

Organização de Computadores

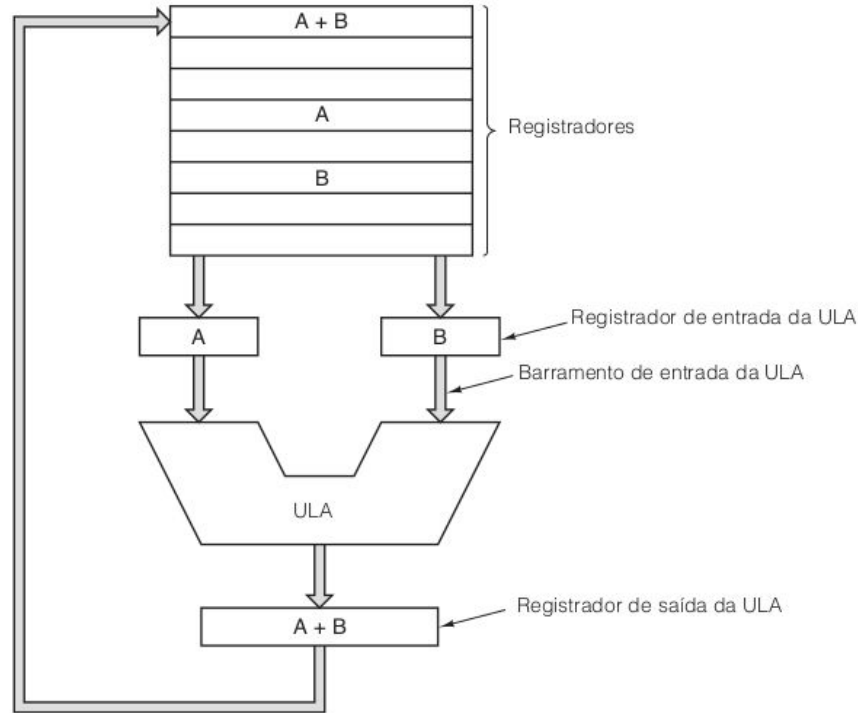
Versão Hands-on com Logisim

Prof. Juan G. Colonna
juancolonna@icomp.ufam.edu.br
Instituto de Computação (IComp)
Universidade Federal do Amazonas (UFAM)
Semestre 2024/01

