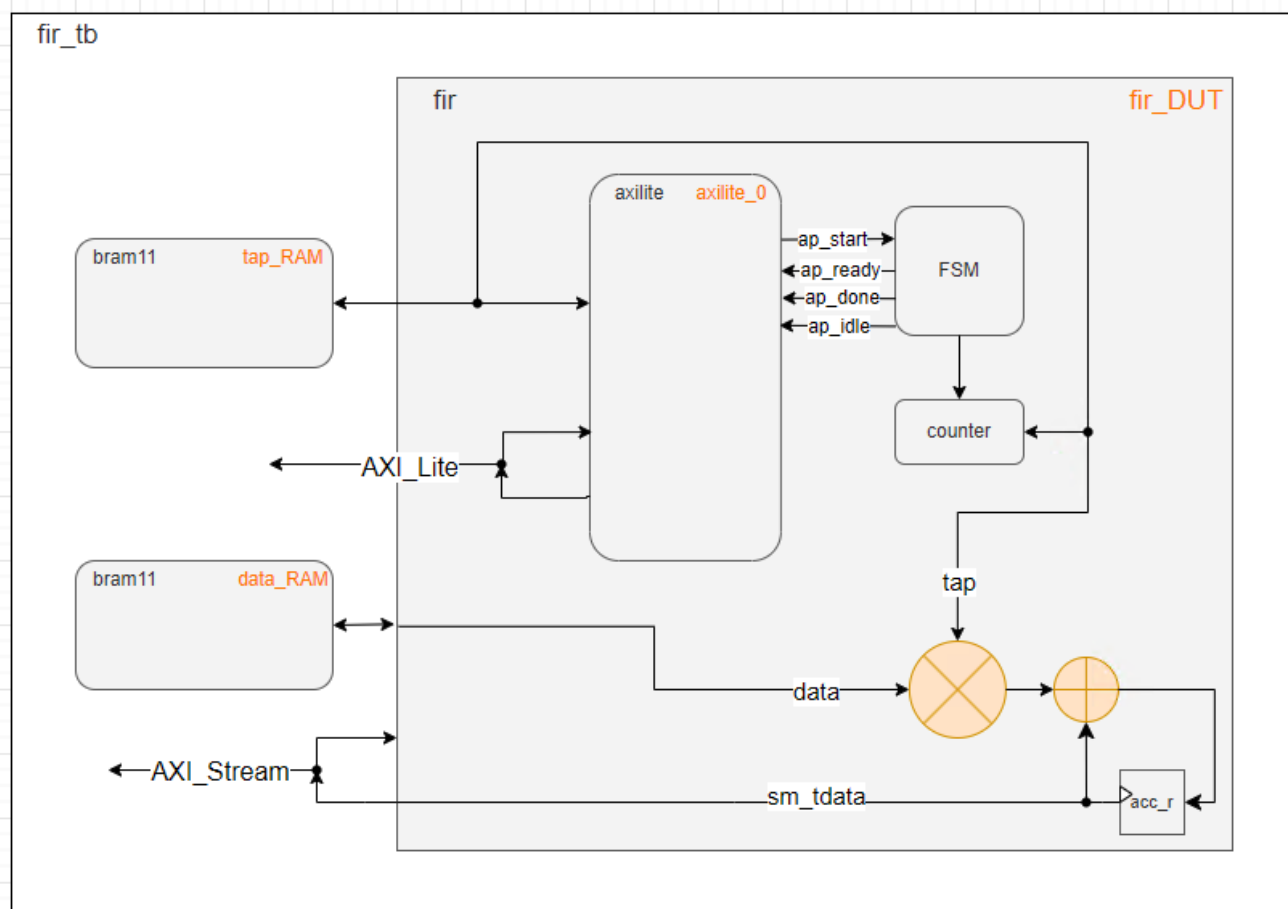


SOC Design Lab - Lab3 Report

R11943022 范詠為

1. Block Diagram



- 我設計了一個 axilite module 負責控制 AXI-Lite 的輸入輸出像是 ap_start、data_len、tap_coeff 等信號，在第二點會針對它做介紹。而 datapath 的實現是用一個 32b x 32b 的乘法器和一個 32b 的加法器，透過 address scheduling 去得到相對應要做乘加的值，並用一個 32b acc_r register 作為暫存和運算完後的輸出資料，data 的輸入輸出則是遵守 AXI-Stream 的規範。

