ECEN 651 Lab Report Pre6

1. Questions

1.1 A data hazard exists in an inorder processor when an instruction reads from a register that is beingwritten to by a previous instruction. This type of hazard is commonly referred to as a Read-After-Write (RAW) hazard. RAW data hazards must be handled properly in order to maintain correct codeexecution. One way to deal with data hazards is to stall the pipeline in the ID stage when the hazardis detected. For the data path shown in Figure 2, how many cycles would the pipeline need to bestalled when a RAW exists between two back to back R-type instructions? What about the case wherean R-type instruction immediately follows a load? Likewise, can a branch instruction cause a RAW?What about a store instruction? Explain your answers for each of these!
Answer: For 2 back to back Rtype instructions, if no hazard, there will be 6 cycles, but with data hazard, 3 extra stalls will be in the middle. So 9 cycles in total. For R type instruction followed by a load instruction, 2 extra stalls will happen, so 8 cycles in total. Because there is no read/ write operations in branch instructions, so no RAW happens.

1.2 Simply stalling the pipeline to resolve data hazards increases the processors CPI (clock cycles perinstruction), thereby reducing its overall efficiency. What improvement might you make to the designto reduce the number of stalls? For back to back R-type instructions, can we eliminate stalls alltogether? Why or why not? What about RAW data hazards involving load instructions?
Answer: We can change the datapath to eliminate stalls. For example, by feeding back outputs of ALU to inputs. Thus the next instruction will operate with new value before stalls. For 2 R-type insts, yes. Because the stalls happen at ID phase, output in EX phase can be used at once. But for R-type and load instructions, stalls cannot be eliminated altogether. Because there will be at least 1 stall in MEM phase.

1.3 Control hazards exist when the processor executes a jump or branch (i.e. changes the path of execu-tion). Explain the reason for a control hazard involving a branch and provide a method for resolvingthis hazard. Do the same for jump.
Answers: For branch and jump instructions, the control hazards happen when next instruction has to wait for MEM cycle of the last instruction. We can resolve this by using prediction unit. Actual PC value can be obtained after EX cycle, feed it back to predict unit, and use the output as next PC.

1.4 Structural Hazards exist when two or more instructions need to use a resource in the data path at thesame time. Explain how having separate memories for data and instructions avoids a structural hazardin the MIPS pipeline. Do any unresolved structural hazards exist?

Answers: If instruction and data memories are not separated, in IF and MEM cycle, 2 load memories instructions will be at the same time, which cause competition for the same resource. No, any structural hazards can be solved.

1.5 What modification to Figure 2 would we have to make to handle the aforementioned hazards? Pleaseprovide some detail.

Answers: We have separated data memories and instruction memories, so structural hazards are no problems. For data hazards, we need a forwarding unit. For control hazards, we need to add a prediction unit.