

ES 4: Finishing touches

Steven Bell

8 December 2021

By the end of class today, you should be able to:

- Explain how a processor implements memory and branch instructions
- Calmly talk about the next two weeks

Load and store

Flashback: to load data, we use another register as an address

LDR R0, [R1] ; Load the word at address R1 and put it into R0
Destination Address

LDR R0, [R1, 16] ; Load the word at address R1+16 into R0
Destination Address Offset

Load with an offset!?

LDR R0, [R1, 16] ; Load the word at address R1+16 into R0



Destination Address Offset

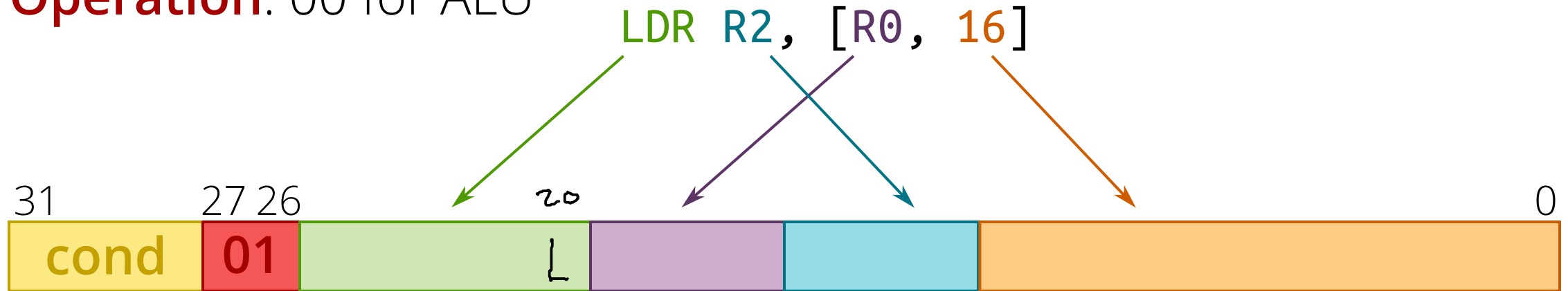
The diagram shows three arrows pointing from labels below to parts of the assembly instruction above. A teal arrow points from 'Destination' to 'R0'. A purple arrow points from 'Address' to 'R1'. An orange arrow points from 'Offset' to '16'.

Write some C code that would make use of this.

(Try it on godbolt.org!)

Memory operations (a simplification)

