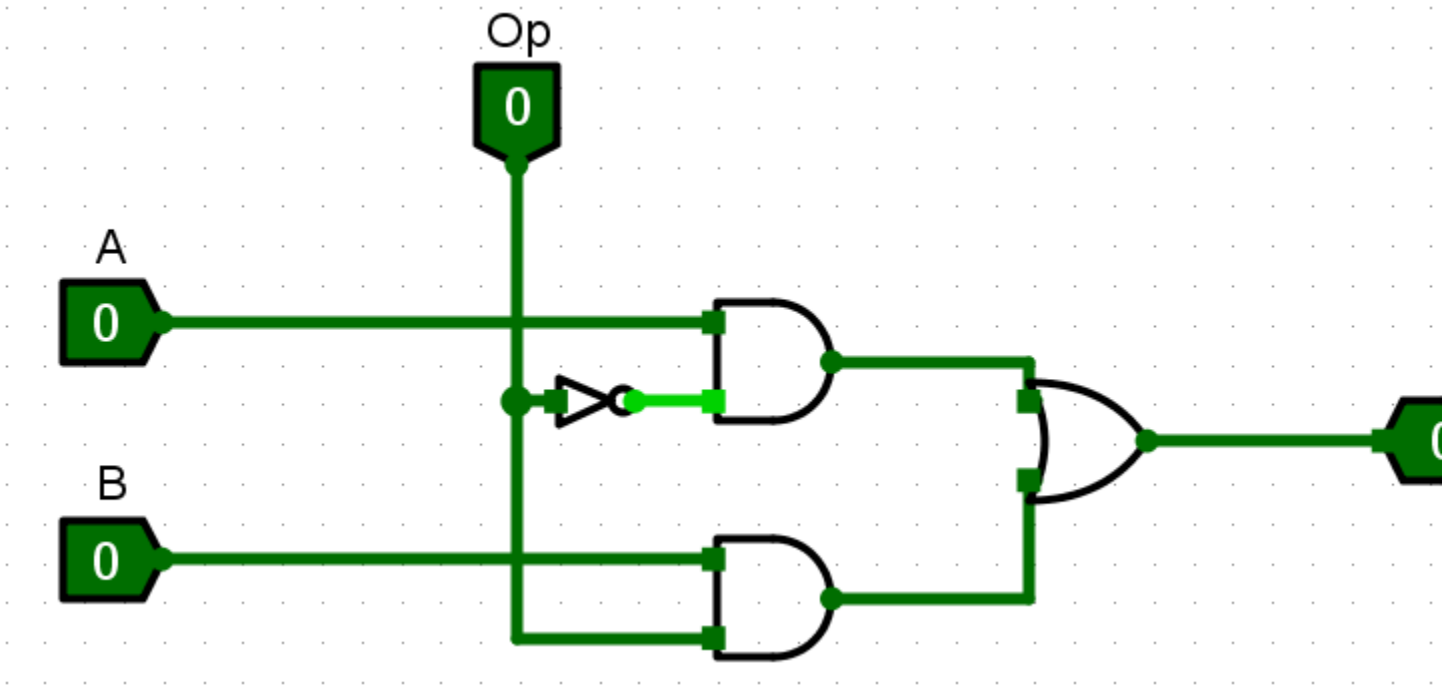
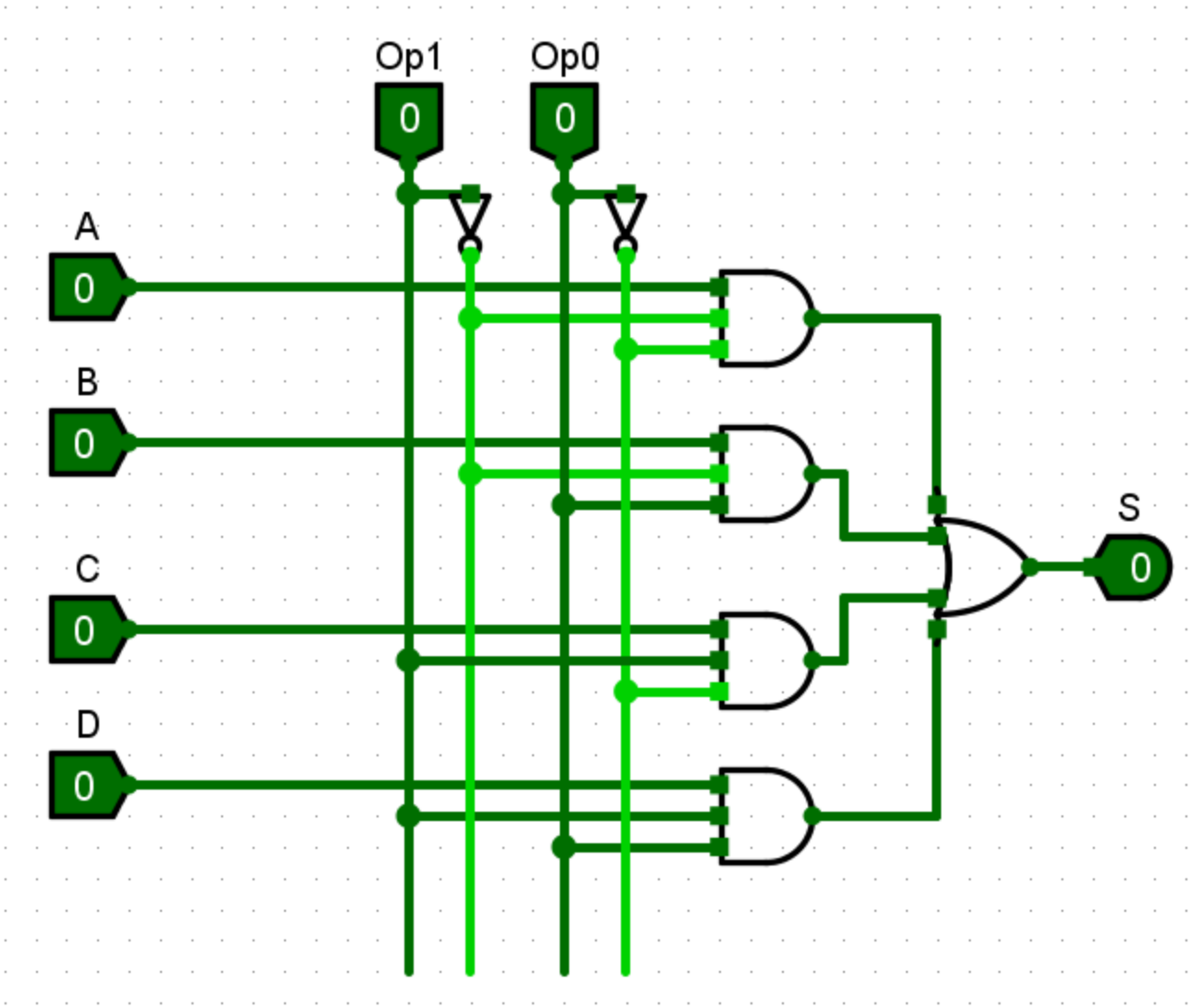
// Giuseppe Cordeiro
// 801779
// Ep02