2. Describe Operation

- FSM
 - 我分成五個 state，S_IDLE、S_LOAD、S_CALC、S_SEND、S_DONE。S_IDLE 為初始狀態，等待 ap_start 來進入 S_LOAD。S_LOAD 收進一筆 streaming 進來的資料並存到 data bram。S_CALC 會花 12 cycles 做單筆 y[n] 的運算，算出來結果後到 S_SEND streaming 一筆出去。若是最後一筆 (ss_tlast) 則進入 S_DONE 並把 ap_done 拉高，否則進入 S_LOAD 繼續運算資料。
- axilite AXI-Lite controller
 - 我參考 lab2 vitis hls 合出來的 rtl 寫出的，read/write 分別有自己的 state machine，分成 IDLE 跟 DATA。前者做 address 的 handshake，後者則是 data 的 handshake。在 AXI-Lite 寫進 fir 時，會 decode address 去存 ap_start、data_len 或 tap_coeff。反之在讀取時會去拿 ctrl (ap_start、ap_idle、ap_done) 或是 tap bram 內的暫存值並輸出。
- AXI-Stream
 - ss_tready 在 state_r == S_LOAD 時拉高，sm_tvalid 則在 state_r == S_SEND 時拉高。在輸入 handshake 成功時從 ss_tdata 寫入一筆 data bram，運算後從 acc_r 輸出至 sm_tdata。
 - Address 控制邏輯
 - 因為 bram 是 single port 不能同時讀取寫入，在設計時需要規劃好每個 cycle 要讀取或寫入，稍微 tricky。我的做法是外面進來的 data 先寫入 bram 0x00，接著進入 S_CALC 後 fir 從 0x04 開始讀，

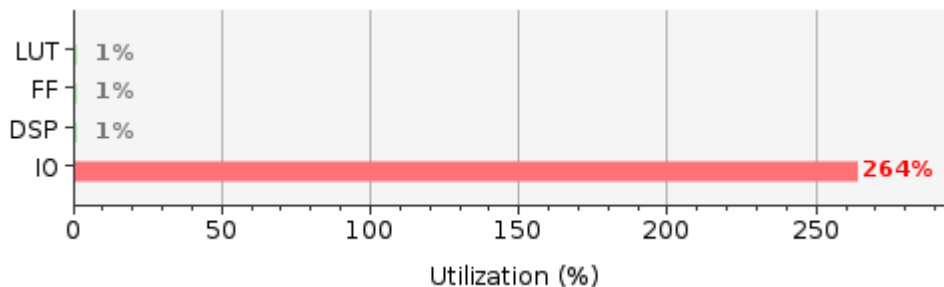
以 circular 的方式讀回 0x00，接著下一筆 data 寫入 0x04，fir 再從 0x08 開始讀一圈回 0x04。這樣的方式可以利用資料的重複性一個 cycle 讀一筆資料來做運算。比較特別的是前十筆因為演算法特性要初始化為 0，但 bram 會拿到 x，所以要特別給乘法器輸入 0。受限於 single port bram 和唯一的乘法器，加上一些 read/write cycle 上的 offset，II=14。

3. Resource Usage

- 總共用了 **283 LUTs**、**134 FFs**、**3DSPs**，未使用 Bram，FF 數量也在合理範圍，而 IO 用了超過板子所提供的量所以顯示紅色。圖中顯示 AXI-Lite 控制器 axilite_0 和乘法器 mul_0 分別的使用量，其中乘法器使用了近四成的 LUT，和全部共三個的 DSP。
- 而我所針對硬體資源使用量所做的優化是把 tap address base 從 0x20 改成 0x80，這麼做的好處是從 AXI-Lite 進來的 tap address 對應到 tap bram address 不用使用減法器，合成後發現 LUT 的數量從 288 減少 5 個到 283，幫助有限。
- 最下圖 synthesis report 可以看到更細節的用量，甚至會區分不同位元數的 component。

