Pipelining a Datapath

Tom Kelliher, CS 220

Nov. 28, 2007

1 Administrivia

Announcements

Assignment

Read 7.1–7.2.

From Last Time

Overview of pipelining.

Outline

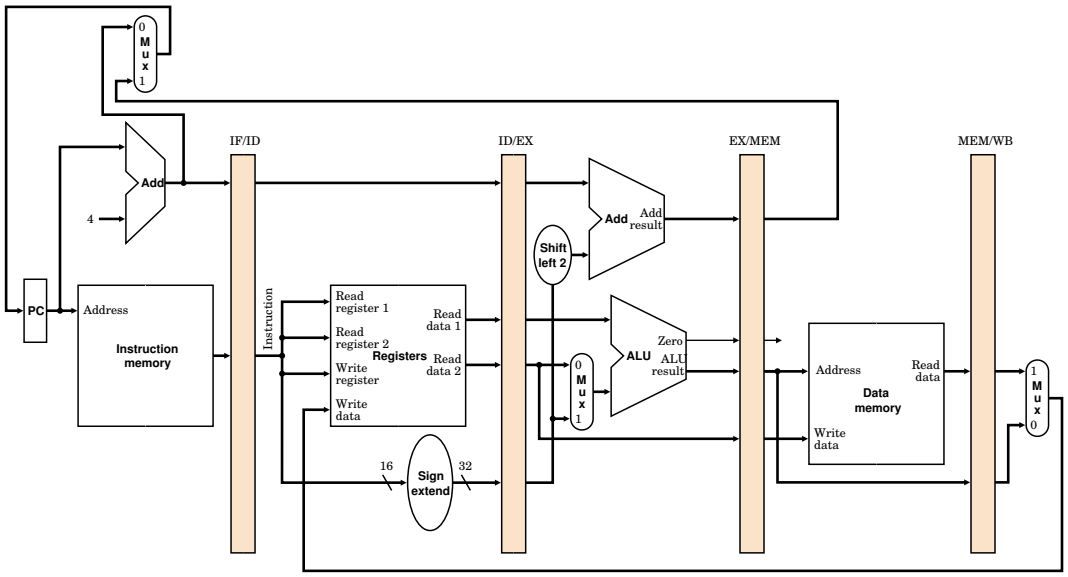
1. Pipelining: a pipelined datapath. Hazards.
2. Simple example: a single `lw`.

Coming Up

Introduction to caches.

2 Pipelining

A pipelined datapath:



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

Observations:

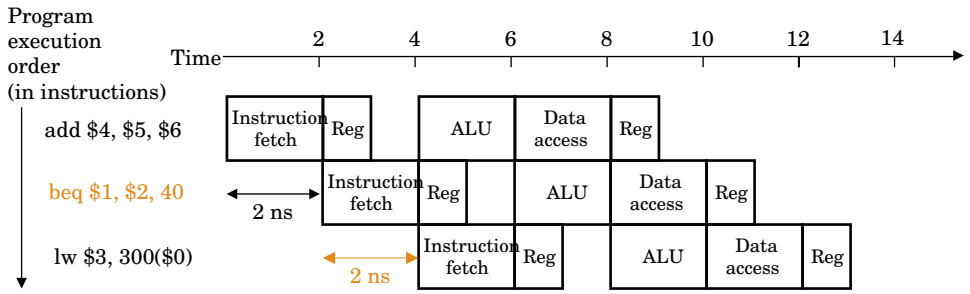
1. Not a true pipeline: feedback.
2. How do we re-design control?

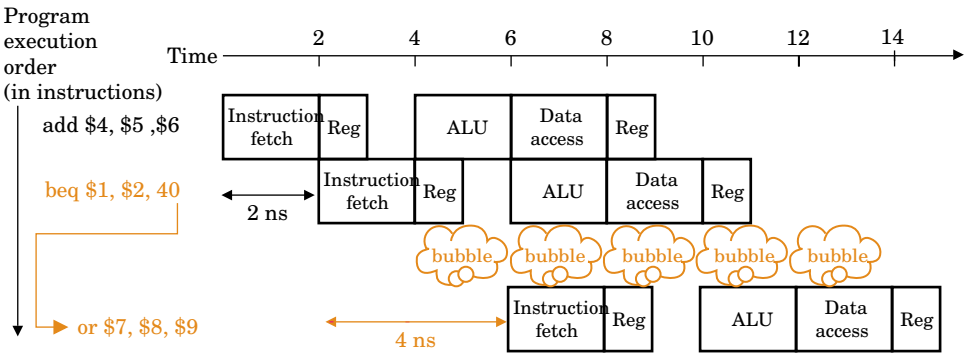
2.1 Hazards

1. Structural hazards.

   Example: unified L1 cache/memory.

