

SOC lab4_2

Caravel_FIR_Design

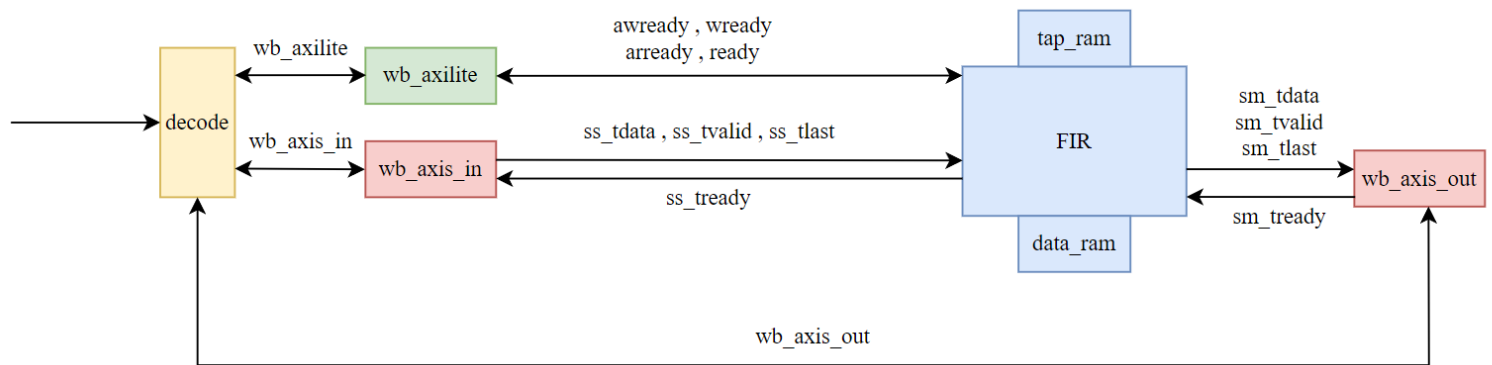
第五組

張耀明

陳佳詳

蕭翔

1. Design block diagram – datapath, control-path



2. The interface protocol between firmware, user project and testbench

Firmware code透過house keeping module傳入soc，系統透過wishbone將firmware code存入user bram。

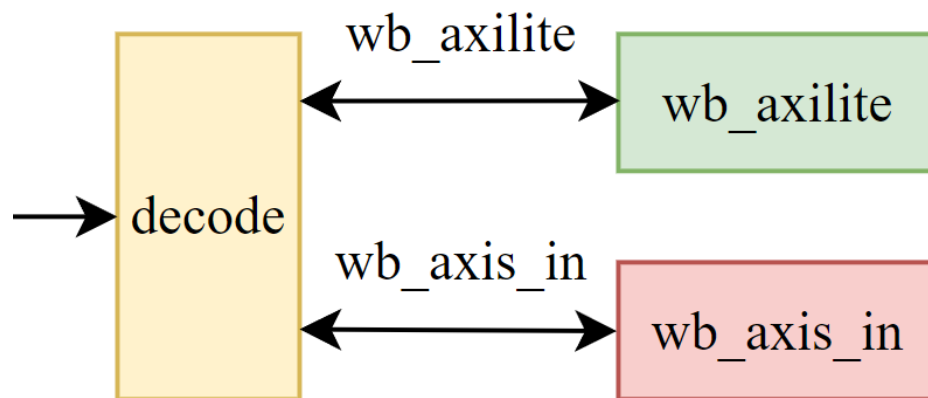
user project 和 testbench(external block)傳輸是使用 mprj_io[15:0]

首先往 0x30000010 送入 data length，再將 tap 一一傳入 0x300000 至 0x3000007F，傳完後，再透過 wb 讀出 0x30000040 至 0x3000007F 的值，並一一透過 mprj_io[31:15] 傳回 testbench 確認值是否正確。

往 mprj_io [31:15] 送 0x00A50000，去通知 testbench 開始 latency timer，並讀取 0x30000000 來看是否為 idle，若為 idle，則透過 wb 傳 ap_start 給 0x30000000，接著開始傳 X 的值進去，透過讀取 0x30000088 的值確認能否繼續讀取資料，若可以則往 0x30000080 送也透過讀取 0x30000090 去判斷是否有算好的值，有的話，則從

