

112 Fall Semester

535226

系統晶片設計

SOC Design Laboratory

LAB3

王語

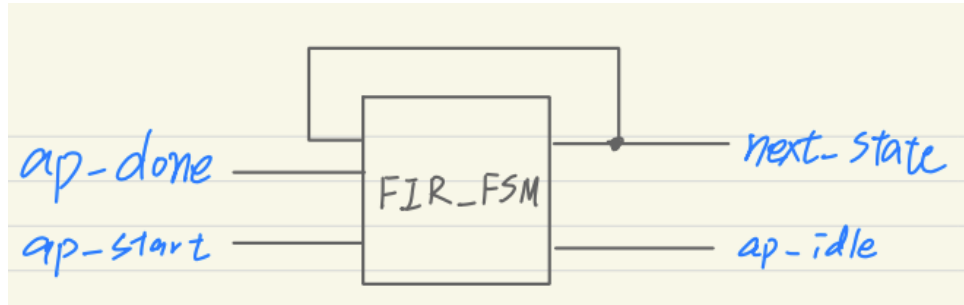
學號：312605003

系級：機器人學程碩一

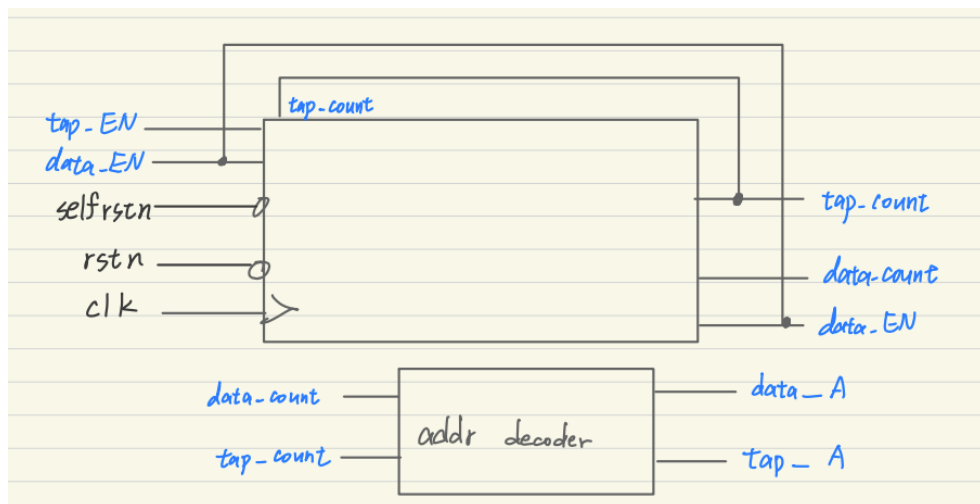
## 一、Block Diagram

詳細電路請見 RTL\_Viewer(schematic).pdf

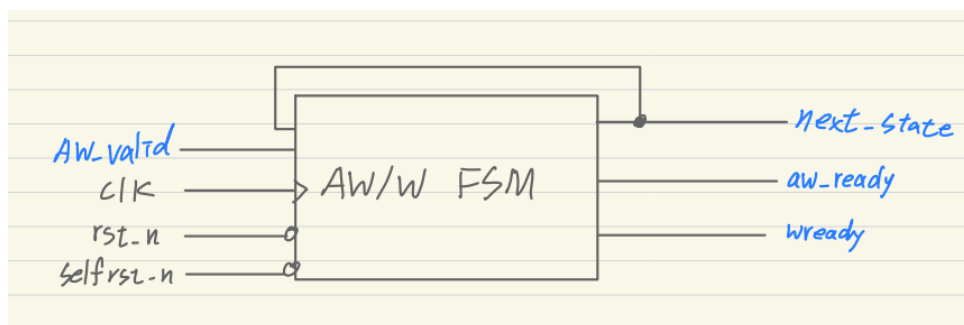
### a. FIR FSM



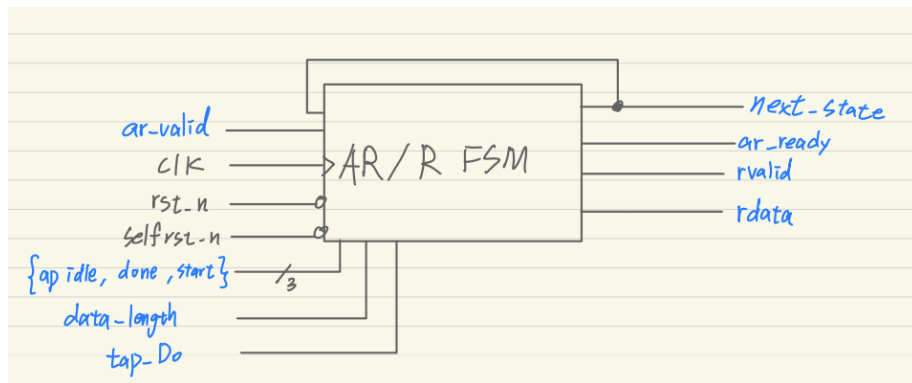
### b. RAM counter & decoder



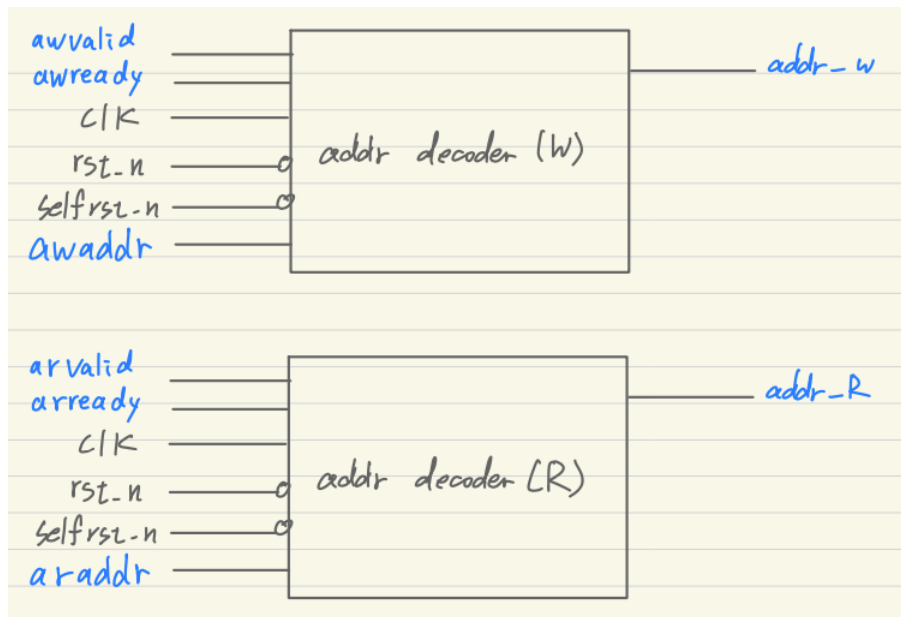
