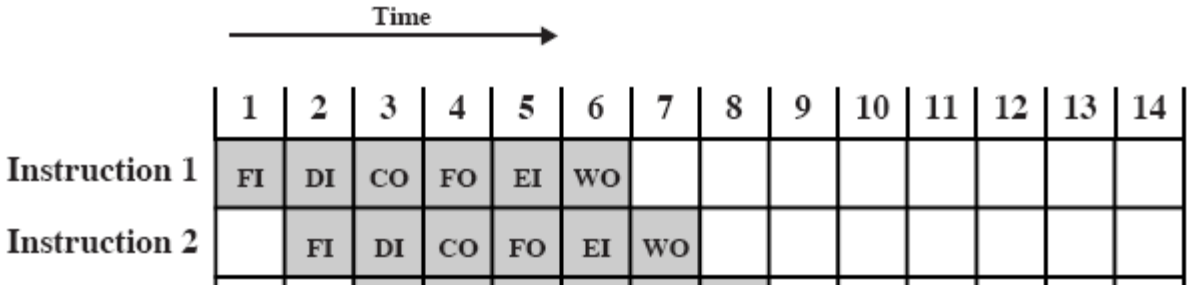
CS430 Computer Architecture

Spring 2015

More Pipelining



- Each instruction does not necessarily go through each stage of the pipeline
  1. Give two different instructions where this is the case
  2. The hope is that all of the stages can be performed in parallel. Give an example where this is not possible.
  3. The conditional branch presents a particular problem. Why?

MIPS Instructions

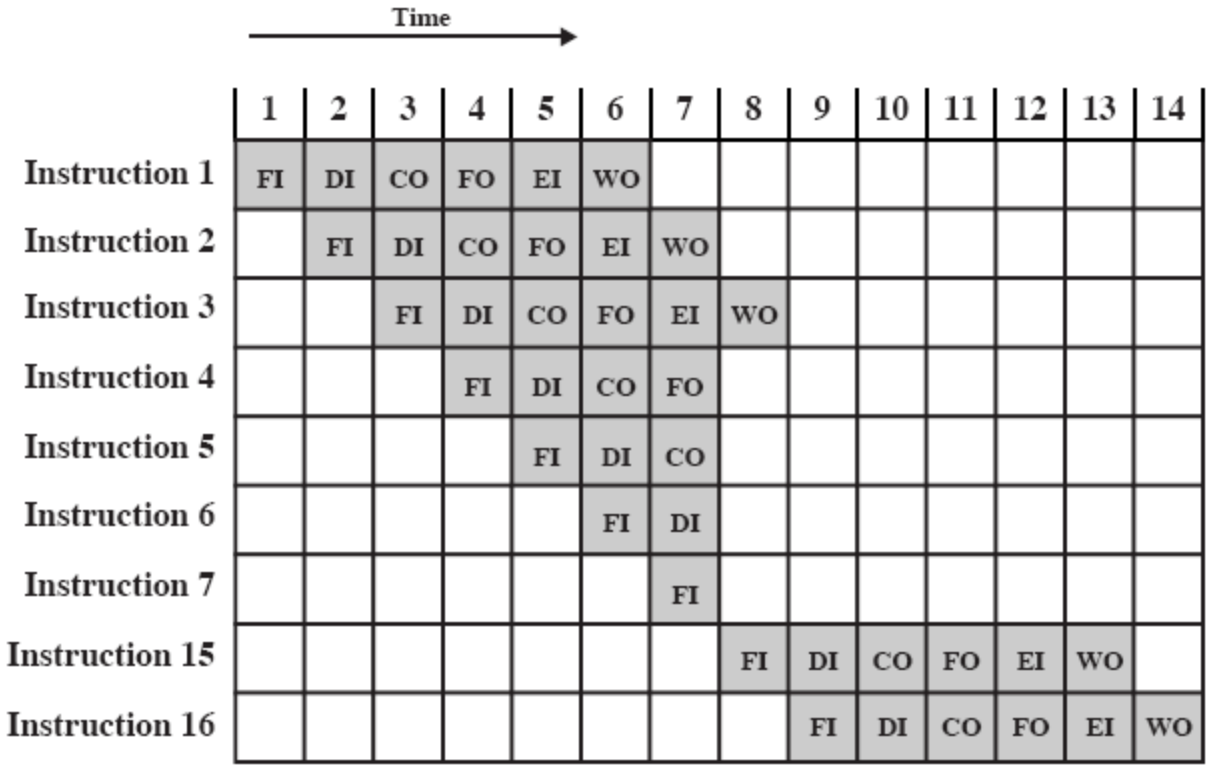
- MIPS instructions classically take five steps:
  1. Instruction Fetch
  2. Instruction Decode & Register Fetch
  3. Execute Instruction & Address Calculation
  4. Memory Access
  5. Write Back Results