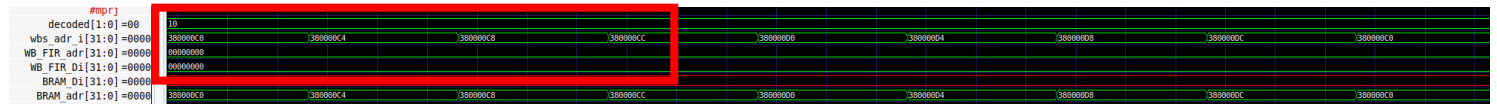
0x30000084 讀取資料，全部算好後，透過讀取 0x30000000 的 ap_done 判斷是結束運算，若結束了，則透過 mprj_io[31:15]傳送 0x005A0000 給 testbench，結束 latency timer，這個動作重複三次。

3. Waveform and analysis of the hardware/software behavior.

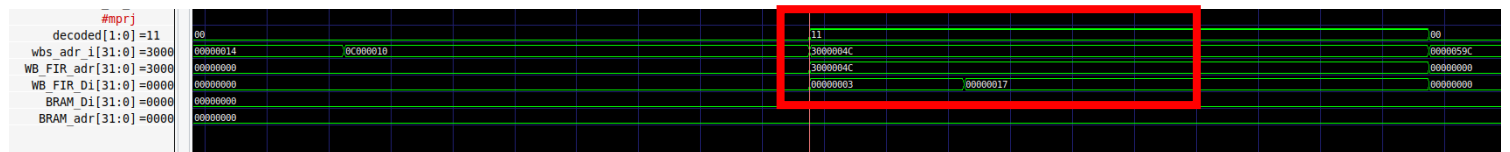


Decode select

12'h380 for BRAM and decoded = 2'b10
12'h300 for FIR and decoded = 2'b11



當 wbs_addr 收到 0x380，使用 decoded=2 來去分流，讓 wbs_addr 傳入 bram_addr，且 FIR_addr 就不會收到資料。

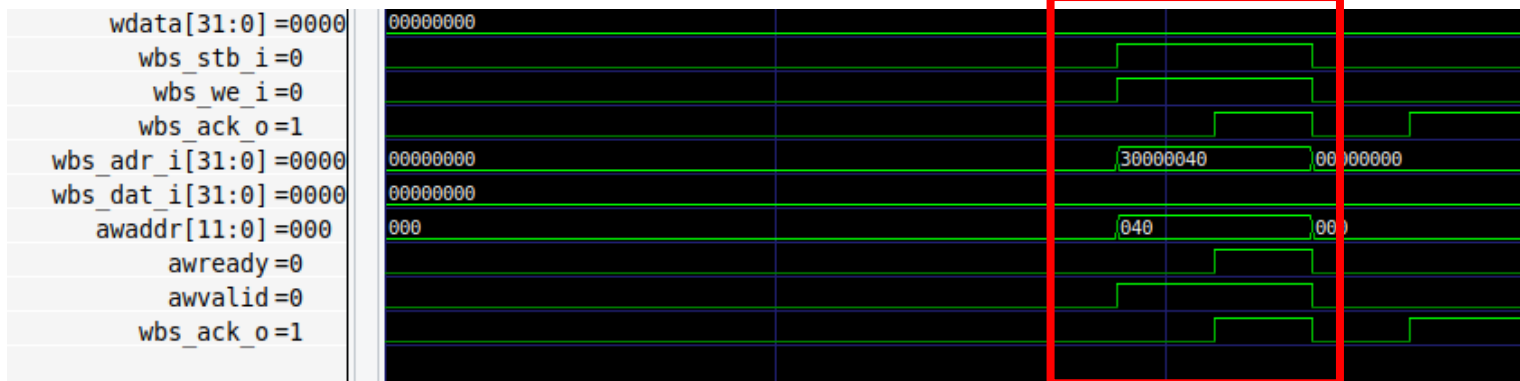
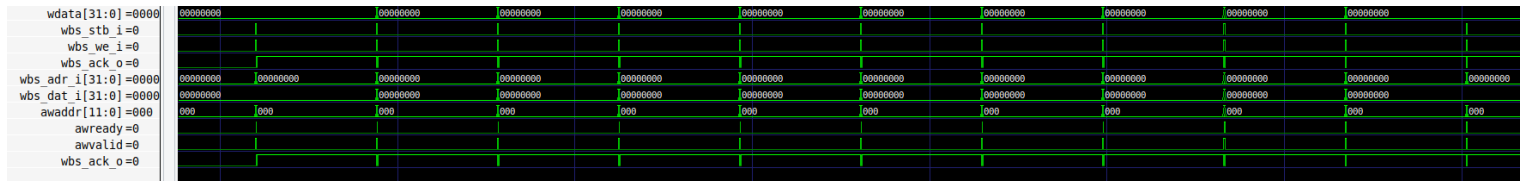