### c. AXI-Lite AW / W FSM



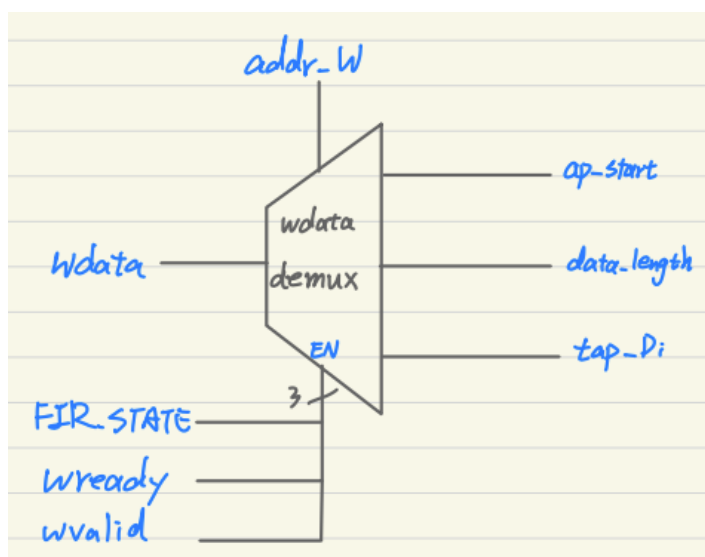
d. AXI-Lite AR / R FSM



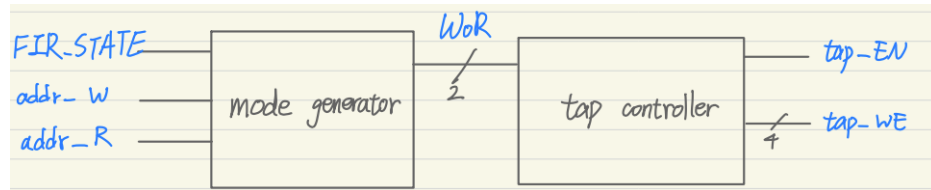
e. Addr decoder



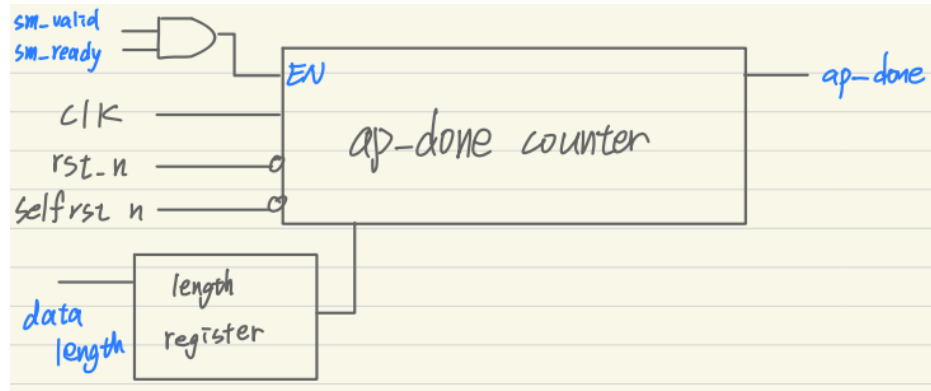
f. AXI-Lite w\_data demux



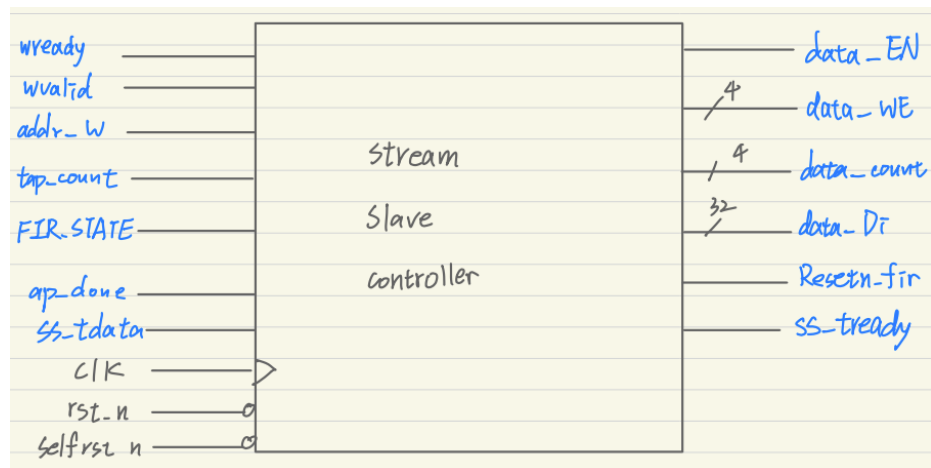
g. Tap RAM controller



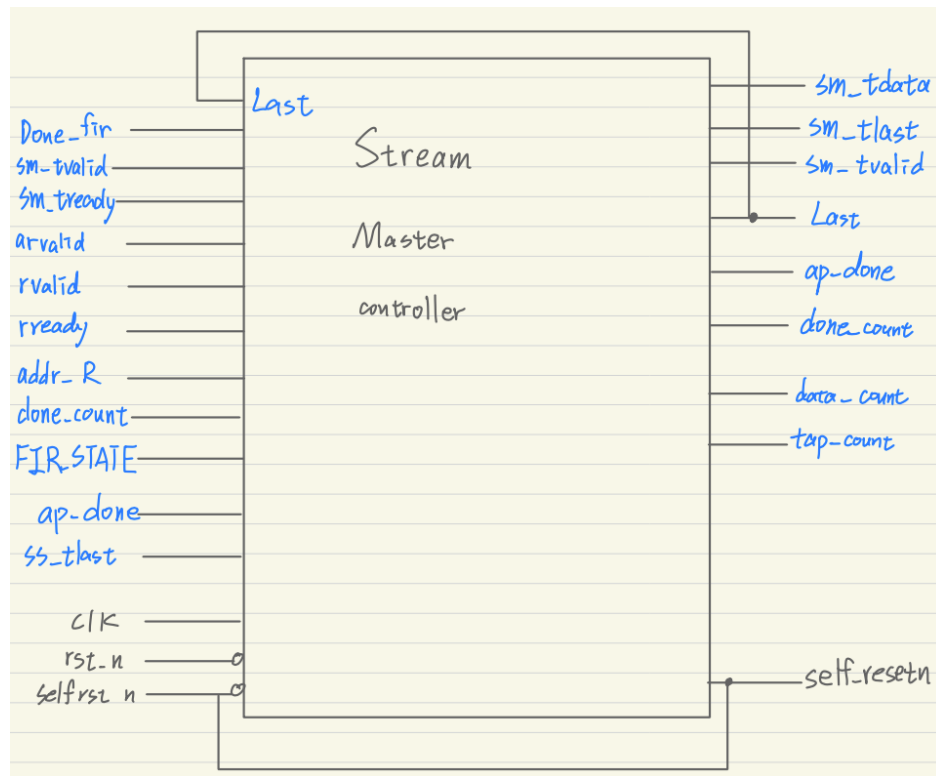
h. ap\_done counter



