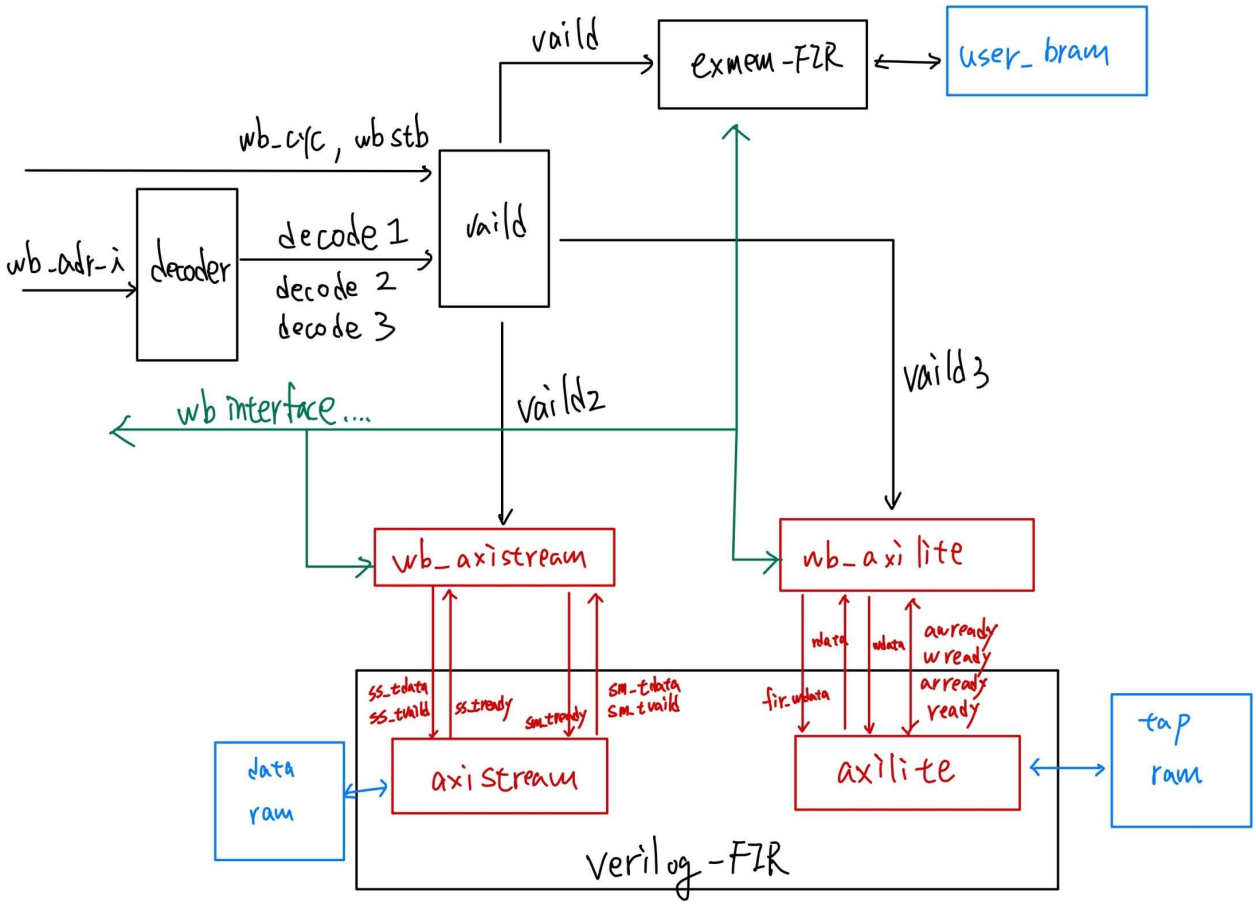
Lab 4-2 (Lab-FIR) Report

Name: 李柏辰/陳揚哲/楊正宇

● Block Diagram



● The interface protocol between firmware, user project and testbench

首先 Firmware code 是透過 house keeping module 傳入 soc，再經由 wishbone 將 firmware code 存入 user bram。

而 user project 和 testbench 之間是使用 mprj_io[15:0] 做傳輸，一開始往 0x30000010 送入 data length，再將 tap 逐一傳入 0x300000~ 0x3000007F，再透過 wishbone 讀取 0x30000040 至 0x3000007F 的值，並一一透過 mprj_io[31:16] 傳回 testbench 確認值是否正確。 CPU 往

