

Introduction to Pipelining

Tom Kelliher, CS 220

Nov. 26, 2007

1 Administrivia

Announcements

Assignment

Read 6.2.

From Last Time

Exceptions.

Outline

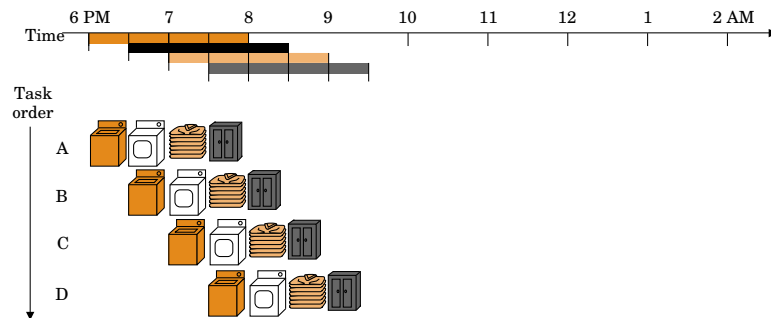
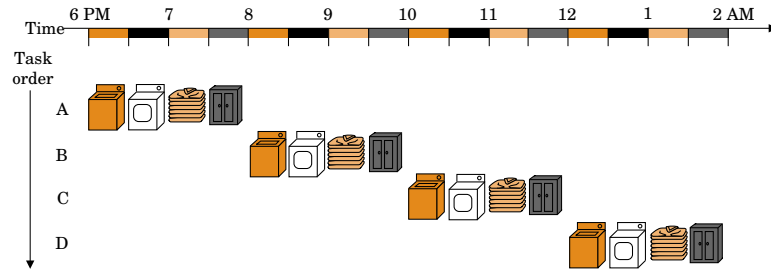
1. Introduction to pipelining; comparison with single-cycle implementation.
2. Architectural features encouraging pipelining.

Coming Up

Pipelined datapath.

2 Introduction to Pipelining

The laundry analogy:



The five stage MIPS pipeline:

1. Instruction fetch.
2. Decode and read registers.
The consistent placement of the source registers permits this.
3. Execute ALU operation or calculate an address.
4. Access memory.
5. Result write-back.

2.1 Comparison of Single-Cycle and Pipelined Performance

Assume:

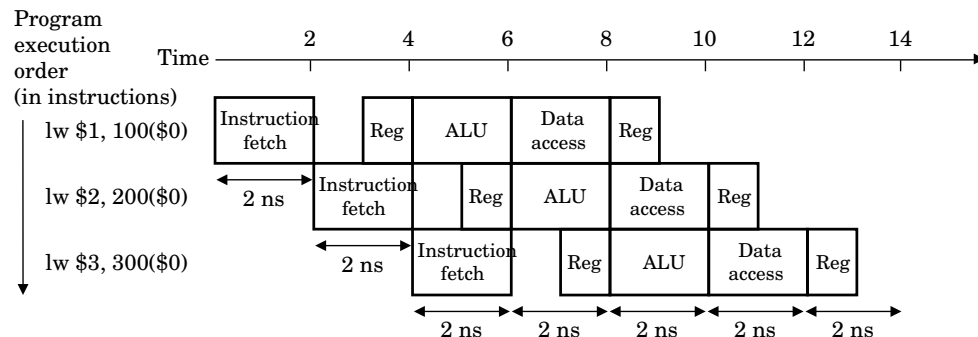
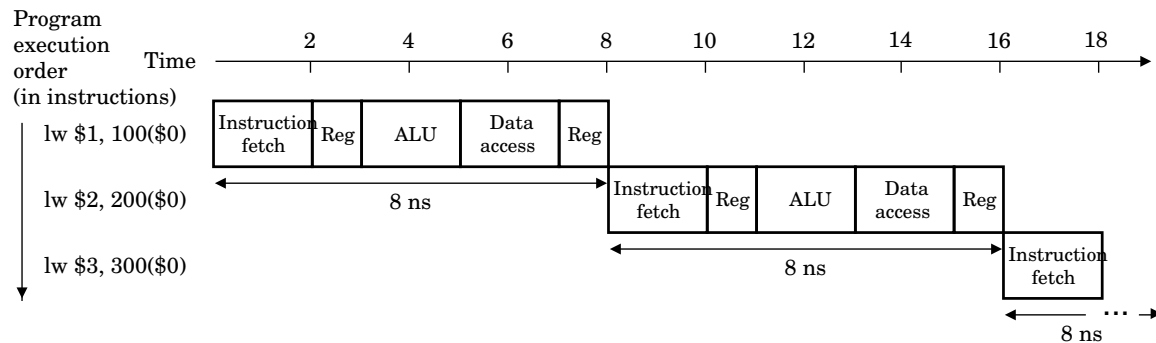
1. Memory access is 2 ns.
2. ALU use is 2 ns.
3. Register file access is 1 ns.

Instruction class times:

Instruction Class	Instruction fetch	Register read	ALU operation	Data access	Register write	Total time
lw	2 ns	1 ns	2 ns	2 ns	1 ns	8 ns
sw	2 ns	1 ns	2 ns	2 ns		7 ns
R-format	2 ns	1 ns	2 ns		1 ns	6 ns
beq	2 ns	1 ns	2 ns			5 ns

Clock periods for the two implementations?

Execution example:



Note that pipelined register file reads are done during the second half of the clock cycle and writes are done during the first half. Why?

Consider the speedup:

1. Assumption: Stages are of equal length. What if they aren't?
2. Speedup is at most the number of pipeline stages.

Do we achieve that?

Consider the execution of 1,000 instructions and compute the actual speedup.

What happened? The cost of the pipeline registers.

Consider:

- How does the speedup occur: shortened instruction execution time or higher instruction bandwidth?

3 Architectural Features Encouraging Pipelining

1. A single instruction size: simplifies instruction fetch.
2. A small number of instruction formats, with register fields in common locations for all formats: simplifies instruction decode and allows register fetch to proceed in parallel.
3. Memory operands occur only in `sw/lw` instructions: simplifies pipeline design and decreases pipeline depth.
4. Memory data is aligned: a memory operation requires only a single memory read or write.