VLSI System Design (Graduate Level)

Fall 2021

HOMEWORK I

REPORT

Must do self-checking before submission:

Compress all files described in the problem into one tar

All SystemVerilog files can be compiled under SoC Lab environment

All port declarations comply with I/O port specifications

Organize files according to File Hierarchy Requirement

No any waveform files in deliverables

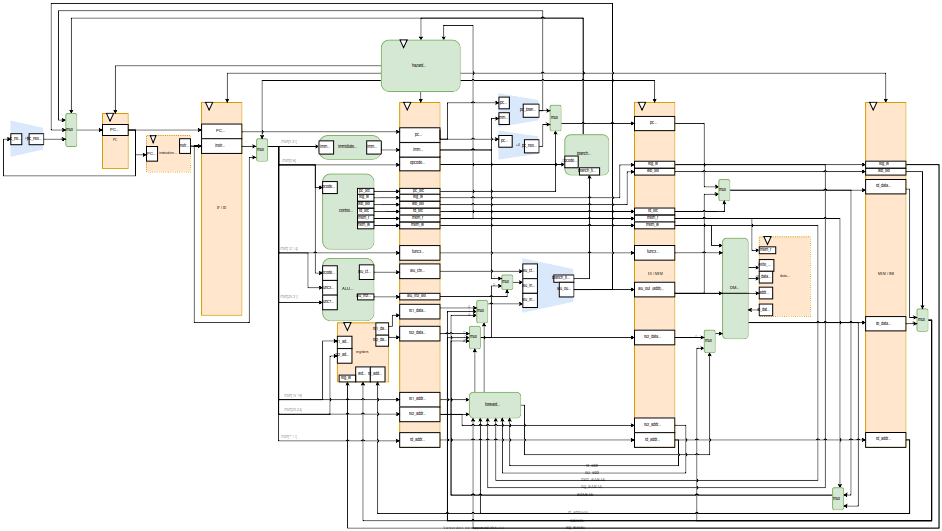
Student name: \_伍志忠\_

Student ID: \_P76104833\_

Summary

完成所有的指定的指令並且合成，合成的cycle設定成11

a , b. Proper explanation of your design,Block diagrams

[](file:///home/cicero/code/VLSI_system_design_2021/hw1/cpu.drawio)如圖(draw.io)，藍色主要是加法器或是ALU，綠色是控制單元或是mux，橘色是register或是memory，有加三角形的表示有接clock。

IF階段會看下個指令的PC是多少以及抓指令。  
ID階段會產生immidiate以及各種控制訊號線

EX階段負責算，如果要跳(branch or jump)的話也會在這階段執行

MEM階段負責把值整理後存進記憶體  
WB階段把值寫回去register

Forward unit如果遇到

1.mem或是wb階段有ex階段需要的值就forward過去

2.wb階段有mem階段需要的值就forward過去

Hazard control unit如果看到memory read會stall，如果遇到要跳指令的狀況就把if\_id和id\_ex的register裡的值(前一個指令)清空。

c. Show your snapshots of the waveforms and the simulation results on the terminal for the different test cases

d.

1.Report the number of lines of your RTL code:

`$wc -l src/\*.sv`

  11 Adder.sv   
  68 ALU\_control.sv   
  24 ALU\_signal\_def.sv   
  57 ALU.sv   
  27 Branch\_control.sv   
  92 Control\_unit.sv   
 285 CPU.sv   
  50 DM\_control.sv   
  47 EX\_MEM\_reg.sv   
  37 Forward\_control.sv   
  50 Hazard\_control.sv   
  71 ID\_EX\_reg.sv   
  55 IF\_ID\_reg.sv   
  21 Immidiate\_generator.sv   
  31 MEM\_WB\_reg.sv   
  13 Mux\_2in.sv   
  20 Mux\_3in.sv   
  20 PC\_reg.sv   
  20 Register.sv   
  99 SRAM\_wrapper.sv   
  56 top.sv   
1154 total

2.the final results of running Superlint

==============================================================

SUMMARY

==============================================================

Properties Considered : 18

assertions : 18

- proven : 2 (11.1111%)

- bounded\_proven (user) : 0 (0%)

- bounded\_proven (auto) : 0 (0%)

- marked\_proven : 0 (0%)

- cex : 16 (88.8889%)

- ar\_cex : 0 (0%)

- undetermined : 0 (0%)

- unknown : 0 (0%)

- error : 0 (0%)

covers : 0

- unreachable : 0

- bounded\_unreachable (user): 0

- covered : 0

- ar\_covered : 0

- undetermined : 0

- unknown : 0

- error : 0

3.3~5 most frequent warning/errors in your code

1.**"Constant '{imm\_,1'b0}' will be left-padded by 4 '0' bits"**

我在ALU\_signal\_def有定義了一個packed struct，是4bits的enum加上1bit的is\_signed，那個constent應該也是4bits enum加1bit的is\_signed，我不懂這錯誤出現的原因，因此沒改。

2.**"Synthesizing 'x'/'z' values in module 'Control\_unit'"**

我確定那些值在那些階段完全不會用到，因此我希望能在合成時讓他自己決定並優化。

3.**"Unequal length operand in bit/arithmetic operator addition in module/design-unit CPU1.dm\_control. LHS operand '{mem\_addr[1:0],3'd0}' is 5 bits, RHS operand '8' is 32 bits"**

我定義了

`define RANGED\_DATA(source,scale,shift) \

source[{shift,3'd0}+scale-1 -:scale]

那個值(8或16)只是表示scale，它最後應該會只是邏輯的一部分而不是線，因此我不想加上bit數。

4.**\*E, "A reachable x-assignment was found"**

同(2)，不管合成出來的值是多少都不會影響結果。

Encountered problem

在寫的過程中最煩的就是看波形debug，看到線接錯、邏輯寫錯、文件沒看仔細等等，發現data memory慢一個clock出來還改了一下架構。幸好main.S有一些註解有標注它應該出來的值是多少，省下了許多自己推演的時間。

這次作業直到合成出來，看到超出預設的clock並明白critical path後跑回去看計機組織他的forward怎設計的，才發現一般的forward沒有從mem讀出來forward。因此我的critical path是  
[mem control]->[forward mux]->[alu]->[branch control]->[hazard control]。我以為mem read的 stall會讓它多一個clock能來得及回去，但是實際上並沒有，幾乎是在一個clock做了2個cycle該做的事，因此預設跑出來的report timing slack=-0.2。這可以藉由修改設計來解決，但是在繳交時間壓力下我決定增加clock的值成11。

learned

複習了verilog的相關知識以及計算機組織，並學習了system verilog的一些寫法、記憶體的delay、相關工具(nWave,superlint,synthesize等等)的使用方法。