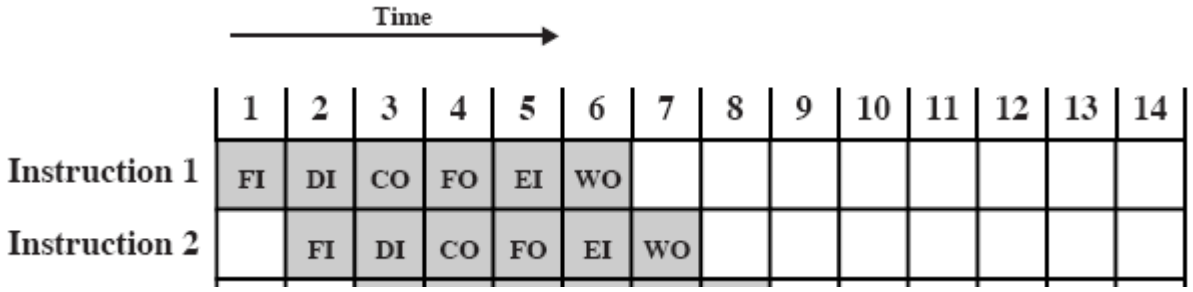
22. Instruction Pipelining II

Chapter 14: section 14.4

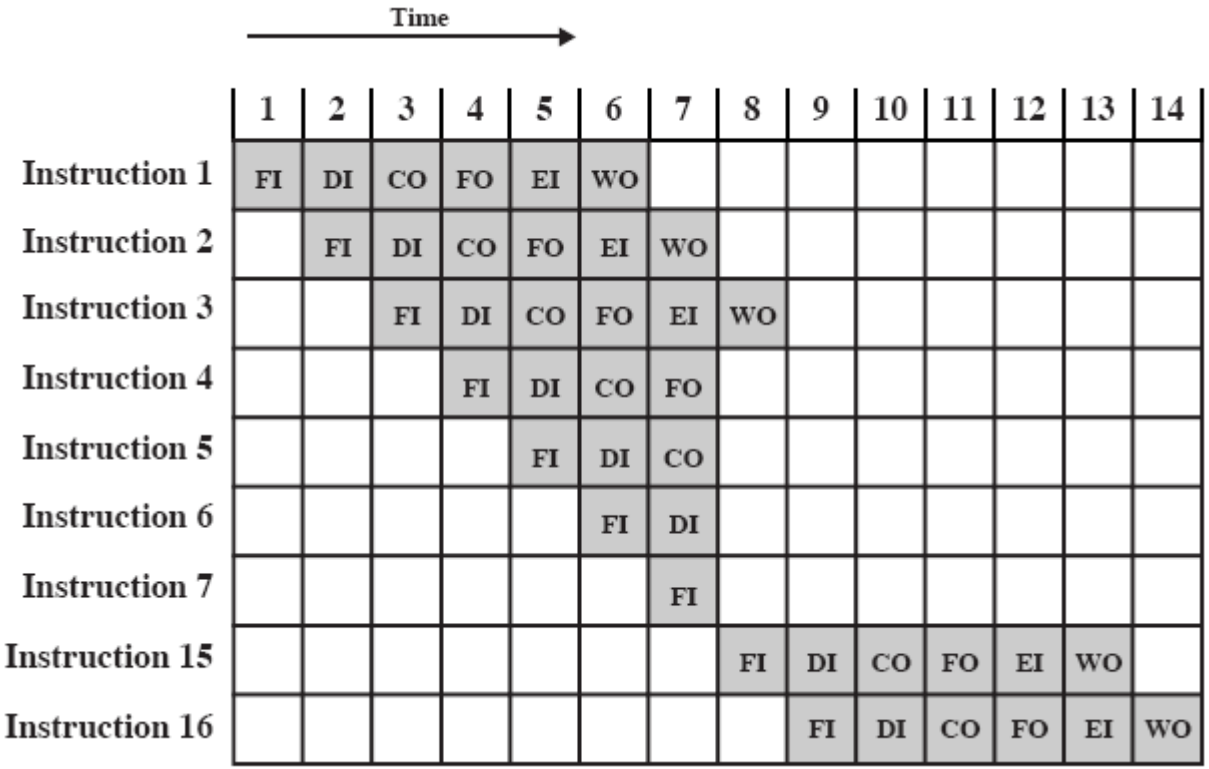
Pipelining Effectiveness



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

Conditional Branch

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



Dependency Delay

- 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

- Give an example of each of the following hazards

- **Resource/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.