CPE 315
Computer Architecture

Number Representation and Intro. to MIPS Instruction Set

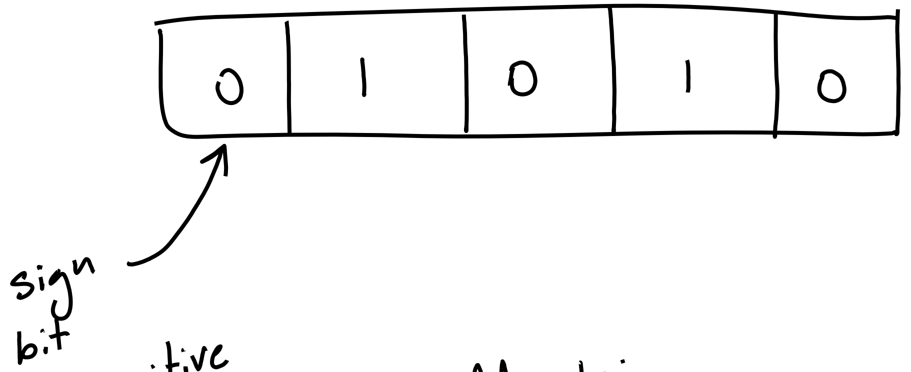
3.2 & 2.1-2.6

Number Representation

- Chapter 3.2

Two’s complement

10



0 = positive

1 = negative

Negate:

1. Flip bits
2. Add 1

$$\begin{array}{r} {}^{1} \\ 10101 \\ +\ 1 \\ \hline \boxed{10110} \end{array}$$

$\uparrow$

−10

Bitwise addition & subtraction

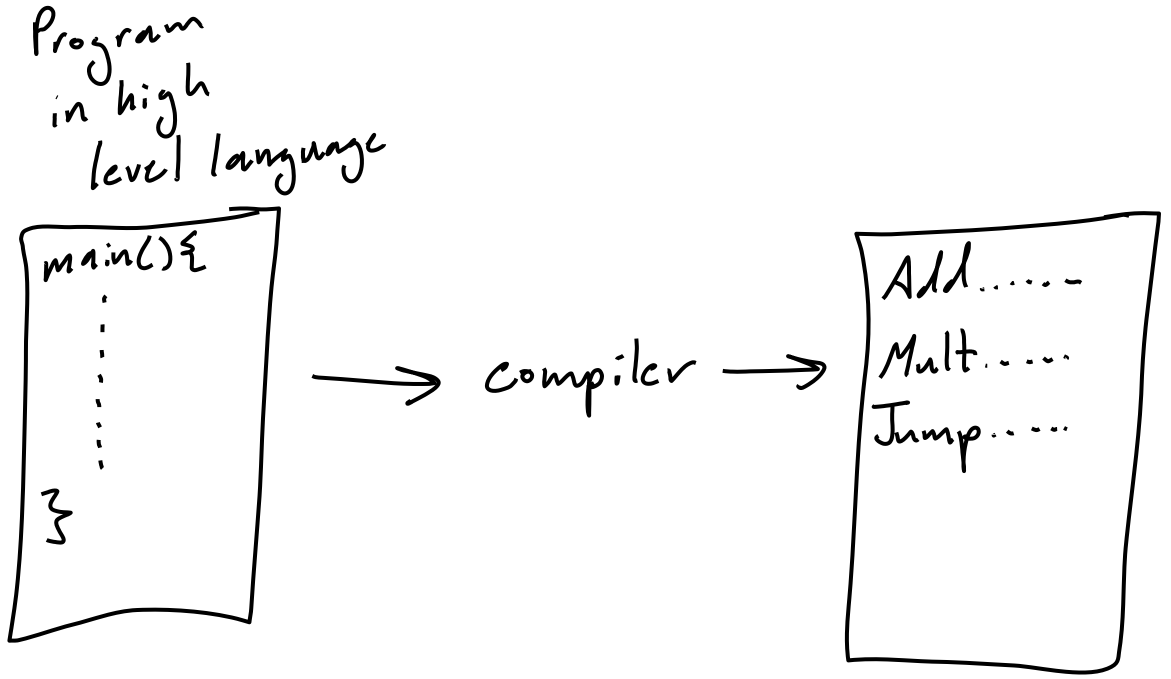
```
 1101
 0010
   +1
 ----
 0011
```

Binary	Signed	Unsigned	Binary	Signed	Unsigned	Binary	Signed	Unsigned
0011	+3	3	0110	+6	6	0110		
+0001	++1	+1	+1101	+-3	+13	+0101	+	+
0100	+4	4	1:0011	+3	19			

4 bit

Instruction Set Architecture (ISA)

