Lab6

https://github.com/v00000v/Soc-Design-Laboratory/tree/main/lab-wlos_baseline

▼ Team members

羅崇榮 R10522526

陳靖雯 R12922a10

張景華 B09505021

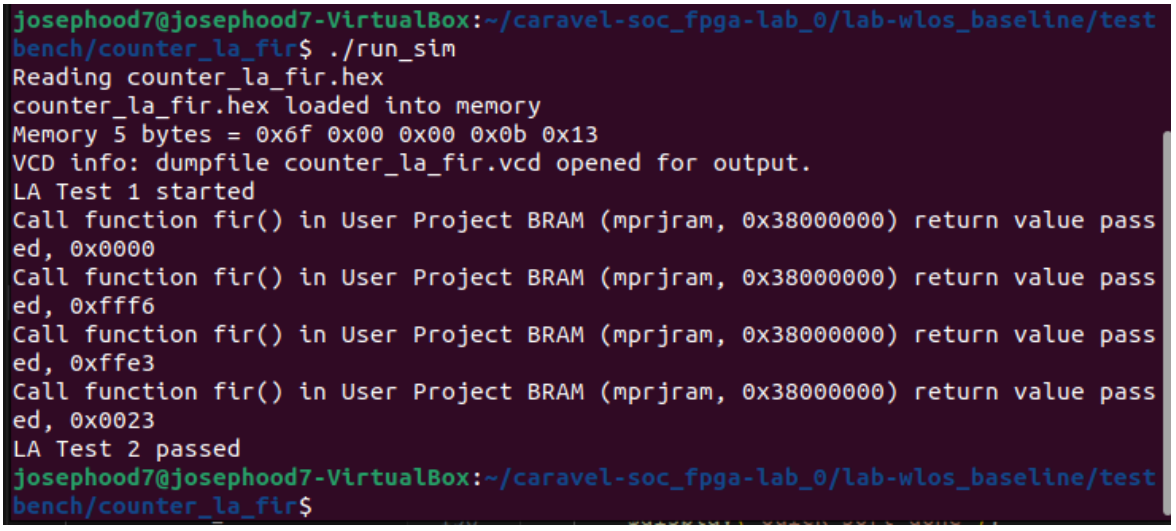
祝華劭 R12945050

▼ firmware combination observation

Initially, we try to wait AB51 for every completion of each firmware function. Soon we found that the *send_data_2* does not end until *tx_busy = 0*. So by the time the test bench runs to wait(AB51), AB51 is already sent before the *send_data_2* ends. To avoid firmware halting, we shall put uart at the end of the operation so that when uart ends, the program ends.

▼ firmware combination log

fir



matrix multiplication

```
josephood7@josephood7-VirtualBox:~/caravel-soc_fpga-lab_0/lab-wlos_baseline/test
bench/counter_la_mm$ ./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 p
assed, 0x003e
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value p
assed, 0x0044
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value p
assed, 0x004a
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value p
assed, 0x0050
LA Test 2 passed
josephood7@josephood7-VirtualBox:~/caravel-soc_fpga-lab_0/lab-wlos_baseline/test
bench/counter_la_mm$
```

