SOC Design

Lab-3 FIR

110590022 陳冠晰

- **Function specification**

  • y[t] = Σ (h[i] * x[t - i])

  In signal processing, a finite-impulse-response, or finite impulse response models are generally linear dynamic models characterized by finite-order moving average representations, implying that their responses to impulse inputs go to zero after a finite number of time steps, equal to the model's memory length.

  FIR filters can be discrete-time or continuous-time, and digital or analog.

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N]$$
$$= \sum_{i=0}^{N} b_i \cdot x[n-i],$$

  Where:

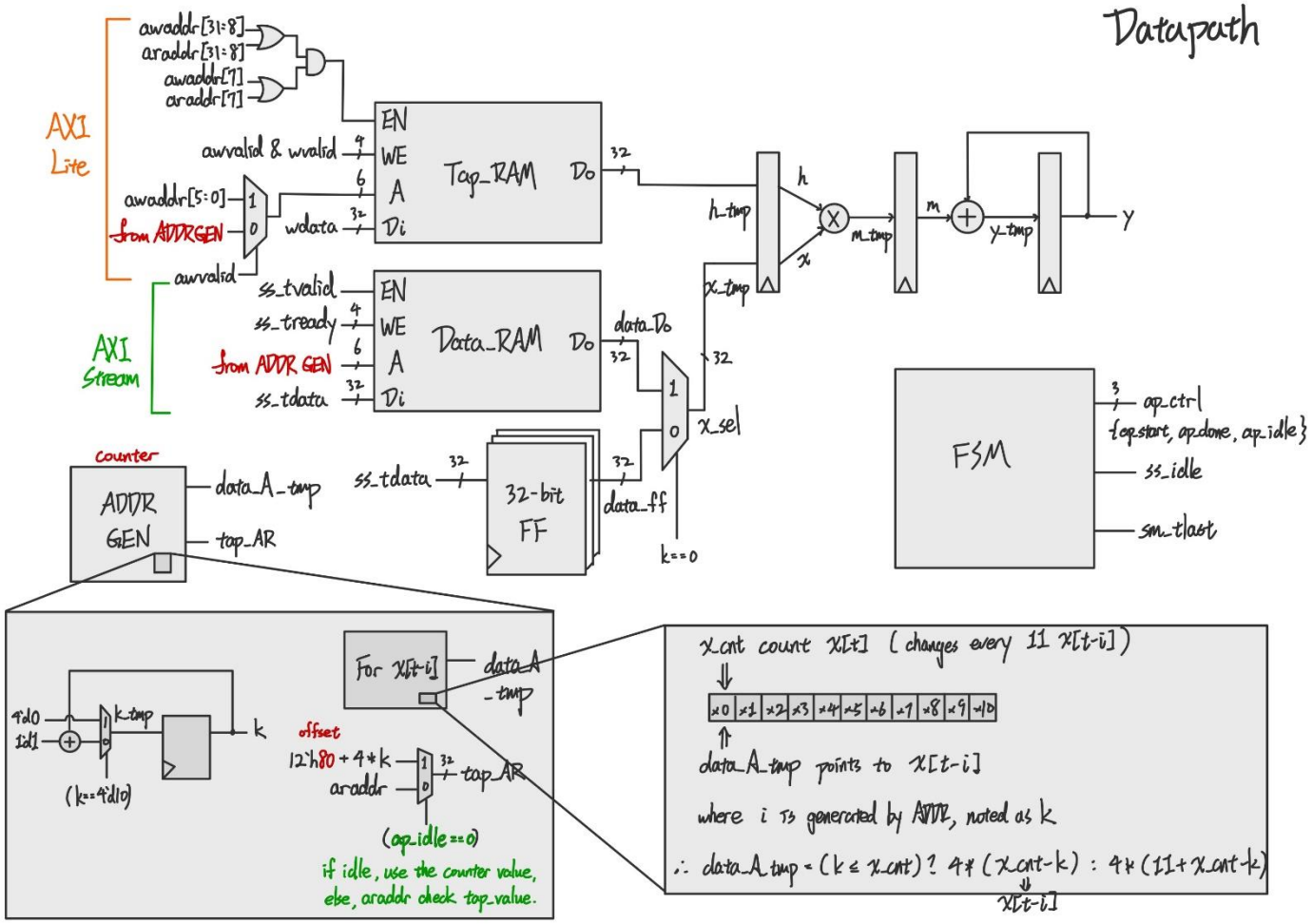
  . x[n] is the input signal,

  . y[n] is the output signal,

  . N is the filter order; an N_{th}-order filter has N + 1 terms on the right-hand side

  . b_i is the value of the impulse response at the i_{th} instant for 0 <= I <= N of an Nth-order FIR filter. If the filter is a direct form FIR filter, then b_i is also a coefficient of the filter.

