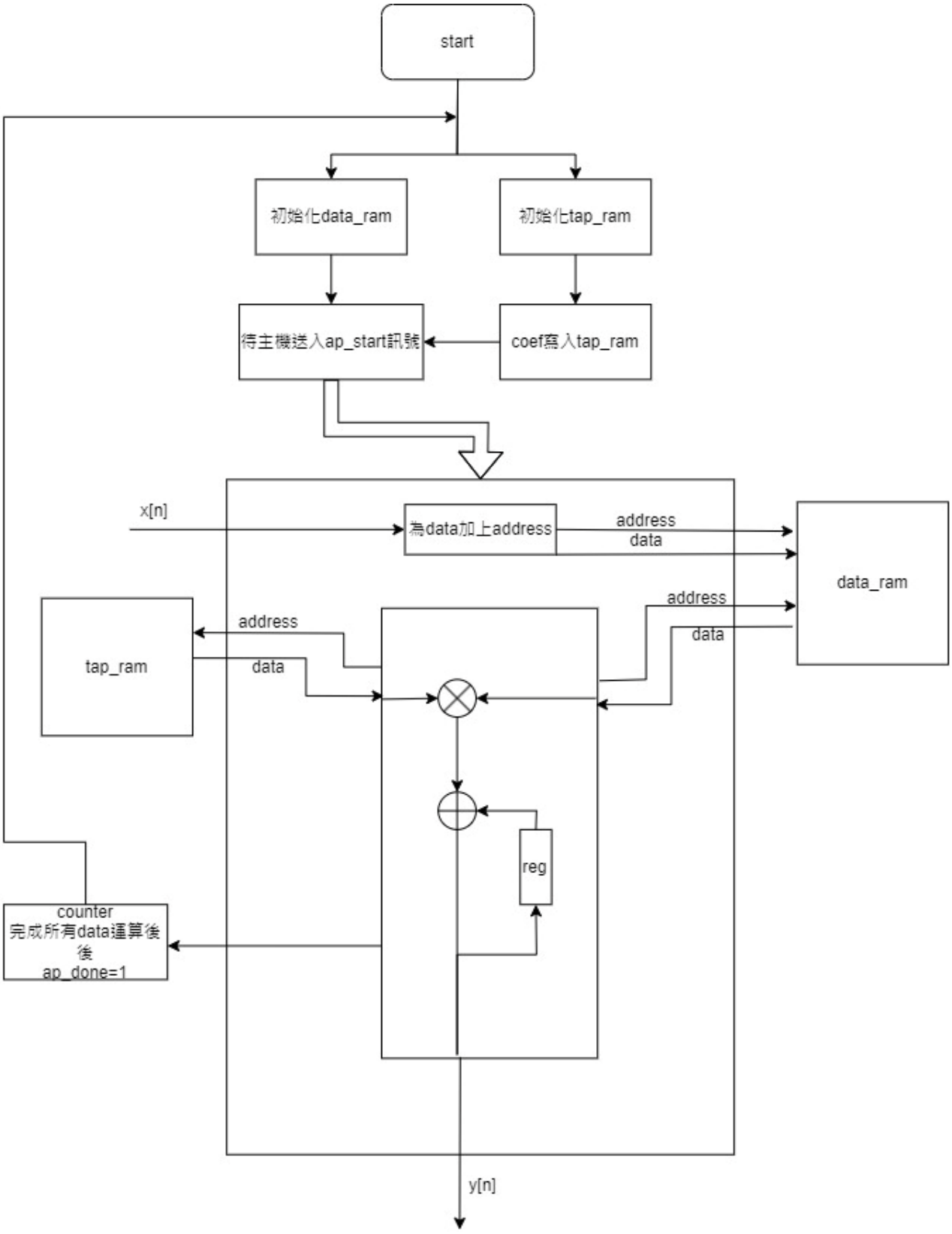
SOC LAB

Lab3 report

ID:R12921076
Name:郭桓愷

總流程圖


start
初始化data_ram
初始化tap_ram
待主機送入ap_start訊號
coef寫入tap_ram
x[n]
為data加上address
address
data
data_ram
address
data
tap_ram
address
data
reg
counter
完成所有data運算後
後
ap_done=1
y[n]

Design conception:

- tap_ram & data_ram皆使用bram11

1. 初始化tap_ram&datat_ram(將0填入)

2. Tap parameters寫入bram:
當讀取到的awaddr位於0x20-FF則判斷資料為Tap parameter。
將address轉換為Bram的address後(awaddr-32)連同data送入bram
此時拉高we進行寫入
完成後回傳write_done訊號至判斷address的block告知已完成寫入
並拉低WE，拉高awready&wready，等待下一筆資料。

3. 待主機發送ap_start

4. Data寫入bram:
當ap_start拉高後開始運作，將data賦予address後送入bram。
此時拉高we進行寫入，待fir完成一次運算後，拉高ss_tready&拉低we，
等待下一筆資料。

5. Fir運算:
進行一次FIR運算需11筆data及coef，我的設計是每一次的iteration都會發出一條tap_ram的address及一條data_ram的address，讀取對應的data後進行運算，並將相乘結果與reg內的上筆數據相加後再存進reg內，完成11次iteration後輸出結果至sm port，並將下一筆data寫入bram。