2. Control hazards. Consider the following example:





Solutions:

(a) Stall.

(b) Predict.

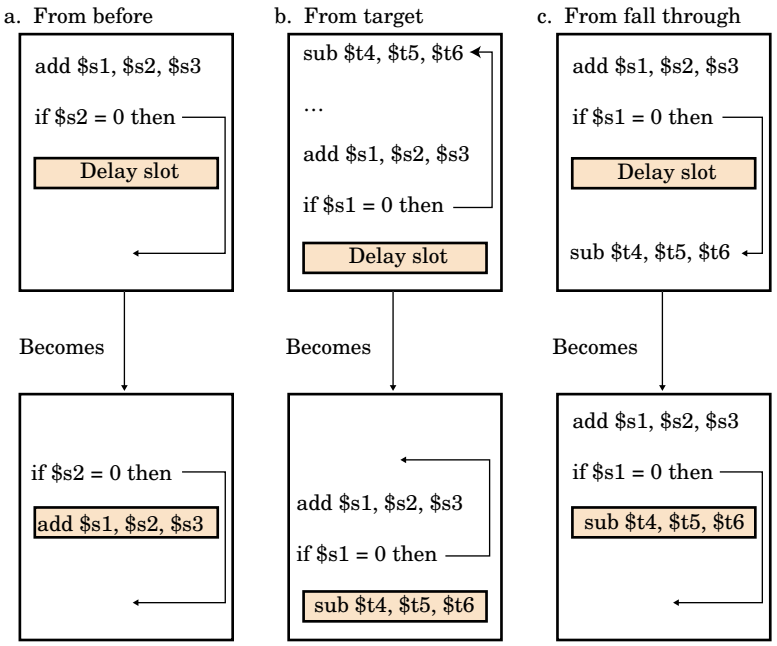
Static prediction. Truly static. Compile-time determined.

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

(c) Delayed branch.

Assumes you know branch outcome early.

Code scheduling:

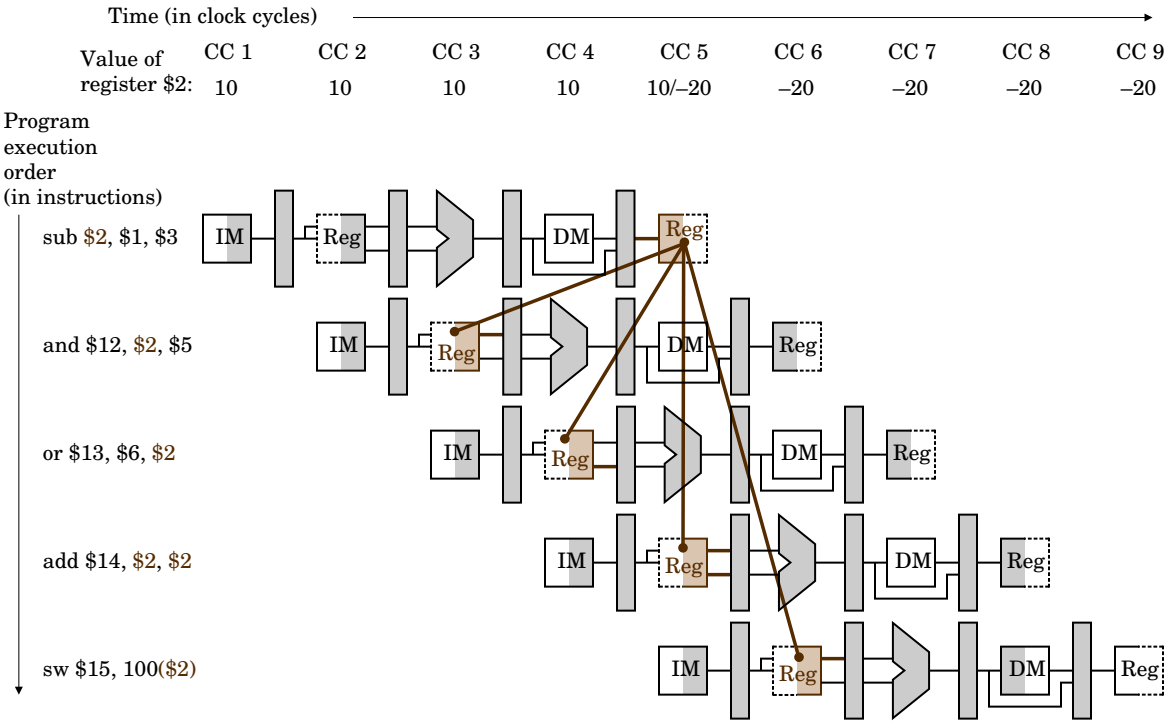


Consideration: deeper pipelines.

