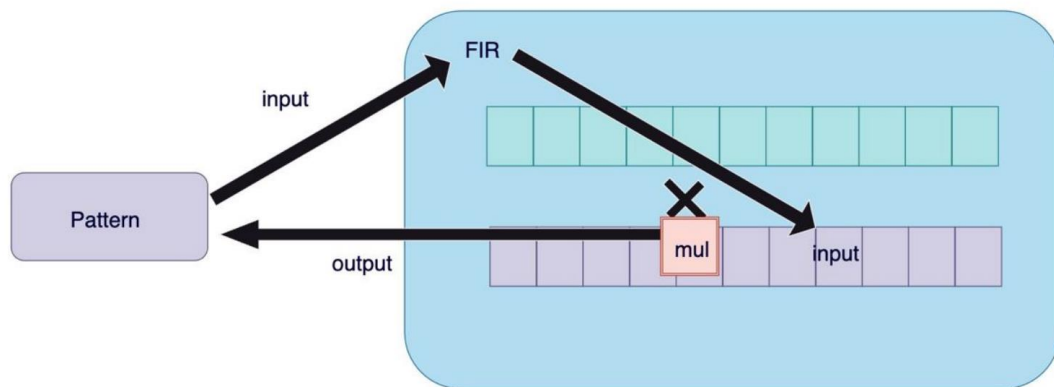
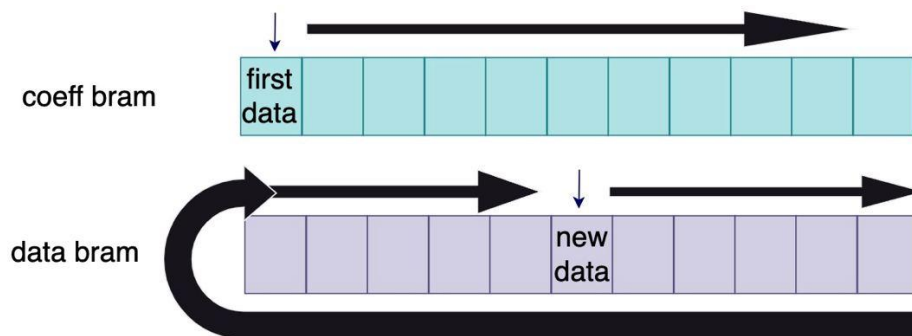


Dataflow:

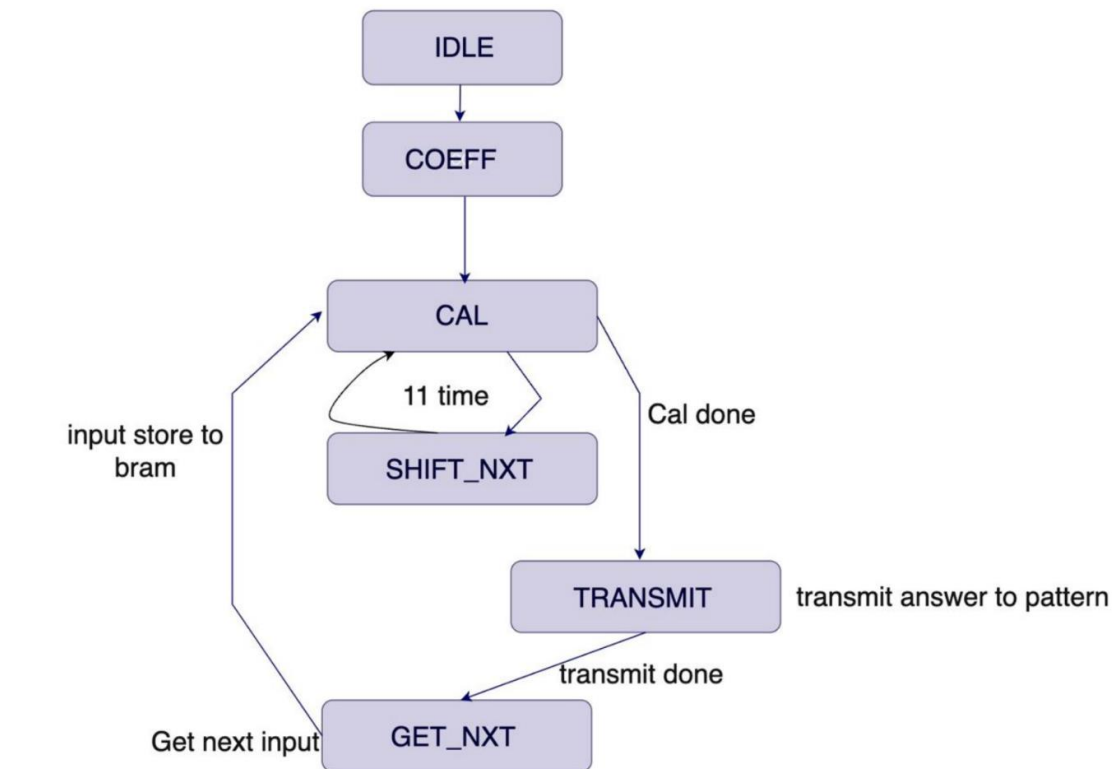


Bram operation



我使用兩個 **bram11** 來分別存 **coefficient** 和 **data**。Coefficient 的部分由於值不會變化，所以比較簡單。而 **data bram** 的部分因為每次新 **data** 都會被存在新的位置，所以用一個 **new data address pointer** 來代表最新的值的 **addr**。再進來新的一個值，就要再往下一個 **addr**。而每次要算出一個結果，就要使用 **address shift register** 取 **bram** 值。取一個值要 **2cycles**，所以算出一筆資料需要 $11 \times 2 = 22\text{cycles}$ 。

FSM:



FSM: 我沒有用 pipeline 加速,由於之後的 lab 還會用到這個 ip,所以我希望他的 state 可以很乾淨,debug 起來不會很麻煩(也是因為還不確定後面的 lab 會如何驅動這個 ip)。所以 state 跟 state 之間劃分得很乾淨,當然,隨之而來的弊端就是 latency 會比較長。各個 state 大致上在做以下的事情:

IDLE: reset 後或是 ap_done 拉起來,就會進入到 idle。Idle 1 cycle 後就會進入 get coefficient。

COEFF: 接收 coefficient 並且存在 coeff bram11 中

CAL: 將 coeff bram 的值與 data bram 的值相乘到 accumulation register 中。如果已經是第 11 個數值就進入 transmit。

SHIFT_NXT: 同時將 coefficient bram 和 data bram shift 到下一個 bram address 取值

TRANSMIT: 將算好的一個值傳送給 Pattern。傳好就進入 GET_NXT。

GET_NXT: 取得下一個 data input。並且存在 data bram 中。

AXI:

AXI-stream: 用來接 data。主要使用 ss_tready 和 ss_tvalid 的 handshake 來判斷。而 ss_tlast 是代表最後一個 input 的訊號。

AXI_lite(configure write): 用來傳 coefficient，使用 aw,ar,w,r 等 channel 來與 pattern 溝通。在這個 lab 助教給的 pattern 中，awready 和 arready 這兩個 signal 沒有用到，不知道之後的 lab 會不會用到。

AP:

都是使用 prototcal 的訊號判斷拉起或歸 0。

Ap_start:

```
if( awvalid==1'b1 && awaddr == 'h00 && apidle == 1'b1) begin //testbench program ap_start
    apstart <= 1'b1;
end
```

Ap_idle:

```
else if(cs == S_TRANSM) begin
    if(sm_tlast == 1'b1 && ss_tlast == 1'b1 && sm_tvalid ) begin //transmit last signal
        apidle <= 1'b1;
    end
end
```

Ap_done:

```
if(cs == S_TRANSM) begin
    if(sm_tlast == 1'b1 && ss_tlast == 1'b1 && sm_tvalid ) begin //transmit last signal
        apdone <= 1'b1;
    end
end
```

Resource Utilization:

Memory: 2 Block RAM

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

Multipiler: 1

Detailed RTL Component Info :

+---Adders :

