Exercício Prático 2 Parte 1
Laboratório de ac2

Objetivo:
Cosntruir uma Unidade Lógica e Aritmética (ULA) de 1 bit, 4 bits e implementar no Logisim e Arduino.

Parte 1 (estudo da ALU usando Logisim):

1. Considere a Unidade Lógica e Aritmética de 1 bit ilustrada na Figura 1 a seguir:

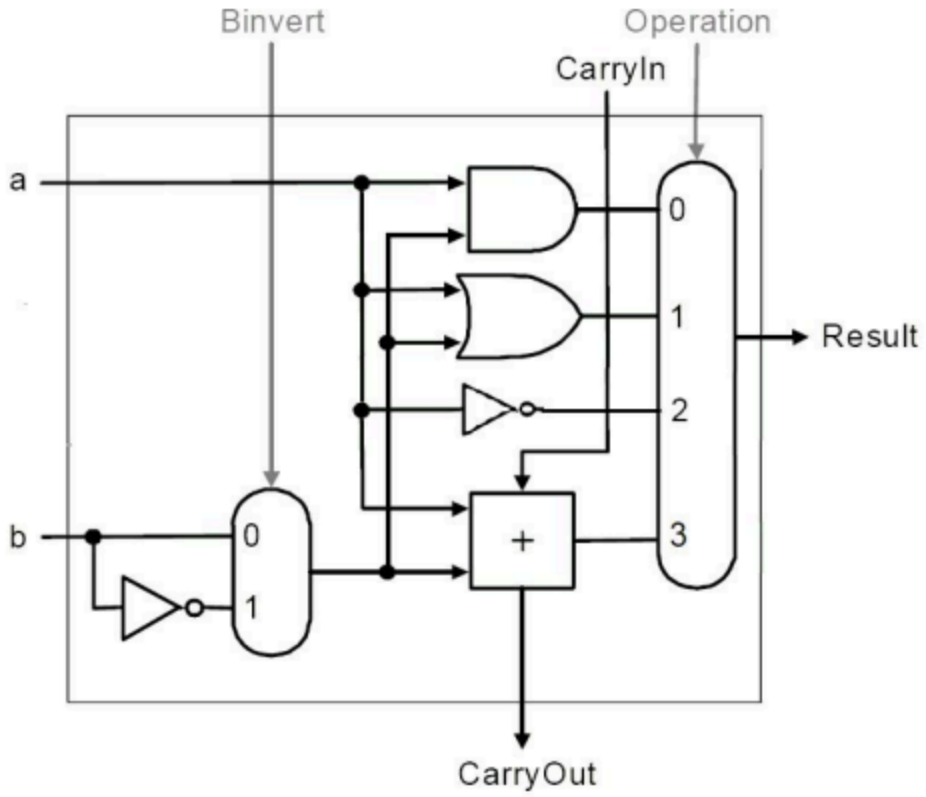
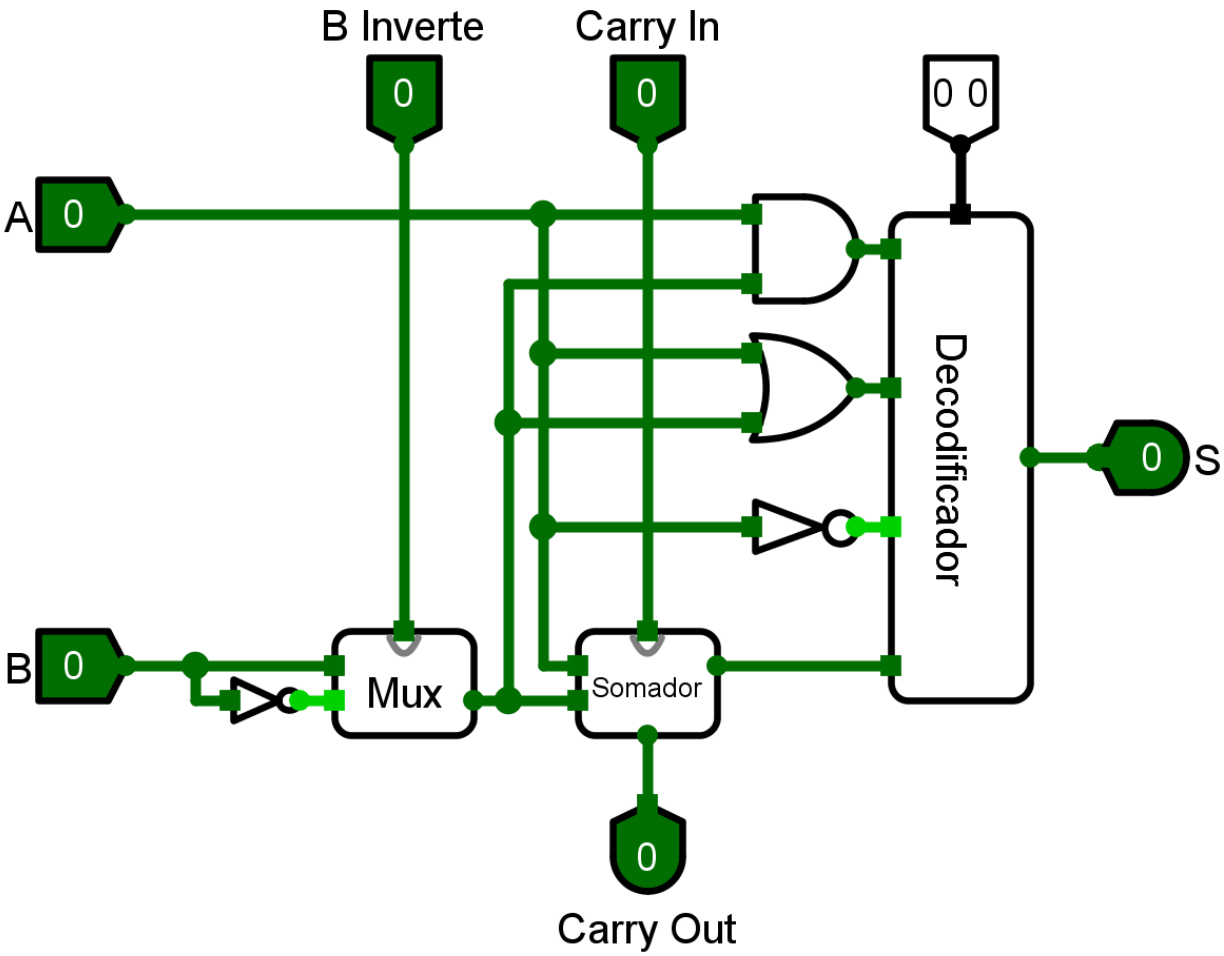


