Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería



**CIRCUITOS DIGITALES AVANZADOS**

**Practica 8**

**Unidad Aritmética y Lógica en Dispositivos Programables**

**Docente:** Lara Camacho Evangelina

**Alumnos:** Gómez Cárdenas Emmanuel Alberto **1261509**

# Contenido

[Objetivo 2](#_Toc73033868)

[Equipo 2](#_Toc73033869)

[Fundamento teórico 2](#_Toc73033870)

[Desarrollo 3](#_Toc73033871)

[ALU 6](#_Toc73033872)

[CONCLUSIONES 8](#_Toc73033873)

# Objetivo

Simular una Unidad Aritmética y Lógica en dispositivos programables FPGA.

# Equipo

Computadora con el IDE Xilinx Vivado u otro software para desarrollo de código para FPGAs.

# Fundamento teórico

Una Unidad Aritmética y Lógica (ALU) es un circuito combinacional que realiza operaciones aritméticas y lógicas sobre números enteros, denominados operandos. La ALU es uno de los componentes básicos de los procesadores.

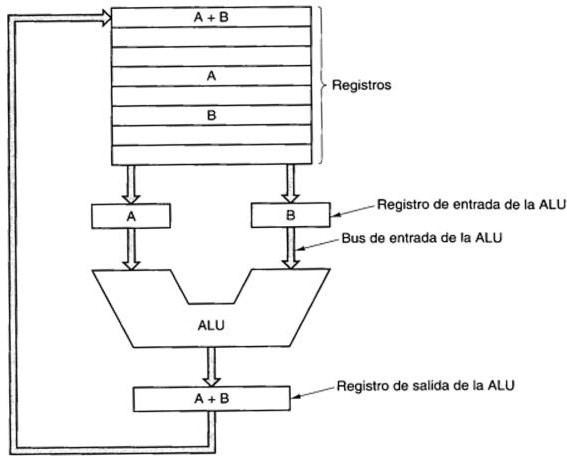


Figura 1. ALU.

Una ALU recibe las siguientes entradas:

* Operandos.
* Acarreo de entrada.
* Función. Este es un código que permite seleccionar la operación que va a realizar la ALU. La cantidad de bits de función determina el número máximo de operaciones diferentes que la ALU puede realizar. Si los bits de función son N, la ALU puede realizar máximo 2N operaciones diferentes.

Una ALU produce las siguientes salidas:

* Resultado de la operación.
* Bits de estado del resultado. También denominados banderas, estos bits son señales individuales del estado del resultado de la última operación realizada por la ALU. Algunos de ellos son: acarreo de salida, sobreflujo, signo, cero y paridad. Cuando la condición asociada a la bandera ocurre, el bit de bandera se activa. Por ejemplo, si existe un sobreflujo después de una operación aritmética, la bandera de sobreflujo se activa, de lo contrario se desactiva.

# Desarrollo

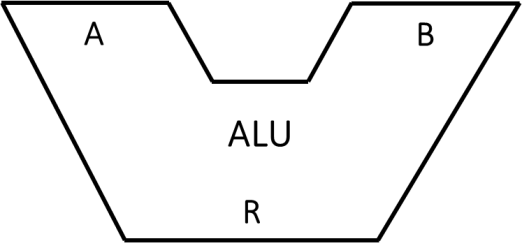
* Revise los siguientes videos sobre los componentes de un sistema computarizado. Favor de revisarlos en el orden indicado:
  1. **ALU**: https://youtu.be/1I5ZMmrOfnA
  2. **Registros y memoria**: https://youtu.be/fpnE6UAfbtU
  3. **CPU**: https://youtu.be/FZGugFqdr60
* Cree en Vivado un nuevo proyecto de código VHDL llamado ALU donde implemente una ALU para operandos de 32 bits.

Figura 2. Símbolo de la ALU. La ALU debe tener las siguientes funciones:

|  |  |  |
| --- | --- | --- |
| **Función** | **Operación** | **Descripción** |
| 0 | AND A,B | 𝑅 = 𝐴 𝖠 𝐵 |
| 1 | OR A,B | 𝑅 = 𝐴 ∨ 𝐵 |
| 2 | XOR A,B | 𝑅 = 𝐴 ⊕ 𝐵 |
| 3 | NOT A | 𝑅 = 𝐴 |
| 4 | ADD A,B | 𝑅 = 𝐴 + 𝐵 |
| 5 | ADC A,B | 𝑅 = 𝐴 + 𝐵 + 𝐶𝑖𝑛 |
| 6 | SUB B,A | 𝑅 = 𝐵 − 𝐴 |
| 7 | SBB B,A | 𝑅 = 𝐵 − 𝐴 − 𝐶𝑖𝑛 |
| 8 | INC A | 𝑅 = 𝐴 + 1 |
| 9 | DEC A | 𝑅 = 𝐴 − 1 |
| A | SHL A | **Corrimiento a la izquierda de A**. Desplaza los bits de **A** una posición a la izquierda y coloca un 0 en la posición menos significativa. El anterior bit más significativo se descarta de **A** y se coloca en **Cout**.  *Ejemplo*:  Sea **A** = 1010 1101.  Se realiza **SHL A**. El resultado es:  **A** = 0101 1010.  **Cout** = 1. |

|  |  |  |
| --- | --- | --- |
| B | SHR A | **Corrimiento a la derecha de A**. Desplaza los bits de **A** una posición a la derecha y coloca un 0 en la posición más significativa. El anterior bit menos significativo se descarta de **A** y se coloca en **Cout**.  *Ejemplo*:  Sea **A** = 1010 1101.  Se realiza **SHR A**. El resultado es: **A** = 0101 0110.  **Cout** = 1. |
| C | ROL A | **Rotación a la izquierda de A**. Desplaza los bits de **A** una posición a la izquierda y coloca el bit más significativo en la posición menos significativa. El bit más significativo se copia a **Cout**.  Puede considerarse como un *corrimiento circular a la izquierda*, donde en lugar de insertar siempre un cero, se inserta el bit **más significativo**.  *Ejemplo*:  Sea **A** = 1010 1101.  Se realiza **ROL A**. El resultado es: **A** = 0101 1011.  **Cout** = 1. |
| D | ROR A | **Rotación a la derecha de A**. Desplaza los bits de **A** una posición a la derecha y coloca el bit menos significativo en la posición más significativa. El bit menos significativo se copia a **Cout**.  Puede considerarse como un *corrimiento circular a la derecha*, donde en lugar de insertar siempre un cero, se inserta el bit **menos significativo**.  *Ejemplo*:  Sea **A** = 1010 1101.  Se realiza **ROR A**. El resultado es: **A** = 1101 0110.  **Cout** = 1. |

Tabla 1. Funciones de la ALU. La ALU debe contar con los siguientes bits de estado:

**Cin:** Acarreo de entrada.

**Cout:** Acarreo de salida.

**Z: Cero.** Se activa si el resultado es cero.

**Ov:** Sobreflujo

**P**: Paridad. Se activa si el resultado tiene un número par de bits activos.

Después de realizar una operación aritmética, los bits **Cout**, **Z**, **Ov** y **P** son actualizados para reflejar el estado de **R**.

Después de realizar una operación lógica, **Z** y **P** son actualizados para reflejar el estado de **R**. Los bits **Cout** y **Ov** son puestos en cero.

# ALU

# 

# ALU\_tb

# 

# CONCLUSIONES

La ALU es de los componentes más importantes de un procesador ya que realiza todas las operaciones aritméticas y lógicas, después de esta práctica logramos aplicar nuestros conocimientos sobre la ALU y lo que hemos aprendido sobre VHDL.