Name	^1	Slice LUTs (53200)	Slice Registers (106400)	DSPs (220)	Bonded IOB (125)	BUFGCTRL (32)
▼ N fir		283	134	3	330	1
axilite_0 (axilite)		67	81	0	0	0
mul_0 (mul)		111	0	3	0	0

Resource	Utilization	Available	Utilization %
LUT	283	53200	0.53
FF	134	106400	0.13
DSP	3	220	1.36
IO	330	125	264.00



```

-----
Start RTL Component Statistics
-----
Detailed RTL Component Info :
+---Adders :
    2 Input  32 Bit    Adders := 2
    2 Input  12 Bit   Adders := 4
    2 Input   5 Bit    Adders := 1
    2 Input   4 Bit    Adders := 1
+---Registers :
           32 Bit    Registers := 3
           12 Bit    Registers := 2
            4 Bit    Registers := 1
            1 Bit    Registers := 7
+---Multipliers :
           32x32    Multipliers := 1
+---Muxes :
    2 Input  32 Bit    Muxes := 8
    2 Input  12 Bit   Muxes := 5
    5 Input   5 Bit    Muxes := 1
    2 Input   5 Bit    Muxes := 1
    2 Input   4 Bit    Muxes := 2
    2 Input   1 Bit    Muxes := 6
    5 Input   1 Bit    Muxes := 1
-----

```

Finished RTL Component Statistics

4. Timing Report

- Max Frequency: 58.824 MHz (Clock Period = 17ns)

Name	Waveform	Period (ns)	Frequency (MHz)
axis_clk	{0.000 8.500}	17.000	58.824

- other constraints: input delay 和 output delay 都設為 4ns

✓ fir.xdc (/home/ywfan/SOCD_Lab/lab_3/lab-fir/fir_vivado/fir_vivado.srscs/constrs_1/new/fir.xdc)	
1	create_clock -period 17.000 -name axis_clk -waveform {0.000 8.500} [get_ports axis_clk]
2	set_output_delay -clock [get_clocks *] 4.0 [get_ports -filter { NAME == "*" && DIRECTION == "OUT" }]
3	set_input_delay -clock [get_clocks *] 4.0 [get_ports -filter { NAME == "*" && DIRECTION == "IN" }]

- Longest Path Slack: 0.528 ns

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock
Path 1	0.528	17	18	32	waddr_r_reg[5]/C	acc_r_reg[29]/D	16.367	10.816	5.551	17.0	axis_clk	axis_clk

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 0.528 ns	Worst Hold Slack (WHS): 0.140 ns	Worst Pulse Width Slack (WPWS): 8.000 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: 526	Total Number of Endpoints: 526	Total Number of Endpoints: 135

- All user specified timing constraints are met.

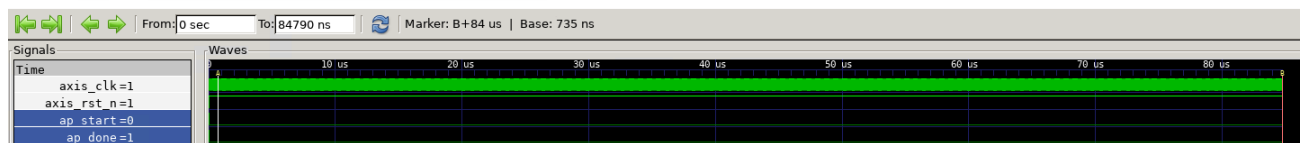
- Longest Path: waddr_r -> acc_r

- critical path 從 waddr_r 到 mul_0 到 acc_r 非常合理，因為為了省 FF 的數量，並沒有在乘法器前的 data 擋 FF，造成從取得 bram data 的 address 開始到乘加完成有較長的路徑，需要較長的 clock period。觀察路徑上的時間發現單純乘法器也需要差不多 10 ns 的時間。