Figura 1

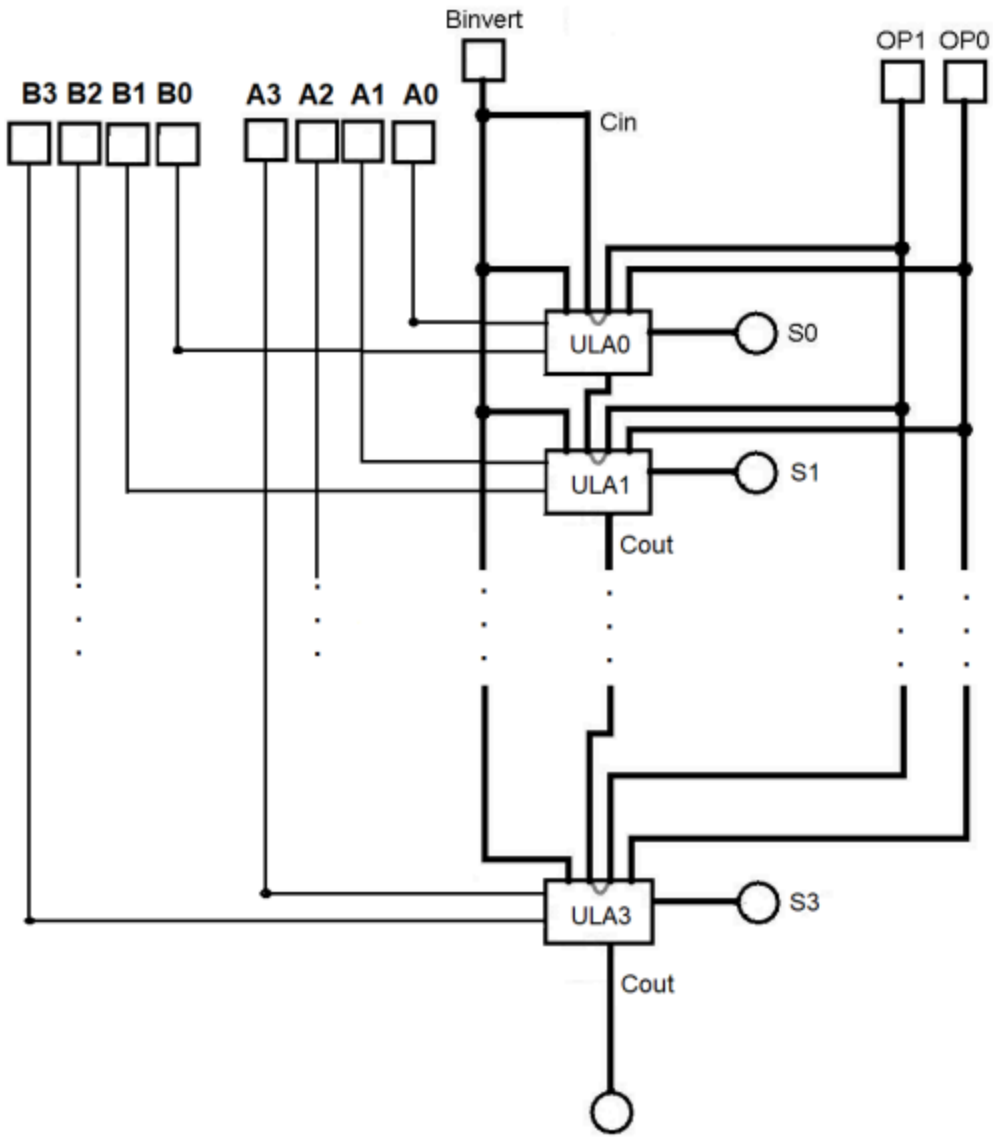
2. Procure entender o esquema, principalmente a subtração.
3. Sua ULA possui a seguinte tabela de opcodes:

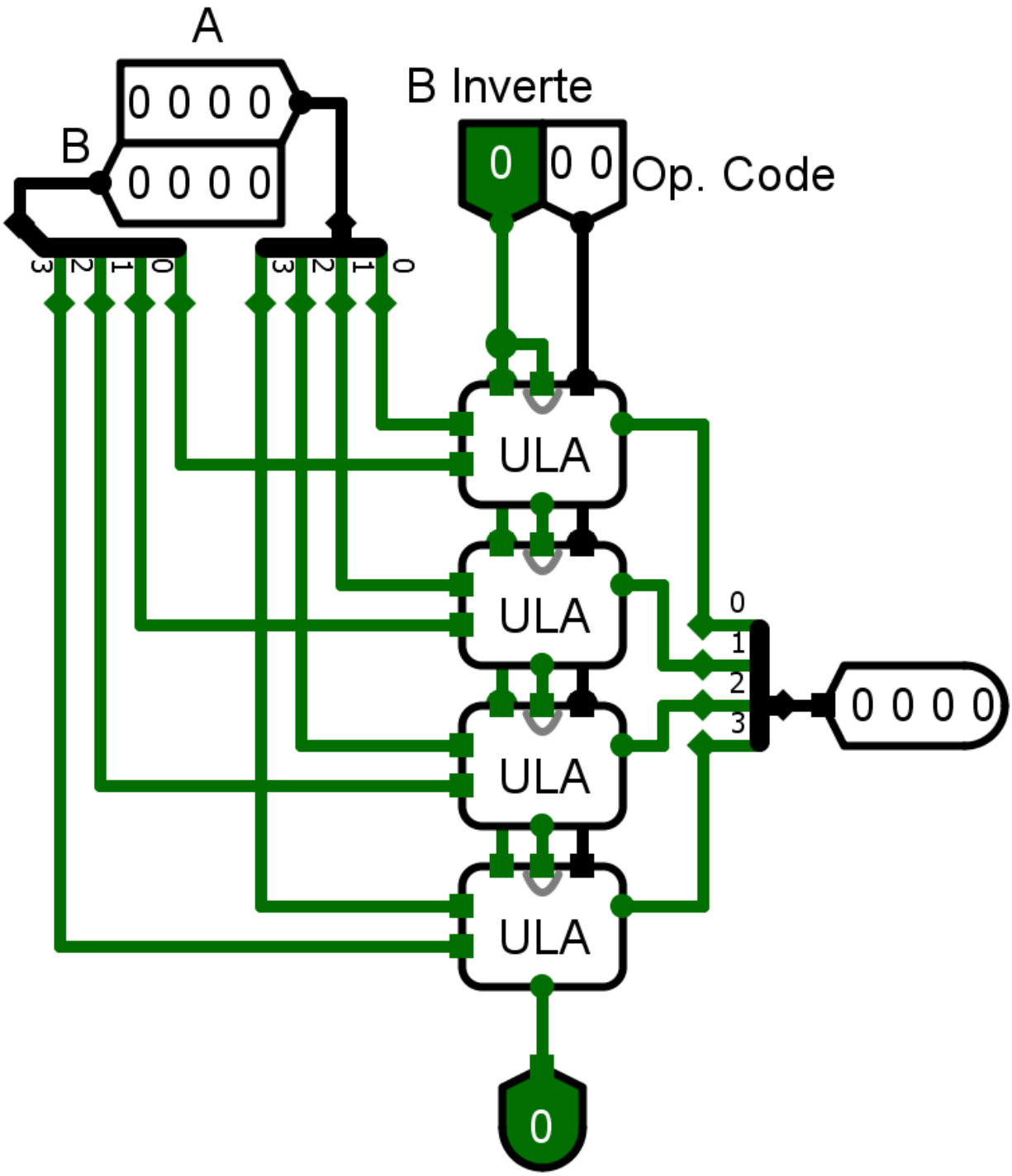
Op. Code (Operation)	Instrução (Result)
0	AND (a,b)
1	OR (a,b)
2	NOT (a)
3	SOMA(a,b)


