

CprE 381: Computer Organization and Assembly Level Programming

Exam 2 Review

Henry Duwe
Electrical and Computer Engineering
Iowa State University

Administrative

- Exam 2
 - When: March 26 (Monday!)
 - Where (by first letter of last name):
 - A-L: 2105 Pearson (**THIS ROOM**)
 - M-Z: 0308 Elings
 - What: MIPS Arithmetic through pipelining (including hazards)
 - Aids:
 - 1 8.5"x11" notes sheet; double-sided; self-generated; hand-written
 - No electronic devices, including calculators
 - Copy of green sheet will be provided on exam
 - TODOs:
 - Review in-class assessments, HWs, **labs**, HW7 exam questions, P&H exercises, lecture slides, readings
 - Ask questions here, OH at 1:30pm, and on Canvas

Exam Focus

1. CPU Performance

- Evaluating computer performance
- CPI
- Amdahl's law
- Critical path and performance

2. Computer Arithmetic

- Basic arithmetic operations
- MIPS ALU design

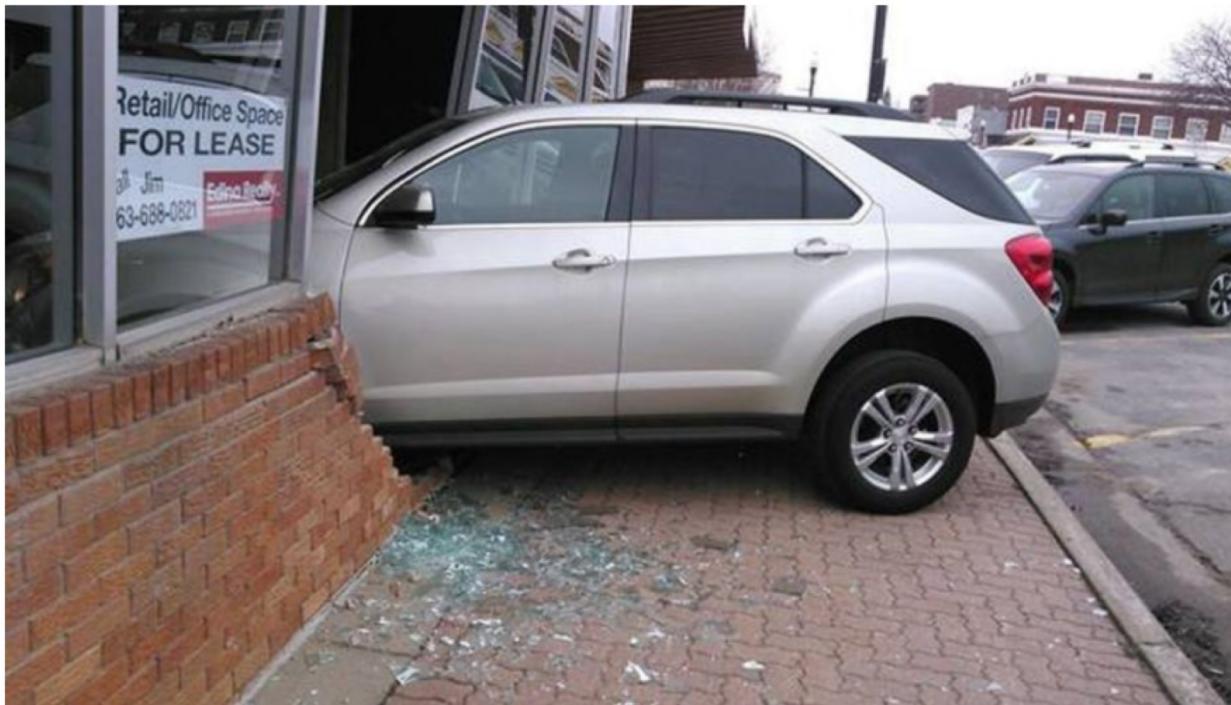
3. Processor Design

- Single cycle datapath and control
- Multi-cycle datapath and control
- Pipelined datapath and control

