

## Diseño ALU 32 bits

Se diseña una ALU de un bit primero para luego instanciarla 32 veces para crear la ALU de 32 bits.

Primero se analizarán las funciones por aparte y luego se unirán para crear la ALU de 1 bit y se agregarán al final ciertos elementos para completar la ALU de 32 bits.

Para la función AND se necesita una compuerta AND que tome ambas entradas y realice la función de la siguiente forma:



Para la función OR se realiza el mismo procedimiento, es necesario únicamente la compuerta:



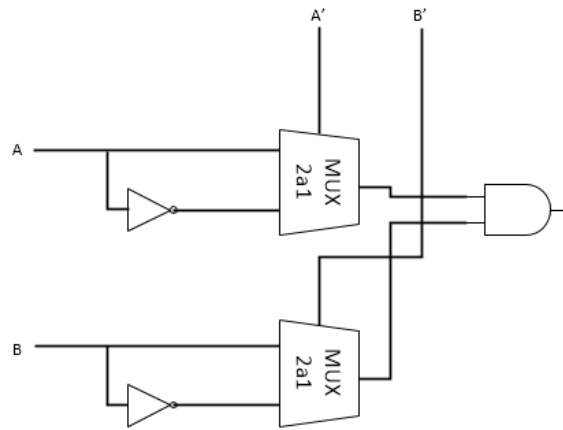
Para la función NOR se tomará la compuerta AND que se tiene y se invertirán ambas entradas para crear una NOR como se puede ver en la siguiente tabla:

NOR			AND con entradas negadas		
A	B	Out	A	B	Out
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

Por lo tanto se debe escoger si se niega la entrada o no, para esto se implementa un MUX de 2 a 1 que presenta la siguiente tabla:

A	B	Sel	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

y el circuito final se ve de la forma:



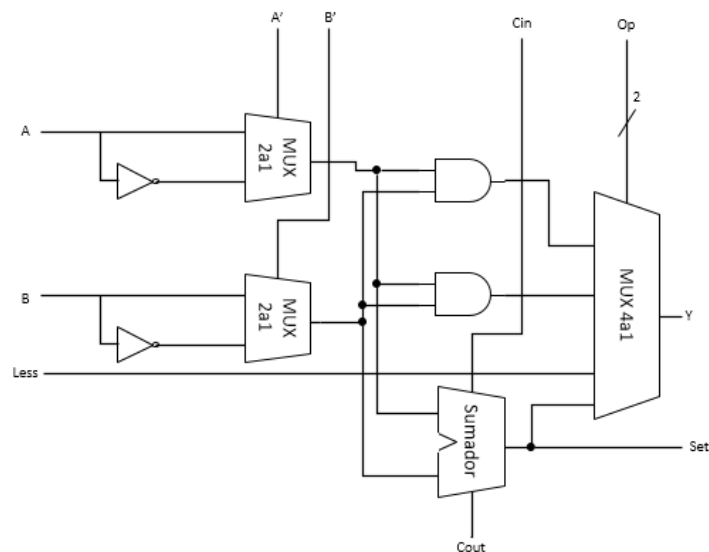
Aquí se agregan 2 señales de control ( $A'$  y  $B'$ ) para activar si se quiere la negación de las entradas y así se puede realizar ambas AND y NOR con una sola compuerta.

Para el add se utiliza un sumador que se presenta en la siguiente tabla:

A	B	Cin	Y	Cout
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

Por último para el Set Less Than se agrega una entrada Less y una salida Set que ayuda a obtener el signo de la operación del sumador, esto al instanciar las ALU's va a tomar más sentido.