B Inverte
Carry In
0
0
0 0
A
0
Decodificador
0
S
B
0
Mux
Somador
0
Carry Out

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:




A
0 0 0 0
B
0 0 0 0
B Inverte
0
0 0
Op. Code
3
2
1
0
3
2
1
0
ULA
ULA
ULA
ULA
0
1
2
3
0 0 0 0
0

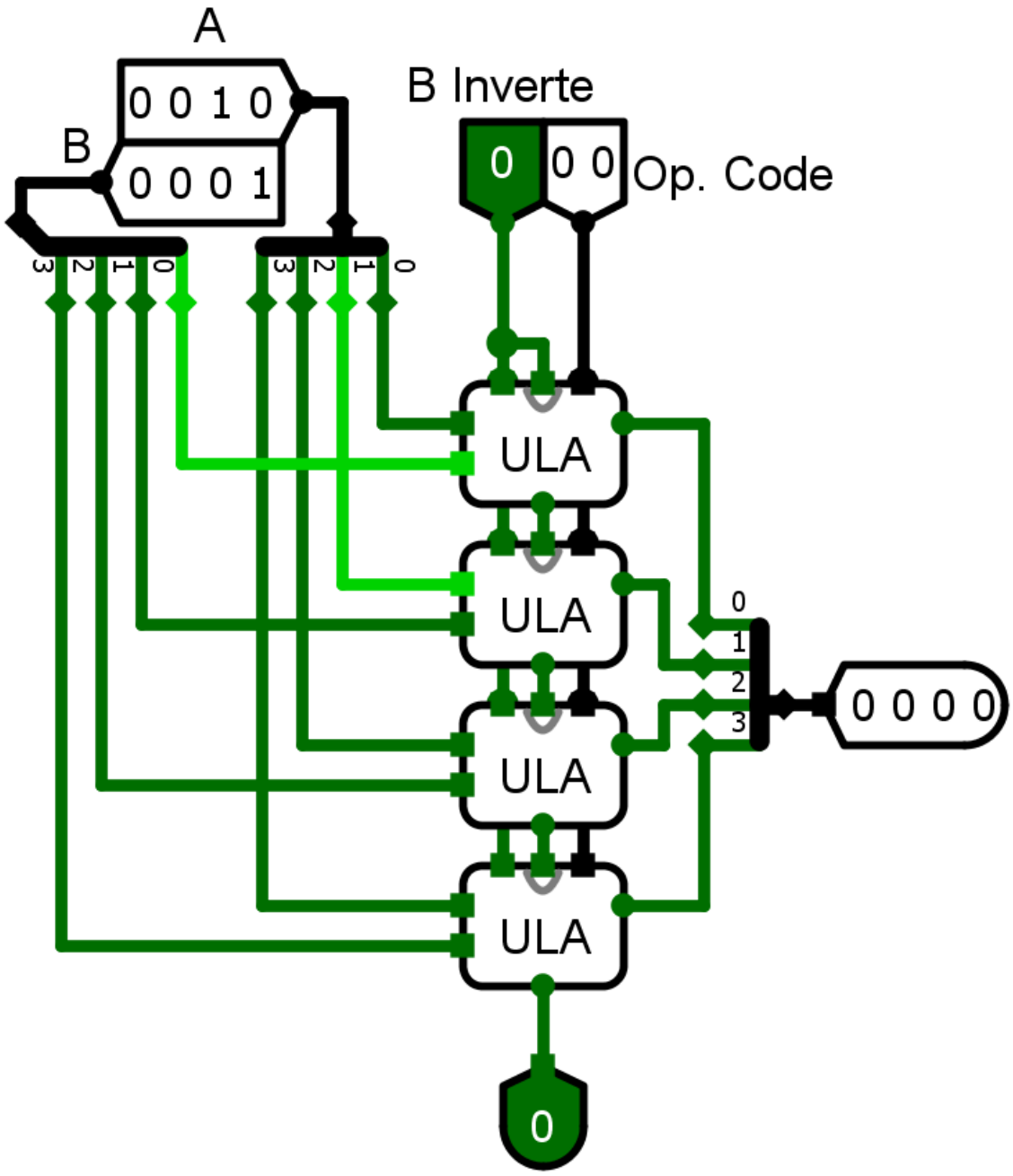
6. Teste a sua ULA de acordo com o seguinte roteiro (considerando os números de 4 bits):

