

SoC Design Laboratory, 2023 Fall

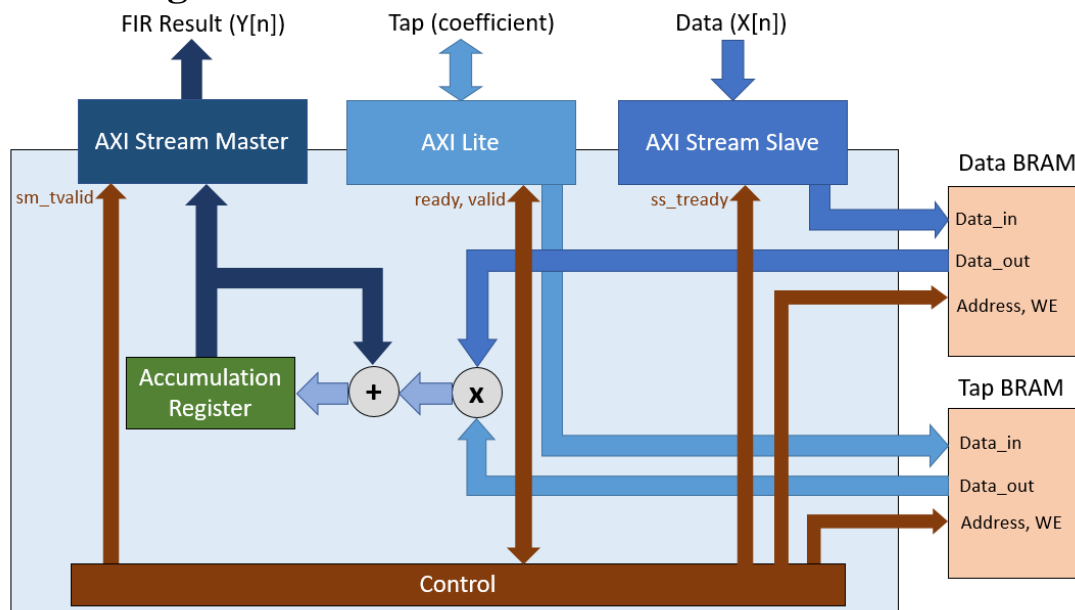
Report of Lab3

R11942072 陳俊宇

Overview

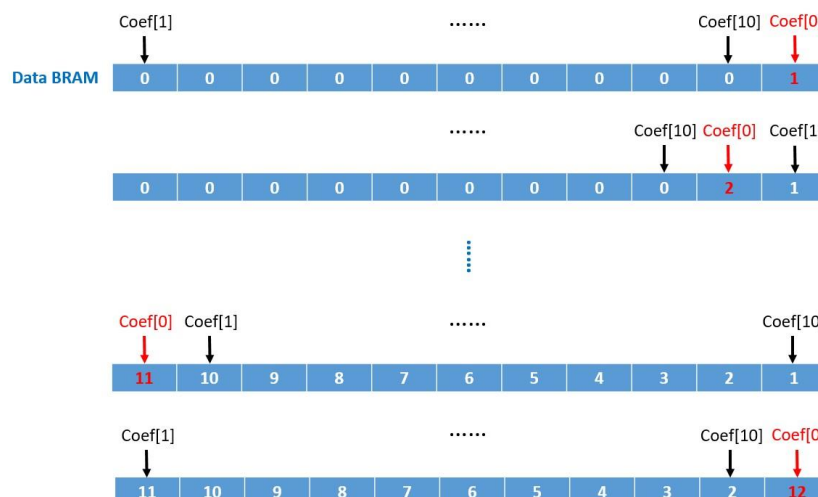
本次 Lab 中，需要人工撰寫 RTL，完成具有 AXI Lite, AXI Stream interface 的 FIR (Finite impulse response) filter module。

Block Diagram



FIR Filter Computation without Shift Register

本次 Lab 要求 data 和 tap 必須存在 BRAM 裡面，由於 BRAM 難以進行 shift 的動作，因此必須重新設計 data 和 tap 的存取控制，我的構思如下圖所示：



從中我們可以觀察到 FIR filter 運算的規律：每次需要讀入一筆新的 data，而這筆 data 需要與 coef[0]相乘(上圖紅色標註)，而上次讀入的 data 則是與 coef[1]相乘，以此類推，直到計算完 11 個相乘。

假如 data 是依照讀入順序存在各個 address，我們就只需要用一個 pointer 指向最新的 data address，從此 pointer 開始依序 accesses data BRAM，將讀出的 data 依序與 coef[0], coef[1]..., coef[11]相乘累加，即可完成 FIR filter 運算。

Resource usage

LUT and Flip Flop						
28	1. Slice Logic					
29	-----					
30						
31	+-----+-----+-----+-----+-----+-----+					
32	Site Type	Used	Fixed	Prohibited	Available	Util%
33	+-----+-----+-----+-----+-----+-----+					
34	Slice LUTs*	212	0	0	53200	0.40
35	LUT as Logic	212	0	0	53200	0.40
36	LUT as Memory	0	0	0	17400	0.00
37	Slice Registers	118	0	0	106400	0.11
38	Register as Flip Flop	115	0	0	106400	0.11
39	Register as Latch	3	0	0	106400	<0.01
40	F7 Muxes	0	0	0	26600	0.00
41	F8 Muxes	0	0	0	13300	0.00
42	+-----+-----+-----+-----+-----+-----+					