quick sort

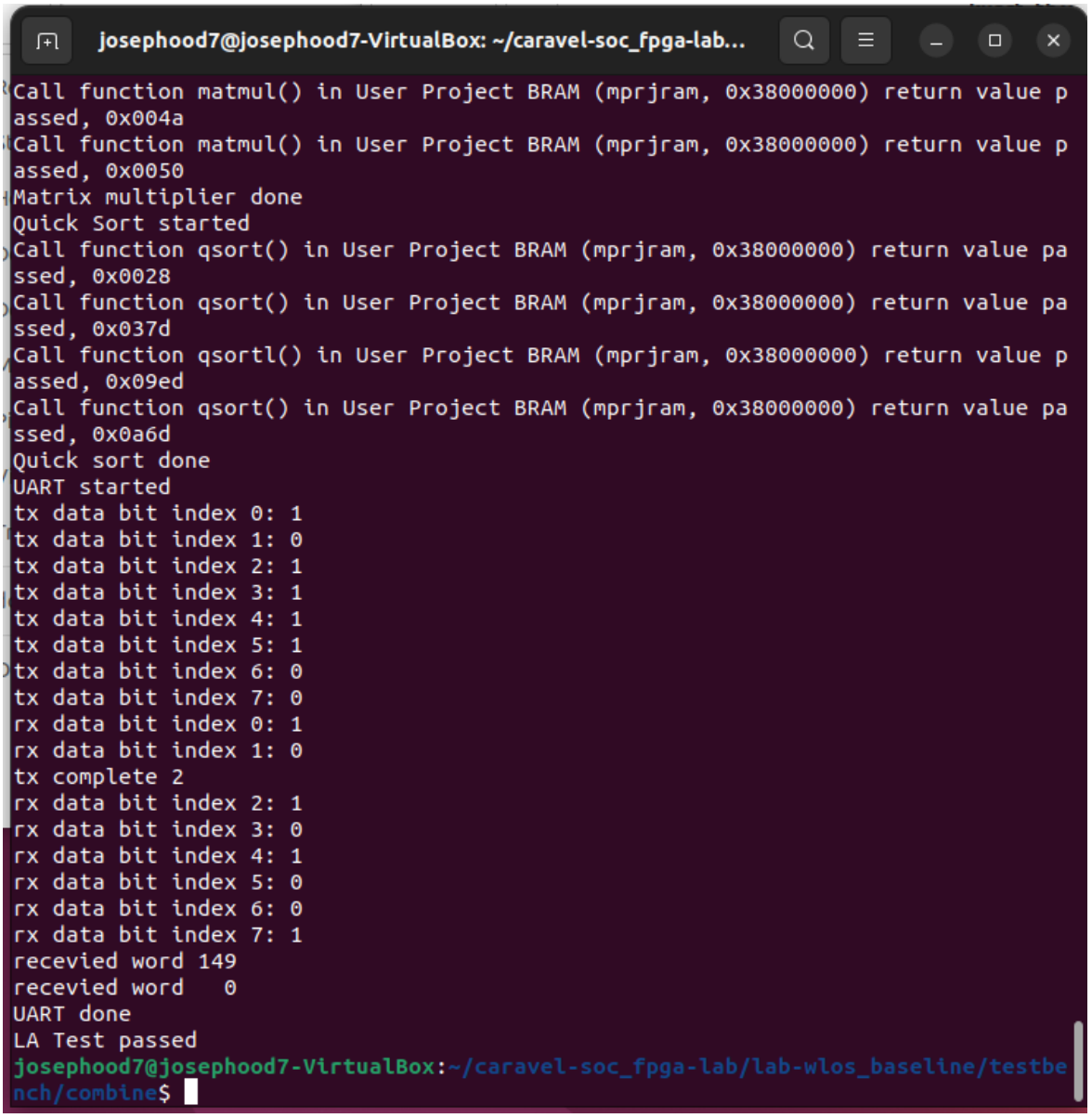
```
josephood7@josephood7-VirtualBox:~/caravel-soc_fpga-lab_0/lab-wlos_baseline/test
bench/counter_la_qs$ ./run_sim
Reading counter_la_qs.hex
counter_la_qs.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_qs.vcd opened for output.
LA Test 1 started
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value pa
ssed, 0x0028
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value pa
ssed, 0x037d
Call function qsortl() in User Project BRAM (mprjram, 0x38000000) return value p
assed, 0x09ed
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value pa
ssed, 0x0a6d
LA Test 2 passed
josephood7@josephood7-VirtualBox:~/caravel-soc_fpga-lab_0/lab-wlos_baseline/test
bench/counter_la_qs$ SS
```

uart

```
josephood7@josephood7-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbe
nch/uart$ ./run_sim
Reading uart.hex
uart.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile uart.vcd opened for output.
LA Test 1 started
tx data bit index 0: 1
tx data bit index 1: 0
tx data bit index 2: 1
tx data bit index 3: 1
tx data bit index 4: 1
tx data bit index 5: 1
tx data bit index 6: 0
tx data bit index 7: 0
tx complete 2
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
LA Test 1 passed
josephood7@josephood7-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbe
nch/uart$
```

combine

```
josephood7@josephood7-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbe
nch/combine$ ./run_clean
josephood7@josephood7-VirtualBox:~/caravel-soc_fpga-lab/lab-wlos_baseline/testbe
nch/combine$ ./run_sim
Reading kuart.hex
kuart.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile uart.vcd opened for output.
FIR started
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value pass
ed, 0x0000
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value pass
ed, 0xfff6
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value pass
ed, 0xffe3
Call function fir() in User Project BRAM (mprjram, 0x38000000) return value pass
ed, 0x0023
FIR done
Matrix multiplier started
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value p
assed, 0x003e
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value p
assed, 0x0044
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value p
assed, 0x004a
Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value p
assed, 0x0050
Matrix multiplier done
Quick Sort started
Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value pa
ssed, 0x0028
```