Caminho dos dados

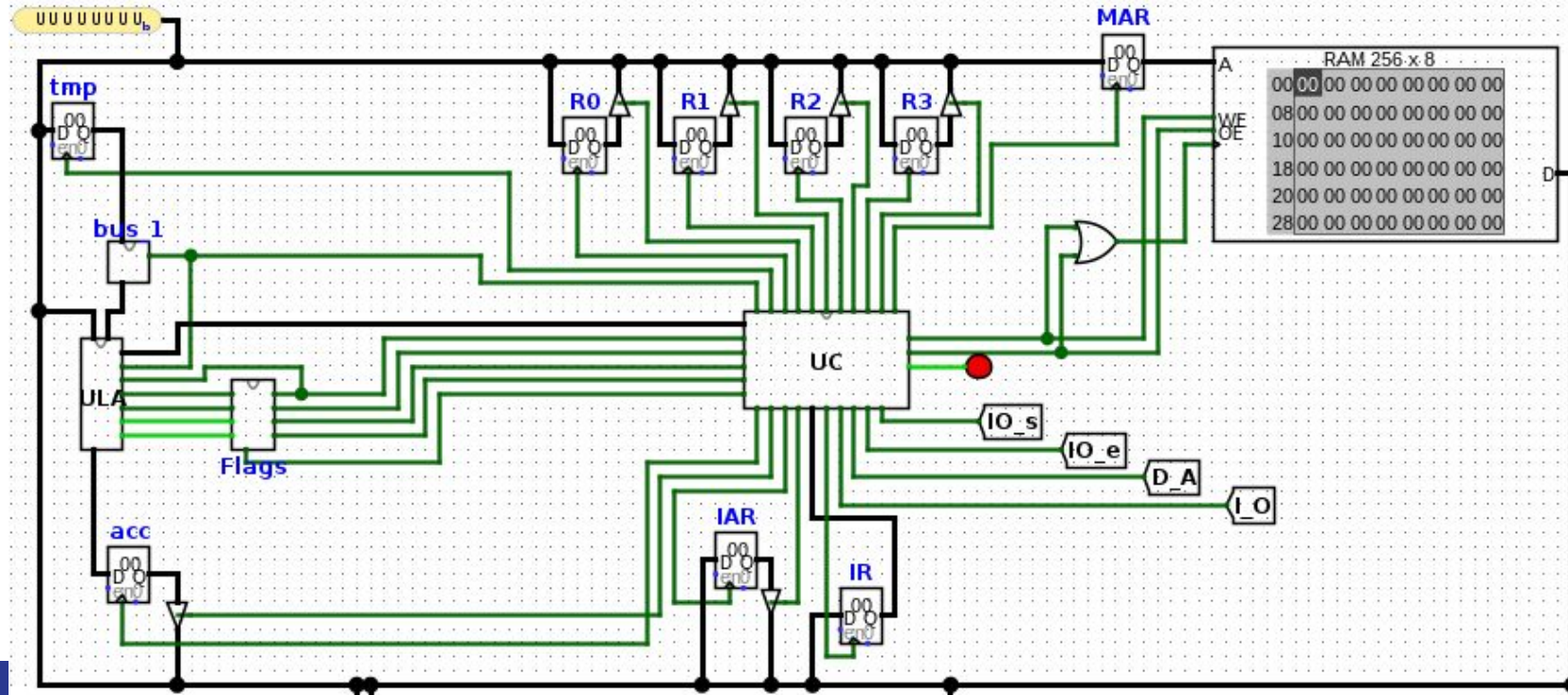
Caminho de Dados

A parte principal, chamada caminho de dados, compreende os registradores e a ULA, onde as operações ocorrem usando os dados dos registradores



Visão geral do computador de 8bits

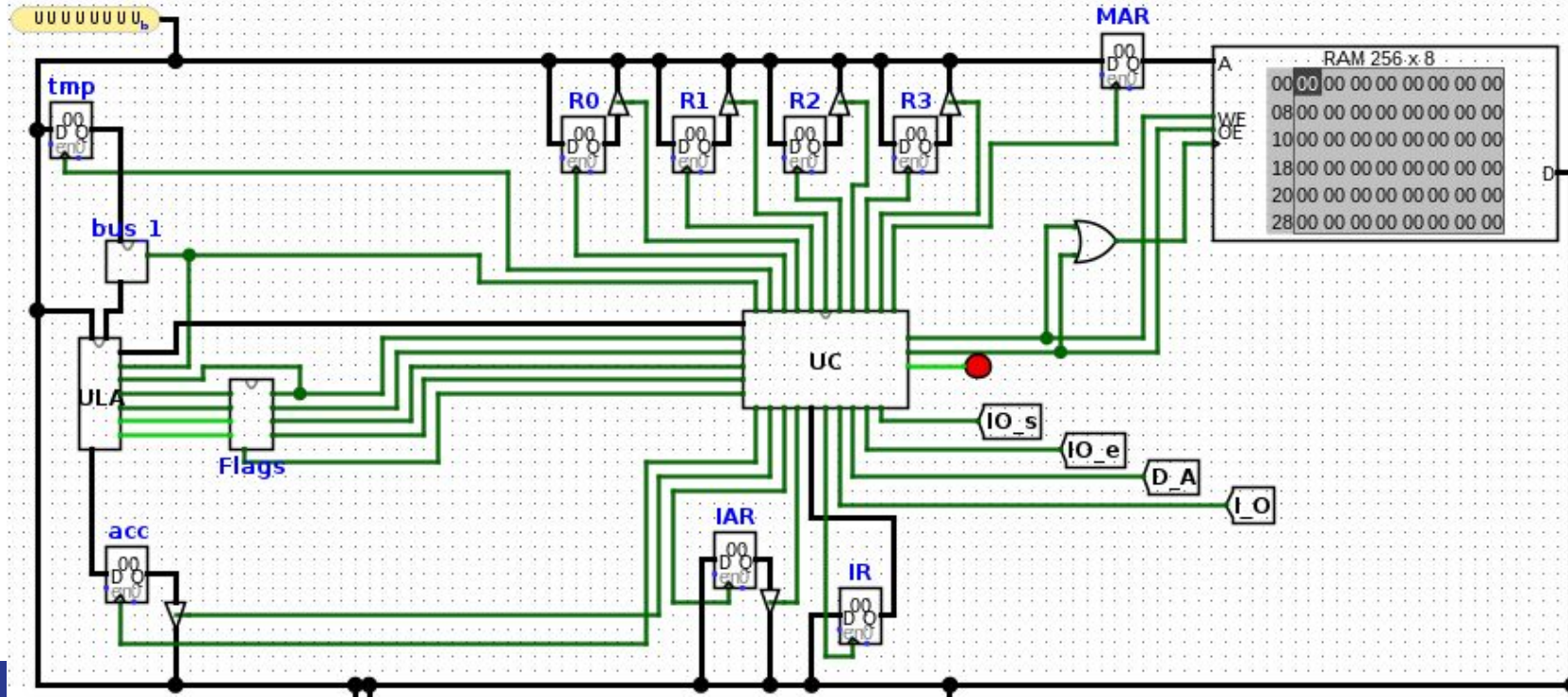
Elementos novos: Unidade controladora, Unidade aritmética, Acumulador, Registrador temporário, barramento de dados



Visão geral do computador de 8bits

Pseudo código com instruções *assembly*:

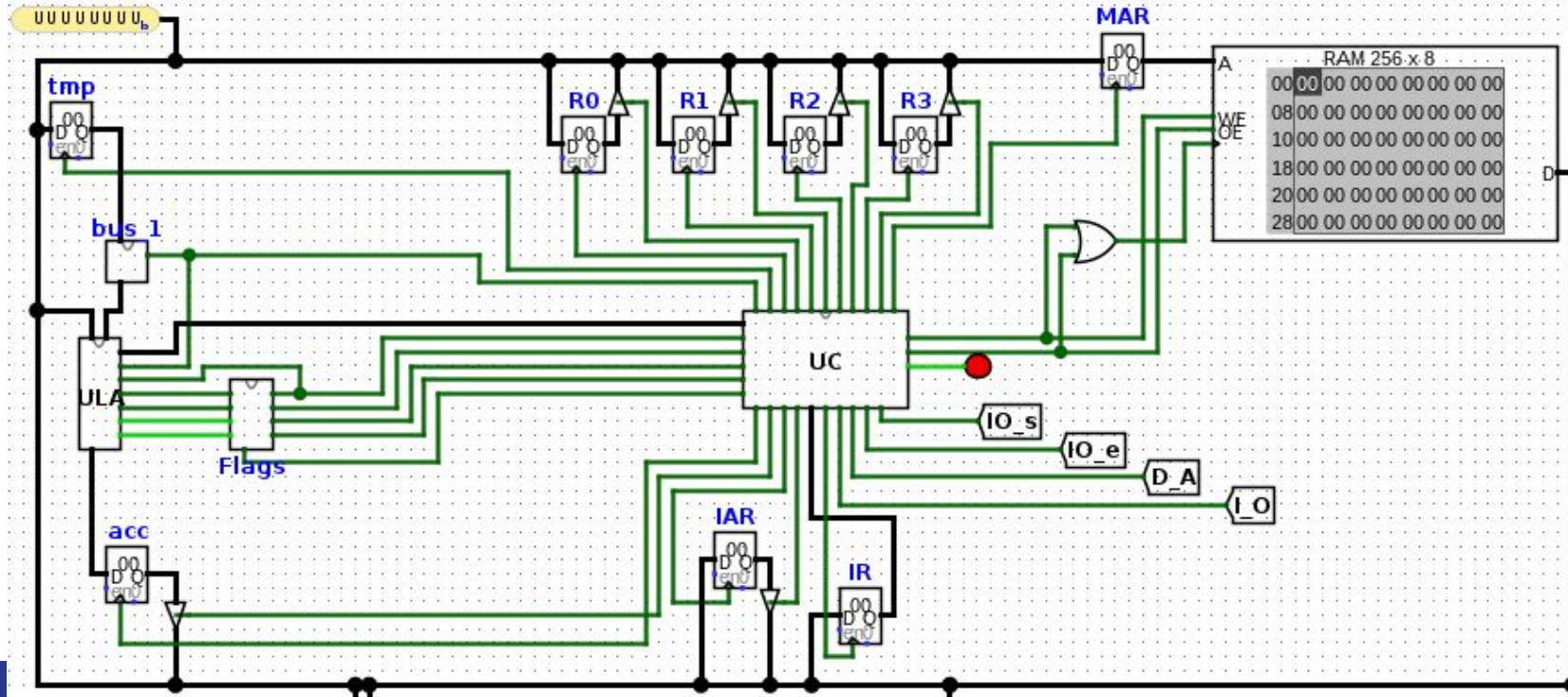
- ADD R0,R1,R2
- ADD R0,R1,R0
- SUB R0,R1,R0



Visão geral do computador de 8bits

Pseudo código com instruções *assembly*:

- LD R0,R1
- STR R0,R1



Visão geral do computador de 8bits

Quantas instruções assembly podemos ter?

Quantas instruções aritméticas podemos ter?



Visão geral do computador de 8bits

Quantas instruções assembly podemos ter?

Quantas instruções aritméticas podemos ter?

| Tipo de instrução | code | OP |
|-------------------|------|-----|
| 1 | 000 | ADD |
| 1 | 001 | SHR |
| 1 | 010 | SHL |
| 1 | 011 | NOT |
| 1 | 100 | AND |
| 1 | 101 | OR |
| 1 | 110 | XOR |
| 1 | 111 | CMP |



Visão geral do computador de 8bits

Quantas instruções assembly podemos ter?

Quantas instruções aritméticas podemos ter?

