

## Resumen de MIPS 32: Registros e Instrucciones

MIPS (Microprocessor without Interlocked Pipeline Stages) es una arquitectura RISC (Reduced Instruction Set Computer).

### Registros

MIPS 32 tiene 32 registros de propósito general (GPRs) de 32 bits cada uno. Además, existen registros especiales para la unidad de punto flotante (FPU), y registros HI y LO para resultados de multiplicación y división.

### Registros de Propósito General (GPRs)

Nombre	Número	Uso	Preservado en llamada
\$zero	0	Valor constante 0	No aplicable
\$at	1	Ensamblador temporal (no usar directamente)	No
\$v0- \$v1	2-3	Valores de retorno de funciones, evaluación de expr.	No
\$a0- \$a3	4-7	Argumentos de funciones (los primeros 4)	No
\$t0-\$t7	8-15	Temporales	No
\$s0-\$s7	16-23	Salvados (preservados a través de llamadas)	Sí
\$t8-\$t9	24-25	Temporales	No
\$k0- \$k1	26-27	Reservados para el núcleo del Sistema Operativo	No aplicable
\$gp	28	Puntero global	Sí
\$sp	29	Puntero de pila (Stack Pointer)	Sí
\$fp	30	Puntero de marco (Frame Pointer)	Sí
\$ra	31	Dirección de retorno	Sí

### Registros Especiales

- **PC (Program Counter):** Puntero a la instrucción actual. No es directamente direccionable por la mayoría de las instrucciones.
- **HI, LO:** Registros de 32 bits que almacenan el resultado de 64 bits de operaciones de multiplicación, o el cociente (LO) y resto (HI) de operaciones de división.

### Registros de Punto Flotante (FPU - Coprocesador 1)

MIPS tiene 32 registros de punto flotante de 32 bits (\$f0 - \$f31).

- Para operaciones de precisión simple (32 bits), se usa un registro (e.g., \$f0).
- Para operaciones de precisión doble (64 bits), se usan pares de registros (e.g., \$f0 y \$f1, referenciados por el registro par \$f0).

- Existe un registro de control/estado para la FPU que incluye flags de condición.

### Formatos de Instrucción

Las instrucciones MIPS tienen una longitud fija de 32 bits y se dividen principalmente en tres formatos:

- **Tipo R (Registro):** Usado para operaciones registro-registro. op (6) | rs (5) | rt (5) | rd (5) | shamt (5) | funct (6)
  - op: Código de operación (opcode).
  - rs: Primer operando fuente (registro).
  - rt: Segundo operando fuente (registro).
  - rd: Registro destino.
  - shamt: Cantidad de desplazamiento (shift amount).
  - funct: Función (varía el opcode).
- **Tipo I (Inmediato):** Usado para operaciones con un valor inmediato, cargas y almacenamientos, y saltos condicionales. op (6) | rs (5) | rt (5) | inmediato (16)
  - op: Código de operación.
  - rs: Operando fuente (registro).
  - rt: Operando fuente/destino (registro).
  - inmediato: Valor inmediato de 16 bits (puede ser una constante o un desplazamiento).
- **Tipo J (Salto):** Usado para saltos incondicionales. op (6) | dirección (26)
  - op: Código de operación.
  - dirección: Parte de la dirección de salto.
- **Formatos de Punto Flotante (FR, FI):** Similares a R e I pero con campos específicos para la FPU.

### Repertorio de Instrucciones MIPS (Núcleo y Comunes)

#### 1. Instrucciones Aritméticas

Mnemónico	Significado	Formato	Ejemplo (Ensamblador)
add	Suma con signo (con detección de desb.)	R	add \$s1, \$s2, \$s3
addu	Suma sin signo (sin detección de desb.)	R	addu \$s1, \$s2, \$s3

Mnemónico	Significado	Formato	Ejemplo (Ensamblador)
sub	Resta con signo (con detección de desb.)	R	sub \$s1, \$s2, \$s3
subu	Resta sin signo (sin detección de desb.)	R	subu \$s1, \$s2, \$s3
addi	Suma con inmediato (con signo, desb.)	I	addi \$s1, \$s2, 100
addiu	Suma con inmediato (sin signo, sin desb.)	I	addiu \$s1, \$s2, 100
mult	Multiplica (resultado en HI, LO)	R	mult \$s2, \$s3
multu	Multiplica sin signo (HI, LO)	R	multu \$s2, \$s3
div	Divide (cociente LO, resto HI)	R	div \$s2, \$s3
divu	Divide sin signo (LO, HI)	R	divu \$s2, \$s3
mfhi	Mover desde HI	R	mfhi \$s1
mflo	Mover desde LO	R	mflo \$s1

