

Exercício Prático 2 Parte 1

Laboratório de AC2

Arthur Martinho Medeiros Oliveira - 813168

Daniel Salgado Magalhães - 821469

Objetivo:

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

Parte 1 (estudo da ALU usando Logisim):

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):

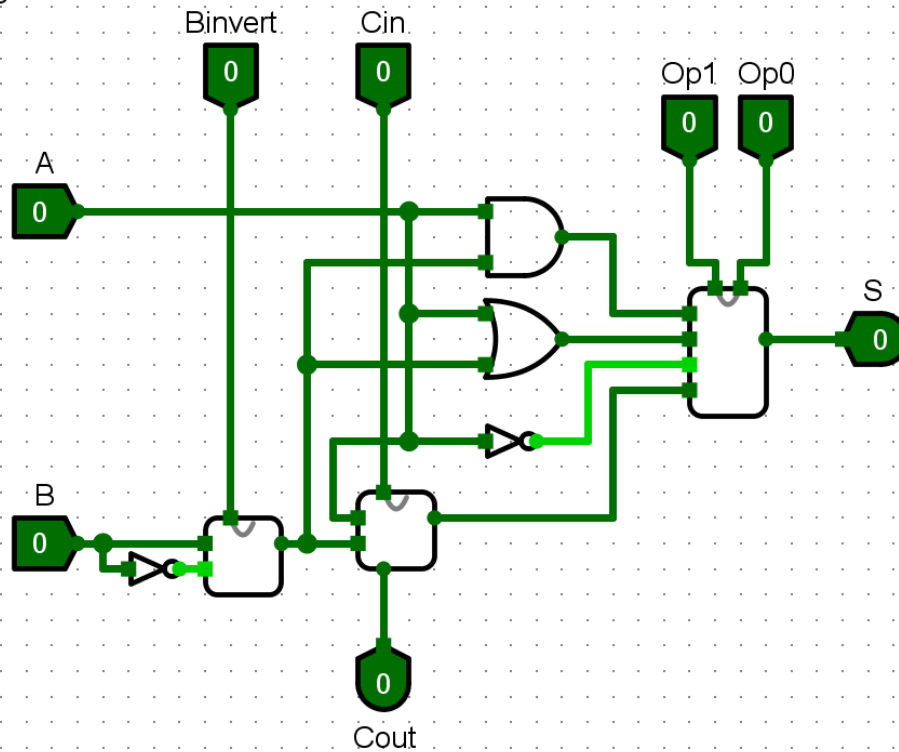
| Instrução realizada | Binário (A,B,Op.code) | Valor em Hexa (0x ...) | Resultado em binário |
|---------------------|--------------------------|---------------------------|-------------------------|
| AND(A,B) | 0010 0001 00 | (0000 1000 0100) = 0x084 | 0000 |
| OR(A,B) | 0010 0011 01 | (0000 1000 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) | 1100 1101 00 | (0011 0011 0100) = 0x334 | 1100 |

Os prints de todos os circuitos elaborados (ULA de 1 bit e ULA de 4 bits).