Parte 1 -

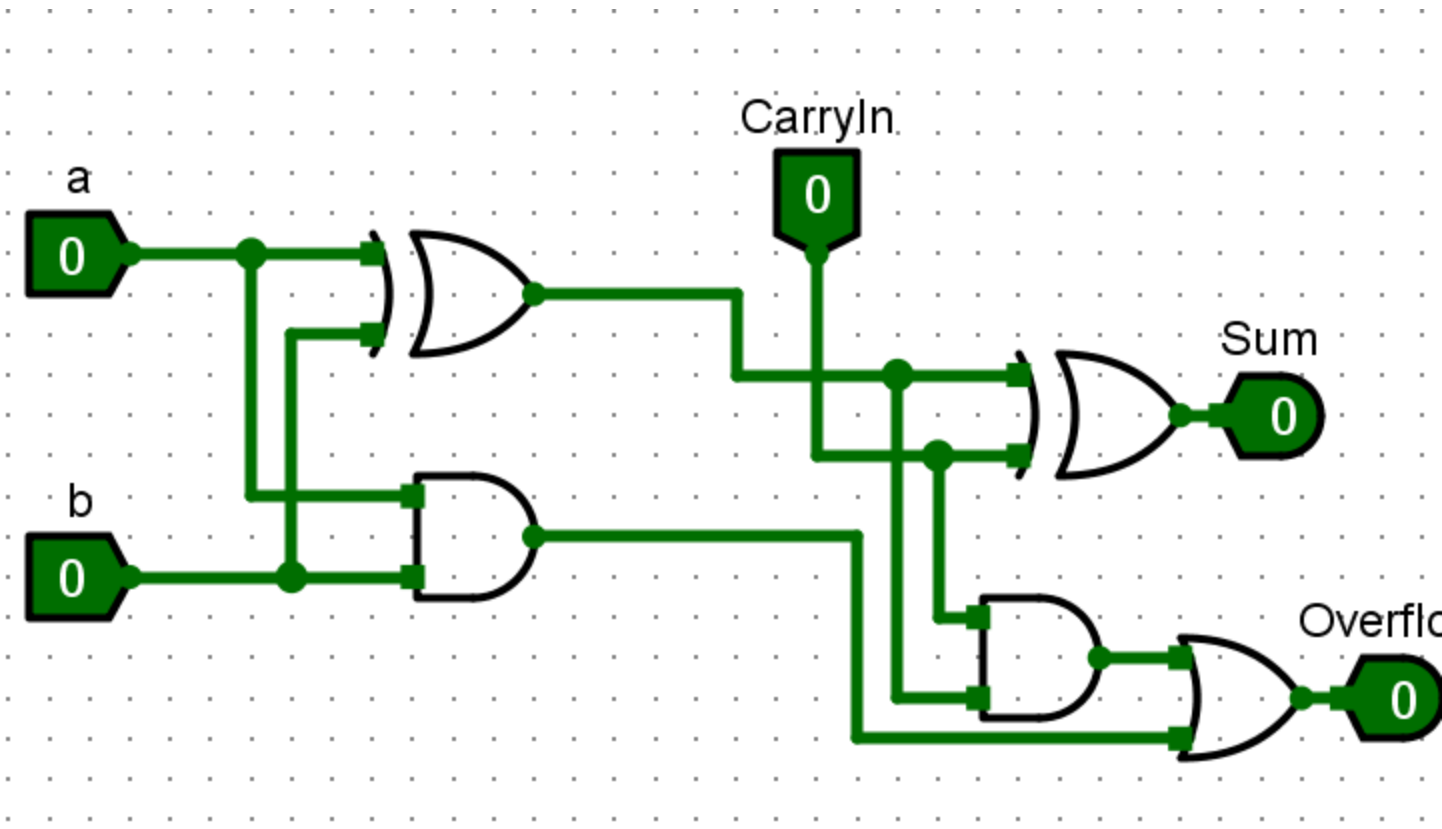
Mux 2x1 -



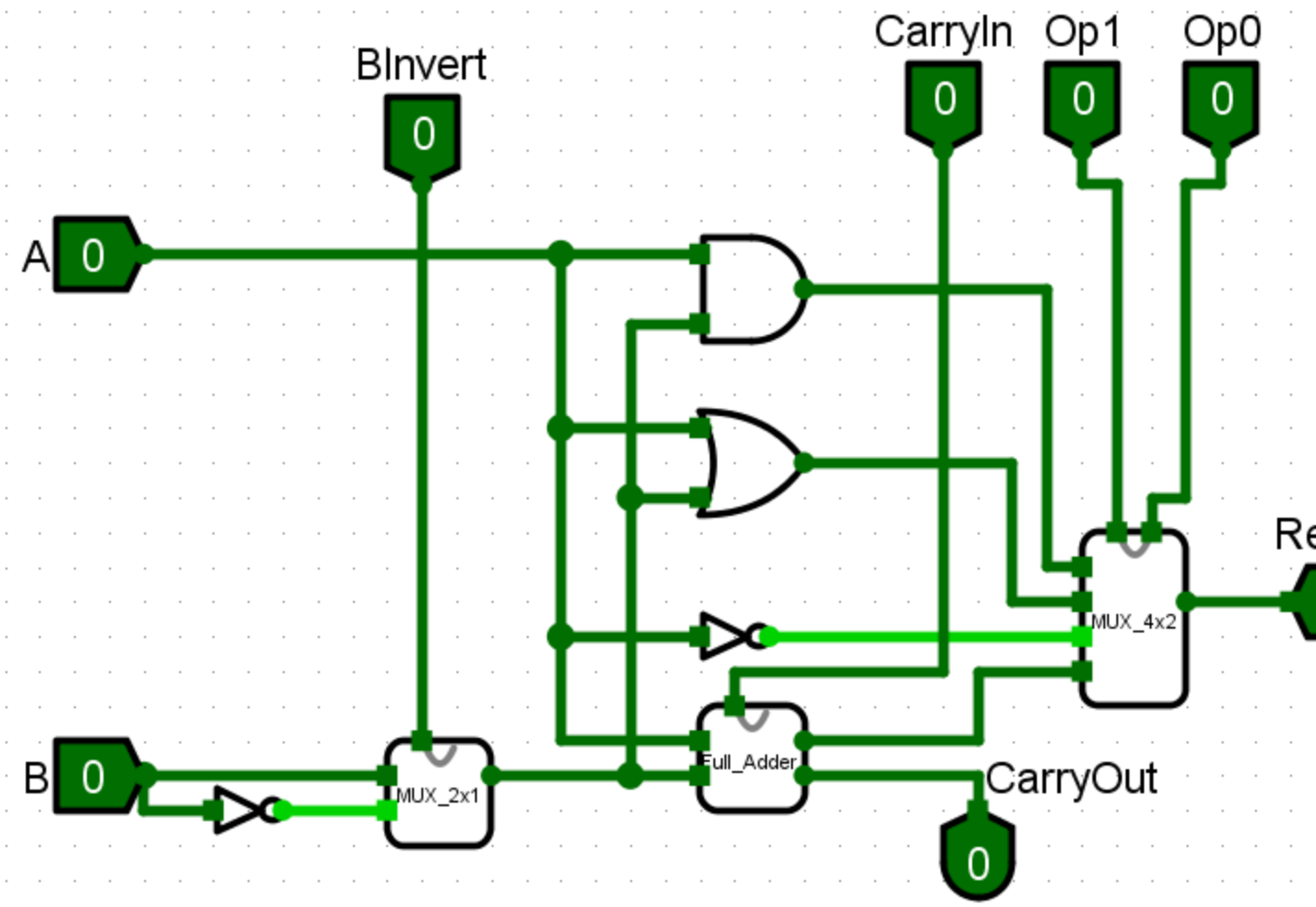
Mux 4x2 -



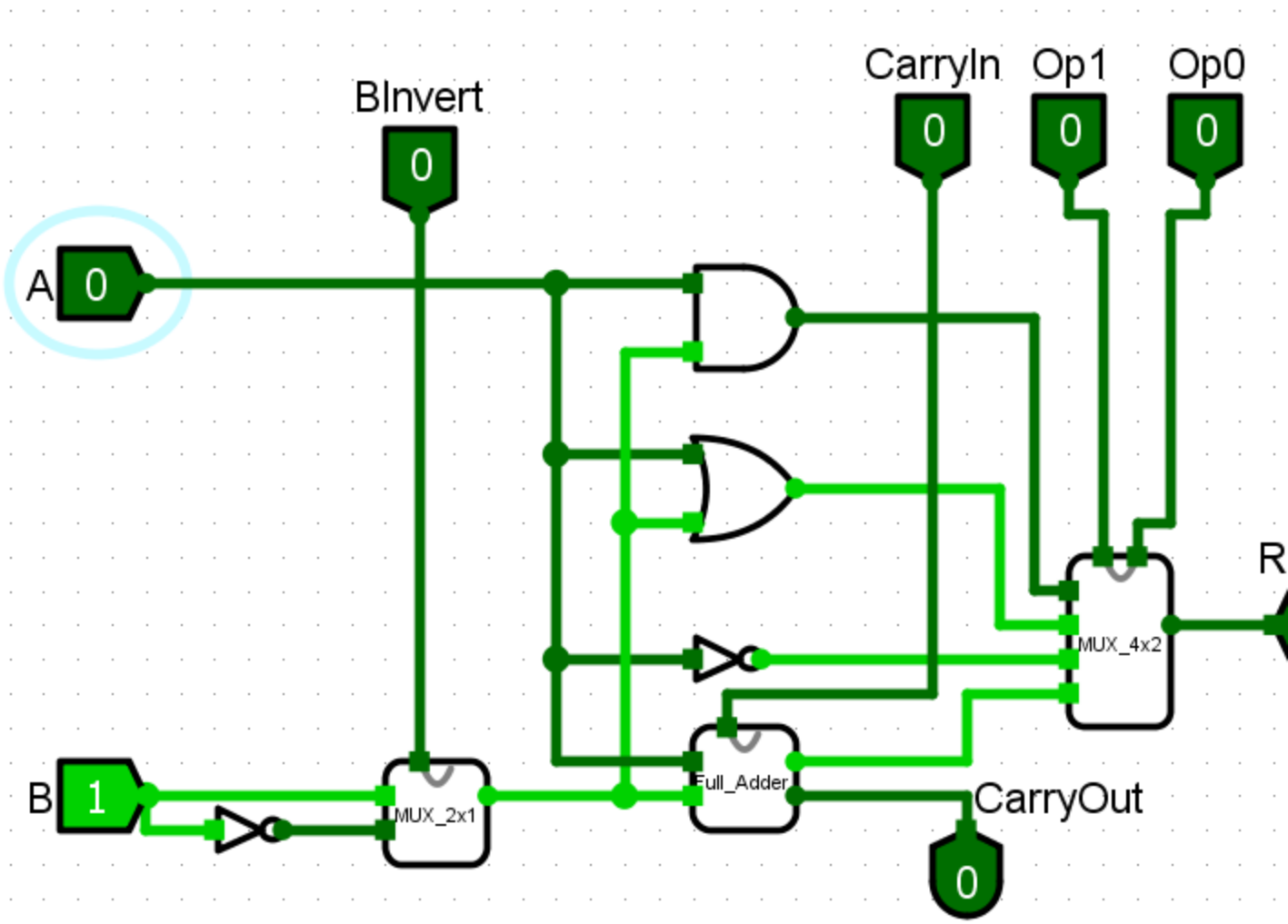
Somador completo -



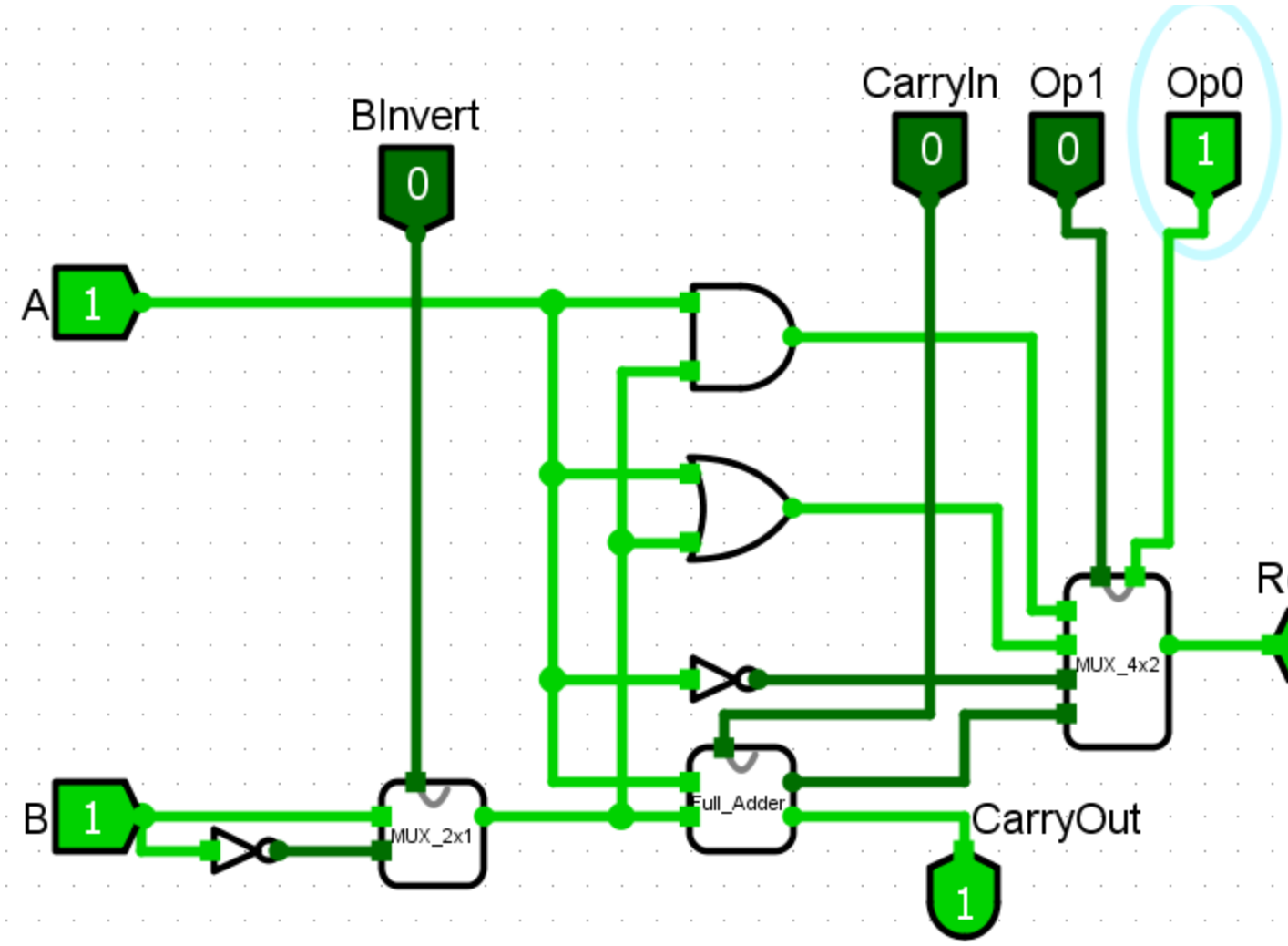
Ula 1 bit -



