

Op. Code (Operation)	Instrução (Result)
0	AND (a,b)
1	OR (a,b)
2	NOT (a)
3	SOMA(a,b)

4. Teste a sua ULA de acordo com o seguinte roteiro:

Inicio:

A=0;

B=1;

AND(A,B);

A=1;

B=1;

OR(A,B);

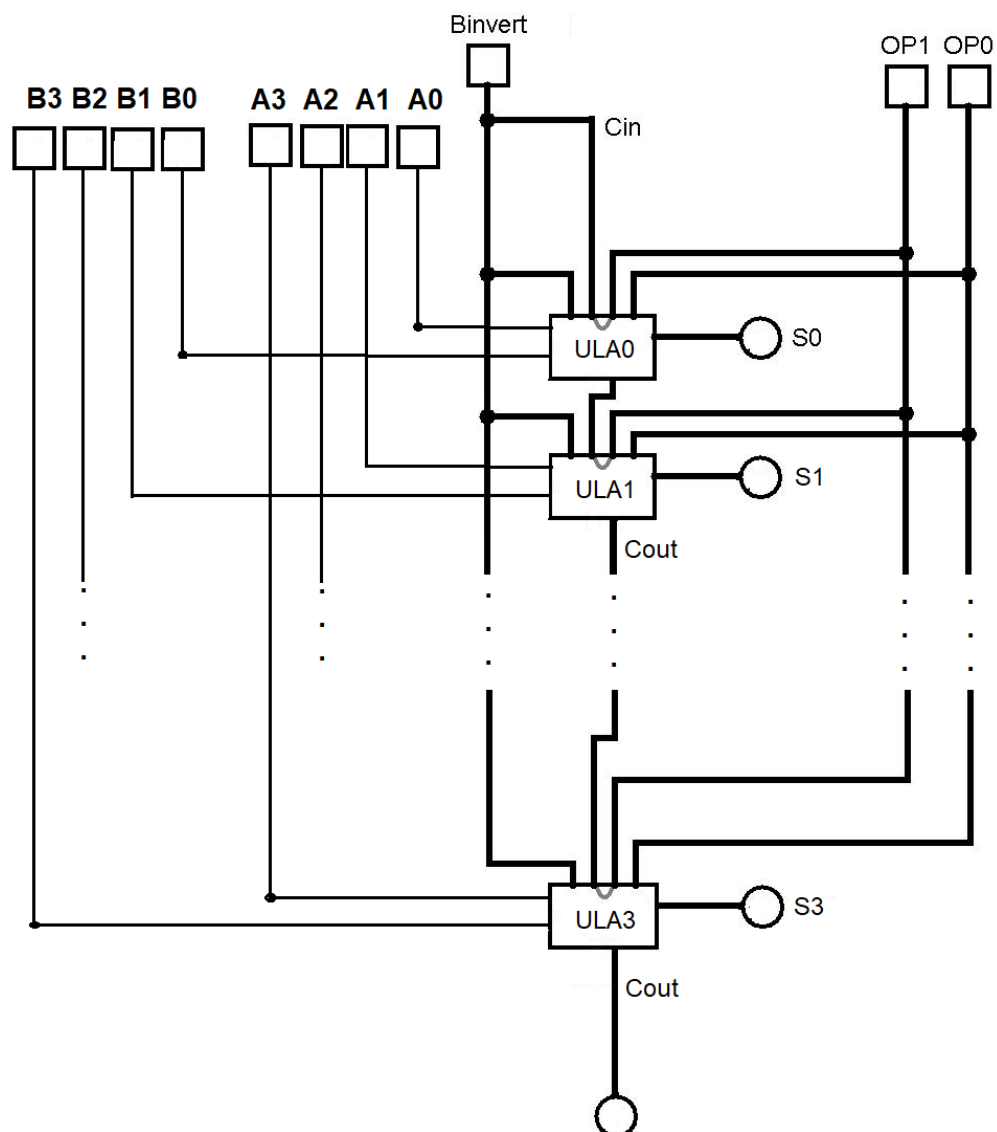
SOMA(A,B);

NOT(A);

SOMA (A,-B);

Fim.

5. Usando essa ula de 1 bit, construa essa ULA para **4 bits** no Logisim e verifique o seu funcionamento. Veja como funciona o barramento de instruções (operation) e o barramento de dados (a e b). Observe a ligação do Binvert ao Carry_in da primeira ULA. Procure usar subcircuitos, seu circuito deverá estar como a figura a seguir:



6. Teste a sua ULA de acordo com o seguinte roteiro (considerando os números de 4 bits):

Início:

A=2; (ou A=0010)
B=1; (ou B=0001)
AND(A,B);
B=3; (ou B=0011)
OR(A,B);
SOMA(A,B);
A=12; (ou A=1100)
NOT(A);
B=13; (ou B=1101)
AND(B,A);

Fim.

Para o programa de teste acima, preencher a tabela a seguir considerando que cada linha corresponderá à execução de uma instrução (a primeira linha já foi realizada, observe que a palavra deverá conter 10 bits, para escrevermos em hexa completamos os dois bits à esquerda com zero):

Instrução realizada	Binário (A,B,Op.code)	Valor em Hexa (0x ...)	Resultado em binário
AND(A,B)	0010 0001 00	(0000 1000 0100) = 0x084	0000
OR(A,B)			
SOMA(A,B)			
NOT(A)			
AND(B,A)			

O que apresentar nesse exercício:

Parte 1 (Logisim)

- Os prints de todos os circuitos elaborados (ULA de 1 bit e ULA de 4 bits).
- Preencher a tabela de resultados para a unidade de 4 bits.
- Um print de cada um dos testes realizados (use o roteiro indicado no item 6). Não é necessário dar prints quando no programa de teste apenas uma atribuição de valores às variáveis for realizada.

