohibited | Available | Util% |
+-------------------+------+-------+------------+-----------+-------+
| Block RAM Tile    |   10 |     0 |          0 |       140 |  7.14 |
|   RAMB36/FIFO*    |    7 |     0 |          0 |       140 |  5.00 |
|     RAMB36E1 only |    7 |       |            |           |       |
|   RAMB18          |    6 |     0 |          0 |       280 |  2.14 |
|     RAMB18E1 only |    6 |       |            |           |       |
+-------------------+------+-------+------------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and therefore can ac


4. DSP
------

+-----------+------+-------+------------+-----------+-------+
| Site Type | Used | Fixed | Prohibited | Available | Util% |
+-----------+------+-------+------------+-----------+-------+
| DSPs      |    0 |     0 |          0 |       220 |  0.00 |
+-----------+------+-------+------------+-----------+-------+


5. IO and GT Specific
--------------------

+-----------------------------+------+-------+------------+-----------+--------+
|          Site Type          | Used | Fixed | Prohibited | Available |  Util% |
+-----------------------------+------+-------+------------+-----------+--------+
| Bonded IOB                  |    0 |     0 |          0 |       125 |   0.00 |
| Bonded IPADs                |    0 |     0 |          0 |         2 |   0.00 |
| Bonded IOPADs               |  130 |   130 |          0 |       130 | 100.00 |
| PHY_CONTROL                 |    0 |     0 |          0 |         4 |   0.00 |
| PHASER_REF                  |    0 |     0 |          0 |         4 |   0.00 |
| OUT_FIFO                    |    0 |     0 |          0 |        16 |   0.00 |
| IN_FIFO                     |    0 |     0 |          0 |        16 |   0.00 |
| IDELAYCTRL                  |    0 |     0 |          0 |         4 |   0.00 |
| IBUFDS                      |    0 |     0 |          0 |       121 |   0.00 |
| PHASER_OUT/PHASER_OUT_PHY   |    0 |     0 |          0 |        16 |   0.00 |
| PHASER_IN/PHASER_IN_PHY     |    0 |     0 |          0 |        16 |   0.00 |
| IDELAYE2/IDELAYE2_FINEDELAY |    0 |     0 |          0 |       200 |   0.00 |
| ILOGIC                      |    0 |     0 |          0 |       125 |   0.00 |
| OLOGIC                      |    0 |     0 |          0 |       125 |   0.00 |
+-----------------------------+------+-------+------------+-----------+--------+


6. Clocking
-----------

+------------+------+-------+------------+-----------+-------+
|  Site Type | Used | Fixed | Prohibited | Available | Util% |
+------------+------+-------+------------+-----------+-------+
| BUFGCTRL   |    5 |     0 |          0 |        32 | 15.63 |
| BUFIO      |    0 |     0 |          0 |        16 |  0.00 |
| MMCME2_ADV |    0 |     0 |          0 |         4 |  0.00 |
| PLLE2_ADV  |    0 |     0 |          0 |         4 |  0.00 |
| BUFMRCE    |    0 |     0 |          0 |         8 |  0.00 |
| BUFHCE     |    0 |     0 |          0 |        72 |  0.00 |
| BUFR       |    0 |     0 |          0 |        16 |  0.00 |
+------------+------+-------+------------+-----------+-------+
```


```
7. Specific Feature
-------------------

+-------------+------+-------+------------+-----------+-------+
|  Site Type  | Used | Fixed | Prohibited | Available | Util% |
+-------------+------+-------+------------+-----------+-------+
| BSCANE2     |    0 |     0 |          0 |         4 |  0.00 |
| CAPTUREE2   |    0 |     0 |          0 |         1 |  0.00 |
| DNA_PORT    |    0 |     0 |          0 |         1 |  0.00 |
| EFUSE_USR   |    0 |     0 |          0 |         1 |  0.00 |
| FRAME_ECCE2 |    0 |     0 |          0 |         1 |  0.00 |
| ICAPE2      |    0 |     0 |          0 |         2 |  0.00 |
| STARTUPE2   |    0 |     0 |          0 |         1 |  0.00 |
| XADC        |    0 |     0 |          0 |         1 |  0.00 |
+-------------+------+-------+------------+-----------+-------+


