CSCI341

Lecture 30, Building a Datapath

RECALL...

- The "datapath" is a representation of the flow of information (data, instructions) through the CPU
- Implemented as combination of circuitry and combinatorial & sequential chips
- "State" is created through clocking and edge-triggered flip-flops

DATAPATH ELEMENTS

- A component that operates on or “holds” data
- Memory, registers, ALU, adders, etc.

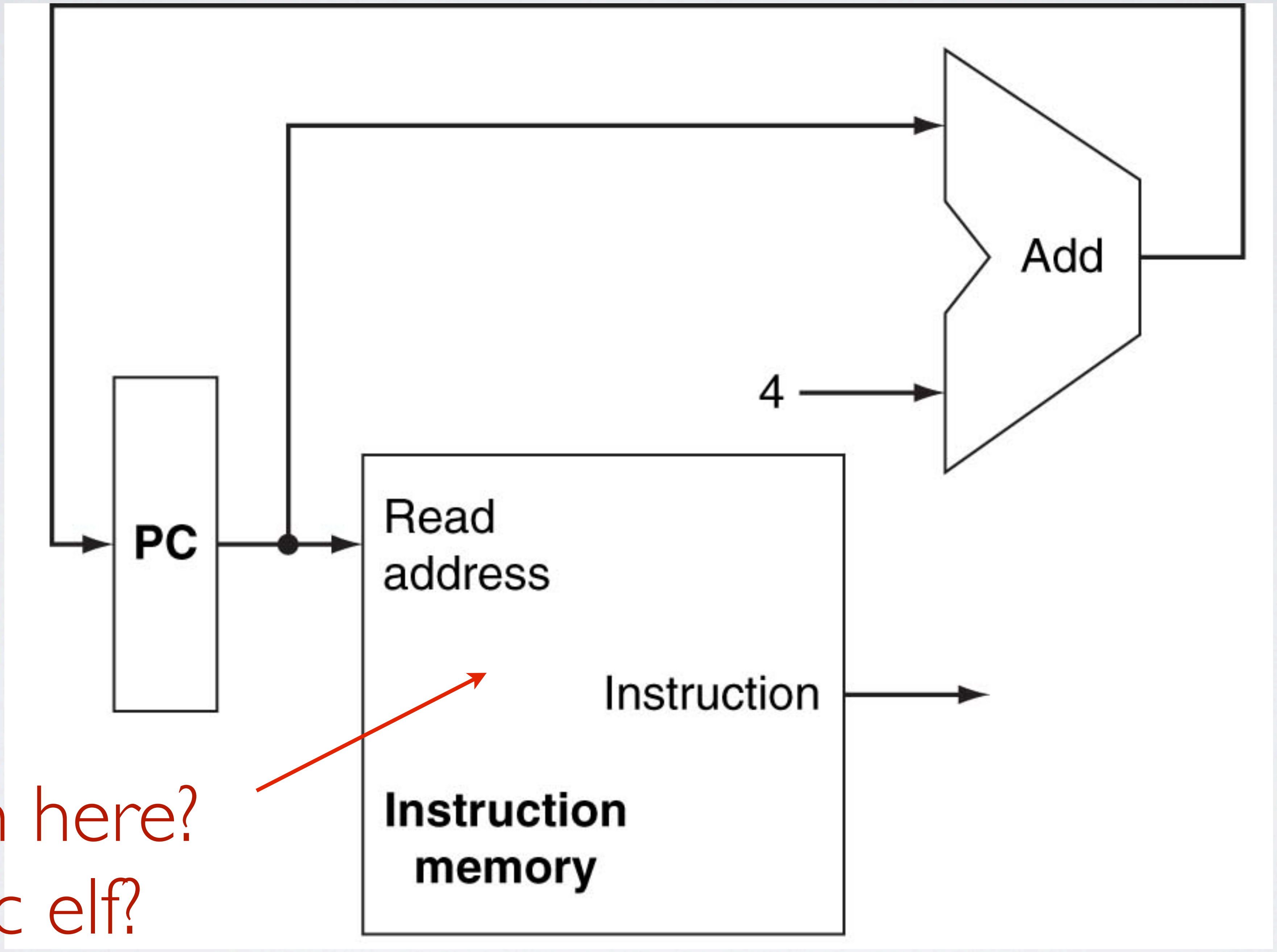
PROGRAM COUNTER (PC)

- Memory unit that increments or can be set to a value
- Value represents an instruction address

INSTRUCTION FETCH

- PC tells memory to send instruction at PC's address to CPU control circuitry
- Increment program counter, in preparation for next instruction

INSTRUCTION FETCH



What's in here?
A magic elf?