當 wbs_addr 收到 0x300，使用 decoded=3 來去分流，讓 wbs_addr 傳入 WB_FIR_addr，且 bram_addr 就不會收到資料。

Wb_axilite:

透過wb_we_i來區分是read or write

Write:



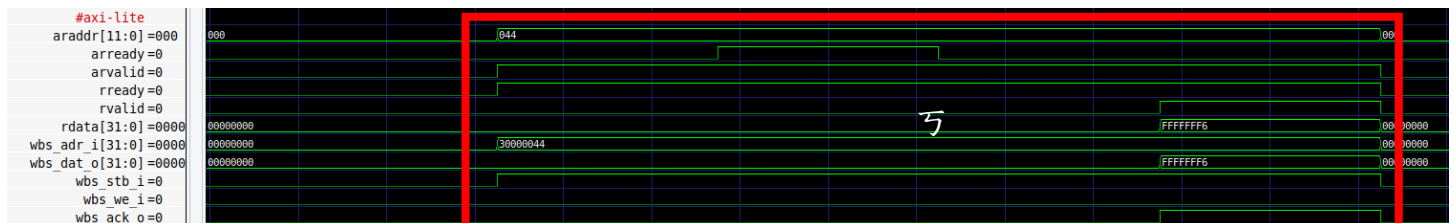
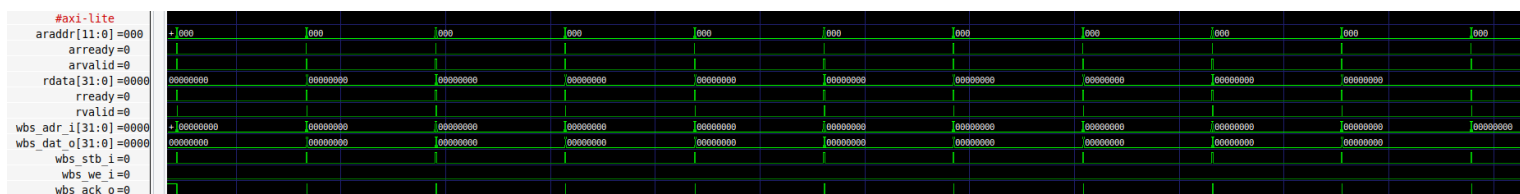
Wb_stb_i接awvalid跟wvalid，表示現在送給FIR的addr跟data是否有

