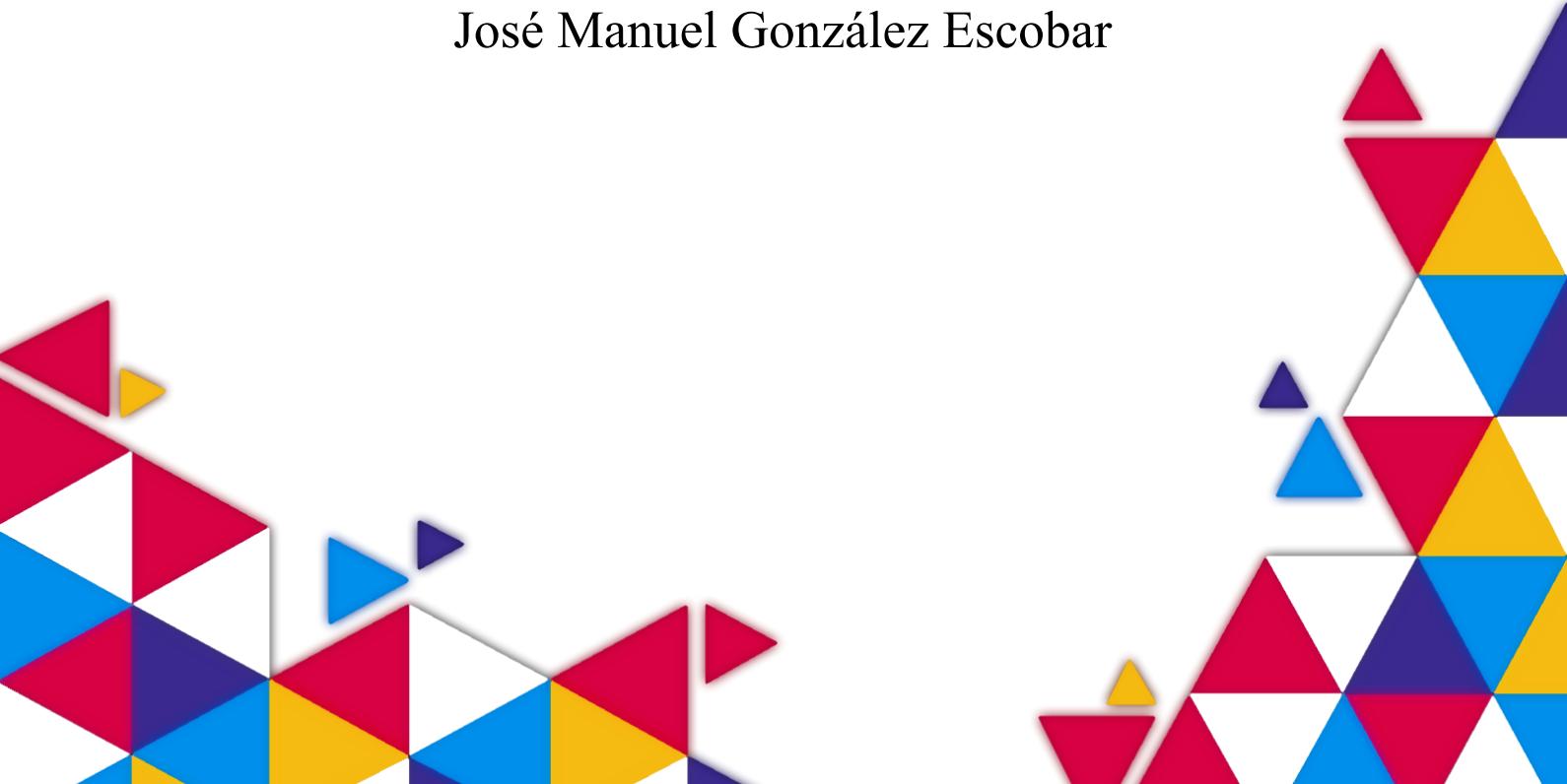


P8:PROBLEMA DE LA PROPAGACIÓN DEL ACARREO PARA ANCHO DE PALABRA GRANDE.

