Universidad de Carabobo

FACYT

Computación

Repertorio de Instrucciones MIPS32

Análisis de las instrucciones fundamentales y formatos

Materia:
Arquitectura del Computador

Elaborado por:
Alessandro Ocanto
C.I: 31.467.550

6 de diciembre de 2025

1. Introducción

La arquitectura MIPS (Microprocessor without Interlocked Pipeline Stages) es uno de los ejemplos más representativos de la filosofía RISC (Reduced Instruction Set Computer). Su diseño se centra en un conjunto de instrucciones simplificado y regular, lo que permite una ejecución eficiente a través de la segmentación (pipelining).

El presente documento tiene como objetivo sintetizar y categorizar las instrucciones más críticas del repertorio MIPS32. Basado en los datos de referencia estándar (Patterson & Hennessy), se desglosan las operaciones aritméticas, lógicas, de transferencia de datos y de control de flujo. Asimismo, se describen los formatos de instrucción básicos (R, I, J) que definen cómo el hardware interpreta estos comandos.

Este resumen sirve como guía técnica para comprender la manipulación de registros, el acceso a memoria y la lógica de ramificación a nivel de lenguaje ensamblador.

2. Formatos de Instrucción

Todas las instrucciones MIPS tienen una longitud fija de 32 bits. La simplicidad del diseño se refleja en el uso de solo tres formatos básicos:

Cuadro 1: Formatos Básicos de Instrucción MIPS

Tipo	**Campos (bits)**					
R	`op` (6)	`rs` (5)	`rt` (5)	`rd` (5)	`shamt` (5)	`funct` (6)
I	`op` (6)	`rs` (5)	`rt` (5)	`inmediato` (16)		
J	`op` (6)	`dirección` (26)				

3. Instrucciones Aritméticas

Estas instrucciones realizan operaciones matemáticas en registros. La mayoría utiliza el formato tipo R (registro a registro) o tipo I (con un operando inmediato).

Cuadro 2: Instrucciones Aritméticas Principales

Nombre	**Mnemónico**	**Fmt**	**Operación (Verilog)**
Suma	`add`	R	$R[rd] = R[rs] + R[rt]$
Suma inmediata	`addi`	I	$R[rt] = R[rs] + ExtSigno(Inm)$
Suma sin signo	`addu`	R	$R[rd] = R[rs] + R[rt]$
Resta	`sub`	R	$R[rd] = R[rs] - R[rt]$
Multiplicación	`mult`	R	$\{Hi, Lo\} = R[rs] * R[rt]$
División	`div`	R	$Lo = R[rs]/R[rt]; Hi = R[rs]\,\%R[rt]$

4. Instrucciones Lógicas

Utilizadas para manipulación de bits. Son esenciales para enmascaramiento y operaciones booleanas.

Cuadro 3: Instrucciones Lógicas

Nombre	**Mnemónico**	**Fmt**	**Operación**
And	`and`	R	$R[rd] = R[rs]$ & $R[rt]$
Or	`or`	R	$R[rd] = R[rs] \mid R[rt]$
Nor	`nor`	R	$R[rd] = \sim (R[rs] \mid R[rt])$
And inmediato	`andi`	I	$R[rt] = R[rs]$ & $ZeroExt(Inm)$
Despl. Izquierda	`sll`	R	$R[rd] = R[rt] \ll shamt$
Despl. Derecha	`srl`	R	$R[rd] = R[rt] \gg shamt$

5. Transferencia de Datos (Memoria)

MIPS es una arquitectura *Load/Store*, lo que significa que solo estas instrucciones pueden acceder a la memoria RAM.

Cuadro 4: Carga y Almacenamiento

Nombre	**Mnemónico**	**Fmt**	**Operación**
Carga palabra	`lw`	I	$R[rt] = M[R[rs] + ExtSigno(Inm)]$
Almacena palabra	`sw`	I	$M[R[rs] + ExtSigno(Inm)] = R[rt]$
Carga byte	`lb`	I	$R[rt] = ExtSigno(M[R[rs] + Inm])$
Almacena byte	`sb`	I	$M[R[rs] + Inm] = R[rt](7:0)$
Carga sup. inm.	`lui`	I	$R[rt] = \{Inm, 16'b0\}$

6. Control de Flujo (Saltos)

Permiten alterar la secuencia de ejecución, vitales para bucles (loops) y condicionales (if/else).

Cuadro 5: Saltos Condicionales e Incondicionales

Nombre	**Mnemónico**	**Fmt**	**Descripción**
Salto si igual	`beq`	I	Si $(R[rs] == R[rt])$ ir a PC + 4 + Dir
Salto si distinto	`bne`	I	Si $(R[rs] != R[rt])$ ir a PC + 4 + Dir
Salto incond.	`j`	J	PC = Dirección Salto
Saltar y enlazar	`jal`	J	$R[31] = PC + 8$; PC = DirSalto
Salto registro	`jr`	R	$PC = R[rs]$
Fijar si menor	`slt`	R	$R[rd] = (R[rs] < R[rt])?1 : 0$

7. Registros Principales

La arquitectura define 32 registros de propósito general con convenciones de uso específicas para preservar la compatibilidad entre llamadas a funciones.

- **$zero (0):** Constante 0. No se puede sobrescribir.
- **$v0 - $v1 (2-3):** Valores de retorno de funciones.
- **$a0 - $a3 (4-7):** Argumentos de funciones.
- **$t0 - $t9:** Registros temporales (no se preservan en llamadas).
- **$s0 - $s7:** Registros salvados (se preservan en llamadas).
- **$gp (28):** Puntero global.
- **$sp (29):** Puntero de pila (Stack Pointer).
- **$ra (31):** Dirección de retorno (Return Address).