



Diseño de una ALU

El objetivo de la práctica 2 es el diseño e implementación de una Unidad Aritmético-Lógica (ALU) con las siguientes características:

Entradas

- Una señal de reloj **clk** común a todos los componentes secuenciales del diseño.
- Una señal **reset** de reset asíncrono conectada a los componentes secuenciales indicados en el esquema RTL simplificado.
- Dos señales **A_in** y **B_in**, ambas de 8 bits, sobre las que se realizan las operaciones.
- Una señal **OP_in** de 5 bits que indica la operación que se realiza sobre los datos de entrada.

Salidas

- Las señales **A_out** y **B_out**, de 8 bits, conectadas a las salidas de los registros **regA** y **regB** respectivamente (ver figura del esquema RTL simplificado).
- La señal **TipoOP_out**, de 2 bits, con el valor en cada instante de la señal de salida del registro **regTipoOP**. La entrada de este registro es la señal interna **TipoOP** (ver figura esquema RTL simplificado). Esta señal, de 2 bits, indica el tipo de operación realizada, y tomará valor "01" si el valor de la señal interna **OP** corresponde a una operación aritmética, "10" si es lógica, "11" si es de comparación y "00" si el valor de **OP** no corresponde a ninguna operación implementada.
- La señal **LEDs**, de 8 bits, con el valor de la señal interna **SALIDA_ALU** (ver figura esquema RTL simplificado) si **TipoOP** indica que la operación realizada sobre las señales de entrada es Lógica o de Comparación; cero en otro caso.
- La señal **RESULTADO**, del tamaño adecuado, con el valor de la señal **SALIDA_ALU** si **TipoOP** indica que la operación realizada sobre las señales de entrada es Aritmética; cero en otro caso.
- La señal **CERO**, de 1 bit, que tomará valor '1' cuando el valor de la señal **SALIDA_ALU** sea cero; '0' en otro caso.
- La señal **SIGNO**, de 1 bit, que valdrá '1' si **TipoOP** indica que la operación realizada es aritmética y el valor de **SALIDA_ALU** es negativo, y '0' en otro caso (tanto si se trata de una operación aritmética que dé un resultado no negativo como si se trata de cualquier otro tipo de operación).

