

VLSI System Design

HW6

Report

系級	電機 113
學號	F64096114
姓名	郭家佑

1. 設計原理(含硬體架構圖)

- general description

non-pipeline CPU 我是利用講義所提供的架構去做修改並延伸，多了 4 個指令，分別為 SUB、J、JSUB、RET，前者可用來相減並比較是否為 0(當作 COMPARE)，後三者則是在遞迴 function 中常會使用到的指令。

IO port 的部分主要是由 CPU 與 IM DM 之間的連接，其他就是由 tb 灌 clk、rst 進來使用，因為設計的 memory 都會有五個 port，所以在建立 instance 時會把不能動的 IR 給一些 const value。

而 CPU 內部的 register 有三大部分，分別為 RFILE 16 個和 return address 64 個(存 jsub pc+1)和 rotate result 33 個外加一個 top ptr(因為 return addr 是 stack structure)。而 RFILE 讀取和寫入都跟 MEMORY 一樣，讀馬上寫下一 cycle，後面三個都是下一個 cycle 才寫入。

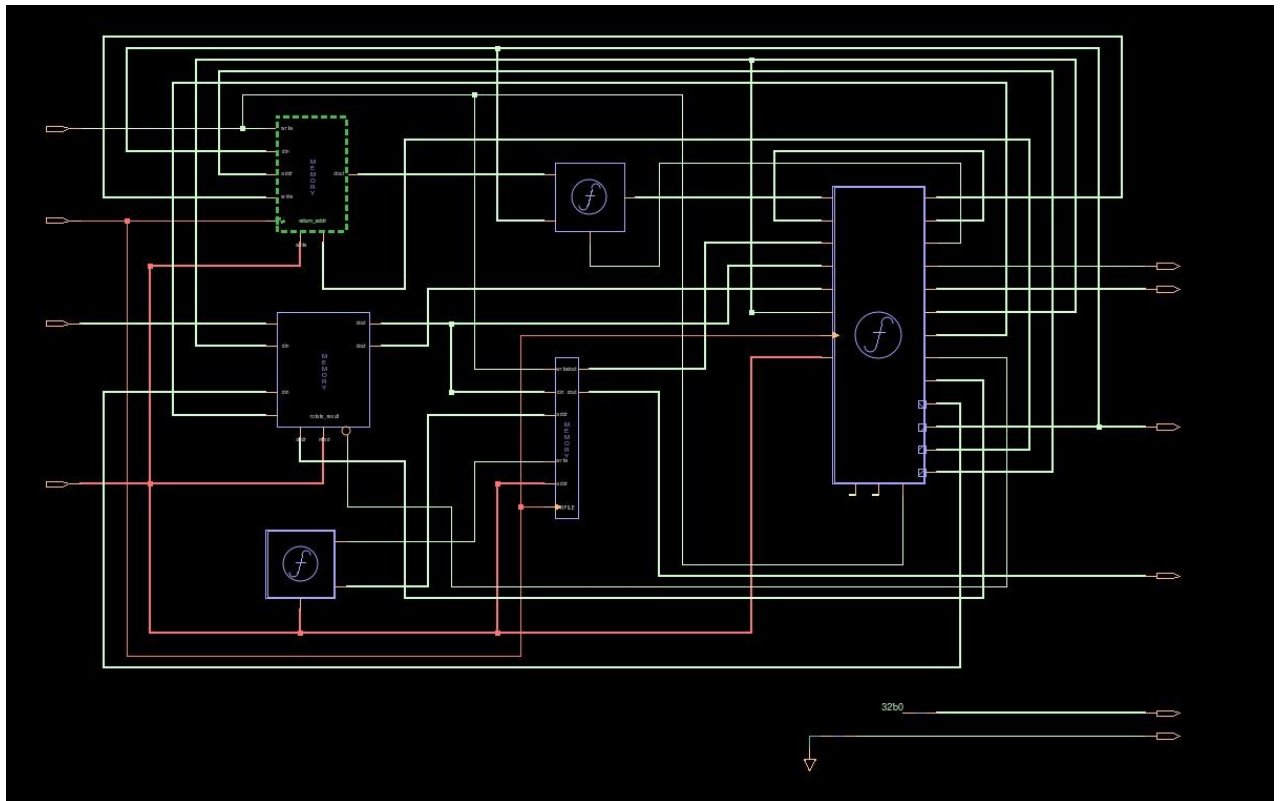
由於改成 single cycle，所以在 BRA 指令必須用到前一 cycle 的 CC，所以我就多加了一個 next_psr 存前一 cycle 的 CC 結果，其他指令就根據要做什麼給什麼訊號。

而 testbench 部分我是建了一個 CPU 和兩個 MEM(IM, DM 但實際上是同一顆)，並將所需要的 ir.prog、data.prog、golden.bin(答案)都讀入指定的位置，並在 reset 後開始執行指令，直到最後一個指令是將 DM 的 4095 位置寫為 FFFFFFFF 就顯示做完了並開始對答案。

