

ES 4: From assembly to bits

Steven Bell

6 December 2021

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

- Explain what a "program counter" does in a processor
- Use a reference (like the textbook) to encode an ARM instruction with bits, or convert a string of bits into ARM instructions.
(you don't need to be good at this, that's what assemblers are for.)

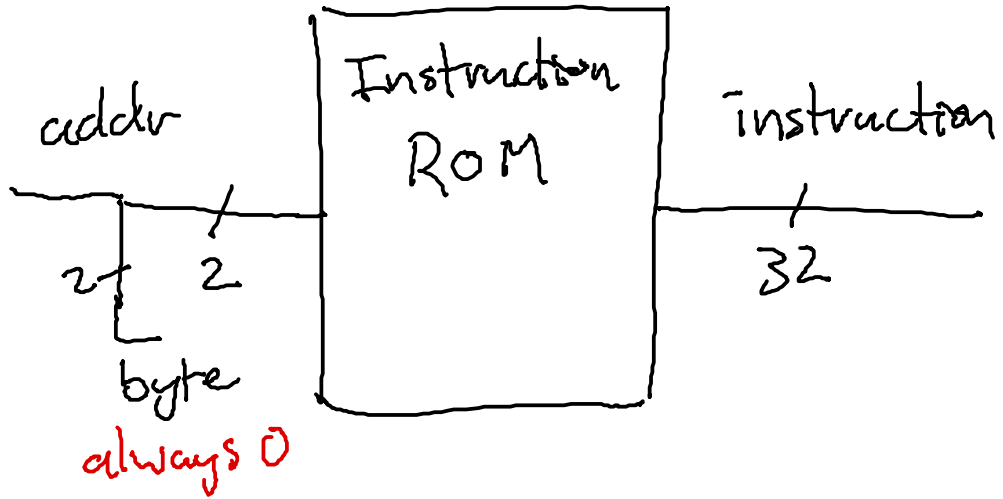
Let's build a computer!

We want to execute the following code:

```
0000  
0x00 MOV R0, #10 ; Put 10 in R0  
0100  
0x04 ADD R1, R1, R0 ; Add R0 to running total in R1  
1000  
0x08 SUB R0, R0, #1 ; Subtract 1 from R0
```

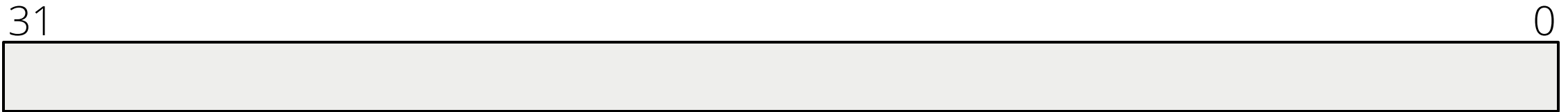
On Wednesday we'll deal with conditions, branching, and memory