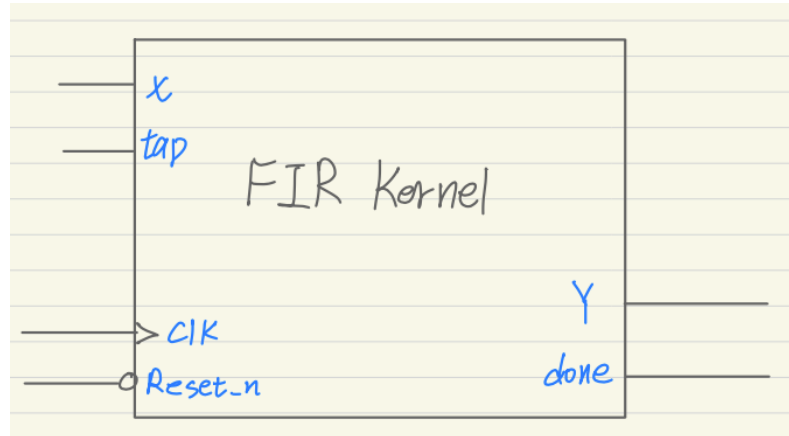
i. Stream Slave Controller



j. Stream Master Controller

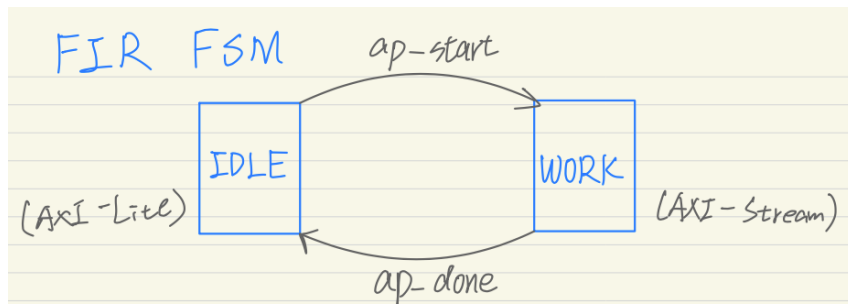


k. FIR Kernel



二、Describe operation

### a. FIR FSM



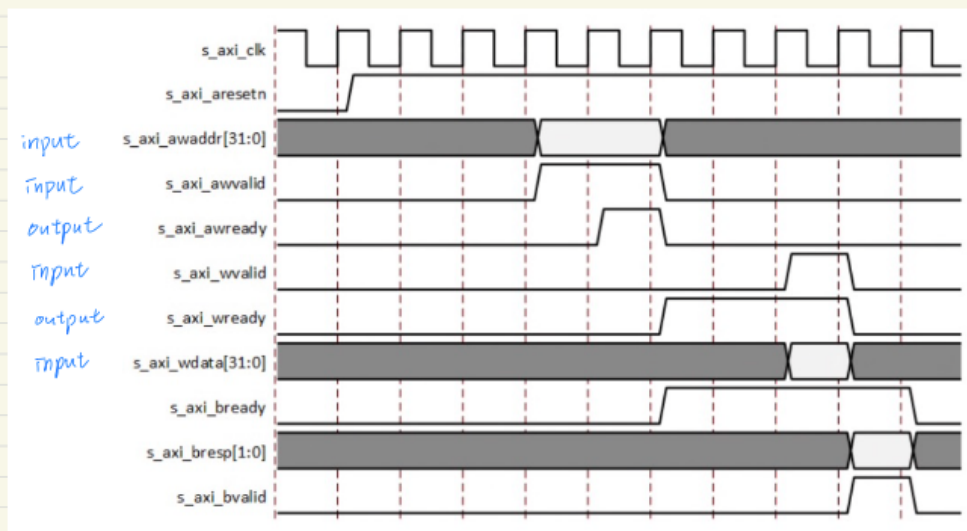
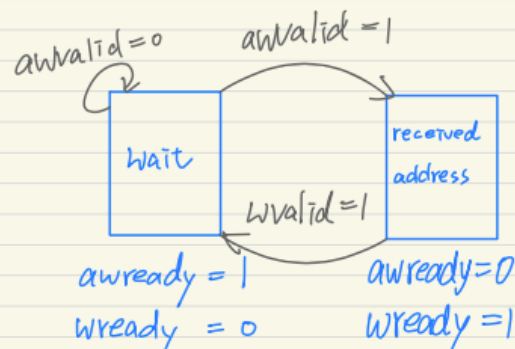
FIR STATE 的切換很簡單，只有兩個 State，分別是 IDLE(0)以及 WORK(1)。當接收到 *ap\_start* 時會由 IDLE 切換至 WORK，反之，當接收到 *ap\_done* 的時候會切換回 IDLE。