8. Primitives
-------------

+----------+------+---------------------+
| Ref Name | Used | Functional Category |
+----------+------+---------------------+
| FDRE     | 4731 |        Flop & Latch |
| LUT6     | 2138 |                 LUT |
| LUT5     | 1147 |                 LUT |
| LUT4     | 1042 |                 LUT |
| FDCE     | 1031 |        Flop & Latch |
| LUT3     |  977 |                 LUT |
| LUT2     |  644 |                 LUT |
| FDPE     |  283 |        Flop & Latch |
| CARRY4   |  236 |          CarryLogic |
| LUT1     |  216 |                 LUT |
| MUXF7    |  170 |               MuxFx |
| SRL16E   |  152 |  Distributed Memory |
| FDSE     |  130 |        Flop & Latch |
| BIBUF    |  130 |                  IO |
| SRLC32E  |   64 |  Distributed Memory |
| MUXF8    |   47 |               MuxFx |
| RAMD32   |   26 |  Distributed Memory |
| RAMS32   |    8 |  Distributed Memory |
| RAMB36E1 |    7 |        Block Memory |
| RAMB18E1 |    6 |        Block Memory |
| BUFG     |    5 |               Clock |
| PS7      |    1 | Specialized Resource |
+----------+------+---------------------+


9. Black Boxes
--------------

+----------+------+
| Ref Name | Used |
+----------+------+
```

四、 Latency for a character loop back using UART

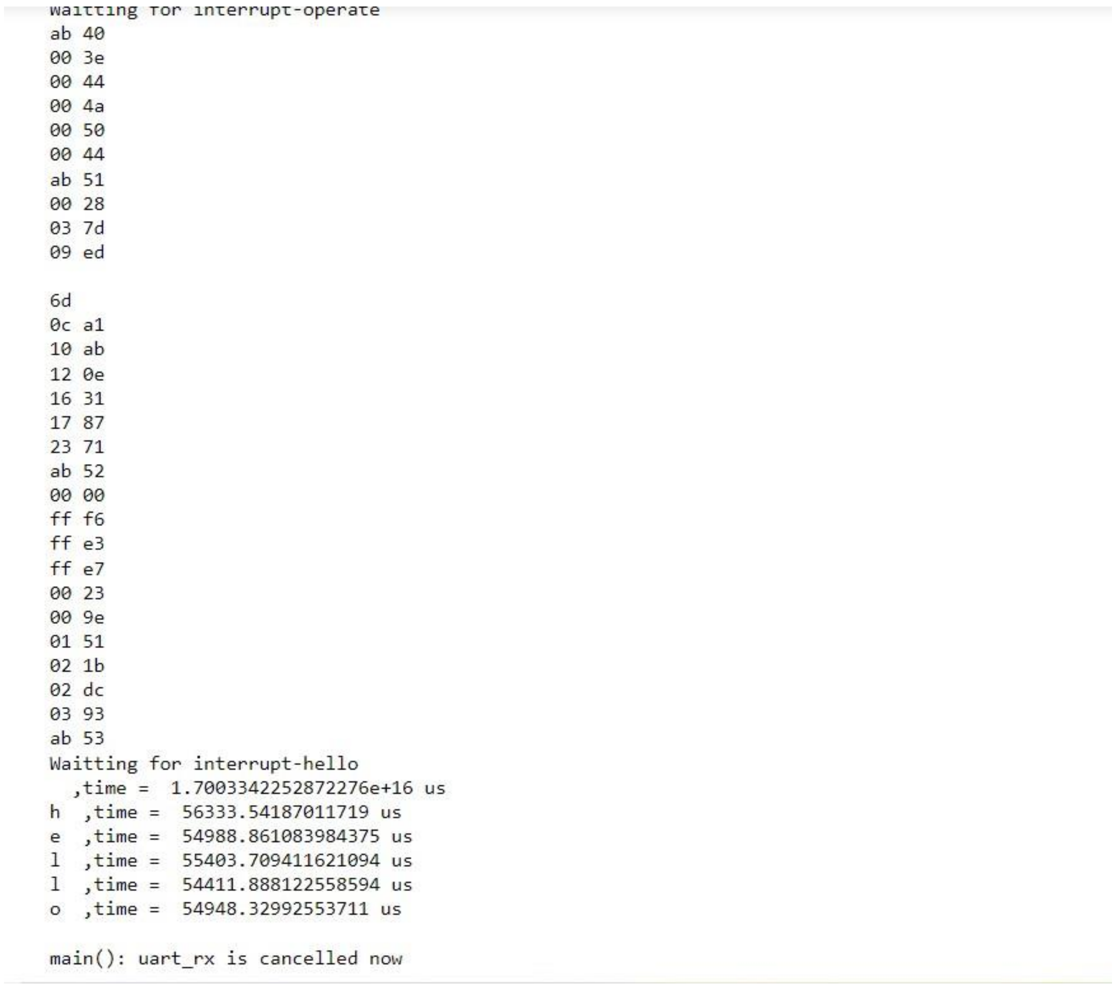