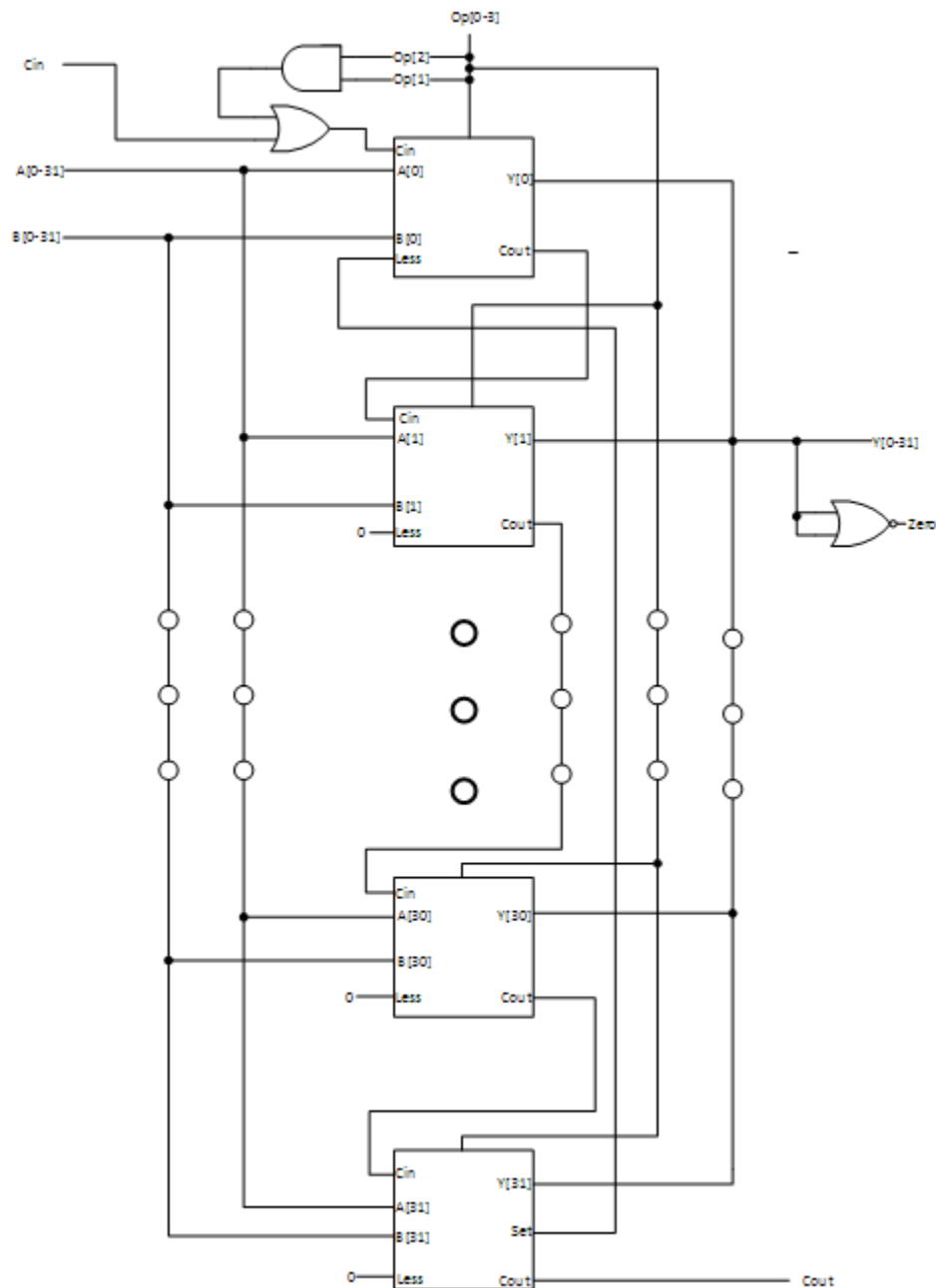
Entonces el diseño final de la ALU de 1 bit es el siguiente:



Como puede verse se utilizó un MUX de 4 a 1 para la elección de la operación, este MUX viene dado por la siguiente tabla:

A	B	C	D	S1	S2	Out
A	X	X	X	0	0	A
X	B	X	X	0	1	B
X	X	C	X	1	0	C
X	X	X	D	1	1	D

Ahora al unir las unidades de 1 bit se obtiene esto:



Aquí se evidencian los Set conectados a 0 y solo el del menos significativo se conecta al Set del más significativo, esto ya que para el Set Less Than se necesita obtener una respuesta de 0 o 1 en 32 bits si el número es menor, estos 0's dan 31 bits del dato y el importante es el signo de la operación para descubrir si es menor o no, por lo tanto este completa los 32 bits dando la respuesta.

También se agrega el módulo de Zero que consiste en pasar los 32 bits de la respuesta por una NOR.

Y por último se agrega una AND y una OR a la entrada Cin del menos significativo, esto es para el control de la operación subtract, Op[2] y Op[1] están activos cuando se da el subtract, y esto pone un carry in de 1 para sumar el uno a B' y obtener su complemento a 2.

Las señales de control se ven de la siguiente forma:

Op	A'	B'	S1	S2
And	0	0	0	0
Or	0	0	0	1
Add	0	0	1	0
Sub	0	1	1	0
SLT	0	1	1	1
Nor	1	1	0	0

Siendo A' y B' los selectores de la entrada para obtener el dato negado y S1 y S2 siendo los selectores del MUX de 4 a 1 para obtener la salida de la función requerida.