- Definition: Instruction-Level Programmer interface to a CPU



HW/SW facts

CPU

2.0 GHz

3.0 GHz

- Processor = fast
- Memory = slow

800 MHz -1300 MHz

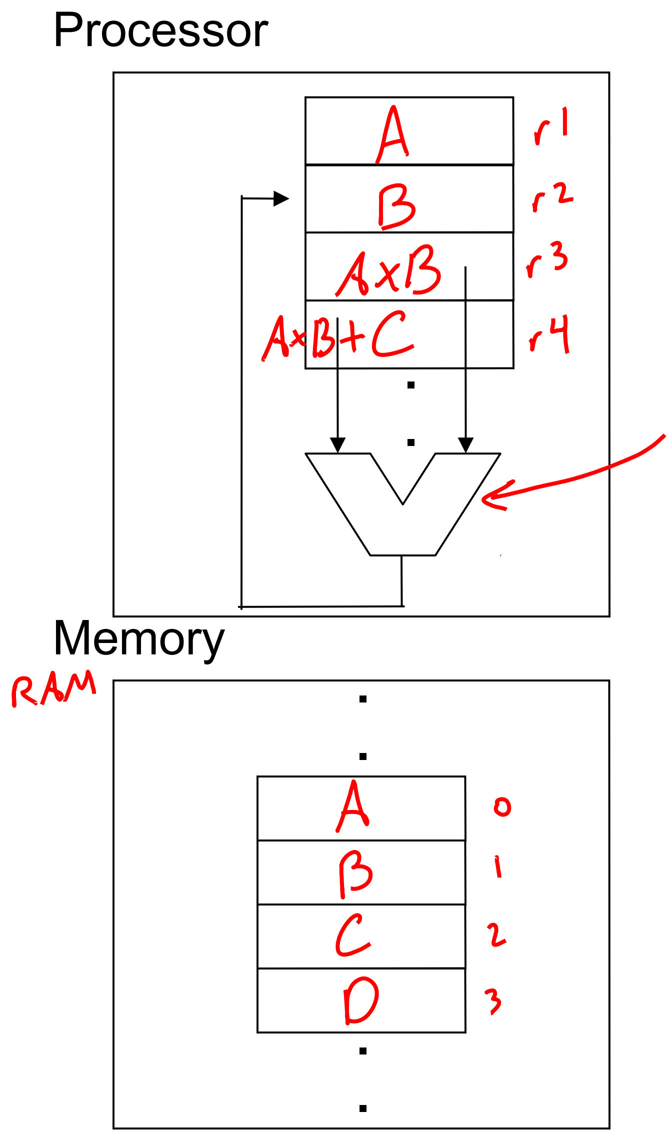
RAM

MIPS

- Load-Store architecture $\leftarrow$
  - Arithmetic operations only on registers
- Simple instructions (RISC) $\longleftarrow$ Reduced Inst. Set Computing
  - Every instruction is a fixed length
- 32 total registers

MIPS is a Load-Store Architecture

CPU



- ALU inputs: Two registers
- Memory access:

Load: copy of data from RAM to register

Store: copy of data from register to RAM

ALU = Arithmetic Logic Unit = does math/logic computation

D = A * B + C

dest. source

```
Load r1, A
Load r2, B
Mult r3, r1, r2
Load r4, C
Add r4, r3, r4
Store D, r4
```