Today's Goal: Exam Hazard Avoidance

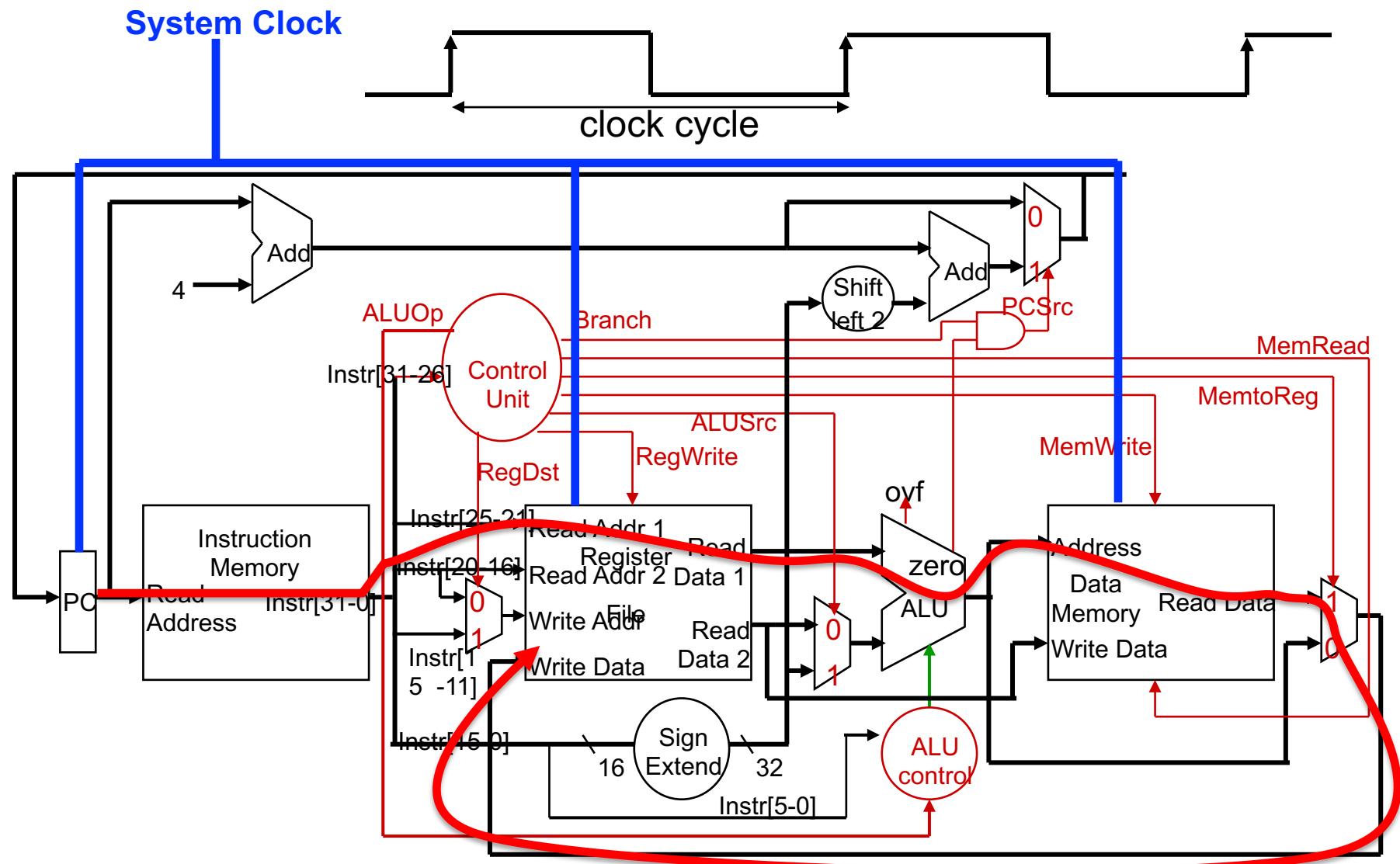
Teen crashes into exam building during her driver's test, police say