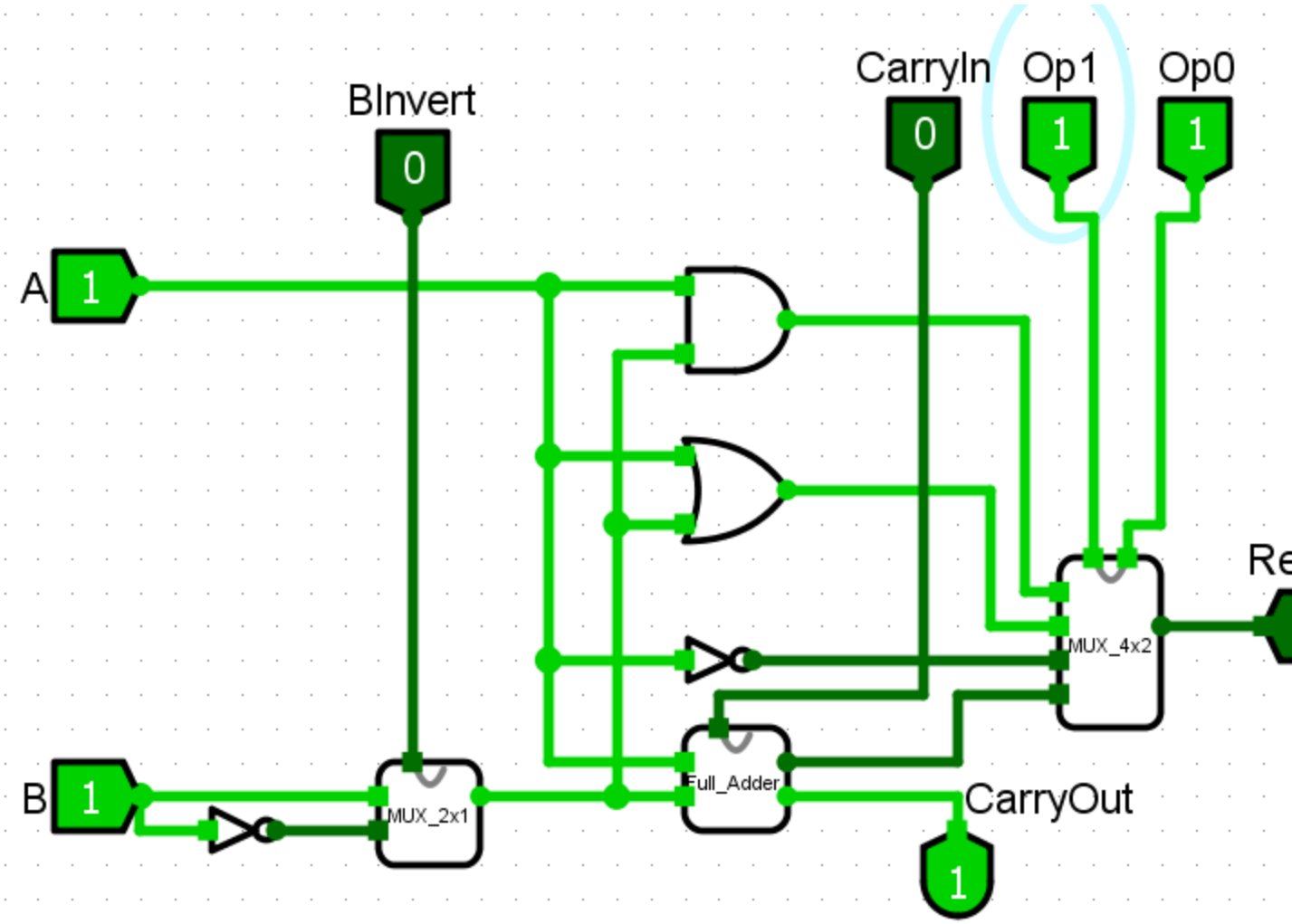
Teste ULA -
AND(A, B) -> A=0 B=1


BInvert
0
CarryIn
0
Op1
0
Op0
0
A
0
B
1
MUX_2x1
Full_Adder
MUX_4x2
CarryOut
0
R

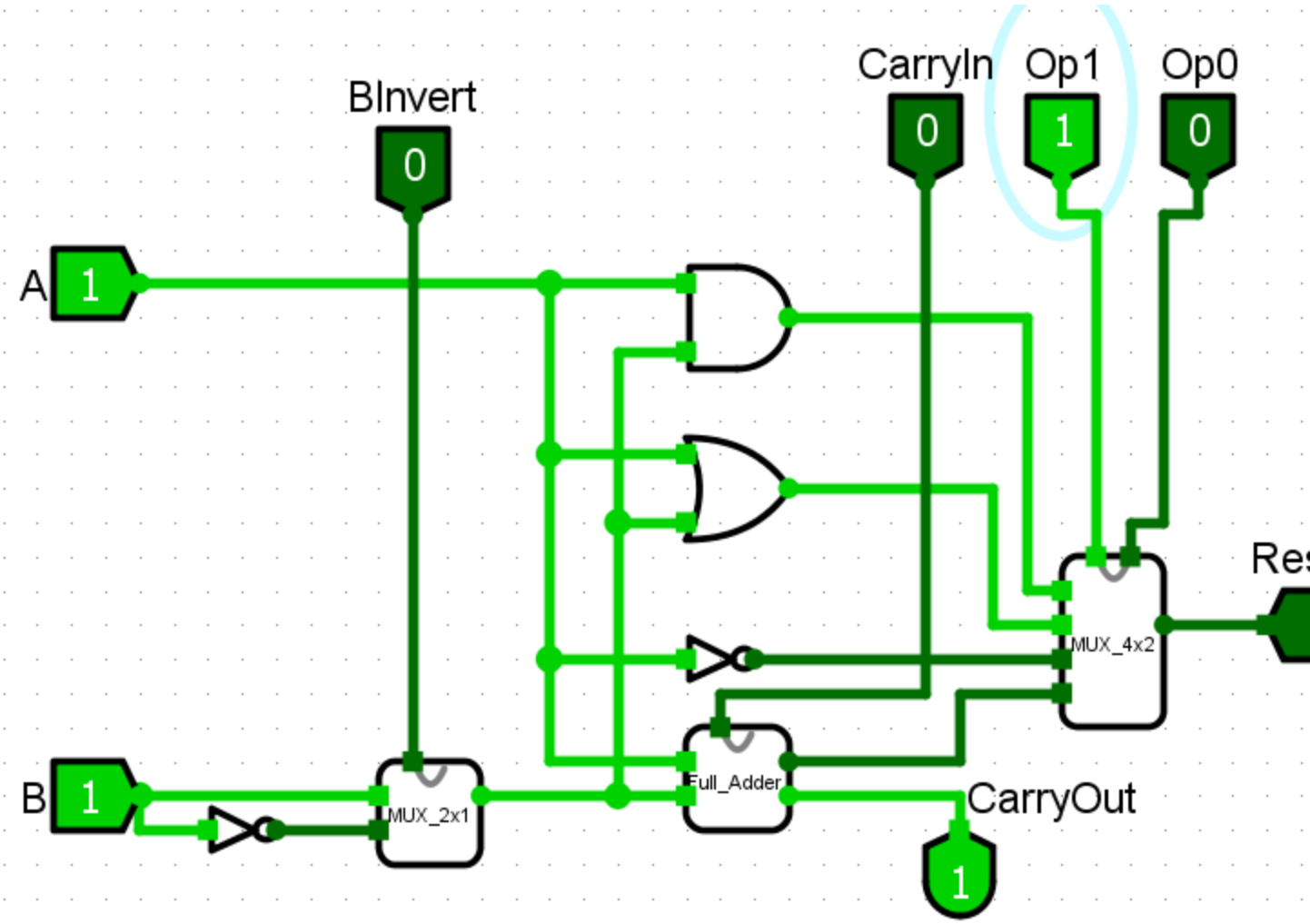
OR(A,B) -> A=1 B=1


BInvert
0
CarryIn
0
Op1
0
Op0
1
A
1