## 2. Instrucciones de Transferencia de Datos

Mnemónico	Significado	Formato	Ejemplo (Ensamblador)
lw	Cargar palabra (Load Word)	I	lw \$s1, 100(\$s2)
sw	Almacenar palabra (Store Word)	I	sw \$s1, 100(\$s2)
lh	Cargar media palabra (con signo)	I	lh \$s1, 100(\$s2)
lhu	Cargar media palabra (sin signo)	I	lhu \$s1, 100(\$s2)
sh	Almacenar media palabra	I	sh \$s1, 100(\$s2)
lb	Cargar byte (con signo)	I	lb \$s1, 100(\$s2)
lbu	Cargar byte (sin signo)	I	lbu \$s1, 100(\$s2)
sb	Almacenar byte	I	sb \$s1, 100(\$s2)
lui	Cargar inmediato superior	I	lui \$s1, 255

## 3. Instrucciones Lógicas

Mnemónico	Significado	Formato	Ejemplo (Ensamblador)
and	AND lógico	R	and \$s1, \$s2, \$s3
or	OR lógico	R	or \$s1, \$s2, \$s3
xor	XOR lógico	R	xor \$s1, \$s2, \$s3
nor	NOR lógico	R	nor \$s1, \$s2, \$s3

Mnemónico	Significado	Formato	Ejemplo (Ensamblador)
andi	AND lógico con inmediato	I	andi \$s1, \$s2, 100
ori	OR lógico con inmediato	I	ori \$s1, \$s2, 100
xori	XOR lógico con inmediato	I	xori \$s1, \$s2, 100
sll	Desplazamiento lógico a la izquierda	R	sll \$s1, \$s2, 4
srl	Desplazamiento lógico a la derecha	R	srl \$s1, \$s2, 4
sra	Desplazamiento aritmético a la derecha	R	sra \$s1, \$s2, 4
sllv	Despl. lógico izq. (variable)	R	sllv \$s1, \$s2, \$s3
srlv	Despl. lógico der. (variable)	R	srlv \$s1, \$s2, \$s3
srav	Despl. aritmético der. (variable)	R	srav \$s1, \$s2, \$s3

#### 4. Instrucciones de Salto Condicional

Mnemónico	Significado	Formato	Ejemplo (Ensamblador)
beq	Saltar si igual (Branch on Equal)	I	beq \$s1, \$s2, Etiqueta
bne	Saltar si no igual (Branch on Not Equal)	I	bne \$s1, \$s2, Etiqueta
slt	Poner si menor que (Set on Less Than)	R	slt \$s1, \$s2, \$s3
sltu	Poner si menor que (sin signo)	R	sltu \$s1, \$s2, \$s3
slti	Poner si menor que (inmediato)	I	slti \$s1, \$s2, 100
sltiu	Poner si menor que (inmediato, sin signo)	I	sltiu \$s1, \$s2, 100

#### 5. Instrucciones de Salto Incondicional

Mnemónico	Significado	Formato	Ejemplo (Ensamblador)
j	Saltar (Jump)	J	j Etiqueta
jr	Saltar a registro (Jump Register)	R	jr \$ra
jal	Saltar y enlazar (Jump And Link)	J	jal Procedimiento

#### 6. Instrucciones de Punto Flotante (Comunes, precisión simple .s, doble .d)

Mnemónico	Significado	Formato	Ejemplo (Ensamblador)
add.s / add.d	Suma FP	FR	add.s \$f0, \$f1, \$f2
sub.s / sub.d	Resta FP	FR	sub.d \$f0, \$f2, \$f4
mul.s / mul.d	Multiplicación FP	FR	mul.s \$f0, \$f1, \$f2
div.s / div.d	División FP	FR	div.d \$f0, \$f2, \$f4
lwc1	Cargar palabra a coprocesador 1 (FPU)	I	lwc1 \$f0, 100(\$s0)
swc1	Almacenar palabra de coprocesador 1 (FPU)	I	swc1 \$f0, 100(\$s0)
c.eq.s / c.eq.d	Comparar FP si igual	FR	c.eq.s \$f0, \$f1
c.lt.s / c.lt.d	Comparar FP si menor que	FR	c.lt.d \$f0, \$f1
bc1t	Saltar si flag de condición FP es verdadero	FI	bc1t EtiquetaFP
bc1f	Saltar si flag de condición FP es falso	FI	bc1f EtiquetaFP

## 7. Pseudo-Instrucciones Comunes

El ensamblador MIPS traduce estas instrucciones a una o más instrucciones MIPS reales.

Mnemónico	Significado	Traducción típica (Ejemplo)
move	Mover el contenido de un registro a otro	addu \$t0, \$zero, \$s0
li	Cargar inmediato (Load Immediate)	addiu \$t0, \$zero, constante (o lui+ori)
la	Cargar dirección (Load Address)	lui \$at, superior(dir); ori \$t0, \$at, inferior(dir)
blt	Saltar si menor que (Branch on Less Than)	slt \$at, \$s0, \$s1; bne \$at, \$zero, Etiqueta
bgt	Saltar si mayor que (Branch on Greater Than)	slt \$at, \$s1, \$s0; bne \$at, \$zero, Etiqueta
ble	Saltar si menor o igual (Branch on Less or Equal)	slt \$at, \$s1, \$s0; beq \$at, \$zero, Etiqueta
bge	Saltar si mayor o igual (Branch on Greater or Equal)	slt \$at, \$s0, \$s1; beq \$at, \$zero, Etiqueta