By Elizabeth Zvirz | Fox News



A 17-year-old reportedly put her car in drive instead of reverse while taking her driving exam. (Buffalo Police Department)

Review: Worst Case Timing (Load Instruction)



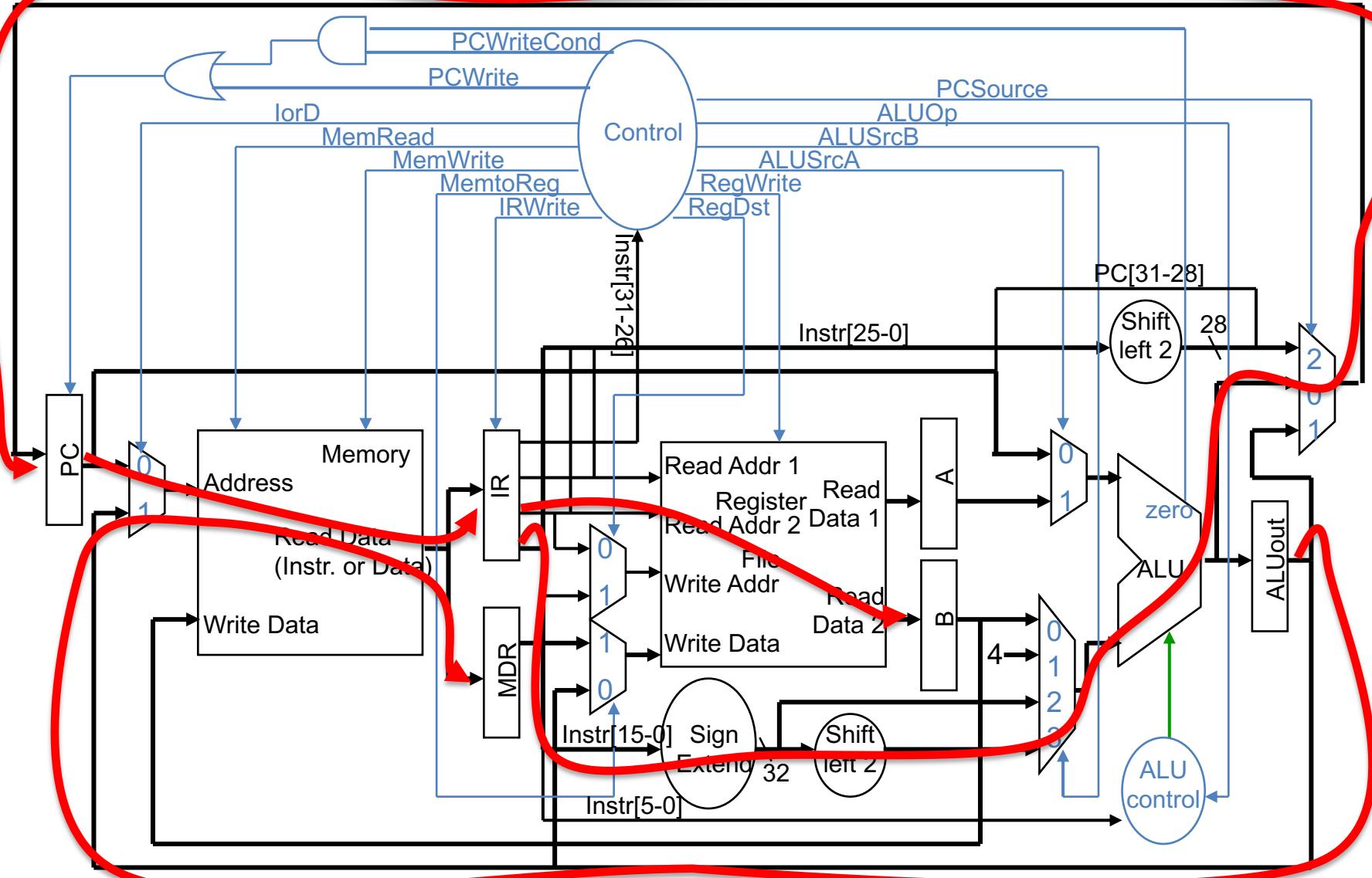
Review: Execution Time

- Drawing on the previous equation:

$$\text{Execution Time} = \# \text{ Instructions} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

- To improve performance (i.e., reduce execution time)
 - Increase clock rate (decrease clock cycle time) OR
 - Decrease CPI OR
 - Reduce the number of instructions
- Designers balance cycle time against the number of cycles required
 - Improving one factor may make the other one worse...

