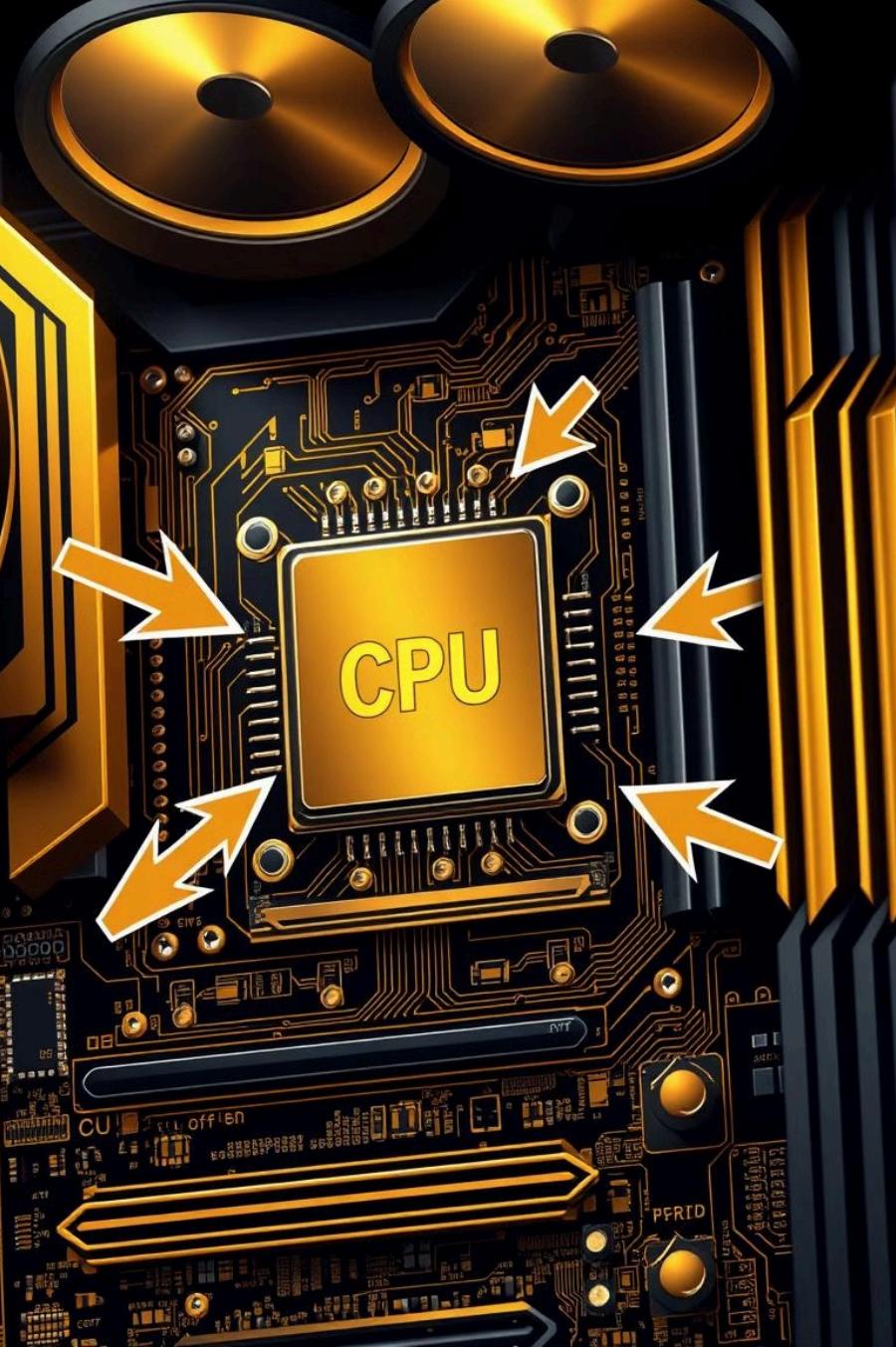


Introducción a ARM64

ARM64 es una arquitectura de procesador de 64 bits desarrollada por ARM Holdings, líder mundial en diseño de microprocesadores. Esta tecnología de vanguardia proporciona un rendimiento y eficiencia energética superiores para una amplia gama de dispositivos, desde teléfonos inteligentes y tabletas hasta servidores y supercomputadoras. En esta presentación, exploraremos en detalle los conceptos clave de ARM64, desde el ensamblador hasta las instrucciones avanzadas, para comprender cómo esta innovadora arquitectura está transformando el paisaje tecnológico.

