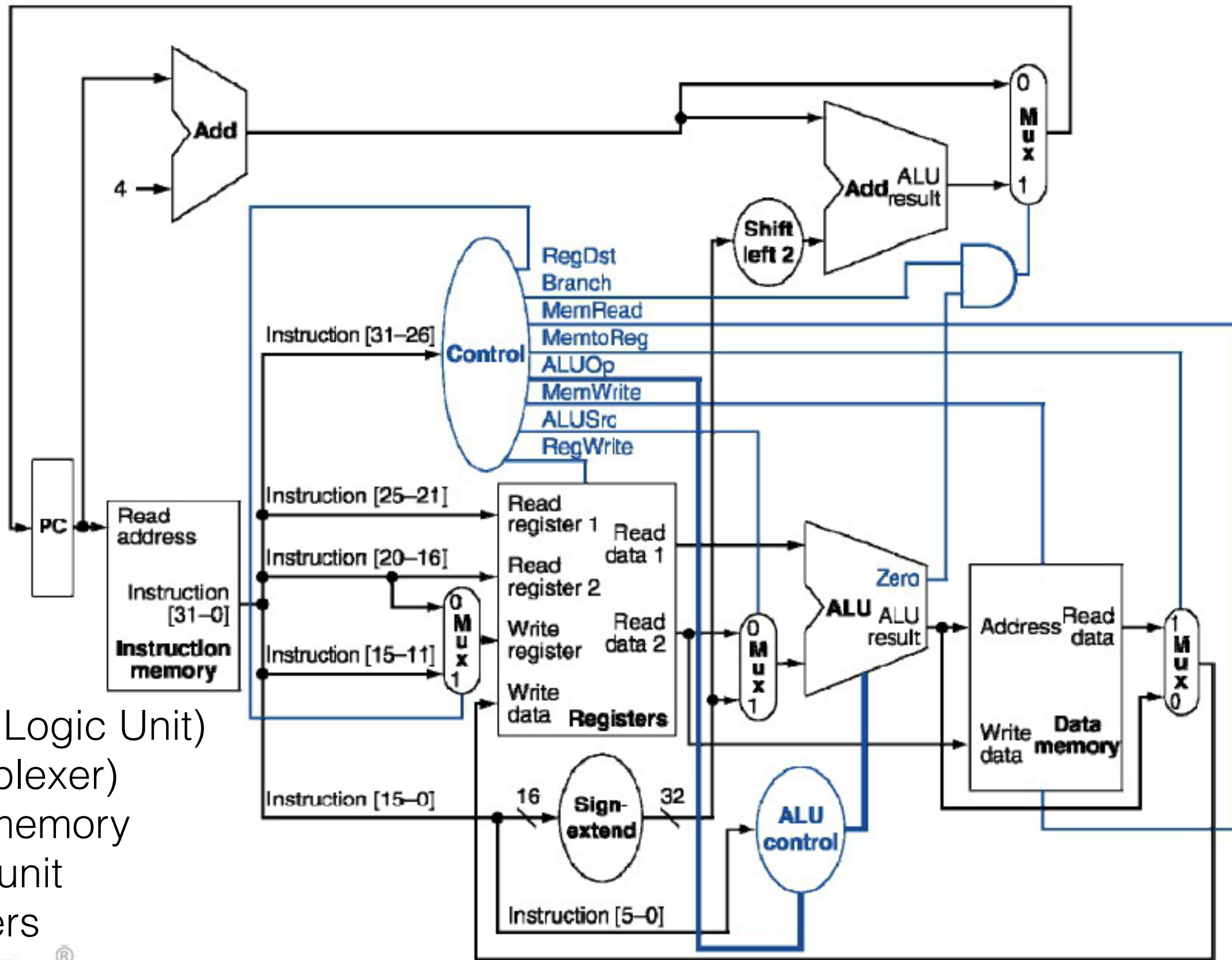


EE116C/CS151B

Fall 2017

Instructor: Professor Lei He
TA: Yuan Liang



ALU (Arithmetic Logic Unit)
 MUX (multiplexer)
 Instruction memory
 Control unit
 Registers
 Clock

....