Block RAM						
65	2. Memory					
66	-----					
67						
68	+-----+-----+-----+-----+-----+-----+					
69	Site Type	Used	Fixed	Prohibited	Available	Util%
70	+-----+-----+-----+-----+-----+-----+					
71	Block RAM Tile	0	0	0	140	0.00
72	RAMB36/FIFO*	0	0	0	140	0.00
73	RAMB18	0	0	0	280	0.00
74	+-----+-----+-----+-----+-----+-----+					

Timing Report

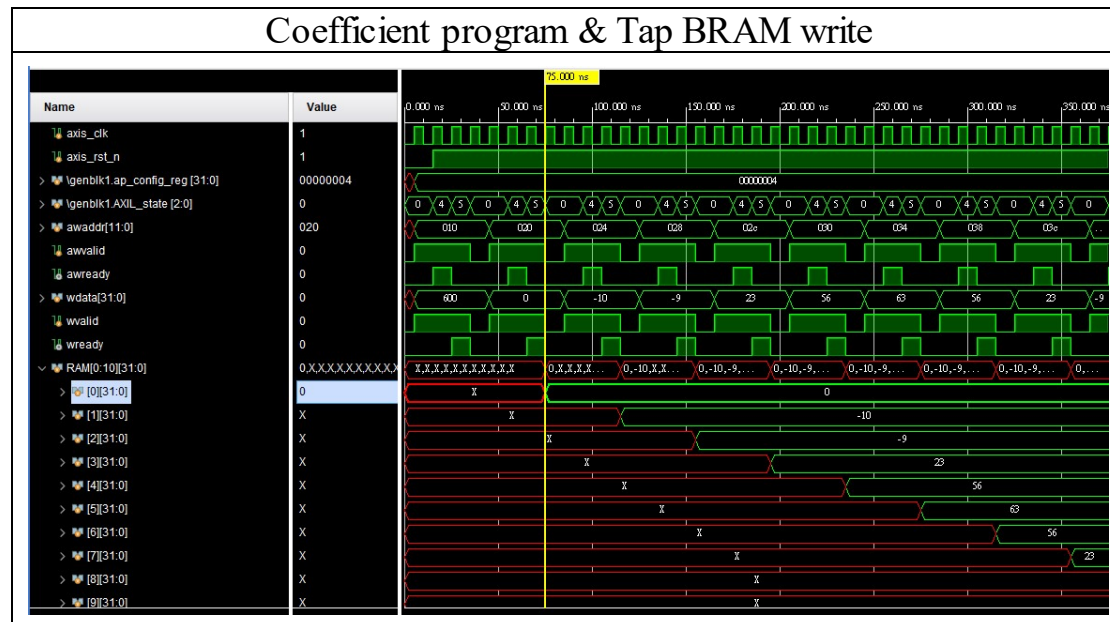
首次嘗試合成時，cycle time 設為 10ns，timing report 顯示 setup slack 約為 6ns，故第二次合成設定 cycle time 為 4ns，timing report 如下：

Timing Summary (Cycle time: 4ns)			
Design Timing Summary			
Setup		Hold	
Worst Negative Slack (WNS):		0.103 ns	Worst Hold Slack (WHS): 0.142 ns
Total Negative Slack (TNS):		0.000 ns	Total Hold Slack (THS): 0.000 ns
Number of Failing Endpoints:		0	Number of Failing Endpoints: 0
Total Number of Endpoints:		152	Total Number of Endpoints: 152
All user specified timing constraints are met.			

