

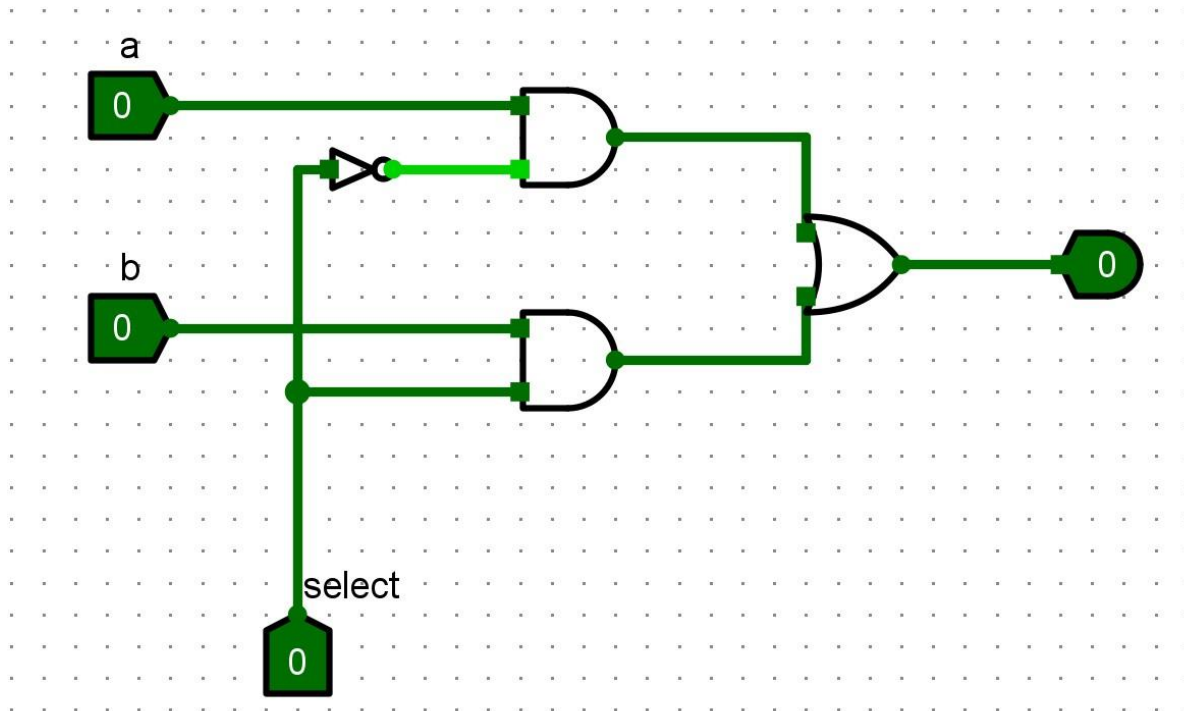
Arquitetura de Computadores 2 - ULA

Victor Souza Lima – 835287

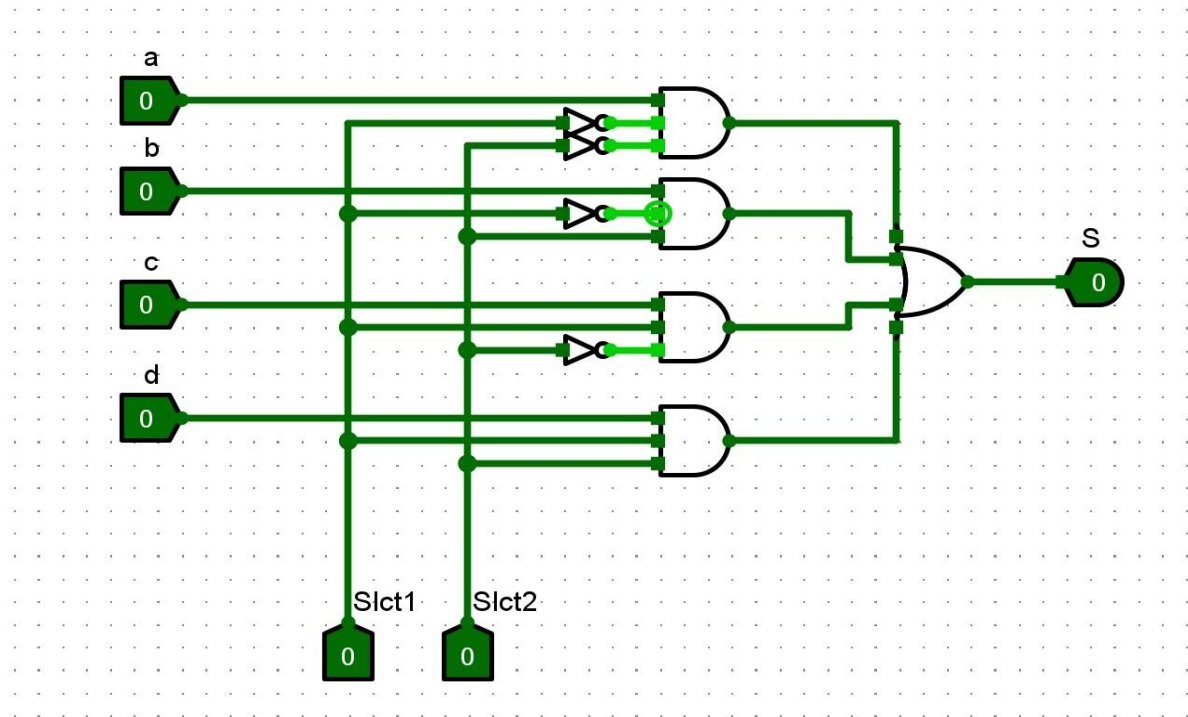
- PARTE 1

Circuitos utilizados:

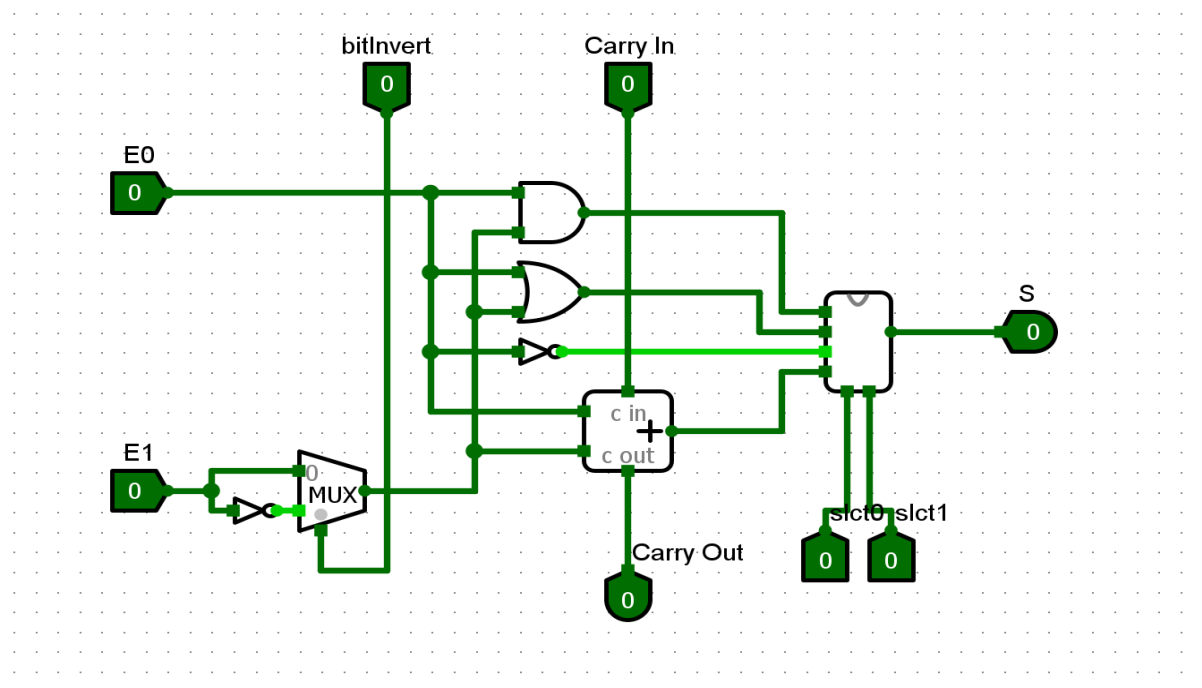
-Multiplexador de 1 seletor:



-Multiplexador de 2 seletores:



-Unidade Lógica e Aritmética (ULA) de 1 bit:



2) Procure entender o esquema, principalmente a subtração.

Explicação: Em nossa ULA, realizaremos o processo de subtração somando, uma vez que $A - B = A + (-B)$. Para isso, realizaremos o Complemento de 2 de uma entrada E:

Seja, $C2 = C1 + 1$; $C1 = E'$;

Logo, $C2 = E' + 1$

Portanto, caso queiramos subtrair um valor em nossa ULA, devemos barrar a entrada E e somar 1.

Essa tomada de decisão é realizada através de um multiplexador (MUX) de 1 seletor.

