Computer Architecture Project 1 Report

楊傑森 Jason Yeo B07902101

OS 作業系統: Windows

Implementation of new modules（比對 HW4: Single cycle datapath):

1) Hazard Detection Unit - for inevitable stalls and flushes if control hazard exists. (& detect load use data hazard). The flush signal will be set to 1 under these 2 conditions together:

a) The previous instruction is a load, when current instruction enters ID stage.

b) Destination register in load = Source register in instruction in ID stage.

If the instruction is flushed, means the decoding will become 0 and stalled for one cycle. The instruction will be decoded again at the next cycle. After stalling for one cycle, the forwarding unit can handle data hazards by forwarding data from load instruction in MEM to next instruction in use in EX stage. This is because the load instruction finished loading from data memory at the end of MEM stage.

2) Branching Unit - if branch signal from control is 1 and registers output are the same then flush signal is 1

3) Forwarding Unit - avoid unnecessary stalls. If there is EX/MEM hazard, it will forward the result from the previous instruction to either input of the ALU. If there is MEM/WB hazard, it will forward the result from the second previous instruction to either input of the ALU.

4) ImmGen - selects a 12-bit field for lw, sw, beq and sign-extend. Left shift operation is already done here.

5) Pipeline Registers - take inputs and temporarily store into registers and pass to the next stage. Values inside the pipeline registers are initialized to 0 in these 4 modules.

6) ALU & ALU_Control - add 3 more conditions for 3 more instructions: lw, sw, beq (refer control signal). Unlike ALU in HW4, Zero (for branch) is removed here.

7) Forward_MUX - Choose either registers output, ALU result or WriteData as output, depending on forwarding signal from Forwarding Unit

8) Adder, MUX32 - description same as HW4 because modules fully imported from HW4

9) Control - Modified from HW4. Added input signal NoOp (no operation) and output signals MemtoReg, MemWrite, MemRead to Register_ID_EX and branch signal to Branching Unit.

10) Data Memory- input address and write data, outputs read result, dependent on MemWrite and MemRead signals.

11) Instruction Memory - Instruction is decoded into 32 bits and fetched to Register_IF_ID

12) PC - Modified from HW4 but dependent on Flush signal from Branch Unit and PCWrite signal from Hazard Detection Unit

13) CPU - declare wires and connect ports

Difficulties while debugging & solutions 除錯時出現的問題和解決方案:

疑:

Registers, PC, Data Memory 的值沒有更新。

有的 registers 和 data memory 輸出在 debug 某個階 段後變很大，大概 5 位數。

翻了 instruction.txt 發現漏掉 or 指令條件(machine code 和 comment 都一樣), 畢竟此 Project 是從 HW4 修改，HW4 也沒有要求/實作 or 指令條件。

解:

immeGen 裡面 sign extend 部分修改 19b’1, 19b’0 成 19b’1111111111111111111, 19'b0000000000000000000

Hazard Detection Unit 裡面要考慮 RS1(instruction[19:15]), RS2(instruction[24:20])非零

ALU_Control 和 Control 按照 slides 提供的 machine code 和 control signal 修 改 lw,sw,beq 的部分

新增 or 在 ALU, ALU_Control 指令條件