Review

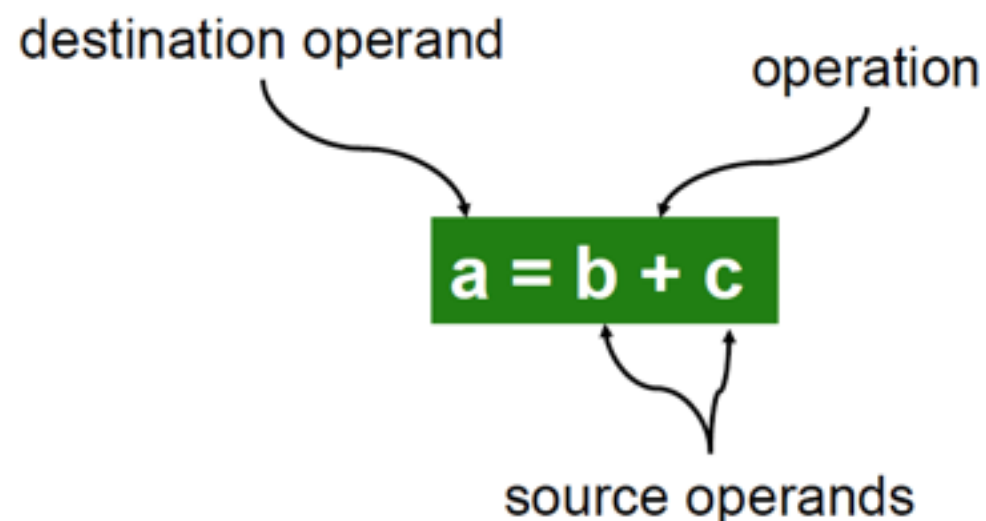
Definition of Instruction set architecture (ISA, or IS, or Instruction set) from Wiki:

An instruction set architecture (ISA) is a group of commands for a CPU in machine language.

It is an abstract model of a computer.

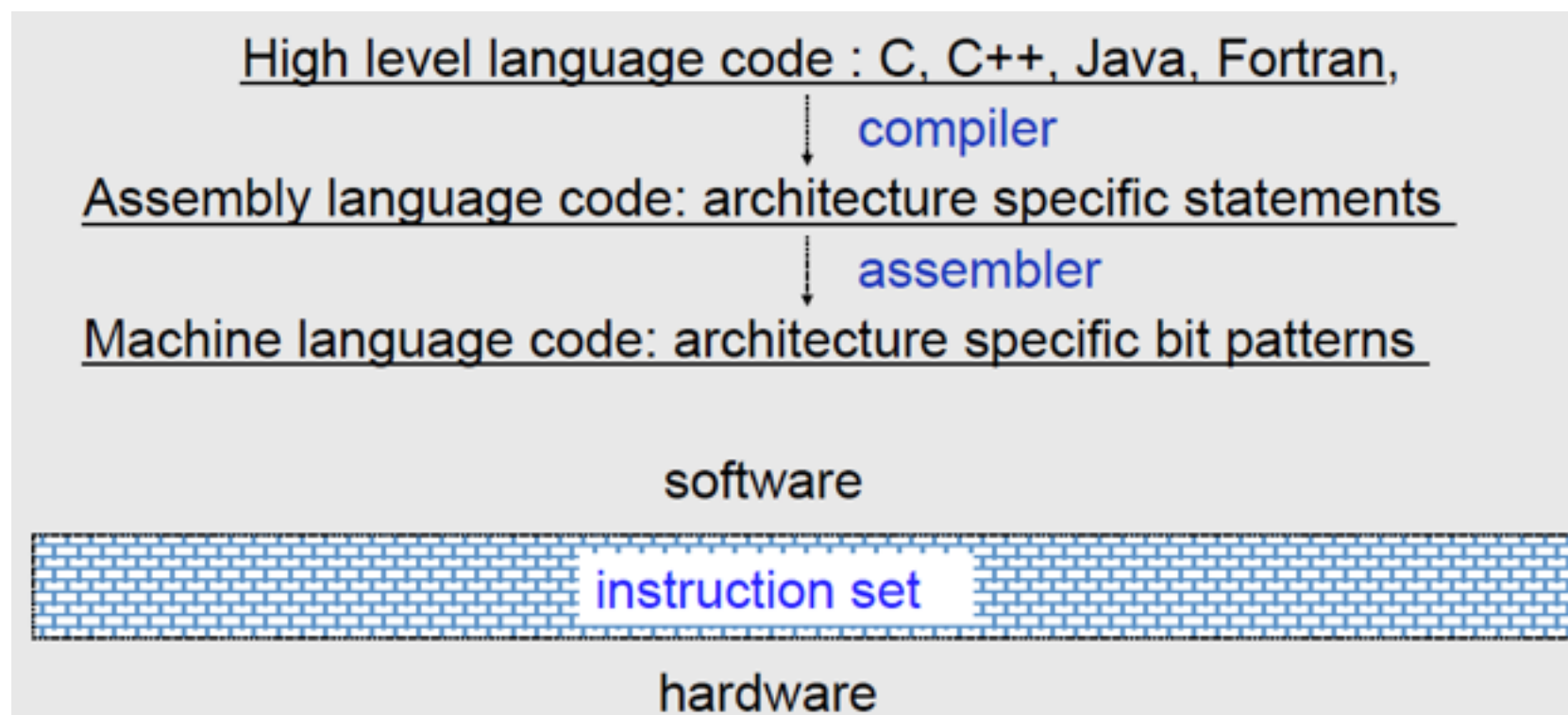
Different computers have different instruction sets.