ULA 1 BIT

Arthur Martinho - 813168

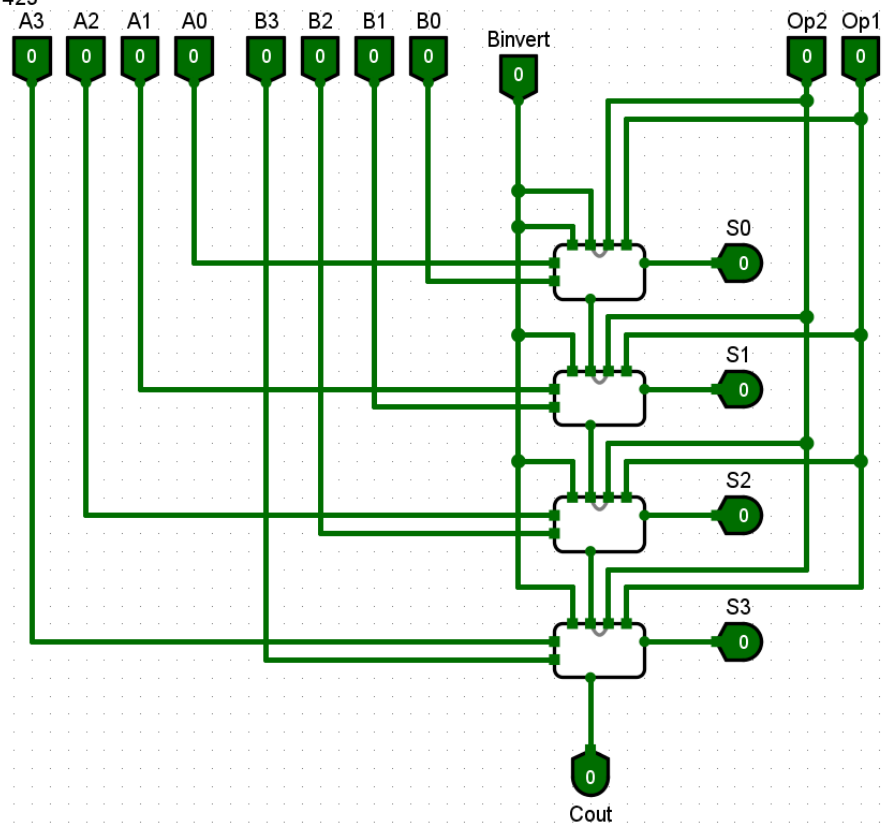
Daniel Salgado - 821429



ULA 4 BITS

Arthur Martinho - 813168

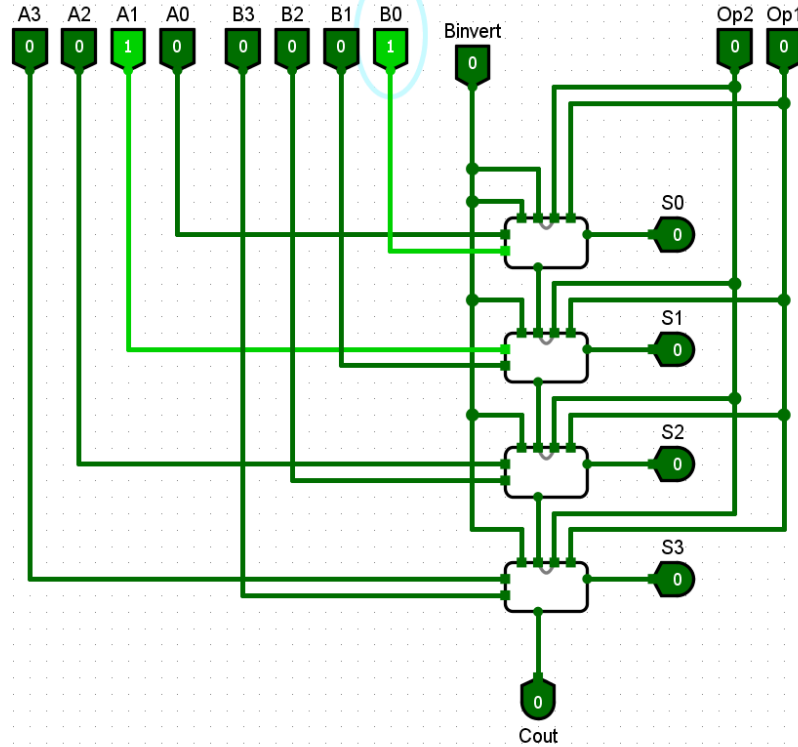
Daniel Salgado - 821429



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.

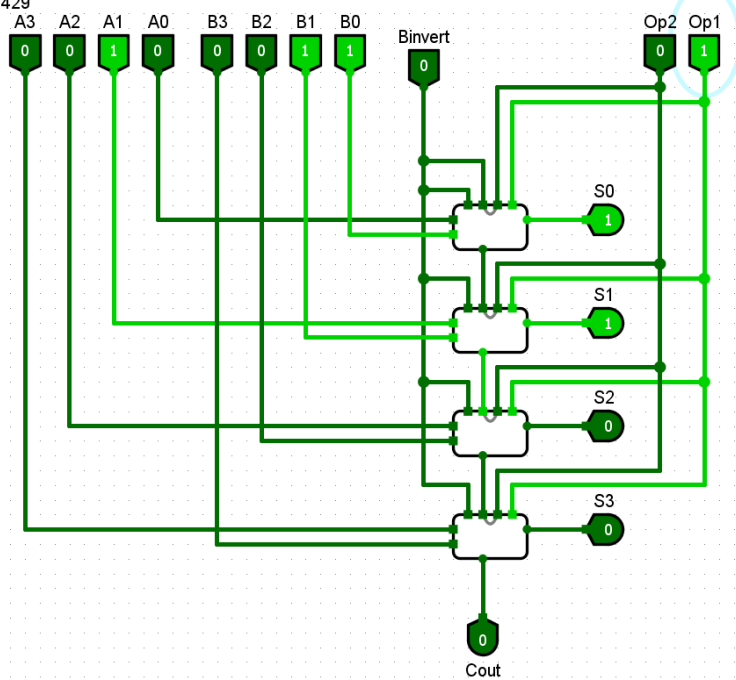
AND(A,B)