Operation: 00 for ALU

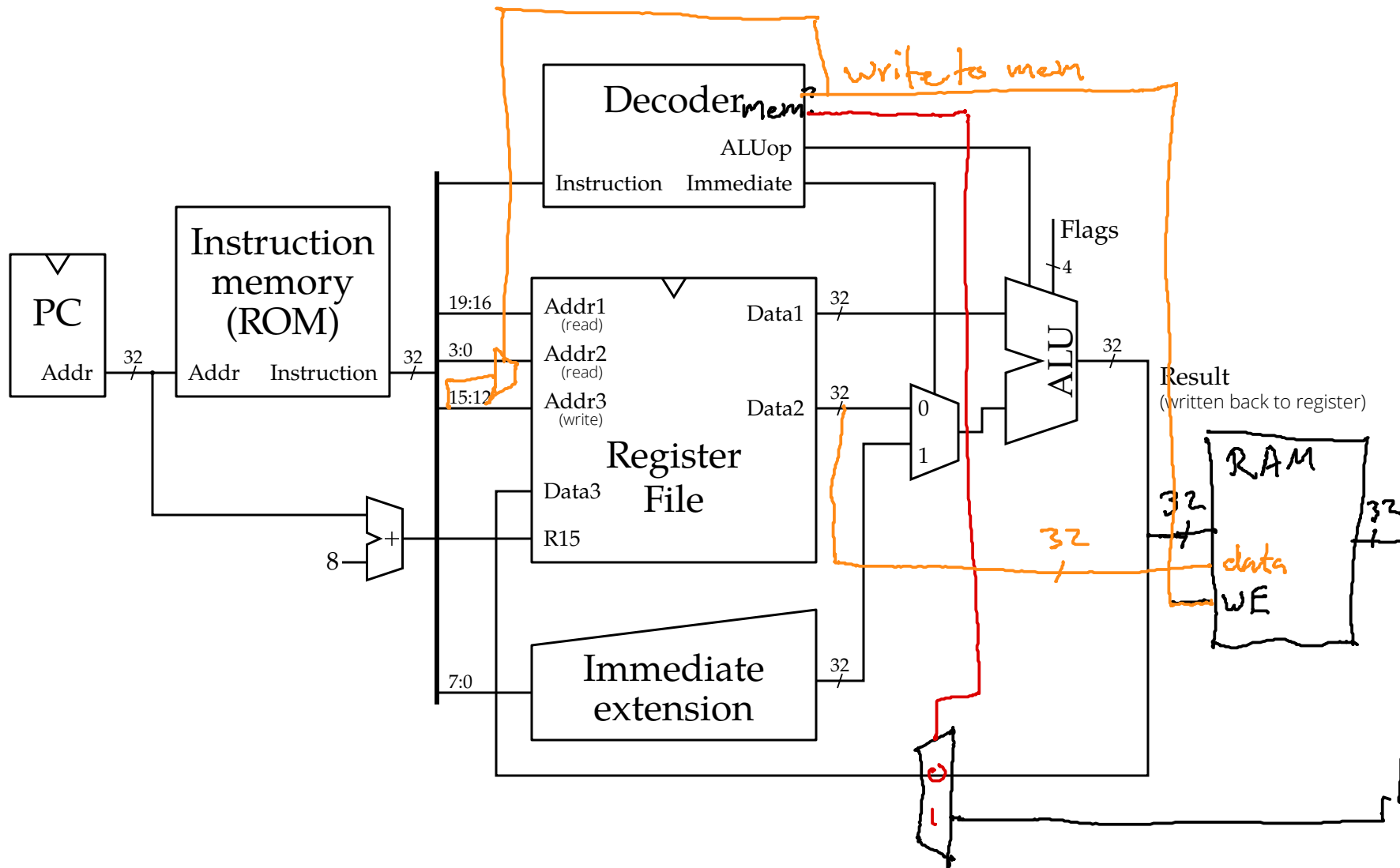


Command: what operation to perform (load/store, byte/word, offsets)

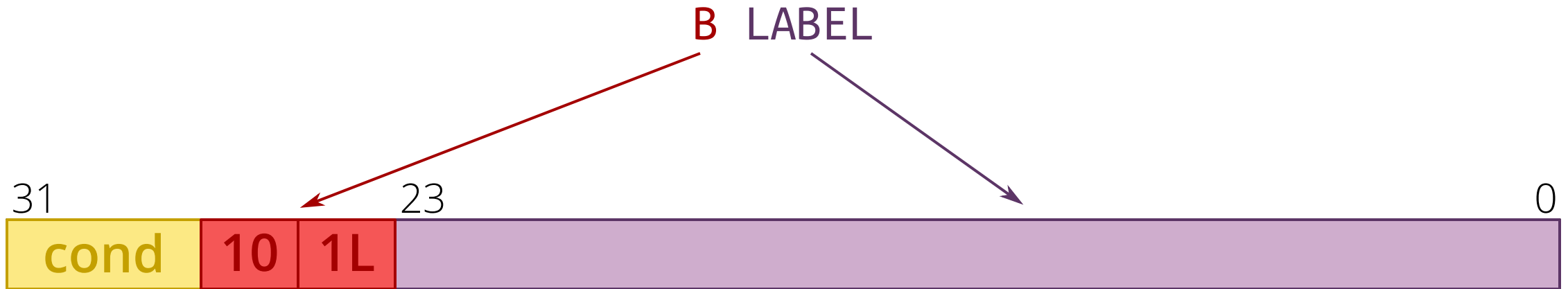
Destination (Rd): which register to put the result in

Address (Rn)

Offset: added to the address



Branch operations



Address (Rn): $(BTA - (PC+8)) / 4$

$(BTA - (PC+8)) / 4??$

Why subtract the PC?

Why PC+8 (vs just the PC)?

And why divide by 4?