EXAMPLE

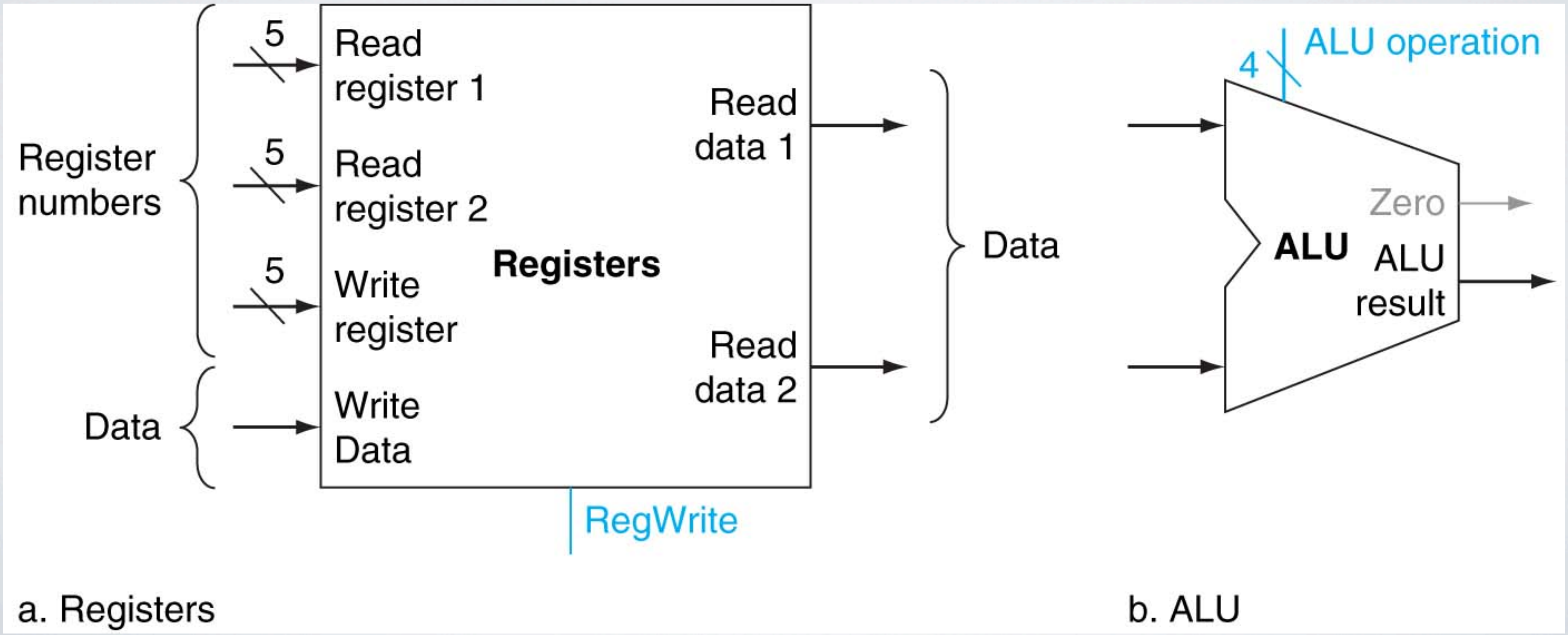
```
add $t1, $t2, $t3
```

"Read $t2 and $t3, add them together, and write to $t1."

EXAMPLE (REGISTER FILE)

```
add $t1, $t2, $t3
```

“Read $t2 and $t3, add them together, and write to $t1.”



EXAMPLE

```
add $t1, $t2, $t3
```

0	9	10	8	0	32
op	rs	rt	rd	shamt	fn

EXAMPLE

```
add $t1, $t2, $t3
```

0	9	10	8	0	32
op	rs	rt	rd	shamt	fn
000000	01001	01010	01000	00000	100000

Let’s wire it up!