Review: Multicycle Processor



Review: Execution Time

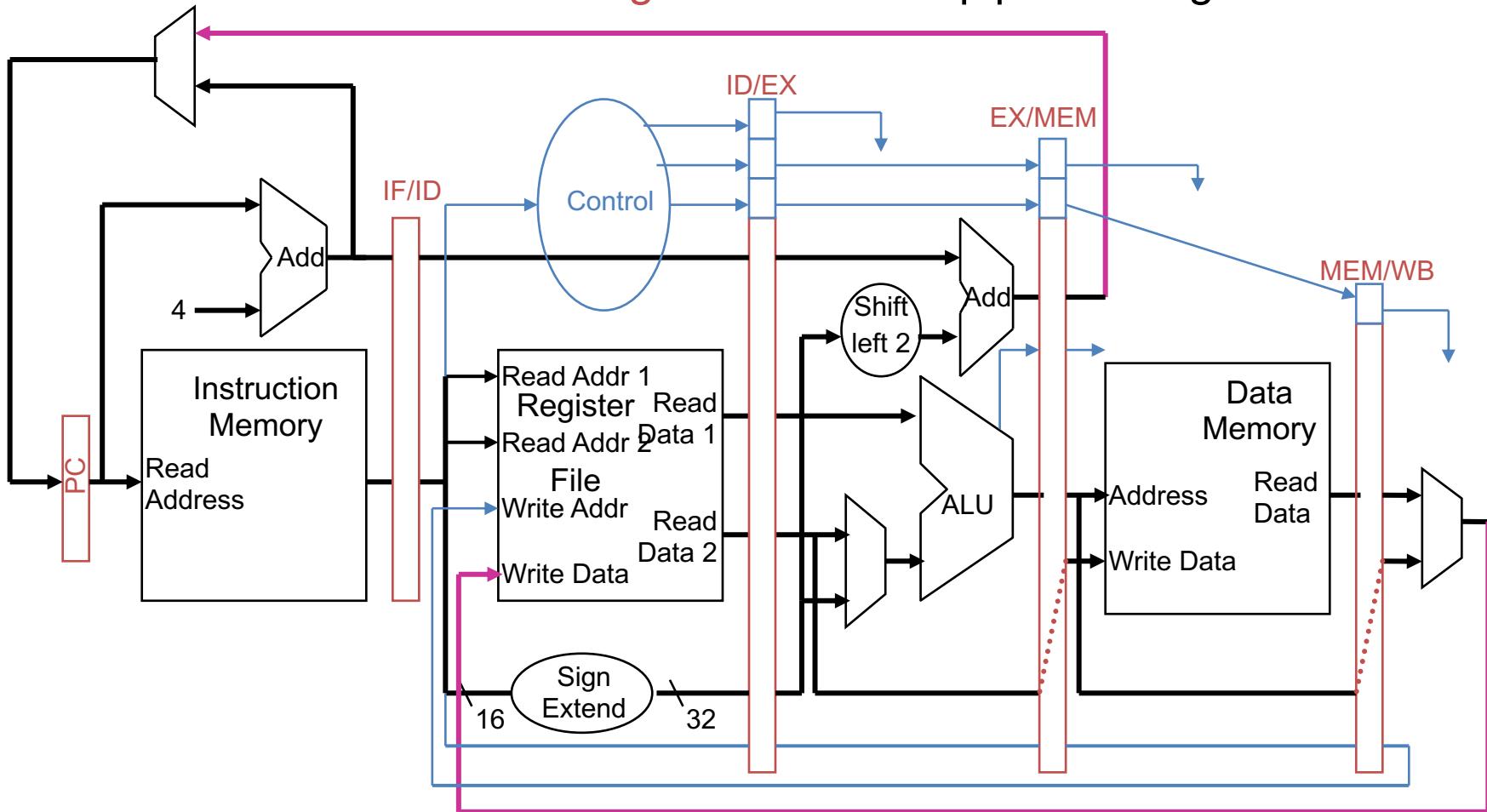
- Drawing on the previous equation:

$$\text{Execution Time} = \# \text{ Instructions} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