- **Block Diagram**

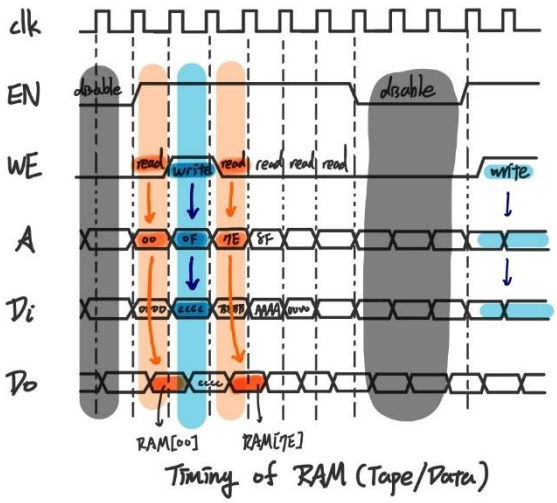


Tap_RAM / Data_RAM

32-bit ↔ 4 byte

reg [3:0] RAM [depth] ⇒ RAM [A>>2] (÷4)

A[10:0] → in order to access byte



Timing of RAM (Tape/Data)

TapeRAM:

Tape address: 0x80 - 0xFF (8'b10000000)

EN = (AW[31:8] == 0) | (AR[31:8] == 0) & (AW[7] | AR[7])

Check if read/write address belongs to TapeRAM

WE = (AWvalid && Wvalid == 1) ? 4'b1111 : 4'b0000; Check if address/data write is valid

A = AW[5:0] AW will >>2, 4-bit left, still can represent the address of 11 tapes

Di = Wdata Data-in equals to wdata of AXI-Lite (h[i] flow through TapeRAM by axilite)

Awready = 1 (RAM 尚有空間)

wready = 1 (Data Buffer 尚有 space)

DataRAM:

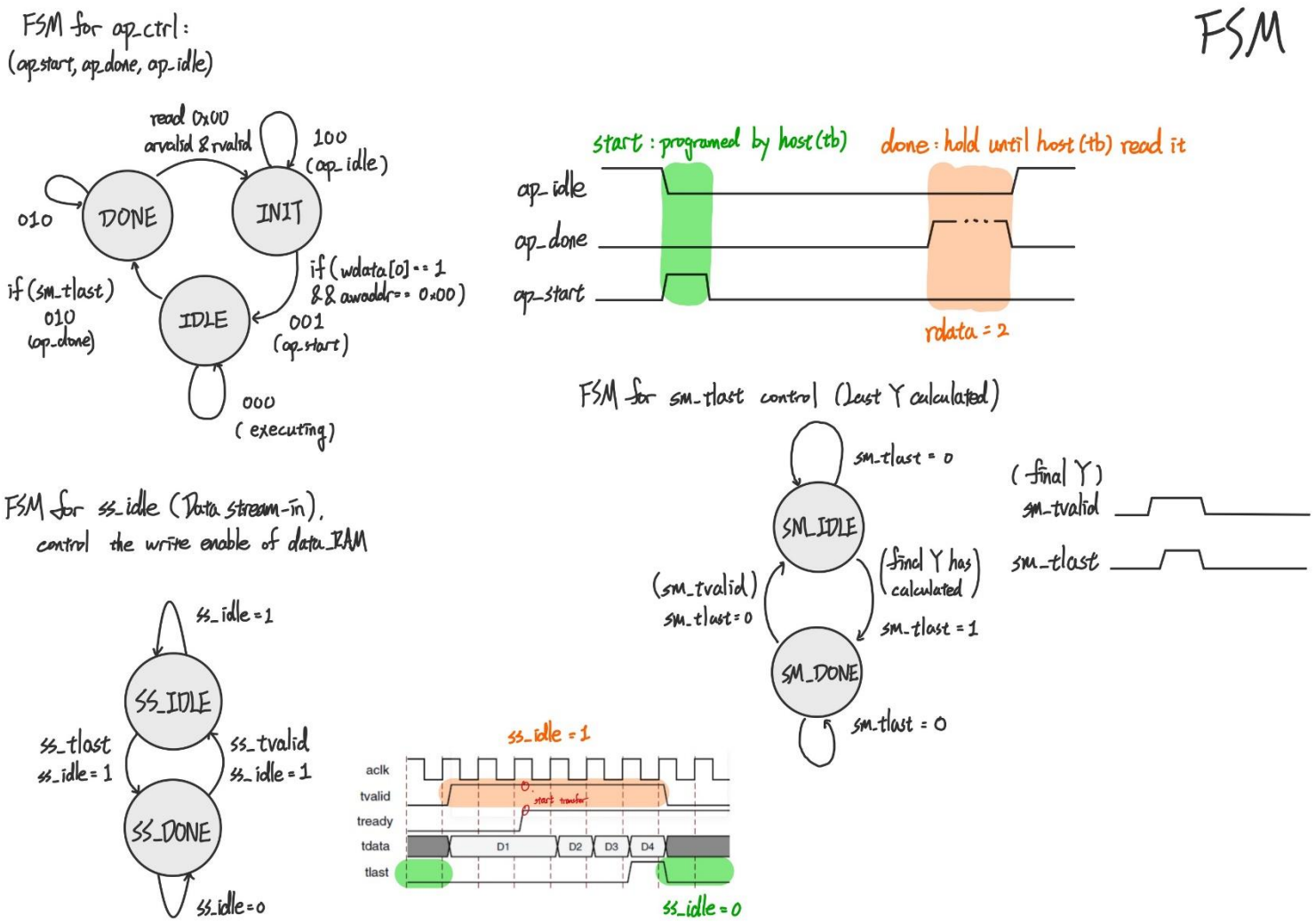
EN = ss_tvalid

WE = (ss_tready & ss_idle) ? 4'b1111 : 4'b0000; (Ready to write "New value" & not finish) (ss_tlast = 0)

A = (ap_ctrl[2] && init_addr < 6'd44) ? init_addr : data_A_tmp;

If ap_idle == 1, initialize the value in dataRAM.