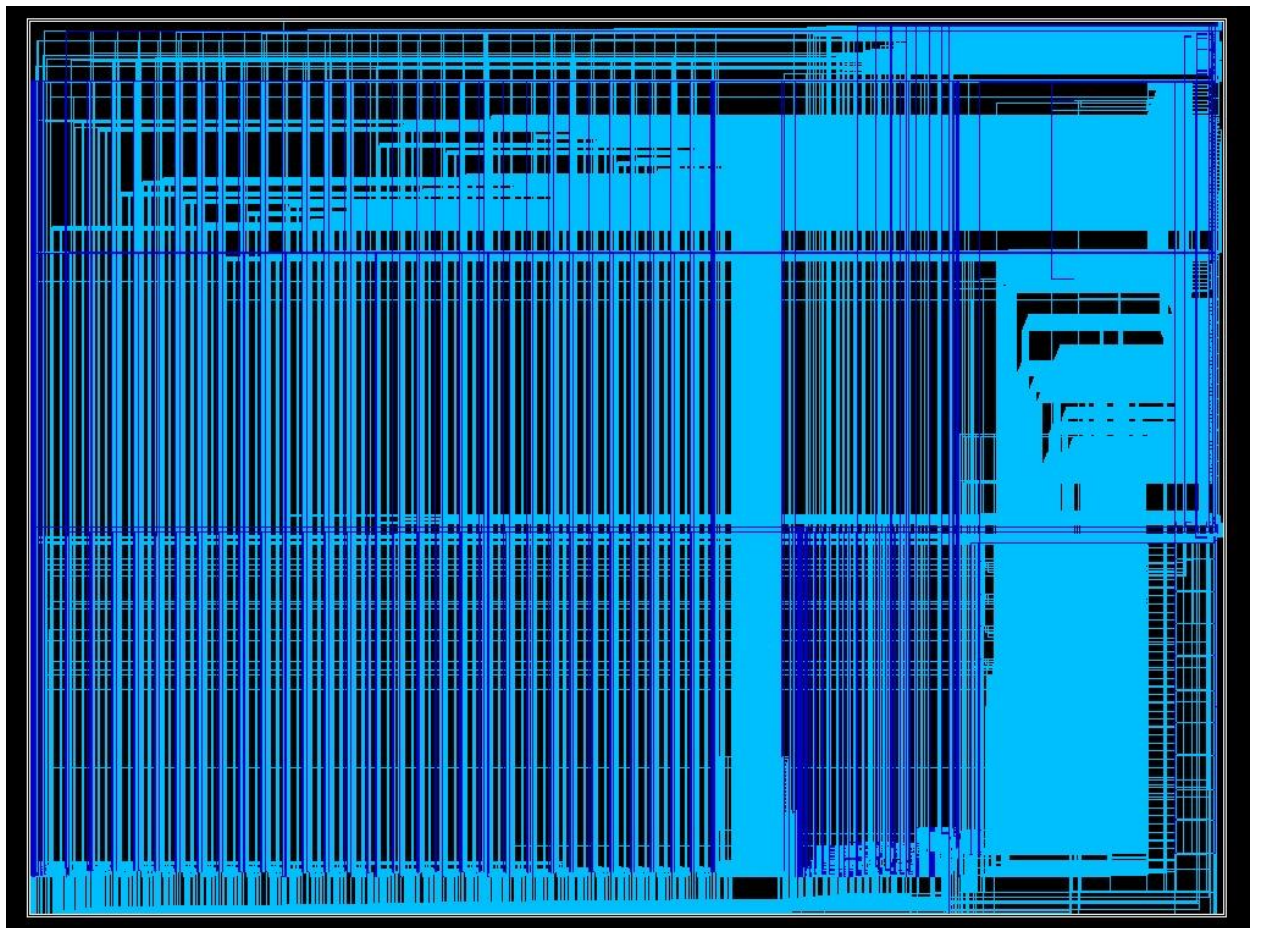
- architecture

- CPU:

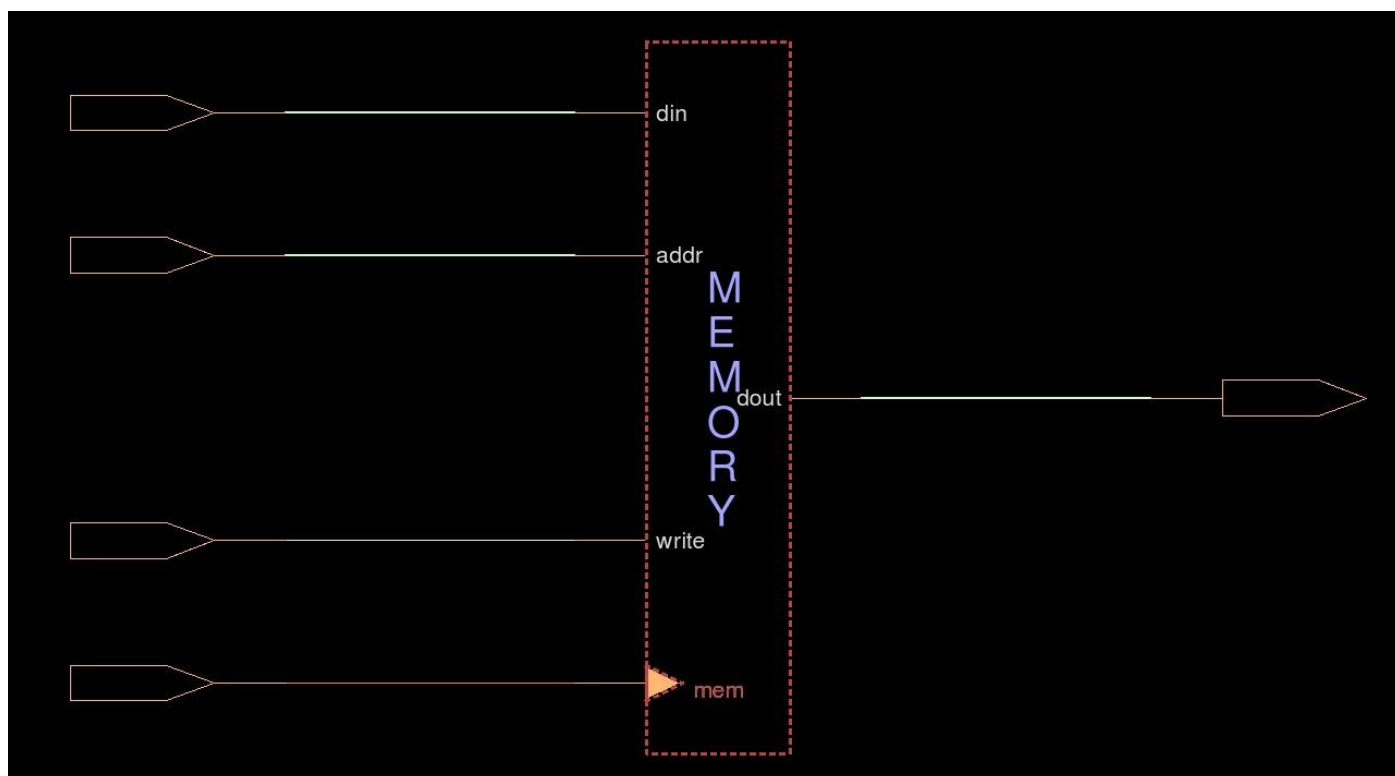
- ◆ Verdi



- ◆ DV



■ MEM:



2. 模擬結果(波型及結果分析)

● The test program and its explanation

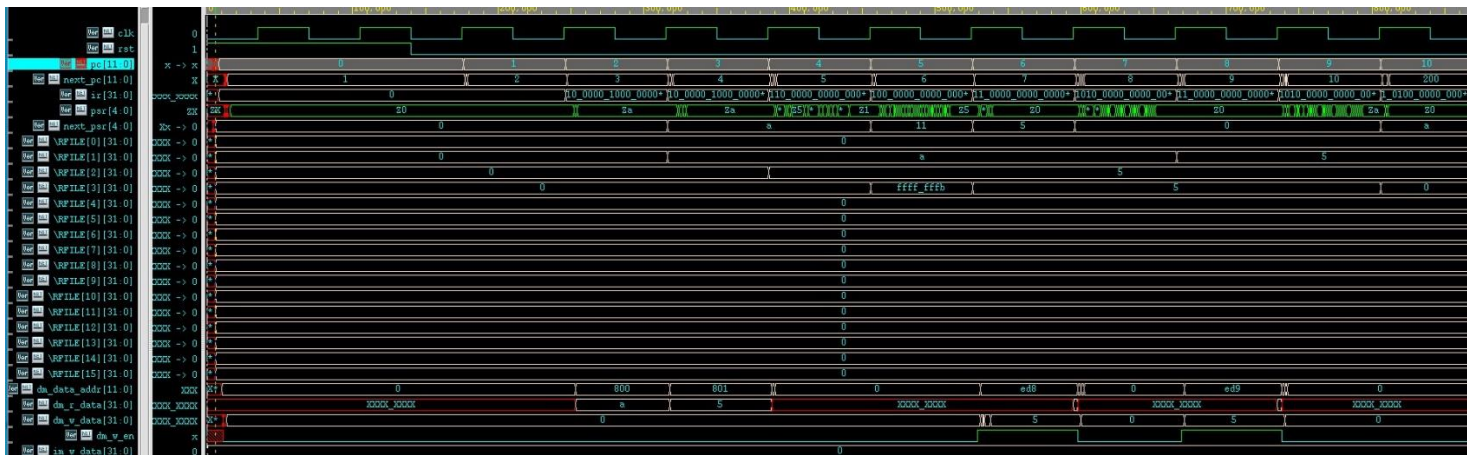
測試程式可分為兩大部分。第一部分是從 DM 讀 input，主要是測 ALU 指令有沒有工作正常；第二部分是直接 load 到 RFILE，是在測我新增加的 JSUB 和 RET 指令，利用遞迴函數來測試有沒有跳到指定的 PC 所對應的 IR。

ALU 測試

1.