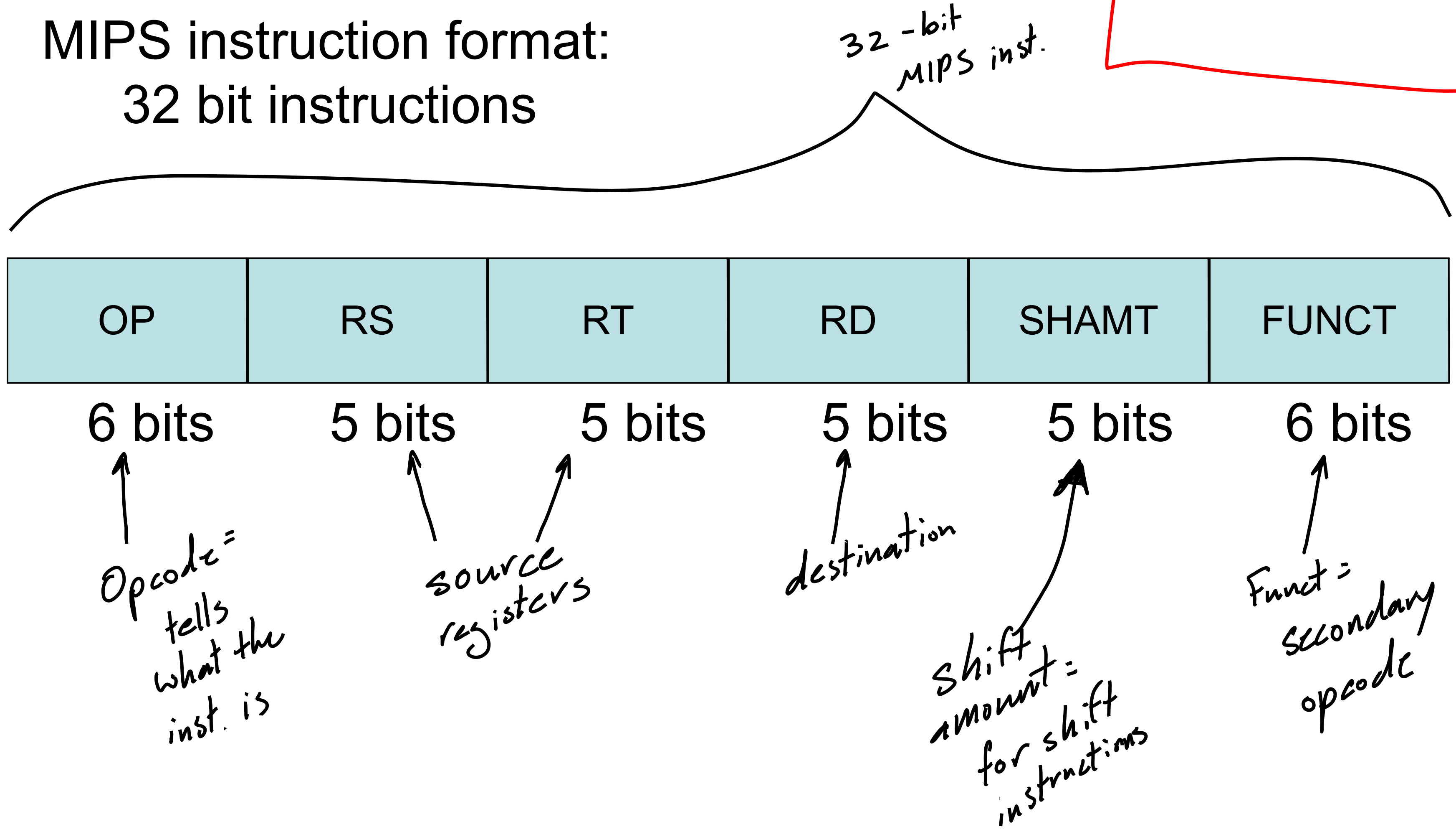
Reasons for Load/Store

- Registers are fast – Memory is slow

MIPS Instructions

- Lab 1 on PolyLearn

MIPS instruction format:
32 bit instructions



MIPS Registers – 32 registers

Name	Reg Number	Usage	Preserved across call?
$zero	0	The constant 0	Yes
$v0-$v1	2-3	Function results	No
$a0-$a3	4-7	Arguments	No
$t0-$t7	8-15	Temporaries	No
$s0-$s7	16-23	Saved	Yes
$t8-$t9	24-25	More temporaries	No
$gp	28	Global pointer	Yes
$sp	29	Stack pointer	Yes
$fp	30	Frame pointer	Yes
$ra	31	Return address	Yes

Pg.121

Register

R-format Instructions

slt = set less than

slt \$2, \$3, \$5

0 or 1

- Arithmetic instructions
  - Two input registers
  - One output register

Operation	rs	rt	rd	shamt	funct	meaning
add \$2,\$3,\$5 \$t2	3	5	2	0	32	\$2 = \$3 + \$5
sub \$2,\$3,\$5	3	5	2	0	34	\$2 = \$3 - \$5
addu \$2,\$3,\$5	3	5	2	0	33	\$2 = \$3 + \$5
slt \$2, \$3, \$5 0 or 1	3	5	2	0	42	if (\$3 < \$5) \$2 = 1 else \$2 = 0

MIPS Example 1

Translate $f = \underbrace{(g + h)}_{1} \underset{\uparrow 3}{-} \underbrace{(i + j)}_{2}$ into MIPS assembly

Assumption: g has been loaded into \$s0, h in \$s1, i in \$s3, and j in \$s4.
We want the result placed in \$v0.

```
1. add $t0, $s0, $s1
2. add $t1, $s3, $s4
3. sub $v0, $t0, $t1
```

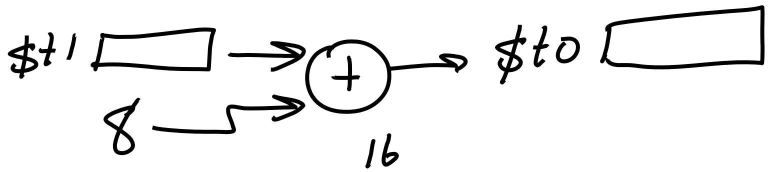
I-format Instructions

Immediate (constant)

- Operations involving register and constant

slti $2, $3, 7

addi $t0, $t1, 8



16

Operation	rs	rt	rd 5	shamt 5	funct 6	meaning
addi $2, $3, 8	3	2	8			$2 = $3 + 8
andi $2, $3, 10	3	2	10			$2 = $3 & 10

slti $2, $3, 7	3	2	7			if ($3 < 7) then $2=1
						else $2=0