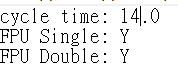
**Computer Architecture HW4 report**

電機四 B05901011 許秉倫

電機四 B05901082 楊晟甫

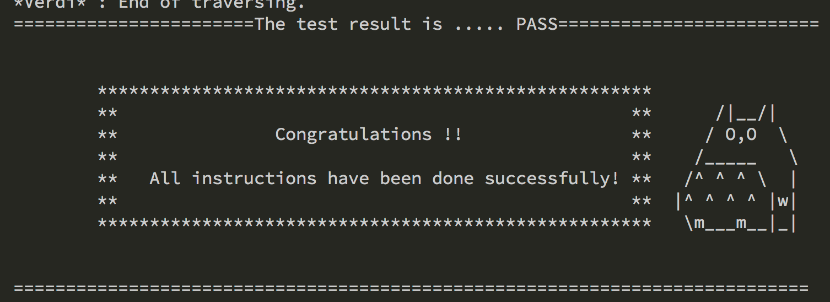
1. Snapshot
   1. Readme.txt content (see sec. 8)



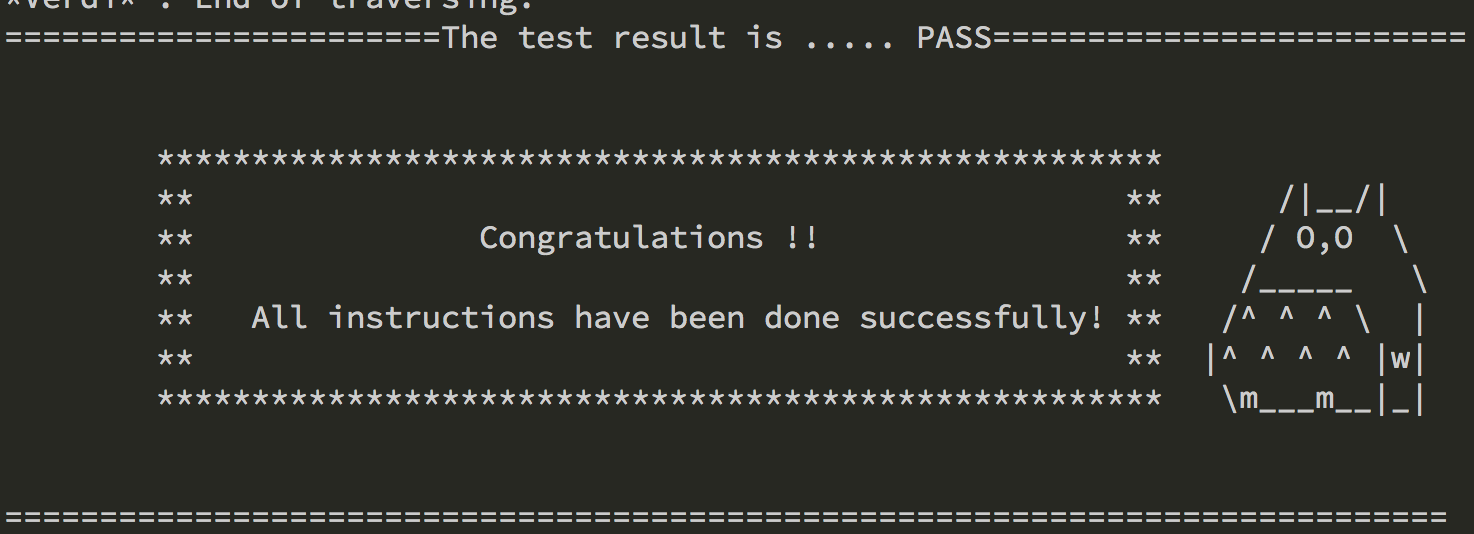
Note: For FPU instructions, minimum cycle time is 100.0

* 1. All RTL Simulation you pass (see Appendix II for example)

