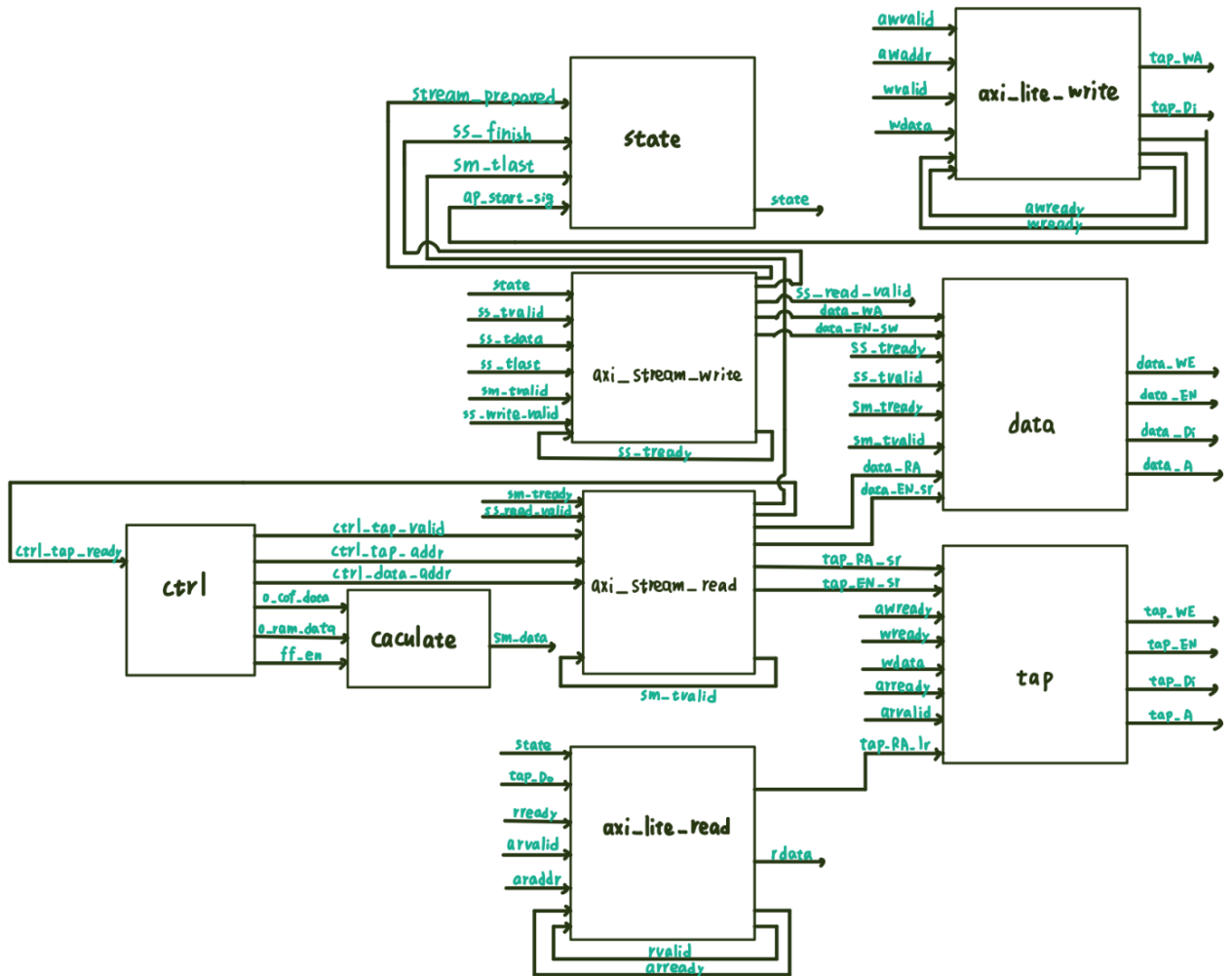


# Lab#3 Report

台科大 M11207002 陳泊佑

- Block diagram



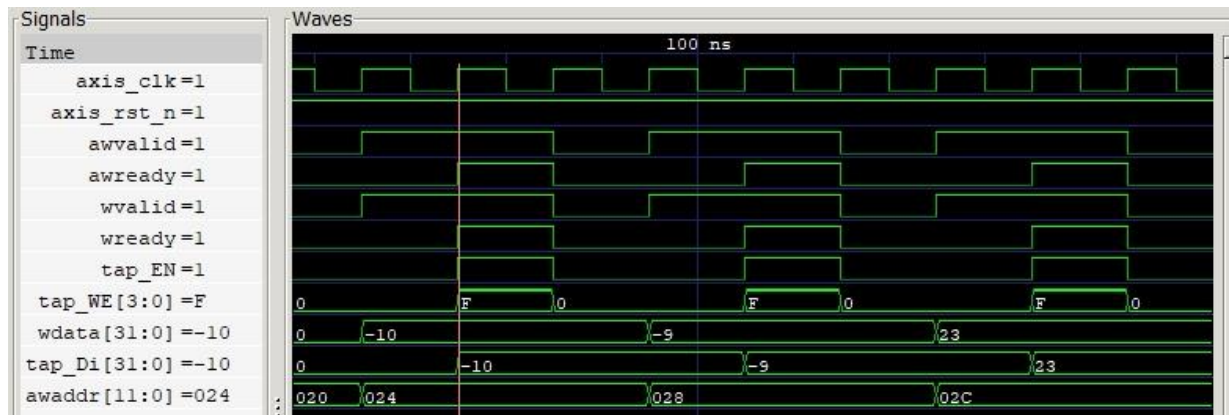
## ● Describe operation

### I. How to receive data-in and tap parameters and place into SRAM

#### Tap-parameters

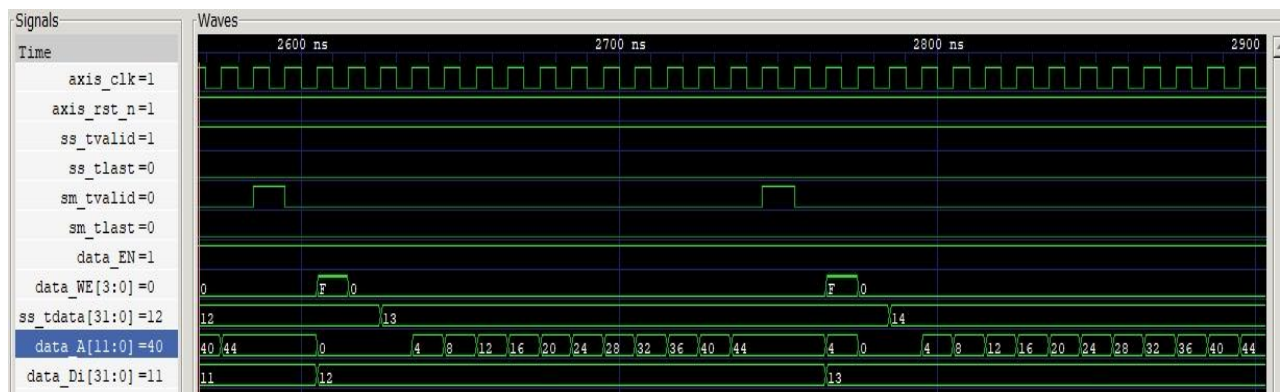
當 awvalid 跟 wvalid 等於 1 時，準備要把 tap 寫入。

當 tap\_EN=1 與 tap\_WE=4'hF 時，會將 wdata 寫進去 awaddr 的位置。



#### data-in

input 透過 stream 傳進來，並先用一個 fifo 來存取 data，要讀資料時，當要讀 data 時，data\_EN=1 與 data\_WE=4'hF，就會開始將 ss\_tdata 寫進去 data\_A 的位置。



## II. How to access shiftram and tapRAM to do computation

為了不動讀取到的 RAM 位置，這裡選擇控制 tap\_addr 跟 data\_addr 來讓它們相乘，下面以 data:1,2,3,4,5 以及 tap:-1,3,-1 來做示範