Those instructions should go in memory

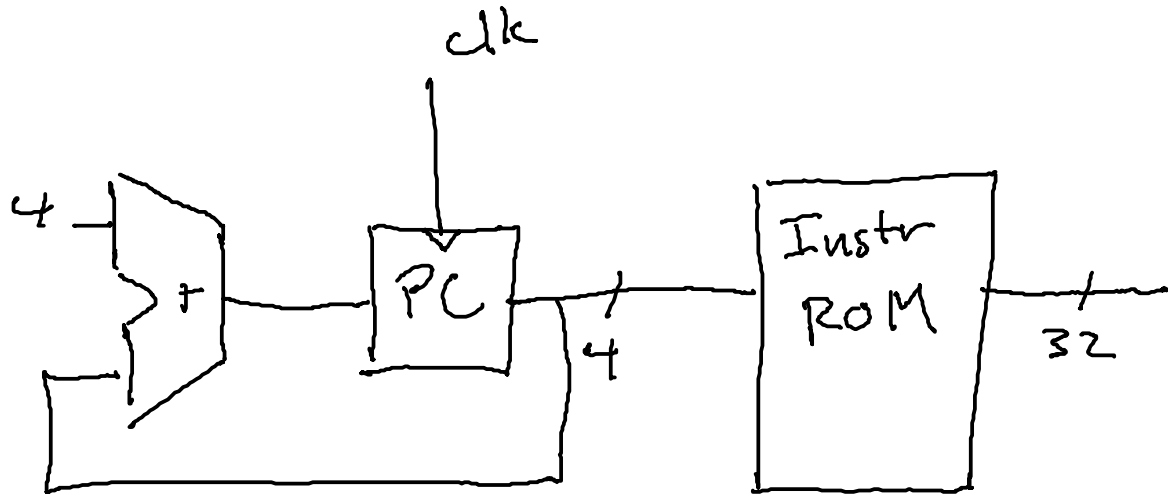


ARM instructions

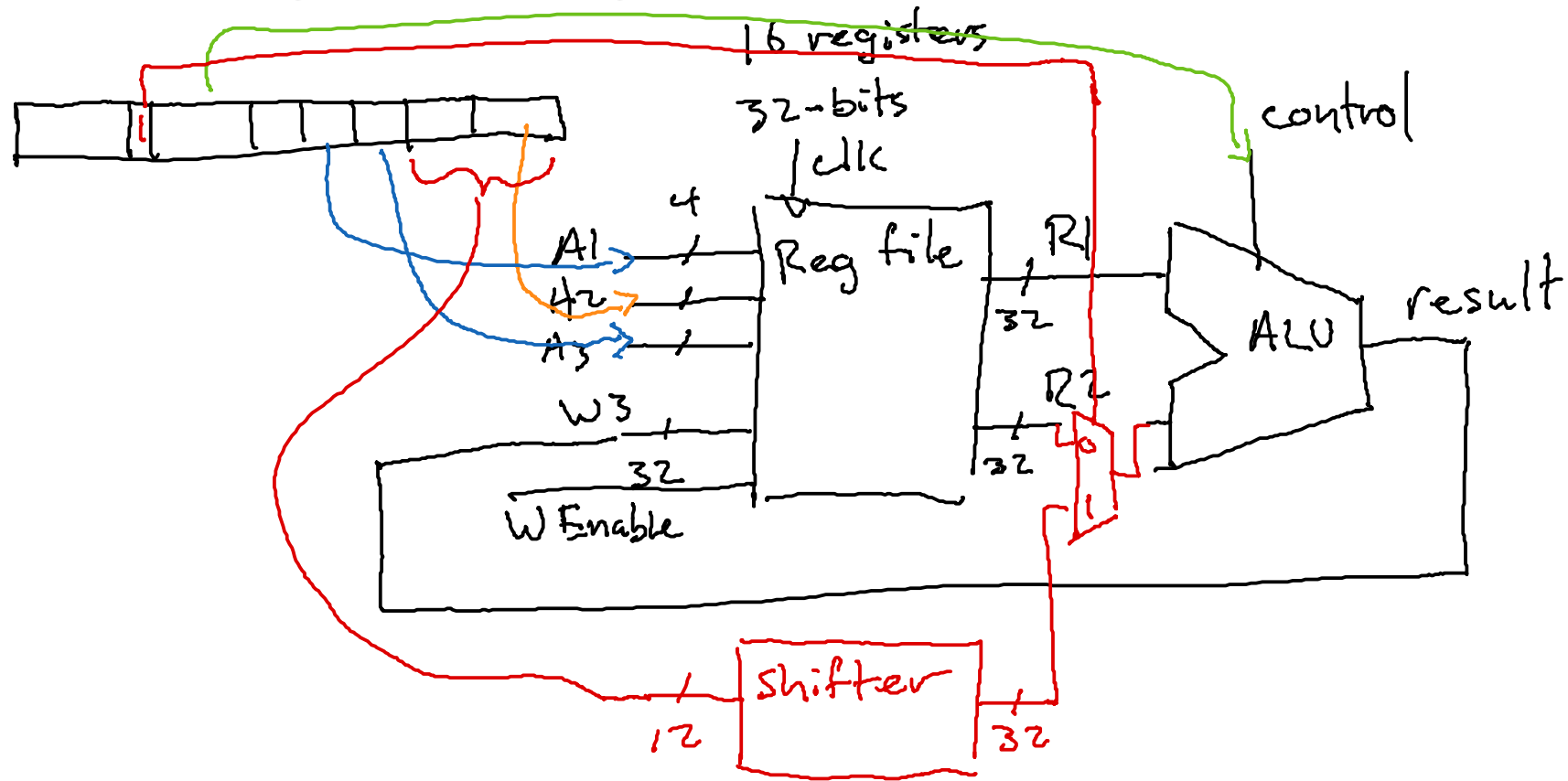
All instructions are 32 bits long (vs. x86)