插入兩個 NOP 開始加減法和 CMP 測試。最後再利用一次 SUB 來比較有沒有相等，有的話就可以用 BRA 跳到下個測試

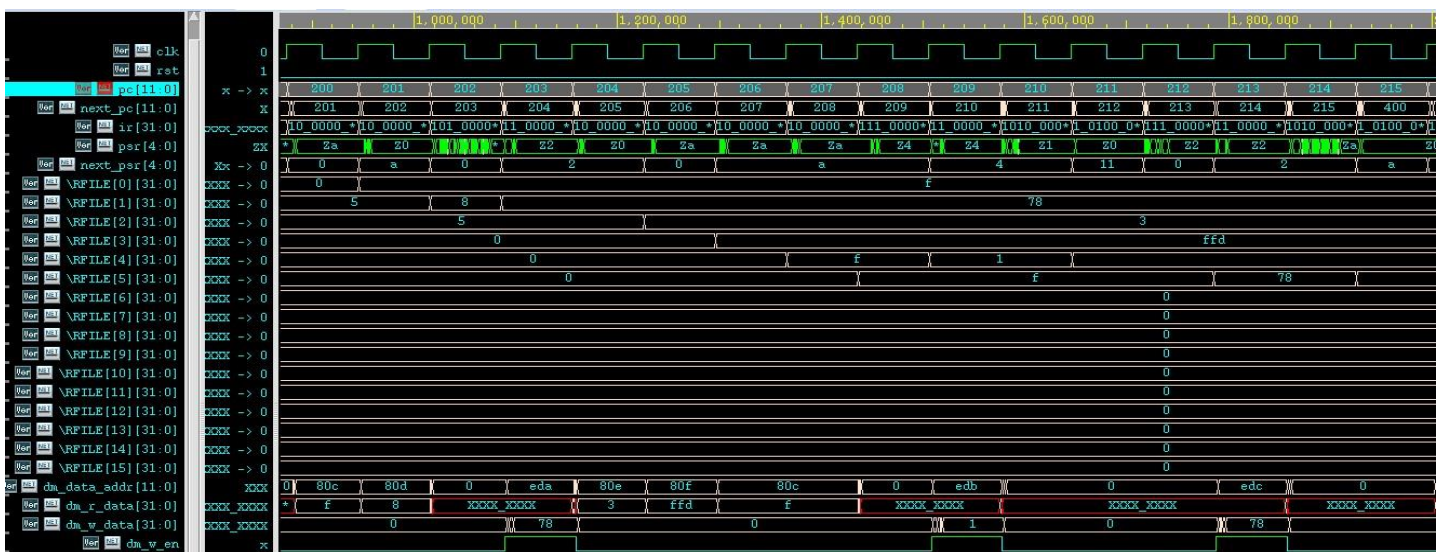
(PC=200)。



▲可看到經過兩個 NOP 閒置，再 LD data 到 reg 裡，就開始算 5 的 2's complement 並與 10 相加存入 reg 裡，接著是 10 與 5 相減並存入 reg 裡，最後做 SUB 並比較，結果為 0 所以就跳到 pc 為 200 的 IR。

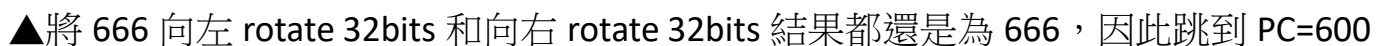
2.

這裡是測試 SHF 的正負數有沒有左移和右移，並與乘法的結果作比較，有相等一樣跳到下一組測試。

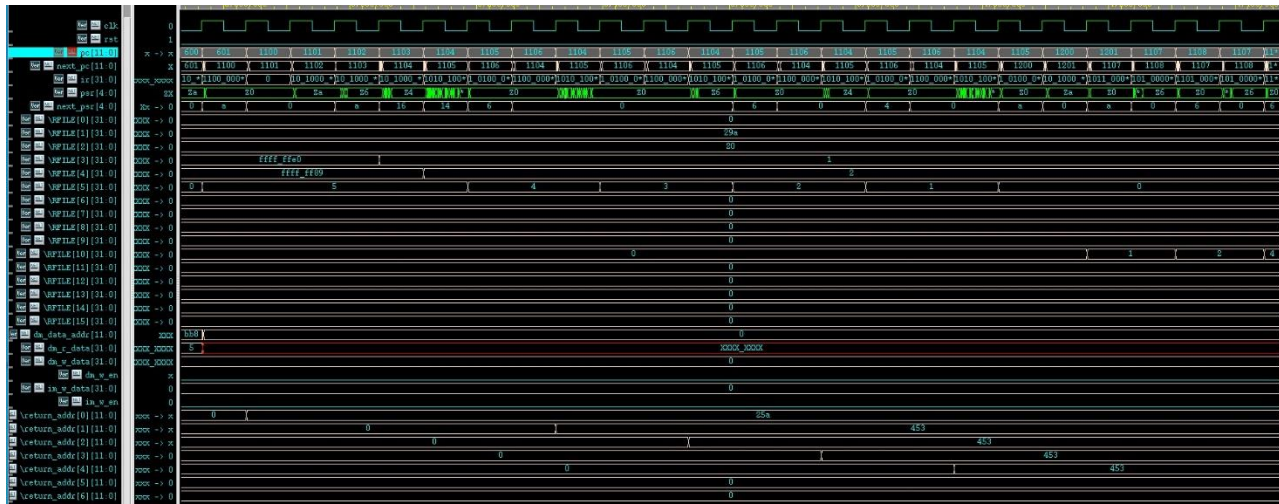


▲可看到我們把 LD 進 reg 的 8 和 15 相乘，並存回 reg 等等做比較。接著 LD +3 和 -3 做>>和<<測試，結果分別為 1 和 120，所以後者會 BRA 至 pc=400 的 IR。