B
1
MUX_2x1
Full_Adder
MUX_4x2
CarryOut
1

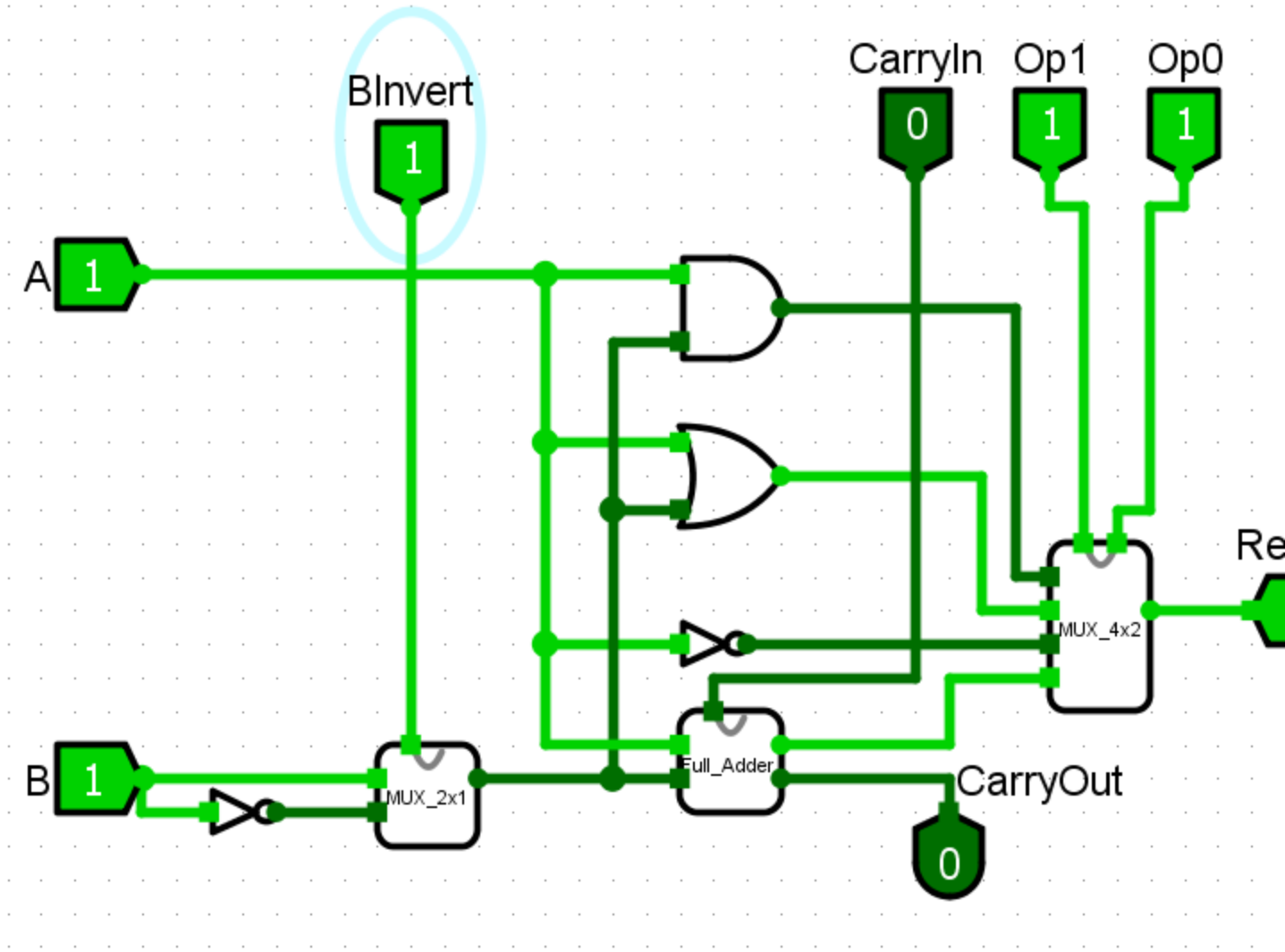
SOMA(A, B) -> A=1 B=1


BInvert
0
CarryIn
0
Op1
1
Op0
1
A
1
B
1
MUX_2x1
Full_Adder
MUX_4x2
CarryOut
1

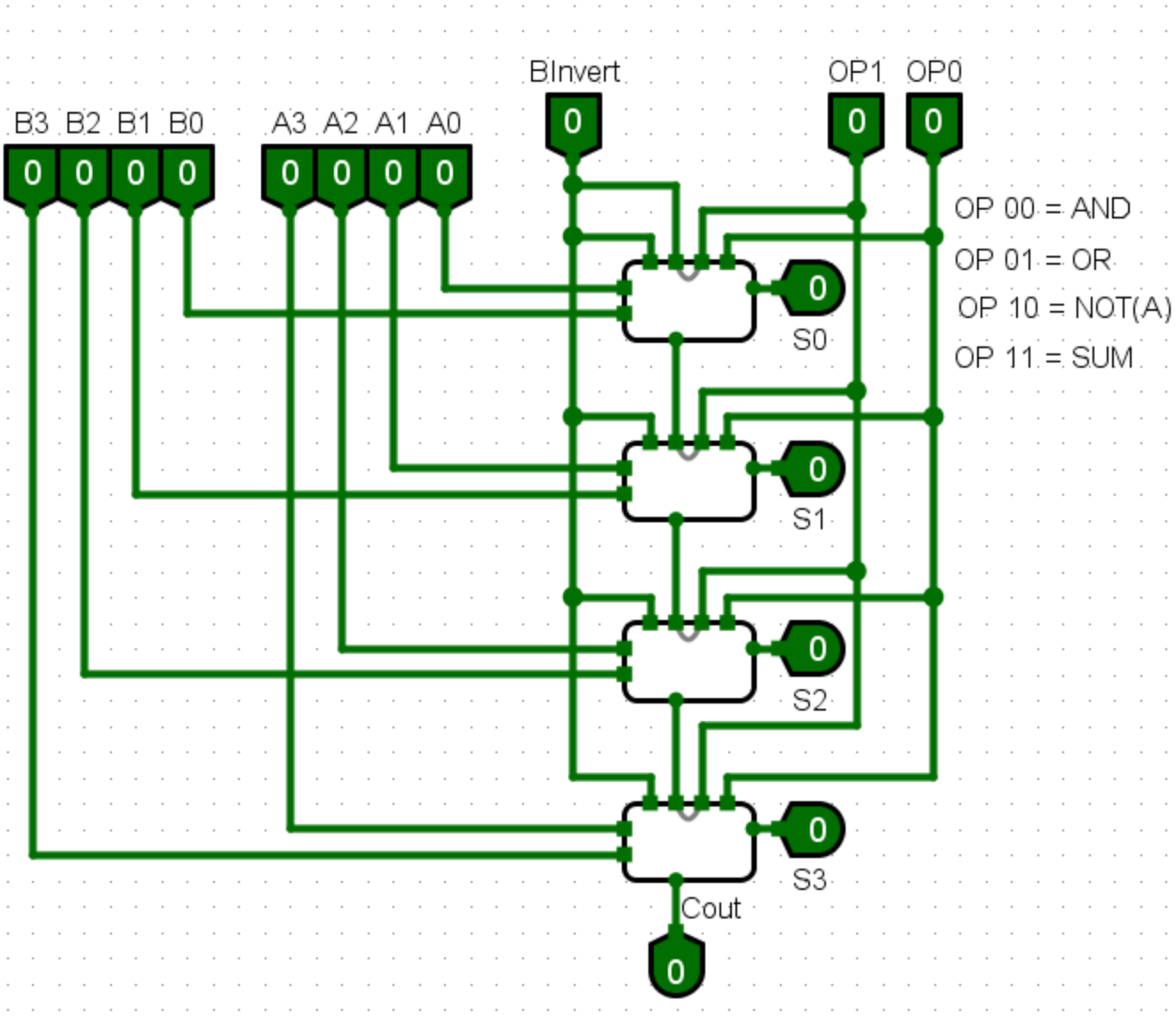
NOT(A) -> A=1