- To improve performance (i.e., reduce execution time)
 - Increase clock rate (decrease clock cycle time) OR
 - Decrease CPI OR
 - Reduce the number of instructions
- Designers balance cycle time against the number of cycles required
 - Improving one factor may make the other one worse...

Review: A Simple MIPS Pipeline

- All control signals can be determined during Decode
 - And held in the **state registers** between pipeline stages



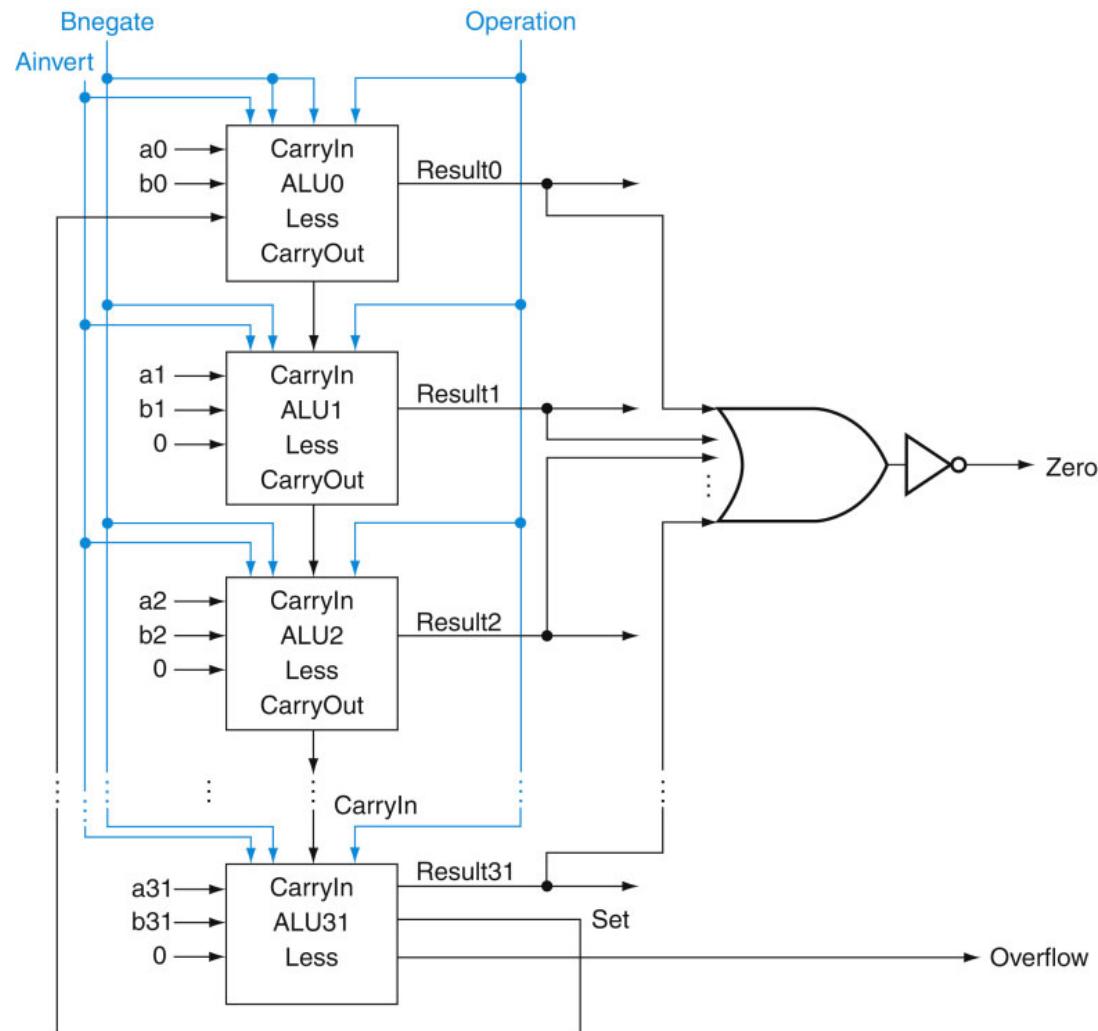
Review: Oh, the *Hazards* I've Seen

- Pipeline Hazards
 - Structural hazards: attempt to use the same resource by two different instructions at the same time
 - Data hazards: attempt to use data before it is ready
 - An instruction's source operand(s) are produced by a prior instruction still in the pipeline
 - Control hazards: attempt to make a decision about program control flow before the condition has been evaluated and the new PC target address calculated
 - Branch and jump instructions, exceptions
- Can always resolve hazards by **waiting**
 - Pipeline control must **detect** the hazard
 - And take action to **resolve** hazards