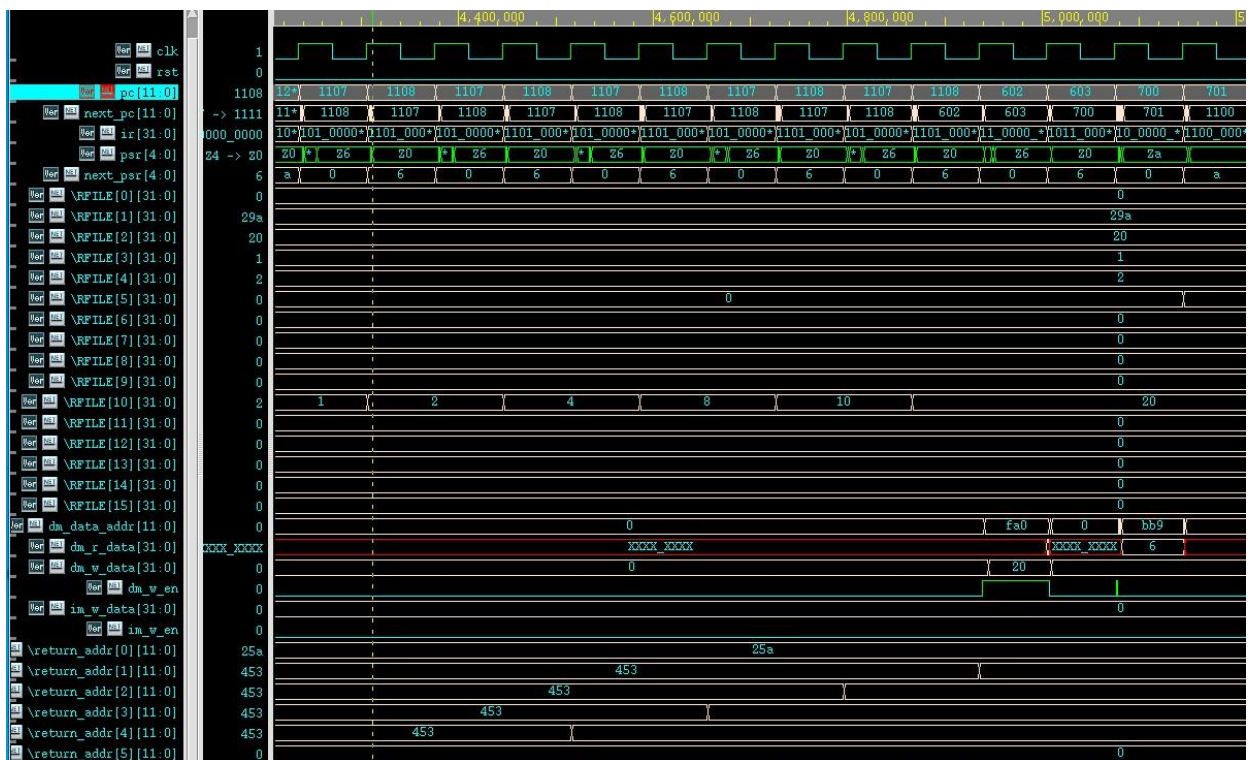
這裡是測試 ROR 的正負數有沒有左移和右移，並與左移 32 位和又移 32 位的結果作比較，有相等一樣跳到下一組測試。



我是利用 `power recursive` 來測的，首先將測試的次方數用 LD 到 `reg 5` 裡，並跳到 `recursive function 1100`。先初始化答案 `reg` 為 0，並將 `decrement` 設為 1 和 `Base` 設為 2，就開始將 `index` 減 1，直到 `index` 為 0 跳到 `return1` 的 label，再跳回到 `recursive function` 做 *2 的動作，根據要 RET 幾次就做幾次 *2 直到跳回到 `call recursive function` 的 `PC+1`。



▲一直 call recursive function，所以會利用 JSUB 跳到指定為至 n 次



▲當 n 被扣到 0 時開始 RET，並開始 *2 n 次



▲最後做完五個 JSB 測資後，在 halt 前利用 STR 將 FFFFFFFF 存入 DM[4095]

- simulation results and explanation

- RTL sim

```
7425 1108
7455 1107
7485 1108
7515 1107
7545 1108
7575 1107
7605 1108
7635 1107
7665 1108
7695 1002
7725 1003

Done

DM[ 3800] = 00000005, pass
DM[ 3801] = 00000005, pass
DM[ 3802] = 00000078, pass
DM[ 3803] = 00000001, pass
DM[ 3804] = 00000078, pass
DM[ 3805] = 0000029a, pass
DM[ 3806] = 0000029a, pass
DM[ 4000] = 00000020, pass
DM[ 4001] = 00000040, pass
DM[ 4002] = 00000080, pass
DM[ 4003] = 00000100, pass
DM[ 4004] = 00000200, pass

*****
**                                     **
**      Congratulations !!           **
**                                     **
**      Simulation PASS!!            **
**                                     **
*****

$finish called from file "testbench.v", line 103.
$finish at simulation time 775500
V C S   S i m u l a t i o n   R e p o r t
Time: 7755000 ps
CPU Time: 0.790 seconds; Data structure size: 0.1Mb
Thu Dec 1 19:02:08 2022
CPU time: .701 seconds to compile + .527 seconds to elab + 10.986 seconds to lin
k + .815 seconds in simulation
[vsd202271@qic-svr RTL]$ verdi
logDir = /home/ncku_class/vsd2022/vsd202271/HW6/HW6/CPU/RTL/verdiLog
```

