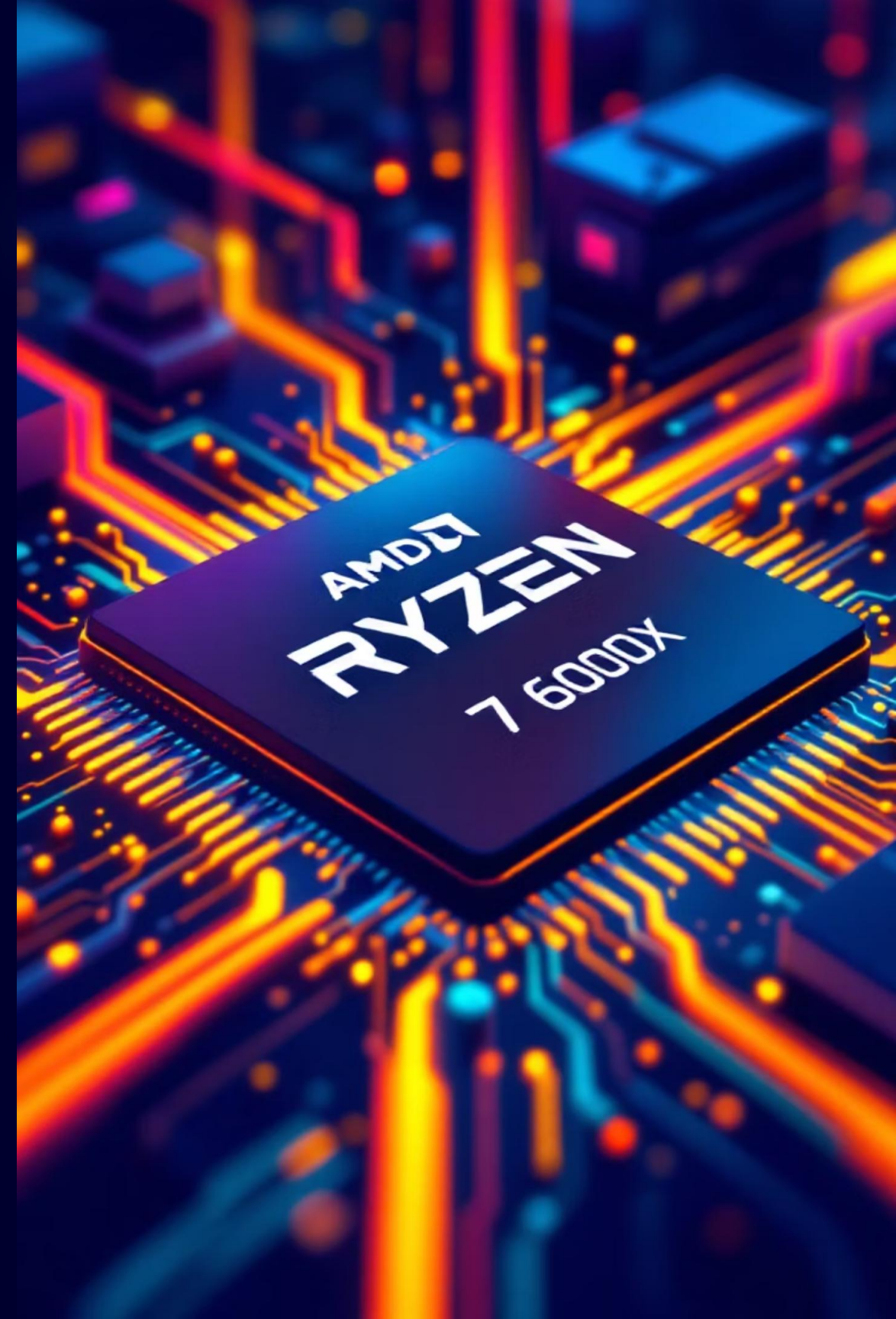


Computación de Altas Prestaciones

Predicción de Saltos: AMD Ryzen 5 7600X

Análisis de Técnicas de Branch Prediction en Arquitectura Zen 4

Miguel Barranquero Díez, Emilio Cobos Álvarez

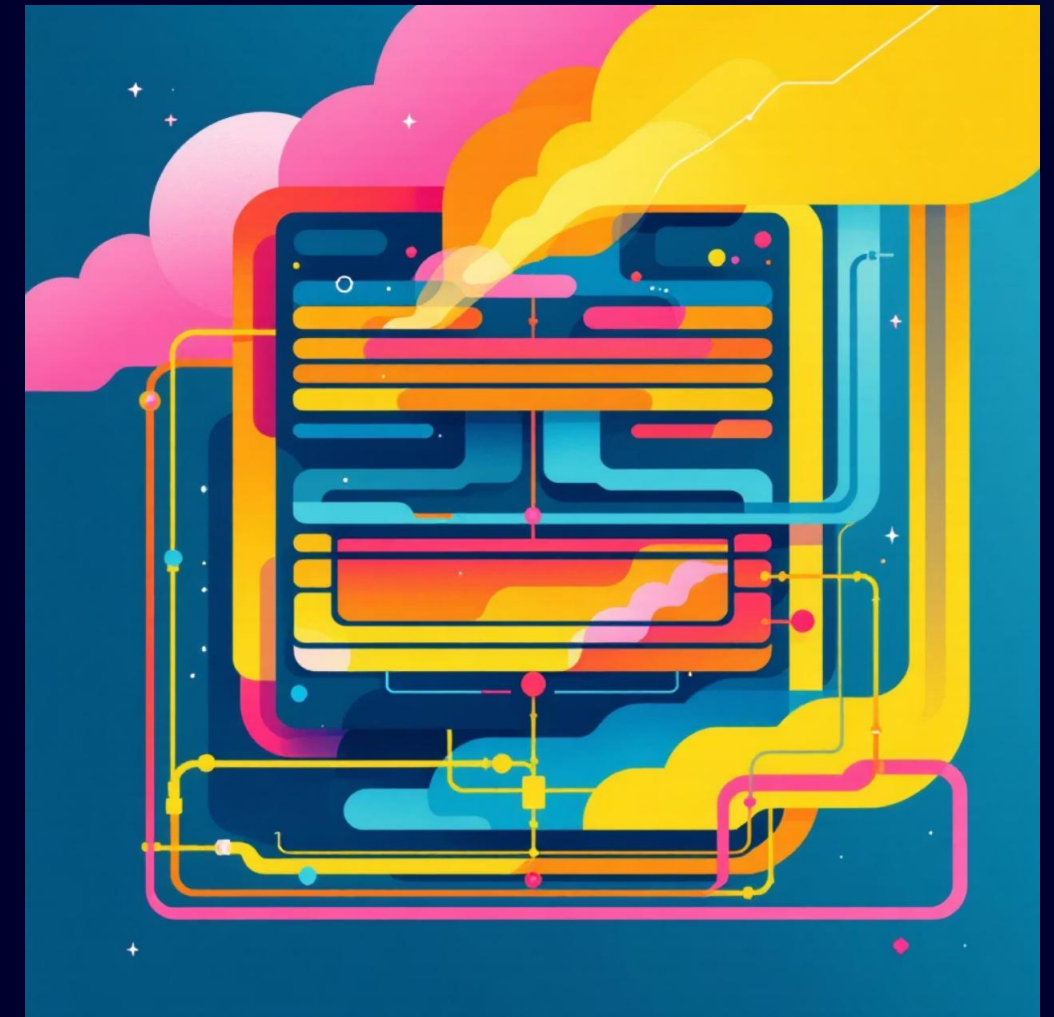


Introducción al AMD Ryzen 5 7600X

La arquitectura Zen 4 de AMD marca un avance significativo en el rendimiento de los procesadores modernos. Su diseño, que incluye un *pipeline* de 19 etapas, subraya la importancia crítica de una predicción de saltos eficaz.