▼ How do you verify your answer from notebook

To meet the testbench, we need to send a “hello\n” from firmware. *uart_write_string(”hello\n”)* should be added to the la.c file

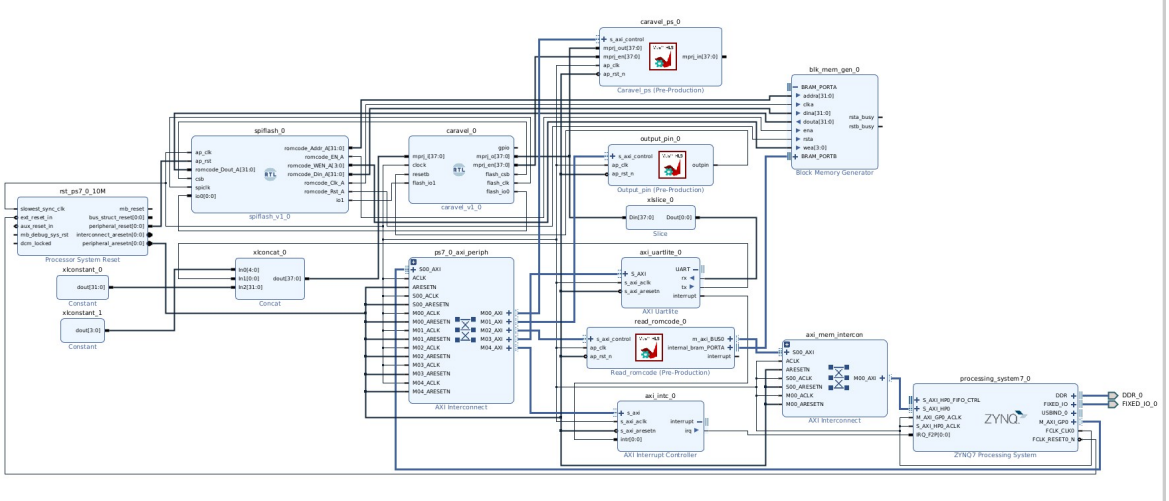
```
reg_la1_data = 0x00000000;

// Configure LA probes from [63:32] as inputs to disable counter write
reg_la1_oenb = reg_la1_iena = 0x00000000;
uart_write_string("hello\n");
//uart_write(1);
```