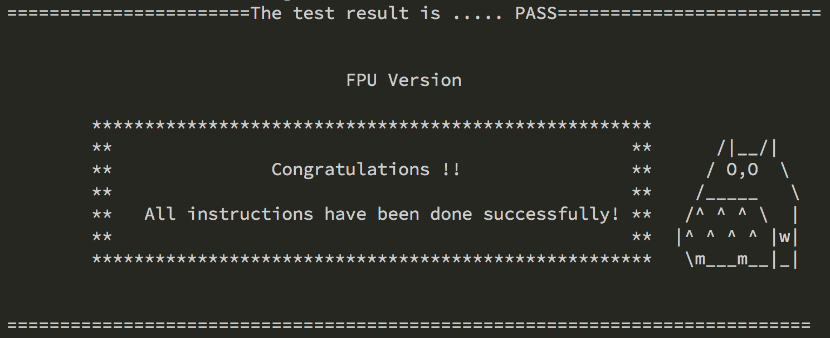
Baseline



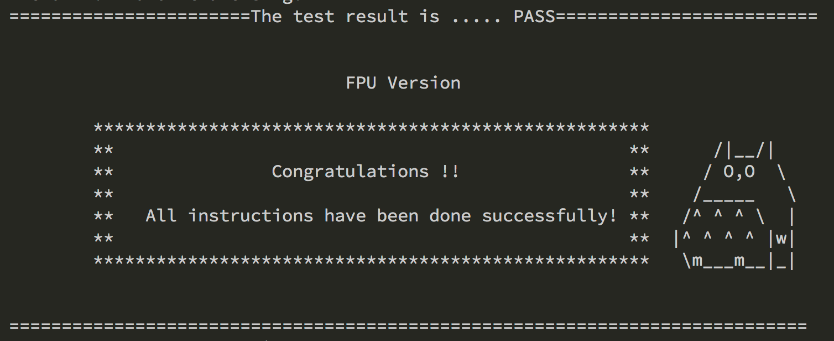
FPU-Baseline



FPU-Single

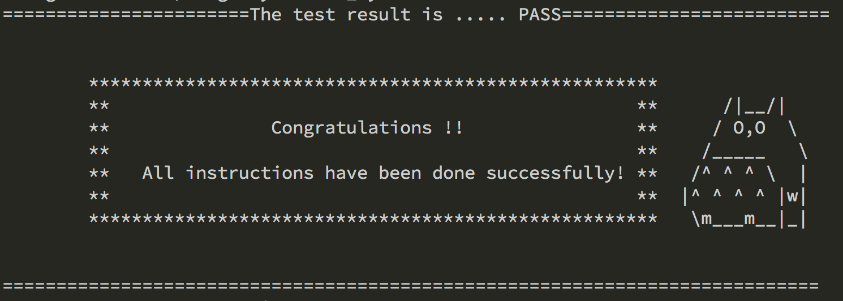


FPU-Double

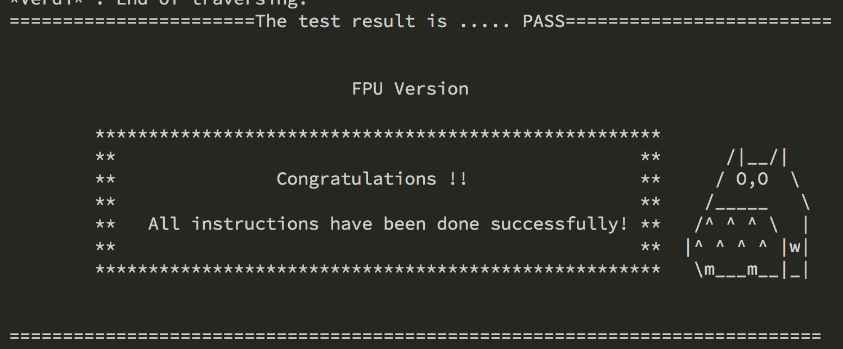


* 1. All Gate-level Simulation you pass

