

# Pipelining a Datapath

Tom Kelliher, CS 220

Nov. 14, 2005

## 1 Administrivia

### Announcements

### Assignment

Read 7.1–7.2.

### From Last Time

Overview of pipelining.

### Outline

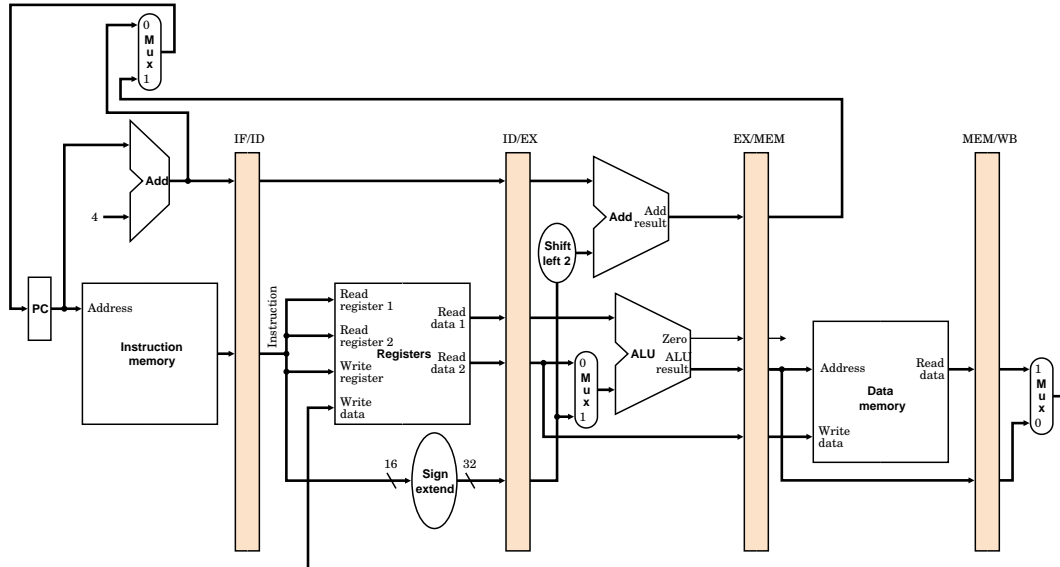
1. Pipelining: a pipelined datapath. Hazards.
2. Simple example: a single `lw`.

### Coming Up

Introduction to caches.

## 2 Pipelining

A pipelined datapath:



Consider four instructions: R-mode, a branch, LW, SW.

Observations:

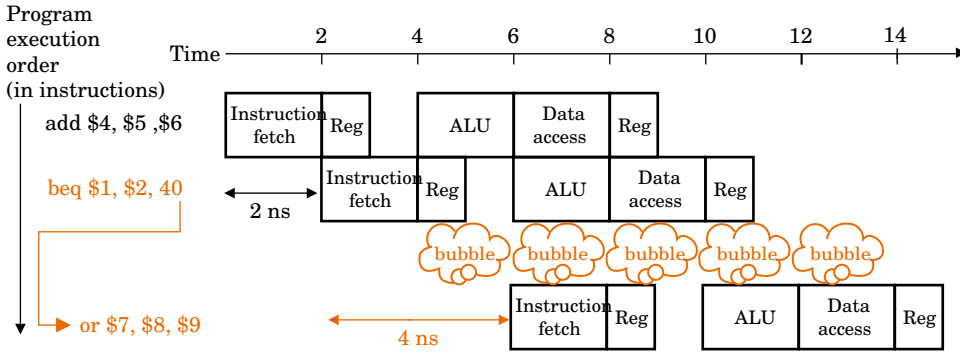
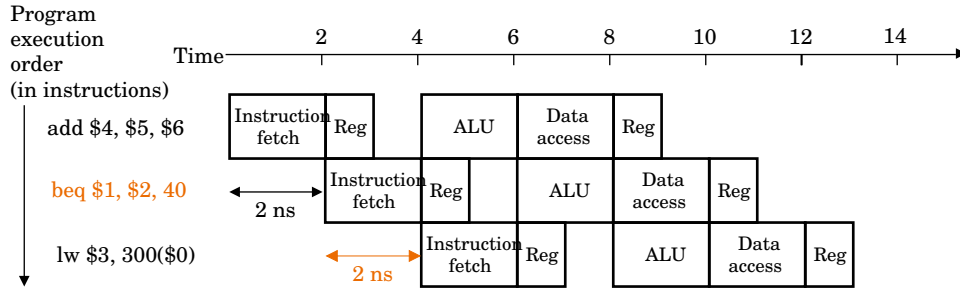
1. Not a true pipeline: feedback.
2. How do we re-design control?