```
In [10]: asyncio.run(async_main())

Start Caravel Soc
Waitting for interrupt
  ,time =  1.7023675509984658e+16 us
h  ,time =  52099.22790527344 us
e  ,time =  67560.67276000977 us
l  ,time =  42128.562927246094 us
l  ,time =  43897.62878417969 us
o  ,time =  50561.42807006836 us

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 =  0xab530040
0x20 =  0x0
0x34 =  0x20
0x38 =  0x3f
```



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

checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0x2dc0040
checkbits =  0xab530040
  ,time =  1.7023676491886058e+16 us
h  ,time =  95908.64181518555 us
e  ,time =  38897.99118041992 us
l  ,time =  43687.82043457031 us
l  ,time =  65224.17068481445 us
o  ,time =  39620.399475097656 us

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 =  0xab530040
0x20 =  0x0
0x34 =  0x20
0x38 =  0x3f
```

上面兩張 jupyter notebook 運行結果圖，為使用降低運行速度方法所得到，所測出來 hello 中的各個 char 傳輸耗時，在這樣的驗證方式中，延遲時間從最低 39000us 到最高 90000us 都有，而平均延遲落在 5000us 附近。



```
Waitting for interrupt-operate
ab 40
00 3e
00 44
00 4a
00 50
00 44
ab 51
00 28
03 7d
09 ed

6d
0c a1
10 ab
12 0e
16 31
17 87
23 71
ab 52
00 00
ff f6
ff e3
ff e7
00 23
00 9e
01 51
02 1b
02 dc
03 93
ab 53
Waitting for interrupt-hello
  ,time =  1.7003342252872276e+16 us
h  ,time =  56333.54187011719 us
e  ,time =  54988.861083984375 us
l  ,time =  55403.709411621094 us
l  ,time =  54411.888122558594 us
o  ,time =  54948.32992553711 us

main(): uart_rx is cancelled now
```

相較於降低運行速度方法，上圖為在 isr 中使用兩個 uart 傳遞兩組 8bit 的 mprj 訊號，這樣的方法會比降速方法稍微增加 loop back latency 的平均值,但卻減少延遲發生極值的狀況,而且平均延遲落在 55000us 附近,對比降速法能得到更穩定的延遲，我們在 isr 中加入讀取 checkbit 的工作來執行，理論上會使得 loop back latency 增加，而觀察結果也顯示了最低延遲時間和較低的平均延遲，是發生在前面方法中，因此也能得證在 isr 中加入工作會讓其執行時間增加，進而對 loop back latency 產生影響。

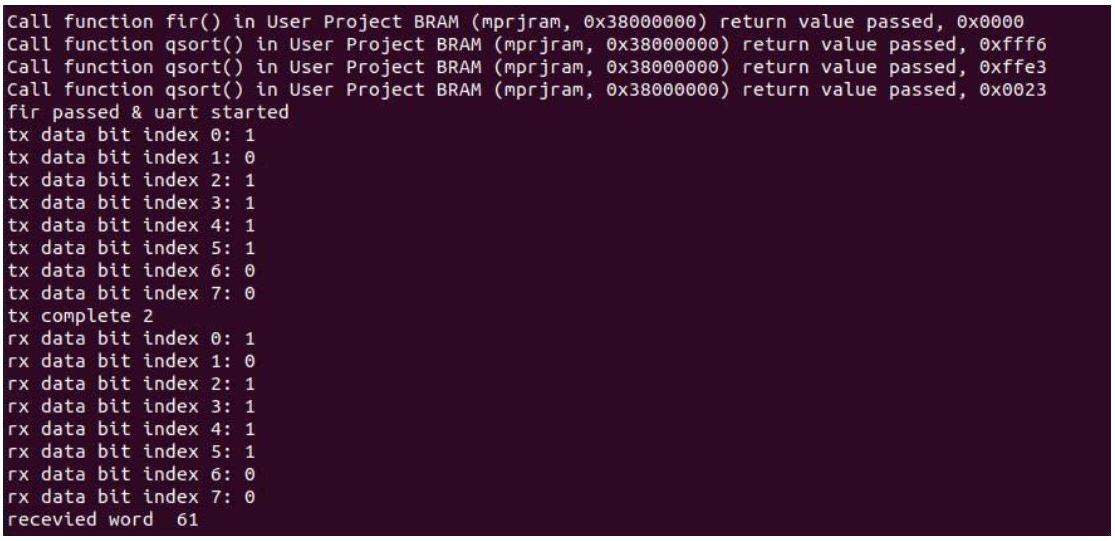
五、Suggestion for improving latency for UART loop back

