

程式碼檔案：pipealu.v

程式碼解說：

(1) Instruction fetch & decode 的部分是直接用 assign 的方法直接所需要的資料，原本是放在第一個 always 內做初始化，不過詢問助教的意見後，移到外面寫比較好。

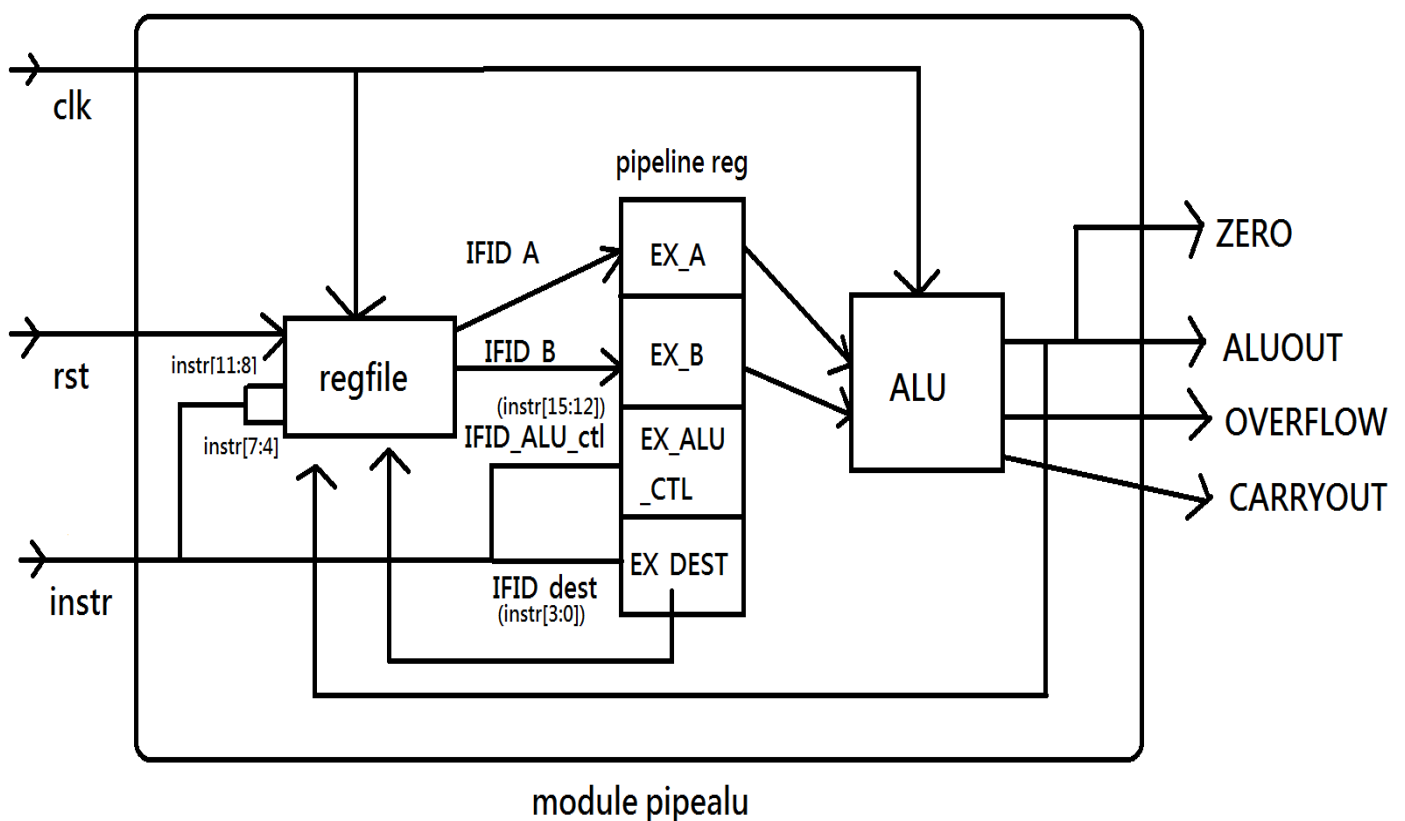
(2) 第一個 always：

1. regfile 的初始化
2. 把 ALUOUT 的結果在算出來之後的下一個 clock 正緣寫回到 regfile

(3) 第二個 always：

1. 將 IF/ID 所獲得的資訊寫到 pipeline 的 register 內部
2. 計算 ALUOUT overflow carry out

結構如下圖：



合成 report 檔案：area.txt    timing.txt

面積：

Number of ports:	53
Number of nets:	1761
Number of cells:	1676
Number of combinational cells:	1125
Number of sequential cells:	550
Number of macros/black boxes:	0
Number of buf/inv:	46
Number of references:	23