EXAMPLE

```
lw $t0, 0($t1)
```

35	9	8	0
op	rs	rt	address
100011	01001	01000	0000 0000 0000 0000

Let's wire it up!

BRANCHING

```
j somewherElse
```

What must happen in order for this to work?

```
beq $t1, $t2, somewhere
```

What about this?

DELAYED BRANCH

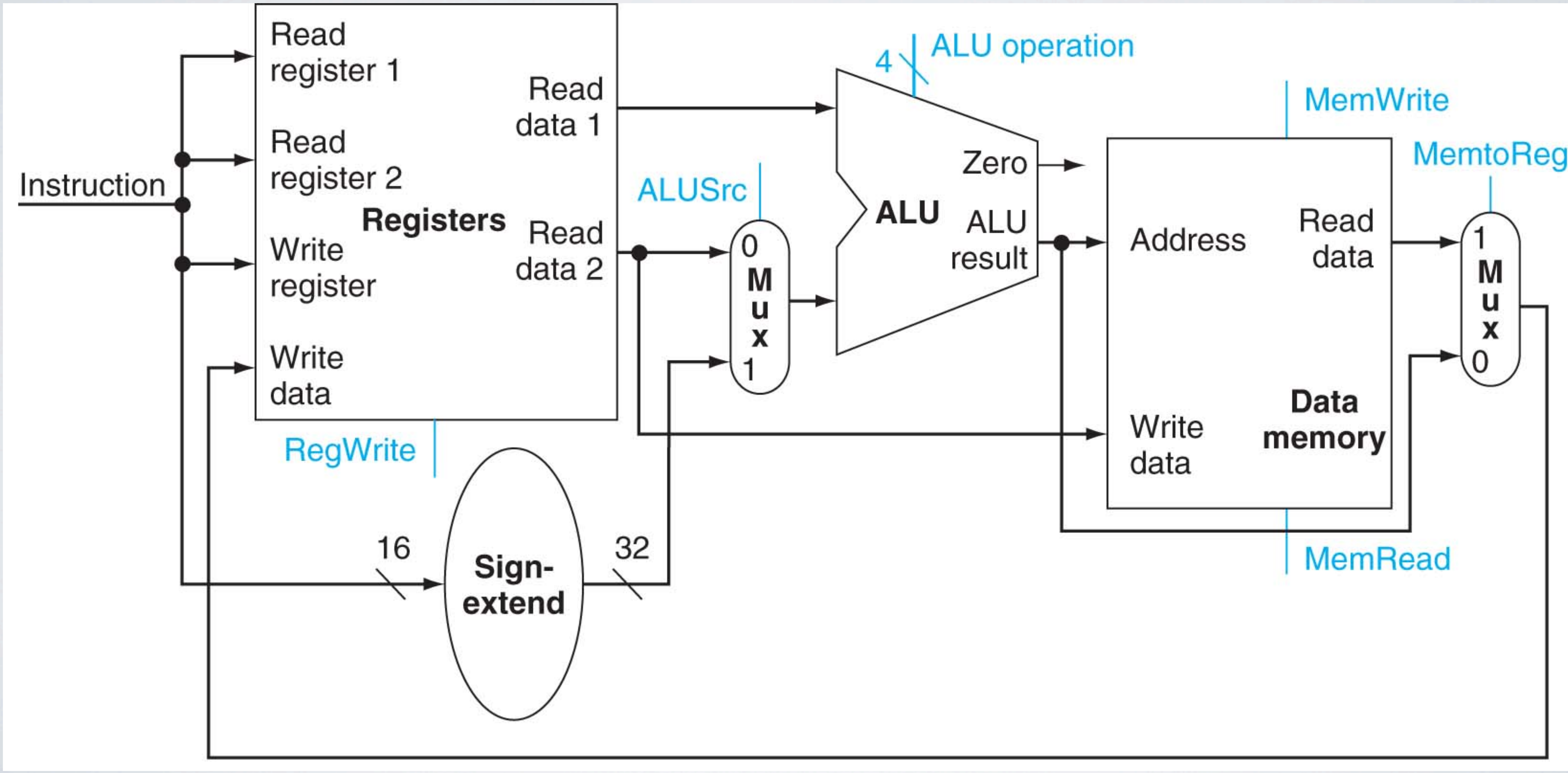
MIPS branching results in the subsequent instruction to be executed, regardless of the branch condition.

