Computer Architecture final project(更新版)

組員：R10945061 林宇恆 B09701248 曾彥禎

1. Verilog 設計理念：我們今天解決了昨天上沒辦法讀取到 SP 的問題，結果是必須將所有的值在 rst_n 前先設定好，避免想讀取到沒辦法讀取的部分，最後在測試第二個 pattern 時也順利跑過了我們主要是按照上課所教的 IF ID EX ME WB 來設計我們的 FSM，而我們把 multicycle 的 mul 當成一般 ALU 單元的 sudmodule，取得 mode 的方式先抓取十位元的 instruction(fun3+opcode)，接著再從 which_mode 這個 module 來轉換成所有指令的 mode，但在實作第二個 pattern 我們目前遇到問題是沒辦法從 sp 這個 reg 讀取值甚至做運算，若之後得到解答會補繳這份投影片，我們也有實作組合語言的部分，無奈時間上安排太緊湊沒有實際去跑過 bonus 的部分。以下圖為我們的實作結果以及遇到的問題。
2. Data path:如同上面所提到，FSM 的部分是從 IDLE IF ID ME WB 結束後又會回到 IDLE 進行下一個 cycle，首先我們先從 mem_addr_I 中獲取指令以及立即數，確認好指令後我從 which_mode 這個 submodule 中將指令轉換為

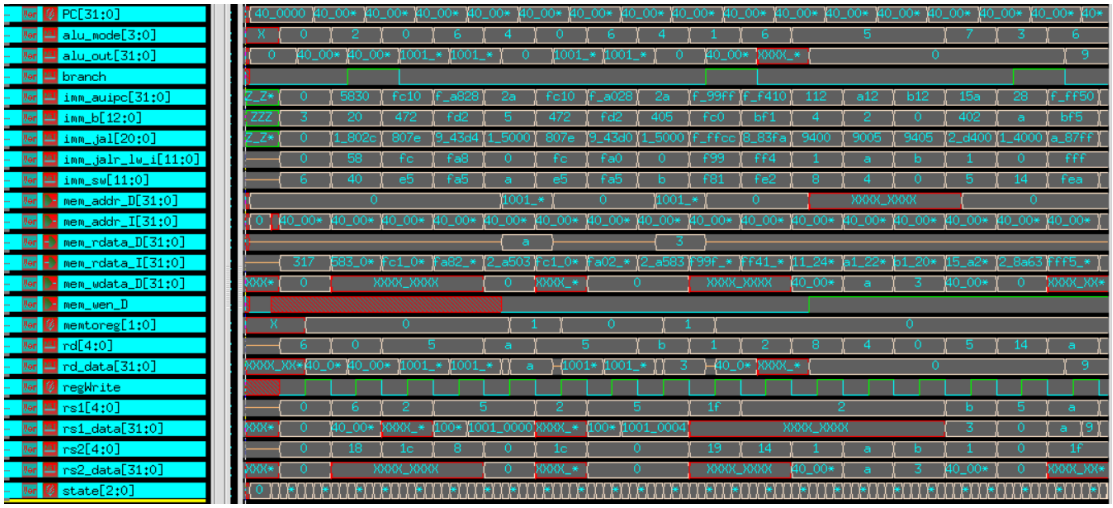
4bit 的指令（較簡潔），而後進入 alu 中進行運算，比較特別要注意的是 jal 跟條件判斷式，這邊針對他們的立即數 0bit 的部分都是 0 所以等同於是直接乘二，這點要注意這邊的立即數為有號數，進行完運算後回到原來的 CHIP.v 中將資料傳回 memory 以及寫進 reg_file，這大概是我實作上的一個 data path 的流程。

3. Multi-cycle:這個部分我是按照作業二的方式來做，我把 multi-cycle 的 module 放在 alu 的 module 裡面，串接起來，首先值會先由 alu 喂進 multi-cycle 的 module 中，隨之隨之給予 valid 信號便開始做 31 個 cycle 的部分，之後會直接連接到 alu 的 output 進行輸出。
4. 此外，我們將指令的使用以最大程度貼合 leaf、perm 所用到的指令，最終於實作中新增的僅有 sub 的指令。隨之給予 valid 信號便開始做 31 個 cycle 的部分，
5. Bonus 組語設計理念：主要組合語言遞迴架構是以 perm 去改寫，並且新增了條件判斷的 testpart 判斷目前條件 n>10 or n<10 or n=0 並跳至對應的運算式中，並且與 perm 一樣持續將 sp 往前推並存入對應的數字，並且於遞迴結束後以 x10=7 開始運算，讓數字以遞迴式加總回

去，最後完成運算將最終的值存回 a1 並結束。

6. 此外，我們將指令的使用以最大程度貼合 leaf、perm 所用到的指令，最終於實作中新增的僅有 sub 的指令。

7. 在 rs1 等於 2 時沒辦法做運算導致最後沒有辦法實作成功，但指令順序都是正確的(此項已解決)。



8. Flipflop:

```
roject/Verilog/CHIP.v'.
============================================================================================
|    Register Name    |    Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
============================================================================================
|      out_r_reg      | Flip-flop |   64   |  Y  | N  | Y  | N  | N  | N  | N  |
|     alu_in_reg      | Flip-flop |   32   |  Y  | N  | N  | N  | N  | N  | N  |
|      state_reg      | Flip-flop |    3   |  Y  | N  | Y  | N  | N  | N  | N  |
|      shreg_reg      | Flip-flop |   64   |  Y  | N  | Y  | N  | N  | N  | N  |
|     counter_reg     | Flip-flop |    5   |  Y  | N  | Y  | N  | N  | N  | N  |
============================================================================================
Presto compilation completed successfully.
Current design is now '/home/raid7_2/user10/r0945061/final_project/2022fall_CA_F
inal_Project/Verilog/CHIP.db:CHIP'
Loaded 5 designs.
Current design is 'CHIP'.
CHIP which_mode alu reg_file mulDiv
design_vision>
```