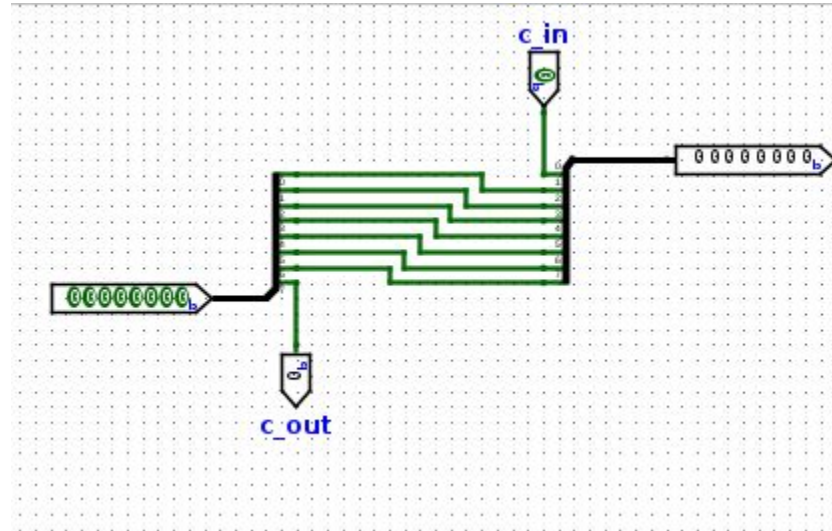
| Tipo de instrução | code | OP |
|-------------------|------|-----|
| 1 | 000 | ADD |
| 1 | 001 | SHR |
| 1 | 010 | SHL |
| 1 | 011 | NOT |
| 1 | 100 | AND |
| 1 | 101 | OR |
| 1 | 110 | XOR |
| 1 | 111 | CMP |

| Tipo de instrução | code | OP |
|-------------------|------|-----|
| 0 | ??? | LD |
| 0 | ??? | STR |
| | | ? |
| | | |
| | | |
| | | |
| | | |
| | | |

