

ECEn 528

Study Guide – Pipelining and hazards

- Read Sections C.1, C.2 (except “Reducing the Cost of Branches through Prediction”), C.3 - C.6, 3.1 of H&P
 - Things to focus on
 - The “normal” five steps of instruction execution on p. C-5 to C-6 and the classic 5-stage pipe
 - Definitions of hazards: structural, data, and control
 - Kinds of data dependences
 - Dealing with exceptions in a pipelined processor, particularly with multicycle operations
 - Identify the checks given on page C-57 in the `stall.c` code from the cache lab (hint: structural hazards are not there explicitly)
 - You do not need to know section C.3 in detail; skimming it is sufficient
 - The implications of control dependences
 - Clarifications
 - The definition of instruction latency on p. C-52 is not universally agreed upon; many architects (and manuals) will define the instruction latency to be one more cycle than this definition (thus integer instructions usually have latency of 1)
 - The buffering of results is essentially adding enough pipe stages to the low-latency functional units so that all instructions take the same number of cycles from IF to WB
 - On p. 150, when it says there three types of dependences, the first two are often called data dependences, and there are three kinds of these (true, anti-, and output).
 - Note that a dependence is a property of a program; a hazard is a property of a pipeline running a program. Dependences which have a distance greater than the pipeline depth (actually, a little less) do not result in hazards.
 - Answer the following questions:
 1. Why does a pipeline improve performance? Assume you have a multi-cycle implementation as a baseline.
 2. Why must a pipeline be restartable to support virtual memory?
 3. Give the pros and cons of permitting multicycle operations.

4. If there are three different functional units (e.g. kinds of instructions) with latencies 1, 2, and 3 and there are six pipeline stages (IF, ID, EX1, EX2, EX3, WB) and the register file is written in the first half of the clock cycle and read in the second half and there are three register file write ports, how many inputs must the bypass muxes have?
5. What are the three kinds of data and name dependences? Give the corresponding hazard name for each.
6. Why do (true) data dependences upper-bound the exploitation of ILP but name dependences do not?
7. Why is RAR not a hazard?