狀態機初始狀態是 IDLE，當接收到 *axis\_reset\_n* 時，也會返回 IDLE 狀態。

### b. AXI AW/W FSM

AXI AW/W FSM (外部寫入 *top.apstart.length*)

Case 1: FIR\_STATE = IDLE



AXI\_Lite 的寫入通道(AW、W)只會在 FIR = IDLE 的狀態啟用，負責寫入係數 tap 以及 ap\_start，當 FIR = WORK 的時候將不再寫入新的資料，因此當 FIR = WORK，awready、wready 皆保持在 0 的狀態。

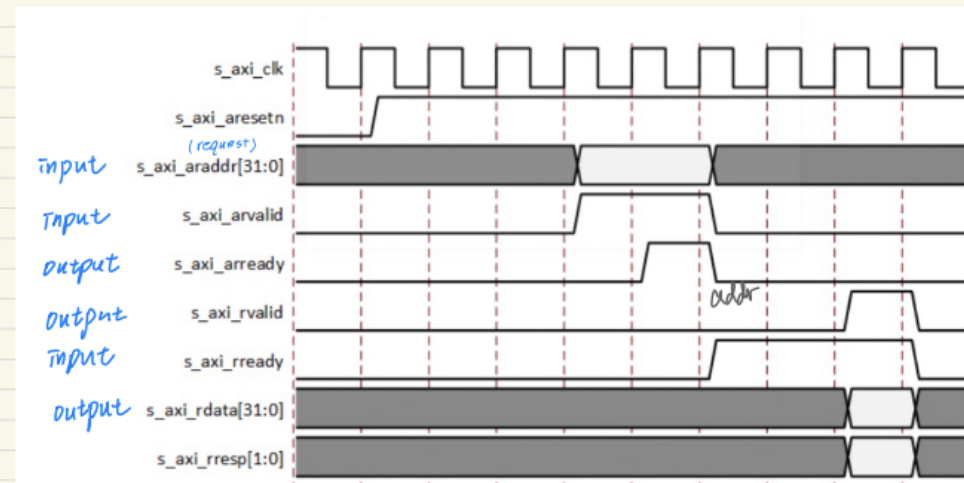
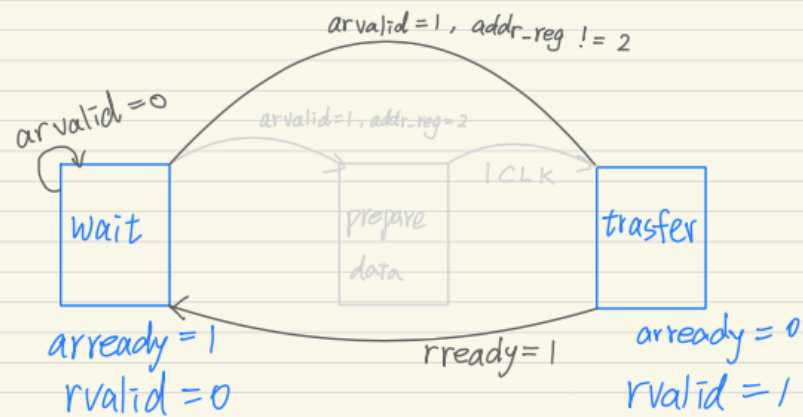
當 FIR = IDLE 時，狀態的切換如上圖所示，若接收到 axis\_reset\_n 時，會返回 WAIT 狀態。

### c. AXI AR/R FSM

#### AXI-Lite (R/AR) FSM

\*: ap-done = r0      ap-idle = r0

讀 ap 信號，coef[0] ~ [10] → 在 Tap bram



AXI\_Lite 的讀取通道(AR、R)在整個流程中皆會被使用，包括系統開始前的 ap\_idle 訊號確認、寫入 tap 後的 tap RAM 資料確認、運算過程的 ap\_done 訊號確認，因此不考慮 FIR STATE 的變化，初始狀態為 WAIT，

等到成功接收 addr 資訊後，即開始傳輸資料，傳輸結束回到 WAIT 狀態，這邊灰色的圖是原先 pipeline 考量，但經改善後資料可以在 1clk 內傳出，因此移除該狀態。

#### 改善：

本 AXI-Lite 設計對於 Aready (awready / arready) 在確認 adata (ardata / awdata) 之前就已經維持在 1 了，這是基於 Testbench 行為設計，但有鑒於 adata 的地址可能不在指定範圍內，將可能接收到既不屬於 ap\_signals、data length 也不屬於 tap 的資料，因此之後會新增一個 confirm 狀態，確認 adata 是有效地址後才進行 Aready 的 set。

#### d. Tap RAM

當 wvalid & wready = 1 時代表接收到外部資料，同時，addr 若為 tap 地址，代表這筆資料是係數，此時 tap\_EN、tap\_WN 為 1 與 1111，會將收到的資料寫入 tap\_bram，因為資料是依序傳入，因此負責管理 tap\_A 的 counter 會+1。

在寫入 tap 時，會順帶寫 0 到 data ram，完成 bram 的 reset，這個動作可以防止運算時讀到 unkown 信號。

如果 tap\_EN=1，則 Tap\_count 遞增，若為 0，則不變化，為了防止生成 latch，之後會將此部分程式改成  $\text{tap\_count} \leq \text{tap\_count} + \text{tap\_EN}$ 。