clock

data: 1,2,3,4,5

tap: -1,3,-1

clock 1

1 0 0  
0 1 2

data-ram-address

0 1 2

tap-ram-address

-1 3 -1

0 1 2

clock 1

1 2 0  
0 1 2

clock 1

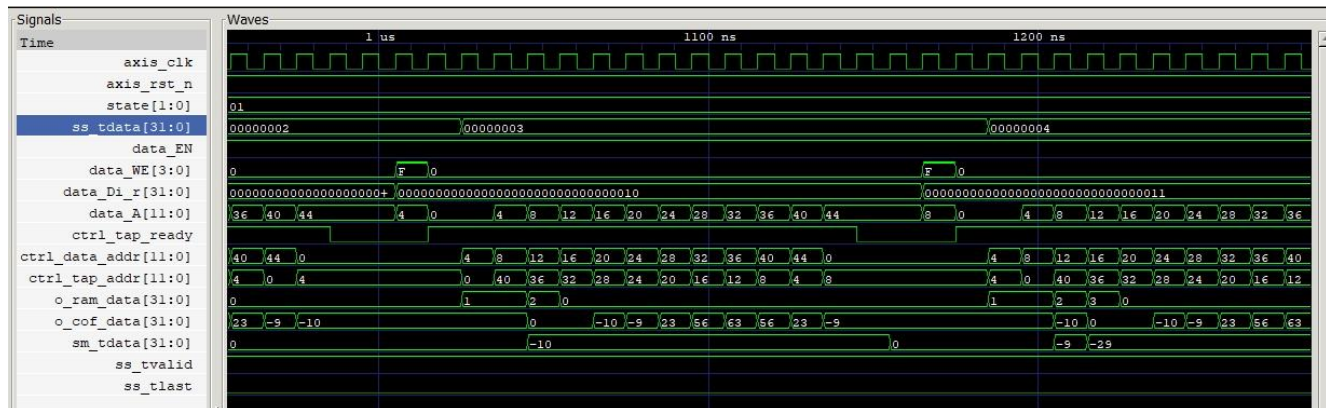
1 2 3  
2 1 0

clock 1

4 2 3  
0 1 2

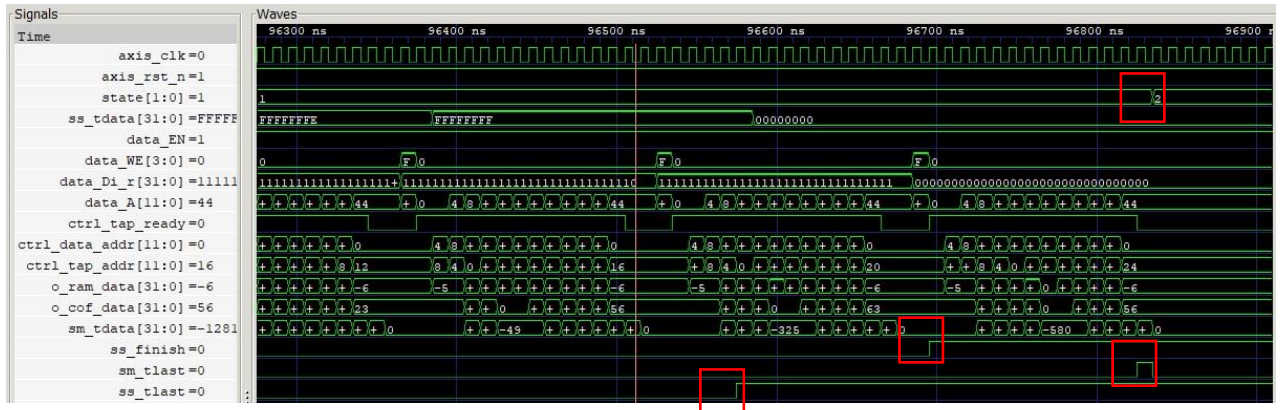
clock 1

5 4 3  
2 1 0



### III. How ap\_done is generated?

當 ss\_tlast 和 sm\_tlast 都為 1 後，ap\_done=2 跳起來，故這邊會把 state = 2'b10。



//code

```
parameter ap_done = 2;

reg [1:0] state;
wire stream_prepared;
reg ap_start_sig;
reg sm_tlast_r;
reg ss_finish_r;
wire ss_finish;
//state
always @(posedge axis_clk) begin
    if (!axis_rst_n) begin
        state <= ap_idle;
    end
    else begin
        case(state)
            ap_idle: begin
                if(stream_prepared && ap_start_sig)
                    state <= ap_start;

                sm_tlast_r <= 0;
            end

            ap_start: begin
                if(ss_finish && sm_tlast)
                    state <= ap_done;

                if(ss_finish && !ctrl_tap_valid)
                    sm_tlast_r <= 1;
                else
                    sm_tlast_r <= 0;
            end
        end
    end
end
```

## ● Resource usage

Utilization      Post-Synthesis   Post-Implementation			
Graph   Table			
Resource	Estimation	Available	Utilization...
LUT	287	53200	0.54
FF	310	106400	0.29
DSP	3	220	1.36
IO	329	125	263.20
BUFG	1	32	3.13