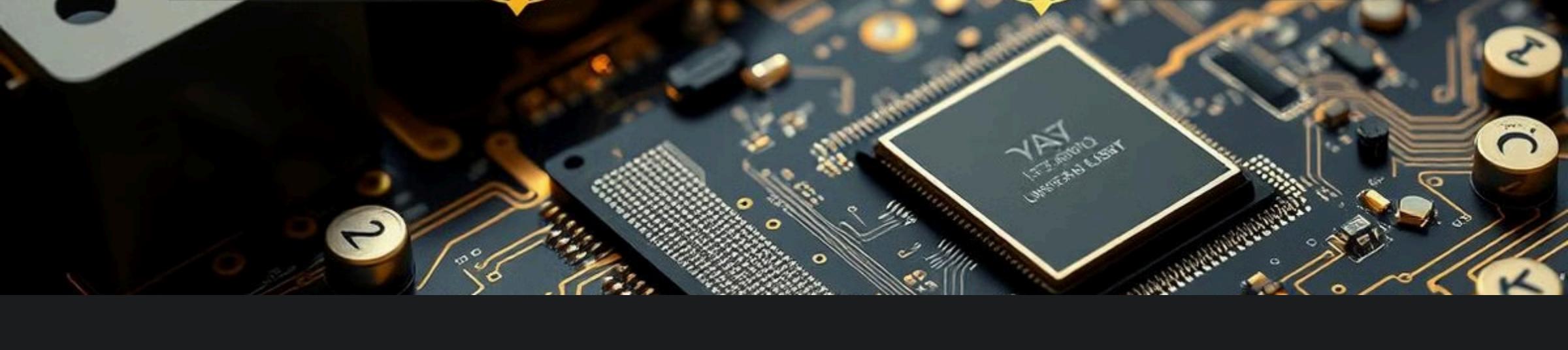
 por Hugo Martinez





Qué es el ensamblador

El **ensamblador** es un lenguaje de programación de bajo nivel que traduce instrucciones en un formato legible por humanos a código máquina entendible por el procesador. A diferencia de los lenguajes de alto nivel como Python o Java, el ensamblador ofrece un control más directo sobre la arquitectura del hardware. Los programas en ensamblador se componen de instrucciones que representan operaciones básicas como movimiento de datos, aritmética y lógica. Aprender ensamblador es fundamental para comprender el funcionamiento interno de los computadores y optimizar el rendimiento de los programas.



Vocabulario básico y tipos de datos

- **Operandos:** Los elementos que participan en una operación, como valores numéricos o variables.
- **Instrucciones:** Comandos que indican al procesador qué operación realizar, como sumar, restar o transferir datos.
- **Tipos de datos:** En ARM64 se manejan enteros, reales, y cadenas de caracteres, cada uno con un tamaño y rango de valores específico.

