Machine Instruction Format

All MIPS machine instructions are 32 bits

- Bits are number 0 to 31 from right to left

Three formats are used:

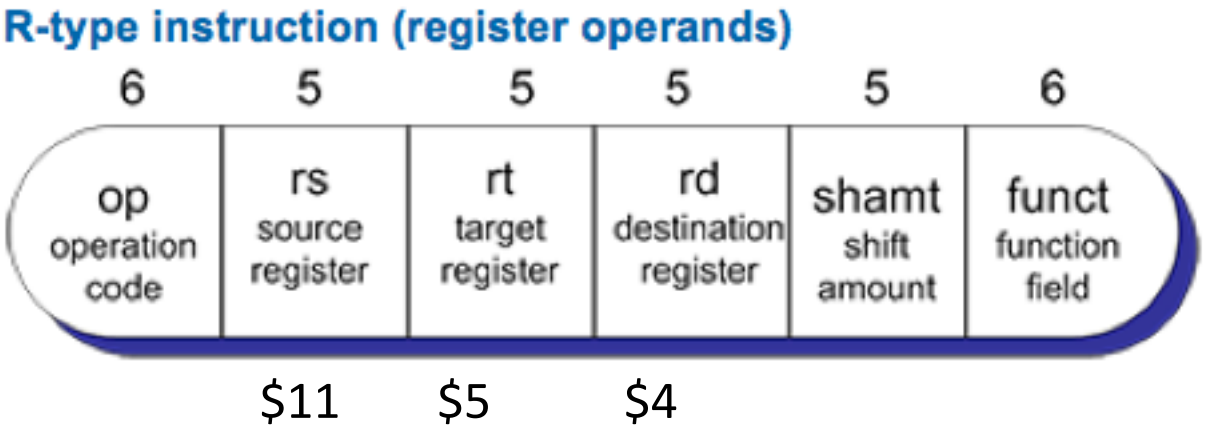
- R-type
  - Arithmetic and logical instructions
- I-Type
  - Instructions using an immediate operand
  - Same format is used for load word (lw) and store word (sw)
- J-type
  - Contains part of jump address in bits 0 through 25
  - Used by jump (j) instruction and jal (to call functions)

Each has a 6-bit opcode in bit positions 26 through 31

R-type Machine Instruction Format

R-type uses:

- 3 register operands
- opcode is always 0
- function is specified by rightmost 6 bits
- shamt field is used with shift instructions

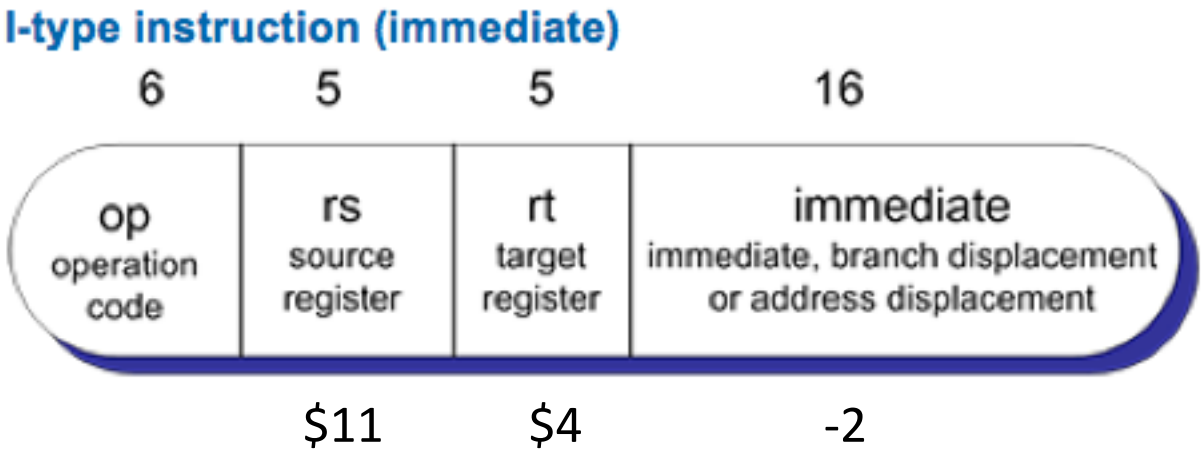


Example: add $4,$11,$5

I-type Machine Instruction Format

I-type uses:

- immediate operand in rightmost 16 bits
- treated as signed operand by arithmetic instructions
- treated as unsigned operand by logical instructions
- each has a unique opcode
- this format is used by lw and sw as well

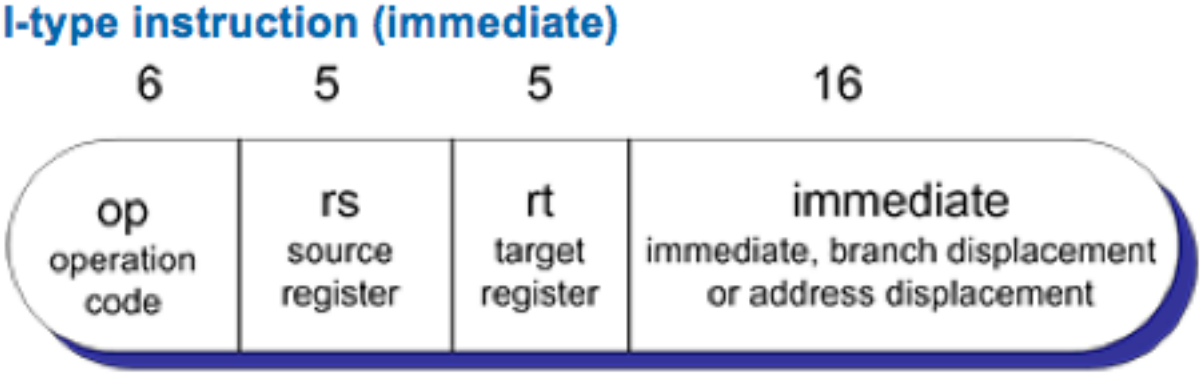


Example: addi $4,$11,-2

I-type Machine Instruction Format

Branch instructions also use the I-type format:

- rightmost 16 bits = # of instructions to branch
- Negative value means branch backwards
- Positive value means branch forward
- Sign-extended to 32 bits & shifted left 2 bits
- then added to the PC to produce branch address
- PC points to location following the branch instruction

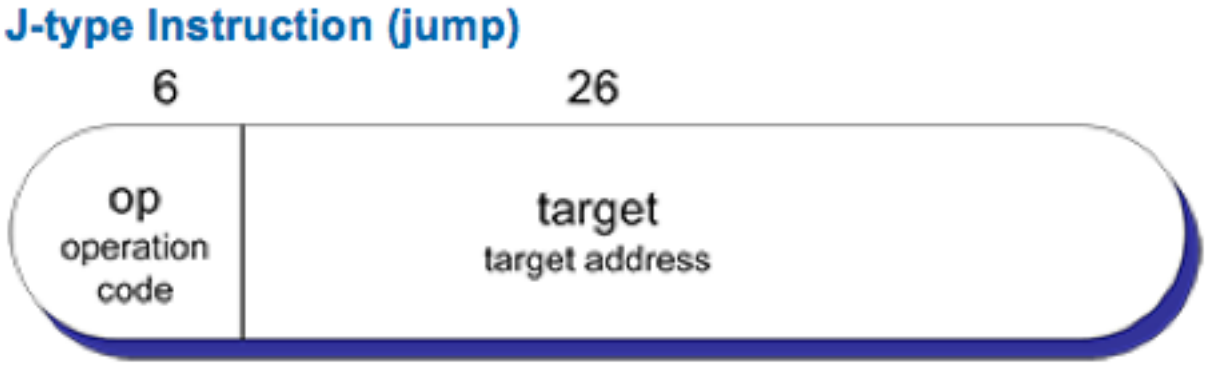


Example: beq $4,$8,exit

J-type Machine Instruction Format

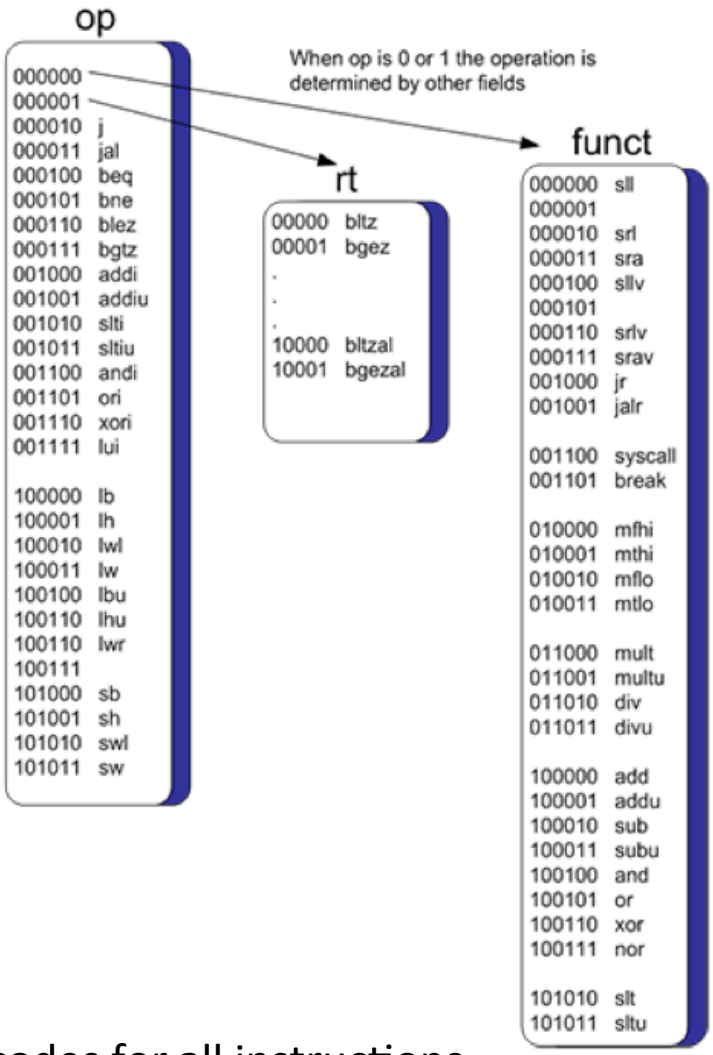
J-type uses:

- Bits 0 through 25 in generating jump address
- Shifted left 2 bits to make it a multiple of 4
- Aligned with a 4-byte instruction boundary
- Prepended with 4 upper bits from PC
- This yields the full 32-bit jump address



Examples: j exit
jal sqrt

Opcode Summary



Appendix A contains opcodes for all instructions