**ORGANIZACION DE COMPUTADORAS**

**TP Nº2**



Alumno: Jeriel Estrada Candiano

Profesor: Santiago Trini

Fecha de Entrega: 06/05/2025

MIPS

En MIPS, las instrucciones se codifican en 32 bits. El formato exacto varía según el tipo de instrucción (tipo R, tipo I, J-type). Las instrucciones aritméticas básicas (add, sub) son de tipo R, mientras que addi y las instrucciones de carga/almacenamiento (lw, sw, li -que se traduce a addi o lui seguido de ori) son de tipo I.

Suponemos para los ejemplos las siguientes posiciones

$1 se refiere al registro número 1 (en binario 00001).

$2 se refiere al registro número 2 (en binario 00010).

$3 se refiere al registro número 3 (en binario 00011).

$4 se refiere al registro número 4 (en binario 00100).

$5 se refiere al registro número 5 (en binario 00101).

$6 se refiere al registro número 6 (en binario 00110).

$7 se refiere al registro número 7 (en binario 00111).

$8 se refiere al registro número 8 (en binario 01000).

$9 se refiere al registro número 9 (en binario 01001).

$10 se refiere al registro número 10 (en binario 01010).

$11 se refiere al registro número 11 (en binario 01011).

$sp (puntero de pila, usualmente $29) se representará como 11101.

$gp (puntero global, usualmente $28) se representará como 11100.

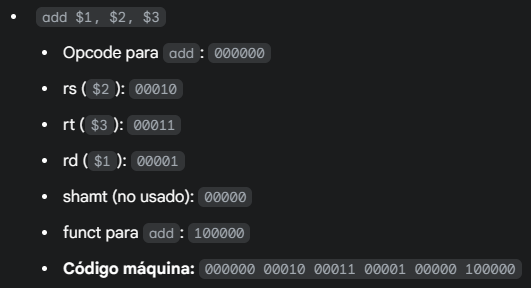
Los valores inmediatos se representarán en binario de 16 bits (para addi) o 32 bits (para la pseudo-instrucción li, aunque se desglosará).

### **Instrucciones Aritméticas (Formato R)**

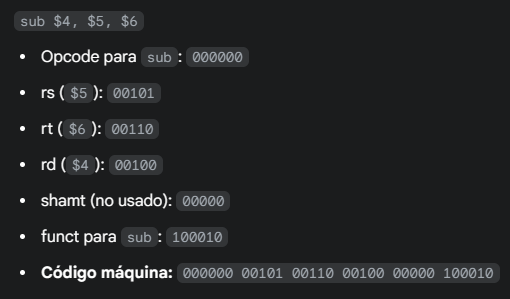
### El formato R tiene la siguiente estructura:



Instrucción Suma (add):



Resta (sub):

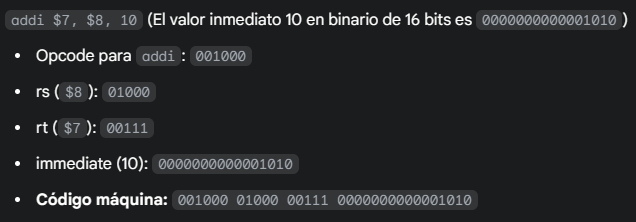


### **Instrucción Suma Inmediata (Formato I)**

El formato I tiene la siguiente estructura:

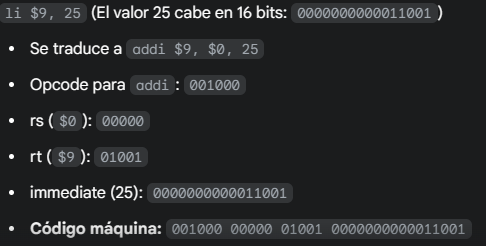


Instrucción Suma Inmediata (addi):



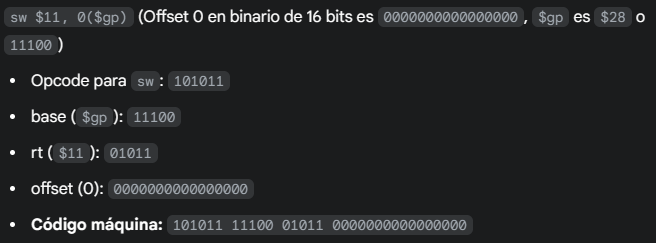
### **Instrucción Cargar Inmediato (Pseudo-instrucción li)**

La pseudo-instrucción li se traduce a una o dos instrucciones reales. Si el valor inmediato cabe en 16 bits con signo, se usa addi con $zero ($0). Si no, se usa lui (load upper immediate) para cargar la parte alta y ori (or immediate) para la parte baja.



Instrucciones de Carga y Almacenamiento (Formato I)





### **Resumiendo, representando en una tabla a las distintas operaciones**

|  |  |  |
| --- | --- | --- |
| Assembler | Lenguaje Maquina (binario) | Pseudocodigo |
| Lw A, 12(B) | 100011 10001 01000 0000000000001100 | A = memoria [B + 12] |
| Addi A, B, 5 | 001000 10011 01010 0000000000000101 | A = B + 5 |
| Sub C, D, A | 000000 01100 01101 01011 00000 100010 | C = D - A |
| J A | 000010 00000000010000000000000100 | Salta a A |
| Sw A, 100(D) | 101011 11101 01001 0000000001100100 | Memoria [D + 100] = A |
| Jr D | 000000 11111 00000 00000 00000 001000 | Salta a la dirección de retorno D |
| Add E, A, D | 000000 00101 00110 00010 00000 100000 | E = A + D |
| Lb T, offset(B) | 100000 11101 00010 0000000000100000 | T = [B + offset] |
| Sb T, offset(H) | 111111 11100 00100 0000000000000000 | Memoria [H] = T |