Seletor	Operação	Valor de E
0	Soma	E
1	Subtração	E'

O valor lógico do seletor está ligado ao Carry in do somador da ULA para reaproveitar a entrada deste. Uma vez que o MUX realiza o complemento de 1, basta apenas somar 1 à resultante.

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

Início:

A=0;

B=1;

AND(A,B);

A=1;

B=1;

OR(A,B);

SOMA(A,B);

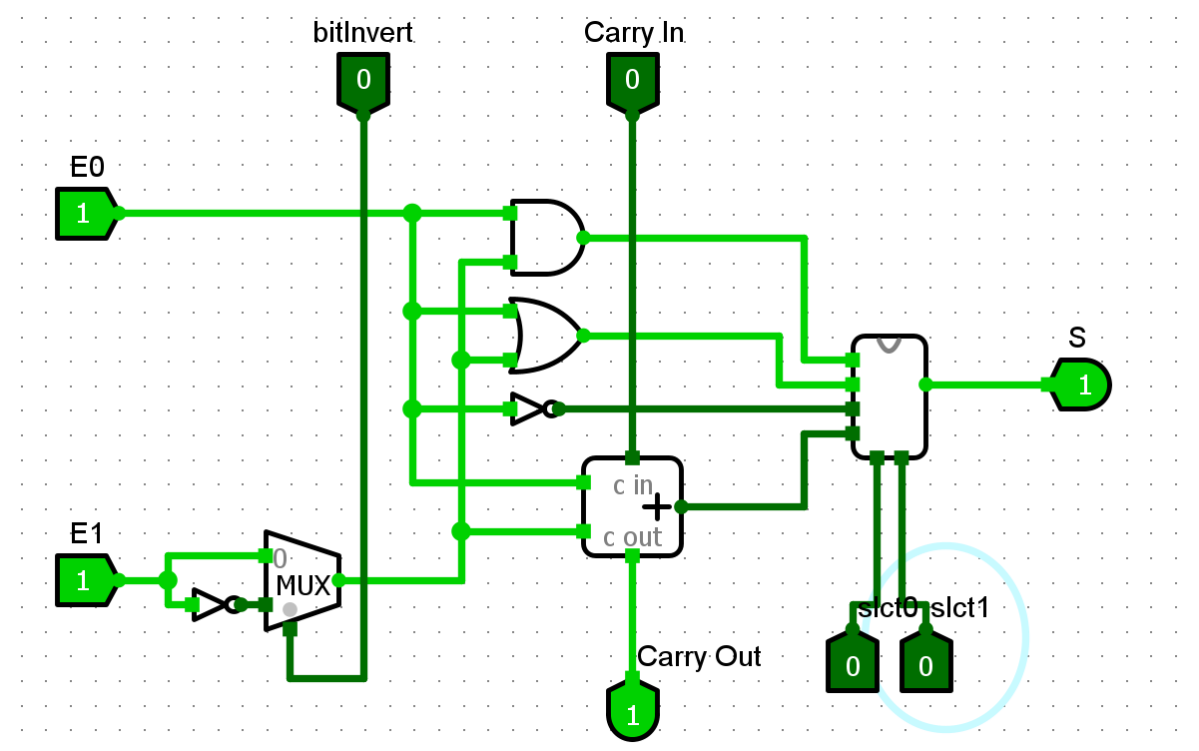
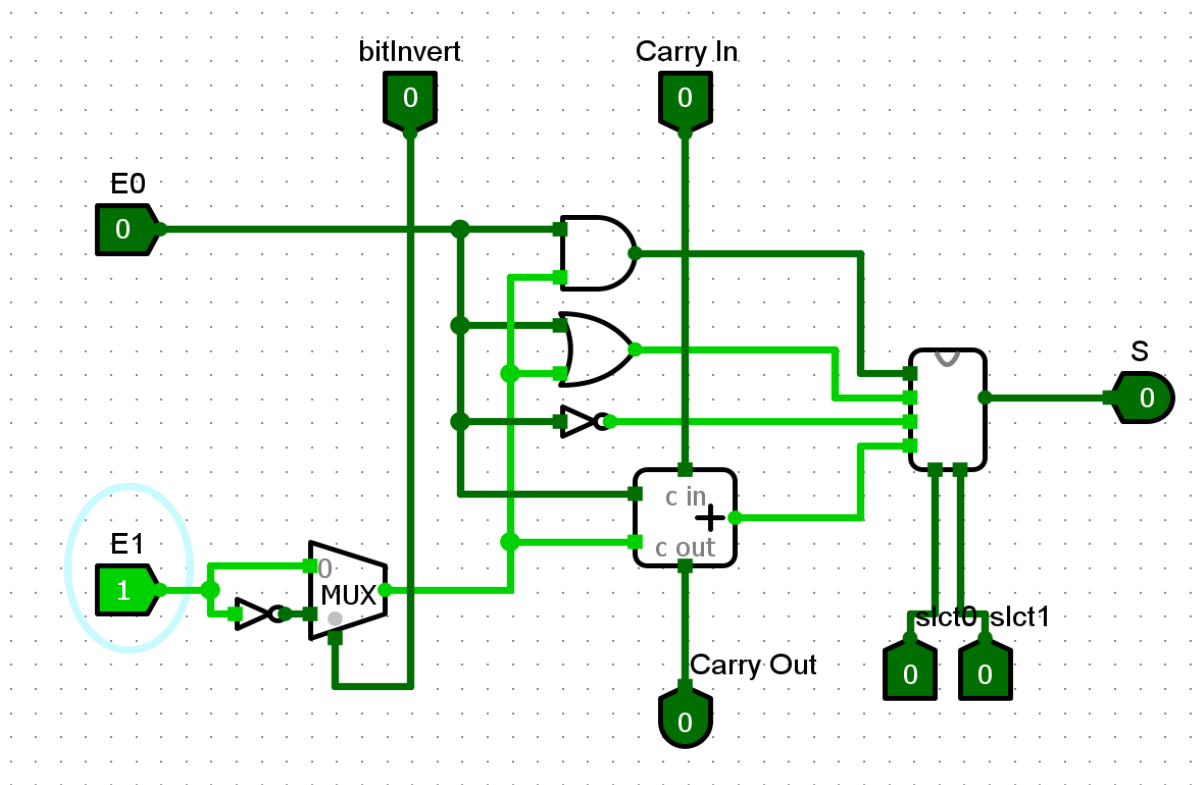
NOT(A);