Format:



Review

Serves as an **interface** between software and hardware



Review

Types of ISA:

CISC (Complex Instruction Set Computers)

large # of instructions

Advantage:

- many specialized instructions

- optimized at the instructions level

Disadvantage:

- various CPI

Intel's x86 (1978)

RISC (Reduced Instruction Set Computers)

relatively fewer instructions

Advantage:

- same CPI -> enable pipelining and parallelism

Disadvantage:

- more instructions.

- more registers for intermediate results.

MIPS

Review

Registers in MIPS:

What is register:

Inside processor, use for frequently accessed data and fast calculation.

How many registers in MIPS:

32

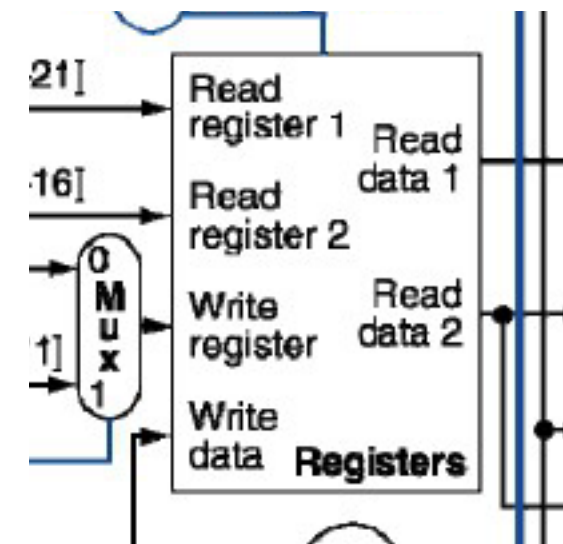
How large are them:

32bit

How they work:

Load values from memory into registers

Store result from register to memory



Review

Registers in MIPS:

C code:

```
f = (g + h) - (i + j);
```

Compiled MIPS code:

```
add $t0, $s1, $s2
```

```
add $t1, $s3, $s4
```

```
sub $s0, $t0, $t1
```