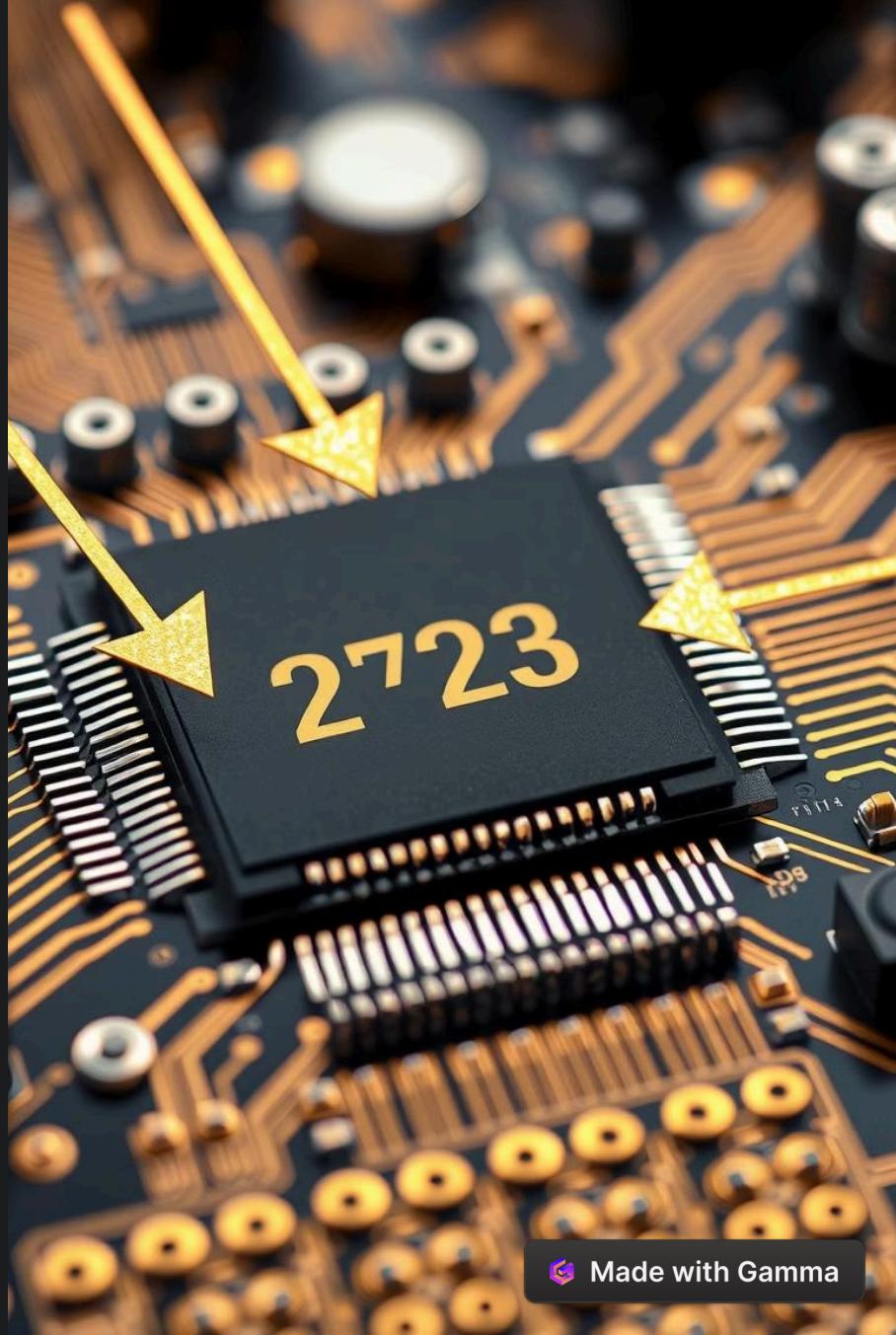
Registros básicos del procesador

1 Registros de alta velocidad

Los registros son ubicaciones de almacenamiento de alta velocidad dentro del procesador ARM64 que permiten realizar operaciones rápidamente.

2 Registros generales y especiales

Existen registros generales como X₀-X₃₀ que se utilizan para almacenar datos, y registros especiales como el contador de programa (PC) y el registro de estado (CPSR).