讀出 tap 的驗證也同理，只是 tap\_WN 換作 0000。

進行 FIR 運算時，tap\_WN 維持 0000，tap\_EN 保持 1，讓每一個 clk 皆讀出係數。

#### 改善：

其實地址可以直接由 wdata 的地址作為 tap\_A 的地址，如此一來就能省去掉 counter。



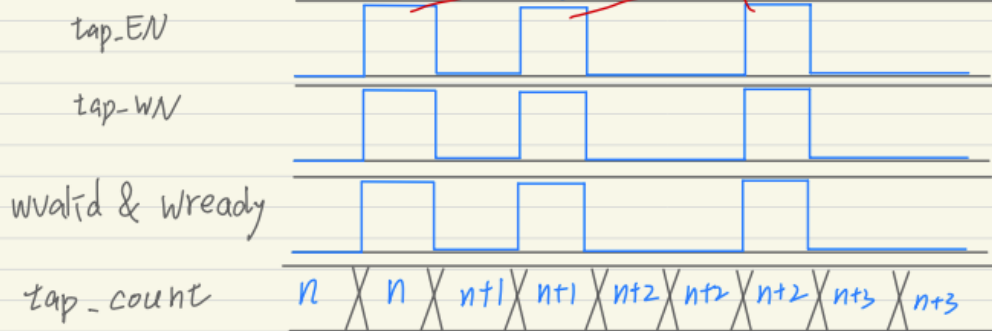
## Tap BRAM controller

key:  $\text{if}(\text{EN}) \{ \text{tap\_count} \leftarrow \text{tap\_count} + 1 \}$

1. 外部傳送  $\text{tap}[0] \sim \text{tap}[10]$ , 寫入 tap bram

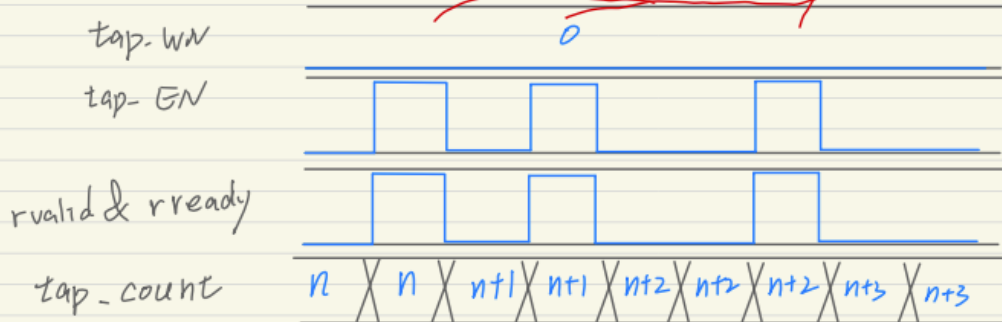
AXI-Lite-AW/W

FIR-STATE = IDLE,  $\text{addr\_reg\_W} = 2$   $\text{WoR} = 2'b10$



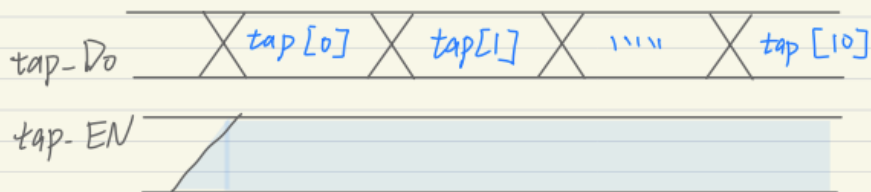
2. 驗證 tap, 從 tap bram 讀出  $\text{tap}[0] \sim \text{tap}[10]$

FIR-STATE = IDLE, AXI-Lite-AR/R  $\text{WoR} = 2'b01$



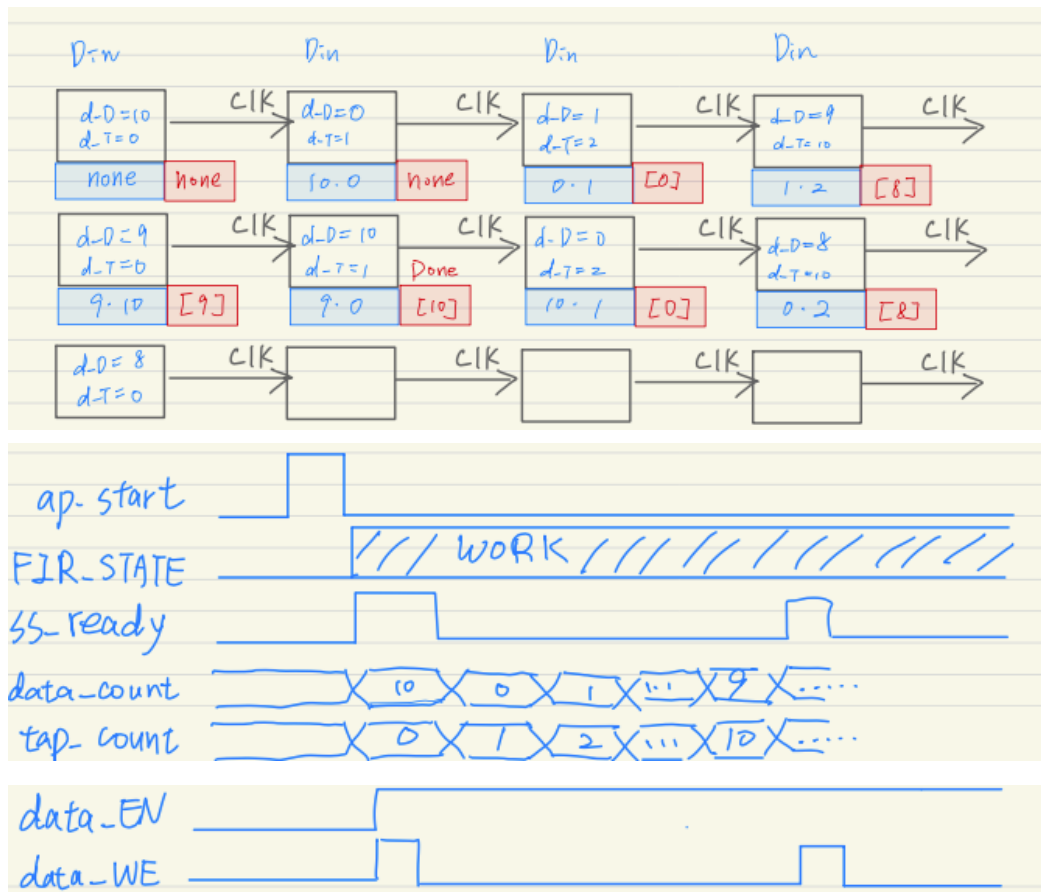
3. 運算, 從 tap bram 讀出  $\text{tap}[0] \sim \text{tap}[10]$