6. 完成所有data運算:
完成一次fir運算counter會記數，當達datalength數量時，發送finish的狀態，此時會拉高sm_tlast，並將ap_done=1寫入rdata，address 0x00處。且進行tap_ram&data_ram初始化，完成後將所有狀態與port回歸初始條件。

7. Run for three times

Fir shift address conception:

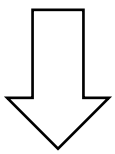
Tap_bram:

Block	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
coef	0	1	2	3	4	5	6	7	8	9	10

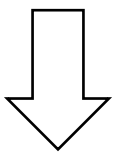
data_bram:

data=x[n]

Block	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n	0	1	2	3	4	5	6	7	8	9	10



Block	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n	11	12	13	14	15	16	17	18	19	20	21



Fir shift address conception:

Tap_parameter=coef[x]
data=x[n]

運算時每個iteration
result=resuil+temp;
Temp[i]=coef[i]*data_block[10-i];
i=i+1

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
i	0	1	2	3	4	5	6	7	8	9	10
x	10	9	8	7	6	5	4	3	2	1	0
n											

Y[10]

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
i	0	1	2	3	4	5	6	7	8	9	10
x	21	20	19	18	17	16	15	14	13	12	11
n											

Y[21]

Fir shift address conception:

i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
x	0	1	2	3	4	5	6	7	8	9	10
第一筆data計算	10	9	8	7	6	5	4	3	2	1	0
	0	10	9	8	7	6	5	4	3	2	1
	1	0	10	9	8	7	6	5	4	3	2
	2	1	0	10	9	8	7	6	5	4	3
						⋮					
	9	8	7	6	5	4	3	2	1	0	1
	10	9	8	7	6	5	4	3	2	1	0
	0	10	9	8	7	6	5	4	3	2	1

Tap_ram地址:
Tap_ram_A<=i<<2

Data_ram地址:
每筆新data{
Z=j
每筆iteration{
data_ram_A<=(z-i)<<2
當zi=0時z=11+j}
J=j+1;j=11時歸零
}

Fir shift address conception:

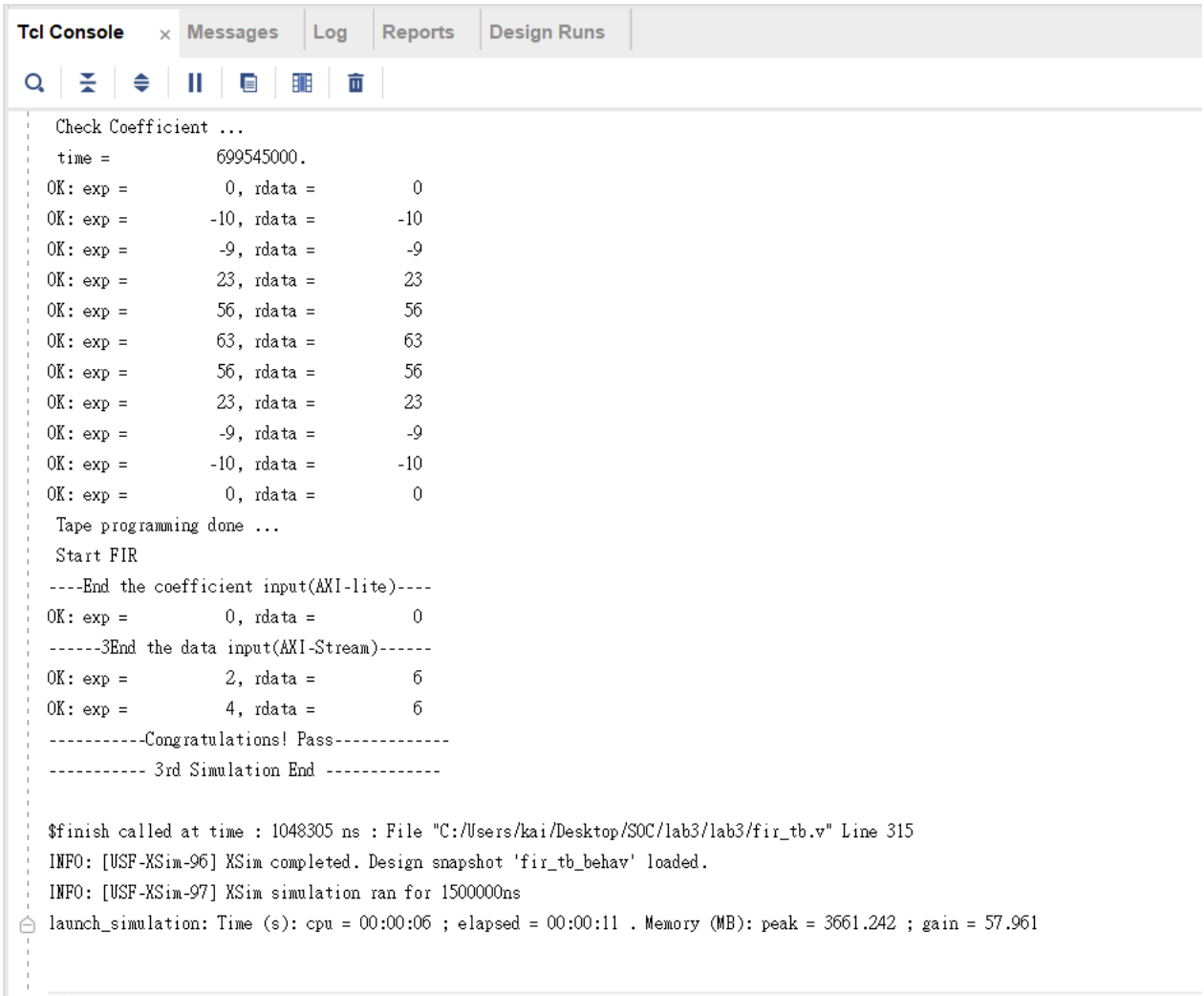
	i	0	1	2	3	4	5	6	7	8	9	10
J=0	z-i	0	10	9	8	7	6	5	4	3	2	1
	z	0	11	11	11	11	11	11	11	11	11	11
J=1	z-i	1	0	10	9	8	7	6	5	4	3	2
	z	1	1	12	12	12	12	12	12	12	12	12
	⋮					⋮						
J=10	z-i	10	9	8	7	6	5	4	3	2	1	0
	z	10	10	10	10	10	10	10	10	10	10	10