### 2.1 Hazards

1. Structural hazards.

Example: unified L1 cache/memory.

2. Control hazards. Consider the following example:



Solutions:

(a) Stall.

(b) Predict.

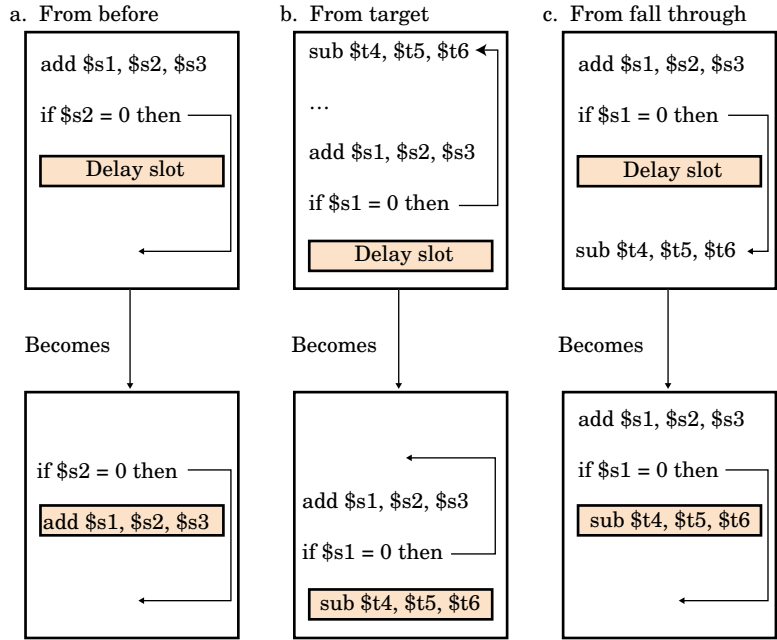
Static prediction. Truly static. Compile-time determined.

Dynamic prediction. Branch history tables. One-, two-bit counters.

(c) Delayed branch.

Assumes you know branch outcome early.

Code scheduling:

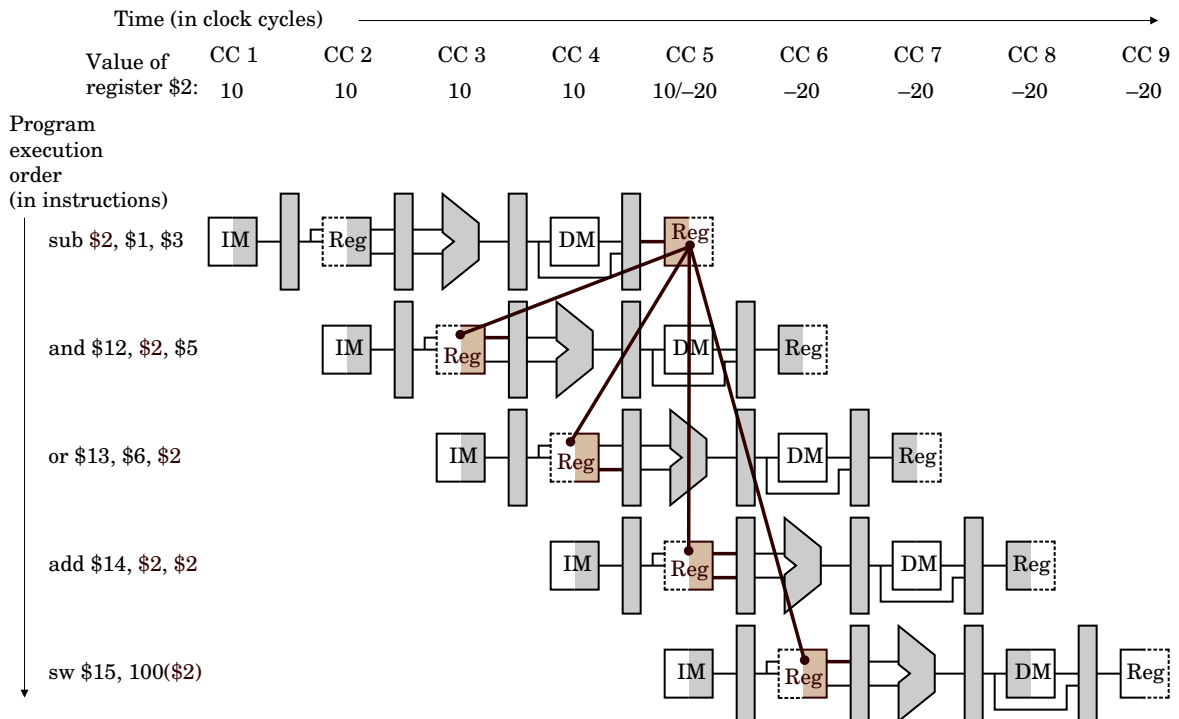


Consideration: deeper pipelines.