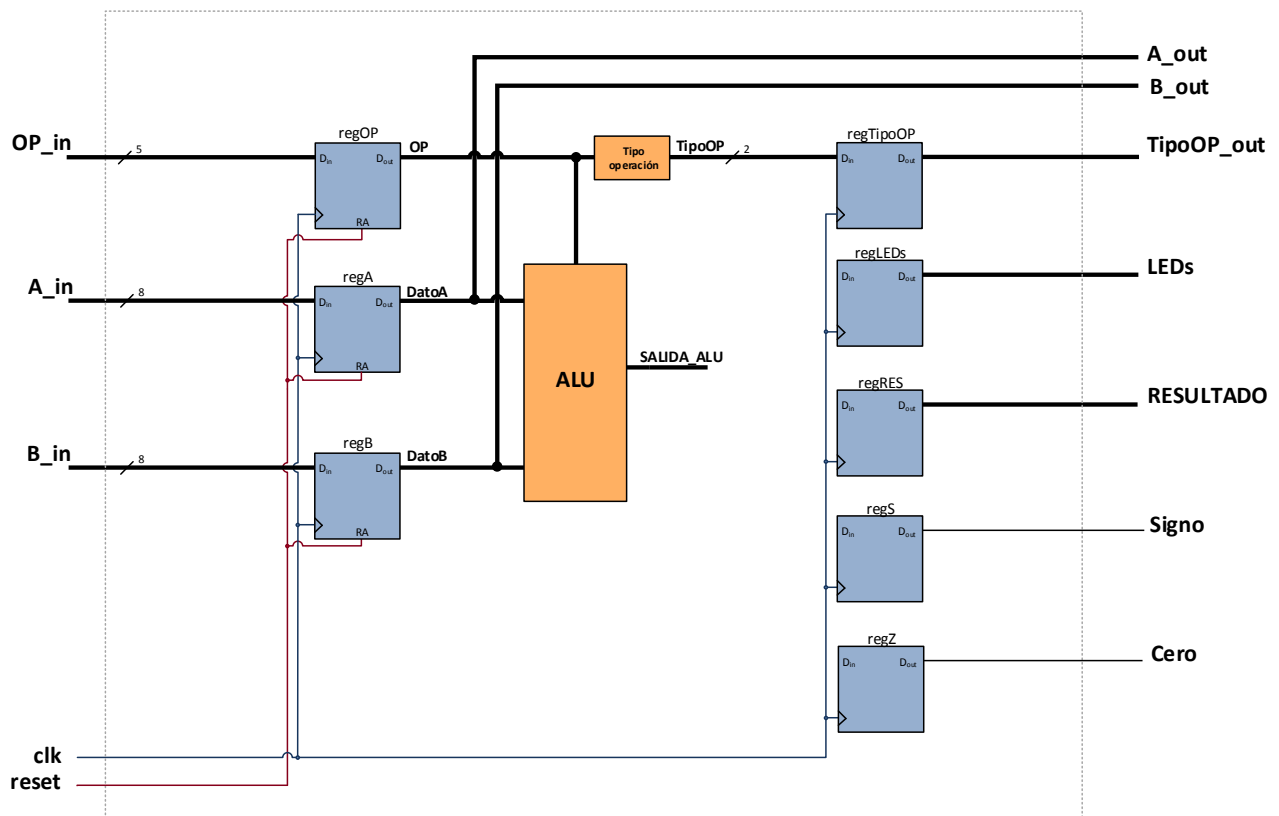


VHDL – LAB2

La ALU realiza las siguientes operaciones:

- Lógicas: OR, AND, XOR, NAND, NOT A, RR A (*Rotate Right*), RL A (*Rotate Left*) (las rotaciones son circulares).
- Aritméticas: $A+0$, $B+0$, $A+B$, $A-B$, $A+1$, $A-1$, $2*A$, $A/2$, $|A|$, $\text{MAX}(A,B)$, $\text{MIN}(A,B)$
- Comparación: $A < B$, $A > B$, $A = B$. En las operaciones de comparación, el bit menos significativo de **SALIDA_ALU** tomará valor '1' si la comparación es verdadera y '0' en otro caso; los restantes bits tomarán valor '0'.

Cada una de estas operaciones corresponderá a un valor determinado de la entrada **OP_in**. El código que corresponde a cada operación es fijado por el diseñador, teniendo en cuenta que el valor "00000" no está permitido.



Esquema RTL simplificado



VHDL – LAB2


















Verifica mediante simulación el funcionamiento realizando las operaciones con los datos y en los instantes de tiempo indicados en la tabla. Al iniciar la simulación, activa el reset asíncrono durante los primeros 85 ns y a continuación desactívalo. Considera una señal de reloj de 50 MHz.

t	A		B		Operación	TipoOP_out	LEDs		S	Z	RESULTADO
ns	DEC	binario	DEC	binario			binario	HEX			DEC
0	113	01110001	-44	11010100	A AND B						
200	113	01110001	-44	11010100	A OR B						
300	113	01110001	-44	11010100	A XOR B						
400	113	01110001	-44	11010100	A NAND B						
500	113	01110001	-44	11010100	NOT A						
600	113	01110001	-44	11010100	RR A						
700	113	01110001	-44	11010100	RL A						
800	113	01110001	55	00110111	A+0						
900	113	01110001	55	00110111	B+0						
1000	113	01110001	55	00110111	A+B						
1100	113	01110001	55	00110111	A-B						
1200	113	01110001	55	00110111	A+1						
1300	113	01110001	55	00110111	A-1						
1400	113	01110001	55	00110111	MAX(A,B)						
1500	113	01110001	55	00110111	MIN(A,B)						
1600	113	01110001	55	00110111	2*A						
1700	113	01110001	55	00110111	A/2						
1800	113	01110001	55	00110111	A						
1900	113	01110001	55	00110111	A<B						
2000	113	01110001	55	00110111	A>B						
2100	113	01110001	55	00110111	A=B						
2200	-15	11110001	55	00110111	A+0						
2300	-15	11110001	55	00110111	A+B						
2400	-15	11110001	55	00110111	A-B						
2500	-15	11110001	55	00110111	A+1						
2600	-15	11110001	55	00110111	A-1						
2700	-15	11110001	55	00110111	MAX(A,B)						
2800	-15	11110001	55	00110111	MIN(A,B)						
2900	-15	11110001	55	00110111	A<B						
3000	-15	11110001	55	00110111	A>B						
3100	-15	11110001	-15	11110001	A=B						
3200	-15	11110001	-15	11110001	2*A						
3300	-15	11110001	-15	11110001	A/2						
3400	-15	11110001	-15	11110001	A						
3500	-15	11110001	-15	11110001	A-B						
3600	-15	11110001	-15	11110001	A+B						
3700	-15	11110001	-15	11110001	RR A						
3800	-15	11110001	-15	11110001	RL A						
3900	-15	11110001	-15	11110001	A XOR B						



VHDL – LAB2

En el simulador, visualiza las señales de la siguiente manera:

-  a_in[7:0]
-  b_in[7:0]
-  op_in[4:0]
-  reset
-  clk
-  datoa[8:0]
-  datob[8:0]
-  salidaalu[8:0]
-  a_out[7:0]
-  b_out[7:0]
-  op[4:0]
-  tipoop[1:0]
-  tipoop_out[1:0]
-  leds[7:0]
-  resultado[8:0]
-  signo
-  cero