

Instruction Set

RISC - reduced instruction set computer

CISC - complex instr. set computer

RISC concepts

- every instr. same size
1 instr = 1 word
- simple instr.
- memory accessed by load/store instr. only.



$$a = b + c$$

add a, b, c
↑ ↑ ↑
instr result operands

$$a = (b + c) - d$$

add t0, b, c
sub a, t0, d

MIPS - 32 registers

saved registers

\$s0... \$s7

reg 16-23

temp registers

\$t0... \$t7

8-15

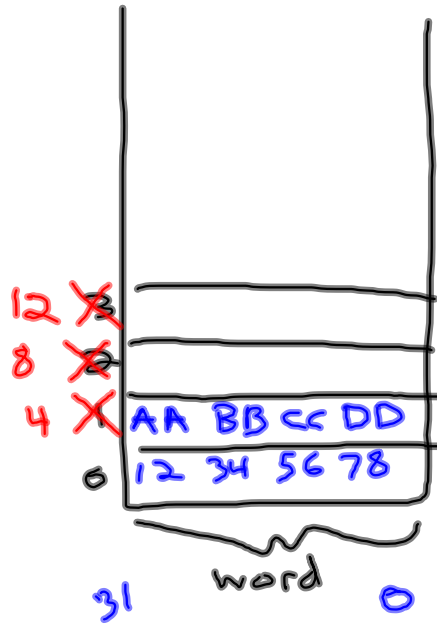
add \$t0, \$s1, \$s2
sub \$s0, \$t0, \$s3

Memory



Byte

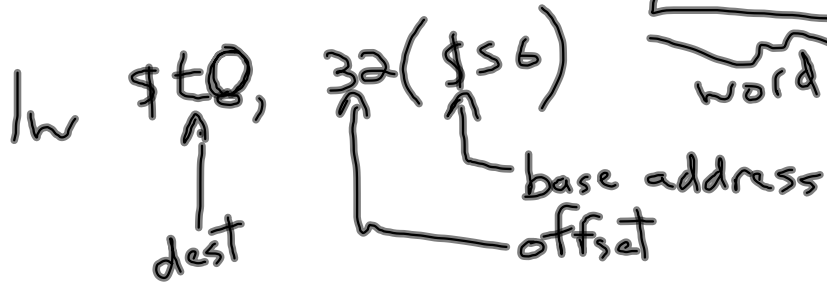
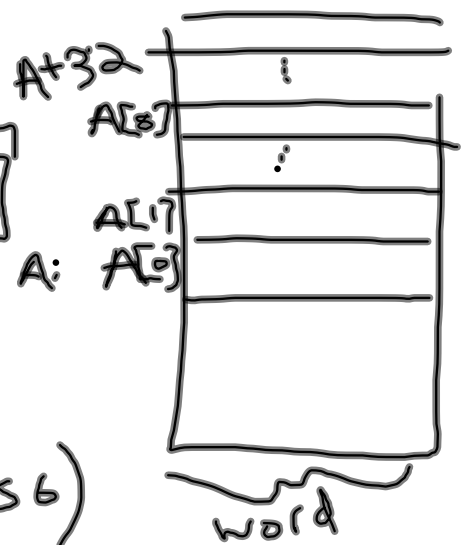
Byte-addressable



load/store

lw - load word

$$\begin{array}{c} \text{word} \\ \uparrow \\ \$s3 \end{array} = \begin{array}{c} h \\ \uparrow \\ \$s4 \end{array} + \begin{array}{c} \boxed{A[8]} \\ \uparrow \\ \$s6 \end{array}$$




add \$s3, \$s4, \$t0

$x = 4;$

Immediate Operands

- store value of operand in instruction

$x = y + 4$


addi \$S4, \$S5, 4

$x = 4$

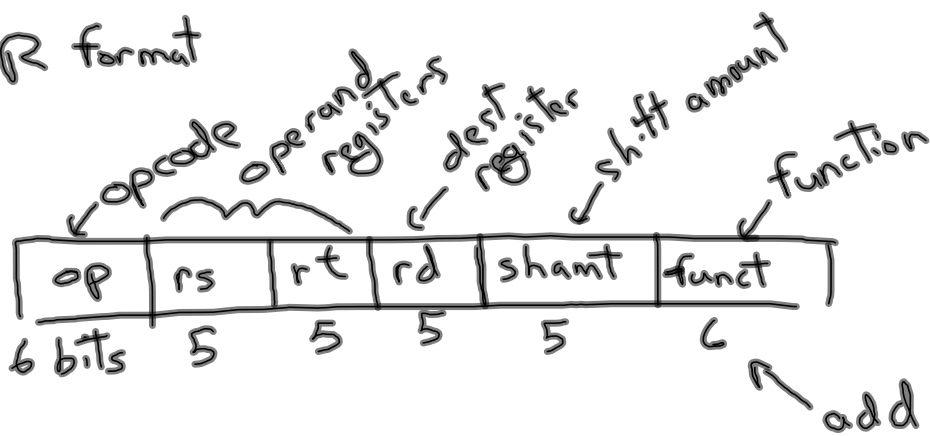
addi \$S4, \$zero, 4

$x = y$

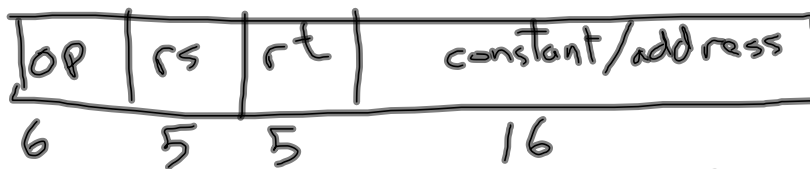
add \$S4, \$S5, \$zero

Instr. Format

R format



I format



	format	op	rs	rt	rd	shamt	func	addr
add	R	0				0	32	X
sub	R	0				0	34	X
addi	I	8			X	X	X	const.
lw	I	35			X	X	X	addr
sw	I	43			X	X	X	addr

Fetch - get the next instr
from memory

Decode - determine operands

Execute - run the instr.

Store - save the result