

# Taller #1

Brayan Esteban Alpízar Elizondo  
Mariana Gonzalez Sanabria

Taller de Diseño Digital  
Instituto Tecnológico de Costa Rica

16 de agosto de 2025

## 1. Problema 2

Diseñe un restador completo de 4 bits con modelo de estructura en VHDL. Parta del diseño un restador completo de 1 bit.

### 1. Propuesta 1

La primera propuesta consiste en implementar el restador de 4 bits a partir de un *full adder* (sumador completo) como bloque base [1]. Este método se fundamenta en el principio de que una operación de resta puede transformarse en una operación de suma si se utiliza el complemento a 2 del sustraendo. Para lograrlo, cada bit del segundo operando se invierte y se introduce un *carry in* inicial con valor lógico '1'. De esta forma, el circuito del sumador completo puede emplearse directamente para realizar restas, sin modificar su lógica interna fundamental. El proceso de diseño inicia con la creación de un módulo de restador de 1 bit basado en un *full adder*, el cual recibe como entradas un bit del minuendo, un bit del sustraendo y el bit de acarreo/préstamo de la etapa anterior. Sus salidas corresponden al bit de diferencia y al acarreo/préstamo hacia la siguiente etapa. Una vez validado el módulo de 1 bit, se interconectan cuatro instancias idénticas en cascada para formar el restador completo de 4 bits.

Este diseño presenta algunas ventajas como la modularidad que permite trabajar con bloques pequeños y reutilizables, lo que facilita la depuración y el mantenimiento del código VHDL. Otra ventaja es la escalabilidad ya que aumentar el número de bits del restador implica únicamente replicar el módulo base y ajustar la conexión de los acarreos, sin necesidad de rediseñar la lógica interna. Además ya que si se dispone de un *full adder* previamente verificado, se reduce el riesgo de errores y se optimiza el tiempo de desarrollo. La verificación también se simplifica, puesto que validar un módulo elemental asegura que su comportamiento se mantendrá en el diseño completo.

### 2. Propuesta 2

La segunda propuesta plantea la implementación directa de un restador de 1 bit a partir de las ecuaciones lógicas mínimas de la operación de resta. En este diseño, se utilizan compuertas XOR, AND y OR para generar las señales de diferencia y de préstamo (*borrow*) sin depender de un sumador completo. Las ecuaciones booleanas utilizadas pueden expresarse como:

$$D = A \oplus B \oplus Bin$$

$$Bout = (\overline{A} \cdot B) + (Bin \cdot \overline{A \oplus B})$$

donde  $D$  representa el bit de diferencia,  $Bout$  es el préstamo de salida,  $A$  y  $B$  son los bits de entrada, y  $Bin$  es el préstamo de entrada. Una vez diseñado este módulo de 1 bit, se interconectan cuatro instancias para obtener el restador de 4 bits.

Aunque esta propuesta permite un control más directo sobre la cantidad y tipo de compuertas utilizadas, lo que en algunos casos puede reducir el retardo de propagación o el consumo de recursos lógicos, también presenta algunos problemas. El diseño y verificación resultan más complejos, ya que no se parte de un bloque previamente validado. Además que la reutilización es limitada, pues las ecuaciones están adaptadas a un fin específico, y cualquier modificación del comportamiento requiere revisar y ajustar la lógica. Si bien es posible replicar este módulo para aumentar el número de bits, no es tan escalable y es menos eficiente que en la primera propuesta.

### 3. Diseño Elegido

Tras analizar ambas alternativas de diseño, creamos la primera propuesta como la más adecuada. Su estructura modular basada en un *full adder* es mas util en cuanto a la reutilización, la escalabilidad y la facilidad de mantenimiento, permitiendo construir un sistema mas robusto a partir de componentes probados y con un menor riesgo de errores.

## Referencias

- [1] S. L. Harris y D. M. Harris, *Digital Design and Computer Architecture: ARM Edition*, 2nd. Cambridge, MA: Morgan Kaufmann, 2021.