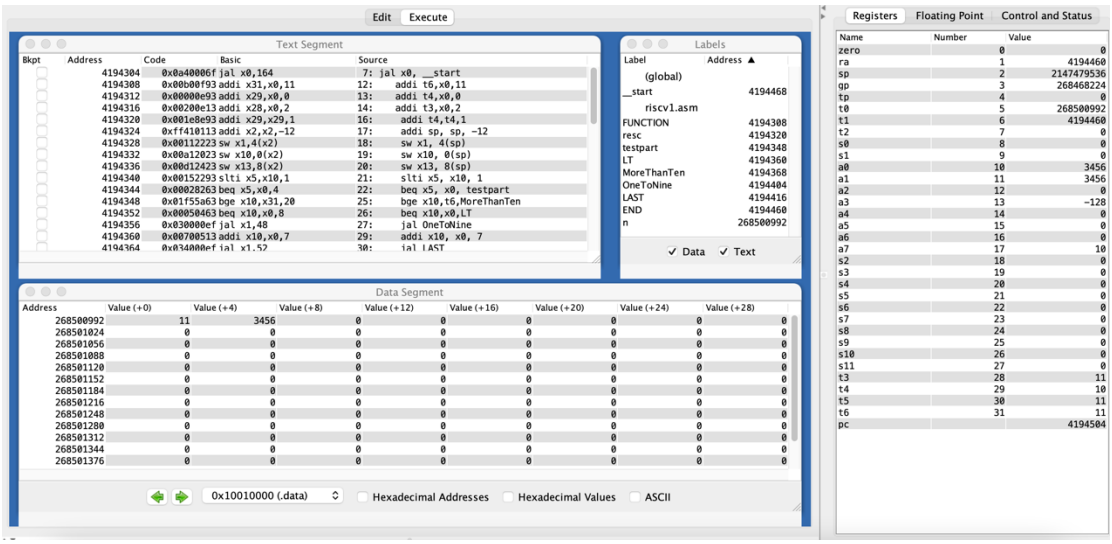
9. Pattern1 結果圖：

```
Loading snapshot worklib.llDiv:v ..................... Done
*Verdi* Loading libsscore_ius152.so
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi_P-2019.06, Linux, 05/26/2019
(C) 1996 - 2019 by Synopsys, Inc.
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the progra
ms that are using this file.
*Verdi* : Create FSDB file 'Final.fsdb'
*Verdi* : Begin traversing the scope (Final_tb), layer (0).
*Verdi* : Enable +mda dumping.
*Verdi* : End of traversing.
-------------------------------------------------------------

START!!! Simulation Start .....

-------------------------------------------------------------


*****************************
*                           *          /|____|\
*    Congratulations !!     *         ((´-___-`))
*                           *        ///       \\\
*    You pass this test!!   *       /||         ||\
*                           *       w|\ m      m /|w
*****************************         \(o)____(o)/

Simulation complete via $finish(1) at time 1545 NS + 0
./Final_tb.v:182            $finish;
ncsim> exit
```

10. Pattern2 結果圖：

```
               Simulation timescale:  10ps
       Writing initial simulation snapshot: worklib.llDiv:v
Loading snapshot worklib.llDiv:v ..................... Done
*Verdi* Loading libsscore_ius152.so
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi_P-2019.06, Linux, 05/26/2019
(C) 1996 - 2019 by Synopsys, Inc.
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file ma
y crash the programs that are using this file.
*Verdi* : Create FSDB file 'Final.fsdb'
*Verdi* : Begin traversing the scope (Final_tb), layer (0).
*Verdi* : Enable +mda dumping.
*Verdi* : End of traversing.
-------------------------------------------------------------

START!!! Simulation Start .....

-------------------------------------------------------------


*****************************
*                           *          /|____|\
*    Congratulations !!     *         ((´-___-`))
*                           *        ///       \\\
*    You pass this test!!   *       /||         ||\
*                           *       w|\ m      m /|w
*****************************         \(o)____(o)/

Simulation complete via $finish(1) at time 5145 NS + 0
./Final_tb.v:182            $finish;
ncsim> exit
[r0945061@cad27 Verilog]$
```

11.組語結果展示：



12.工作分配：

林宇恆撰寫 verilog 架構 討論 pattern 以及 debug

曾彥楨 :撰寫組合語言以及 verilog multicycle 的部分以及討論協助 verilog debug.

13.心得：

非常感謝教授以及助教們學期的付出，起初完全沒有背景知識的情況下來修了這門課，到現在對硬體架構有了嗷深入的了解，都多虧了教授以及助教的回答與講解，雖然到頭來成績不是很理想，Verilog 也是寫得一塌糊塗，但很感謝這堂課讓我學到非常多東西，變得更有學習的熱忱了，希望未來還有機會能夠修到這樣有內容的課程。最後祝教授與助教們新年快樂萬事如意！