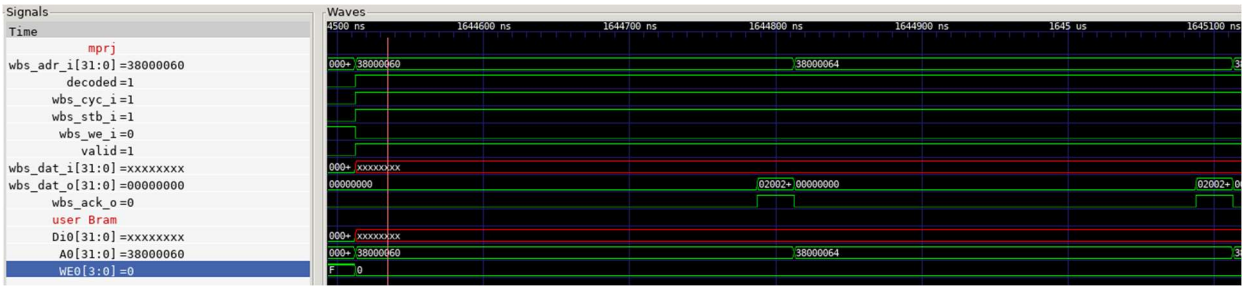
mprj_io [31:16] 送 0x00A50000 通知 testbench 開始 latency timer，而 testbench 會讀取 0x30000000 來看是否為 ap_idle，若為 ap_idle，則透過 wishbone 傳 ap_start 給 0x30000000，接著開始傳 X data 的值給 FIR，透過讀取 0x30000088 的值確認能否繼續讀取資料，若可以則往 0x30000080 送 也透過讀取 0x30000090 來判斷是否有算好的值，有的話，則從 0x30000084 讀取資料，全部算好後，透過讀取 0x30000000 的 ap_done 判斷是結束運算，若結束了，則透過 mprj_io[31:16]傳送 0x005A0000 給 testbench，結束 latency timer ，以上動作重複三次。

- **Waveform and analysis of the hardware/software behavior.**
  - decoded 為判讀 wbs_adr_i[31:24]是否為 3800，是的話同時與 wbs_cyc_i && wbs_stb_i 來 assign vaild 訊號，此 vaild 訊號是給 firmware 端的 wishbone interface 進行傳輸。
  - decoded2 則判讀 wbs_adr_i[31:24]是否為 30000080 或 30000084，接著 assign vaild2 訊號是給 wb_axi_stream 進行 data 的傳輸
  - decoded3 是判讀 wbs_adr_i[31:24]是否為 30000000 到 3000007F 的範圍，接著 assign vaild3 訊號是給 wb_axi_lite 進行 tap 的傳輸

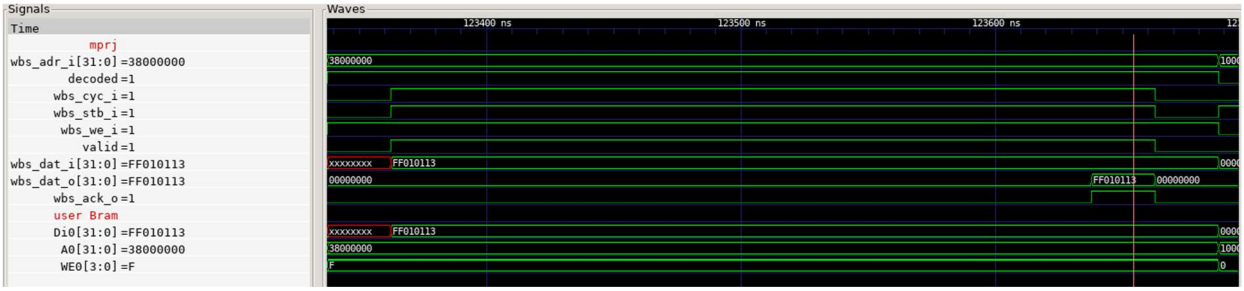
```
assign decoded  = wbs_adr_i[31:24] == 8'h38 ? 1'b1: 1'b0;
assign decoded2 = (wbs_adr_i[31:0] == 32'h30000080 | wbs_adr_i[31:0] == 32'h30000084) ? 1'b1: 1'b0;
assign decoded3 = (wbs_adr_i[31:0] >= 32'h30000000 & wbs_adr_i[31:0] <= 32'h3000007F) ? 1'b1: 1'b0;
```

```
assign valid  = wbs_cyc_i && wbs_stb_i && decoded;
assign valid2 =  wbs_cyc_i && wbs_stb_i && decoded2;
assign valid3 =  wbs_cyc_i && wbs_stb_i && decoded3;
```

firmware read:

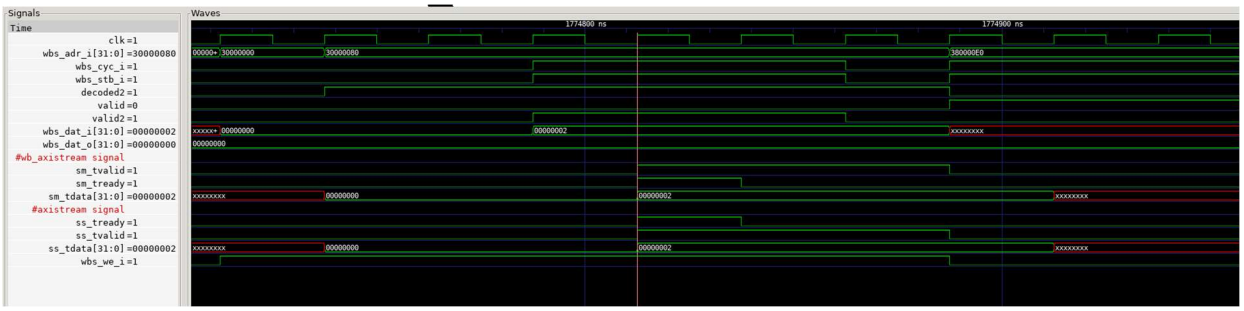


firmware write:

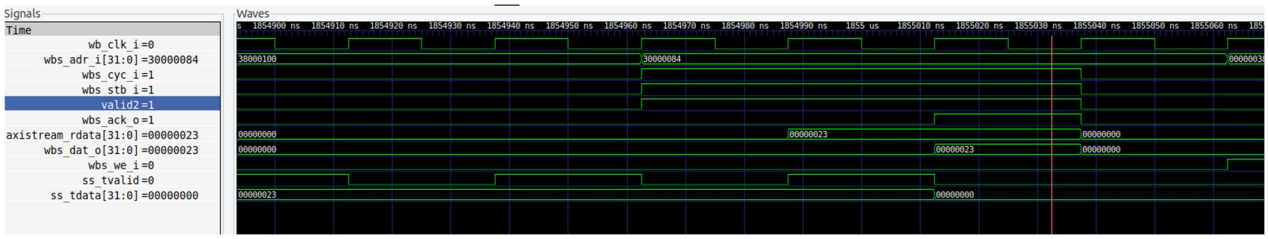


- wb_axistream/axilite 會將 wb 的訊號轉成 axi 的控制訊號來控制 wb 的 data 做寫入及讀出

transmit data from wb_axistream to fir



recive data from fir to wb_axistream



transmit tap from wb_axilite to fir

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

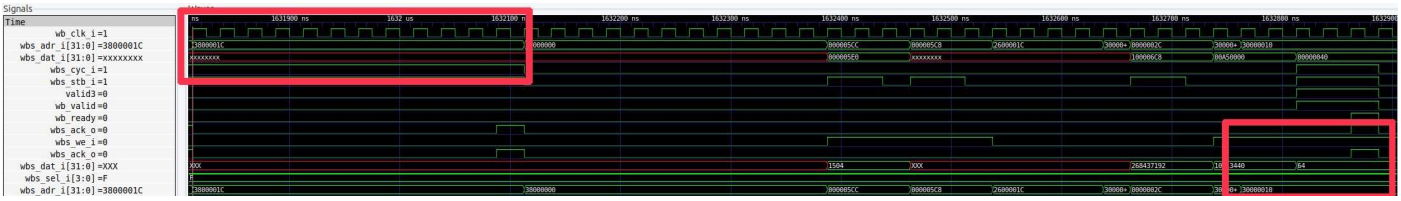
  測量從 checkbits[7:0]從 A5 到 5A 所需的 clock cycle
  測出來的 lantency = 46261 cycles
  每筆 data 所需時間 (46261/64)=722.8

```
LA Test 1 started
lantency =      46261 clock cycles
```

- **What is latency for firmware to feed data?**
  從 firmware 發起指令到 fir 收到 data 時所經過的時間，從下圖組語可以看到在 0x3800001c 的位置會執行 sw 指令將 data length 寫進 0x30000010。

```
531   38000014: 01078793          addi  a5,a5,16 # 30000010 <_esram_rom+0x1ffff938>
532   38000018: 02c02703          lw  a4,44(zero) # 2c <data_length>
533   3800001c: 00e7a023          sw  a4,0(a5)
```

  由下圖可看出，Wishbone addr 指到 3800001c 讀取 firmwave 指令開始，表示要寫值 64(data length)到 30000010 直到 ack_o 拉起表示收到值，共經過 1050ns，為 42 cycles



- **What techniques used to improve the throughput?**
  1. 增加硬體資源，使用多個乘加器
  2. FIR 算完直接用 wb 輸出
- **Does bram12 give better performance, in what way?**
  可能不會，因只有各一個乘法與加法的限制還是必須花 11 個 cycles 進行計算，使用 bram12 對此設計並不會使運算更快。

- **Can you suggest other method to improve the performance?**
  若是 CPU 與 user 之間不共用 wishbone interface，而 FIR 輸出資

料時就不需要再等好幾個 cycles 的 handshake protocol，或是 FIR 直接使用 wishbone 做輸出，但就需要改變原先的 FIR IP 設計。

- **Github Link**

https://github.com/ZheChen-Bill/SoC_Design_Lab4_2.git