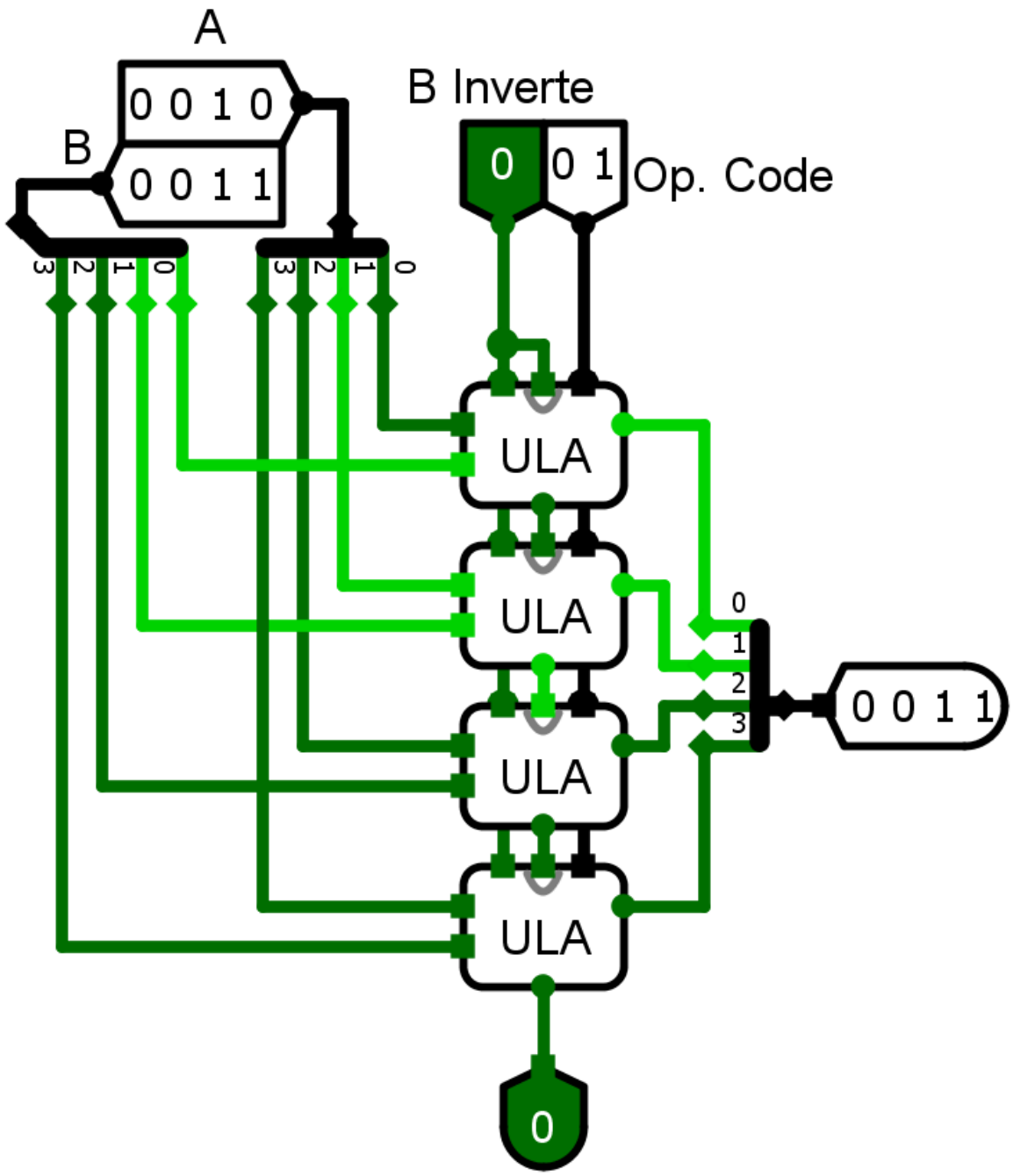
Inicio:
A=2; (ou A=0010)
B=1; (ou B=0001)
AND(A,B);
B=3; (ou B=0011)
OR(A,B);
SOMA(A,B);
A=12; (ou A=1100)
NOT(A);
B=13; (ou B=1101)
AND(B,A);
Fim.

