
	UNIVERSIDAD DE SANTANDER – UDES	Página 1 de 2	
	ACTIVIDAD	Versión: 01	
	PROGRAMA INGENIERÍA DE SOFTWARE	Semestre B-2024	

Nombre: Santiago Alexander Ospina Pabón

DESARROLLE LOS CONCEPTOS DE SUMA BINARIA Y DEZPLAZAMIENTO A TRAVES DE EJERCICIOS USANDO COMPUERTAS LÓGICAS COMO LAS VISTAS EN CLASE.

1. Suma binaria

La suma binaria es una operación fundamental en sistemas digitales. Para realizarla, se usa un circuito conocido como Sumador Completo (Full Adder), que suma tres bits: dos bits de entrada y un bit de acarreo de una posición anterior.

Ejemplo: Suma de dos bits con acarreo

Para sumar dos bits A y B y tener en cuenta un acarreo de entrada C_{in} , necesitamos:



- **Bit de suma (S):** El resultado de la suma de los bits A , B , y C_{in}
- **Acarreo de salida (C_{out}):** El acarreo que se lleva a la siguiente posición.

La tabla de verdad para un Sumador Completo es la siguiente:

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

Circuito de suma usando compuertas:

- Para el bit de suma (S): Se utiliza una compuerta XOR que calcula $S = A \oplus B \oplus C_{in}$
- Para el **acarreo de salida** (C_{out}): Se utiliza una combinación de compuertas AND y OR para calcular $C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B))$

	UNIVERSIDAD DE SANTANDER – UDES	Página 2 de 2	
	ACTIVIDAD	Versión: 01	
	PROGRAMA INGENIERÍA DE SOFTWARE	Semestre B-2024	

Ejercicio práctico: Supongamos que queremos sumar $A = 1$, $B = 1$ y $C_{in} = 1$

- Primero, calculamos el bit de suma $S = 1 \oplus 1 \oplus 1 = 1$
- Luego, calculamos el acarreo de salida $C_{out} = (1 \cdot 1) + (1 \cdot (1 \oplus 1)) = 1$

Resultado: $S = 1$ y $C_{out} = 1$, lo que significa que la suma total es 11_2 (3 en decimal).

2. Desplazamiento usando Compuertas Lógicas

El desplazamiento en sistemas digitales puede ser hacia la izquierda o derecha, y existen dos tipos principales:

- **Desplazamiento lógico:** Rellena con ceros los espacios vacíos.
- **Desplazamiento aritmético:** Conserva el bit de signo en desplazamientos hacia la derecha (para números con signo).

Ejemplo: Desplazamiento lógico a la izquierda

Para realizar un desplazamiento lógico a la izquierda, cada bit en un número binario se mueve una posición hacia la izquierda, y el bit vacío se llena con un 0. Este desplazamiento equivale a multiplicar el número binario por 2.

Ejercicio práctico: Supongamos que tenemos el número binario 101010101010 (10 en decimal) y lo desplazamos a la izquierda una posición:

1. El número original es 101010101010.
2. Después del desplazamiento a la izquierda, obtenemos 101001010010100 (20 en decimal).

Implementación con compuertas:

Un desplazamiento lógico a la izquierda se puede implementar mediante conexiones y compuertas:

- **Compuerta AND:** Para eliminar el último bit.
- **Compuerta OR:** Si se desea combinar valores de dos bits de entrada en el nuevo bit de salida.