Review

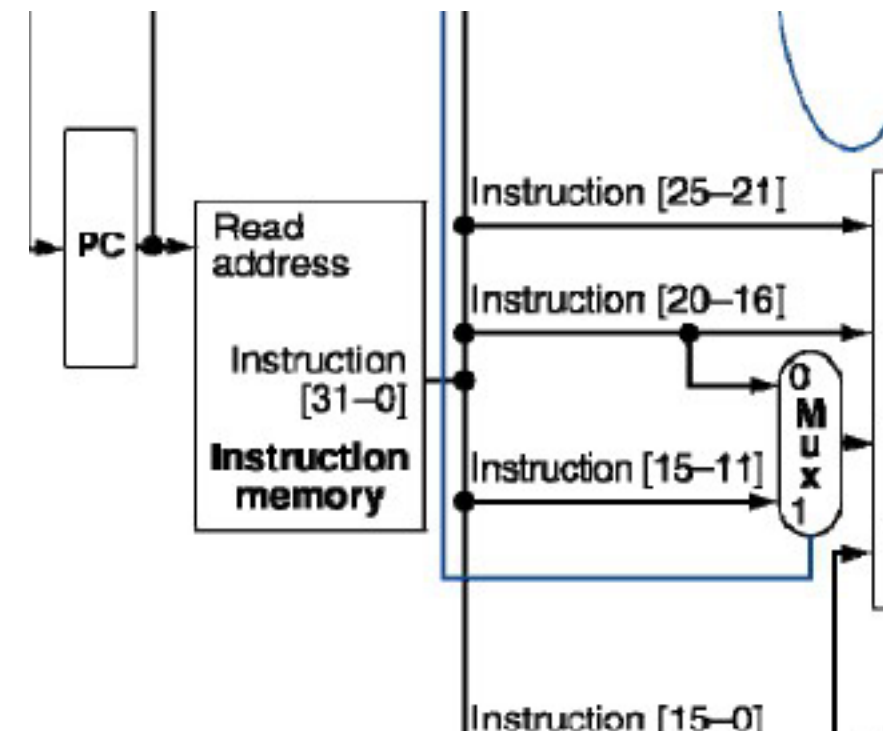
Instructions in MIPS

Instructions formats are hard encoded in binary

For MIPS instructions

Encoded as 32-bit instruction words

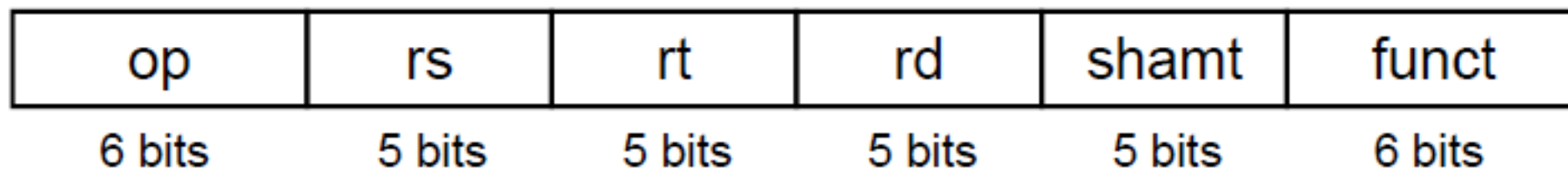
Small number of formats encoding operation code (opcode), register numbers, ...



Review

Instructions in MIPS

R-format Instructions



op: operation code (opcode)

rs: first source register number

rt: second source register number

rd: destination register number

shamt: shift amount (00000 for now)

funct: function code (extends opcode)

add \$t0, \$s1, \$s2

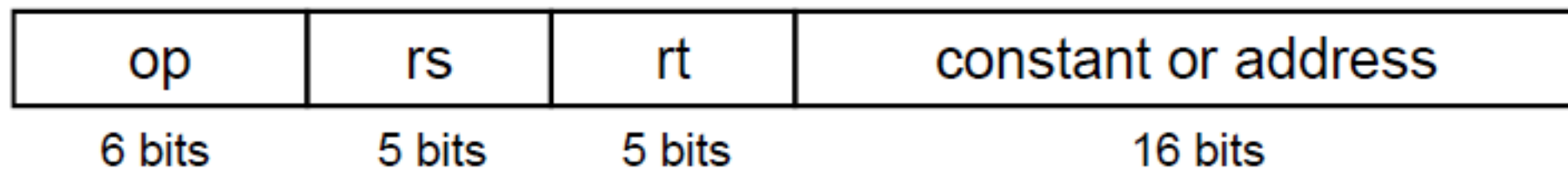


00000010001100100100000000100000₂

Review

Instructions in MIPS

I-format Instructions



op: operation code (opcode)

rs: first source register number

rt: destination or source register number

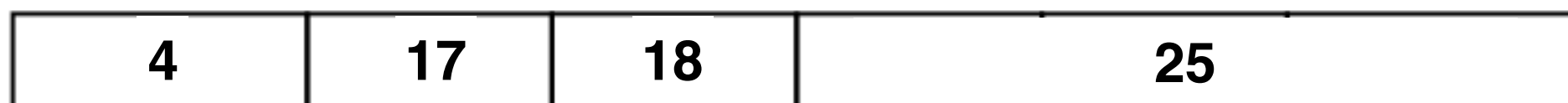
Constant: -2^{15} to $+2^{15} - 1$

Address: offset added to base address in rs

lw \$s1, 100(\$s2)