3. Data hazards.

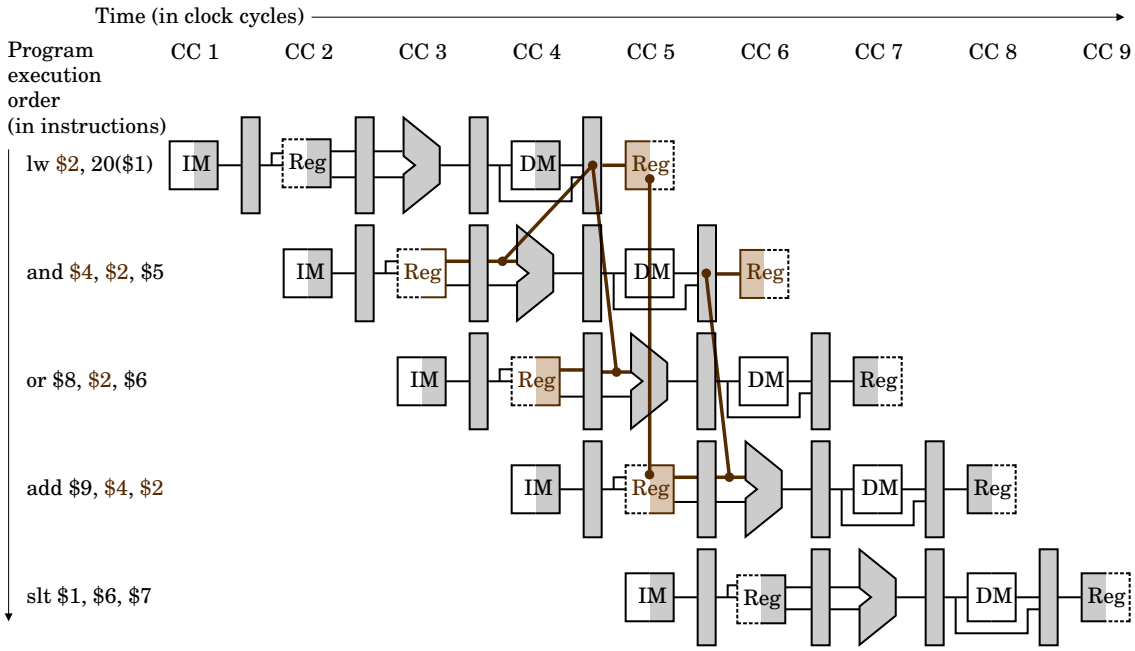
   Data not available when needed.

   ALU example:



Fixed by forwarding.

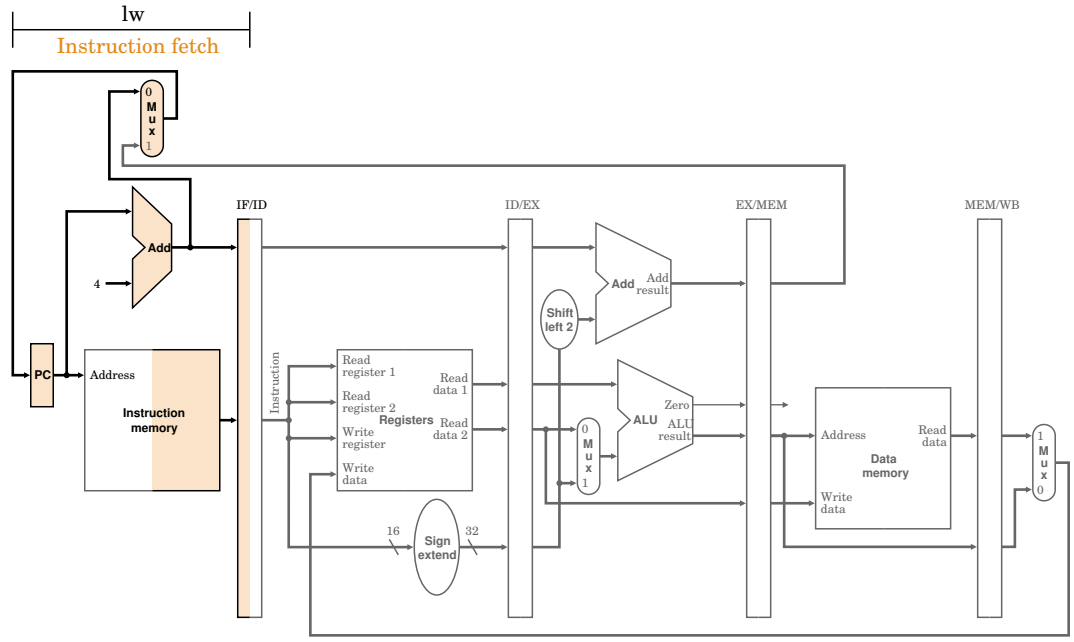
Memory example:

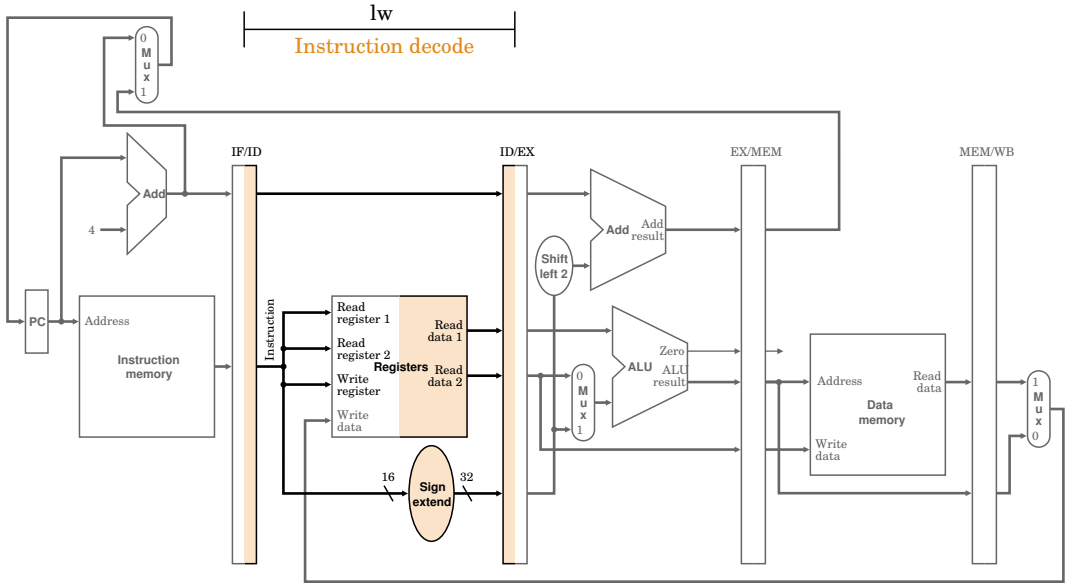


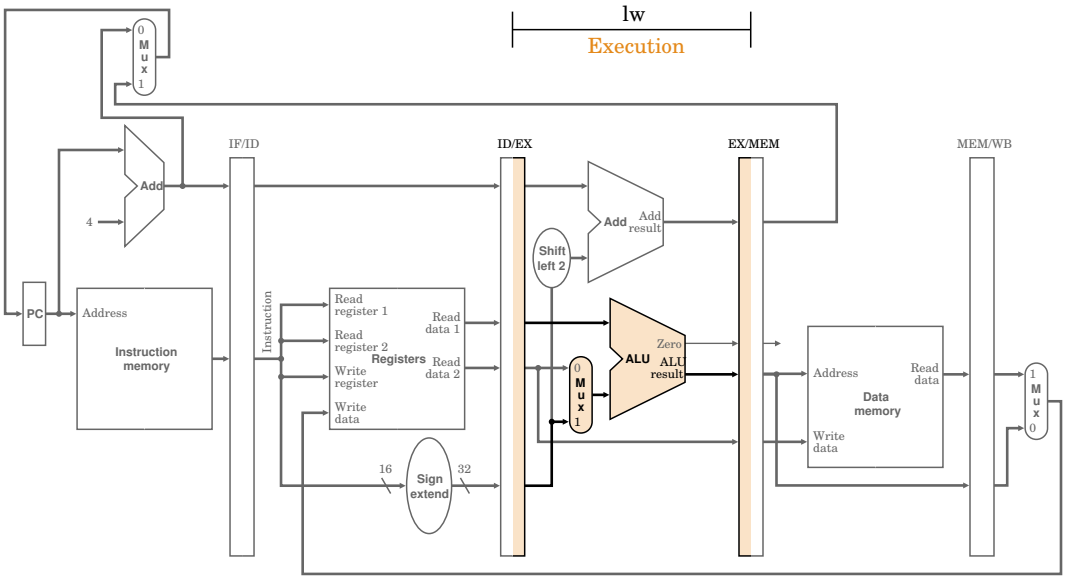
How can this be fixed?

