

Diseños para resolver problema 1 laboratorio 2

Emanuel Chavarría Hernández
Fernando Andrés Fuchs Mora

CE 3201 - Taller de diseño digital
Instituto Tecnológico de Costa Rica

4 de septiembre de 2025

1. Diseños

1.1. Pregunta 1

El sistema diseñado para esta sección tiene como objetivo ser una ALU parametrizable de N bits, en este caso una calculadora con las siguientes operaciones disponibles: suma, resta, multiplicación, división, módulo, and, or, xor, shift left y shift right. Además se deben de realizar las banderas de estados de la ALU como lo son: Negativo (N), Zero (Z), Acarreo (C) y Desbordamiento (V). Diseñar la solución de este sistema tiene cierto grado de complejidad ya que para tres operaciones aritméticas no se puede utilizar el símbolo de operador que brinda SystemVerilog.

Propuestas

- ✓ **Propuesta A para el sumador:** Ripple Carry. Esta alternativa utiliza una cadena de full-adders de un bit a base de compuertas lógicas como lo son AND, OR, XOR y luego estos se conectan en cascada que instancia el full-adder N veces para poder hacerlo parametrizable. En el caso del full-adder se tienen 3 entradas A, B, Cin y en base a ellas se tienen las siguientes ecuaciones:

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = (A \cdot B) + (A \cdot C_{in}) + (B \cdot C_{in})$$

En base a las ecuaciones mostradas anteriormente se tiene la siguiente tabla de verdad:

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

En la tabla mostrada anteriormente cada fila representa el comportamiento de un full-adder de un solo bit. Sin embargo al conectar N de estos bloques se obtiene un sumador ripple-carry de N bits. Se seleccionó esta propuesta de diseño ya que se trata de una arquitectura simple y que requiere un mínimo de compuertas lógicas, por lo tanto utiliza menos ALMs y menos conmutación en la FPGA, lo que indica un menor consumo de recursos y de energía. Si bien es cierto este diseño tiene una desventaja como lo es que el retardo crece de forma lineal con N, para este laboratorio no es necesario utilizar un N muy grande para comprobar el buen funcionamiento. En conclusión esta propuesta se ajusta de la mejor manera a los requerimientos del laboratorio.

- ✓ **Propuesta B para el sumador:** Carry Select Adder. Esta alternativa es un sumador rápido que acelera la suma binaria procesando múltiples valores posibles de acarreo en paralelo y seleccionando el resultado correcto una vez conocido el acarreo real. Esto se logra utilizando múltiples sumadores de ripple carry para realizar el cálculo dos veces simultáneamente. Esta opción fue descartada porque representa una complejidad innecesaria ya que se necesita duplicar los bloques de full-adders e implementar lógica extra de multiplexores para seleccionar el resultado completo. Este diseño es muy complejo para valores de N pequeños para cuatro u ocho bits que pueden ser el rango de operación del laboratorio. El beneficio de la velocidad no compensa el desarrollo y complejidad que se requiere para lograr el carry-select-adder por lo que no resulta adecuado para los requerimientos del laboratorio.
- ✓ **Propuesta A para el restador:** Two's Complement Subtractor. Esta alternativa utiliza los mismos principios de la suma binaria para calcular la resta, implementando un sistema llamado complemento a dos. Esta expresión se puede interpretar de la siguiente manera:

$$A - B = A + (-B) = A + \overline{B} + 1$$

Se puede observar en esta expresión como se establece que los enteros negativos se pueden representar tomando su valor binario positivo invirtiendo todos los bits y sumando uno. Se tiene la siguiente tabla de verdad para la propuesta de diseño:

A	B	Bin	DIFF	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Esta tabla de verdad corresponde a un restador de 1 bit sin embargo, el restador parametrizable de N bits se construye encadenando N celdas de 1 bit propagando el borrow entre ellas. Este diseño es adecuado para el laboratorio por su simplicidad, reutilización de módulos (se puede utilizar el código de sumador parametrizable). Además este método es un estándar universal en procesadores y ALUs modernas.

- ✓ **Propuesta B para el restador:** Borrow Lookahead Subtractor (BLS). Esta alternativa realiza restas binarias a alta velocidad mediante una lógica paralela para generar señales de préstamo, en lugar de propagarse secuencialmente bit a bit. Este método minimiza los retardos de propagación, lo que aumenta significativamente la velocidad de las operaciones de resta, especialmente en aplicaciones multibit como la ALU. Esta propuesta fue rechazada ya que incrementa significativamente la complejidad y el uso de recursos en la FPGA, sin aportar beneficios relevantes para anchos pequeños
- ✓ **Propuesta A para la multiplicación:** Multiplicación por suma de productos parciales. Esta alternativa es un circuito combinacional donde cada bit del multiplicando se multiplica por cada bit del multiplicador para generar productos parciales individuales, que luego se suman usando sumadores y desplazamientos para producir el resultado final. Para poder crearlo se utilizan compuertas AND para los productos parciales y full-adders para la suma. Se tienen las siguientes ecuaciones para el diseño:

$$P = \sum_{i=0}^{N-1} (b_i \cdot A) 2^i$$

$$pp_{i,j} = a_j \cdot b_i \quad (0 \leq i, j \leq N-1)$$

Dada la ecuación anterior se puede demostrar la siguiente tabla de verdad:

a_1	a_0	b_1	b_0	p_3	p_2	p_1	p_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	1	1	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1

Esta tabla es una multiplicación completa de dos operandos de 2 bits. Sin embargo para un N parametrizable la idea es la misma, el arreglo escala repitiendo el patrón de AND + sumadores por columnas. Este diseño fue seleccionado para implementarlo ya que es un diseño estándar y además se puede reutilizar el código ya implementado de los full-adders.

- ✓ **Propuesta B para la multiplicación:** Multiplicador con codificación de Booth y árbol de reducción (Wallace/Dadda). Esta alternativa es un algoritmo para multiplicar eficientemente números binarios con signo que están en complemento a dos. Está basado en un método para sumar o restar el multiplicando al producto parcial, dependiendo de los bits del multiplicador y luego desplazar el producto, reduciendo el número de sumas y restas necesarias en comparación con la multiplicación binaria estándar. Esta propuesta es rechazada ya que requiere una lógica adicional de recodificación con CSA, multiplexores y varias capas de sumadores. Además este diseño tiene una mejor eficiencia cuando el N es mediano o grande, por lo tanto no se adapta a los requerimientos del laboratorio.