Combinational area:	17103.974851
Buf/Inv area:	255.283206
Noncombinational area:	17430.221043
Macro/Black Box area:	0.000000
Net Interconnect area:	3889.269826

Total cell area:	34534.195894
Total area:	38423.465720

時間：

Des/Clust/Port	Wire Load Model	Library
-----		
pipealu	35000	saed90nm_typ

Point	Incr	Path
-----		
AluOut_reg[28]/CLK (DFFX1)	0.00	0.00 r
AluOut_reg[28]/QN (DFFX1)	0.13	0.13 r
U1866/QN (NAND4X0)	0.09	0.21 f
U1865/QN (NOR4X0)	0.12	0.33 r
U1859/Q (AND2X1)	0.10	0.43 r
Zero (out)	0.00	0.43 r
data arrival time		0.43
-----		

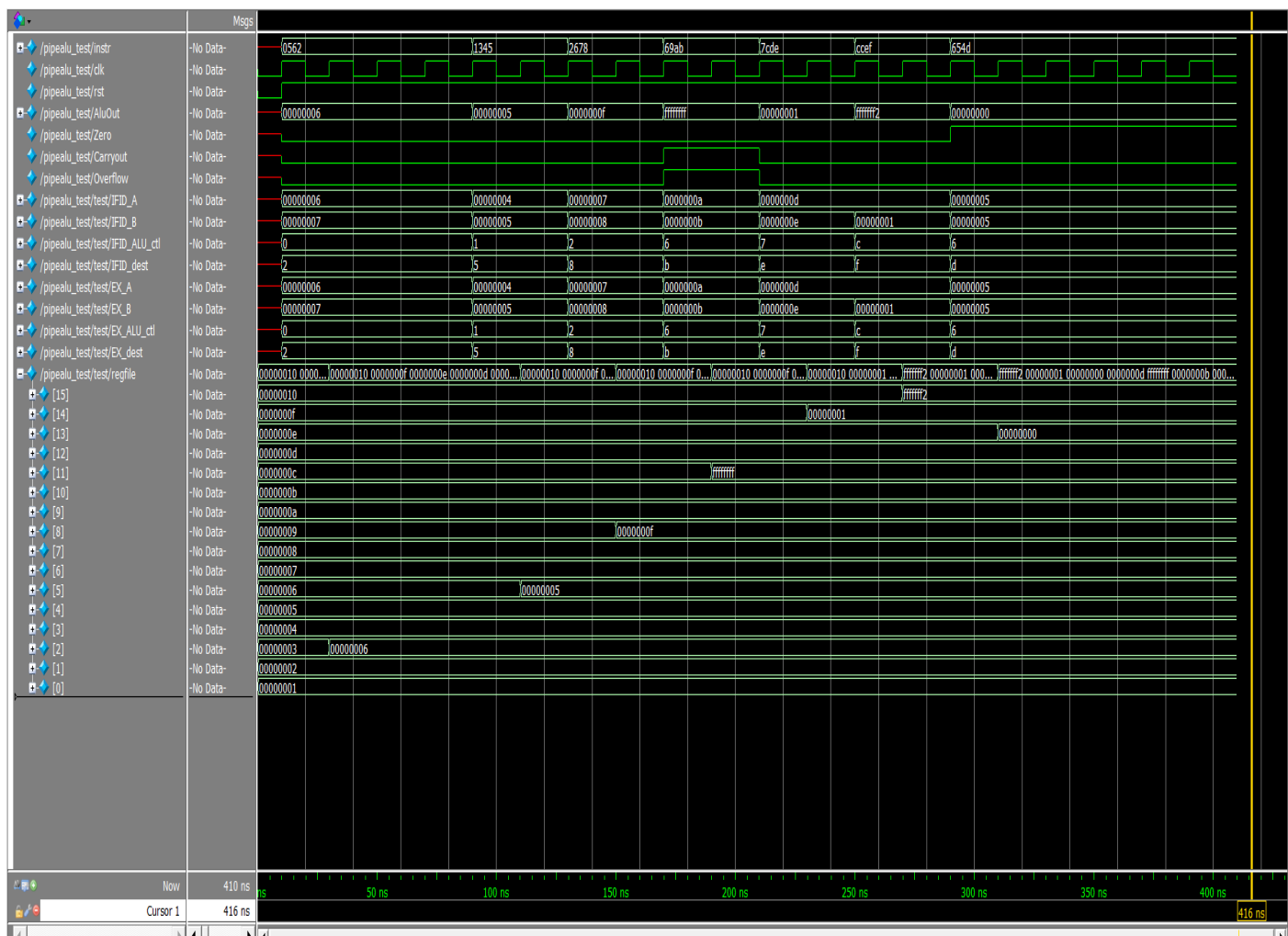
## Testbench 檔案：pipealu\_test.v

我的 testbench 的 regfile 初始化分別為 1~16

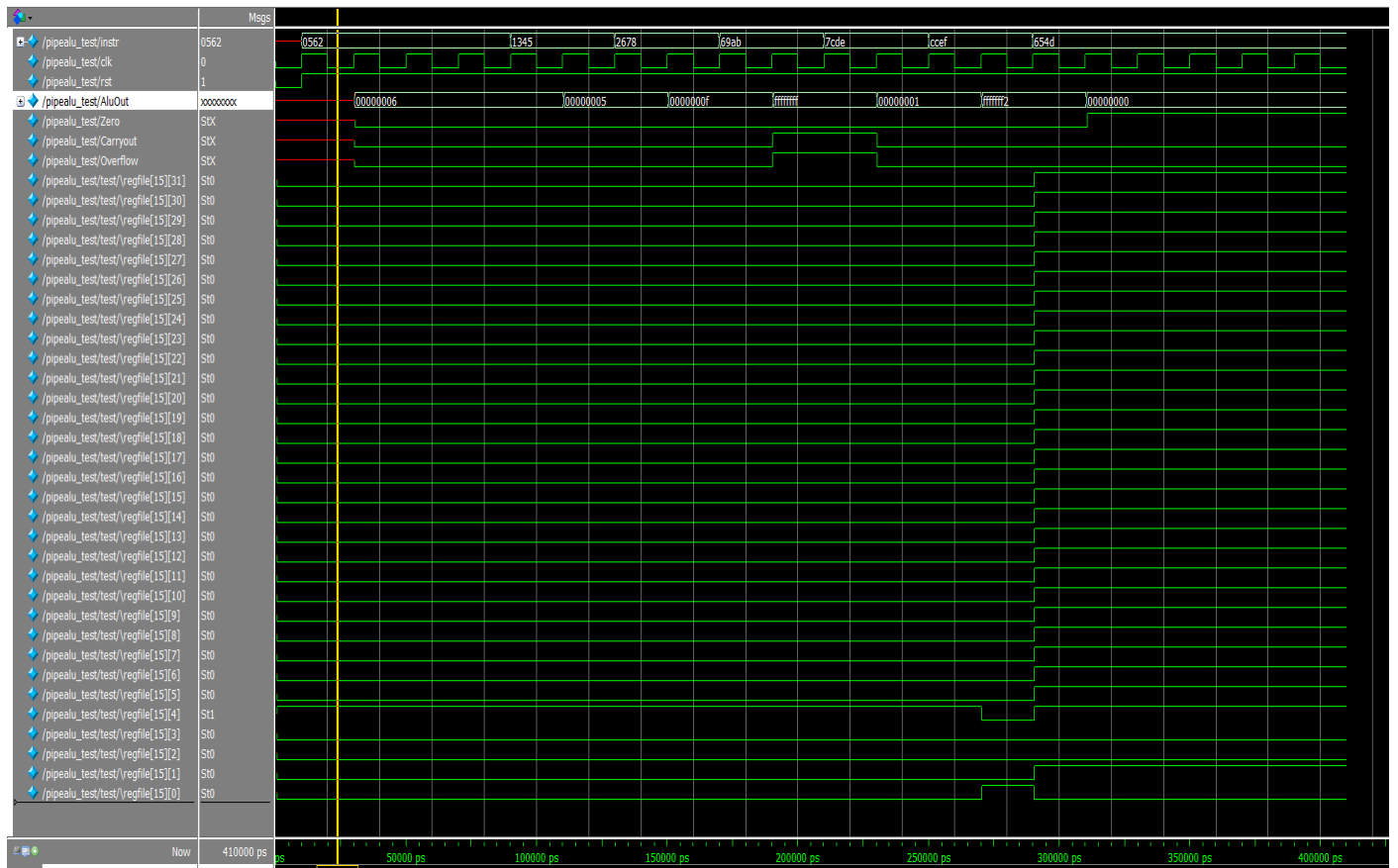
測試的指令有：

```
instr = 16'h0562;//R2 = R5 & R6
instr = 16'h1345;//R5 = R4 | R3
instr = 16'h2678;//R8 = R6 + R7
instr = 16'h69ab;//R11 = R10 -R9
instr = 16'h7cde;//R14 = R12)<R13
instr = 16'hcccf;//R15 = R12 nor R14
instr = 16'h654d;//R13 = R4 - R5 = 0
```

