

Computer Architecture Project 1

Member : B04705003 林子雋 B04902021 陳弘梵 B04902099 黃嵩仁

Team Work :

B04705003 林子雋	關鍵 module 實作(Hazard Detection, Forward Unit, Data Memory)、修正接線錯誤部分、使用 gtkwave debug 找出所有錯得到最終答案
B04902021 陳弘梵	cpu.v 接線、debug、report
B04902099 黃嵩仁	cpu.v 接線、debug、report

How do you implement this Pipelined CPU :

先以上次 single cpu 作業為基礎，先加上 load、store、branch、jump 的功能(新增 Data_memory.v、Jump.v、eq.v)，再將其沿伸至 pipeline-cpu 的架構(新增 IF_ID.v、ID_EX.v、EX_MEM.v、MEM_WB.v、MUX8.v、MUX_forward.v、HD.v、FW.v、OR.v)。

Explain the implementation of each module :

Data_memory.v	實作 data memory，根據傳入的 signal 及 address 決定傳出/寫進的資料。
Jump.v	將 mux1 (mux_branch)的 output 取其最高 4 個 bits，與(從 IF_ID output 的 instruction)shift left 2 bits 後的 address concatenate 後輸出。

Eq.v	判斷從 rs 與 rt 讀出的 data 值是否相同，相同輸出 1。
IF_ID.v	instruction memory 與其他 module 的中繼，若收到 OR.v 傳來的 flush 信號，則將輸出的 instruction、pc 歸 0。
ID_EX.v	instruction、register output 與 control signal 的中繼站。
EX_MEM.v	control signal (memory、write back 相關)、memory address(ALU result)、write data (from mux_forward7)、register address (from mux3)。
MEM_WB.v	作為 write back signal、data memory、register address 的中繼站。
MUX8.v	若無 hazard，則輸出從 control 傳入的 signal，若有 hazard，則輸出 0。
MUX_forward.v	接受從 forward unit 傳來的 signal (2 bits)，來決定輸出為 alu result、mux5 result、register data (rs data or rt data)三種中其中一種。
HD.v	檢查有無 hazard 發生，若檢查到 hazard，則通知 mux8、IF_ID、PC。
FW.v	依照上課講義撰寫。

OR.v	or jump/branch signal 。
------	-------------------------

Problems and solution of this project :

1. 線路過多，接線時很容易混淆，需要有清楚的邏輯。
2. 線路接錯，gtkwave debug 好用。
3. Verilog 的 compiler 所提供的資訊過於簡略。
4. 一開始不知道 wire 的實際運作，後來看了 gtkwave 才比較知道 wire 是一瞬間就改變的
5. 每個 latch(IF/ID.....)module 中，都要做 initial begin end 來初始化 output 的 reg 值，才不會造成在 Register.v 同時寫入和讀取時的錯誤。