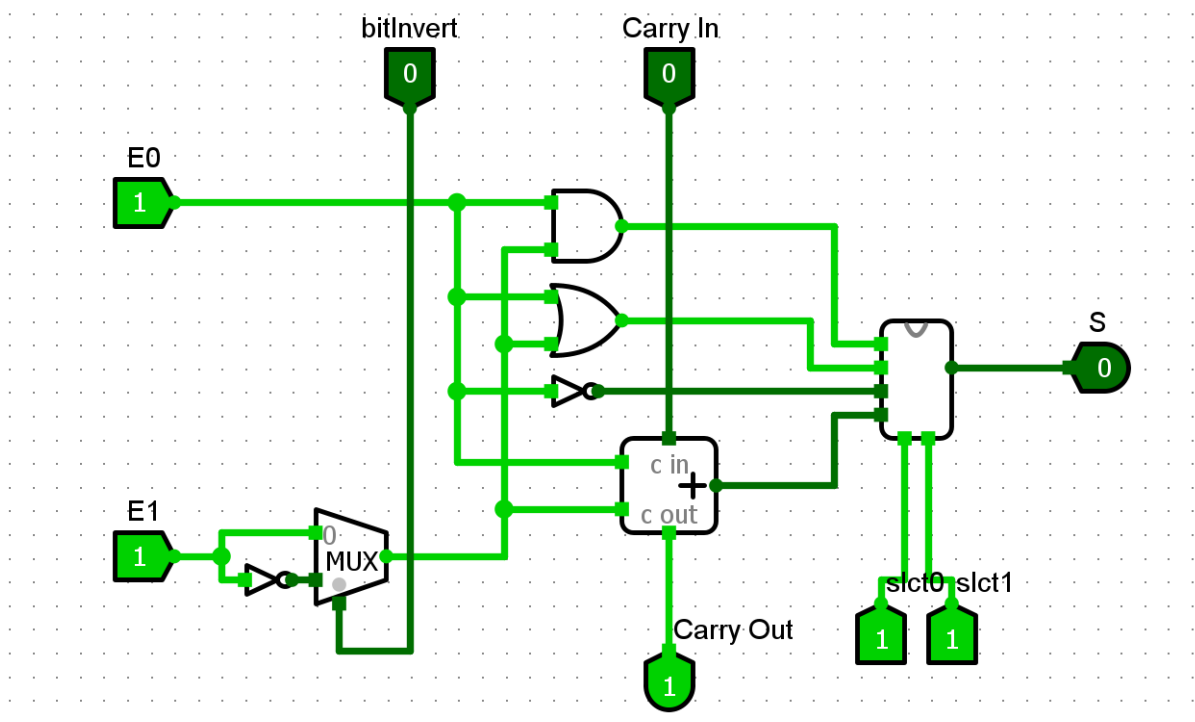
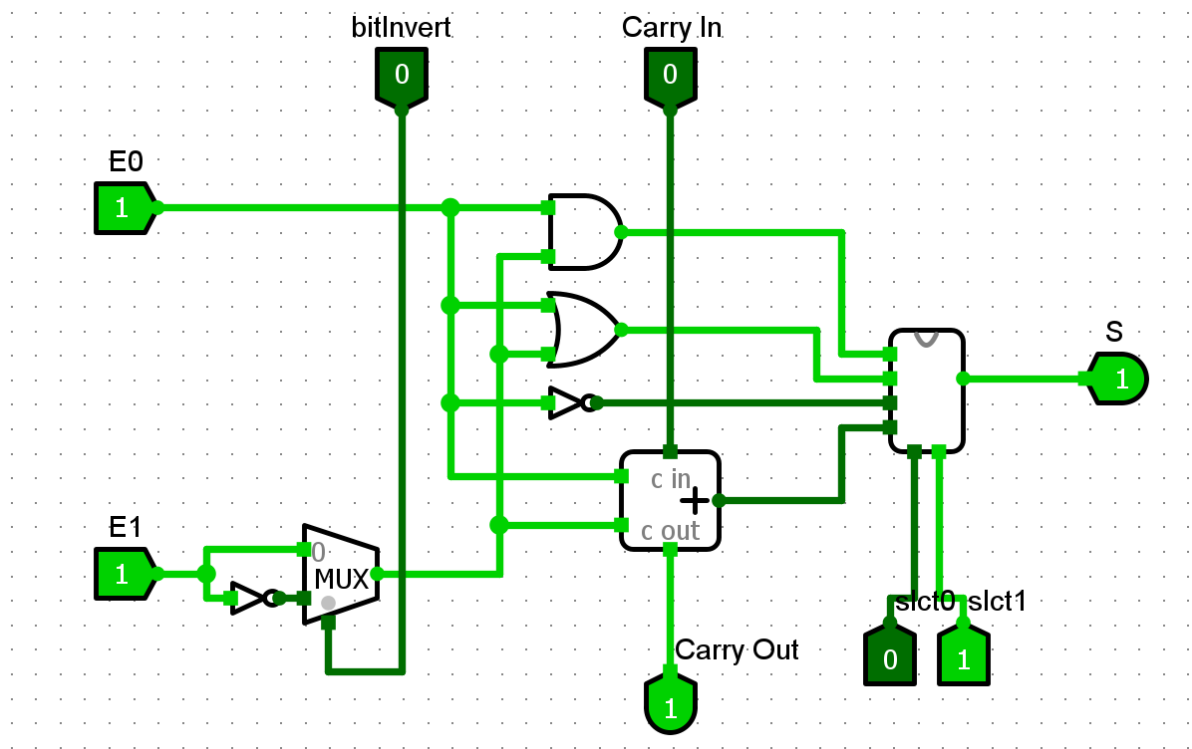
SOMA (A,-B);