### 3. Data hazards.

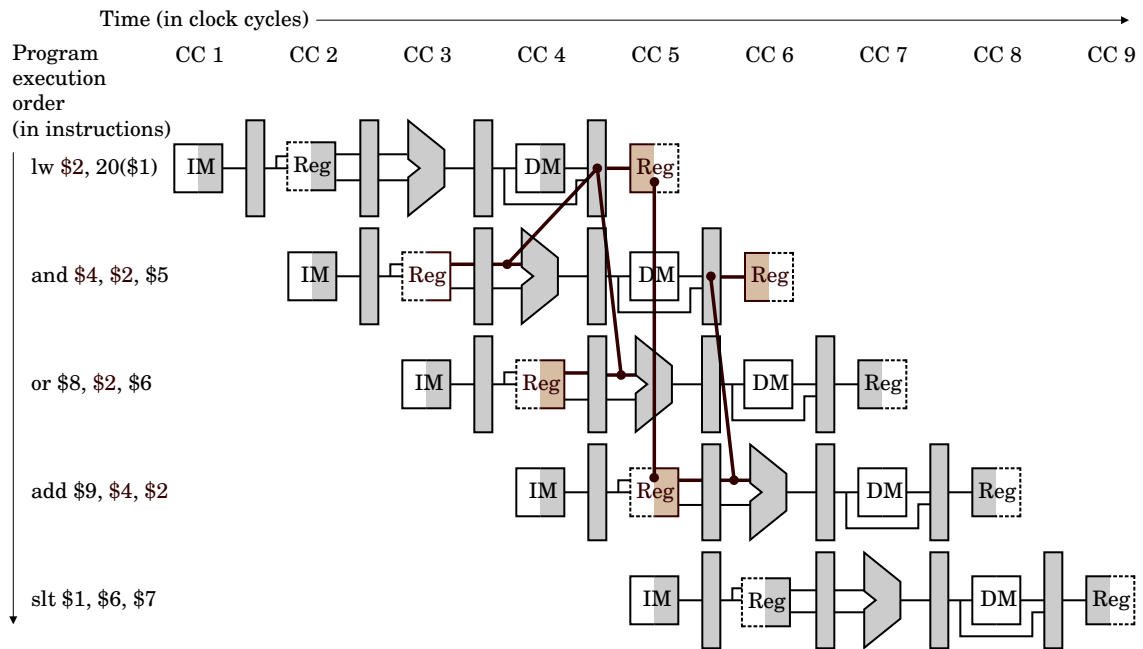
Data not available when needed.

ALU example:



Fixed by forwarding.