## ● Timing report

### I. Try to synthesize the design with maximum frequency

Tcl Console   Messages   Log   Reports   Design Runs   Timing x			
Clock Summary			
Name	Waveform	Period (ns)	Frequency (MHz)
axis_clk	{0.000 6.000}	12.000	83.333

### II. Slack

Tcl Console   Messages   Log   Reports   Design Runs   Timing x			
Design Timing Summary			
Setup      Hold      Pulse Width			
Worst Negative Slack (WNS): 0.586 ns		Worst Hold Slack (WHS): 0.070 ns	Worst Pulse Width Slack (WPWS): 5.500 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: 500		Total Number of Endpoints: 500	Total Number of Endpoints: 311
All user specified timing constraints are met.			

### III. longest path

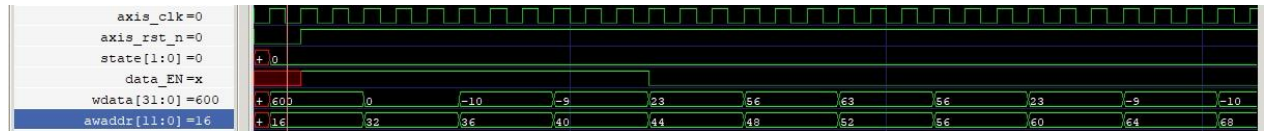
Tcl Console   Messages   Log   Reports   Design Runs   Timing x												?		_		□												
Intra-Clock Paths - axis_clk - Hold																												
General Inform												Name						Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clk
Timer Settings												Path 11	0.070	1	2	3	sm_tdata_r_reg[10]C	sm_tdata_r_reg[11]D	0.328	0.239	0.089	0.0	axis_clk					
Design Timing												Path 12	0.070	1	2	3	sm_tdata_r_reg[12]C	sm_tdata_r_reg[13]D	0.328	0.239	0.089	0.0	axis_clk					
Clock Summar												Path 13	0.070	1	2	3	sm_tdata_r_reg[14]C	sm_tdata_r_reg[15]D	0.328	0.239	0.089	0.0	axis_clk					
Methodology Si												Path 14	0.070	1	2	3	sm_tdata_r_reg[16]C	sm_tdata_r_reg[17]D	0.328	0.239	0.089	0.0	axis_clk					
Check Timing												Path 15	0.070	1	2	3	sm_tdata_r_reg[18]C	sm_tdata_r_reg[19]D	0.328	0.239	0.089	0.0	axis_clk					
Intra-Clock Pat												Path 16	0.070	1	2	3	sm_tdata_r_reg[0]C	sm_tdata_r_reg[1]D	0.328	0.239	0.089	0.0	axis_clk					
axis_clk												Path 17	0.070	1	2	3	sm_tdata_r_reg[20]C	sm_tdata_r_reg[21]D	0.328	0.239	0.089	0.0	axis_clk					
Setup C												Path 18	0.070	1	2	3	sm_tdata_r_reg[22]C	sm_tdata_r_reg[23]D	0.328	0.239	0.089	0.0	axis_clk					
Timing Summary - timing_1																												

Timing									
Intra-Clock Paths - axis_clk - Pulse Width									
Clock Summary (1)	Check Type	Corner	Lib Pin	Reference Pin	Required	Actual	Slack	Location	Pin
Methodology Summ	High Pulse Width	Fast	FDRE/C	n/a	0.500	6.000	5.500		FSM_onehot_state_reg1J/C
> Check Timing (326)	High Pulse Width	Fast	FDRE/C	n/a	0.500	6.000	5.500		FSM_onehot_state_reg2J/C
▼ Intra-Clock Paths	High Pulse Width	Fast	FDRE/C	n/a	0.500	6.000	5.500		ap_start_sig_reg/C
▼ axis_clk	High Pulse Width	Fast	FDRE/C	n/a	0.500	6.000	5.500		aready_r_reg/C
Setup 0.586	Low Pulse Width	Fast	FDSE/C	n/a	0.500	6.000	5.500		FSM_onehot_state_reg0J/C
Hold 0.070	Low Pulse Width	Fast	FDRE/C	n/a	0.500	6.000	5.500		FSM_onehot_state_reg1J/C
Pulse Width	Low Pulse Width	Fast	FDRE/C	n/a	0.500	6.000	5.500		FSM_onehot_state_reg2J/C
Inter-Clock Paths	Low Pulse Width	Fast	FDRE/C	n/a	0.500	6.000	5.500		ap_start_sig_reg/C