- Post-syn sim

```
16906 1108
16976 1107
17046 1108
17116 1107
17186 1108
17256 1107
17326 1108
17396 1107
17466 1108
17536 1107
17606 1108
17676 1107
17746 1108
17816 1107
17886 1108
17956 1002
18026 1003

Done

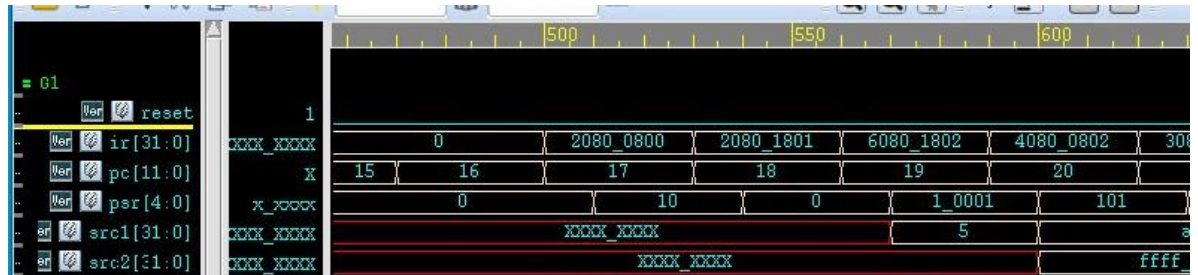
DM[ 3800] = 00000005, pass
DM[ 3801] = 00000005, pass
DM[ 3802] = 00000078, pass
DM[ 3803] = 00000001, pass
DM[ 3804] = 00000078, pass
DM[ 3805] = 0000029a, pass
DM[ 3806] = 0000029a, pass
DM[ 4000] = 00000020, pass
DM[ 4001] = 00000040, pass
DM[ 4002] = 00000080, pass
DM[ 4003] = 00000100, pass
DM[ 4004] = 00000200, pass

*****
**                                     **
**      Congratulations !!           **
**                                     **
**      Simulation PASS!!            **
**                                     **
*****

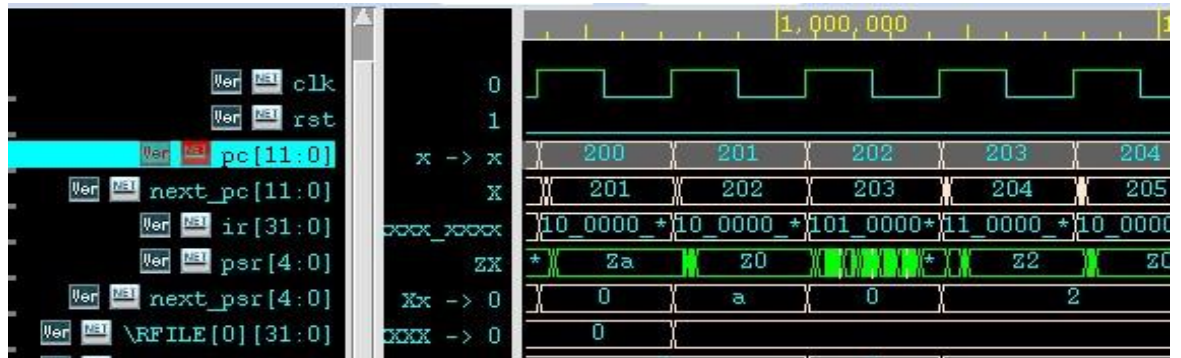
$finish called from file "testbench.v", line 103.
$finish at simulation time 18095000
V C S   S i m u l a t i o n   R e p o r t
Time: 18095000 ps
CPU Time: 1.400 seconds; Data structure size: 3.1Mb
Thu Dec 1 19:32:57 2022
CPU time: 5.281 seconds to compile + 1.881 seconds to elab + 12.407 seconds to link + 1.438 seconds in simulation
[vsd202271@qic-svr postsim]$ +lint=TFIPC-L
+lint=TFIPC-L: Command not found.
[vsd202271@qic-svr postsim]$ nWave &
[5] 33492
[vsd202271@qic-svr postsim]$ logDir = /home/ncku_class/vsd2022/vsd202271/HW6/HW6/CPU/postsim/nWaveLog
```


■ Wave

◆ Rtl



◆ Post_syn



可看到合成後的波型會因為各個 gate 的 delay 而會有暫態反應，像是圖中 psr 因為是在 result 計算完後才能計算，所以會一直受到前面各個值引響 result 值而變更 CC code。

● synthesis results and explanation

■ Timing (slack=32.39 for period=70)