CarryIn
Op1
Op0
BInvert
0
1
0
A
1
B
1
MUX_2x1
Full_Adder
MUX_4x2
CarryOut
1

SOMA(A,-B) → A = 1, B = -(1)

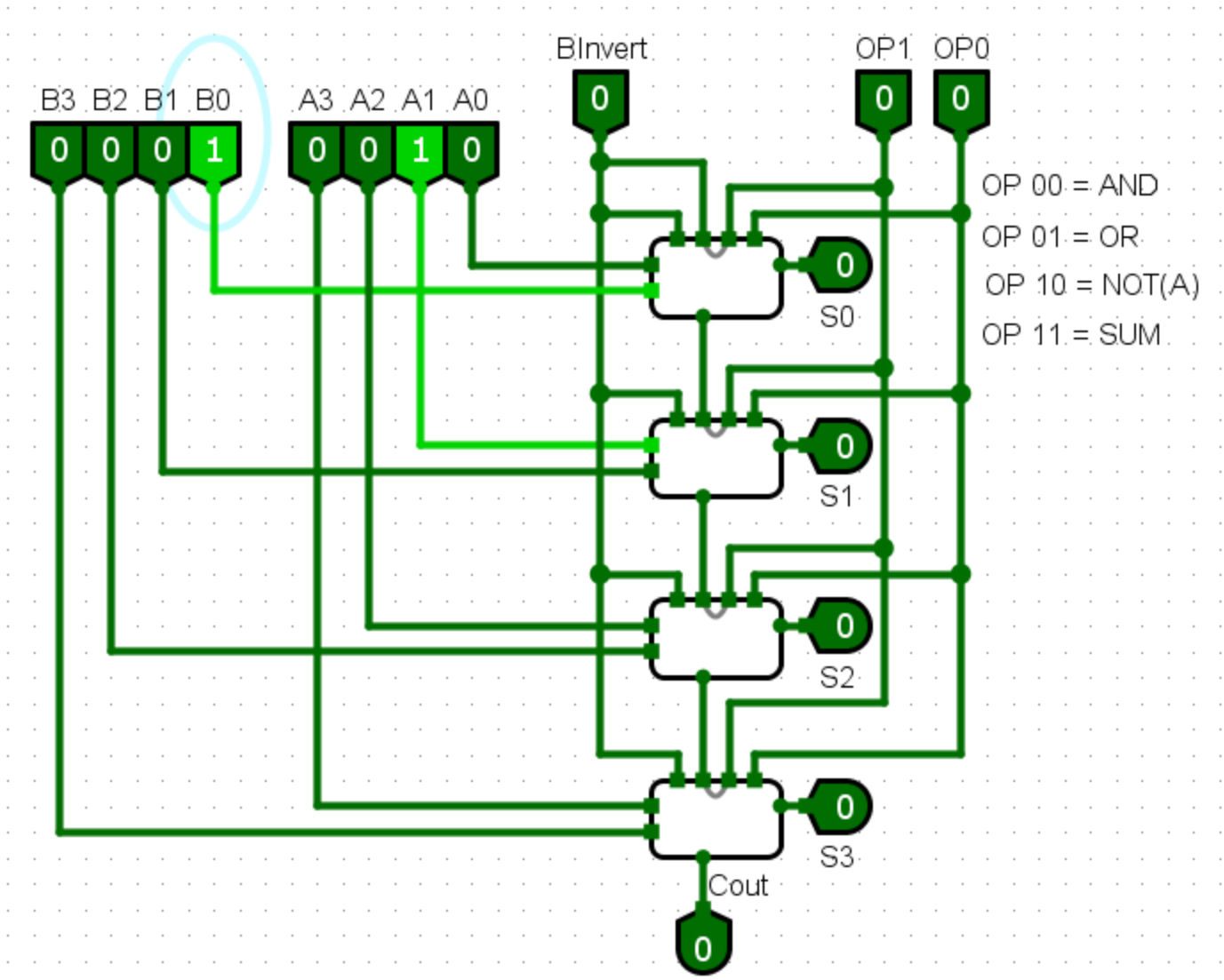


ULA 4 bits: circuito combinatório responsável pela execução de somas, subtrações e funções lógicas