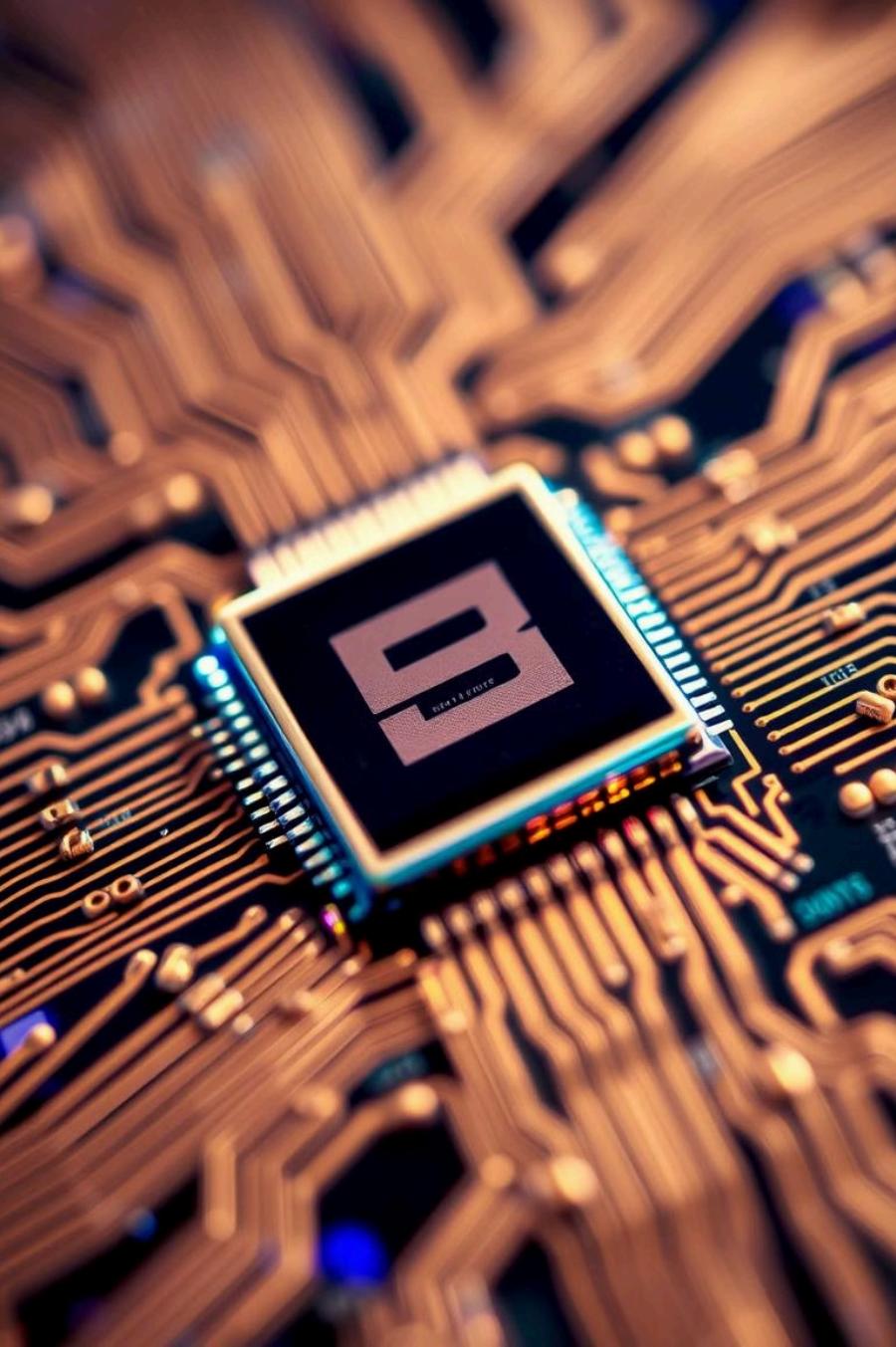
Instrucciones Aritméticas

Las **instrucciones aritméticas básicas** como ADD, SUB, MUL y DIV permiten realizar cálculos matemáticos sobre los registros del procesador.

Instrucciones como ADDS y SUBS **actualizan el registro de estado** con información sobre el resultado, como si fue positivo, negativo o cero.

Existen también **instrucciones avanzadas para operaciones con punto flotante**, como FADD y FSUB, que manipulan los registros vectoriales.





Instrucciones Lógicas

Operaciones Bit a Bit

Instrucciones lógicas como AND, ORR y EOR permiten realizar operaciones bit a bit para aplicar lógica booleana a los registros del procesador.

Extracción de Información

Estas instrucciones son fundamentales para manipular y extraer información específica de los datos almacenados.

Desplazamiento de Bits

Además, instrucciones como LSL y LSR se utilizan para desplazar bits a izquierda o derecha, lo que permite realizar operaciones de multiplicación y división por potencias de 2.



Instrucciones de carga y almacenamiento



Las instrucciones LDR y STR permiten cargar y almacenar datos en la memoria del sistema desde y hacia los registros del procesador.