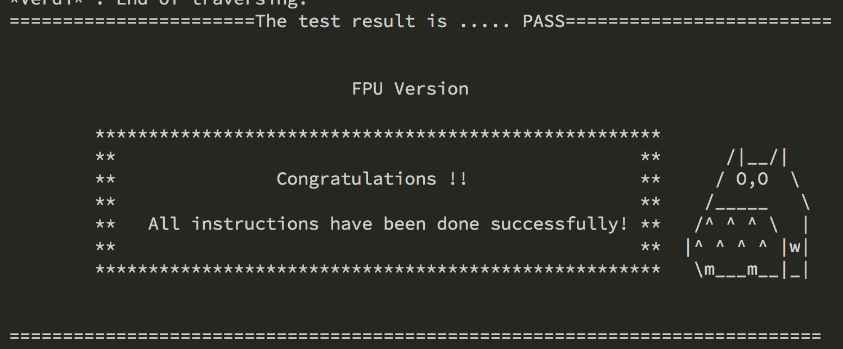
Baseline



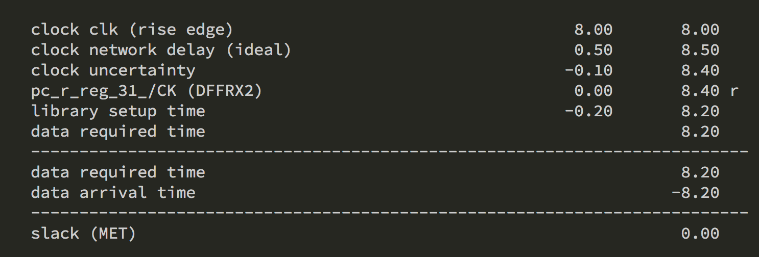
FPU-Single



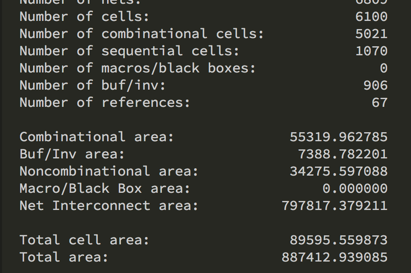
FPU-Double



* 1. Timing report



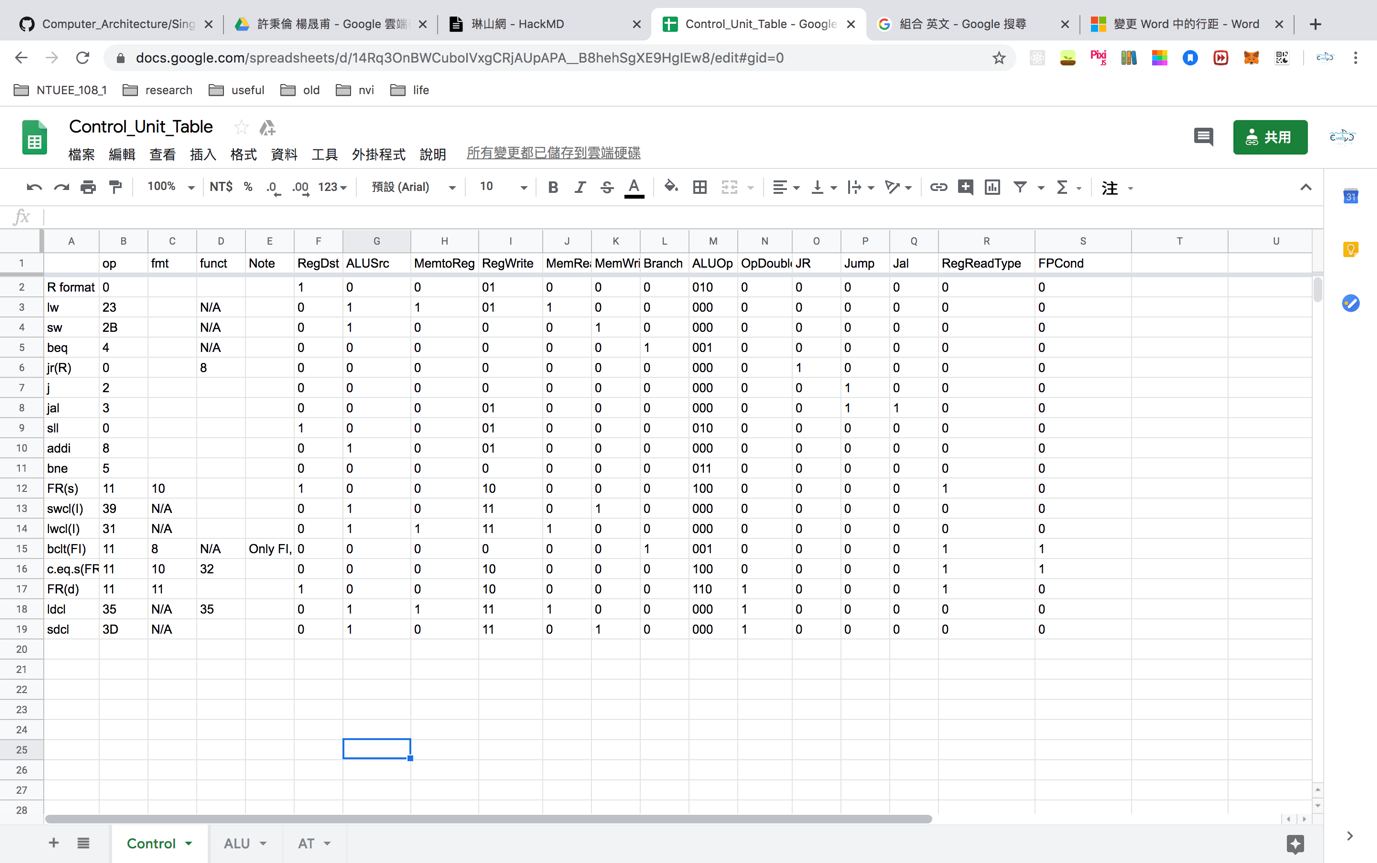