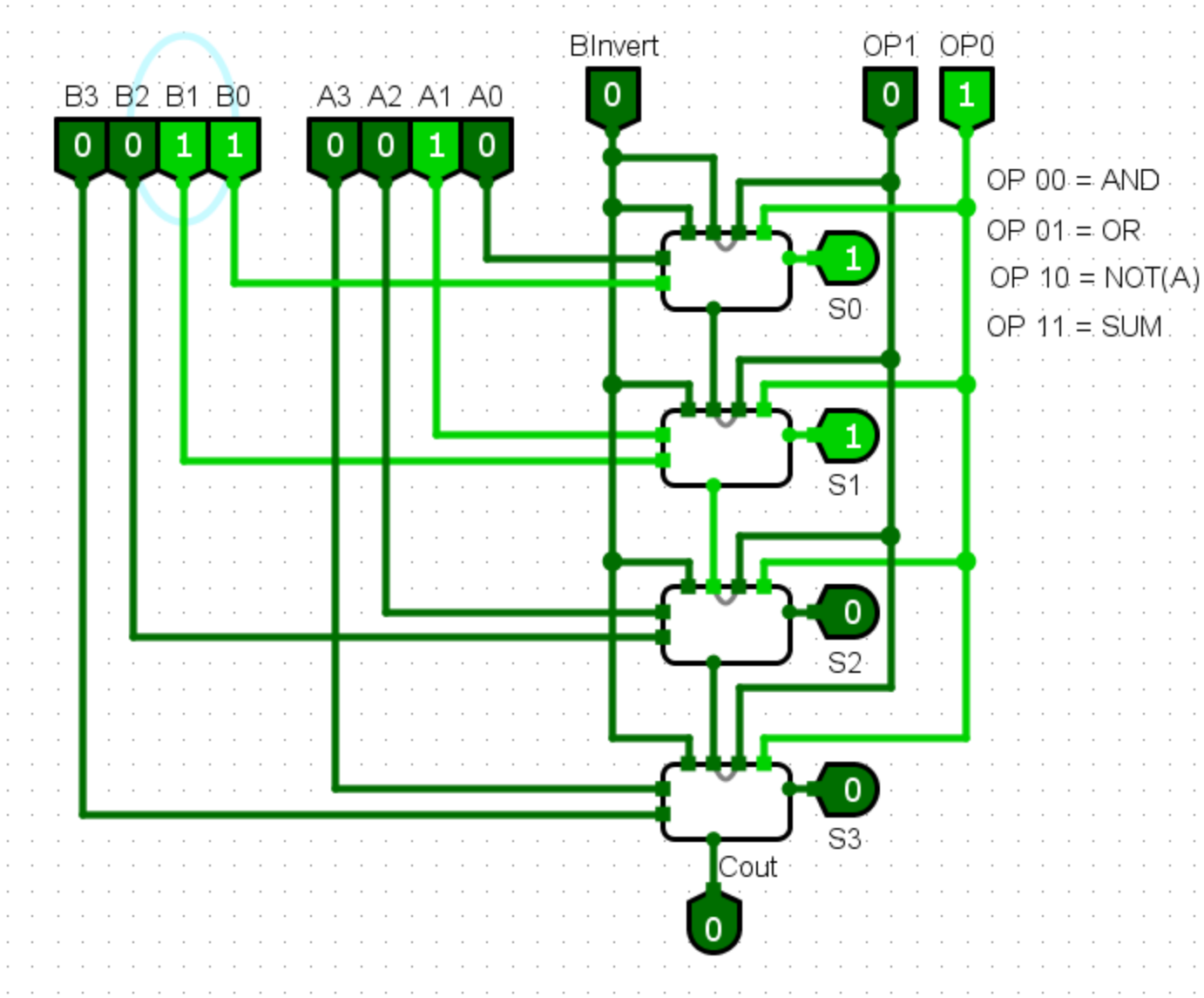


Testes da ULA 4 bits

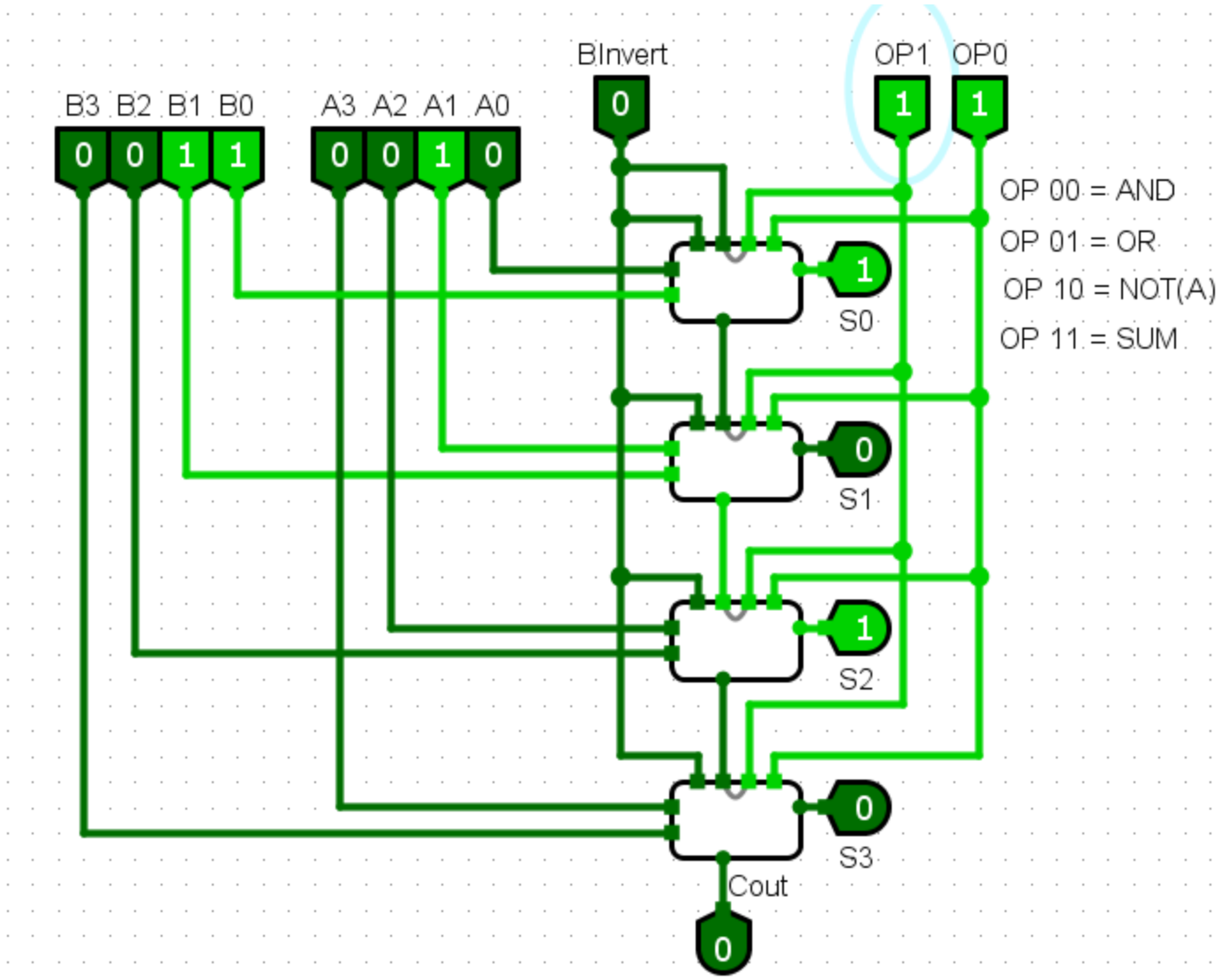
AND(A,B) → A = 2, B =1


B3 B2 B1 B0
0 0 0 1
A3 A2 A1 A0
0 0 1 0
BInvert
0
OP1 OP0
0 0
OP 00 = AND
OP 01 = OR
OP 10 = NOT(A)
OP 11 = SUM
0
S0
0
S1
0
S2
0
S3
Cout
0