FIR-STATE = WORK  $\text{WoR} = 2'b01$



## e. FIR calculation

FIR 的運算是連續無中斷的，下圖中黑色方框內代表 data ram 和 tap ram 輸出哪個地址的資料，藍色方框代表 FIR Kernel 正在進行哪個地址的 data 和哪個地址的 tap 運算，紅色方框代表已經完成多少筆 FIR 累加，當 FIR 累加 11 筆資料時會輸出 DONE 信號。



當接收到 done 信號時，會將運算結果透過 stream master 傳輸出去。

每個 loop tap\_count 皆從 0 開始遞增至 10，當 tap\_count=0 時 data\_WE 會 set，寫入新的資料，data\_count 的起始值由最舊的資料開始，loop 0、loop 1、loop 2 起始值依序為 10、9、8，以此類推。

### 改善：

要讓 FIR 運算不間斷的關鍵其實是  $x[n]$  是否準時接收、 $y[n]$  是否順利被收走，其實這樣的設計存在巨大風險，必須在每輪 loop 開始前接收到新的  $x[n]$ ，並且在當下這輪 loop 結束前將前一筆  $y[n]$  傳送出去，否則運算結果會出錯，因此應該新增暫停狀態，並且在資料的接收端、輸出端新增 FIFO，減少資料傳遞造成的耽誤。

### f. self\_reset\_n

為了確保整個系統可以暢通無阻的運行，每當結束一批資料的 FIR 運算後，會透過自我重置的方式，回到 IDLE 狀態，會將暫存器的數值回歸初始狀態。

### 三、Resource usage

#### 1. Slice Logic

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	129	0	0	53200	0.24
LUT as Logic	129	0	0	53200	0.24
LUT as Memory	0	0	0	17400	0.00
Slice Registers	226	0	0	106400	0.21
Register as Flip Flop	159	0	0	106400	0.15
Register as Latch	67	0	0	106400	0.06
F7 Muxes	0	0	0	26600	0.00
F8 Muxes	0	0	0	13300	0.00

#### 2. Memory

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

#### 3. DSP

Site Type	Used	Fixed	Prohibited	Available	Util%
DSPs	3	0	0	220	1.36
DSP48E1 only	3				

#### 4. IO and GT Specific

Site Type	Used	Fixed	Prohibited	Available	Util%
Bonded IOB	318	0	0	125	254.40
Bonded IPADs	0	0	0	2	0.00
Bonded IOPADs	0	0	0	130	0.00
PHY_CONTROL	0	0	0	4	0.00
PHASER_REF	0	0	0	4	0.00
OUT_FIFO	0	0	0	16	0.00
IN_FIFO	0	0	0	16	0.00
IDELAYCTRL	0	0	0	4	0.00
IBUFDS	0	0	0	121	0.00
PHASER_OUT/PHASER_OUT_PHY	0	0	0	16	0.00
PHASER_IN/PHASER_IN_PHY	0	0	0	16	0.00
IDELAYE2/IDELAYE2_FINEDELAY	0	0	0	200	0.00
ILOGIC	0	0	0	125	0.00
OLOGIC	0	0	0	125	0.00

## 5. Clocking

-----

Site Type	Used	Fixed	Prohibited	Available	Util%
BUFGCTRL	3	0	0	32	9.38
BUFIO	0	0	0	16	0.00
MMCME2_ADV	0	0	0	4	0.00
PLLE2_ADV	0	0	0	4	0.00
BUFMRCE	0	0	0	8	0.00
BUFHCE	0	0	0	72	0.00
BUFR	0	0	0	16	0.00

## 6. Specific Feature

-----

Site Type	Used	Fixed	Prohibited	Available	Util%
BSCANE2	0	0	0	4	0.00
CAPTUREE2	0	0	0	1	0.00
DNA_PORT	0	0	0	1	0.00
EFUSE_USR	0	0	0	1	0.00
FRAME_ECCE2	0	0	0	1	0.00
ICAPE2	0	0	0	2	0.00
STARTUPE2	0	0	0	1	0.00
XADC	0	0	0	1	0.00

### 7. Primitives

---

Ref Name	Used	Functional Category
OBUF	169	IO
IBUF	149	IO
FDCE	94	Flop & Latch
LDCE	67	Flop & Latch
FDRE	59	Flop & Latch
LUT6	48	LUT
LUT3	45	LUT
LUT4	26	LUT
LUT2	24	LUT
CARRY4	13	CarryLogic
LUT5	9	LUT
FDPE	6	Flop & Latch
LUT1	4	LUT
DSP48E1	3	Block Arithmetic
BUFG	3	Clock