Z-j → iteration

Data_ram地址:
每筆新data{
Z=j;
J=j+1;
J=10後歸零}

每筆iteration{
data_ram_A<=(z-i)<<2
當zi=0時z=11+j}

result


Tcl Console
Messages
Log
Reports
Design Runs
Check Coefficient ...
time = 699545000.
OK: exp = 0, rdata = 0
OK: exp = -10, rdata = -10
OK: exp = -9, rdata = -9
OK: exp = 23, rdata = 23
OK: exp = 56, rdata = 56
OK: exp = 63, rdata = 63
OK: exp = 56, rdata = 56
OK: exp = 23, rdata = 23
OK: exp = -9, rdata = -9
OK: exp = -10, rdata = -10
OK: exp = 0, rdata = 0
Tape programming done ...
Start FIR
----End the coefficient input(AXI-lite)----
OK: exp = 0, rdata = 0
------3End the data input(AXI-Stream)------
OK: exp = 2, rdata = 6
OK: exp = 4, rdata = 6
----------Congratulations! Pass----------
---------- 3rd Simulation End ----------
$finish called at time : 1048305 ns : File "C:/Users/kai/Desktop/SOC/lab3/lab3/fir_tb.v" Line 315
INFO: [USF-XSim-96] XSim completed. Design snapshot 'fir_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1500000ns
launch_simulation: Time (s): cpu = 00:00:06 ; elapsed = 00:00:11 . Memory (MB): peak = 3661.242 ; gain = 57.961

Synthesis report

```
---------------------------------------------------------------------------------
Start RTL Component Statistics
---------------------------------------------------------------------------------
Detailed RTL Component Info :
+---Adders :
       2 Input   32 Bit       Adders := 5
       2 Input   12 Bit       Adders := 6
       3 Input   12 Bit       Adders := 1
+---Registers :
                 32 Bit    Registers := 5
                 12 Bit    Registers := 2
                  1 Bit    Registers := 8
+---Multipliers :
                32x32  Multipliers := 1
+---Muxes :
       3 Input   32 Bit        Muxes := 1
       2 Input    1 Bit        Muxes := 13
---------------------------------------------------------------------------------
Finished RTL Component Statistics
---------------------------------------------------------------------------------
```

```
Report BlackBoxes:
+-+--------------+----------+
| |BlackBox name |Instances |
+-+--------------+----------+
+-+--------------+----------+

Report Cell Usage:
+------+-------+------+
|      |Cell   |Count |
+------+-------+------+
|1     |BUFG   |     1|
|2     |CARRY4 |    21|
|3     |LUT1   |     3|
|4     |LUT2   |    19|
|5     |LUT4   |     3|
|6     |LUT5   |     4|
|7     |LUT6   |     3|
|8     |FDRE   |    70|
|9     |FDSE   |     3|
|10    |IBUF   |     3|
|11    |OBUF   |   169|
+------+-------+------+
---------------------------------------------------------------------------------
Finished Writing Synthesis Report : Time (s): cpu = 00:00:23 ; elapsed = 00:00:24 . Memory (MB): peak = 2535.(
---------------------------------------------------------------------------------
```

Synthesis report

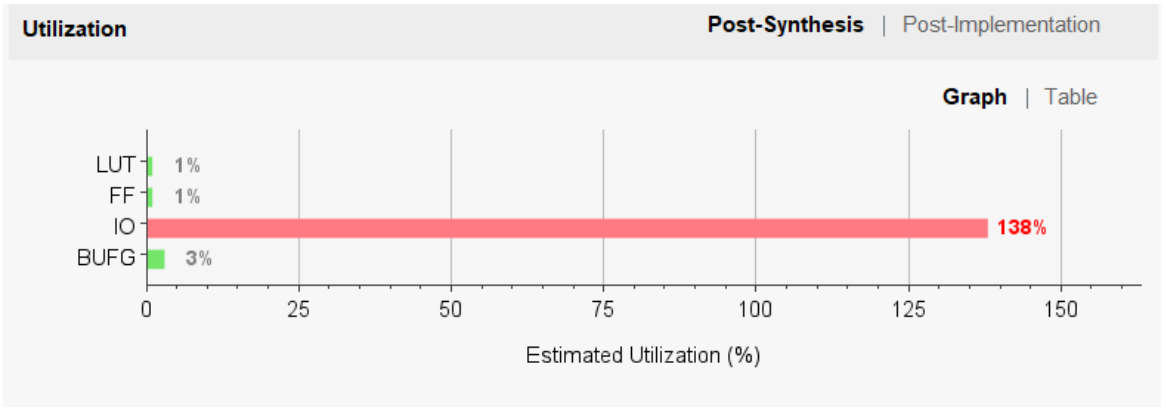
1. Slice Logic

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	30	0	0	53200	0.06
LUT as Logic	30	0	0	53200	0.06
LUT as Memory	0	0	0	17400	0.00
Slice Registers	73	0	0	106400	0.07
Register as Flip Flop	73	0	0	106400	0.07
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

* Warning! The Final LUT count, after physical optimizations and full implementation, is typically lov
Warning! LUT value is adjusted to account for LUT combining.