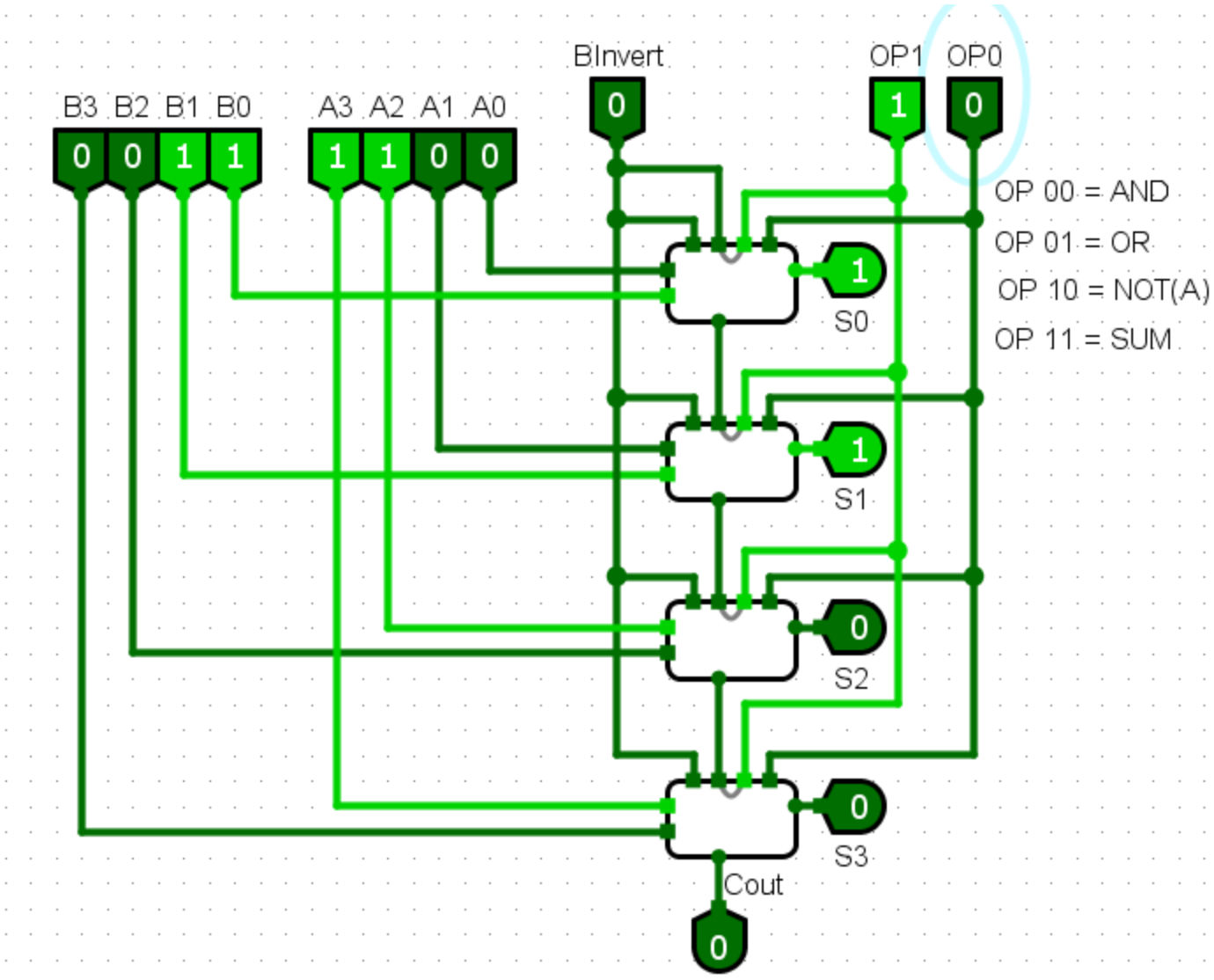
OR(A,B)→A=2,B=3


B3 B2 B1 B0
0 0 1 1
A3 A2 A1 A0
0 0 1 0
BInvert
0
OP1 OP0
0 1
OP 00 = AND
OP 01 = OR
OP 10 = NOT(A)
OP 11 = SUM
1
S0
1
S1
0
S2
0
S3
Cout
0

SOMA(A,B) → A = 2, B = 3


BInvert
OP1
OP0
B3 B2 B1 B0
A3 A2 A1 A0
0
1
1
0 0 1 1
0 0 1 0
OP 00 = AND
OP 01 = OR
OP 10 = NOT(A)
OP 11 = SUM
1
S0
0
S1
1
S2
0
S3
Cout
0

NOT(A) → A = 12



AND(A, B) → A = 12, B = 13

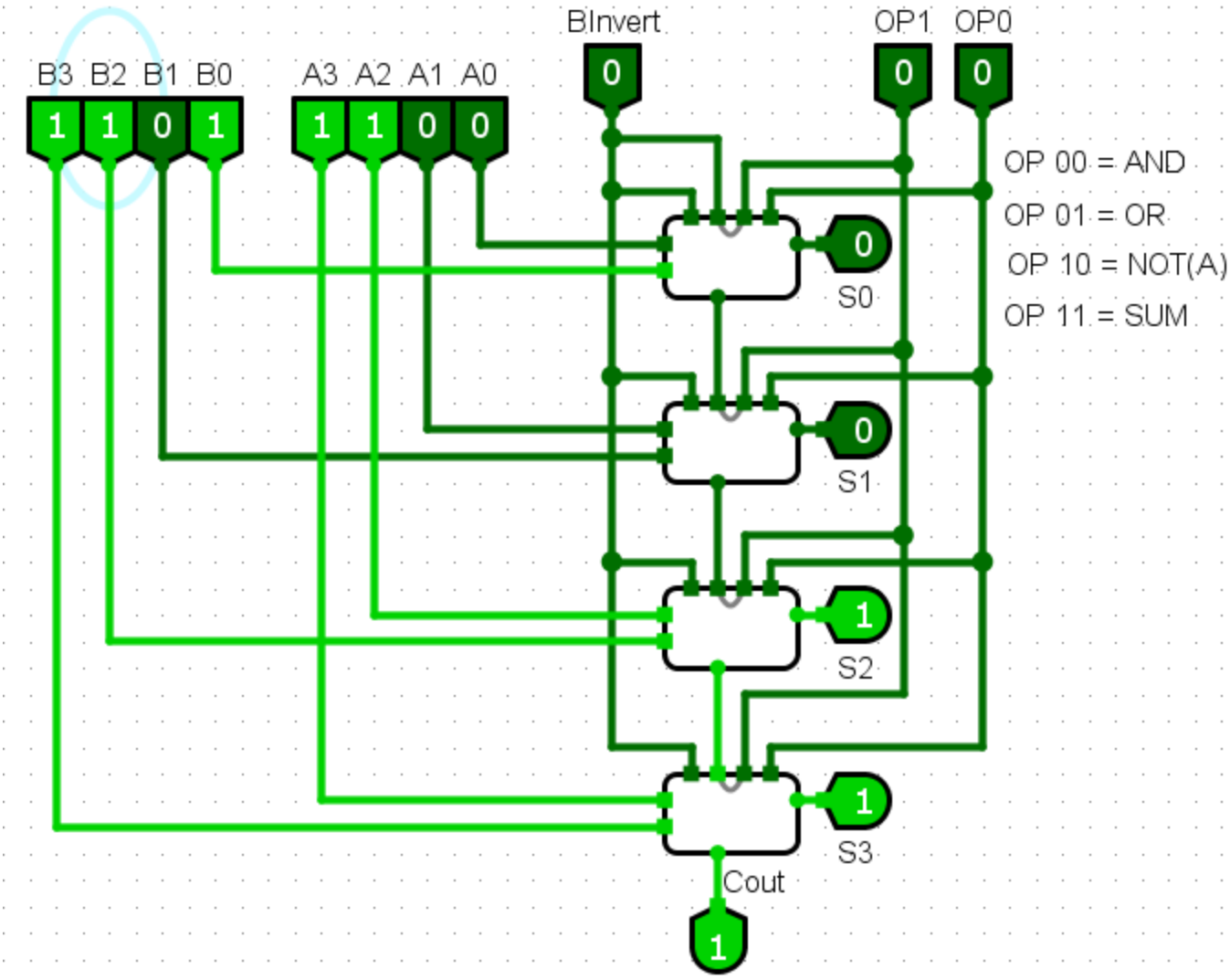
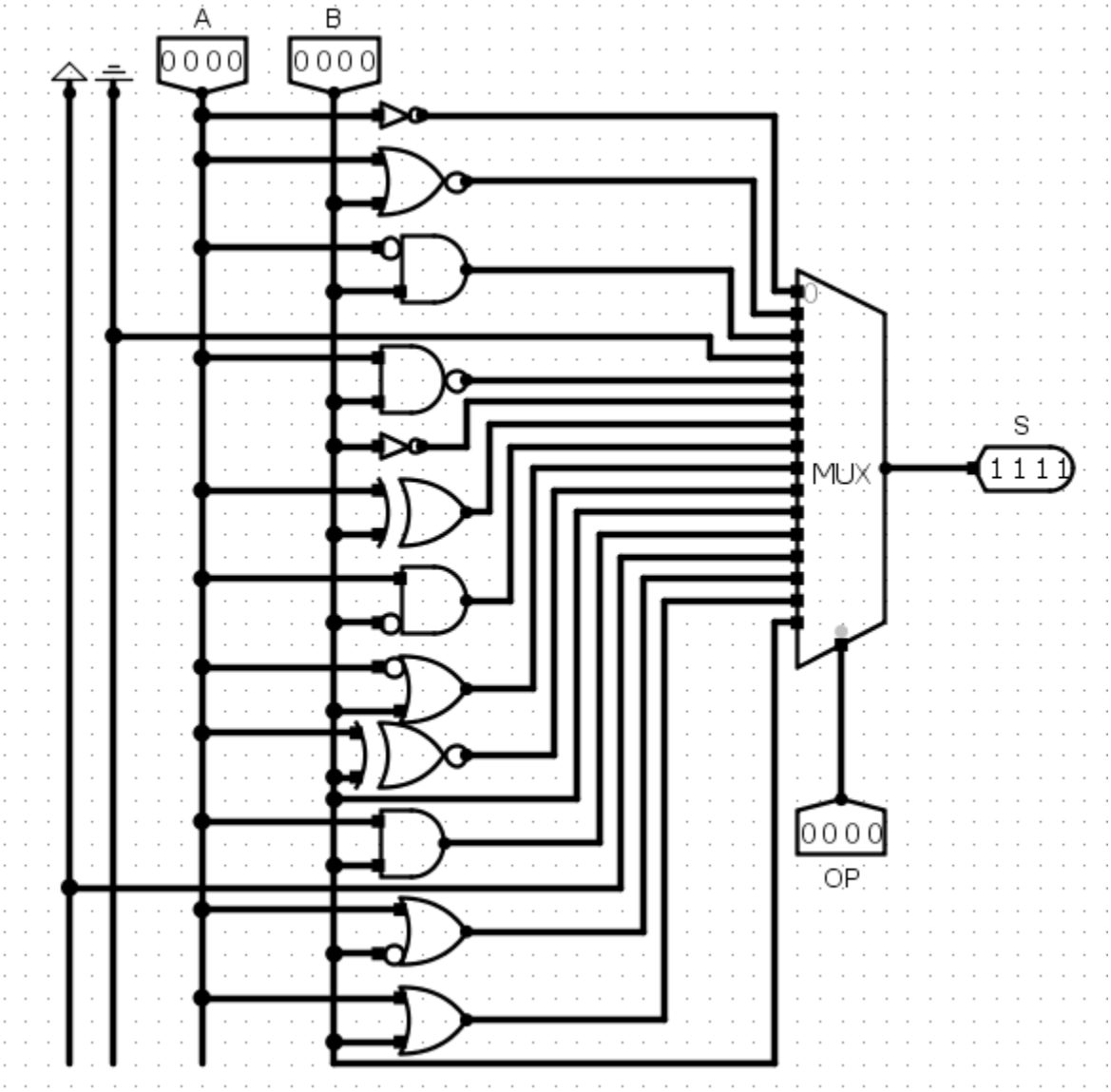