U4758/Y (XOR2X1)	0.36	70.59	r
U4756/Y (XOR2X1)	0.36	70.95	r
U4755/Y (XOR2X1)	0.37	71.32	r
U4926/Y (XOR2X1)	0.34	71.65	f
U4924/Y (AOI22X1)	0.35	72.00	r
U5949/Y (NAND4X1)	0.22	72.22	f
next_psr_reg[2]/D (DFFRHQX1)	0.00	72.22	f
data arrival time		72.22	
clock CLK (rise edge)	105.00	105.00	
clock network delay (ideal)	0.00	105.00	
next_psr_reg[2]/CK (DFFRHQX1)	0.00	105.00	r
library setup time	-0.39	104.61	
data required time		104.61	
data required time		104.61	
data arrival time		-72.22	
slack (MET)		32.39	

***** End Of Report *****

■ Area =226943

```

Report : area
Design : CPU
Version: Q-2019.12
Date   : Thu Dec  1 19:55:46 2022
*****

Library(s) Used:

    slow (File: /home/ncku_class/vsd2022/vsd202200/dv_data/db/slow.db)

Number of ports:          576
Number of nets:          9040
Number of cells:         8276
Number of combinational cells: 6964
Number of sequential cells: 1303
Number of macros/black boxes: 0
Number of buf/inv:       957
Number of references:     49

Combinational area:      135923.356926
Buf/Inv area:            6446.563282
Noncombinational area:   91020.284027
Macro/Black Box area:    0.000000
Net Interconnect area:   1234899.552307

Total cell area:         226943.640953
Total area:              1461843.193260

***** End Of Report *****

```

■ Power =1.363mW

```

Leakage Power Units = 1pw

Cell Internal Power   = 1.1392 mW   (84%)
Net Switching Power  = 216.9850 uW  (16%)

Total Dynamic Power   = 1.3561 mW  (100%)
Cell Leakage Power    = 7.4569 uW

Power Group    Internal Power    Switching Power    Leakage Power    Total Power    ( % )    Attrs
-----
io_pad         0.0000         0.0000         0.0000         0.0000    ( 0.00%)
memory         0.0000         0.0000         0.0000         0.0000    ( 0.00%)
black_box      0.0000         0.0000         0.0000         0.0000    ( 0.00%)
clock_network  0.0000         0.0000         0.0000         0.0000    ( 0.00%)
register       1.0735         3.3312e-03     2.8936e+06     1.0797    ( 79.18%)
sequential     0.0000         0.0000         0.0000         0.0000    ( 0.00%)
combinational  6.5674e-02     0.2137         4.5634e+06     0.2839    ( 20.82%)
-----
Total          1.1392 mW      0.2170 mW      7.4569e+06 pW  1.3636 mW
***** End Of Report *****

```

我的週期之所以需要比別人的大才不會有 time violation 是因為我把原本的 fetch/execute/write 濃縮成一個 cycle，利用 single-cycle CPU 就可解決原先三個 cycle unbalance 會浪費時間等問題，這樣就不會有不 balanced 的問題而讓費時間。優化過後與原先做的需要 fetch/execute/write 相比(period=40 total time

為 31020000)，快了 1.7 倍左右，總模擬時間變為:18095000。

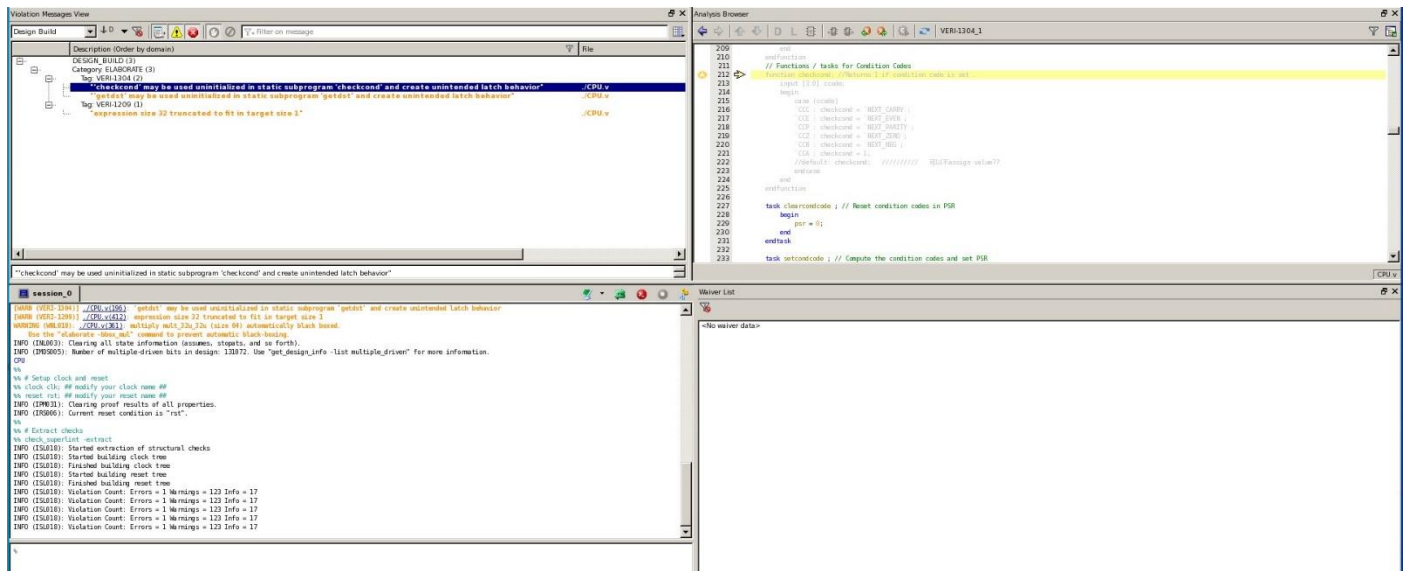
面積的話，因為我有 `jsub` 和 `ret` 指令，為了不怕會有超大遞迴導致出現 error，所以我開了 64 個 12bit 的 register 來存，因此會大了一些。