Why? Pipelining & branches (more on this later).

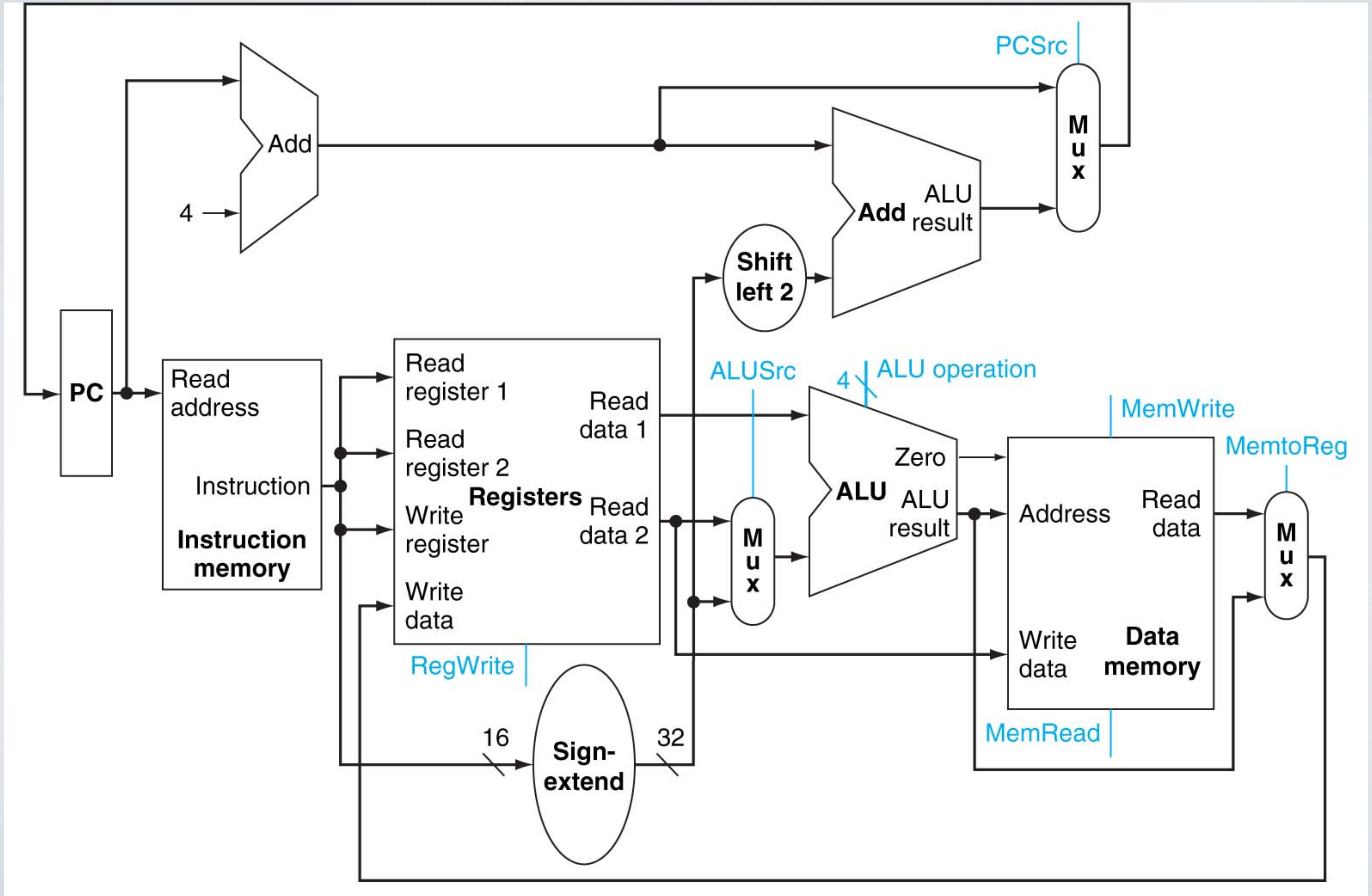
MIPS DATAPATH

- Try to execute all instructions in one clock cycle

MIPS DATAPATH (R-TYPE)



MIPS DATAPATH (GENERAL)



HOMEWORK

- Reading 26
- Finish Project 6, "Healthy" Beverages

no such
thing as
magic.