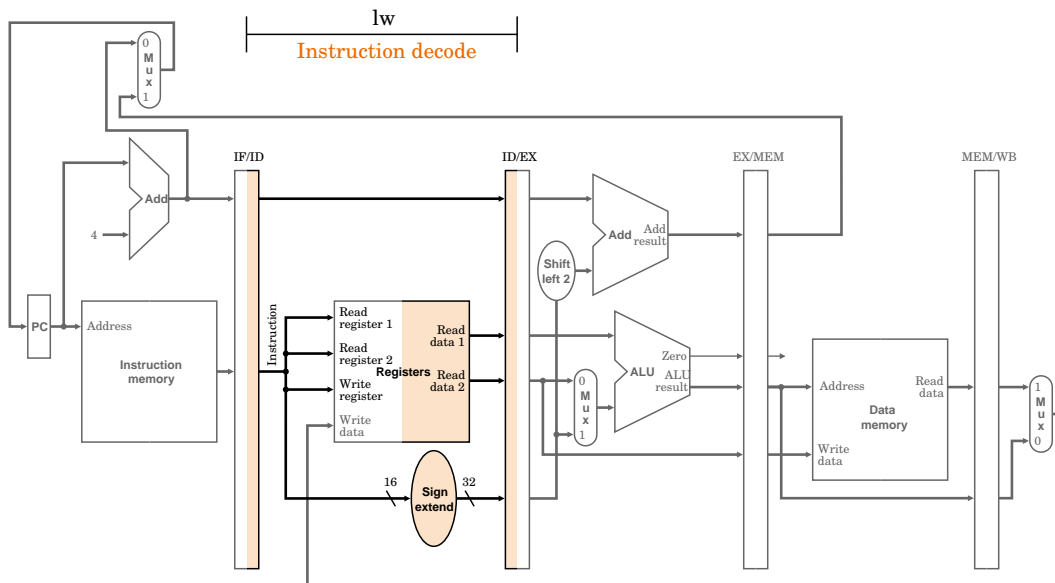
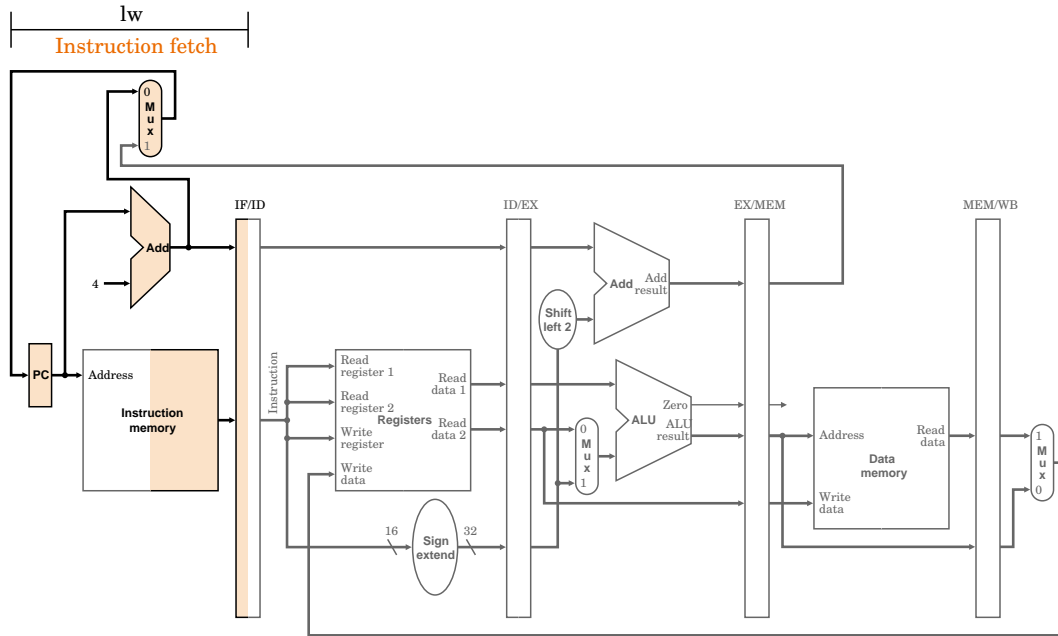
Memory example:

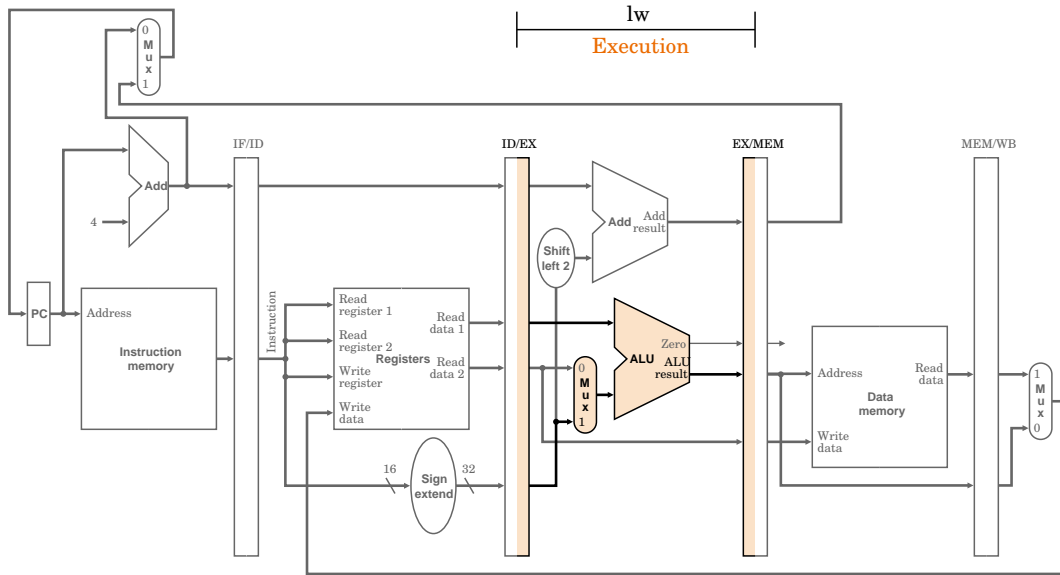


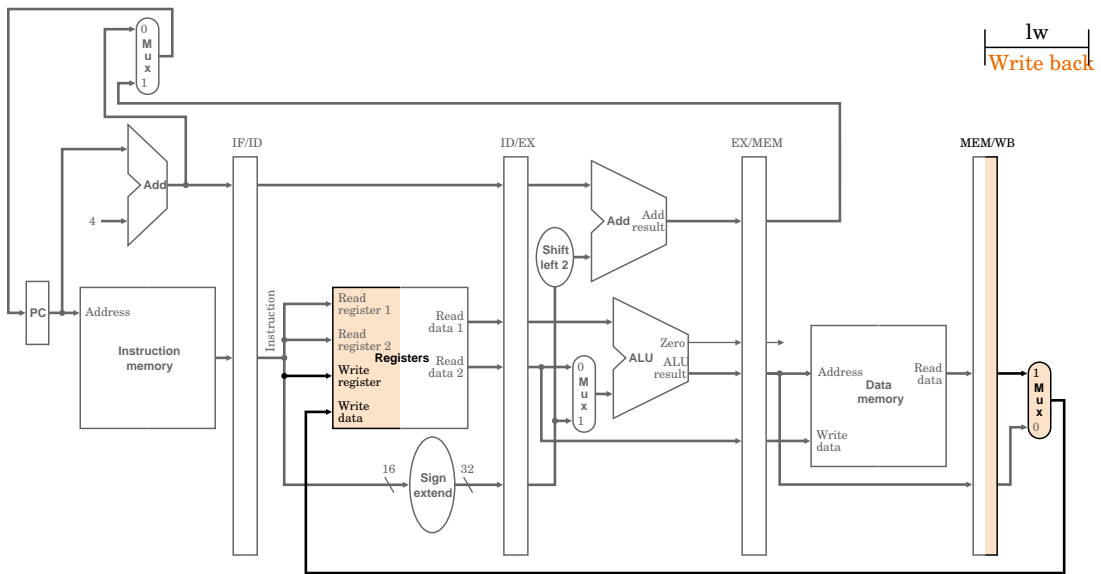
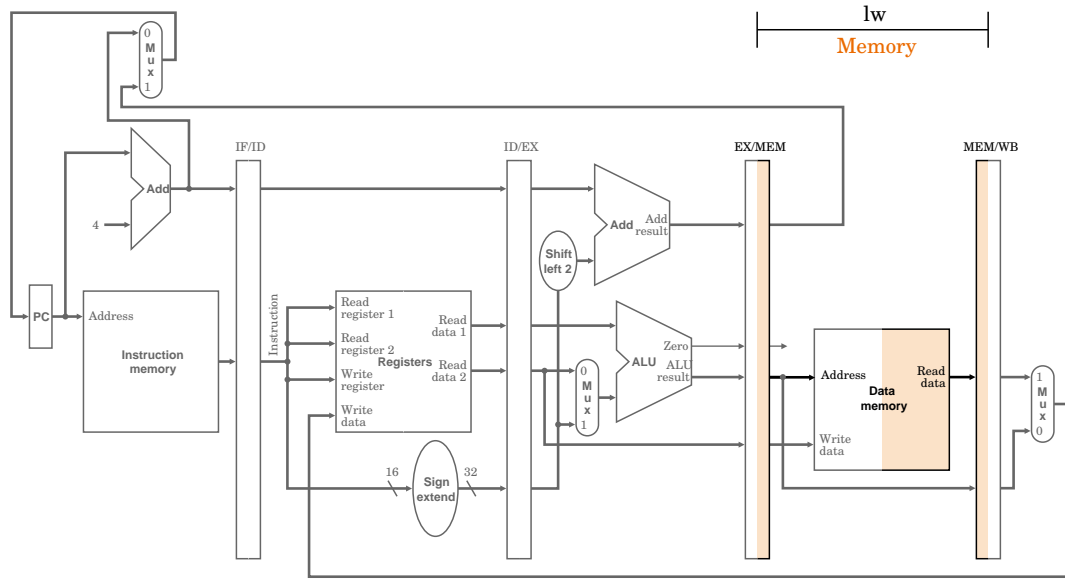
How can this be fixed?

## 2.2 Simple Example

Let's follow a lw. What's going on during each clock cycle?







97108/Patterson  
Figure 06.15