

# MIPS Addressing Modes

## Accessing Data in Memory (1)

MIPS is a load-store architecture

- Only load and store instructions access memory.
- Computation instructions operate only on values in registers.

The bare machine provides only one memory-addressing mode: **c(rx)**

- Uses the sum of the immediate *c* and register *rx* as the address.

# MIPS Addressing Modes

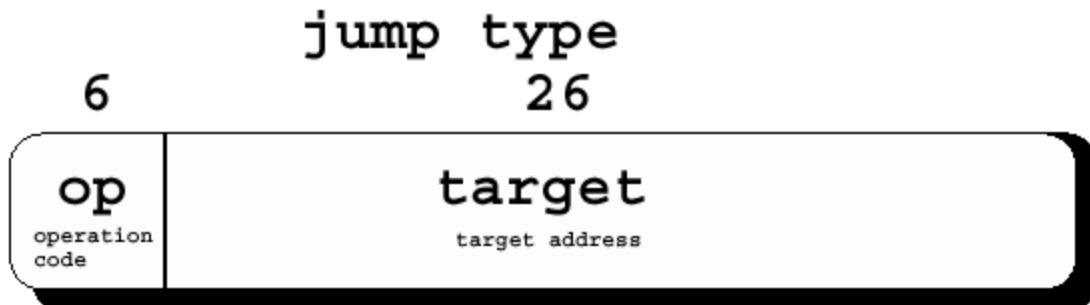
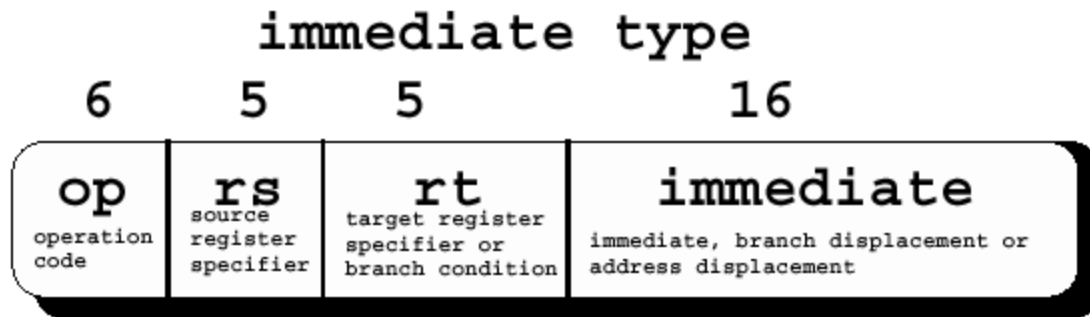
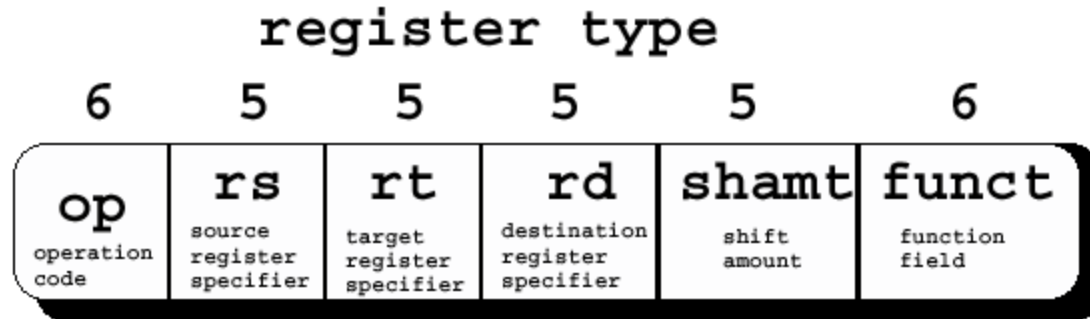
## Accessing Data in Memory (2)

The **virtual machine** provides the following addressing modes for load and store instructions:

Format	Address computation
(register)	contents of register
imm	immediate
imm (register)	immediate + contents of register
label	address of label
label $\pm$ imm	address of label + or – immediate
label $\pm$ imm (register)	address of label + or – (immediate + contents of register)

This virtual computer appears to have non delayed branches and loads and a richer instruction set than the actual hardware

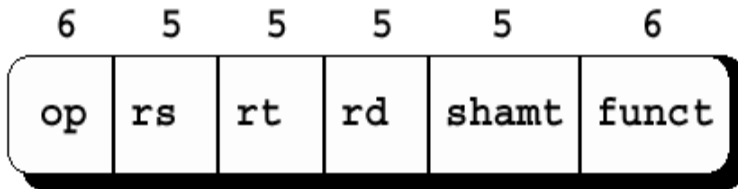
# MIPS Instruction Format



# MIPS Addressing Modes

## Accessing Instructions and Data (1)

### Register Addressing

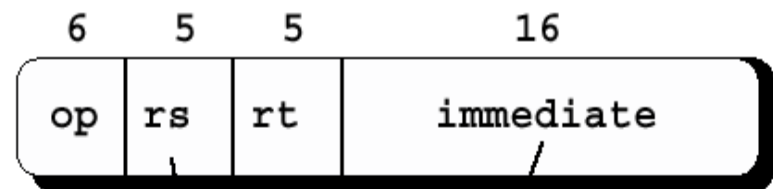


000000 01001 01010 01000 00000 100000

add \$t0,\$t1,\$t2

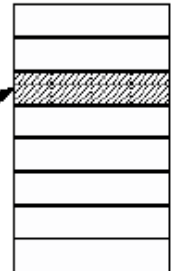
function for  
add is 32

### Base Addressing



register

+



100000 01010 01000 0000000000000000

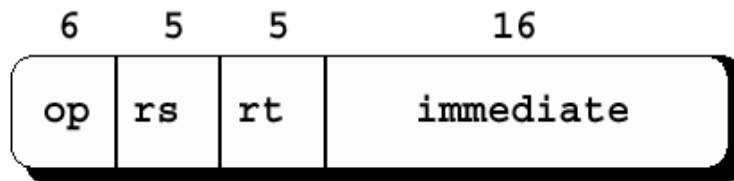
lb \$t0,(\$t2)

opcode for  
lb is 32

# MIPS Addressing Modes

## Accessing Instructions and Data (2)

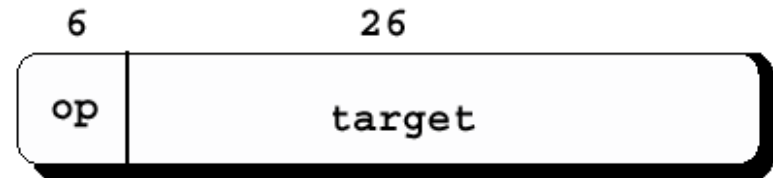
### Immediate Addressing



001000 01001 01001 0000000000000001

opcode for  
addi is 8

addi \$t1,\$t1,1



000010 0000010000000000000000001111

opcode for  
j is 2

j nextCh

destination  
0x4003c. add  
2 zeros since  
instructions  
word-aligned

# MIPS Addressing Modes

## Accessing Instructions and Data (3)

### PC-Relative Addressing