Arthur Martinho - 813168
Daniel Salgado - 821429



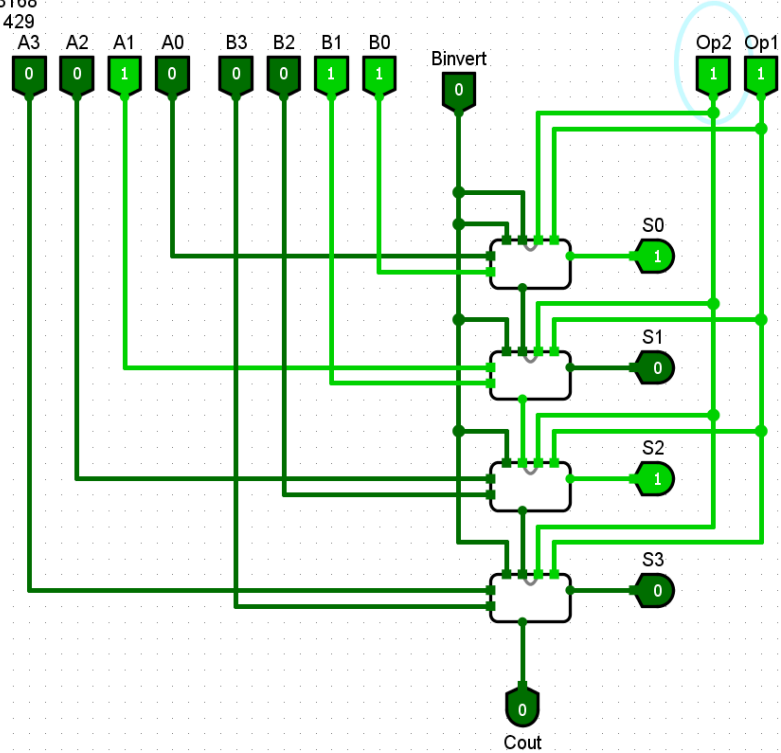
OR(A,B)

Arthur Martinho - 813168
Daniel Salgado - 821429



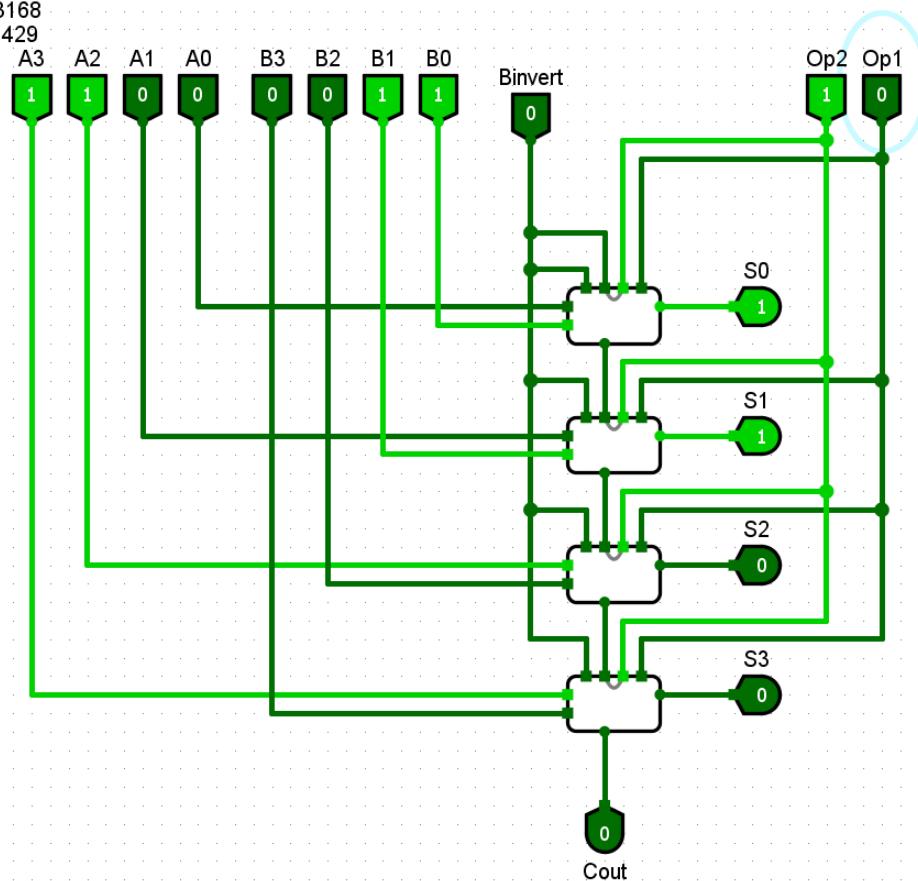
SOMA(A,B)