2 Input	32 Bit	Adders := 1
2 Input	12 Bit	Adders := 2
2 Input	4 Bit	Adders := 4
2 Input	2 Bit	Adders := 2

+---Registers :

32 Bit	Registers := 4
12 Bit	Registers := 2
4 Bit	Registers := 7
2 Bit	Registers := 2
1 Bit	Registers := 8

+---Multipliers :

32x32 Multipliers := 1

LUT and FF

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	223	0	0	53200	0.42
LUT as Logic	223	0	0	53200	0.42
LUT as Memory	0	0	0	17400	0.00
Slice Registers	177	0	0	106400	0.17
Register as Flip Flop	177	0	0	106400	0.17
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

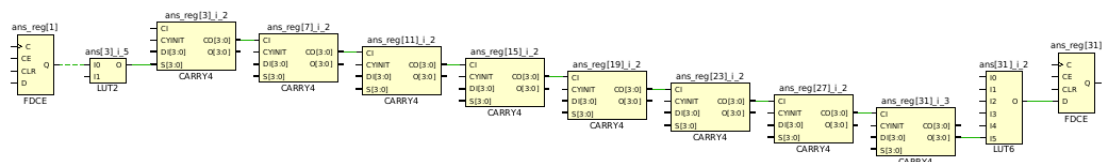
Timing:

就是在取 bram 值並且相乘再存到 accumulated register 的那條 path。

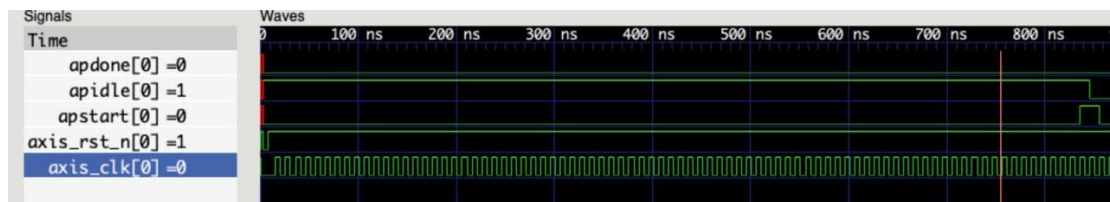
花費 3.761ns 留下了 6.103ns 的 slack

表示最快大約可以 synthesis 到 4ns 左右(留一點 timing 給 APR)

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock
Path 1	6.103	10	11	3	ans_reg[1]/C	ans_reg[31]/D	3.761	2.646	1.115	10.0	axis_clk	axis_clk



Waveform:



From ap_start to ap_done: 10209 cycles for 600 inputs ◦ Average 17.6 cycles for an output

Simulation result:

```
[PASS SM] [Pattern 588] Golden answer: x, Your answer: x
[PASS SM] [Pattern 589] Golden answer: x, Your answer: x
[PASS SM] [Pattern 590] Golden answer: x, Your answer: x
[PASS SM] [Pattern 591] Golden answer: x, Your answer: x
[PASS SM] [Pattern 592] Golden answer: x, Your answer: x
[PASS SM] [Pattern 593] Golden answer: x, Your answer: x
[PASS SM] [Pattern 594] Golden answer: x, Your answer: x
[PASS SM] [Pattern 595] Golden answer: x, Your answer: x
[PASS SM] [Pattern 596] Golden answer: x, Your answer: x
[PASS SM] [Pattern 597] Golden answer: x, Your answer: x
----Done data input(AXI-Stream)----
---- Start Config Read check (ap_idle)----
OK: exp = 0, rdata = 0
---- Done Config Read check (ap_idle)----
[PASS SM] [Pattern 598] Golden answer: x, Your answer: x
-----End the data input(AXI-Stream!) -----
[PASS SM] [Pattern 599] Golden answer: x, Your answer: x
---- All transmit DONE, Start Config Read check (ap_done )----
OK: exp = 2, rdata = 6
---- Ap_done check pass, Start Config Read check (ap_idle )----
OK: exp = 4, rdata = 6
---- Ap_done check pass, Ap_idle check pass ----
-----Congratulations! Pass-----
```