Componentes da ULA

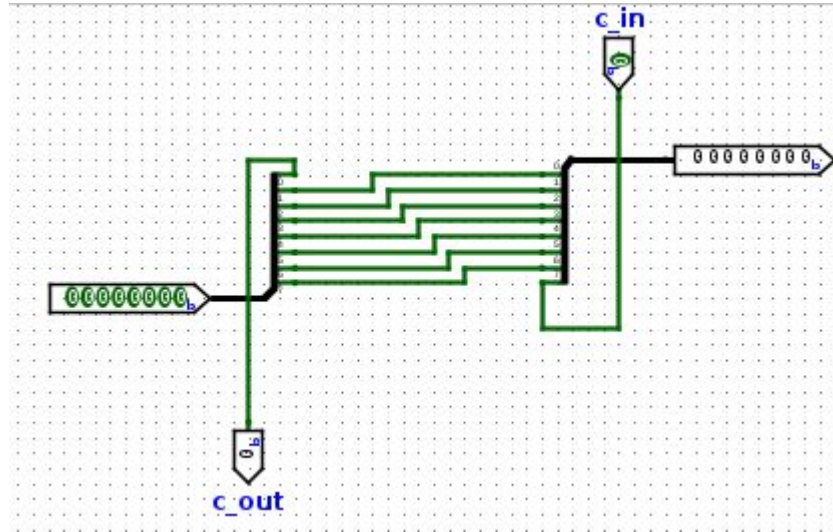
Shifter

Left shift: 0100 0010 \rightarrow 1000 0100

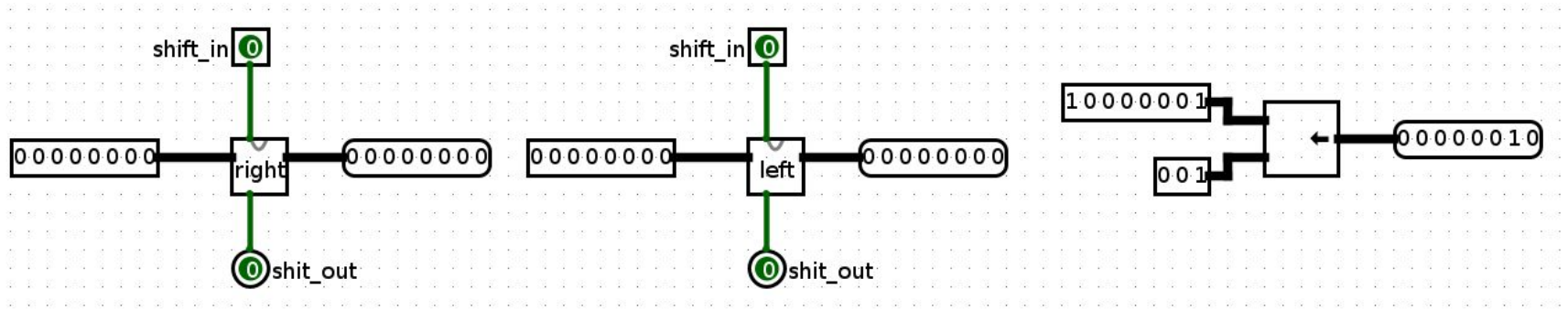


Shifter

Right shift: 0100 0010 \rightarrow 0010 0001

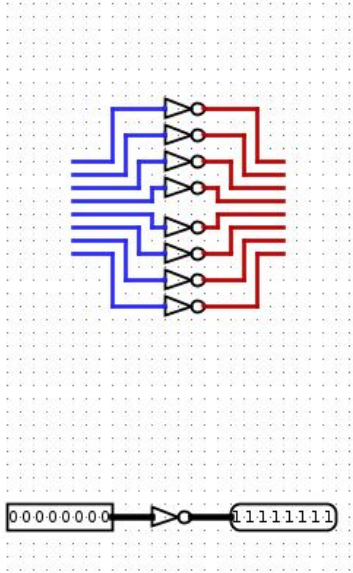


Shifter



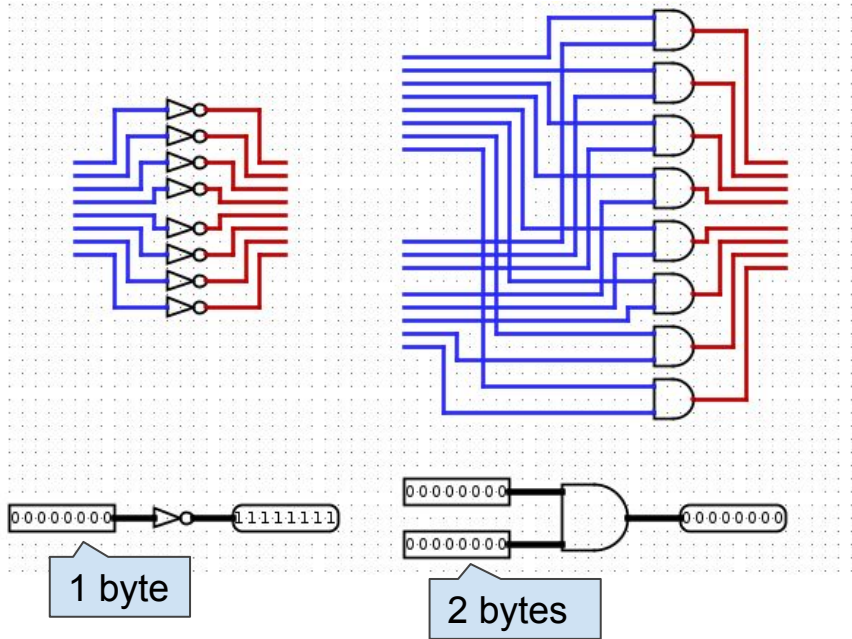
O módulo do Logisim não tem *shift_in* e *shift_out*.

NOTer, ANDer, ORer e XORer



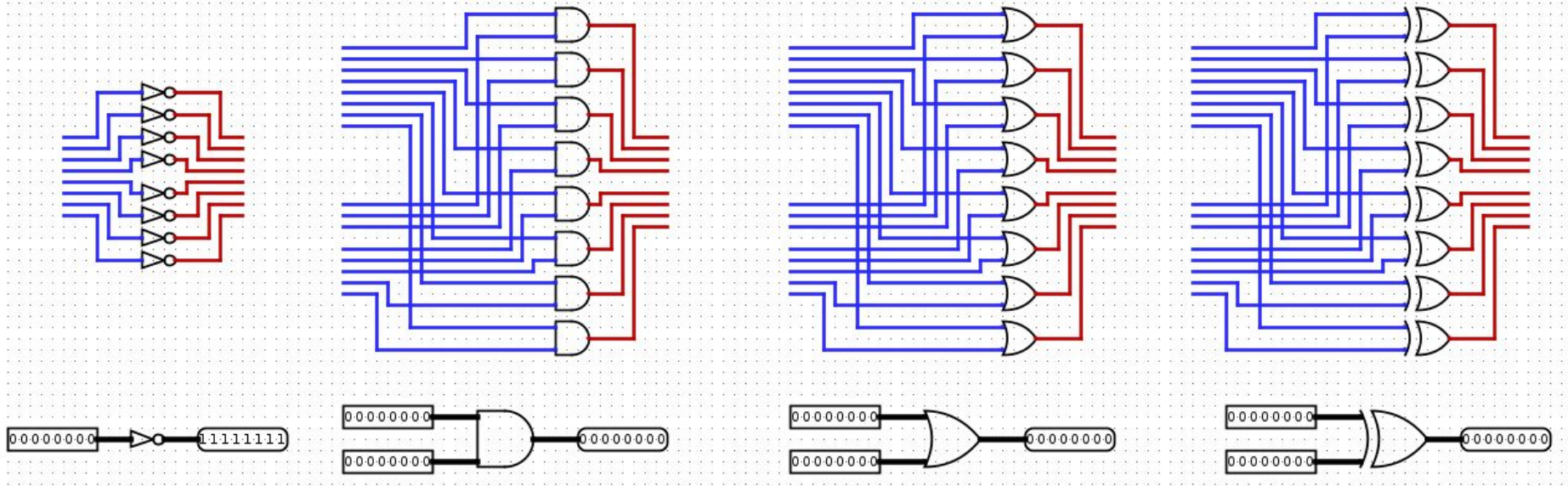
As portas lógicas geralmente funcionam com um bit, mas nossa ULA vai trabalhar com bytes, por isso vamos estender os módulos.