Di = ss_tdata

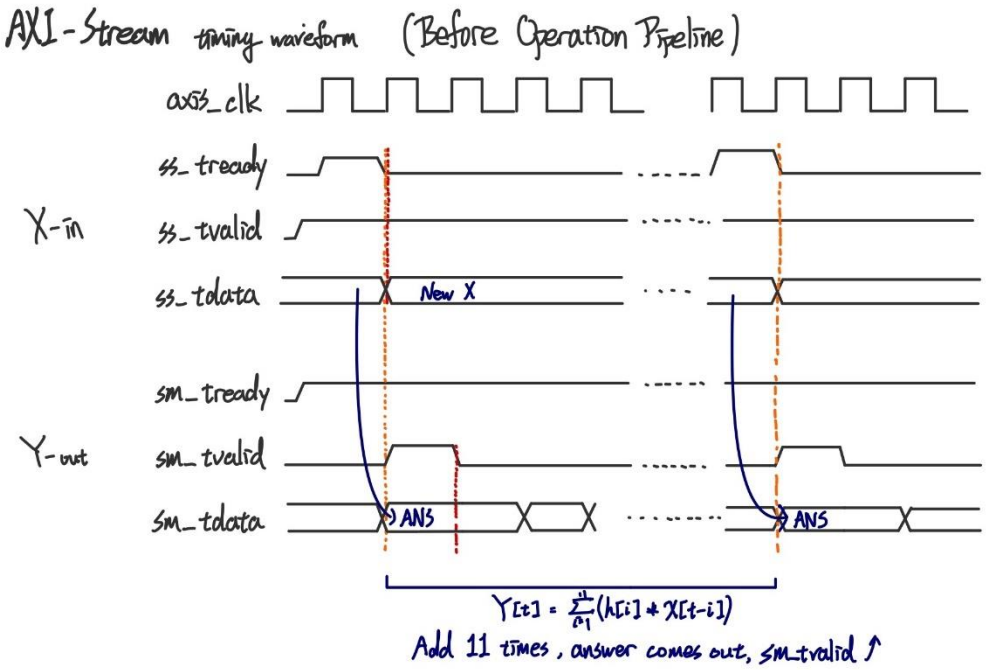


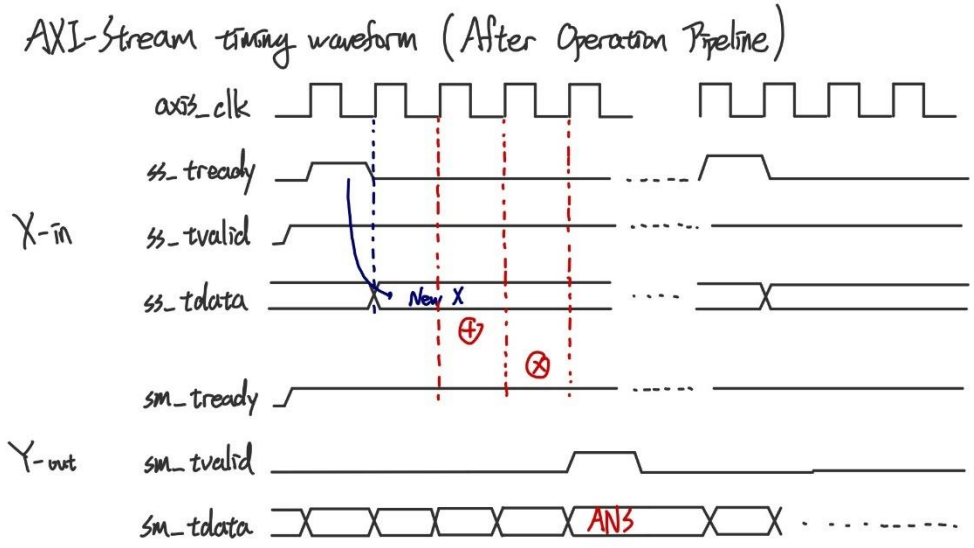
1. 左上角的 FSM 是用來給 ap_start, ap_done, ap_idle 的，一開始處在 INIT state，ap_ctrl = {ap_idle, ap_done, ap_start} = 3'b100，當我們 Host 端 (testbench) program ap_start，代表 FIR 開始運作，ap_idle 降下，進到 IDLE state，ap_ctrl = {ap_idle, ap_done, ap_start} = 3'b000。當我們完成最後一個 Y 的計算，傳出去給 testbench 比對，同時拉高 sm_tlast，就代表 FIR 做完運算，進到 DONE state，ap_ctrl = {ap_idle, ap_done, ap_start} = 3'b010。由於 testbench 需要讀取 ap_done 的訊號，所以要等到 read address == 0x00 讀取到 ap_done 過後再回到 initial 的 state。
2. 左下角的 FSM 專門產生 ss_idle 的訊號拿去 data_RAM 當成 write enable，在還未收到 ss_tlast 的期間，都會是 1，等到收到 ss_tlast，才會進到 SS_DONE state，ss_idle = 0，代表不能再寫了。
3. 右邊的 FSM 是在產生 sm_tlast，當最後一個 Y 計算完，counter 數到 data length，就進到 SM_DONE state，sm_tlast 拉高一個 cycle。