- El coste de un error de predicción puede oscilar entre 15 y 20 ciclos de reloj, impactando directamente en la eficiencia.
- Aun así, la tasa de acierto supera el 95% en programas típicos.
- Este enfoque ha permitido un incremento del 13% en Instrucciones por Ciclo (IPC) respecto a Zen 3.

Su optimización beneficia aplicaciones clave como compiladores, simulaciones científicas y videojuegos.



Arquitectura del Sistema de Predicción

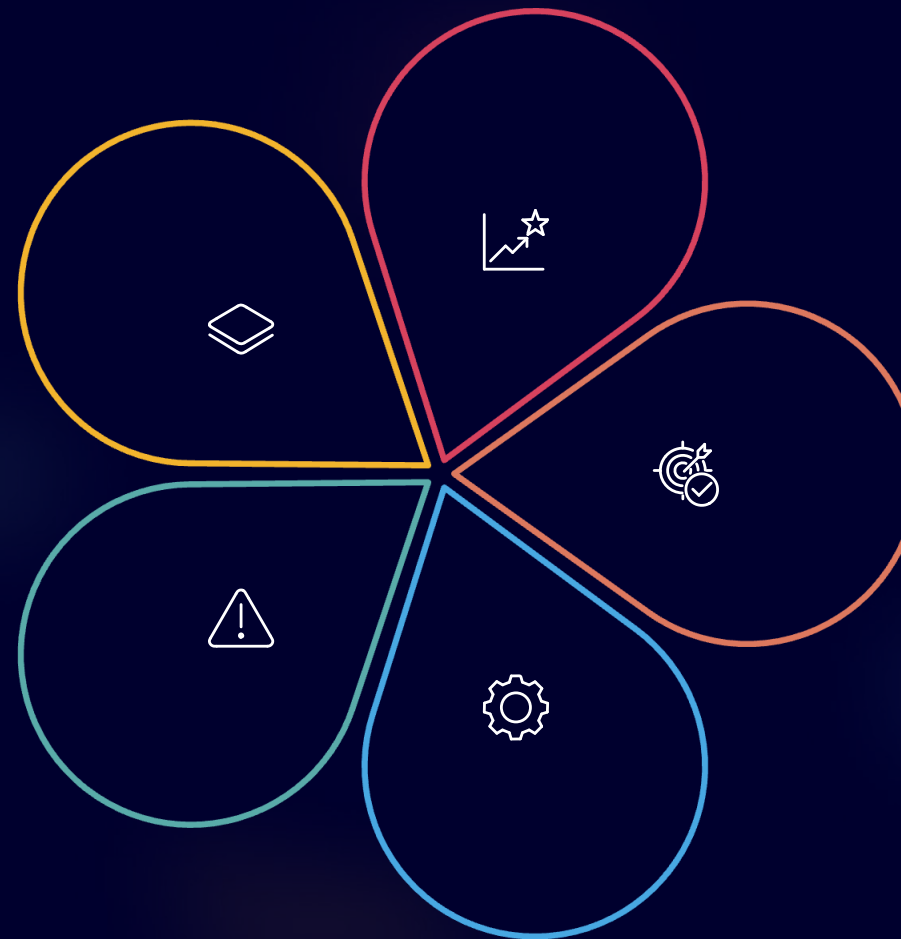
El AMD Ryzen 5 7600X emplea un sofisticado sistema jerárquico de predicción de saltos de dos niveles para optimizar tanto la velocidad como la precisión.

Sistema Jerárquico

Diseño de dos niveles (L1 y L2) para una aproximación balanceada.

Fallback Estático

Un mecanismo de predicción estática como respaldo para casos inusuales o de "código frío".



Nivel 1 (L1)

Un Predictor Perceptrón con Hash enfocado en la velocidad de respuesta.

Nivel 2 (L2)

Un Predictor TAGE para garantizar la máxima precisión en escenarios complejos.