四、Timing Report

a. 最大频率

```
Project Summary x Device x constrain_01.xdc x
/home/a605/soclab/mylab3/lab3_vivado/lab3_vivado.srscs/constrs_1/new/constrain_01.xdc
1: create_clock -period 4.10 -name axis_clk -waveform {0.000 2.05} [get_ports axis_clk]
2:
```

b. Slack

Design Timing Summary

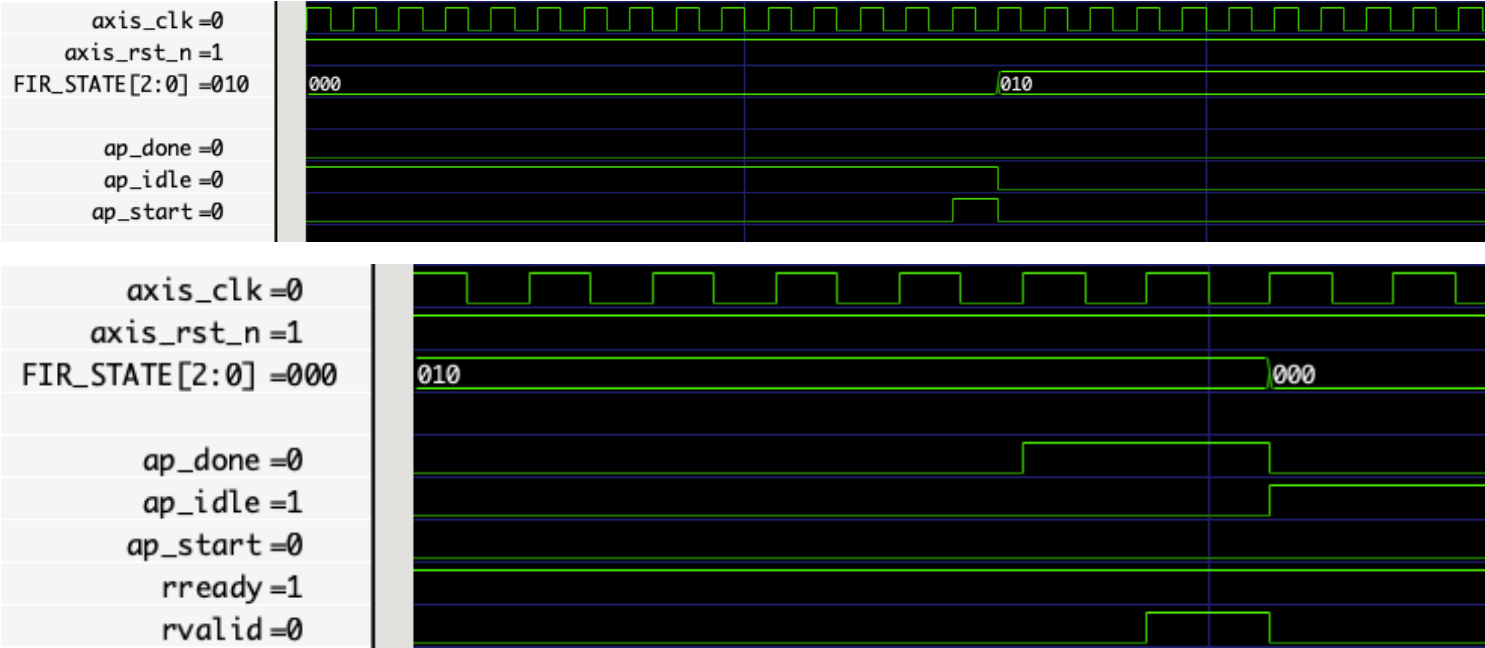
Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 0.035 ns	Worst Hold Slack (WHS): 0.113 ns	Worst Pulse Width Slack (WPWS): 1.550 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 275	Total Number of Endpoints: 275	Total Number of Endpoints: 160

All user specified timing constraints are met.

