# Trabajo Práctico CLP

Contenido

[Trabajo Práctico CLP 1](#_Toc210935797)

[1. Introducción 1](#_Toc210935798)

[2. Implementación 1](#_Toc210935799)

[2. Diagrama de Bloques 2](#_Toc210935800)

[3. Simulaciones 3](#_Toc210935801)

[4. Tabla de uso de recursos 5](#_Toc210935802)

[5. Limitaciones 6](#_Toc210935803)

## 1. Introducción

El objetivo de este trabajo fue implementar un sistema de comunicación UART con una Unidad Aritmeticológica (ALU) de 4 bits en una FPGA utilizando VHDL y el entorno de desarrollo Vivado.

El sistema permite recibir datos a través de la UART, procesarlos mediante la ALU según la operación seleccionada, y visualizar el resultado en los 4 LEDs de la FPGA.

Se integró un módulo Virtual I/O (VIO) para permitir la verificación y prueba remota del diseño en el hardware.

## 2. Implementación

El sistema está compuesto por los siguientes módulos principales:

* **UART\_RX:** encargado de recibir los datos seriales enviados por el usuario.
* **ALU:** realiza las operaciones aritméticas y lógicas entre dos operandos de 4 bits, con soporte para suma, resta, AND, OR, XOR, NOT, desplazamiento a la izquierda y a la derecha.
* **UART\_ALU\_TOP:** módulo superior que integra la recepción de datos vía UART, el control de flujo y la ALU.
* **VIO:** módulo de depuración que permite controlar las señales de entrada (reset, RXD) y observar las salidas (LEDs) directamente desde Vivado Hardware Manager.

El flujo de operación inicia con la recepción de **tres** bytes por UART: los operandos A, B y el código de operación Op. Una vez recibidos, se calcula el resultado y se muestra en los 4 LEDs de la placa. La ALU permite las siguientes operaciones:

|  |  |
| --- | --- |
| Operación | Código Op |
| + | 0 |
| - | 1 |
| AND | 2 |
| OR | 3 |
| XOR | 4 |
| NOT | 5 |
| << | 6 |
| >> | 7 |

## 2. Diagrama de Bloques

Diagrama de bloques de la implementación:

A screenshot of a computer

AI-generated content may be incorrect.

Esquemático generado por el entorno de desarrollo Vivado incluyendo el módulo VIO:

A diagram of a computer

AI-generated content may be incorrect.

## 3. Simulaciones

Simulación *de uart\_alu\_top* para probar la operación de suma:

* 20us: señal de RESET
* 94us: se recibe el primer byte por la UART con valor ‘3’ y se almacena en *A\_buf*
* 180us: se recibe el segundo byte por la UART con valor ‘5’ en *B\_buf*
* 267us: se recibe el tercer byte por la UART con valor ‘0’ y se hace *alu\_ready* = 1 para que la ALU opere.
* La ALU opera con A = 3, B = 5, Op = 0 (suma), Result = 8
* El resultado ‘8’ se refleja en los *leds*

A screenshot of a computer program

AI-generated content may be incorrect.

Simulación *de uart\_alu\_top* para probar la operación de resta:

* 454us: se recibe el primer byte por la UART con valor ‘7’ y se almacena en *A\_buf*
* 541us: se recibe el segundo byte por la UART con valor ‘2’ en *B\_buf*
* 628us: se recibe el tercer byte por la UART con valor ‘1’ y se hace *alu\_ready* = 1 para que la ALU opere.
* La ALU opera con A = 7, B = 2, Op = 1 (resta), Result = 2
* El resultado ‘2’ se refleja en los *leds*

A screenshot of a computer

AI-generated content may be incorrect.

Simulación de *uart\_alu\_top* para probar la operación de AND:

* 815us: se recibe el primer byte por la UART con valor ‘9’ y se almacena en *A\_buf*
* 902us: se recibe el segundo byte por la UART con valor ‘3’
* en *B\_buf*
* 988us: se recibe el tercer byte por la UART con valor ‘1’ y se hace *alu\_ready* = 1 para que la ALU opere.
* La ALU opera con A = 9(1001), B = 3(0011), Op = 1(0001), Result = 1
* El resultado ‘1’ se refleja en los *leds*

A screenshot of a computer

AI-generated content may be incorrect.

## 4. Tabla de uso de recursos

La siguiente tabla muestra los recursos utilizados en la FPGA de la tarjeta Arty Z7-10:

A screenshot of a computer

AI-generated content may be incorrect.

## 5. Limitaciones

El sistema es una implementación básica de una ALU con propósitos educativos y enfocados en reforzar los conocimientos adquiridos en VHDL y programación de FPGA durante el curso de Circuitos Lógicos Prgramables. A pesar de cumplir con los objetivos propuestos, el sistema presenta algunas limitaciones que podrían abordarse en versiones futuras:

* Ancho de palabra fijo: La ALU implementada opera únicamente con operandos de 4 bits, lo que restringe el rango de valores procesables. Una ampliación a 8 o 16 bits permitiría realizar operaciones más representativas.
* Sin control de errores en UART: La comunicación UART no incluye verificación de paridad ni manejo de errores de recepción, por lo que bytes corruptos podrían afectar el resultado de la operación.
* Latencia de procesamiento: La sincronización entre la recepción de datos y la ejecución de la operación depende del reloj del sistema y puede introducir un pequeño retardo, perceptible en configuraciones de mayor frecuencia o ancho de datos.
* Interfaz limitada: El sistema actual muestra el resultado únicamente mediante los Leds, lo cual es útil para pruebas básicas, pero poco práctico para aplicaciones complejas.
* Ausencia de retroalimentación al transmisor: El sistema no envía información de confirmación o resultado por UART, limitando su uso a entornos unidireccionales de prueba.