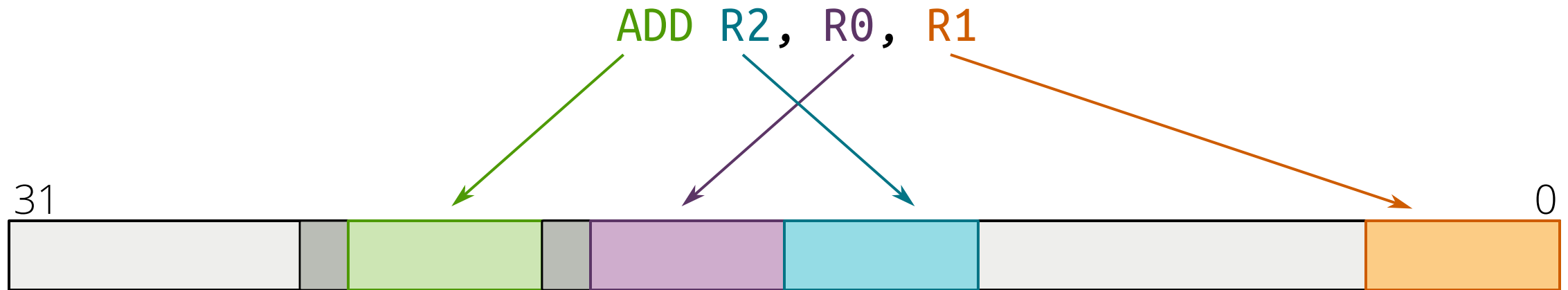
The program counter tracks the current address



We've got a register file, and an ALU



Data operations



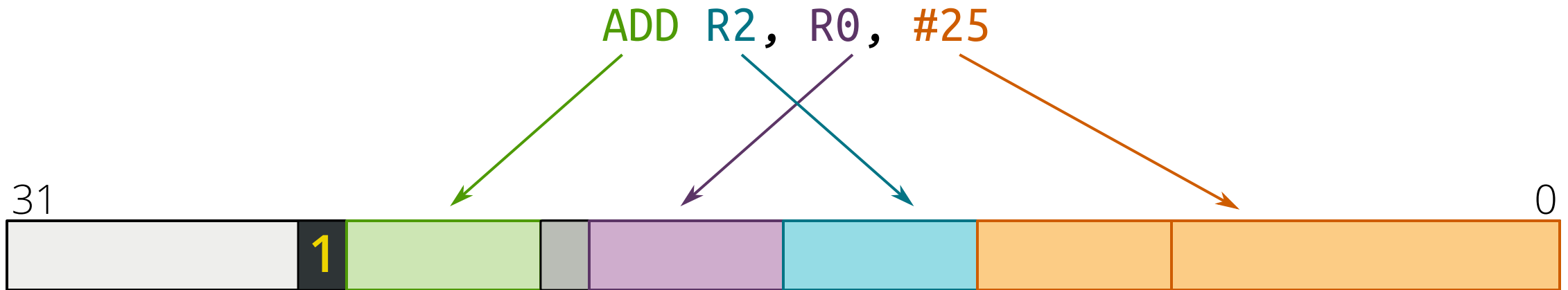
Command: what data operation to perform

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

First operand (Rn)