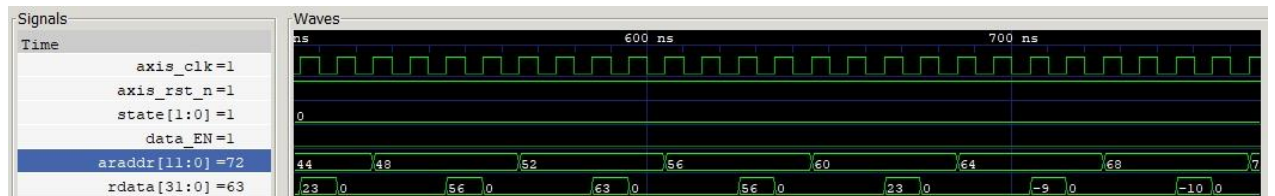


## ● Simulation Waceform

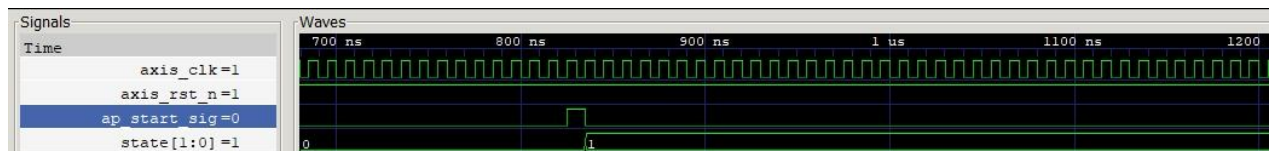
1.  $ap\_idle = 1 \rightarrow state = 0$ ，開始輸入 tap。



2. testbench 開始檢查 tap。



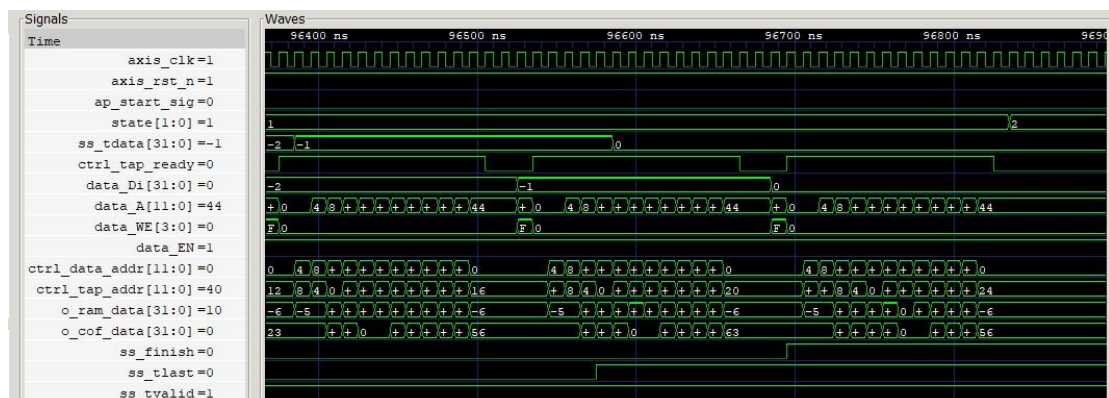
3. 收到  $ap\_star\_sig$  訊號開始做運算， $state = 2'b01$ 。



4.  $state = 2'b01$  後，會開始將  $ss\_tdata$  寫入至 RAM，並且  $ctrl\_tap\_ready = 1$  (下圖中紅色框框)，接著控制模組開始輸出對應的  $tap\_addr$  與  $data\_addr$ ，之後再 RAM 讀出數值做相乘。



5. 當  $ss\_tlast$  和  $sm\_tlast$  都為 1 後， $ap\_done=2$  跳起來，故這邊會把  $state = 2'b10$ 。



- **observed & learned**

Lab3 的作業很具有挑戰性，我們花了許多天討論才討論出處理資料的方式，特別是把資料從 RAM 讀出來做相乘，這邊需要寫控制訊號來控制，我覺得也是最難的部分，所幸最後有想出辦法，但最後用到的 LUT 以及 FF 也還是很多，還要另找方法來解決，上課時有聽老師分享其他學校同學的方法，感覺可以試試看!!