

MIPS

WANG Xiyu

August 27, 2024

Contents

1	Instruction Set Architecture	2
2	Machine Code vs Assembly Language	3
3	Walkthrough	3
4	General Purpose Register	3
5	MIPS Assembly Language	3
5.1	General Instruction Syntax	3
5.2	Arithmetic Operation: Addition	3
5.3	Arithmetic Operation: Subtraction	3
5.4	Complex Expressions	3
5.5	Constant/Immediate Operand	3
5.6	Register Zero (\$0 or \$zero)	3
5.7	Logical Operation: Overview	3
5.8	Logical Operation: Shifting	3
5.9	Logical Operation: Bitwise AND	3
5.10	Logical Operation: Bitwise OR	3
5.11	Logical Operation: Bitwise NOR	3
5.12	Logical Operation: Bitwise XOR	3
6	Large Constant: Case study	3
7	MIPS Basic Instruction Checklist	3
7.1	Subsection Title	3
7.1.1	Subsubsection Title	3

1 Instruction Set Architecture

Instruction Set Architecture includes everything programmers need to know to make the machine code work correctly, and allows computer designers to talk about functions independently from the hardware that performs them. This abstraction allows many implementations of varying cost and performance that can run identical software. Basically an API between software and hardware.

2 Machine Code vs Assembly Language

3 Walkthrough

4 General Purpose Register

5 MIPS Assembly Language

5.1 General Instruction Syntax

5.2 Arithmetic Operation: Addition

5.3 Arithmetic Operation: Subtraction

5.4 Complex Expressions

5.5 Constant/Immediate Operand

5.6 Register Zero (\$0 or \$zero)

5.7 Logical Operation: Overview

5.8 Logical Operation: Shifting

5.9 Logical Operation: Bitwise AND

5.10 Logical Operation: Bitwise OR

5.11 Logical Operation: Bitwise NOR

5.12 Logical Operation: Bitwise XOR

6 Large Constant: Case study

7 MIPS Basic Instruction Checklist

7.1 Subsection Title

7.1.1 Subsubsection Title

// Add your code example here

Column 1	Column 2	Column 3
Data 1	Data 2	Data 3

Table 1: Table caption

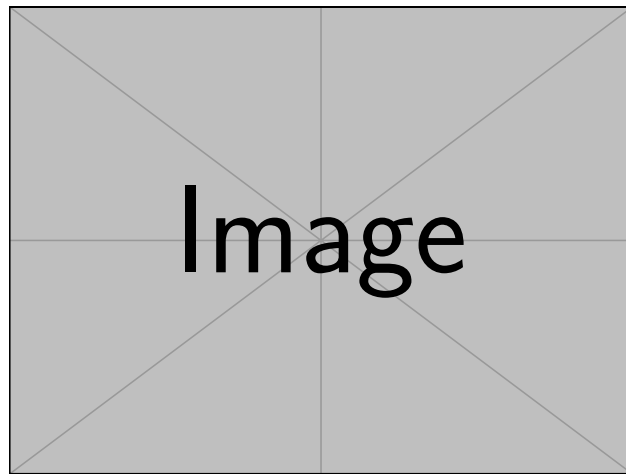


Figure 1: This is a sample caption.