MIPS Instruction Set

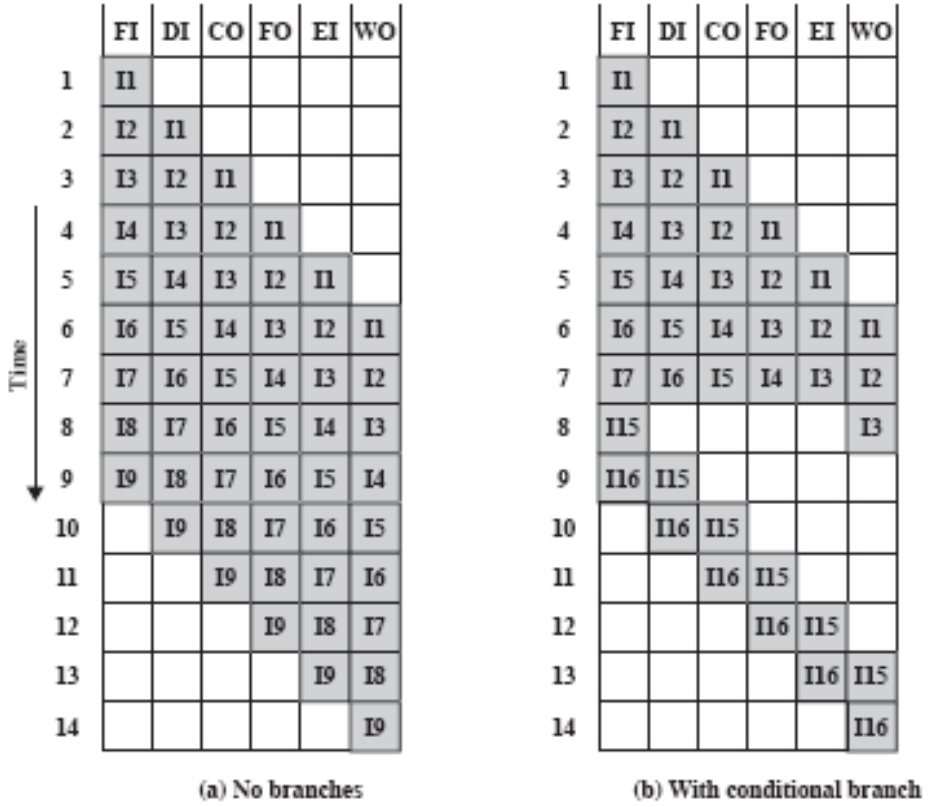
- The MIPS instruction set was designed for pipeline execution
  1. MIPS instructions are the same length. x86 instructions vary from 1 byte to 17 bytes and pipelining is much more challenging.
  2. MIPS has only a few instruction formats, with the source operand being located in the same place in each instruction.
  3. Memory operands only appear in loads or stores in MIPS.
  4. Operands must be aligned in memory.

Conditional Branch

- Explain the following diagram
  1. which instruction is the branch
  2. what is the branch penalty



Conditional Branch



Conditional Branch

- Any issues with the following diagram?

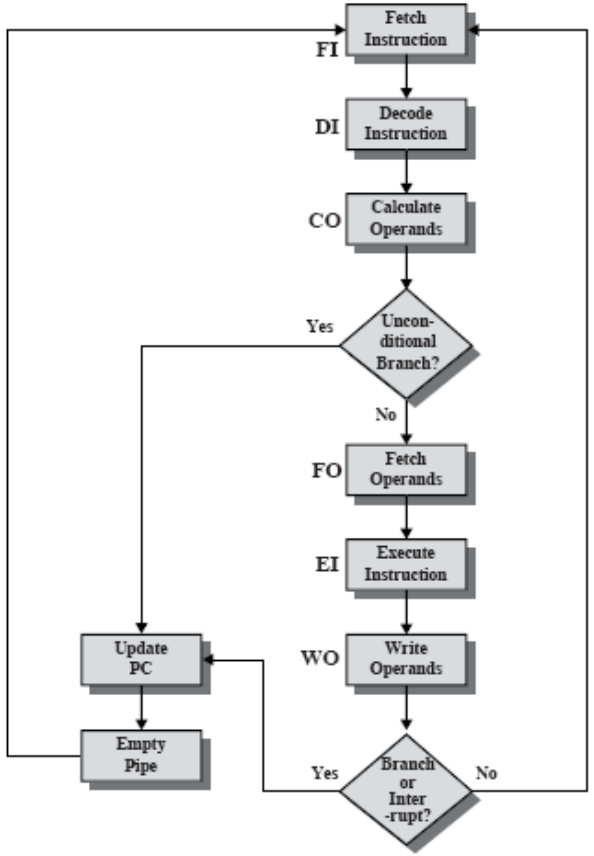


Figure 14.12. p. 499

Question

- The CO stage might be delayed because the value used as part of the calculation depends on a register that will be effected by an instruction still in the pipeline.
  - Give an example of this and show the pipeline as it would look when the problem is encountered.

Pipeline Hazards

1. Give an example of each of the following hazards

- Structural hazards: Hardware cannot support the combination of instructions because the instructions need the same resource.
- Control hazards: Need to make a decision based on the results of one instruction while others are executing.
- Data hazards: An instruction depends on the results of a previous instruction still in the pipeline.

Pipeline Observations

- It is not necessarily the case that adding more stages to the pipeline increases performance because of the overhead involved with moving data along in the various buffers of the pipeline.

- Increasing the number of stages increases the overhead of the control logic used to handle memory & register dependencies and in optimizing the pipeline.