

Resumen de operandos y lenguaje ensamblador MIPS

Michelle Sayas

June 2025

1 Registros y Memoria

El procesador MIPS cuenta con 32 registros que incluyen \$10, \$20, \$0, \$40, etc. Algunos registros tienen funciones especiales como \$7 (no siempre es 0) y \$1 (reservado para manejar constantes grandes).

Memoria: La memoria principal en MIPS tiene un espacio de direcciones de aproximadamente 4GB. Es accesible solo mediante instrucciones de transferencia de datos. Usa direccionamiento por byte, con direcciones de palabras consecutivas separadas por 4 unidades y almacena estructuras de datos complejas, tablas de valores y registros desbordados para llamadas a procedimientos.

2 Instrucciones del Ensamblador MIPS

2.1 Aritméticas

Operaciones básicas como add, sub (Suma y resta con tres operandos en registros) y addi (Suma con un valor inmediato constante).

2.2 Transferencia de Datos (Carga y almacenamiento)

lw, sw: Transferencia de palabras completas entre memoria y registros. lh, sh, lb, sb: Transferencia de medias palabras y bytes. lui: Carga de constantes en los 16 bits superiores de un registro.

2.3 Lógicas

Para operaciones bit-a-bit se utilizan and, or, nor (Operaciones lógicas con tres registros), andi, ori (Operaciones lógicas con registros y constantes), sll, srl (Desplazamientos lógicos a izquierda y derecha), para saltos condicionales beq, bne (Saltan si los registros son iguales o diferentes) y slt, slti (Comparan si un valor es menor que otro) y para los incondicionales j (Salto a una dirección específica), jr (Salto a la dirección almacenada en un registro) y jal (Salto con almacenamiento de la dirección de retorno).

3 Consideraciones de Diseño

3.1 Filosofía RISC

Todas las instrucciones tienen formato simple y tamaño fijo.

3.2 Separación load/store

Solo instrucciones específicas acceden a memoria.

3.3 Uso de registros

La mayoría de operaciones trabajan exclusivamente con registros.

3.4 Manejo de constantes

Mecanismos especiales (addi, lui) para trabajar con valores inmediatos.

Este conjunto de instrucciones permite implementar operaciones complejas mediante combinaciones de estas instrucciones básicas, siguiendo los principios de diseño de arquitecturas RISC. La separación clara entre operaciones aritméticas, transferencias de datos y control de flujo facilita tanto la implementación hardware como la programación en ensamblador