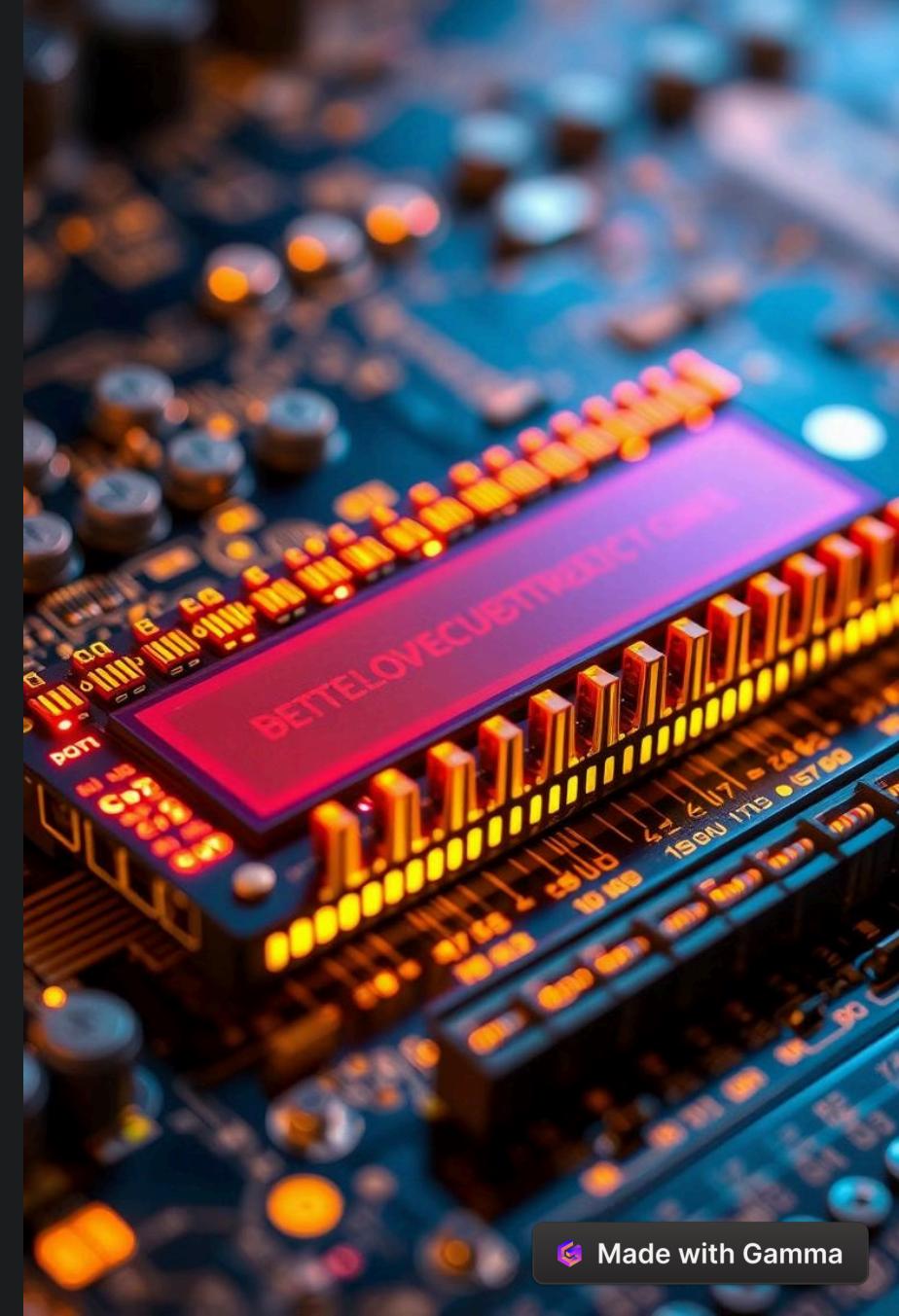
Existen variantes de estas instrucciones, como LDUR y STUR, que facilitan el acceso a datos alineados y no alineados en memoria.