效，wb_addr_i 0x30000000接awaddr去對上addr，wb_dat_i接wdata，

wb_ack_o接wready，表示已經收到wdata，可以再送下一筆，

wb_dat_o就寫個0。

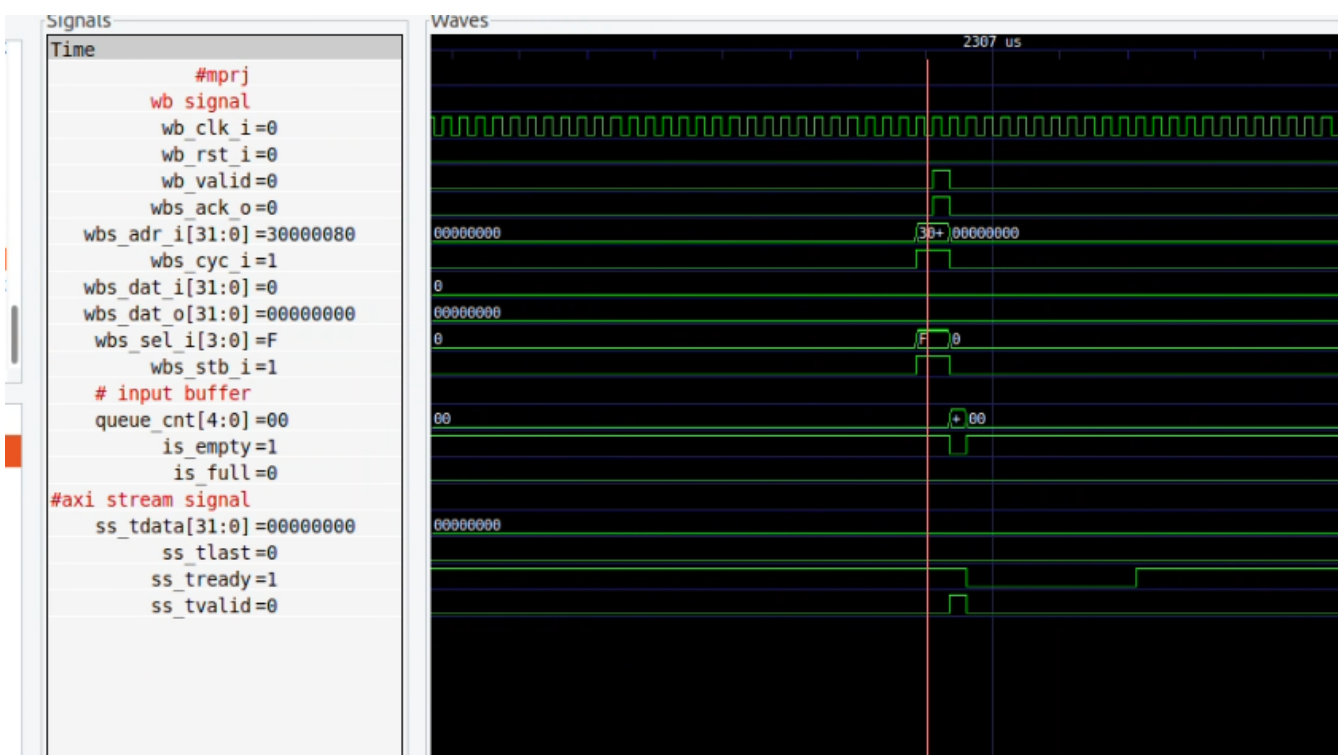
READ:



wb_stb_i接arvalid跟rready，表示現在送給FIR的addr是否有效，以及是否已經做好了讀資料的準備，wb_addr_i接awaddr去對上addr，rval id接wb_ack_o，表示收到了讀的要求並已經給好了rdata，wb_dat_o接rdata。

wb_axi_strm_in:

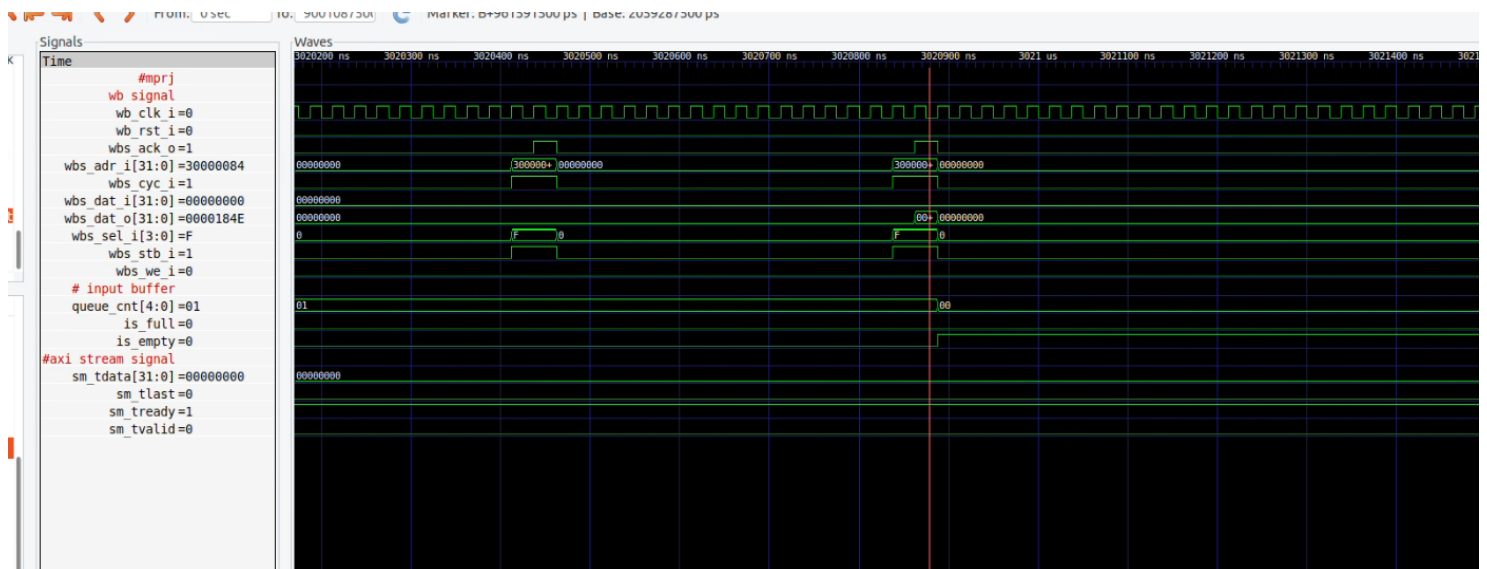
這個東西的設計理念是在中間放一個 buffer，wishbone 將資料寫入的時候將其暫存在 buffer 裡面。一旦 buffer 裡面有資料，ss_valid 就會設 1 並等 fir，module 將 ss_ready 拉起來收資料，並且最先進來的資料會被預備在 ss_tdata。一旦 ss_valid = ss_ready = 1，在 buffer 裡面最先進來的資料會被讀出去(first in first out)。



波形如上圖，由於 firmware 餵資料的速度太慢，所以基本上這個 buffer 沒有甚麼用。

Wb_axi_strm_out:

Wb_axi_strm_out 接收 fir 的輸出，並將其暫存在一個 buffer 裡面等待 wishbone 的讀取。當 buffer 為空的時候，wishbone 沒辦法從裡面取得資料。當 buffer 還有空間的時候，sm_tready 為 1，等待 sm_tvalid 被拉起來使自己可以收資料。當 buffer 裡面有資料的時候，他會響應 wishbone 的 read request，將 buffer 裡面最先進來的資料輸出出去。



**4.What is the FIR engine theoretical throughput, i.e. data rate?
Actually measured throughput?**

47264 cycle

738.5 cycle per data

```
ubuntu@ubuntu2004:~/caravel-soc_fpga-lab/SOC_lab_caravel_fir/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
checkbits is 765a
checkbits is 765a
latency = 47268 clock cycles
checkbits is 765a
LA Test 2 passed
```

5.What is latency for firmware to feed data?

從 firmware 下送資料的指令開始，到 fir 實際收到資料為止所經過的 cycle 數。對於韌體(CPU)來講，送/收 wishbone 訊號把 x 給 fir，僅僅是對某個記憶體位置的存取。在 C 語言的 firmware code 中，我們對指標做取值以便對記憶體做存取，把 data length=64 寫入 0x30000010 這個位置。這在硬體中代表將 64 寫到 fir 裡面。