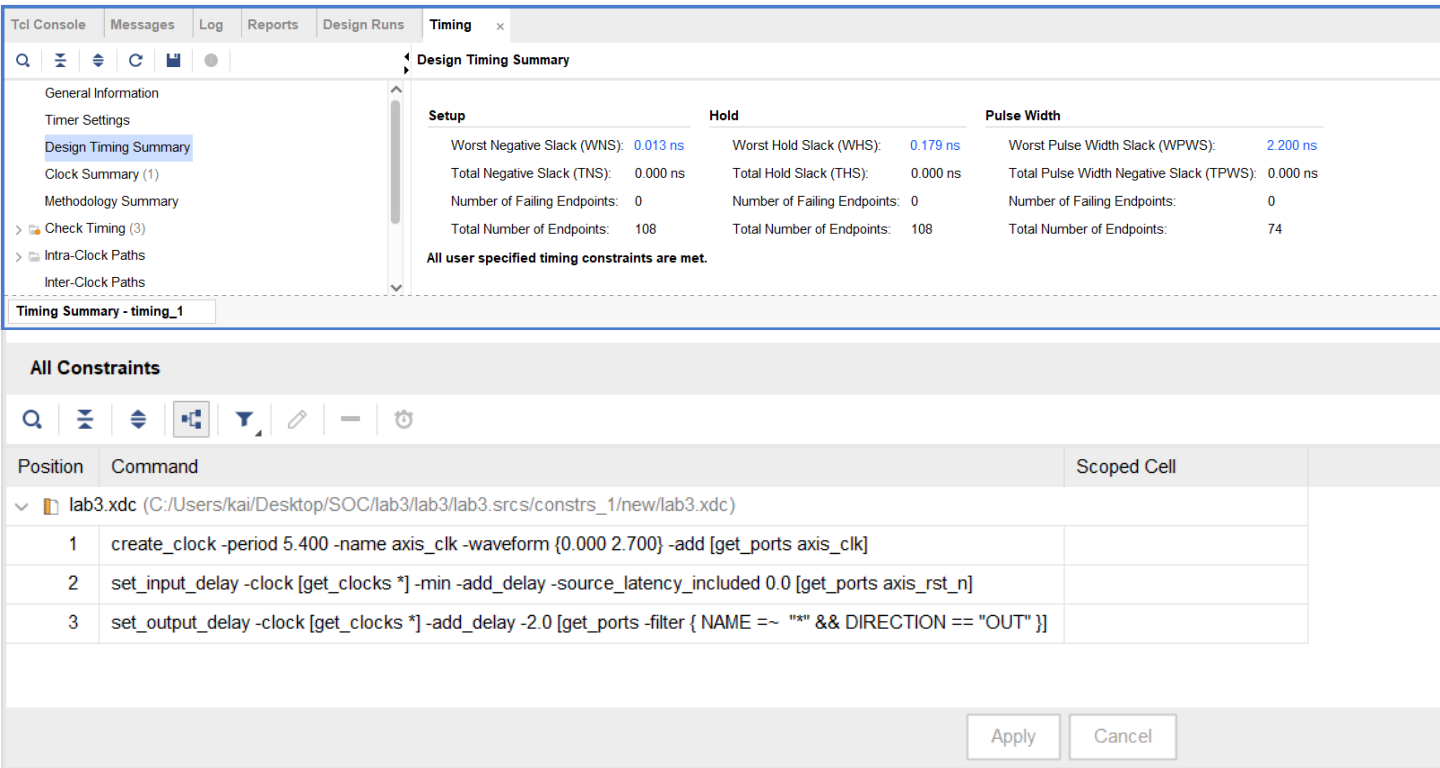
2. Memory

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



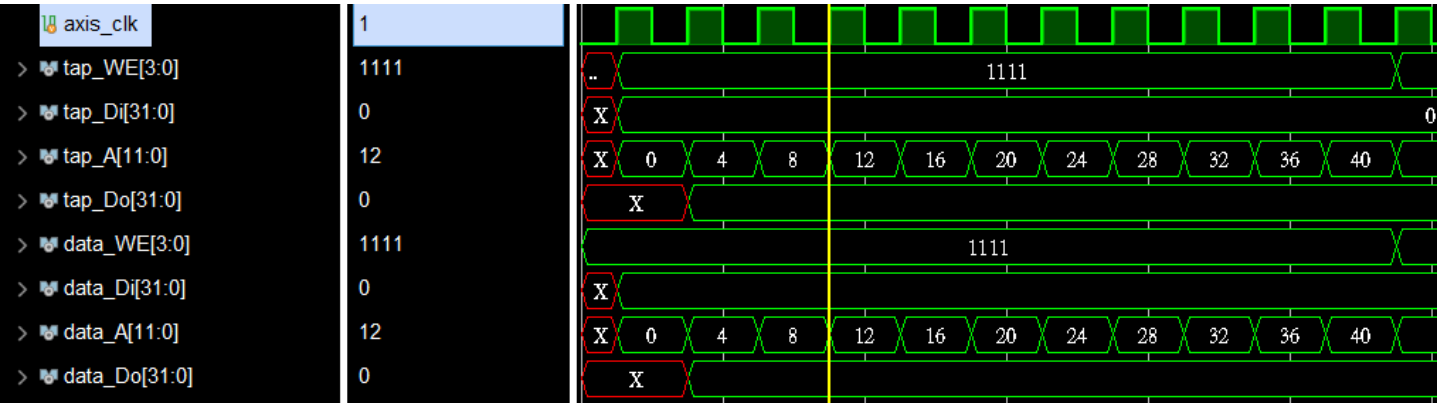
Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	synth_design Complete!												29	73	0	0	0	10/22/23, 4:10 AM	00:00:34	Vivado Synthesis Defaults (Vivado Synthesis 2023)