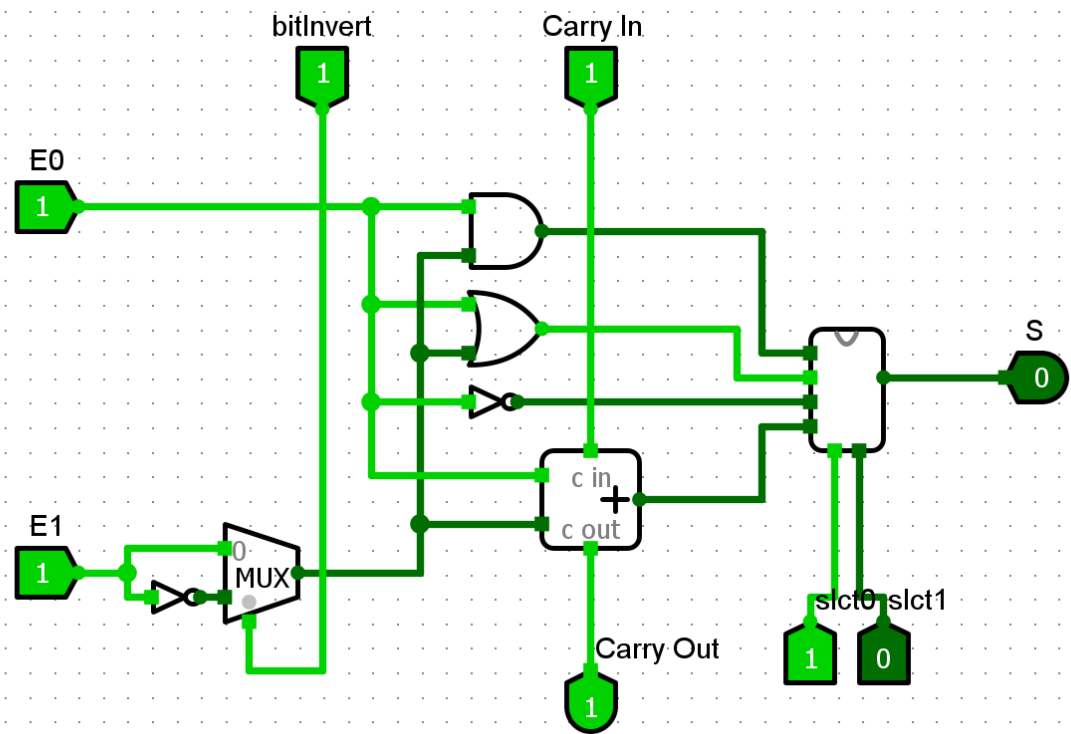
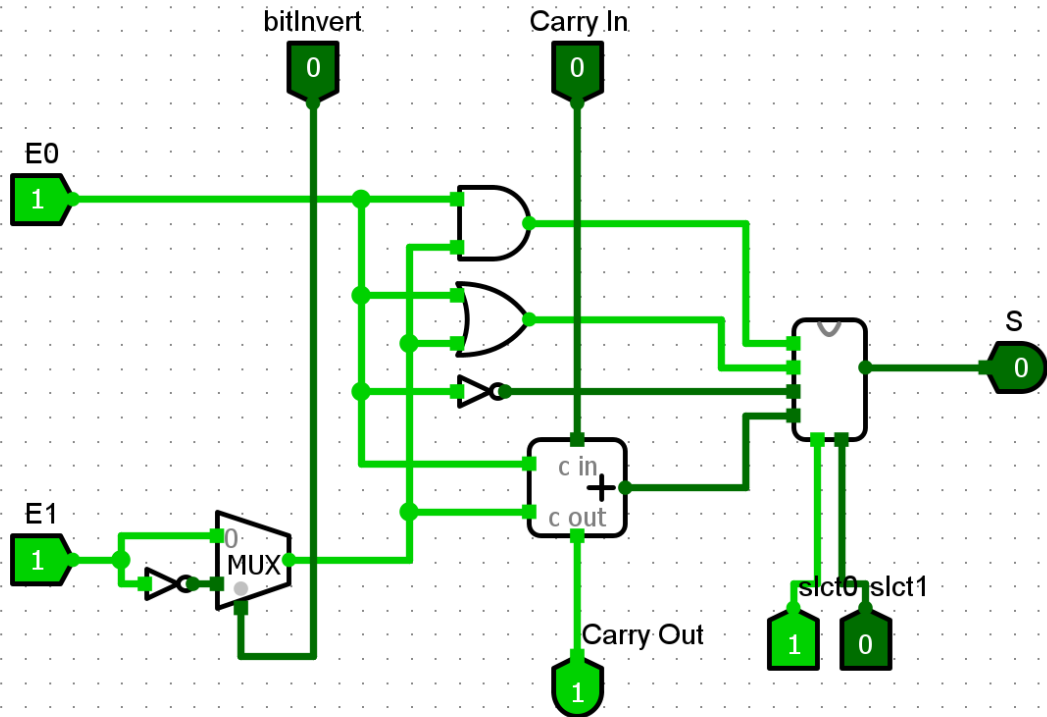
Resp:

Saída = 0, 1, 1, 0, 0, 0

Fluxo:



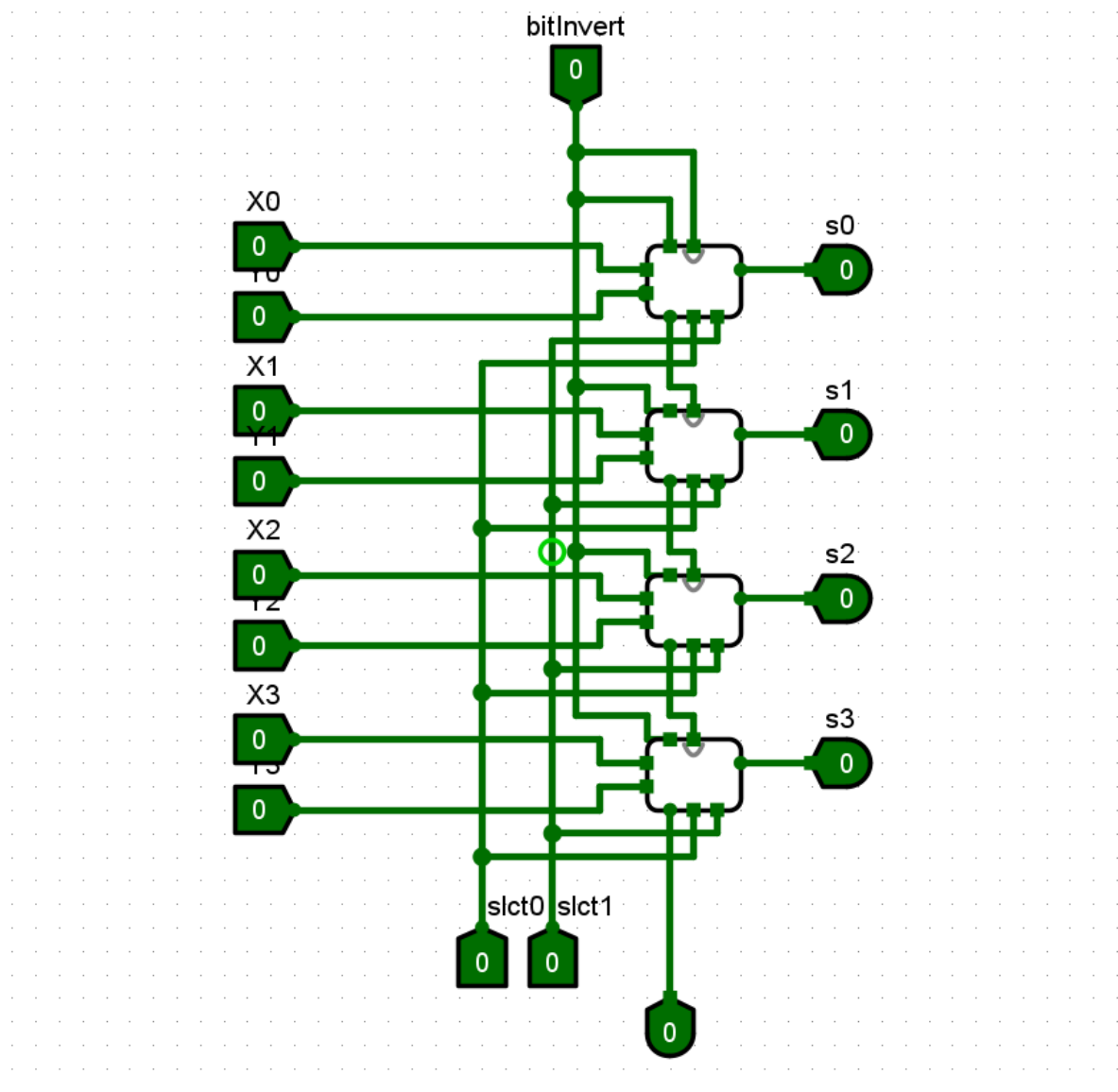




5) Usando essa ula de 1 bit, construa essa ULA para 4 bits no Logisim e verifique o seu funcionamento. Veja como funciona o barramento de instruções (operation) e o barramento de dados (a e b). Observe a ligação do Binvert ao Carry_in da primeira ULA. Procure usar subcircuitos, seu circuito deverá estar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

Resp:

-Unidade Lógica e Aritmética (ULA) de 4 bits:



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.