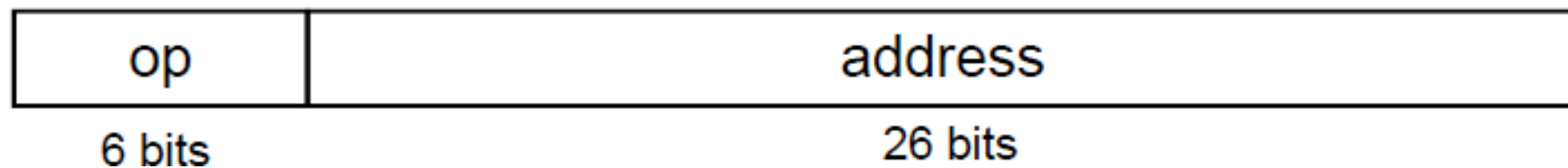
beq \$s1, \$s2, 100



Review

Instructions in MIPS

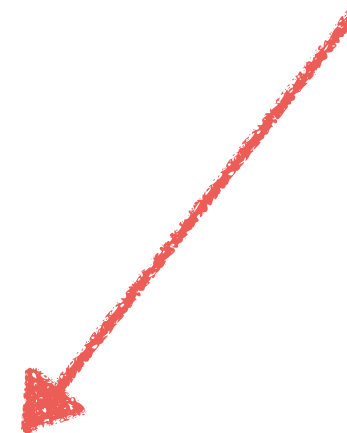
J-format Instructions



op: operation code (opcode)

Address: absolute address

j 10000



Review

Instructions in MIPS

How does processor know what kind of instructions it is?

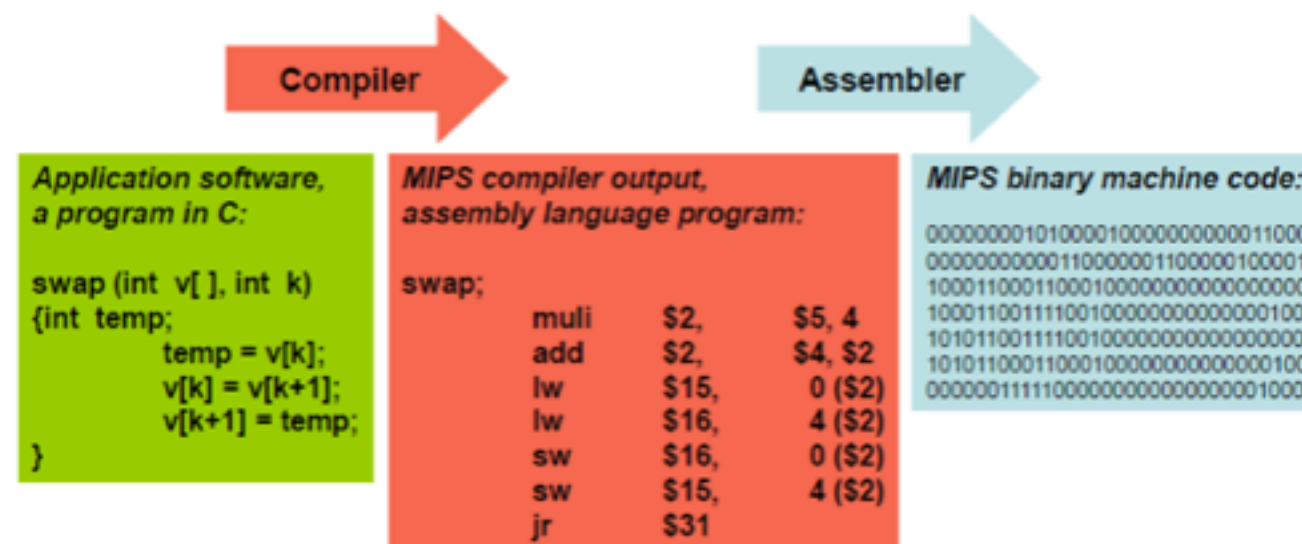
And how to parse it.

Review

- arithmetic
 - add, subtract, multiply, divide, ...
- logical
 - and, or, shift left, shift right, ...
- data transfer
 - load word, store word
- conditional branch
 - beq & bne are PC-relative, since most targets are nearby
- unconditional jump
 - jump, jump register, branch and link

Review

The big picture:



What is ISA
ISA style
ISA for MIPS

ISA in processor