timing report

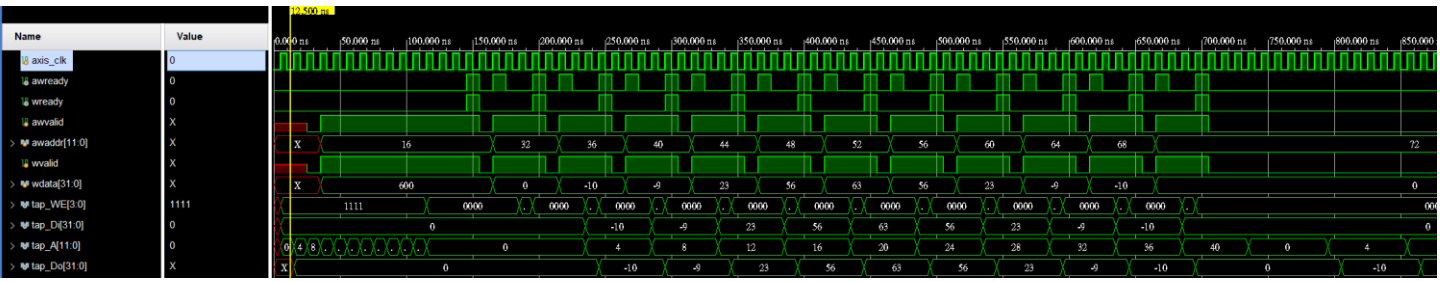


waveform

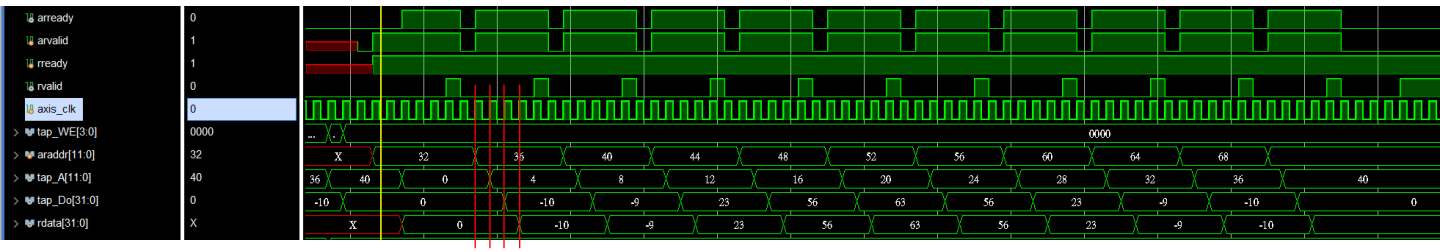
1.tap_ram&data_ram initialization



2.Tap parameter write



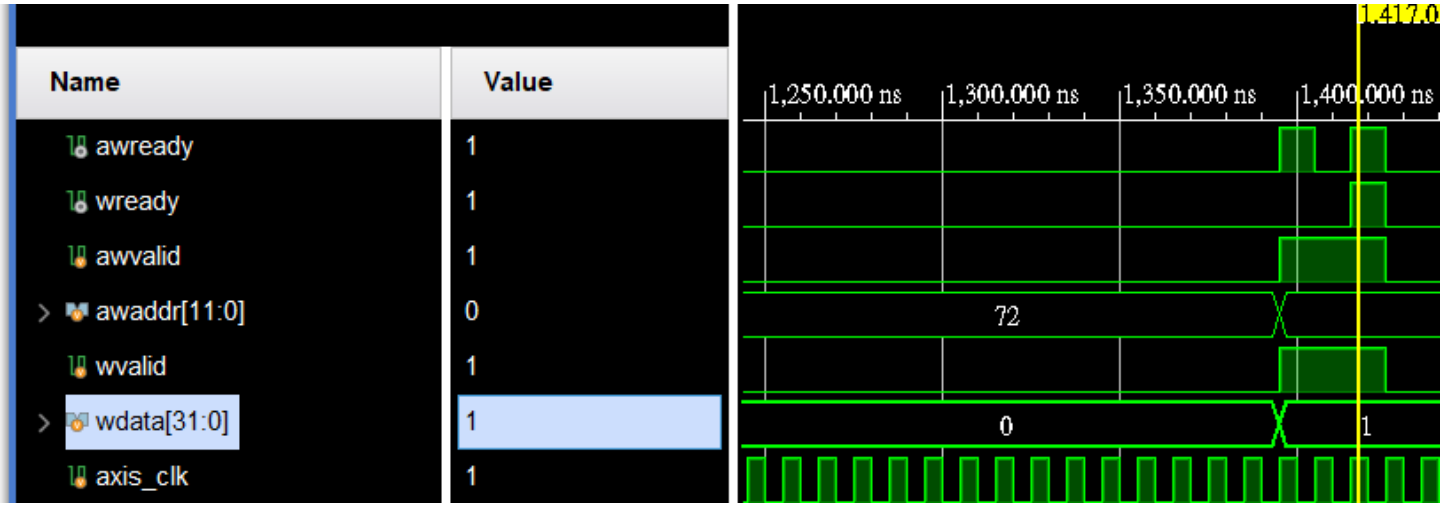
3.Tap parameter read



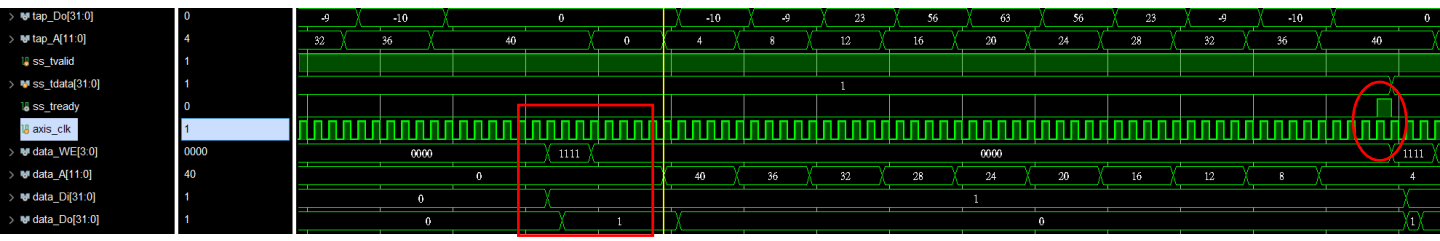
從送入araddr到送出rdata共會有4個clk的延遲，故在設計時有使rvalid delay四個cycle發送以同步信號。