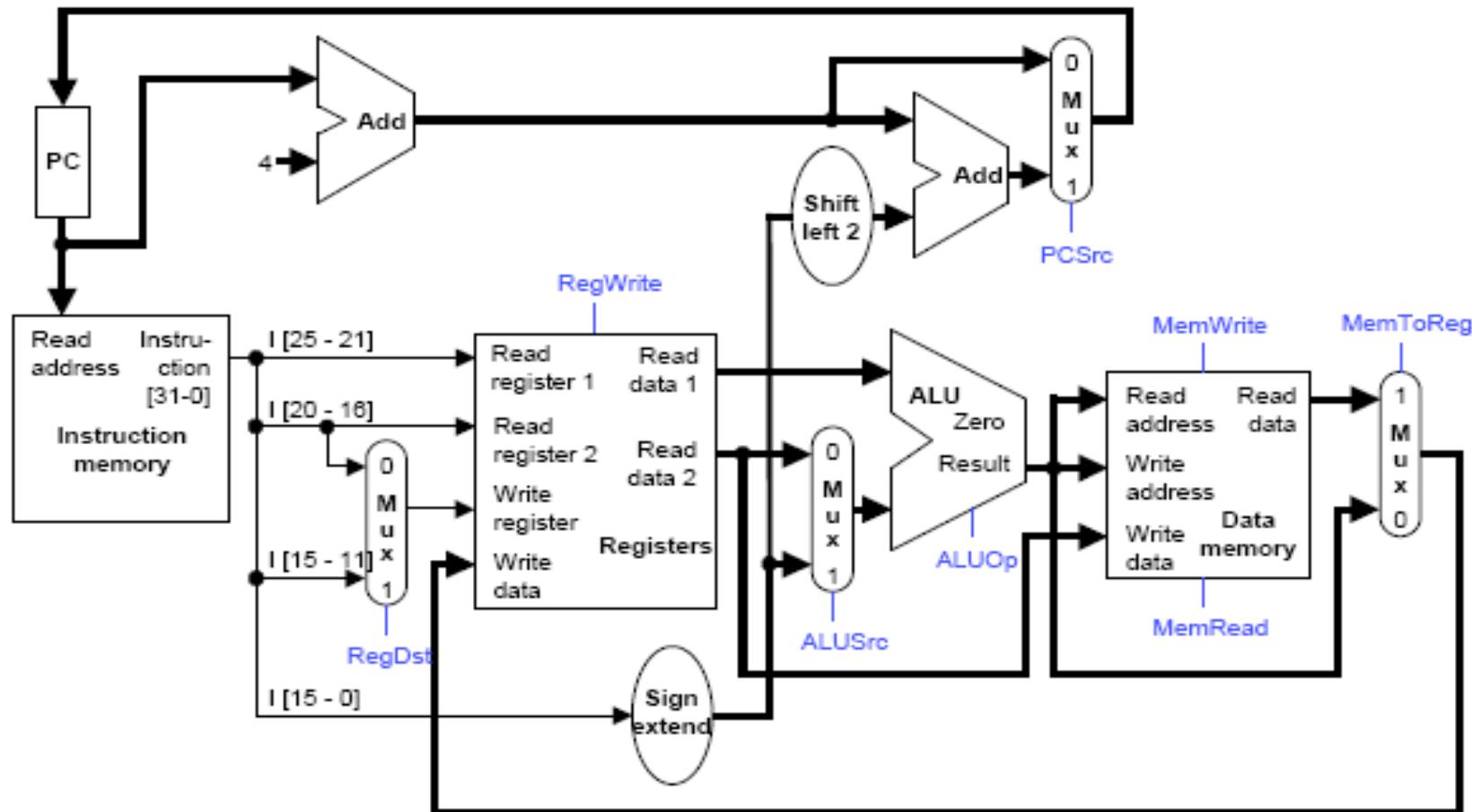
MIPS ALU Design

- When does overflow occur?
- How do you detect overflow?
- How do you perform slt?
- How do you test for zero?
- What is the propagation delay?



Sample Questions

- Add the **sc** ($M[R[Rs]] \leftarrow \text{SignExtImm16}$) instruction to the single-cycle datapath below:



Sample Questions

- What is the performance of this code on the following processor? What is the CPI?

Func: **addu** \$t0, \$a0, \$zero

addu \$t1, \$zero, \$zero

addu \$t3, \$zero, \$zero

L1: **lw** \$t2, 0(\$t0)

addu \$t3, \$t3, \$t2

addiu \$t1, \$t1, 1

addiu \$t0, \$t0, 4

bne \$t1, \$a1, L1

addu \$v0, \$t3, \$zero

jr \$ra

Instruction Type	CPI
Arithmetic (add, sub)	3
Memory (load, store)	5
Control (branches, jumps)	4

Sample Questions

- Suppose 40% of your program's runtime is spent in a single function.
- Assuming that software optimization is applied only to this function, how much faster can we make the overall program?

Pipelining Review

- What is the CPI of an ideal pipelined processor?
- What prevents having the ideal CPI?
- Does pipelining help latency or throughput?
- What are pipeline registers?
- What limits deeper levels of pipelining?
- What is the ideal speedup from pipelining?
What limits this ideality?
- How does the MIPS ISA inherently support pipelining?
- What are the three kind of hazards in pipelining?

Pipelining Review