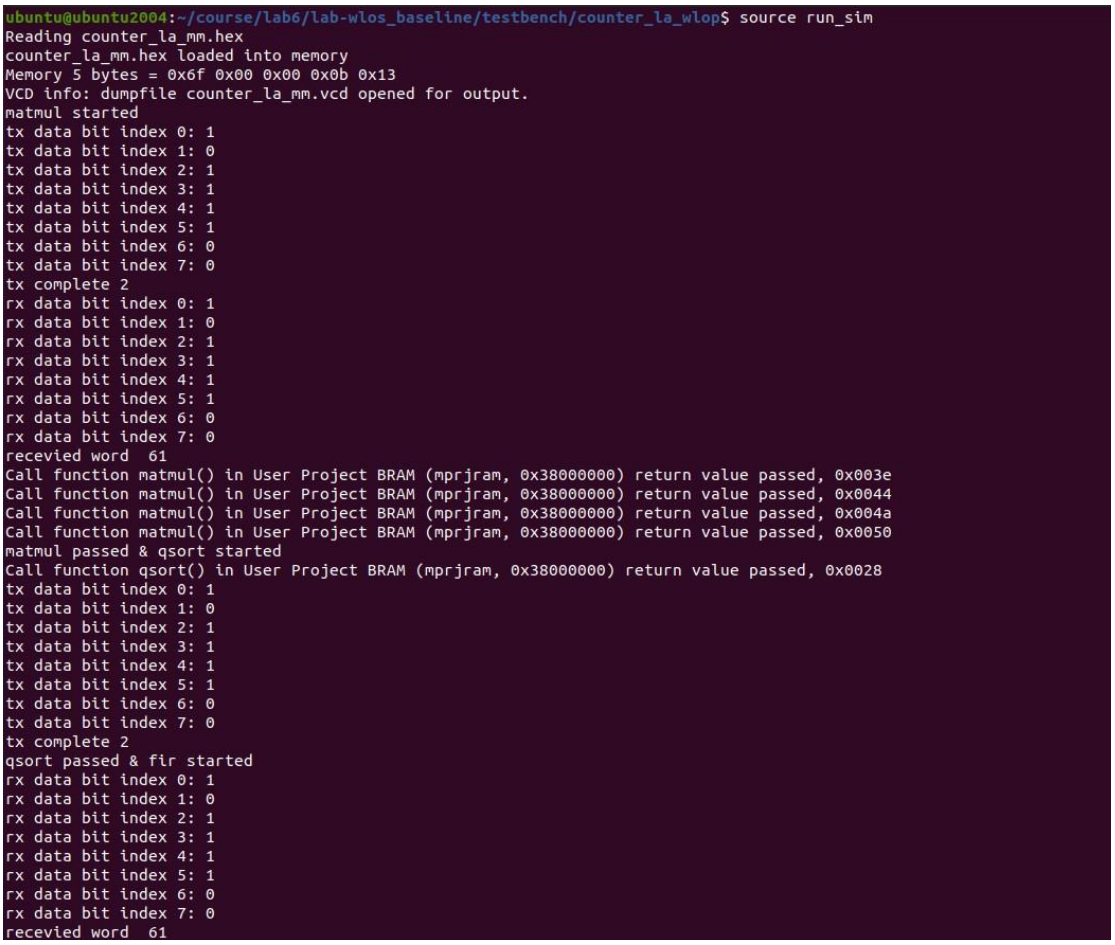
在改善 uart loop back latency 方面，應該能夠透過兩種層面來優化延遲，首先在軟體層面中，如果能夠在數據到達時立即進入 ISR 程序，便能夠有效減少 cpu 對數據的輪詢次數，除了減輕 cpu 工作壓力外，也能使延遲因此降低。除了減少資料抵達延遲外，與 loop back latency 最直接相關的便是 ISR 內部需要執行的程式，應該盡量確保 ISR 中的程式碼是高效率的，其中執行的內容盡量精簡，就能最大程度減少處理中斷的來回延遲。

在硬體方面，最直接的方式是調高 uart 的時脈以增加每秒可傳輸的位元數，或是在 cpu 的資料接收端加上 FIFO，如此一來能夠控制 FIFO 在收到一定數量的數據後再執行中斷，能減少 cpu 被中斷的次數，也能使得 cpu 利用更大的 block 一次性處理更多數據，在降低延遲方面，由於 FIFO 可以減少中斷發生的頻率和提高數據處理效率，也就代表著整體傳輸延遲降體，尤其當大量數據到達時所減少的延遲會更為明顯。