waveform

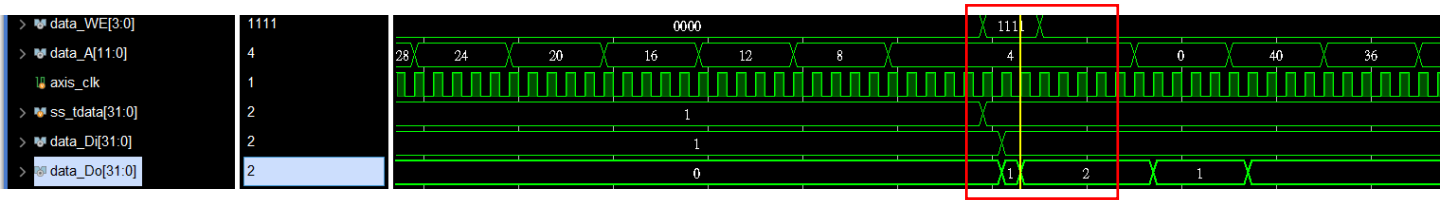
4.Write ap_start



5.Data write



Data寫入後立即進行fir運算，完成FIR運算後READY拉高，進行下一筆data讀取

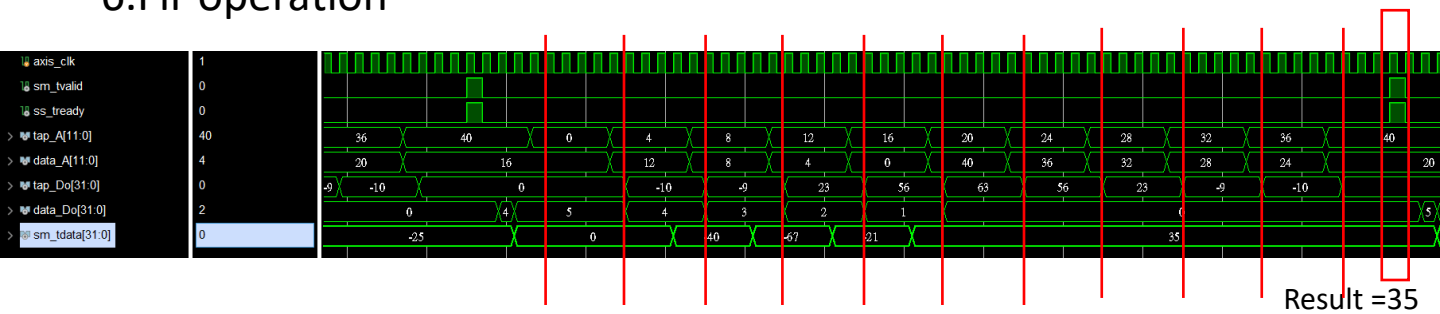


此為第二筆data，從rdata至data_in有1個clk的延遲，因此在此有延長we開啟的時間以確保bram寫入正確的data而不是上筆data。

