

Trabajo Práctico CLP

Contenido

Trabajo Práctico CLP	1
1. Introducción	1
2. Implementación.....	1
2. Diagrama de Bloques	2
3. Simulaciones.....	3
4. Tabla de uso de recursos.....	5
5. Limitaciones	6

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.

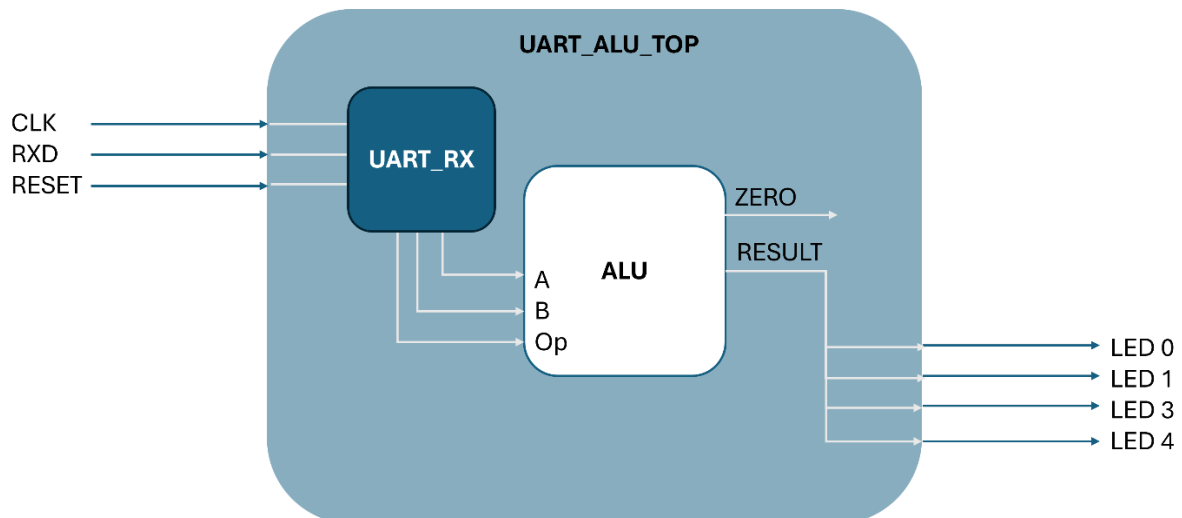
- **UART_ALU_TOP**: módulo superior que integra la recepción de datos vía UART, el control de flujo y la ALU.
- **VIO**: módulo de depuración que permite controlar las señales de entrada (reset, RXD) y observar las salidas (LEDs) directamente desde Vivado Hardware Manager.

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:

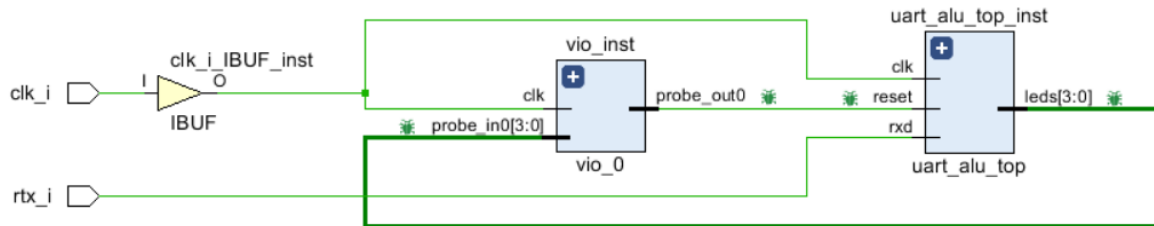
Operación	Código Op
+	0
-	1
AND	2
OR	3
XOR	4
NOT	5
<<	6
>>	7

2. Diagrama de Bloques

Diagrama de bloques de la implementación:



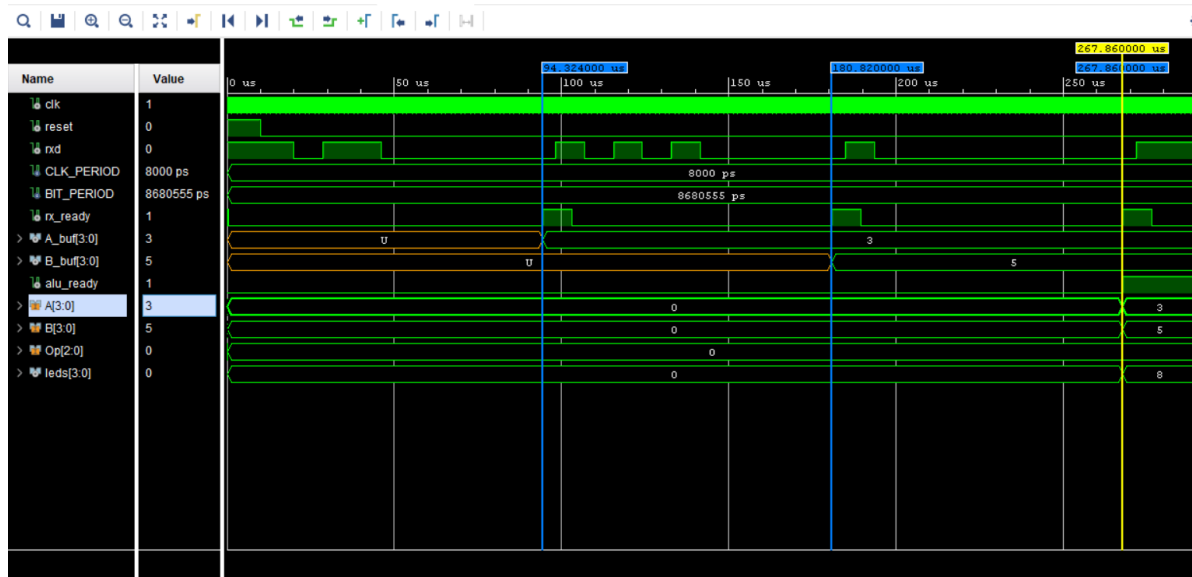
Esquemático generado por el entorno de desarrollo Vivado incluyendo el módulo VIO:



3. Simulaciones

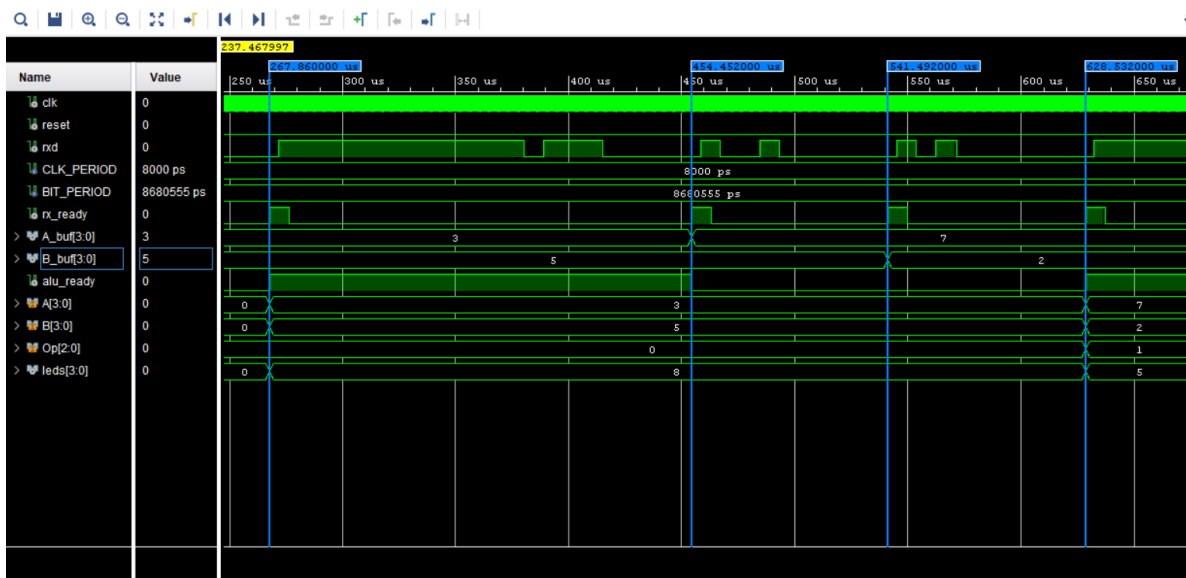
Simulación de `uart_alu_top` para probar la operación de suma:

- 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`



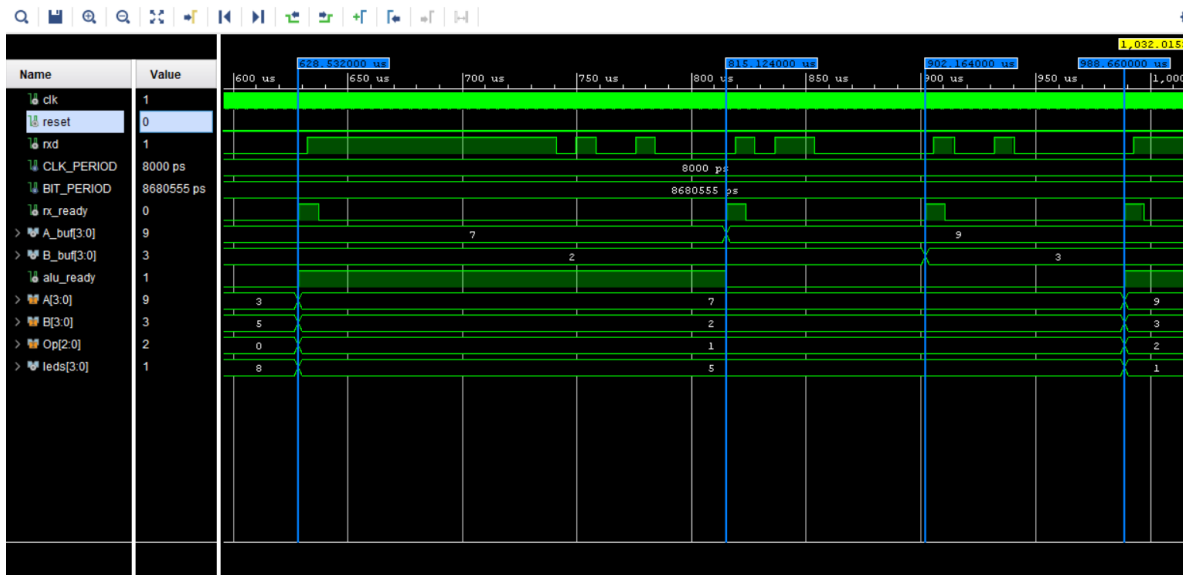
Simulación de *uart_alu_top* para probar la operación de resta:

- 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*



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*



4. Tabla de uso de recursos

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

Hierarchy					
Name	Slice LUTs (17600)	Slice Registers (35200)	F7 Muxes (8800)	Bonded IOB (100)	BUFGCTRL (32)
uart_alu_top_VIO	159	319	3	2	1
dbg_hub (dbg_hub_CV)	0	0	0	0	0
uart_alu_top_inst (uart_alu_top)	49	53	3	0	0
UartRX (uart_rx)	37	26	0	0	0
meta_harden_rxd_i0 (meta_harden)	0	2	0	0	0
uart_baud_gen_rx_i0 (uart_baud_gen)	8	8	0	0	0
uart_rx_ctl_i0 (uart_rx_ctl)	29	16	0	0	0
vio_inst (vio_0)	110	266	0	0	0
inst (vio_0_vio_v3_0_18_vio)	110	266	0	0	0
DECODER_INST (vio_0_vio_v3_0_18...	21	49	0	0	0
PROBE_IN_INST (vio_0_vio_v3_0_18...	10	34	0	0	0
PROBE_OUT_ALL_INST (vio_0_vio_v3...	2	6	0	0	0
G_PROBE_OUT[0].PROBE_OUT0_I...	1	2	0	0	0
U_XSDB_SLAVE (vio_0_xsdb_v1_0_...	77	161	0	0	0

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 Programables. 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.