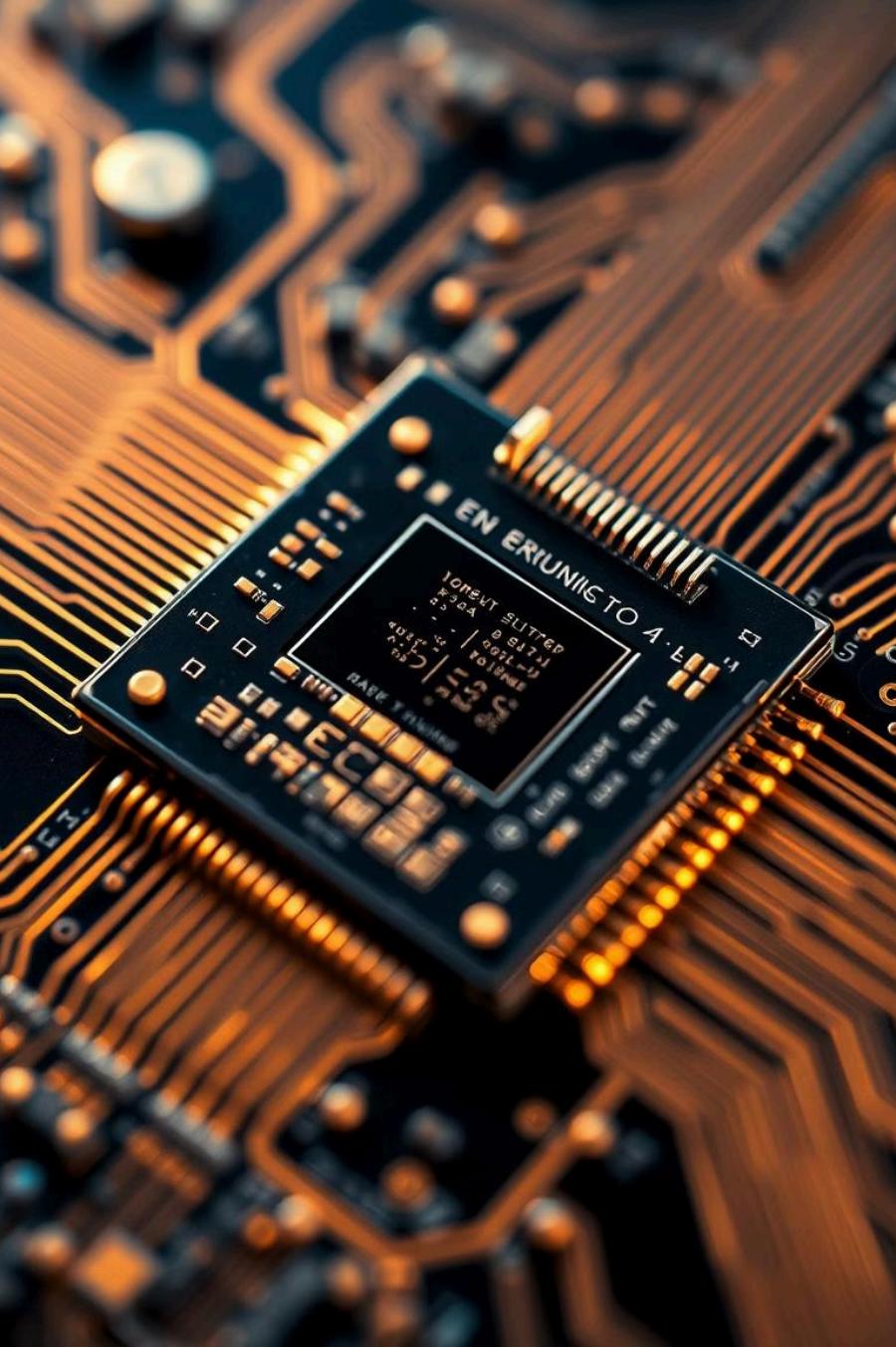


Instrucciones como LDRSW y LDRSB realizan operaciones de extensión de signo para cargar datos de diferentes tamaños en los registros.

Dirección de Memoria y Punteros

Los punteros son registros que almacenan direcciones de memoria, permitiendo acceder a datos de manera indirecta. En ARM64, los punteros utilizan el prefijo X o W, seguido de un número, para referirse a registros de 64 o 32 bits respectivamente. El manejo eficiente de punteros es clave para navegar la jerarquía de memoria y optimizar el rendimiento de los programas.