* 1. Area report



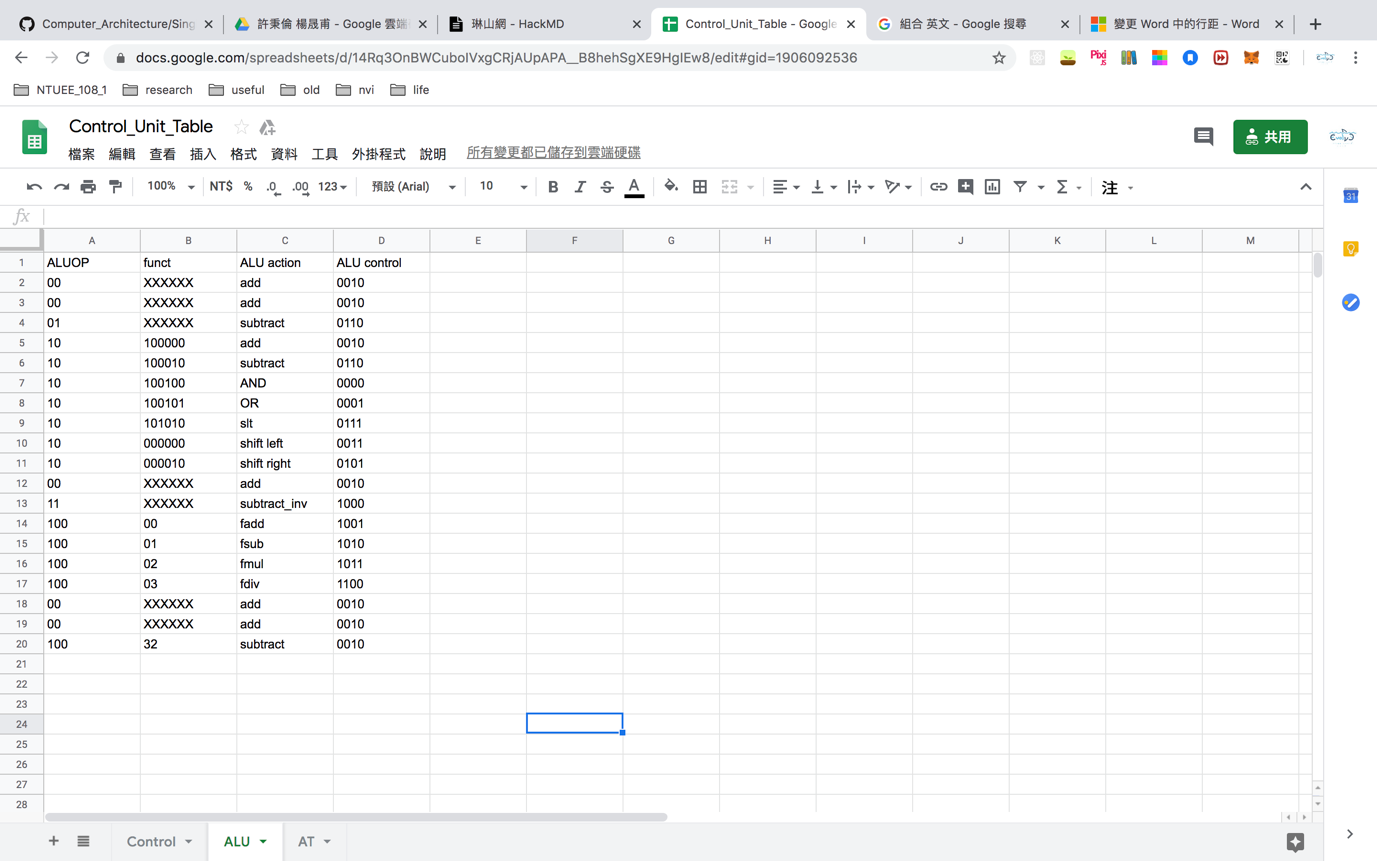
1. Your baseline A\*T value calculated from numbers in the snapshot
2. Please describe how you design this circuit and what difficulties you encountered when working on this exercise.