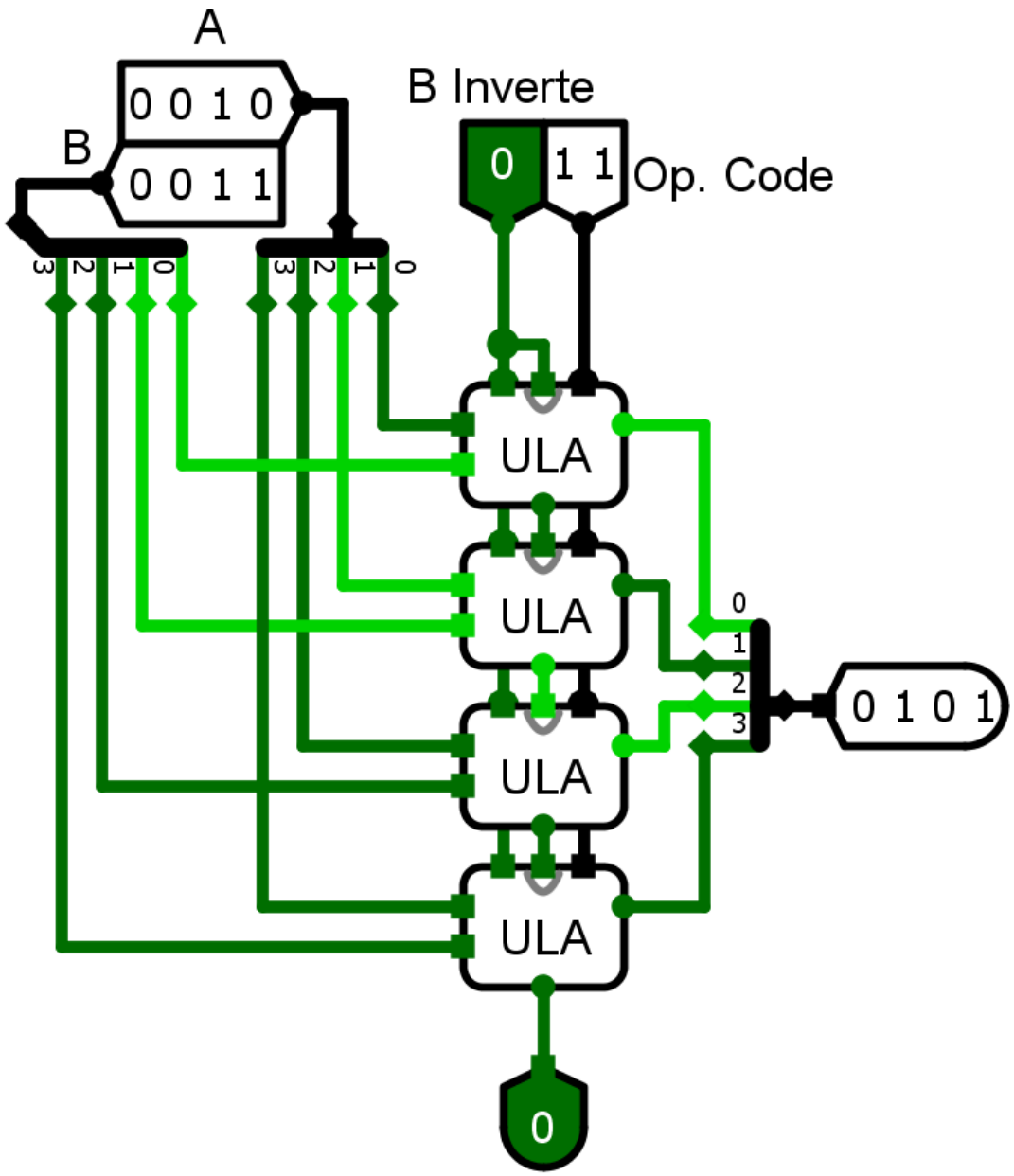
Para o programa de teste acima, preencher a tabela a seguir considerando que cada linha corresponderá à execução de uma instrução (a primeira linha já foi realizada, observe que a palavra deverá conter 10 bits, para escrevermos em hexa completamos os dois bits à esquerda com zero):