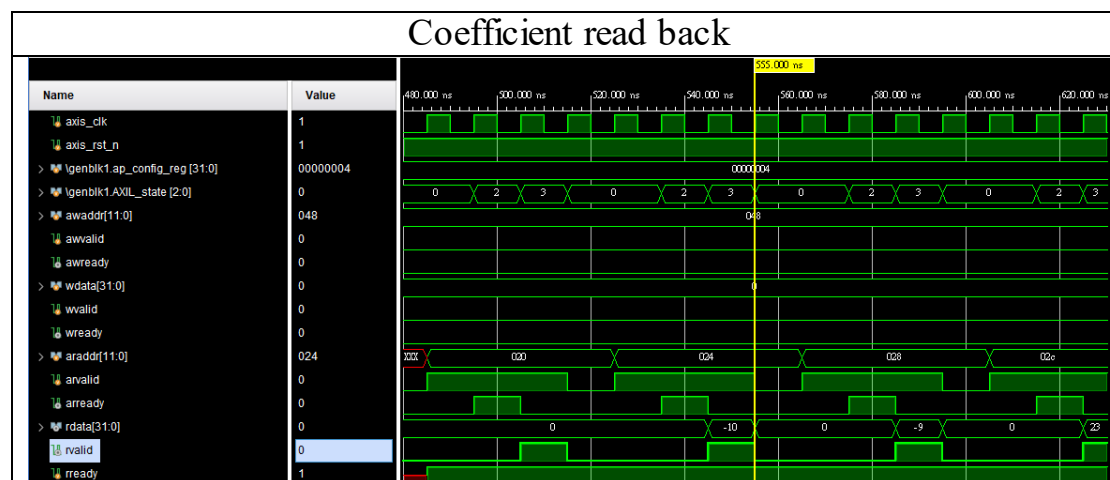
Max Delay Path	
483	Max Delay Paths
484	-----
485	Slack (MET) : 0.103ns (required time - arrival time)
486	Source: genblk1.accumulate_reg[1]/C
487	(rising edge-triggered cell FDCE clocked by axis_clk {rise@0.000ns fall@2.000ns period=4.000ns})
488	Destination: genblk1.accumulate_reg[31]/D
489	(rising edge-triggered cell FDCE clocked by axis_clk {rise@0.000ns fall@2.000ns period=4.000ns})
490	Path Group: axis_clk
491	Path Type: Setup (Max at Slow Process Corner)
492	Requirement: 4.000ns (axis_clk rise@4.000ns - axis_clk rise@0.000ns)
493	Data Path Delay: 3.761ns (logic 2.646ns (70.354%) route 1.115ns (29.646%))
494	Logic Levels: 10 (CARRY4=8 LUT2=2)
495	Clock Path Skew: -0.145ns (DCD - SCD + CPR)
496	Destination Clock Delay (DCD): 2.128ns = (6.128 - 4.000)
497	Source Clock Delay (SCD): 2.456ns
498	Clock Pessimism Removal (CPR): 0.184ns
499	Clock Uncertainty: 0.035ns ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
500	Total System Jitter (TSJ): 0.071ns
501	Total Input Jitter (TIJ): 0.000ns
502	Discrete Jitter (DJ): 0.000ns
503	Phase Error (PE): 0.000ns

觀察 MaxDelay Path 的 Source 與 Destination，可知電路的 critical path 在於計算 data, coefficient 相乘並與累加的部分
Setup slack 為 0.103ns。

Simulation Waveform



Coefficient 由 AXI lite interface 輸入，fir 接收後直接寫入 Tap BRAM。從波形可以看到，每當 wready=1 時，下個 cycle 就會有一筆 coefficient 寫入 Tap BRAM。



將 coefficient 從 Tap BRAM 讀出，並且從 AXI lite interface 輸出。從波形可以看到，當 rvalid=1 時，代表讀出的 coefficient 已經放在 rdata 上，準備好被讀出。

Data stream in & Data BRAM write

The timing diagram illustrates the interaction between various signals during a data stream input and Data BRAM write operation. The horizontal axis represents time in nanoseconds (ns), ranging from approximately 950,000 ns to 1,250,000 ns. The vertical axis lists the signals being monitored.

Signals and their values:

- axis_clk:** A periodic clock signal.
- axis_rst_n:** A reset signal, shown as a low pulse at the beginning.
- lgenblk1_AXIS_state [1:0]:** A 2-bit state signal, transitioning through values 0, 1, 2, 3, 1, 2, 3, 1, 2.
- ss_idata[31:0]:** A 32-bit data stream input, shown as a sequence of 0s and 1s.
- RAM[0:10][31:0]:** A 32-bit Data BRAM write signal, shown as a sequence of 0s and 1s.
- ss_valid:** A valid signal, shown as a high pulse.
- ss_tready:** A ready signal, shown as a high pulse.
- ss_last:** A last signal, shown as a high pulse.
- sm_idata[31:0]:** A 32-bit data stream input, shown as a sequence of 0s and 1s.
- sm_valid:** A valid signal, shown as a high pulse.
- sm_tready:** A ready signal, shown as a high pulse.
- sm_last:** A last signal, shown as a high pulse.

The diagram shows that the data stream input and Data BRAM write signals are synchronized with the clock and other control signals.

需要進行 FIR filter 計算的 input data 是由 AXI stream slave interface 輸入，每當 fir 準備好接收下一筆 data 時 (前一筆 data 已經存入 Data BRAM)，就把 ss_tready 拉到 1，當 testbench 看到 ss_tready=1 就會輸入下一筆 data。

Data stream out

The timing diagram displays the data stream output for various signals over time. The signals are listed on the left, and their values are shown in the middle column. The right column shows the corresponding waveforms. A yellow vertical line marks the time 2,915,000 ns.

Name	Value
axis_clk	1
axisrst_n	1
lggenblk1_AXIS_state [1:0]	3
ss_idata[31:0]	16
RAM[0:10][31:0]	12,13,14,15,5,6,7,8
ss_valid	1
ss_tready	0
ss_tlast	0
sm_idata[31:0]	1830
sm_valid	1
sm_tready	1
sm_tlast	0

運算完成的 output data 由 AXI stream slave interface 輸出，每當 fir 算完一筆 data 的結果，並且已經將結果放到 sm_tdata 上時，就可以把 sm_tvalid 拉到 1，當 testbench 看到 sm_tvalid=1 就會讀出 data 並且與 golden value 進行比對。