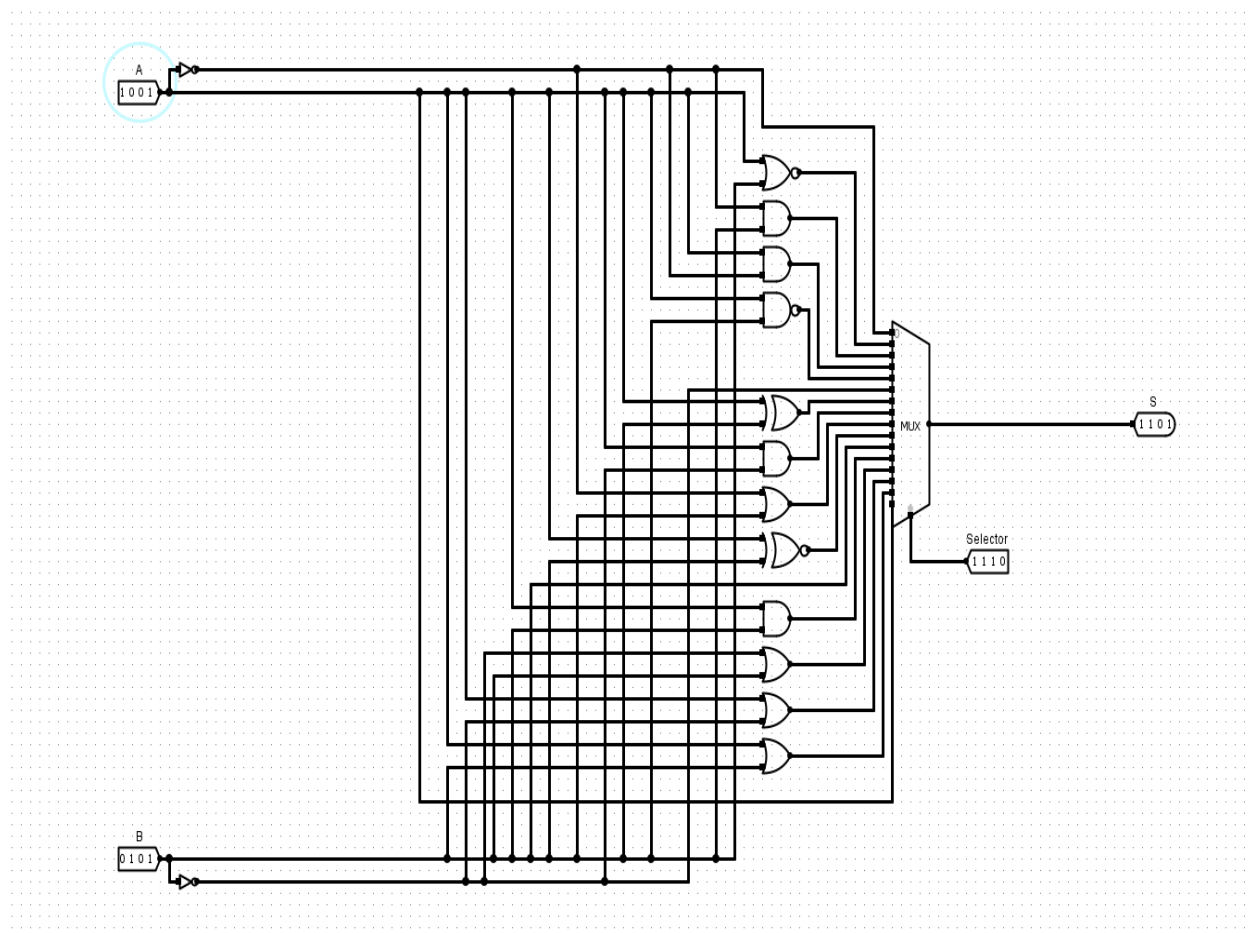
Para o programa de teste acima, preencher a tabela a seguir considerando que cada linha corresponderá à execução de uma instrução (a primeira linha já foi realizada, observe que a palavra deverá conter 10 bits, para escrevermos em hexa completamos os dois bits à esquerda com zero):

Instrução realizada	Binário: A,B,Op.cod	Valor em Hexa (0x...)	Resultado em binário
AND(A,B)	0010 0001 00	0000 1000 0100= 0x084	0000
OR(A,B)	0010 0011 01	0000 1000 1101= 0x08D	0011
SOMA(A,B)	0010 0011 11	0000 1000 1111= 0x08F	0101
NOT(A)	1100 0011 10	0011 0000 1110= 0x30E	0011
AND(B,A)	1101 1100 00	0011 0110 0000= 0x360	1100

- PARTE 2

Circuitos utilizados:

Unidade Lógica e Aritmética (ULA) 74181:



Instruções	Binário	Resultado da operação
450	0100 0101 0000	1011
CB1	1100 1011 0001	0000
A32	1010 0011 0010	0001
C43	1100 0100 0011	0000
124	0001 0010 0100	1111
785	0111 1000 0101	0111
9B6	1001 1011 0110	0010
CD7	1100 1101 0111	0000
FE8	1111 1110 1000	1110
649	0110 0100 1001	1101
D9A	1101 1001 1010	1001
FCB	1111 1100 1011	1100
63C	0110 0011 1100	1111
98D	1001 1000 1101	1111
76E	0111 0110 1110	0111
23F	0010 0011 1111	0010

Se o objetivo fosse realmente testar esta ULA, quantas linhas a nossa tabela verdade deveria ter, ou seja na verdade a tabela que você preencheu deveria ter quantas linhas?

$$A * B * \text{NumOpe} = 2^4 * 2^4 * 2^4 = 2^{12} = 4096 \text{ linhas}$$