Arquitectura Avanzada de Procesadores

José Manuel González Escobar



¿Cómo se puede resolver el problema del acarreo propagado en la unidad aritmética cuando el ancho de palabra es muy grande como ocurre en el procesador MIPS?

Mostrar la solución mediante un ejemplo.

El problema del acarreo propagado en la unidad aritmética es una limitación que puede surgir en procesadores con palabras de ancho grande, como el procesador MIPS. La presencia de acarreos puede introducir retardos significativos en la ejecución de operaciones aritméticas, ya que cada bit de resultado depende del acarreo de su bit correspondiente en la operación anterior. Para resolver este problema, se utilizan técnicas como el lookahead carry adder (LCA) o carry-lookahead adder.

Un adder convencional, como un ripple-carry adder, genera el acarreo bit a bit, lo que puede ser lento cuando la palabra es extensa. El carry-lookahead adder es una solución que intenta abordar este problema, calculando los acarreos de manera más eficiente para mejorar la velocidad de operación. Este tipo de adder se basa en la idea de "mirar hacia adelante" para precalcular los acarreos antes de que se necesiten en el sumador.

Voy a mostrar un ejemplo simplificado para ilustrar cómo funciona el carry-lookahead adder. Supongamos que estamos sumando dos números de 4 bits en un procesador MIPS con un adder convencional y luego con un carry-lookahead adder.

Ejemplo con Adder Convencional:

$$\begin{array}{r} 1101 \text{ (13 en decimal)} \\ + 1011 \text{ (11 en decimal)} \\ \hline 10000 \end{array}$$

Aquí, cada bit en la suma depende del acarreo del bit anterior, lo que podría resultar en un retraso significativo si el ancho de palabra es grande.

Ejemplo con Carry-Lookahead Adder:

$$\begin{array}{r} 1101 \text{ (13 en decimal)} \\ + 1011 \text{ (11 en decimal)} \\ \hline 10000 \end{array}$$

En el carry-lookahead adder, los acarreos se calculan de antemano utilizando fórmulas específicas. Aunque los detalles precisos pueden ser más complejos, la idea general es que los acarreos se generan en paralelo, lo que reduce la dependencia secuencial entre los bits.

Este enfoque permite acelerar la generación de resultados y reducir los retrasos asociados con el acarreo en palabras extensas. Es importante tener en cuenta que en implementaciones reales, la lógica del carry-lookahead adder puede ser más complicada, y se pueden utilizar técnicas adicionales como el pipeline para mejorar aún más el rendimiento. Estas técnicas buscan aprovechar la parallelización y reducir los tiempos de propagación para las operaciones aritméticas.

El carry-lookahead adder mejora la velocidad de operación en comparación con un adder convencional al reducir la dependencia secuencial entre los bits y acelerar la generación de resultados. En un adder convencional, como el ripple-carry adder. Como cada suma depende del acarreo del bit anterior desemboca en un retraso significativo en la generación de resultados, especialmente para palabras de ancho grande.

Por otro lado, el carry-lookahead adder calcula los acarreos de manera más eficiente mediante el uso de fórmulas específicas que permiten precalcular los acarreos antes de que se necesiten en el sumador. Esta técnica reduce la dependencia secuencial entre los bits, ya que los acarreos se generan en paralelo, lo que significa que cada acarreo no depende del cálculo del acarreo anterior. Como resultado, se logra una mejora significativa en la velocidad de operación, ya que los cálculos se realizan de forma más eficiente y con menos retrasos.

En resumen, el carry-lookahead adder logra mejorar la velocidad de operación al minimizar la dependencia secuencial entre los bits y al calcular los acarreos de manera más eficiente, lo que reduce los retrasos asociados con el acarreo propagado en palabras de ancho grande.