Second operand (Rm)

Data operations




Immediate: 1 when second operand is an immediate

Load and store

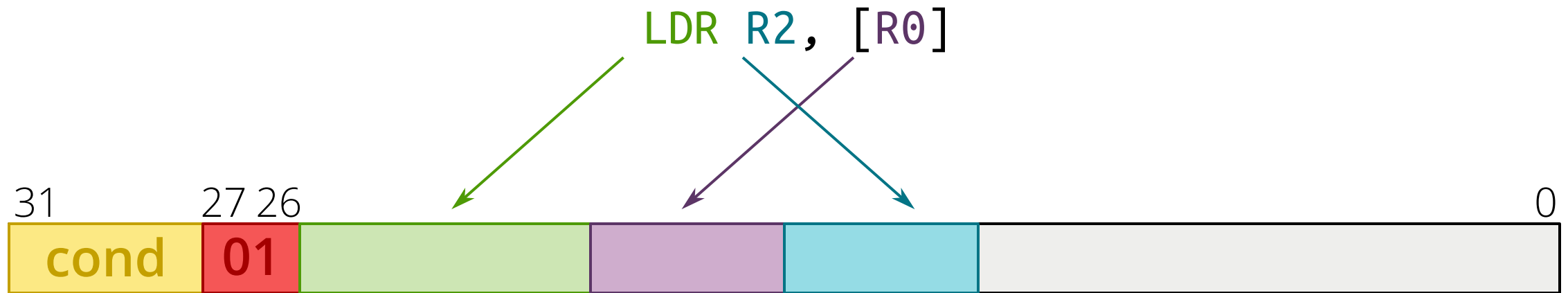
To load data, we use another register as an address

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

Destination Address



Memory operations (a simplification)

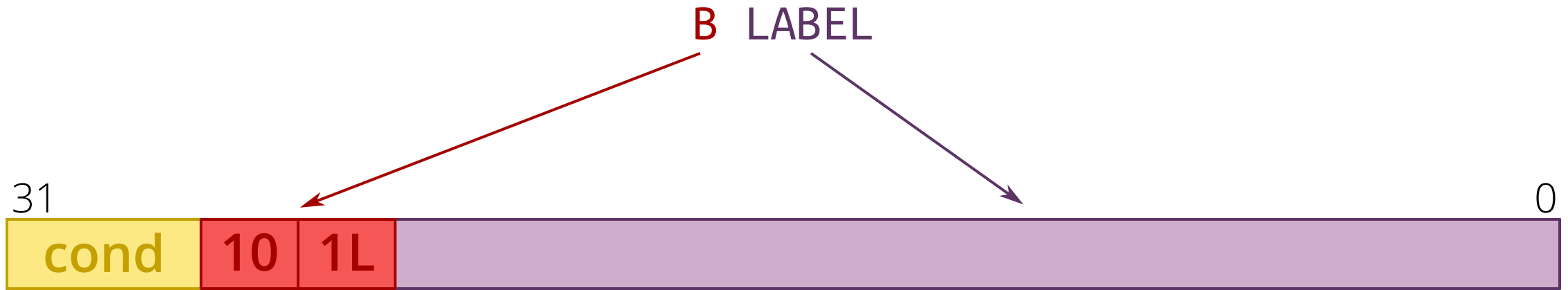


Command: what operation to perform (load/store, byte/word, offsets)

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

Address (Rn)

Branch operations (a simplification)



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

For Wednesday

1. Project, project, project!

No quiz, hooray!