NOTer, ANDer, ORer e XORer



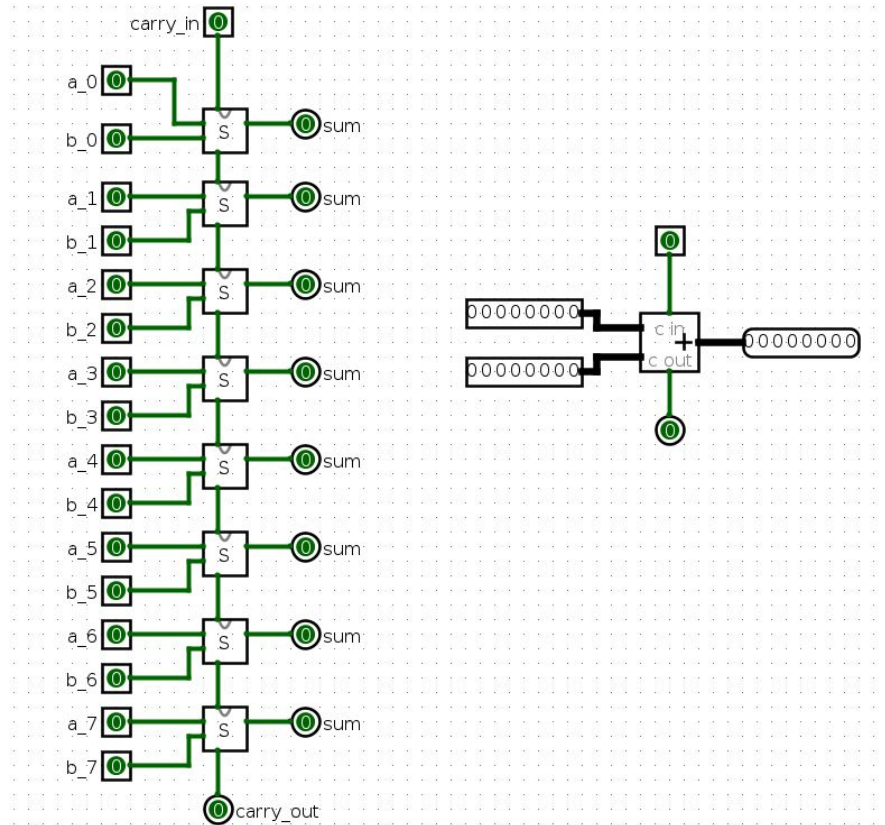
As portas lógicas geralmente funcionam com um bit, mas nossa ULA vai trabalhar com bytes, por isso vamos estender os módulos.

NOTer, ANDer, ORer e XORer

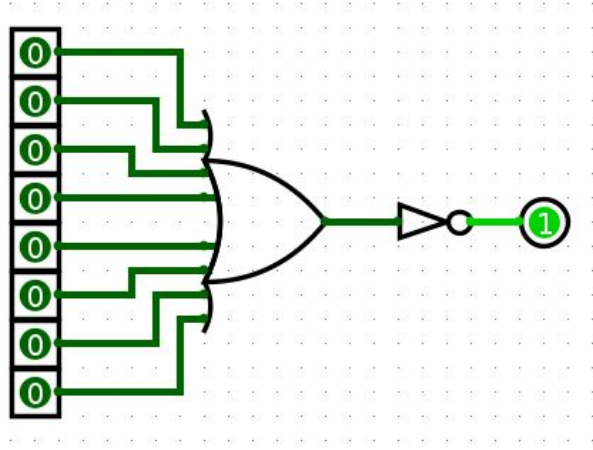


As portas lógicas geralmente funcionam com um bit, mas nossa ULA vai trabalhar com bytes, por isso vamos estender os módulos.