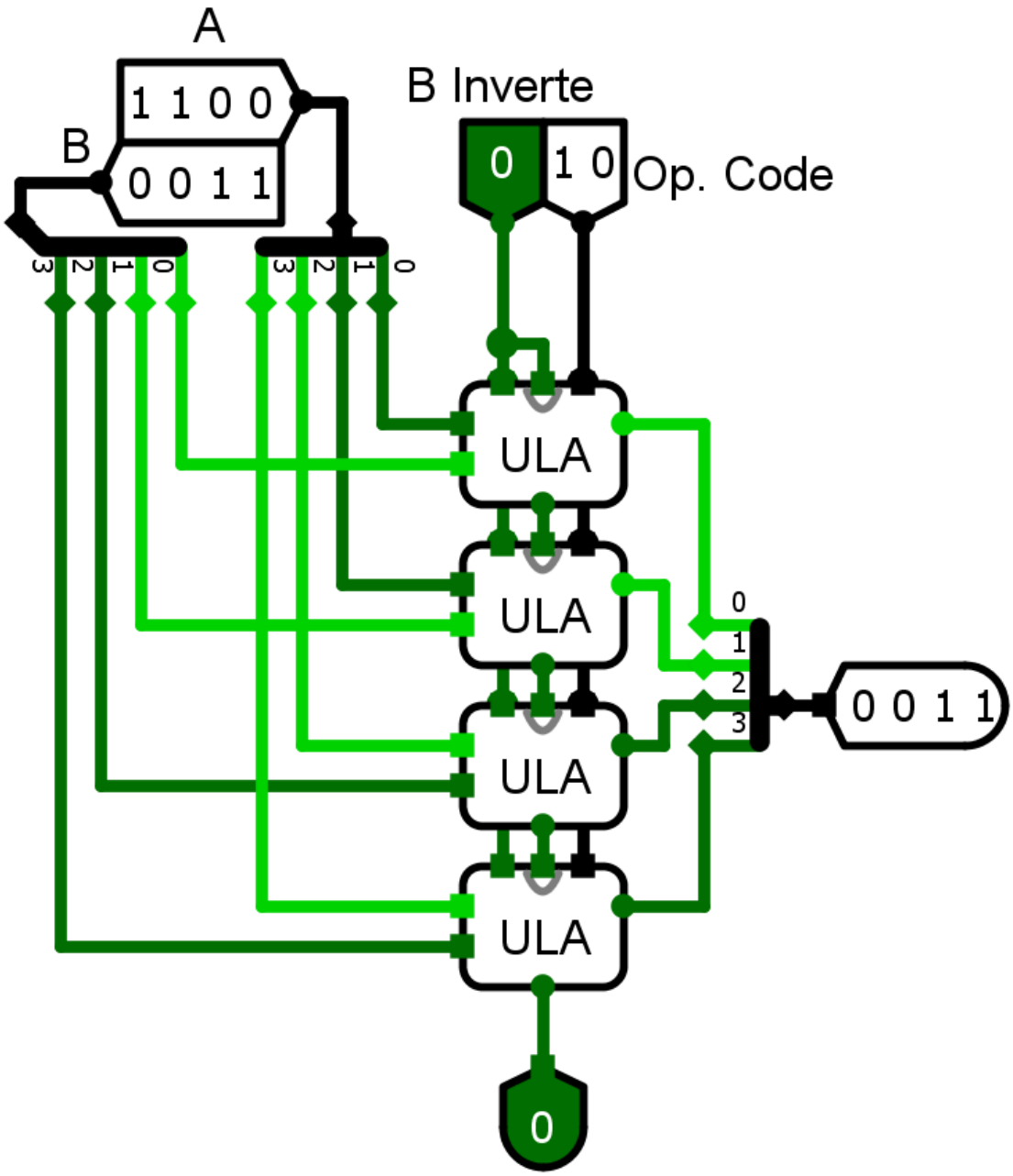
Para o programa de teste acima, preencher a tabela a seguir considerando que cada linha corresponderá à execução de uma instrução (a primeira linha já foi realizada, observe que a palavra deverá conter 10 bits, para escrevermos em hexa completamos os dois bits à esquerda com zero):