waveform

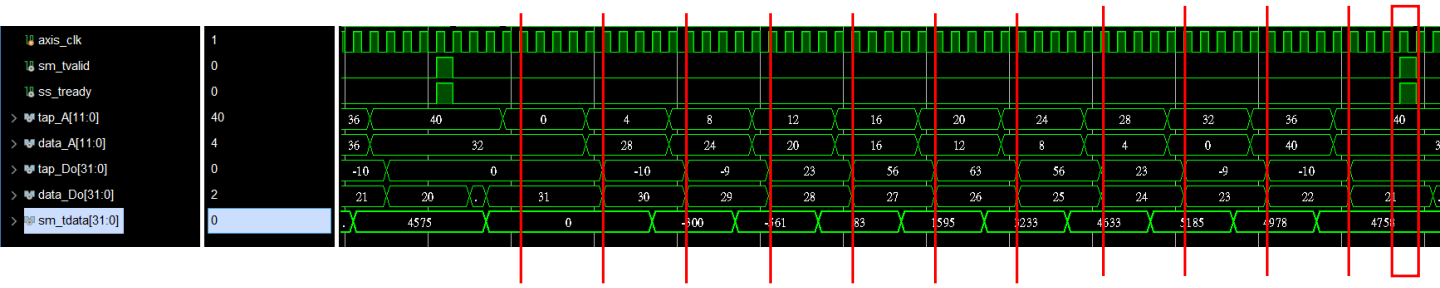
6.Fir operation

Sm_tvalid拉高輸出結果



Result =35

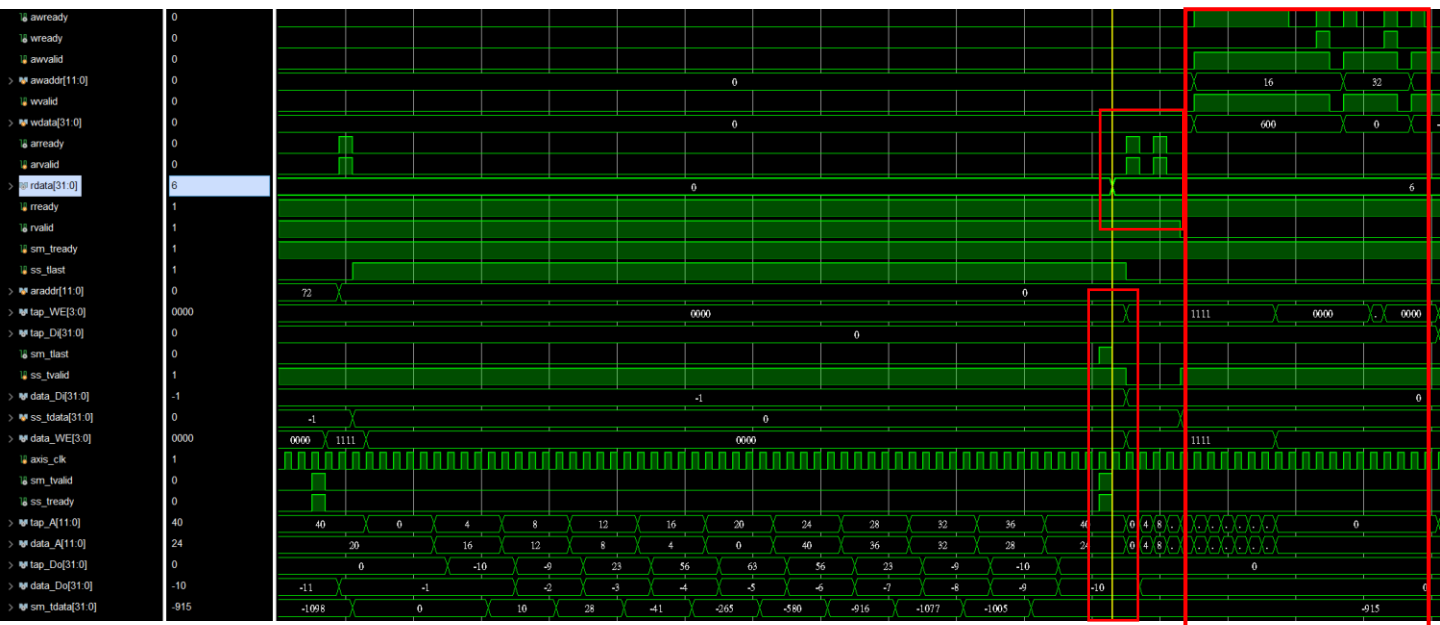
=0*5-10*4-9*3+23*2+56*1+63*0+56*0+23*0-9*0-10*0+0*0=35