Somador de 8 bits

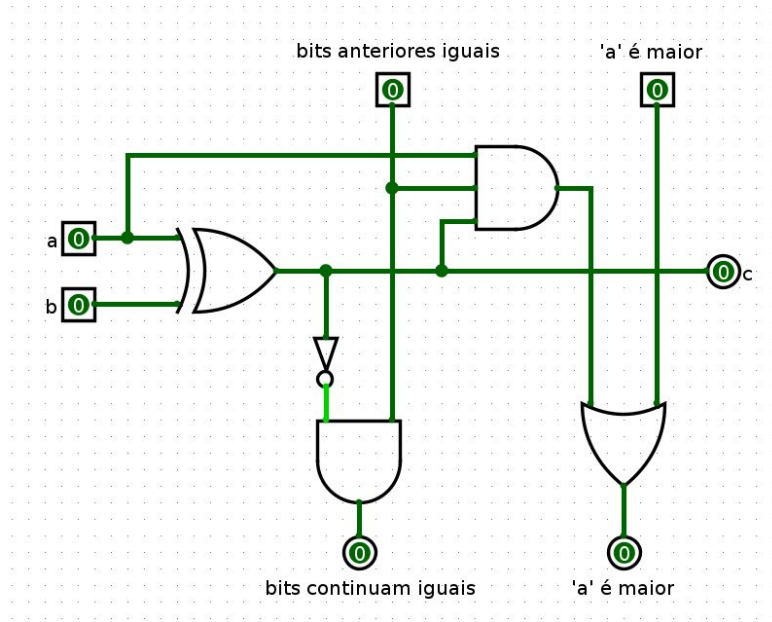


Comparador com zero



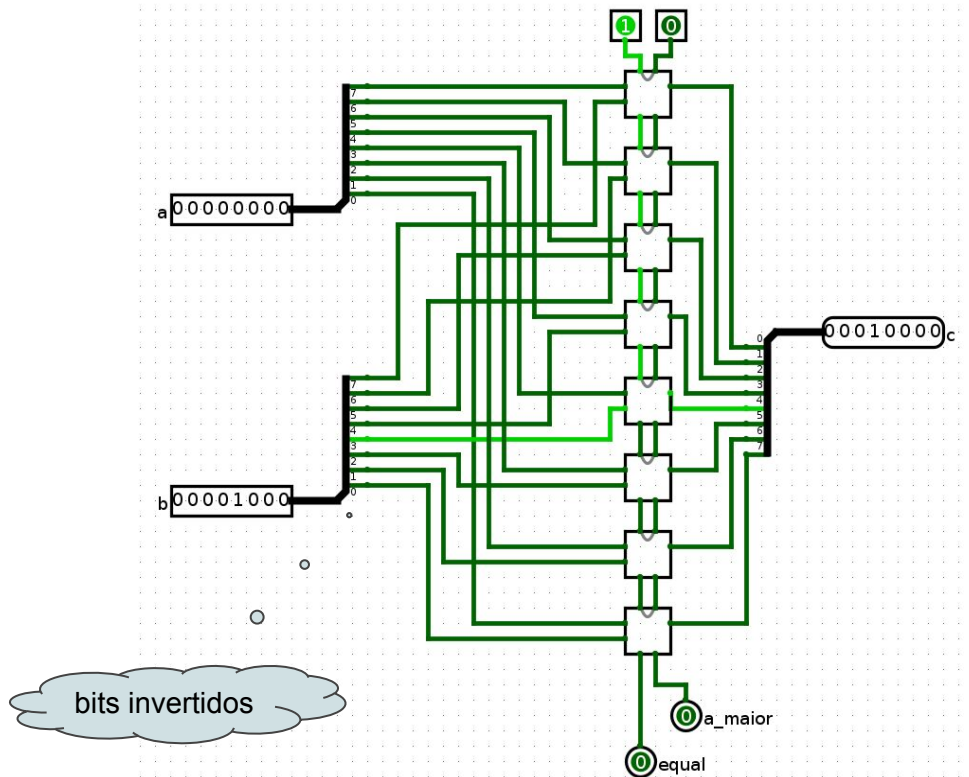
Se algum bit for 1, então a saída do comparador é 0.

Comparador geral

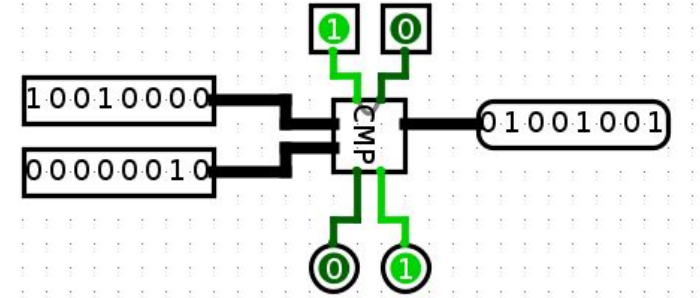
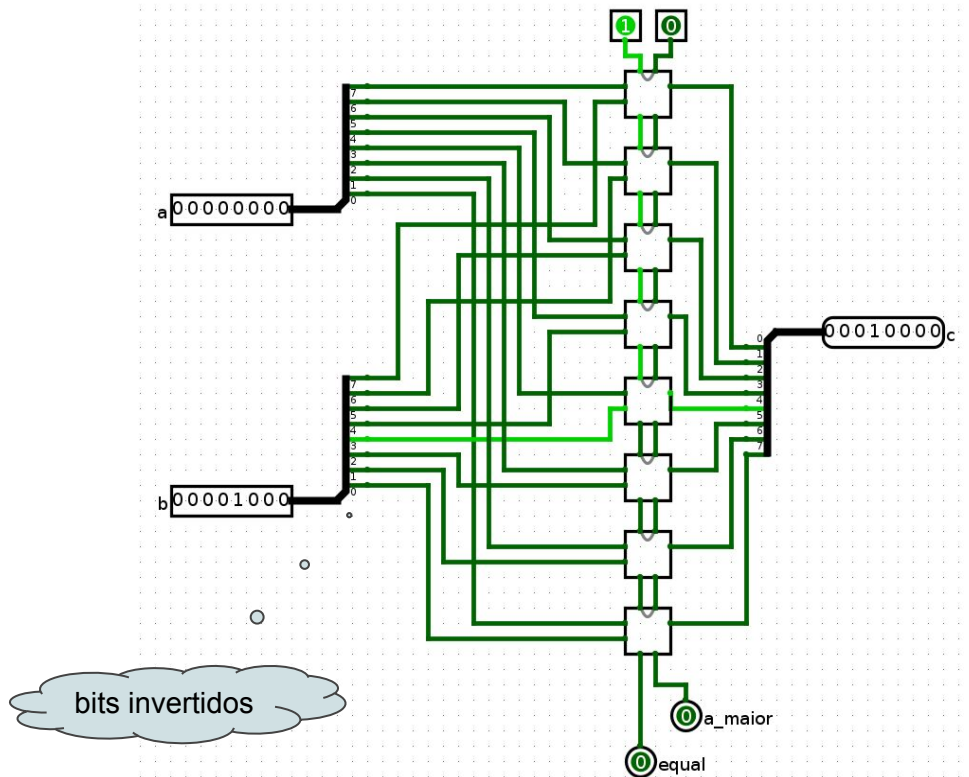