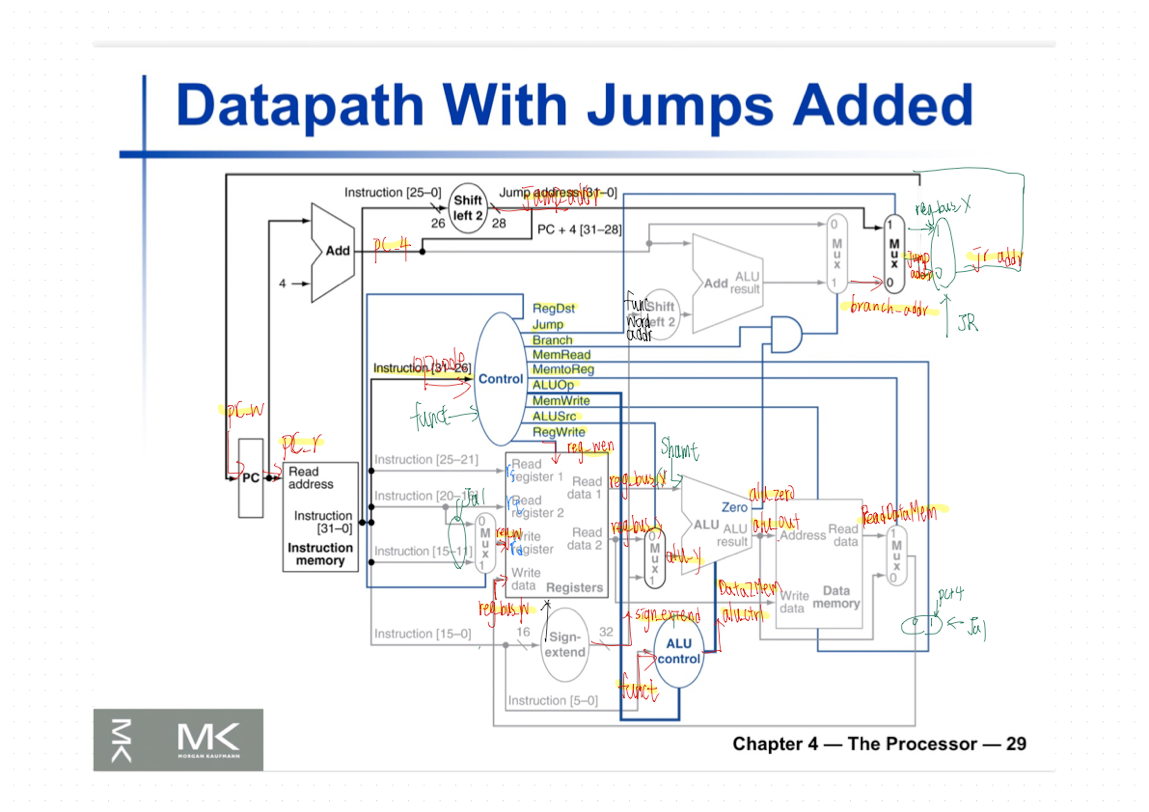
**Design: 我們是照著課本給的電路，把control及alu table擴充，來達成baseline及FPU的功能。**

Control Table:



ALU Control Table:



Baseline: 紅色的字為wrie，綠色的部分是比課本多加的東西  


FPU: FPU的工作主要可以分成以下幾個步驟:

1. 擴充原本的Ctrl Unit, 新增了OpDouble來判斷是否為double instruction，RegWrite判斷是何種寫入模式，(R/I/FR/I(FP))，RegReadType判斷要讀fs/ft/fd或是rs/rt/rd，以及FPCond判斷是否需要access FP Cond
2. 利用原本4bit的 alu control剩餘幾個沒用到的值，分別assign給Single的加減乘除、double的加減以及
3. 將reg的output以及ALU的input、output擴充為64bit，並對沒有用到的32bit補0
4. 在Double的lw sw時，因為有64bit要讀，需分為兩次cycle讀入，所以需要讓PC停住一次
5. 在alu unit裡面，需要調用工作站裡的檔案，這時候需要先instantiation，並依照single double的不同而給不同的parameter

**Difficulties:** 由於我們兩個都對verilog沒有經驗，所以前期花了很多時間上網觀摩別人的code，  
但當大架構出來了之後，就變得比較得心應手。其中最大的困難可以分為兩個:

1. Design: 一開始對processor的運作只停留在instruction怎麼走、什麼時候要write memory, reg，但後來試著把要不同指令要做的事情寫成table之後，整個邏輯就清晰許多，要implement 以及debug都相對方便。
2. Debug: 由於上次作業只有大致了解nWave怎麼debug，這次debug起來有點無從入手，最後發現看instruction的address可以有效發現bug出在哪，怎麼會run time error，也更熟悉nWave的操作介面。
3. Work distribution

許秉倫：Baseline  
楊晟甫：FPU

1. (optional) how you improve your A\*T value

* 將 “control\_table”, “alu\_table”化簡，減少bit用數