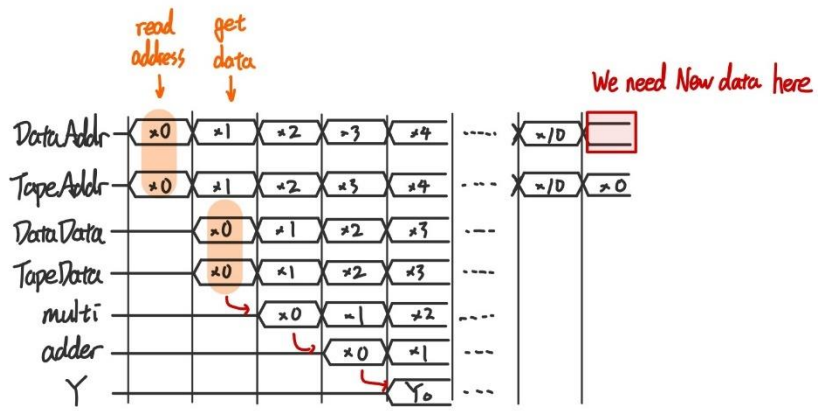
- **Operation**

圖片中上半部分是還未做 operation pipeline 會呈現出的 y 波形圖。

ss_tready 每次拉高，代表 stream-in 新的 x[t]進來 RAM。再下個 cycle 可以發現







sm_tvalid 拉高，代表計算出的 Y 被拿去 testbench 做比對。

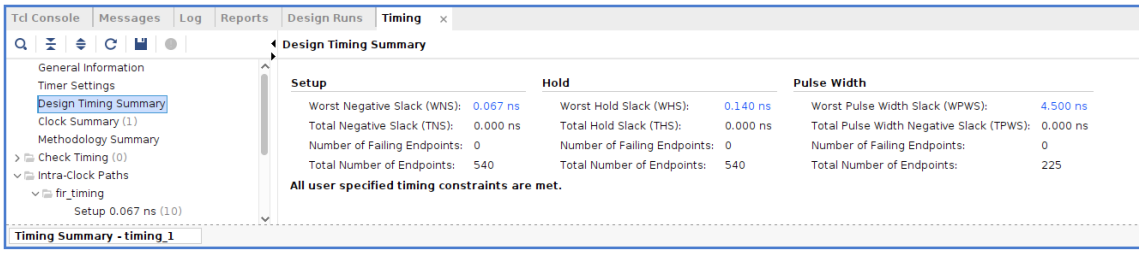
上圖最下面是做完 operation pipeline 後呈現出的 timing waveform，可以看到中間圖片中，X stream-in 和 Y stream-out 間變化。

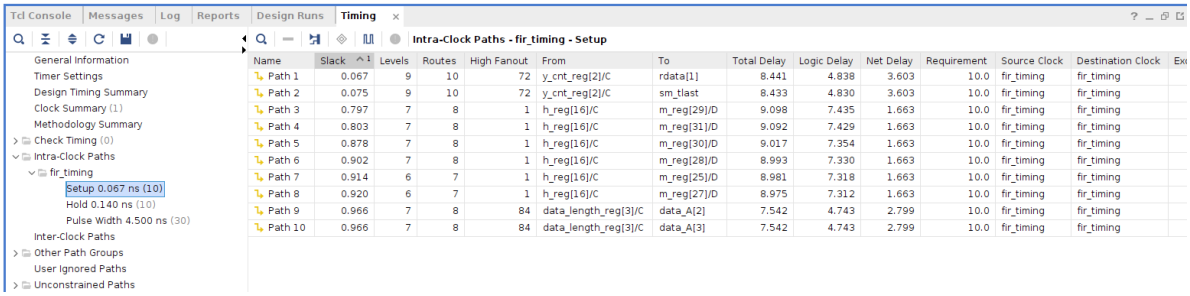
- **Resource usage**

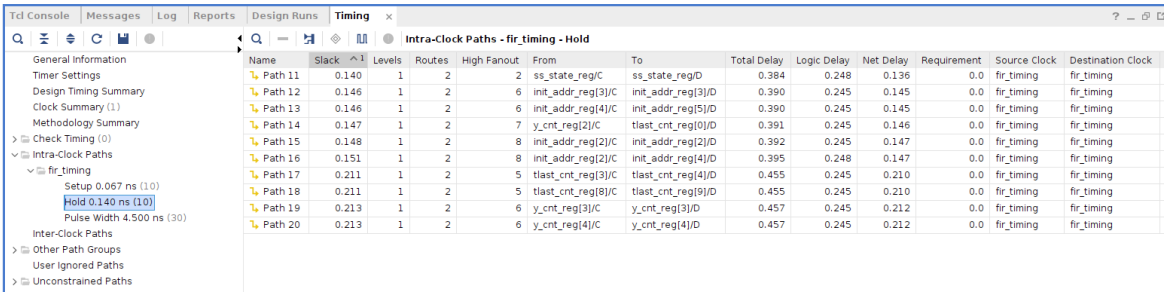
Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	190	0	0	53200	0.36
LUT as Logic	190	0	0	53200	0.36
LUT as Memory	0	0	0	17400	0.00
Slice Registers	224	0	0	106400	0.21
Register as Flip Flop	224	0	0	106400	0.21
Register as Latch	0	0	0	106400	0.00
F7 Muxes	0	0	0	26600	0.00
F8 Muxes	0	0	0	13300	0.00