The Acorn RISC Machine used a 5-stage pipeline

1) Read 0x00

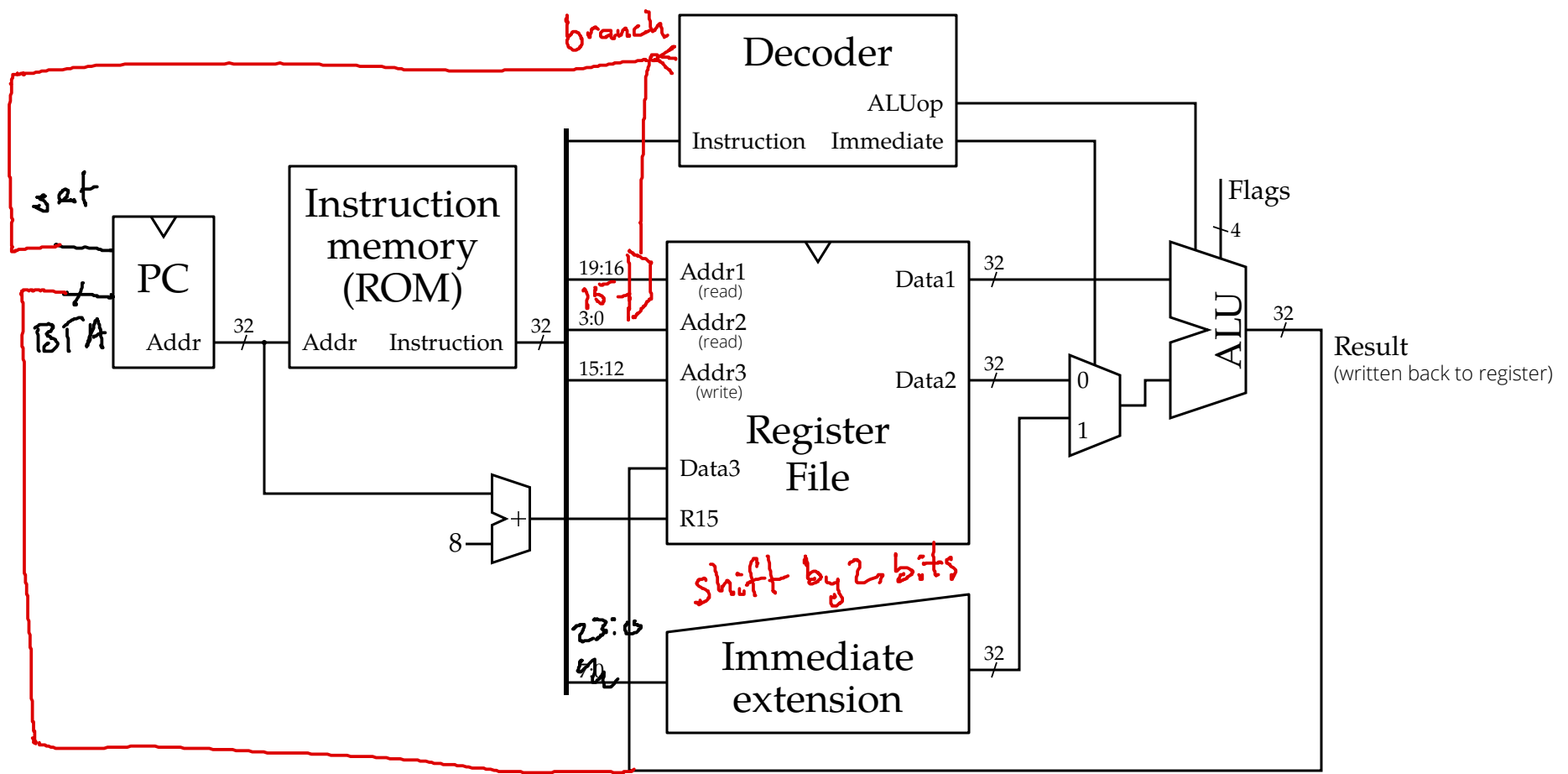
2) Read 0x04

Decode 0x00

3) Read 0x08

Decode 0x04

Execute 0x00



We made it!

Never stop improving!

How can I make this course more inviting/accessible beyond ECE?

Demo logistics

Set up starting at 11am in Halligan 102

Open to the public 12-1:15pm

Bring diagrams and pictures to explain your project

Spend some time with your project, some time wandering around

Final reports

Explain how your system works, at a level another ES4 student could pick up and follow.

Not a build journal, or a novel about your successes and failures.

Follow the handout for specific structure

Good reports are like a photo tour

ES 4

INTRODUCTION TO

~~DIGITAL LOGIC~~

RETRO VIDEO GAME
DESIGN



FINAL PROJECT

DEMO PARTY

MON. DEC 29

12-1:30PM

HALLIGAN 102