Arthur Martinho - 813168
Daniel Salgado - 821429



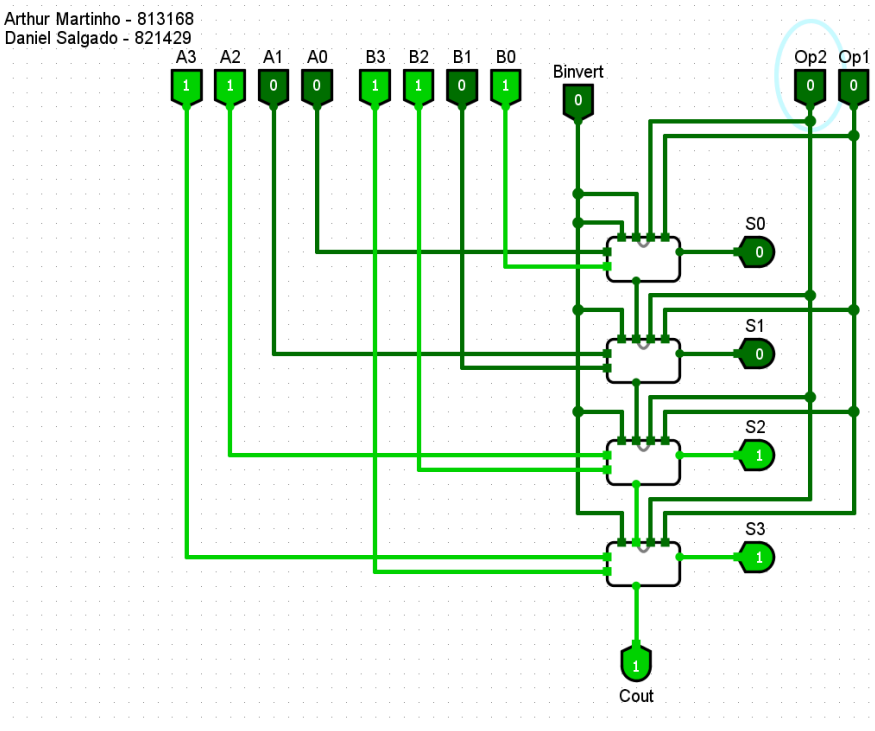
NOT(A)

Arthur Martinho - 813168
Daniel Salgado - 821429



AND(B,A)