Estructuras de Control de Flujo

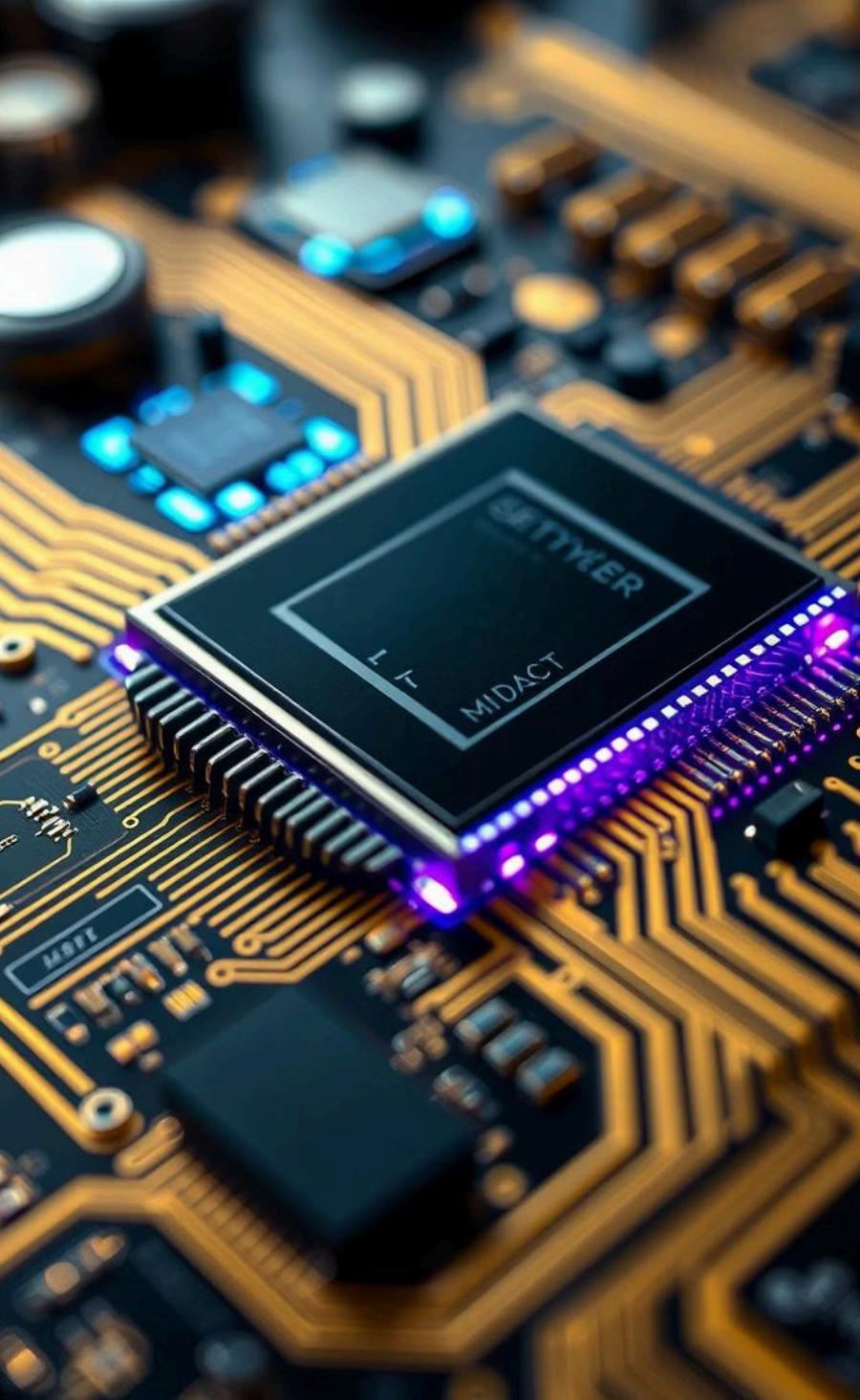


Saltos Incondicionales y Condicionales

Instrucciones como B (incondicional), B.cond (condicional) y BL (llamada a subrutina) permiten saltar a diferentes secciones del código, controlando el flujo de ejecución.

Estructuras de Control

El uso de estructuras como IF-THEN-ELSE y bucles WHILE o FOR posibilitan la toma de decisiones y la ejecución iterativa en los programas ensamblador.



Conclusiones y Resumen

1

Arquitectura de Alto Rendimiento

ARM64 es una arquitectura de 64 bits que ofrece un alto rendimiento y eficiencia energética para una amplia gama de dispositivos.

2

Control Directo del Hardware

El **ensamblador** brinda un control directo sobre el hardware, permitiendo optimizar el desempeño de los programas a través de instrucciones básicas.

3

Dominio de Conceptos Clave

El dominio de conceptos como registros, instrucciones aritméticas y lógicas, y estructuras de control de flujo, es clave para aprovechar al máximo las capacidades de ARM64.