Site Type	Used	Fixed	Prohibited	Available	Util%
Block RAM Tile	0	0	0	140	0.00
RAMB36/FIFO*	0	0	0	140	0.00
RAMB18	0	0	0	280	0.00

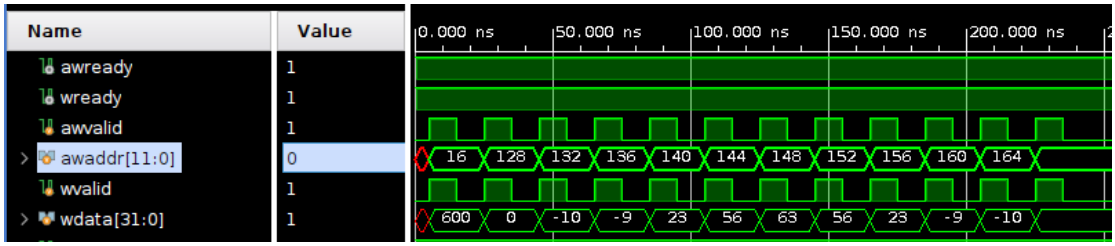
- **Timing report**



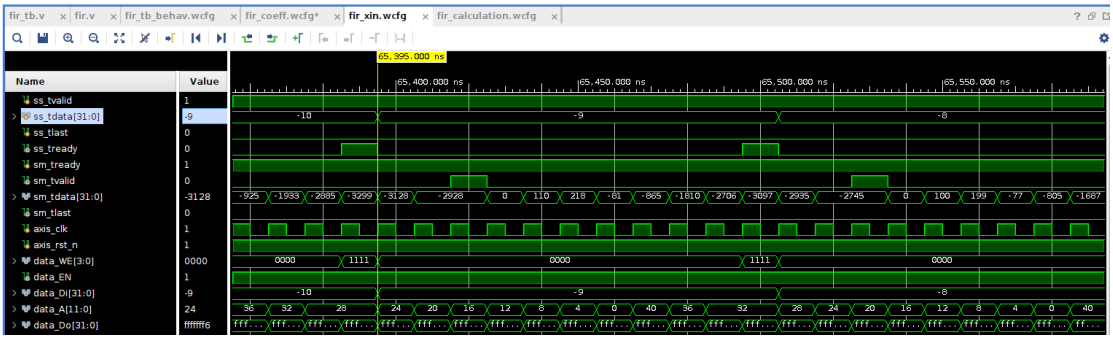




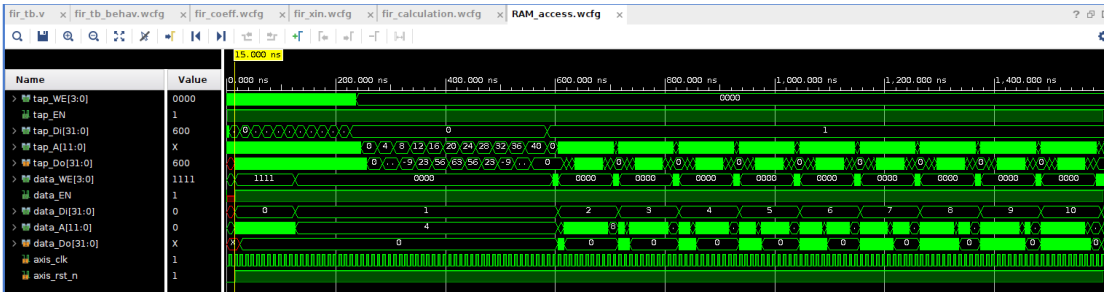
- **Simulation waveforms**
  - Coefficient



  - X-in / Y-out



  - RAM access control



可以看到一開始 tap_Di 在 load in coefficients，再來是 data 的 stream in。