TABELA DE TESTES DA ULA 4 BITS

ão realizada	Binário (A, B, OP)	Valor em Hexa	Resultado em Binario
B)	0010, 0001, 00	0x084	0000
)	0010, 0011, 01	0x08D	0011
A,B)	0010, 0011, 11	0x08F	0011
	1100, 0011, 10	0x30E	0011
A)	1100, 1101, 00	0x334	1100

Parte 2-

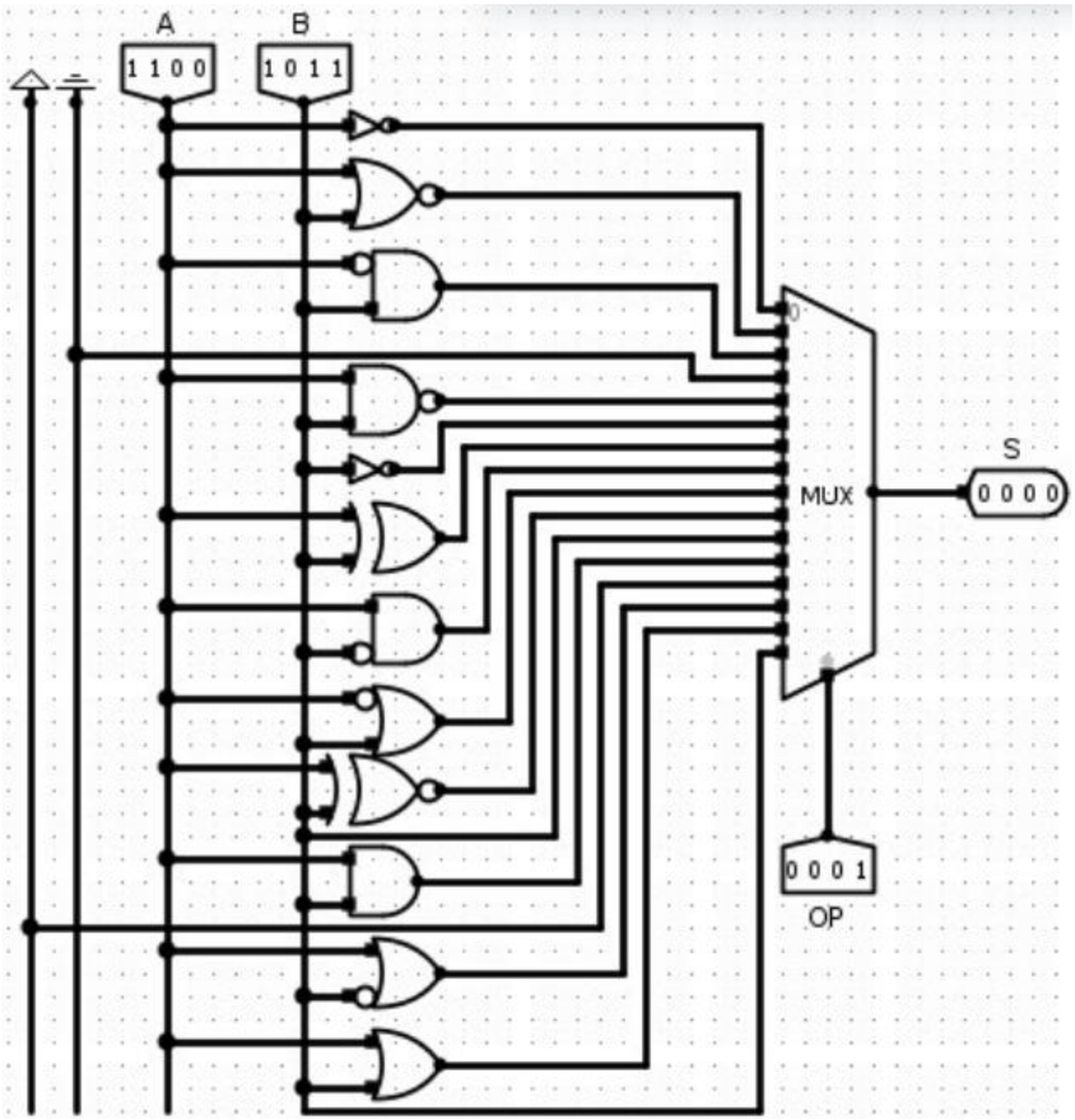
ULA 74LS181: circuito combinatório responsável pela execução de somas, subtrações e funções lógicas.



Instrucao	Binario	Resultado
450	010001010000	1011
CB1	110010110001	0000
A32	101000110010	0001
C43	110001000011	0000
124	000100100100	1111
785	011110000101	0111

9B6	100110110110	0010
CD7	110011010111	0000
FE8	111111101000	1110
649	011001001001	1101
D9A	110110011010	1001
FCB	111111001011	1100
63C	011000111100	1111
98D	100110001101	1111
76E	011101101110	0111
23F	001000111111	0011

Teste (CB1) -> (OR(A,B))’ A=1100, B = 1011 /./ OP = 0000



Pergunta:

A tabela verdade teria 4096 linhas, pois são 4 bits da entrada A, 4 bits da entrada B e 4 bits de seleção, totalizando 12 bits, ou seja, 2 elevado a 12 combinações.