For Wednesday

1. Project, project, project!

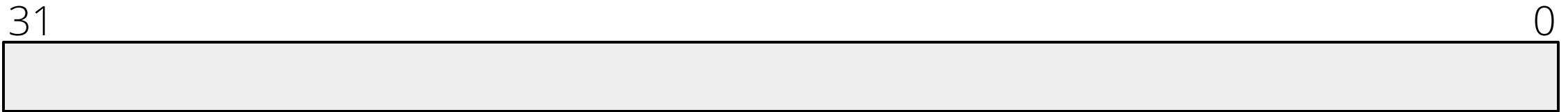
No quiz, hooray!



Condition: 4 bits specifying under what condition the instruction should execute

ARM instructions

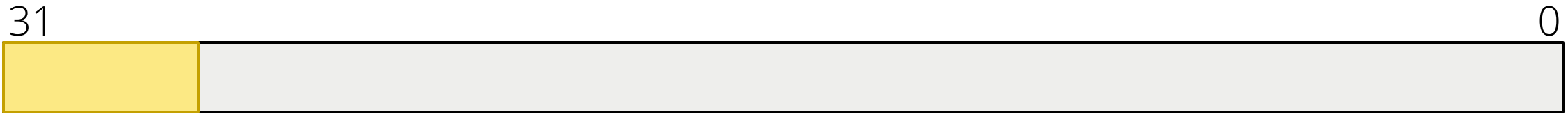
All instructions are 32 bits long (vs. x86)



Condition codes (table 6.3 in Harris)

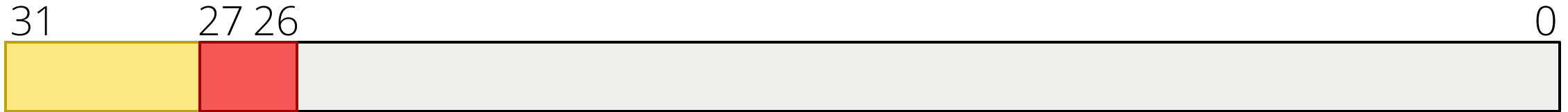
0000	EQ	Equal	Z
0001	NE	Not equal	\bar{Z}
0010	CS/HS	Carry set / Unsigned higher or same	C
0011	CC/LO	Carry clear / Unsigned lower	\bar{C}
0100	MI	Minus	N
0101	PL	Plus	\bar{N}

ARM instructions



Condition: 4 bits specifying under what condition the instruction should execute

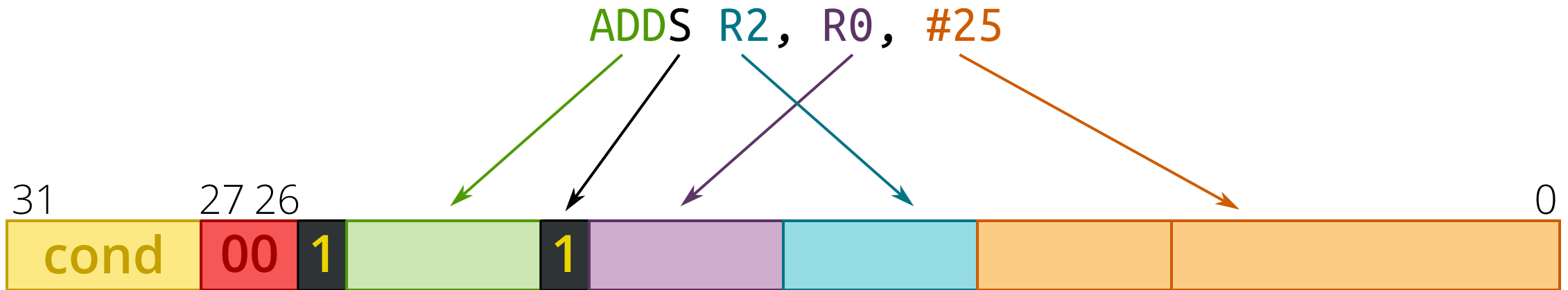
ARM instructions



Condition: 4 bits specifying under what condition the instruction should execute

Operation: what kind of instruction this is
Data operation, memory operation, branch, etc.

Data operations



Set CPSR: 1 when ALU flags should be saved