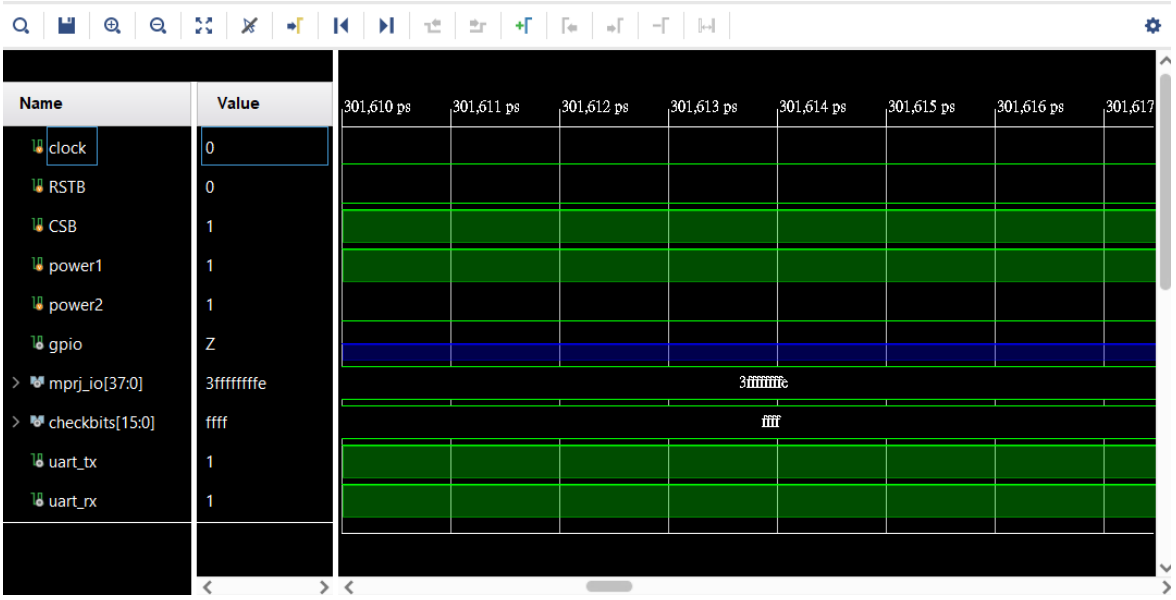
```
async def fir():
    while((ipPS.read(0x1c) & 0xffff0000) != 0xFFF60000):
        continue
    print("Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed, 0xFFF6")
    while((ipPS.read(0x1c) & 0xffff0000) != 0xFFE30000):
```

```
        continue
    print("Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed, 0xFFE3")
    while((ipPS.read(0x1c) & 0xffff0000) != 0x00230000):
        continue
    print("Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0023")
    while((ipPS.read(0x1c) & 0xffff0000) != 0xAB510000):
        continue
    print("Call function fir() in User Project BRAM (mprjram, 0x38000000) return value passed, 0xAB51")
async def mm():
    while((ipPS.read(0x1c) & 0xffff0000) != 0x003E0000):
        continue
    print("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x003E")
    while((ipPS.read(0x1c) & 0xffff0000) != 0x00440000):
        continue
    print("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0044")
    while((ipPS.read(0x1c) & 0xffff0000) != 0x004A0000):
        continue
    print("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x004A")
    while((ipPS.read(0x1c) & 0xffff0000) != 0x00500000):
        continue
    print("Call function matmul() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0050")
async def qs():
    while((ipPS.read(0x1c) & 0xffff0000) != 0x00280000):
        continue
    print("Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0028")
    while((ipPS.read(0x1c) & 0xffff0000) != 0x037D0000):
        continue
    print("Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x037D")
    while((ipPS.read(0x1c) & 0xffff0000) != 0x09ED0000):
        continue
    print("Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x09ED")
    while((ipPS.read(0x1c) & 0xffff0000) != 0x0A6D0000):
        continue
    print("Call function qsort() in User Project BRAM (mprjram, 0x38000000) return value passed, 0x0A6D")
```

▼ Block design



▼ Timing report/ resource report after synthesis



Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	1.327 ns	Worst Hold Slack (WHS):	0.035 ns	Worst Pulse Width Slack (WPWS):	6.250 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:	6762	Total Number of Endpoints:	6762	Total Number of Endpoints:	2804

All user specified timing constraints are met.

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	3994	0	0	53200	7.51
LUT as Logic	3940	0	0	53200	7.41
LUT as Memory	54	0	0	17400	0.31
LUT as Distributed RAM	16	0			
LUT as Shift Register	38	0			
Slice Registers	4131	0	0	106400	3.88
Register as Flip Flop	4026	0	0	106400	3.78
Register as Latch	105	0	0	106400	0.10
F7 Muxes	164	0	0	26600	0.62
F8 Muxes	44	0	0	13300	0.33

```
INFO: [Project 1-111] Unisim Transformation Summary:
  A total of 147 instances were transformed.
  IOBUF => IOBUF (IBUF, OBUFT): 38 instances
  LD => LDCE (inverted pins: G): 30 instances
  LDC => LDCE: 75 instances
  RAM32M => RAM32M (RAMD32(x6), RAMS32(x2)): 4 instances

Synth Design complete, checksum: bf8a757b
INFO: [Common 17-83] Releasing license: Synthesis
162 Infos, 389 Warnings, 1 Critical Warnings and 0 Errors encountered.
synth_design completed successfully
synth_design: Time (s): cpu = 00:01:59 ; elapsed = 00:02:16 . Memory (MB): peak = 2092.918 ; gain = 455.895
INFO: [runtcl-6] Synthesis results are not added to the cache due to CRITICAL_WARNING
INFO: [Common 17-1381] The checkpoint 'C:/Users/User/Desktop/SoC/SoC_lab/caravel-soc_fpga-lab-main/lab-wlos_baseline/project_1/project_1.runs/synth_1/caravel.dcp' ha
INFO: [runtcl-4] Executing : report_utilization -file caravel_utilization_synth.rpt -pb caravel_utilization_synth.pb
INFO: [Common 17-206] Exiting Vivado at Sat Dec  9 22:15:08 2023...
```