- Give two examples of how structural hazards have been overcome in a MIPS pipeline
- How can you stall the pipeline?
- What is a branch-delay slot?
- ~~How does predict taken differ from predict not taken?~~

Sample Questions

- The 5 stage pipeline is elongated to 8 stages as shown below.
 - IF ID RR EX1 EX2 MEM1 MEM2 WB
- Results are available only after EX2 and MEM2 stages
- How many inputs does a forwarding mux on the input of EX1 have?
- How many load use delay slots will this pipeline have?

Sample Questions

- Indicate all the dependencies in the code below

```
loop: sub $t3, $t3, 1
      add $t0, $t0, 4
      add $t1, $a0, $t0
      lw   $t2, 0($t1)
      add $t2, $t2, 1
      sw   $t2, 0($t1)
      bgt $t3, $zero, loop
```

- How many stalls does one iteration of the above loop incur? Assume branches are resolved in the decode stage.

Sample Questions

- How long does this code take to execute on a standard 5-stage MIPS pipeline?
- Can we make it run faster?

```
Loop: lw      $t0, 0($s1)
      addu   $t0, $t0, $s2
      sw      $t0, 0($s1)
      addiu  $s1, $s1, -4
      bne    $s1, $zero, Loop
      nop
```

Acknowledgments

- These slides contain material developed and copyright by:
 - Joe Zambreno (Iowa State)
 - David Patterson (UC Berkeley)
 - Mary Jane Irwin (Penn State)
 - Christos Kozyrakis (Stanford)
 - Onur Mutlu (Carnegie Mellon)
 - Krste Asanović (UC Berkeley)
 - Morgan Kaufmann