六、What else do you observe

在這次 lab6 中，首先要以模擬驗證 firmware 能正常運作，因此我們在 mprj 為特定值時會印出許多資訊來驗證他運行的狀況，首先是依序執行 mm、fir、qsort 等 firmware 確認功能正常，最後呼叫 uart 來傳遞測試 rx、tx 是否能正常傳遞資料。在來是將 uart 的傳遞放在穿插在其中，確保中斷後能回到原本運行的地方繼續工作下去，如第二張圖中所示，這樣的結果表示中斷後能夠回到原本的 firmware 繼續執行。

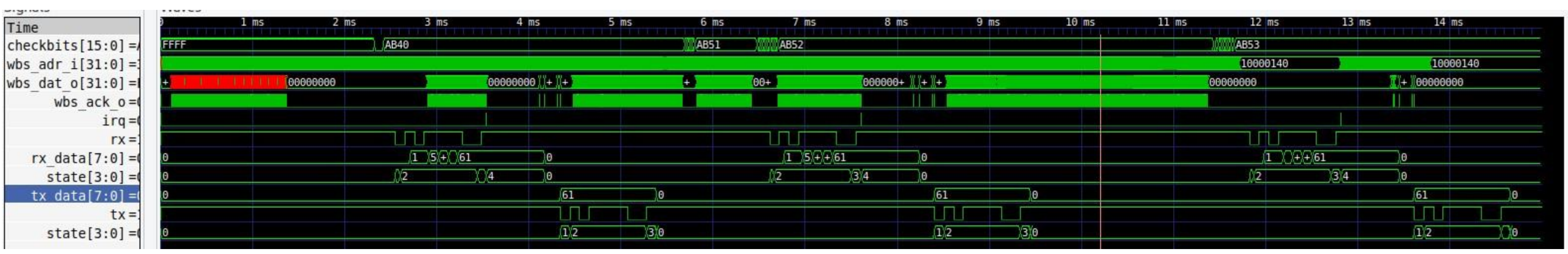

```
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0000
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, 0xfff6
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, 0xffe3
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0023
fir passed & uart 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
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
```
圖(Firmware 模擬依序驗證功能)



```
ubuntu@ubuntu2004:~/course/lab6/lab-wlos_baseline/testbench/counter_la_wlop$ 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.
matmul 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
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
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
matmul passed & qsort started
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0028
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
qsort passed & fir started
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
```

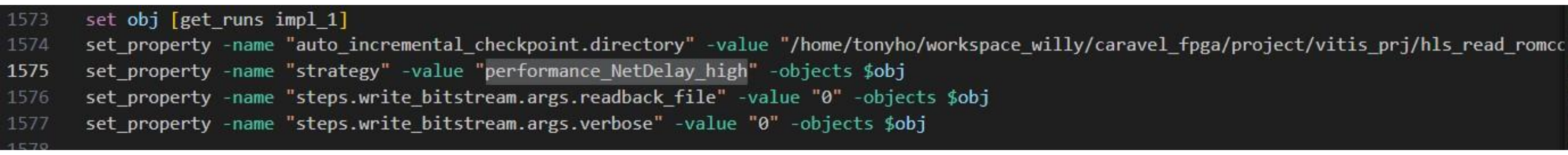
圖(Firmware 模擬中穿插 uart 傳遞)

在模擬跑出的波形裡，能夠仔細觀察到 uart 的運作方式，首先 61 這個訊號會在 rx 端被分成 1bit 進行傳遞到 cpu，接著 cpu 就能按照 state 的步驟將收到的 bit 組合完整的 rx_data，而 tx 端也相反的將資料拆分成 1bit 進行傳輸，除了從模擬結果驗證 uart 功能，觀察波形更能深入理解其運作原理。



圖(uart 模擬波形)

在合成 hex 檔時我們遇到了 timing violation 的問題，從 timing report 中觀察到總會有負 slack 的狀況發生，但若直接將出現 timing violation 的.hex 燒錄到 fpga 上，卻也能正常運作，盡管能正常運作，但是若硬體不滿足 timing request，仍然是錯誤的設計。我們後來在 vvd_caravel_fpga_40mhz.tcl 中修改合成時的 property，把”Vivado Synthesis Default” 改成”performance netdelay_high” 便能有效解決 timing 上的問題。



```
set obj [get_runs impl_1]
set_property -name "auto_incremental_checkpoint.directory" -value "/home/tonyho/workspace_willy/caravel_fpga/project/vitis_prj/hls_read_romcc
set_property -name "strategy" -value "performance_NetDelay_high" -objects $obj
set_property -name "steps.write_bitstream.args.readback_file" -value "0" -objects $obj
set_property -name "steps.write_bitstream.args.verbose" -value "0" -objects $obj
```