Arthur Martinho - 813168
Daniel Salgado - 821429



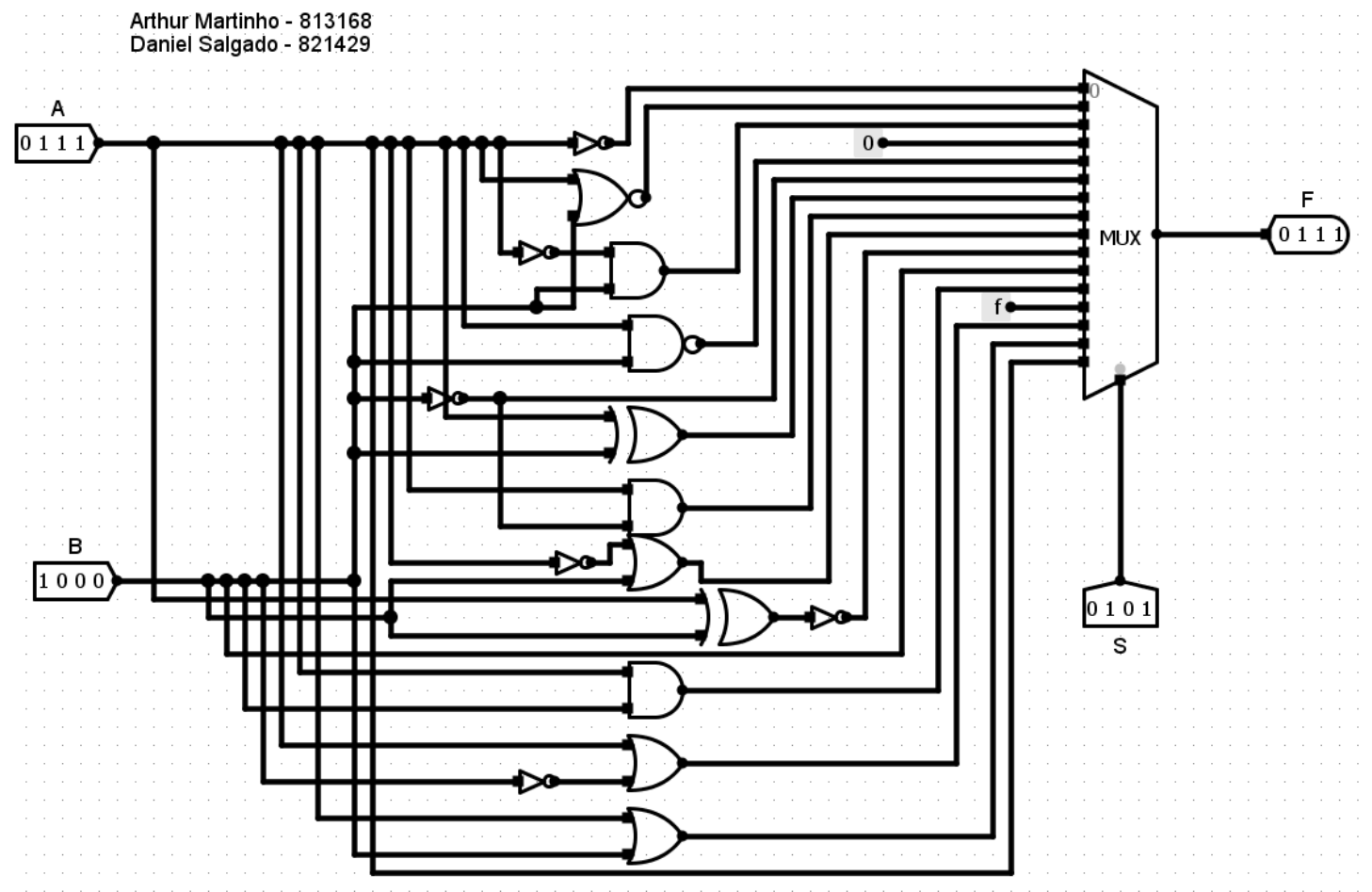
Parte 2

Complete agora a tabela a seguir onde todas as instruções que a ULA pode fazer serão testadas.

| Instruções | Binário | Resultado da operação |
|------------|----------------|-----------------------|
| 450 | 0100 0101 0000 | 1011 - B |
| CB1 | 1100 1011 0001 | 0000 - 0 |
| A32 | 1010 0011 0010 | 0001 - 1 |
| C43 | 1100 0100 0011 | 0000 - 0 |
| 124 | 0001 0010 0100 | 1111 - F |
| 785 | 0111 1000 0101 | 0111 - 7 |
| 9B6 | 1001 1011 0110 | 0010 - 2 |
| CD7 | 1100 1101 0111 | 0000 - 0 |
| FE8 | 1111 1110 1000 | 1110 - E |
| 649 | 0110 0100 1001 | 1101 - D |
| D9A | 1101 1001 1010 | 1001 - 9 |
| FCB | 1111 1100 1011 | 1100 - C |
| 63C | 0110 0011 1100 | 1111 - F |
| 98D | 1001 1000 1101 | 1111 - F |
| 76E | 0111 0110 1110 | 0111 - 7 |
| 23F | 0010 0011 1111 | 0010 - 2 |

O projeto da ULA no Logisim com um printscreen de alguma instrução da tabela sendo executada.

Execução: $785 = 7$



Responder:

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?

Resposta:

A tabela verdade deveria ter 2^{12} (4096) linhas