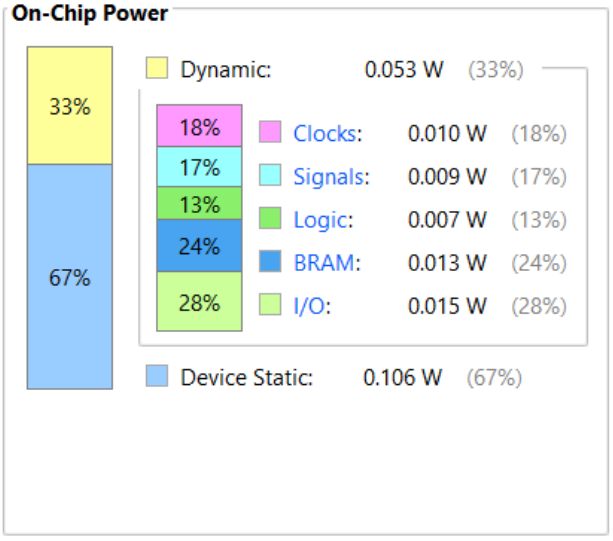
```
INFO: [Project 1-853] Binary constraint restore complete.
Reading XDEF placement.
Reading placer database...
Reading XDEF routing.
Read XDEF Files: Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.486 . Memory (MB): peak = 2535.742 ; gain = 0.000
Restored from archive | CPU: 0.000000 secs | Memory: 0.000000 MB |
Finished XDEF File Restore: Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.487 . Memory (MB): peak = 2535.742 ; gain = 0.000
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.002 . Memory (MB): peak = 2535.742 ; gain = 0.000
INFO: [Project 1-111] Unisim Transformation Summary:
  A total of 42 instances were transformed.
  IOBUF => IOBUF (IBUF, OBUFT): 38 instances
  RAM32M => RAM32M (RAMD32(x6), RAMS32(x2)): 4 instances

open_run: Time (s): cpu = 00:00:28 ; elapsed = 00:00:20 . Memory (MB): peak = 2691.117 ; gain = 1114.570
```

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.158 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**26.8°C**
Thermal Margin:	58.2°C (4.9 W)
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity



▼ Latency for a character loop back using UART

```
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]))
    UartStarts = time.time()
    i = 1
    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))
            UartEnds = time.time()
            if i<len(tx_str):
                ipUart.write(TX_FIFO, ord(tx_str[i]))
                i=i+1
        Latency = UartEnds - UartStarts
        print(buf, end='')
        print("Uart latency = ", Latency)

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

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

Start Caravel Soc
Waitting for interrupt
Uart latency =  0.018100500106811523
Uart latency =  0.018827438354492188
heUart latency =  0.02610468864440918
Uart latency =  0.027285337448120117
Uart latency =  0.027841806411743164
Uart latency =  0.028374433517456055
```

▼ Suggestion for improving latency for UART loopback

1. We suggest implementing FIR using hardware because it is faster to use AXI and calculate directly instead of using UART to transmit TX and RX to the firmware.
2. Secondly, we can use UART to communicate between hardware and firmware. In this case, we utilize the functions *uart_write* and *uart_read*.

▼ What else do you observe

- If concurrent assignment errors occur, replace `*default_nettype none* with `*default_nettype wire.*
- When we decode addresses 0x3000 and 0x 3800, ensure wires and registers are not assigned to the same resource simultaneously. I suggest using

```
if(address == 3800) begin
    io_out_r <= count;
end else if(address == 3000) begin
    io_out_r[6] <= tx;
end
```

Rather than

```
io_out_r <= (address == 3800)? count:((address == 3000)? tx:0);
```

Otherwise, an error might occur while running the implementation.

- The Vivado might not recognize the define.v, so we can copy-paste the definitions to the .v files directly.