Result =4758

=0*31 -10*30 -9*29 +23*28 +56*27 +63*26 +56*25 +23*24 -9*23 -10*22 +0*21=4758

7.ap_done

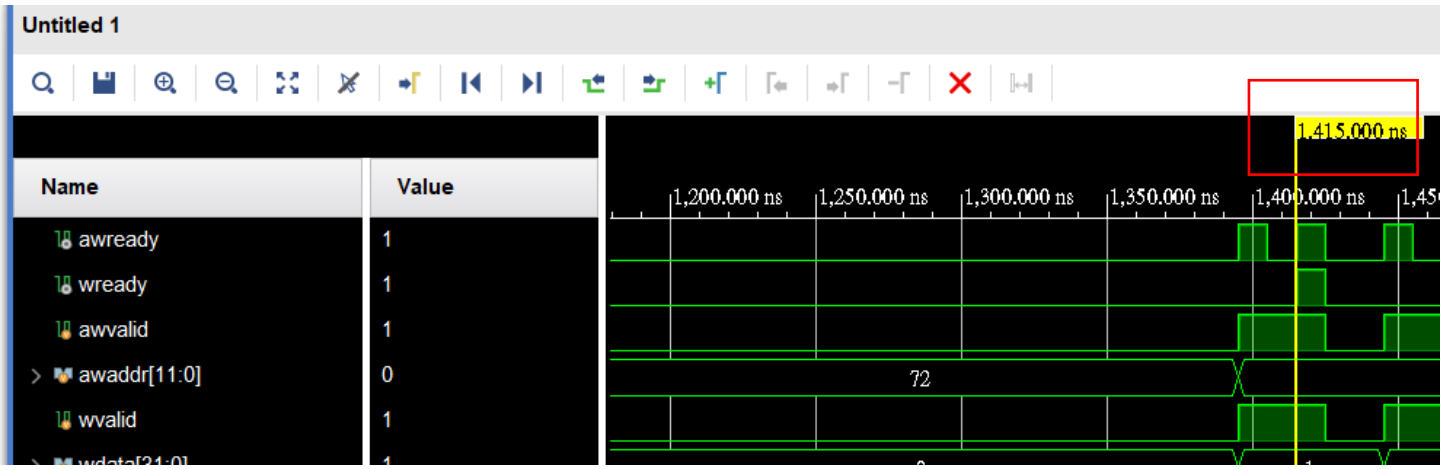


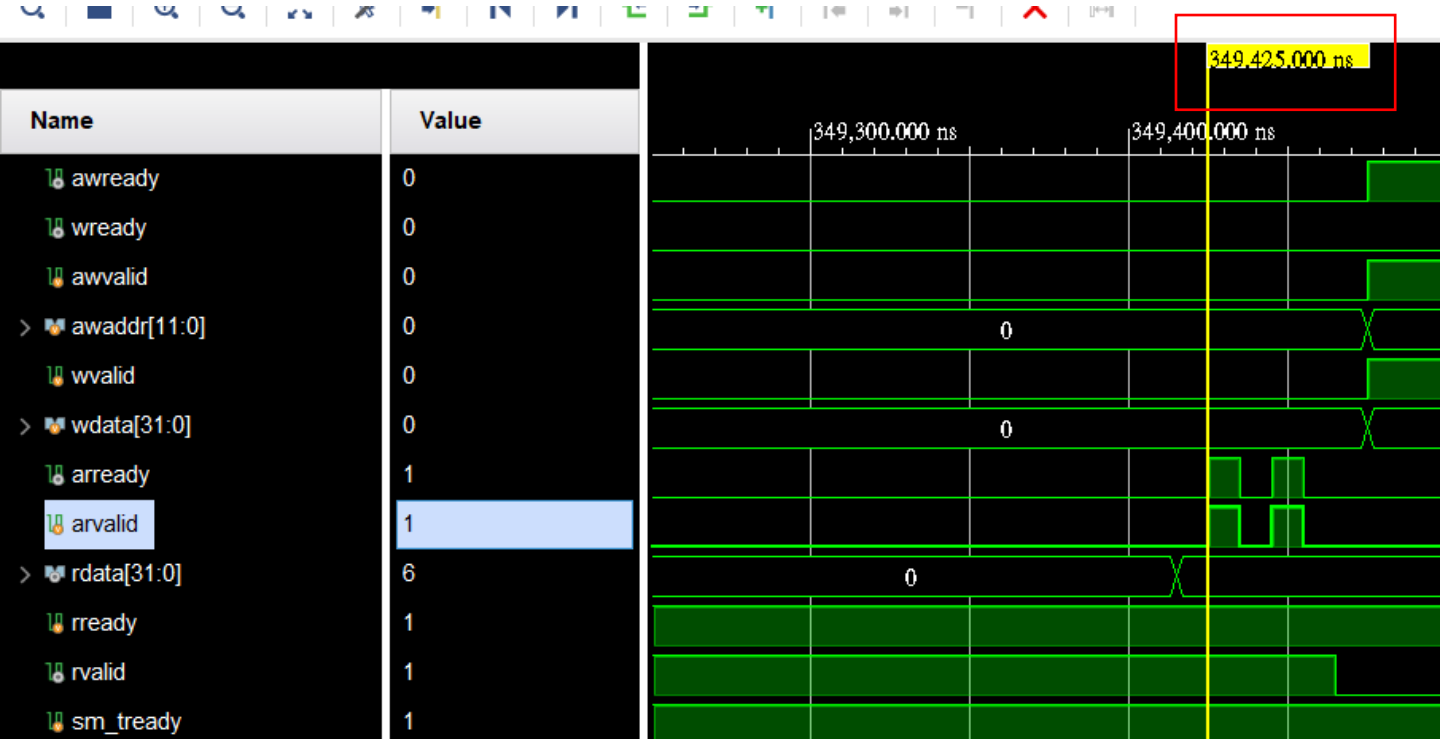
完成後回傳ap_done=1,ap_idle=0。
復原所有狀態

進行第二run

waveform

8. clock cycles from ap_start to ap_done





Start:1415
Done:349425

#Takes 348010 cycles