3. Coding style

原本未修改：因為電路中 `rotate` 指令是用 `while` 來實現，而這個語法是不可合成的，所以會有 error 導致無法分析 coding style 及合成電路。還有很多 `display`, `monitor`, `$stop`, 等指令，我都用 `//synopsys translate_off` 和 `//synopsys translate_on`，把他們註記起來了，也修改了原本會有 `latch` 的所有情形和 `blocking`, `non-blocking` 混用的錯誤。

```
%% # import and elaborate design //
%% analyze -v2k ./instruction_set_model.v; ## modify your file name ##
[WARN (VERI-9025)] ./instruction_set_model.v(8): Empty Port in Module Declaration
%% elaborate -bbox true -top instruction_set_model; ## modify your top module ##
[WARN (VERI-9033)] ./instruction_set_model.v(23): array NEM (size 131072) automatically BLACK-BOXED. Use the "elaborate -bbox_a" command to prevent automatic black-boxing.
[WARN (VERI-9033)] ./instruction_set_model.v(30): array return_addr (size 12288) automatically BLACK-BOXED. Use the "elaborate -bbox_a" command to prevent automatic black-boxing.
[WARN (VERI-1060)] ./instruction_set_model.v(338): 'initial' construct ignored
[WARN (VERI-1060)] ./instruction_set_model.v(345): 'initial' construct ignored
[WARN (VERI-1209)] ./instruction_set_model.v(156): expression size 32 truncated to fit in target size 12
[WARN (VERI-1330)] ./instruction_set_model.v(166): actual bit length 4 differs from formal bit length 5 for port 'ccode'
[WARN (VERI-1304)] ./instruction_set_model.v(120): 'checkcond' may be used uninitialized in static subprogram 'checkcond' and create unintended latch behavior
[WARN (VERI-1209)] ./instruction_set_model.v(180): expression size 33 truncated to fit in target size 32
[WARN (VERI-1142)] ./instruction_set_model.v(115): system task 'display' ignored for synthesis
[WARN (VERI-1304)] ./instruction_set_model.v(106): 'getdst' may be used uninitialized in static subprogram 'getdst' and create unintended latch behavior
[WARN (VERI-1209)] ./instruction_set_model.v(224): expression size 32 truncated to fit in target size 1
[WARN (VERI-1209)] ./instruction_set_model.v(225): expression size 32 truncated to fit in target size 12
[WARN (VERI-1209)] ./instruction_set_model.v(238): expression size 32 truncated to fit in target size 12
[WARN (VERI-1209)] ./instruction_set_model.v(239): expression size 33 truncated to fit in target size 32
[ERROR (VERI-1080)] ./instruction_set_model.v(226): non-constant loop condition not supported for while
ERROR (ENL134): Unable to elaborate design as top module "instruction_set_model" is black boxed.
Summary of errors detected:
[ERROR (VERI-1080)] ./instruction_set_model.v(226): non-constant loop condition not supported for while
ERROR (ENL134): Unable to elaborate design as top module "instruction_set_model" is black boxed.
ERROR at line 17 in file /home/ncku_class/vsd2022/vsd202271/HW5/HW5/CPU_be/superlint.tcl, more info in Tcl-variable errorInfo
ERROR (ENL034): 2 errors detected in the design file(s).
```

修改完後：剩三個不好的 style。



利用兩層 for 將所有 rotate 結果算出來並儲存在大小為 33 的 array 中，就不會 while 中有未知的變數導致電路無法合成。

4. 心得

這次 lab 讓我好好學習到了哪些語法是可以合成的，而哪些語法是不可合成的，因為講義的架構和 code 會有非常多不可合成、blocking, nonblocking 混用，bits 不同 truncate 再一起

討論

1. Vcs 的 timescale 要放在最前面

6、VCS编译注意问题:

使用VCS编译时, 必须先含有`timescale或者宏定义的文档放在前面, 不然会报错误
Error-[ITSFM] Illegal `timescale for module
router_test_top.sv, 7
Module "router_test_top" has `timescale but previous module(s)/package(s) do not.
Please refer LRM 1364-2001 section 19.8.

2. 底下為疑問，待詢問並解決

[illegible]

ELAPSED TIME	AREA	WORST NEG SLACK	TOTAL SETUP COST	DESIGN RULE COST	ENDPOINT	MIN DELAY COST
0:00:06	1071.1	0.00	0.0	0.0		0.00
0:00:06	1071.1	0.00	0.0	0.0		0.00
0:00:06	1071.1	0.00	0.0	0.0		0.00

[illegible]

Optimization Complete

5

[illegible]

```
Updating timing information
Information: Updating design information... (UID-85)
```


[illegible]