

HOJA DE REFERENCIA MIPS

FORMATOS DE INSTRUCCIÓN

Tipo	Estructura (31←0)	¿Para qué sirve?	Regla Memotécnica
R (Registro)	op(6) rs(5) rt(5) rd(5) shamt(5) funct(6)	Operaciones entre registros	Registros con R esultado en rd
I (Inmediato)	op(6) rs(5) rt(5) valor_inmediato(16)	Operaciones con constante	Inmediato = valor directo
J (Salto)	op(6) dirección_de_salto(26)	Salto largo	J ump = brinca lejos

Clave: El `opcode` (primeros 6 bits) siempre dice el formato

INSTRUCCIONES ESENCIALES

Aritméticas/Lógicas (Hex en mente)

Mnemónico	Operación	¿Qué hace?	Código Hex	Truco Memotécnico
add / addu	rd = rs + rt	Suma (con/sin signo)	0x20 / 0x21	add = 20 (como 20+20)
addi / addiu	rt = rs + Imm	Suma con constante	0x08 / 0x09	i de inmediato = 8/9
sub / subu	rd = rs - rt	Resta	0x22 / 0x23	sigue a add (22,23)
and / or	rd = rs & rt	AND/OR bit a bit	0x24 / 0x25	24=AND, 25=OR
sll / srl	rd = rt <</>> shamt	Desplazamientos	0x00 / 0x02	shift left=00, right=02
nor	rd = ~(rs rt)	NOT OR	0x27	Única forma de NOT

Memoria (Load/Store)

Mnemónico	Operación	¿Para qué?	Hex	Memoria Visual
lw	rt = Mem[rs+Off]	Cargar palabra	0x23	L oad W ord = 23
sw	Mem[rs+Off] = rt	Guardar palabra	0x2B	S tore W ord = 2B
lui	rt = {Imm, 16'b0}	Cargar parte alta	0x0F	Load U pper I mmEDIATE

Salto y Control

Mnemónico	Operación	Condición	Hex	Recordatorio
beq	if(rs==rt) PC+=0ff	Salta si igual	0x04	Branch E Qual = 04
bne	if(rs!=rt) PC+=0ff	Salta si distinto	0x05	Branch N ot E qual = 05
slt	rd = (rs<rt)?1:0	Compara menor que	0x2A	S et L ess T han
j	PC = DirSalto	Salto absoluto	0x02	J ump directo
jal	\$ra=PC+8; PC=Dir	Llamar función	0x03	J ump A nd L ink
jr	PC = rs	Salto por registro	0x08	J ump R egister

REGISTROS - CONVENCIÓN CRÍTICA

Nombre	Número	Uso Principal	¿Se guarda?	Regla Memotécnica
<code>\$zero</code>	0	Siempre vale 0	N/A	Cero = origen
<code>\$v0-\$v1</code>	2-3	Valores de retorno	No	V alor de vuelta
<code>\$a0-\$a3</code>	4-7	Argumentos	No	A rgumentos
<code>\$t0-\$t9</code>	8-15,24-25	Temporales	No	T emporales (tirados)
<code>\$s0-\$s7</code>	16-23	Variables importantes	Sí	S alvados (se guardan)
<code>\$sp</code>	29	Puntero de pila	Sí	S tack P ointer

Nombre	Número	Uso Principal	¿Se guarda?	Regla Memotécnica
\$fp	30	Marco de función	Sí	F rame P ointer
\$ra	31	Dirección retorno	Sí	R eturn A ddress

TRUCO: Los que empiezan con **S** se **S**alvan (*s0—s7, \$sp, \$fp, \$ra*)

PUNTO FLOTANTE (IEEE 754)

Fórmula:

Valor = (-1)^S × (1.Fracción) × 2^(Exponente - Bias)

Sesgos (Bias):

- Simple precisión (32 bits):** Bias = 127 (1-2-7)
- Doble precisión (64 bits):** Bias = 1023 (10-23 como los bits de fracción simple)

Casos especiales:

Exponente	Fracción	Resultado	Mnemotécnico
0	0	Cero	Todo en 0 = 0
Máximo	0	Infinito	Exp al máximo = ∞
Máximo	≠ 0	NaN	No es número válido

CÓDIGOS DE EXCEPCIÓN

Código	Nombre	Significado	Causa común
0	Int	Interrupción hardware	Timer, dispositivo
4	AdEL	Error lectura	Dirección inválida en lw
5	AdES	Error escritura	Dirección inválida en sw
8	Sys	Llamada sistema	Instrucción <code>syscall</code>
12	Ov	Overflow	Suma excede 32 bits

RESUMEN RÁPIDO

- Formatos:** R=registros, I=inmediato, J=salto lejano
- Instrucciones clave:** add(20), lw(23), sw(2B), beq(04)
- Registros:** Los `$s...` se SALVAN, los `$t...` se TIRAN
- Flotante:** Bias=127 (simple), casos: 0, ∞, NaN
- Excepciones:** 4=LEE mal, 5=ESCRIBE mal