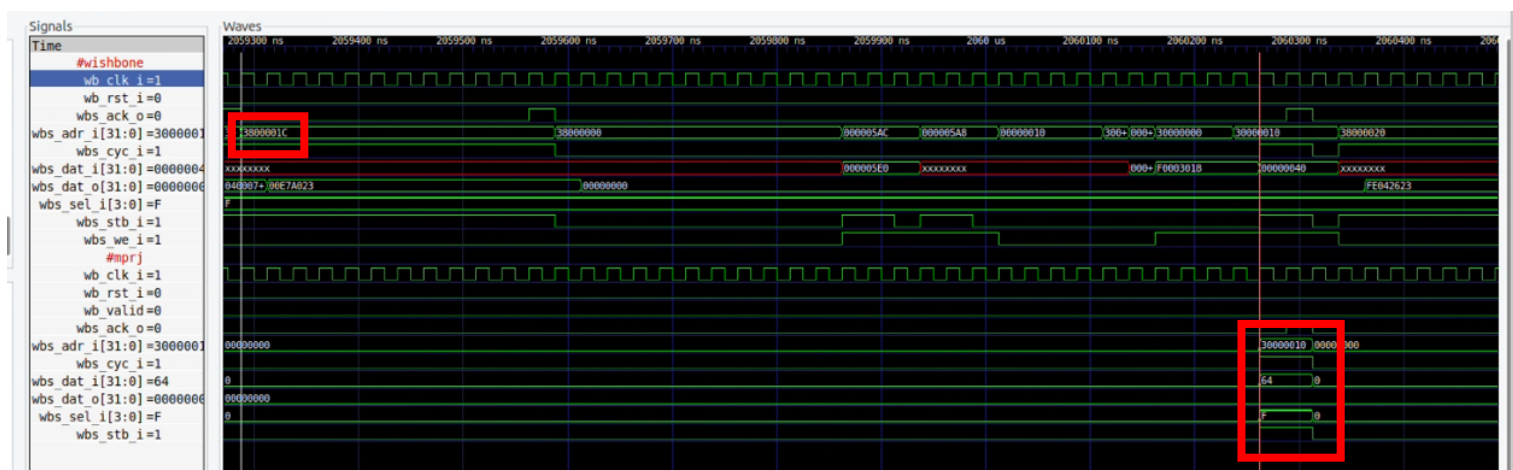
```
29     send_wb(mprj_datlen, data_len);
30     /*(volatile uint32*)(0x30000010) = 64;
```

而這段 code 在組合語言中會被編譯為以下的指令

```
38000010: 300007b7          lui    a5,0x30000
38000014: 01078793          addi   a5,a5,16 # 30000010 <_esram_rom+0x1ffff938> #把0x30000010這個地址寫入a5 register
38000018: 04000713          li     a4,64    # 把datalength寫入a4 register
3800001c: 00e7a023          sw     a4,0(a5) # 將 data length 寫入 0x30000010
```

其中，sw 這個指令一旦被執行，就會有一個 wishbone write 的訊號從 soc 被發送出來，夾帶著 64 這個值，寫入 user project 裡面。而這個指令所在的記憶體位置是 0x3800001c。

以下量測送 data length 所花的時間。與 lab3 不同的是，我們將 data length 送到 wb_axis 的 module，所以和送 input 的時間會一樣。



如上圖，從 wishbone addr 指到 03800001c 去讀取 sw 這個指令開始，到該指令寫入的值 64 經由 wishbone 往 0x30000010 這個 address 發出 write 請求並收到 ack 為止，總共經過了 $975\text{ns}/25\text{ns} = 39$ 個 cycle。這長度包含了 CPU 讀取 user bram 所花的時間。



6. What techniques used to improve the throughput?

在 output 加上一個長度為 2 的 buffer。因為 firmware 使用 polling 的方式讀取 output，每次讀取 output 的指令都會經過相當長的時間，導致 fir 算好一個輸出後可能要等一段時間才能把資料送出去。所以 output 放一個 buffer，讓 fir 只要有資料就可以先放進去，這樣一來 fir 就可以不用為了等 wb_read signal 而卡在那邊不能做下一筆運算。

7. Does bram12 give better performance, in what way?

沒有使用到Bram12。

8. Can you suggest other method to improve the performance?

使用 interrupt 的方式去讀取 Output，避免 polling 所帶來的時間延遲。當 output 可讀取的時候，透過發送一筆 interrupt request 給 soc，並在 interrupt service routine 裡面設計收 output data 的 code。