**Final Project**

**112學年度第二學期**

**老師：朱守禮　老師**

**學生：**

**資訊二乙11027204鄭宇傑**

**資訊二乙11027205蔡宗勳**

**資訊二乙11027207彭靖傑**

**資訊二乙11027209巫年巨**

**一 設計重點說明 :**

IF / ID :

透過 ec\_reg決定是否要將PC地址的output寫入，

以及在此抓取指令、解碼指令，並輸出register file 的RD1, RD2。

並在此將instruction memory傳入的OP code 輸出給control unit

離開 IF / ID 後，在第二個clk正緣期間，

擴充immediate value，與設定control unit、register file的RN1、RN2

RD1、RD2。

ID / EX :

Fetch control unit傳入的 RegWrite、MemToReg、MemRead、

MemWrite、ALUO、ALUSrc、RegDst並輸出。

並接收擴充後的immediate value 與 RD1、RD2 Address，

與RT、RD ( 之後用於決定WN該為哪個register )。

離開 ID / EX 後，在第三個clk正緣期間，

依照ALU OP code 讓ALU執行運算，並且將PC位置左移2。

EX / MEM :

Fetch MemRead、MemWrite、RegWrite、MemToReg，

與 PC位置左移2後的PC位置( 用於beq )，

以及ALU經過execute後的結果，與WN的register。

離開 EX / MEM 後，在第四個clk正緣期間，

更新Data Memory，如遇到LW，要將RD輸出給 MEM / WB，

此時透過AND gate 確認目前是否為branch指令，並輸出給PCSrc

的MUX。

MEM / WB :

Fetch RegWrite、MemToReg，

以及經過Data Memory 輸出的ReadData與經過ALU後的

Address 與 WN。

離開 MEM / WB後，在第五個clk正緣期間，

由MemToReg決定是否要將結果寫入WN，在回傳WD，

這樣即可完成一個cycle。

**二 心得感想 :**

當處理MIPS Pipeline CPU時，遇到的困難點主要有以下幾個方面：

時脈處理：Pipeline中的時脈同步是一個複雜的問題。需要確保各個階段在正確的時脈下進行操作，並解決時脈衝突的問題。

Pipeline結構：MIPS Pipeline的結構比較為複雜，包括Instruction Fetch、Instruction Decode、Execute、Memory Access和Write Back等階段。確保各個階段的數據和控制信息正確傳遞，以及處理branch等特殊情況，是一場挑戰。

二進制與十六進制的轉換：MIPS指令使用32位二次制數據表示，而在程序中經常使用8個十六進制數據表示。需要熟知二進制與十六進制之間的轉換關係，以便正確理解和編寫指令。

Little Endian排序：MIPS實體系統採用Little Endian的順序，即低位元存儲在低地址，高位位元存儲在高地址。

這些在過程中，模擬波形圖是很好的debug方法。通過觀察和分析波形圖，可以檢查各個接線是否正確、數據是否正確等。此外，過程中使用display指令可以幫助減輕一些負擔。

完成這次的實驗，讓我們更加了解到設計一個CPU中，不只需要注意到Pipeline的效率是否有提升，還要注意會不會有時脈的衝突。

**三 分工表 :**

程式撰寫 : 彭靖傑、巫年巨

報告製作 : 蔡宗勳、鄭宇傑