Sistema Híbrido

Una configuración en cascada que integra y complementa ambos niveles.

Predictor Perceptrón (L1): Velocidad Optimizada

El Predictor Perceptrón de Nivel 1 (L1) es la primera línea de defensa para la predicción de saltos, priorizando la rapidez.

Características Clave

- Utiliza un modelo de **machine learning simple**.
- Opera con un **historial corto** (8-16 bits) de saltos previos.
- **Latencia mínima** de 1 ciclo de reloj, con una tasa de acierto del 90-92%.
- Incorpora una **función hash** para minimizar colisiones en la tabla.
- Integra un **mecanismo de confianza** para evaluar la fiabilidad de sus propias predicciones.

Implementación Técnica

- La tabla está **indexada por hash** (XOR de PC e historial).
- Se almacena en **memoria SRAM** de doble puerto para acceso paralelo.
- El cálculo se resuelve mediante una **suma ponderada vs. umbral** en un único ciclo de reloj.

Predictor TAGE (L2): Precisión Avanzada

El Predictor TAGE (Tagged Geometric Predictor) constituye el segundo nivel, centrado en lograr la máxima precisión, incluso con patrones de saltos complejos.

Concepto "Tagged Geometric Predictor"

- Diseñado para la **precisión superior** en patrones de saltos intrincados.
- Emplea **múltiples tablas** con historiales de longitud variable (8-128 bits).
- Utiliza **etiquetas (tags)** de 10-14 bits para evitar colisiones entre entradas.
- Alcanza una **tasa de acierto del >97%** en patrones de ejecución complejos.

Detalles de Implementación

- Sus tablas están **organizadas geométricamente**, de historiales cortos a largos.
- La consulta se realiza de forma **secuencial y pipelineada** (1-2 ciclos).
- Implementa un **esquema de promoción** para la actualización dinámica de las entradas.
- Permite el **particionamiento por hilo** para optimizar el soporte a SMT (Simultaneous Multithreading).

Sistema Híbrido en Cascada

La integración del Predictor Perceptrón (L1) y el TAGE (L2) forma un sistema híbrido en cascada, optimizando conjuntamente velocidad y precisión.



Integración L1 y L2

El Perceptrón L1 y el TAGE L2 trabajan en paralelo y en secuencia.



Meta-Predictor Dinámico

Un meta-predictor selecciona dinámicamente la predicción más fiable.



Validación por Confianza

L1 genera predicciones en paralelo; L2 valida si la confianza de L1 es baja.



Soporte Multisalida

Capaz de soportar hasta 2 saltos por ciclo en paralelo, mejorando el rendimiento.

Este diseño incluye una pila de retornos (RAS) de 32 entradas y utiliza **clock-gating** en L2 para ahorro energético.



Static branch-predictions



Predicción Estática (Fallback)

En situaciones donde los predictores dinámicos no tienen suficiente información, la predicción estática actúa como un mecanismo de respaldo simple pero esencial.

Mecanismo de Respaldo

- **Reglas sin aprendizaje:** "saltos hacia atrás → tomados" (ideal para bucles).
- "Saltos hacia adelante → no tomados" (común en bifurcaciones condicionales).
- Su uso es **marginal (<1%)**, principalmente en "código frío" o inicialización.
- **Latencia adicional nula**, ya que no requiere ciclos de cálculo complejos.

Casos Especiales

Return Stack Buffer (RSB)

Return Stack Buffer (RSB)

- 32 direcciones de retorno almacenadas
- Predice con precisión instrucciones return
- Sin cambios respecto a Zen 3 (diseño probado)

Predicción de Saltos Indirectos

- **Array dedicado** para branches indirectos (ej: llamadas virtuales)
- **Capacidad total:** 3,072 destinos posibles
- **Ventaja clave:** Hasta 32 destinos por salto sin pérdida de rendimiento
- **Uso:** Optimiza código con polimorfismo y funciones virtuales