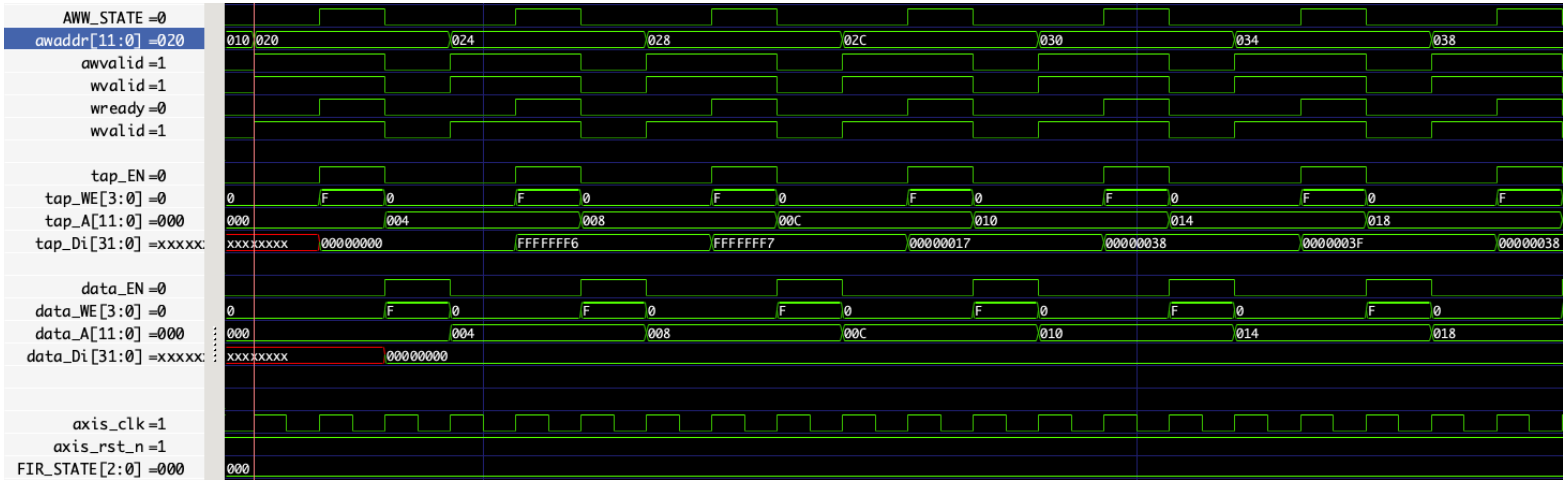
五、Waveform

a. FIR\_STATE

當 ap\_start，FIR\_STATE 切換到 WORK，當 ap\_done 被成功接收後，回到 IDLE

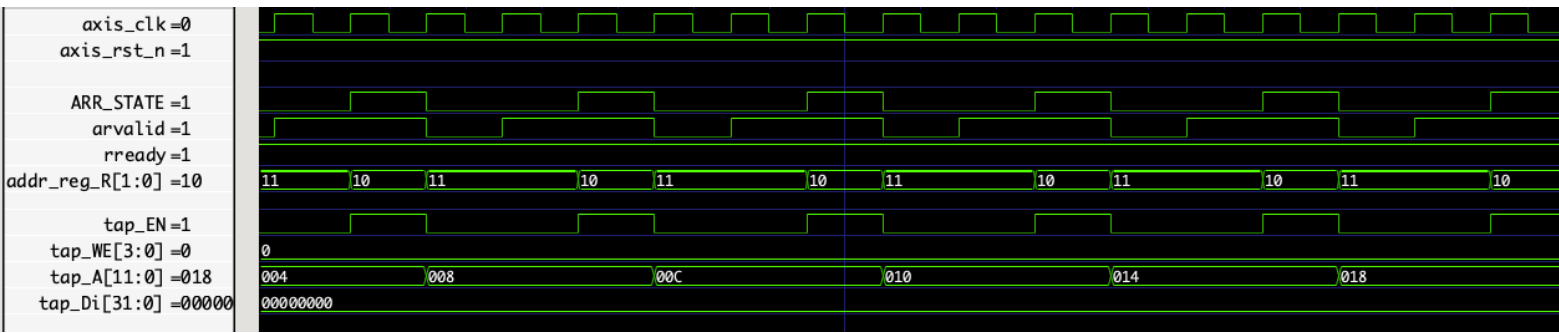


b. AXI-AW/W



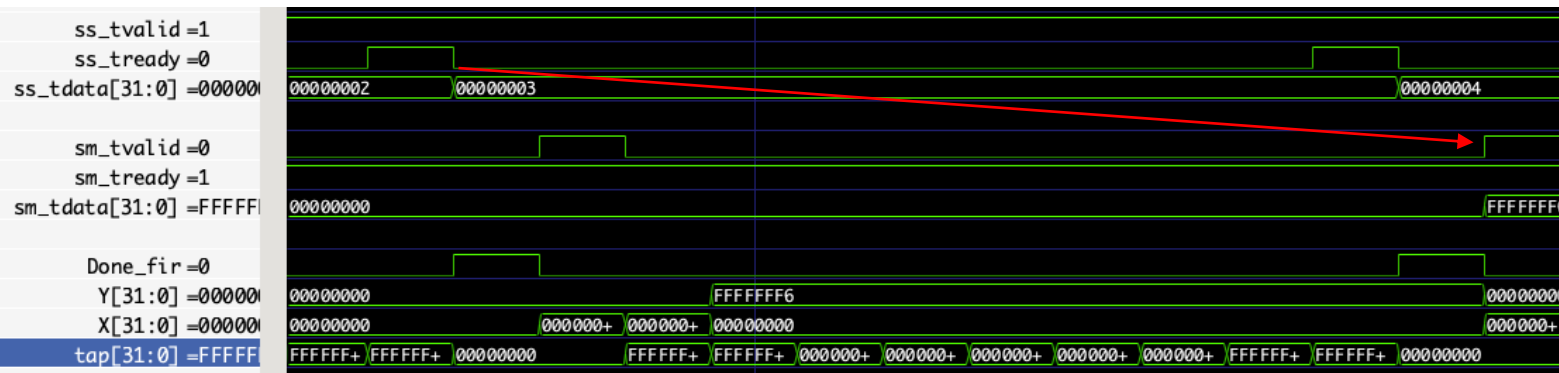
該波形圖說明了接收 tap 並且存入 tap ram，同時順帶在 data ram 中存入 0 完成初始化。

### c. AXI-AR/R



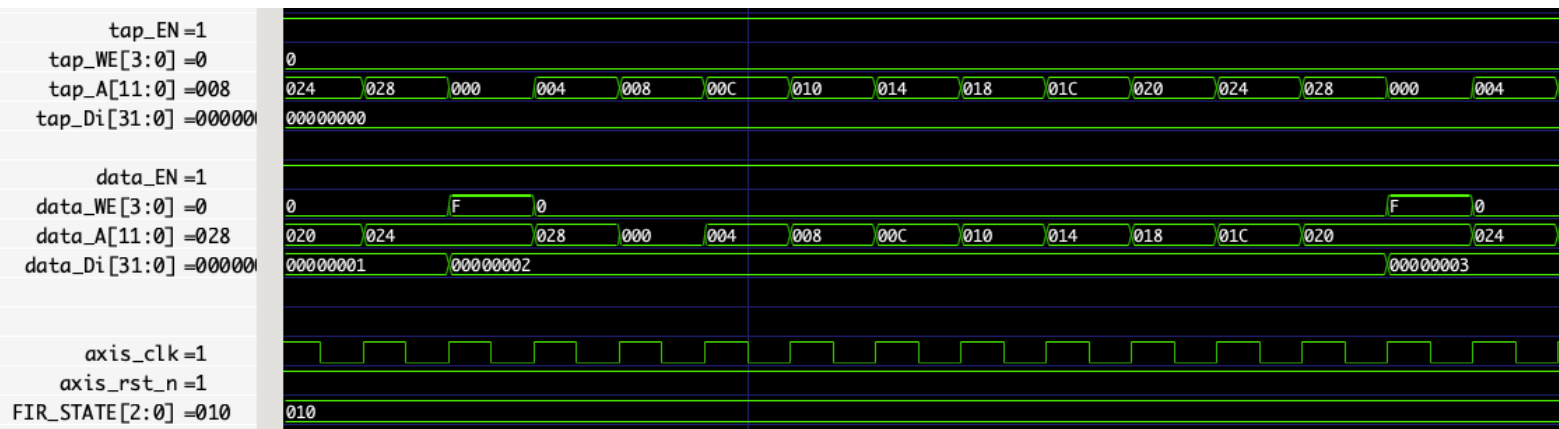
此波形圖描述透過 axi-lite 讀取 tap ram 中的數值，檢驗 tap 數值是否正確

### d. Stream Slave / Master



此波形圖描述透過 stream-slave 接收資料、運算，再透過 stream-master 將資料輸出的過程。

### e. RAM Address



這張波形圖描述進行 FIR 運算時，兩個 RAM 的輸出以及地址依序為何。