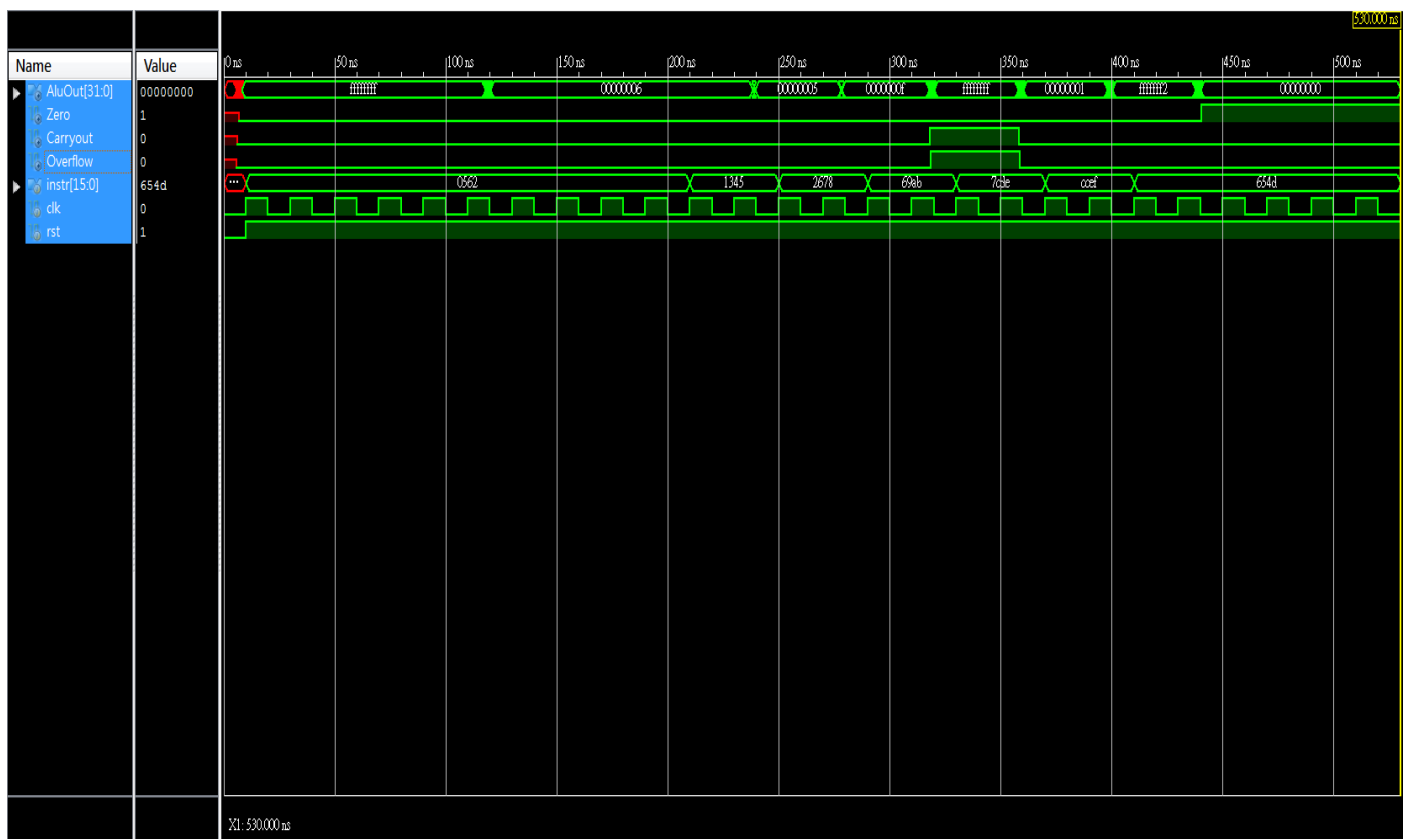
### (1)RTL 驗證



### (2)design compiler 生成的 gatelevel 驗證



### (3)ISE Xilinx FPGA 驗證



我給 FPGA 的 testbench 有差異：我第一個指令給了之後，給了很久的 delay 才給第二個，這樣才能符合延遲，不然後面的指令會沒有讀到。

寫這份作業遇到的困難：

1. 不太會用 `regfile`。
2. 對於 `blocking vs non-blocking` 的理解不夠深。
3. 不太懂 `pipeline` 如何實作。
4. 寫出來的 `code` 不能合成。
5. 合成出來的硬體跑起來不對。
6. 寫的硬體是對的，結果卻因 `testbench` 寫錯，跑出不正確的結果。
7. 不知道 `FPGA` 模擬的 `delay` 要給很大，一開始都跑不出來，試很久才發現。
8. `FPGA` 跟 `design complier` 合成優化的機制不太相同，常常出現不同的結果。

解決方法：

1. 自己慢慢試
2. 問助教