

## MIPS Instructions

5 min

The MIPS ISA is a simple instruction set that is broken up into three distinct types of instructions, all 32-bits in length:

- *R-Type* or *Register* MIPS instructions are used for most arithmetic and logic operations
- *I-Type* or *Immediate* instructions are used primarily for data transfer and immediate operations using constants
- *J-Type* or *Jump* instructions are used to jump the program to the specific instruction, such as in a loop

Along with the instruction types, it also details that each

Preview: Docs Loading link description

### [CPU](#)

will have 32 registers, each capable of holding a 32-bit piece of data. MIPS operates on data that is stored in the register or with a 16 bit 'immediate' piece of data. Immediate data is typically a constant that can be sent to the

Preview: Docs Loading link description

### [processor](#)

so it doesn't need to take up space in a register.

MIPS is often used in distributed/embedded technologies because of its RISC architecture and concise instruction set. Some advantages to this in a small system include limited space requirements, increased battery life, and little to no customer interaction.

## Instruction Format

Here is a sample R-type instruction:

```
000000 00000 00000 00000 00000 000000
op  rs  rt  rd  shamt func
```

Copy to Clipboard

All R-type instructions use this instruction format according to the MIPS documentation. This example introduces several abbreviations above in the machine code/instructions. They will be used throughout the rest of this lesson, so let's define them now:

Abbreviation	Definition
op	OPCODE
rs	first source register
rt	second source register
rd	destination register

Abbreviation	Definition
shamt	bit shift amount
func	extra bits for additional functions

### Instructions

1. Checkpoint 1 Passed

#### 1.

Create a new variable, answer1, and set it equal to the answer of:

- How many bits long is a MIPS instruction?

2. Checkpoint 2 Passed

#### 2.

Create another variable, answer2, and set it equal to the answer of:

- How many types of instructions are there in MIPS?

3. Checkpoint 3 Passed

#### 3.

Finally, create another variable, answer3, and set it equal to the answer of:

- What is the abbreviation used for the destination register?

### script.py

```
answer1 = 32
```

```
answer2 = 3
```

```
answer3 = "rd"
```