

Análisis de la Arquitectura MIPS: Referencia y Algunos Conceptos Fundamentales

Samuel Cordero, CI:31678592

May 31, 2025

Introducción

Buenas tardes, profe Aquí le va mi resumen de algunas partes clave del libro de Patterson y Hennessy, ese de "Estructura y Diseño de Computadores" que nos trae de cabeza. La idea es entender cómo piensa una computadora, desde las instrucciones más básicas hasta cómo maneja los números decimales y las llamadas a funciones. Nos vamos a enfocar en las "chuletas" de MIPS y una página del capítulo 2 que nos aterriza un poco la cosa. Es un vistazo rápido, pero intentando no perdernos en lo que nos quiere explicar el libro.

Resumen: La Chuleta de MIPS (Parte 1)

Esta página es nuestra biblia rápida de MIPS. Básicamente, te da todas las instrucciones de enteros que vas a usar (suma, resta, cargar y guardar datos, saltos, etc.), con su nombre, cómo se escriben y qué hacen. Lo más importante, al menos para mí, es que te muestra cómo se ven esas instrucciones en binario (los formatos R, I y J). Así entiendes que no es magia, sino bits bien organizados. Y para no hacer la vida tan complicada, también nos recuerda las "pseudoinstrucciones" que el ensamblador convierte por nosotros. Finalmente, lo clásico: la lista de los 32 registros de MIPS y para qué se usa cada uno, lo de $s0, t0, a0$, etc., *que esclave para no hacer una lista de cosas para programar*.

- **Instrucciones Básicas:** Sumar, restar, mover datos de memoria a registros y viceversa, y los saltos para que el programa no sea lineal.
- **Formatos Binarios (R, I, J):** La estructura que le da sentido a cada instrucción en la máquina.
- **Pseudoinstrucciones:** Un "atajo" del ensamblador para operaciones comunes.
- **Registros MIPS:** Los 32 casilleros de almacenamiento rápido en el procesador y su uso "etiquetado".

Mini-resumen: La primera "chuleta" de MIPS organiza las instrucciones de enteros, sus formatos binarios, las pseudoinstrucciones y el rol de los 32 registros. ¡Un manual de supervivencia para el ensamblador y para nosotros!

Resumen: La Chuleta de MIPS (Parte 2)

Esta segunda parte de la "chuleta" es un poco más complicado de entender. Primero, te da el mapa completo de los códigos de operación y funciones de MIPS, o sea, el número binario exacto de cada instrucción. Después, y esto es clave para la vida real (y las científicas), se mete de lleno con el ****estándar IEEE 754 de punto flotante****. Nos explica la fórmula ($N = (-1)^S \times (1 + \text{Fracción}) \times 2^{(\text{Exponente} - \text{Sesgo})}$), cómo se guardan los números decimales (en simple o doble precisión) y qué significan cosas raras como "infinito" o "NaN" (Not a Number). También detalla las convenciones para llamar a funciones: dónde van los argumentos ($a0$), los resultados ($v0$), y qué registros hay que guardar sí o sí ($s0$). Para cerrar, te muestra los registros del coprocesador 0, que son los que usa el sistema operativo para manejar errores (como cuándo algo sale mal y se genera una excepción), y finalmente, nos recuerda los prefijos de tamaño (kilo, mega, giga) en potencias de 2 y de 10.

- **Códigos de Operación Detallados:** El "ADN" binario de cada instrucción MIPS.
- **Punto Flotante IEEE 754:** Cómo la computadora representa y maneja números decimales, incluyendo los "raros" (infinito, NaN).
- **Llamadas a Funciones:** El protocolo de cómo se pasan datos y se manejan los registros entre funciones.
- **Manejo de Excepciones (CP0):** Cómo el procesador y el SO gestionan los errores y las interrupciones.
- **Prefijos de Medida:** Definición de kilo, mega, etc., en sus versiones binaria y decimal.

Mini-resumen: La segunda "chuleta" cubre la codificación completa de instrucciones, el estándar de punto flotante IEEE 754, las reglas para llamadas a funciones y el control de excepciones con el Coprocesador 0. ¡Ya casi vemos la métrica a la hora de programar en lenguaje ensamblador!

Resumen: Página 78, Capítulo 2

Esta página es nuestra primera aproximación a cómo programar en ensamblador MIPS. Lo primero que te explica es dónde vive la información: en los **registros** (esos 32 espacios superrápidos dentro de la CPU) y en la **memoria principal** (el disco duro, por así decirlo, pero de trabajo). Nos recalca que la memoria se organiza en "palabras" y que hay que acceder a ellas de 4 en 4 bytes (por eso lo de 'Memory[0]', 'Memory[4]', etc.). Y lo más útil de la página es la tabla (Figura 2.1) que te da ejemplos claros de las instrucciones más básicas de MIPS: cómo sumar, cómo cargar datos de memoria a un registro, cómo guardarlos de vuelta, operaciones lógicas y, por supuesto, los saltos condicionales e incondicionales para controlar el flujo del programa. Es la base para entender cómo las instrucciones de un lenguaje de alto nivel se convierten en el "idioma" de la máquina.

- **Dónde Están los Datos:** Registros (rápidos) y Memoria (grande, organizada por palabras).
- **Instrucciones MIPS Ilustradas:** Ejemplos prácticos de las operaciones fundamentales (aritméticas, de E/S, lógicas y de control).

Mini-resumen: La página 78 introduce cómo MIPS maneja los datos (registros y memoria) y presenta ejemplos básicos de sus instrucciones clave, sentando las bases del ensamblador.

Conclusión

Después de revisar estas páginas, queda claro que MIPS no es solo un conjunto de instrucciones, ¡es una forma de pensar la arquitectura de computadores! Empezamos por entender los bloques básicos: dónde se guardan los datos (registros y memoria) y cómo se manipulan con las instrucciones más elementales. Luego, nos metimos en la parte "seria" de cómo se ven estas instrucciones en binario y cómo se manejan los números decimales complicados con el IEEE 754. También vimos que no todo es ejecutar código, sino también saber pasarle el control a otras partes del programa (las funciones) y cómo el sistema operativo se encarga de los errores. En resumen, estas referencias son como la "ficha técnica" de la computadora, indispensable para cualquier programador o diseñador que quiera ir más allá de lo superficial y realmente entender cómo funciona el "cerebro" electrónico. ¡Ahora, solo queda ponerlo en práctica con algunos ejercicios que consigamos en el libro que nos mandó!