Summary	
Name	Path 1
Slack	0.528ns
Source	waddr_r_reg[5]/C (rising edge-triggered cell FDCE clocked by axis_clk {rise@0.000ns fall@8.500ns period=17.000ns})
Destination	acc_r_reg[29]/D (rising edge-triggered cell FDCE clocked by axis_clk {rise@0.000ns fall@8.500ns period=17.000ns})
Path Group	axis_clk
Path Type	Setup (Max at Slow Process Corner)
Requirement	17.000ns (axis_clk rise@17.000ns - axis_clk rise@0.000ns)
Data Path Delay	16.367ns (logic 10.816ns (66.082%) route 5.551ns (33.918%))
Logic Levels	17 (CARRY4=11 DSP48E1=2 LUT2=2 LUT6=2)
Clock Path Skew	-0.145ns
Clock Uncertainty	0.035ns

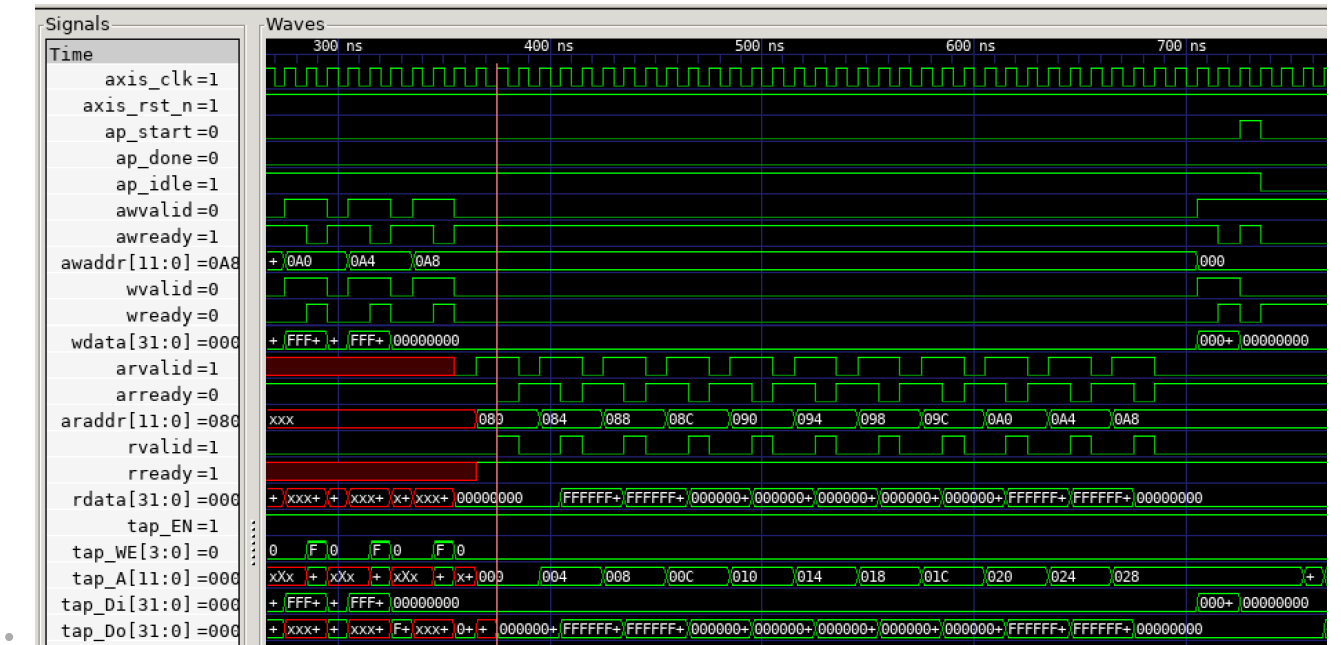
5. Simulation Waveform

- # of clock cycles from ap_start to ap_done (RTL): 84000 ns / 10 ns = 8400 cycles
- 分析 II，8400 cycles / 600 data = 14。



- Coefficient Program and Read Back

- 因為 AXI-Lite 需要做 address handshake 和 data handshake，所以一筆資料需要 3 個 cycles。



-

- 000 -> S_IDLE, 001 -> S_LOAD, 010 -> S_CALC, 011 -> S_SEND, 100 -> S_DONE