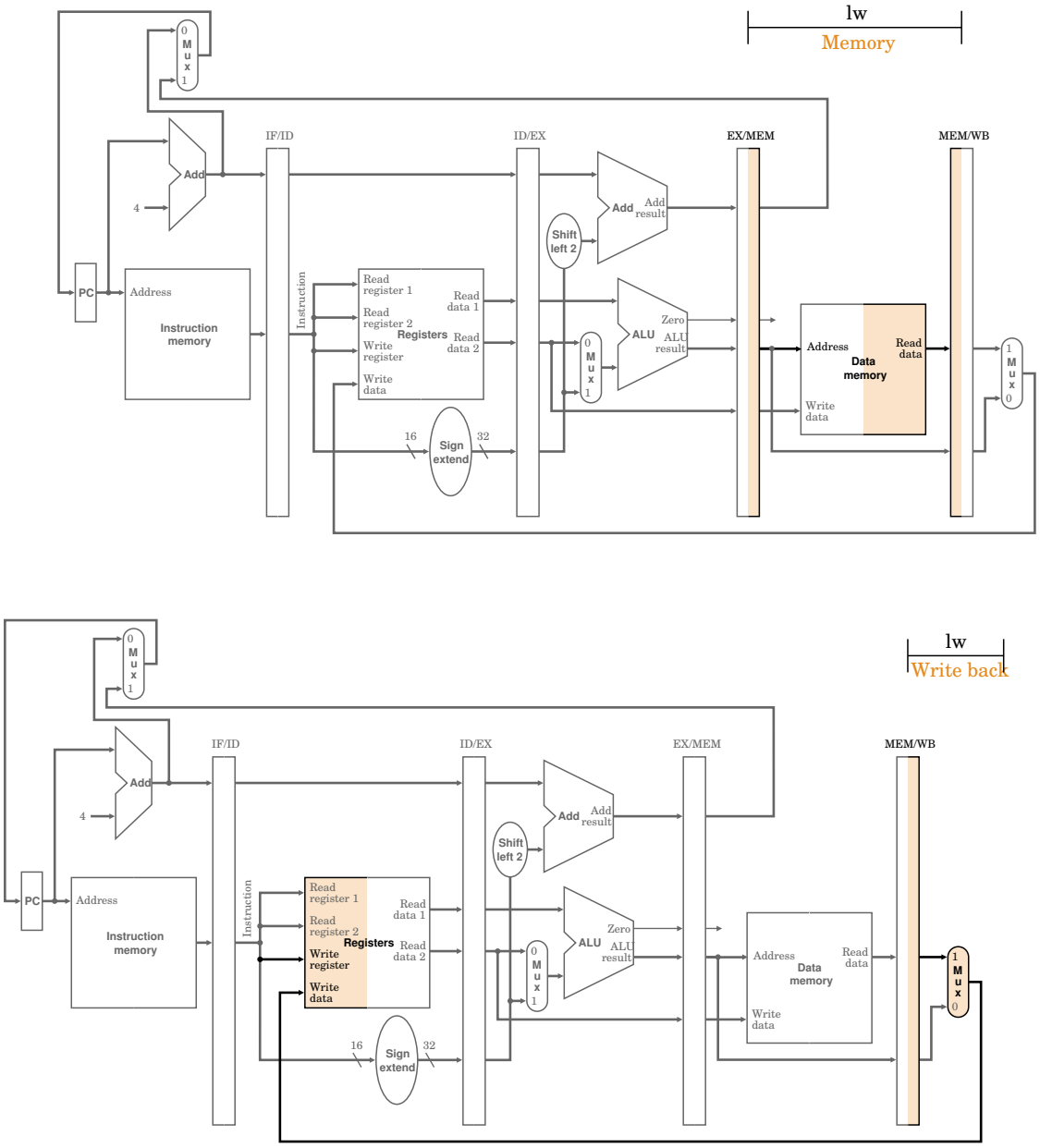
2.2 Simple Example

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


lw
Instruction fetch
IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
PC
Address
Instruction memory
Instruction
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Shift left 2
Add result
ALU
Zero
ALU result
Address
Data memory
Read data
Write data
16
Sign extend
32
M u x
0
1


lw
Instruction decode
IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
PC
Address
Instruction memory
Instruction
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Shift left 2
Add result
ALU
Zero
ALU result
Address
Data memory
Read data
Write data
16
Sign extend
32
M u x
0
1


lw
Execution
IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
PC
Address
Instruction memory
Instruction
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Shift left 2
Add
Add result
ALU
Zero
ALU result
Sign extend
16
32
Address
Data memory
Read data
Write data
M u x



97108/Patterson
Figure 06.15