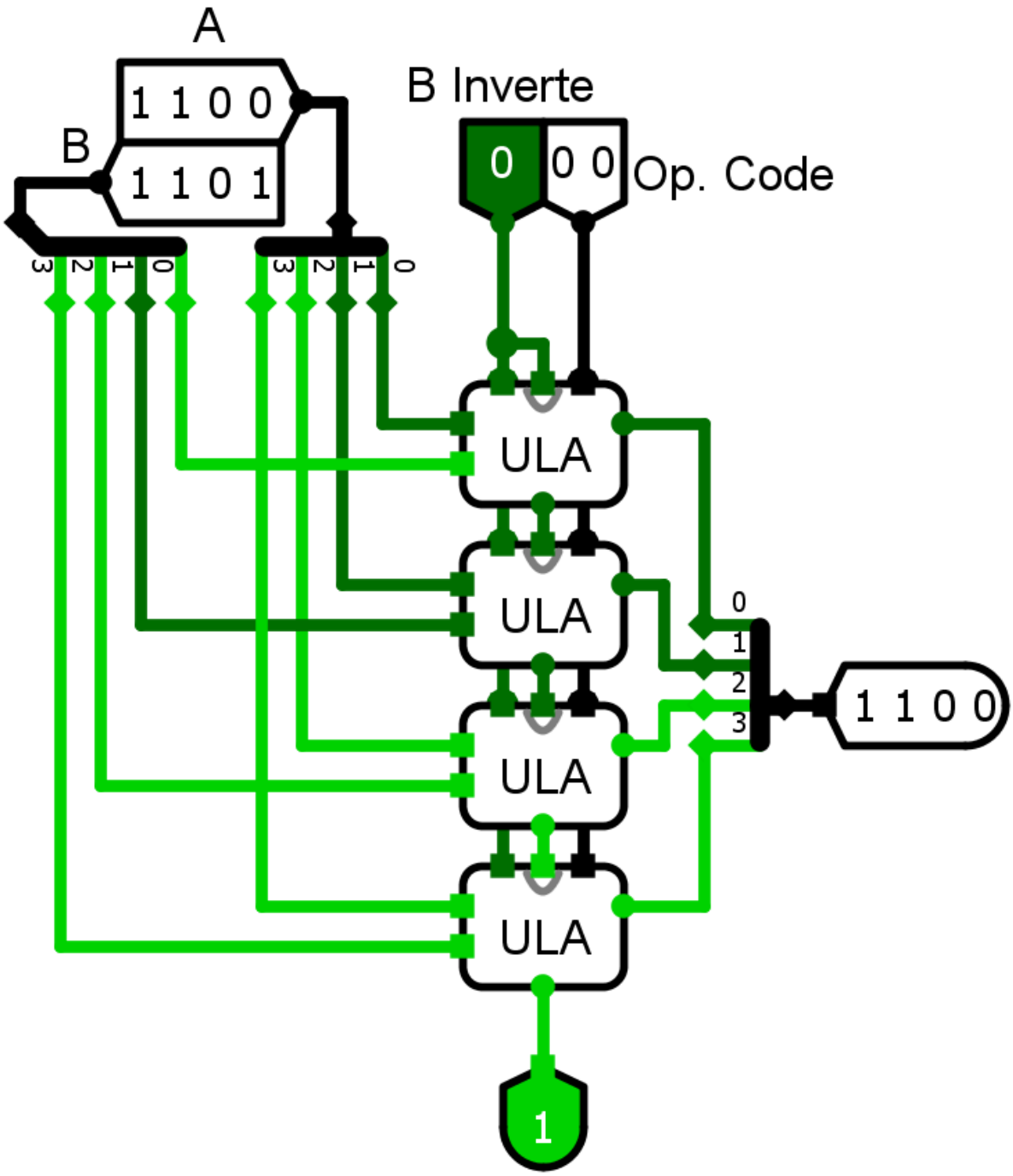
Instrução realizada	Binário (A,B,Op.code)	Valor em Hexa (0x …)	Resultado em binário
AND(A,B)	0010 0001 00	(0000 1000 0100) = 0x084	0000
OR(A,B)	0010001101	(000010001101) = 0x08D	0011
SOMA(A,B)	0010001111	(000010001111) = 0x08F	0101
NOT(A)	1100001110	(001100001110) = 0x30E	0011
AND(B,A)	1100110100	(001100110100) = 0x334	1100


A
0 0 1 0
B
0 0 0 1
B Inverte
0
0 0
Op. Code
3
2
1
0
3
2
1
0
ULA
ULA
ULA
ULA
0
1
2
3
0 0 0 0
0


A
0 0 1 0
B
0 0 1 1
B Inverte
0
0 1
Op. Code
3
2
1
0
3
2
1
0
ULA
ULA
ULA
ULA
0
1
2
3
0 0 1 1
0


A
0 0 1 0
B
0 0 1 1
B Inverte
0
1 1
Op. Code
3
2
1
0
3
2
1
0
ULA
ULA
ULA
ULA
0
1
2
3
0 1 0 1
0


A
1 1 0 0
B
0 0 1 1
B Inverte
0
1 0
Op. Code
3 2 1 0
3 2 1 0
ULA
ULA
ULA
ULA
0
1
2
3
0 0 1 1
0


A
1 1 0 0
B
1 1 0 1
B Inverte
0
0 0
Op. Code
3
2
1
0
3
2
1
0
ULA
ULA
ULA
ULA
0
1
2
3
1 1 0 0
1