- *c* indica se 'a' e 'b' são diferentes
- *c* -> 1 e 'a' maior -> 0, então 'b' é maior que 'a'
- *c* -> 1 e 'a' maior -> 1, então 'a' é maior que 'b'

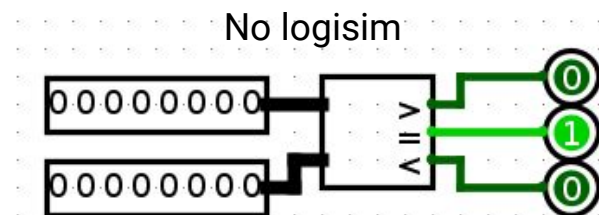
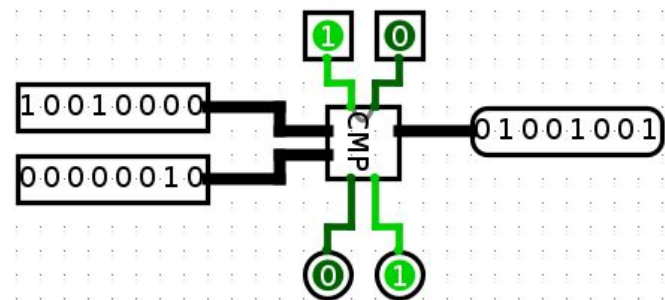
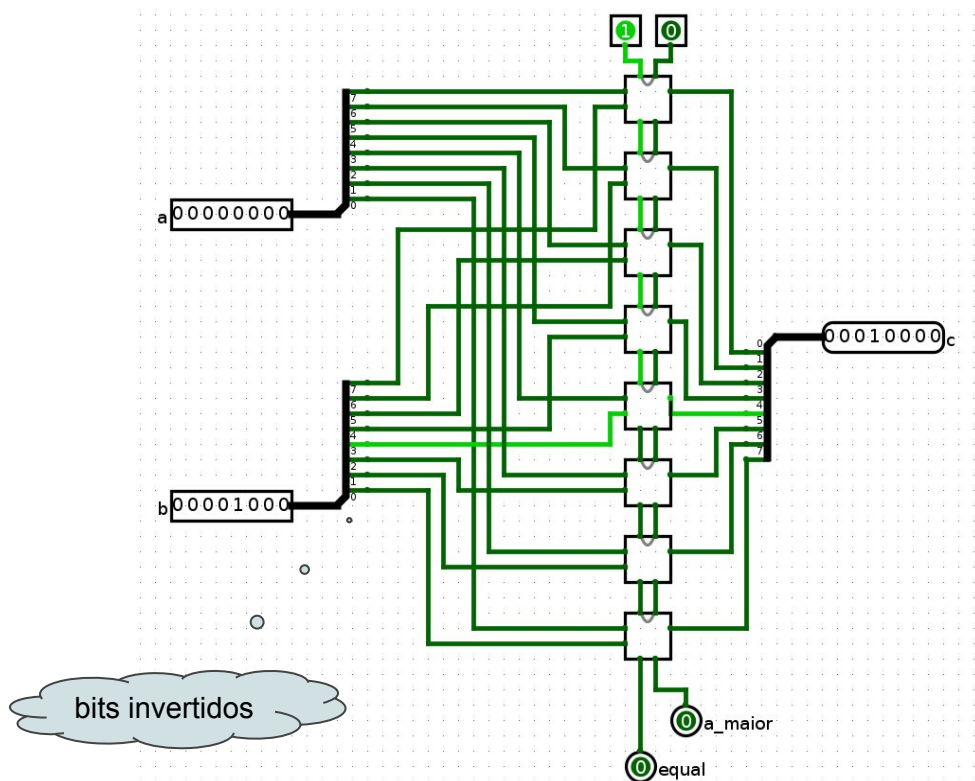
Comparador geral



Comparador geral

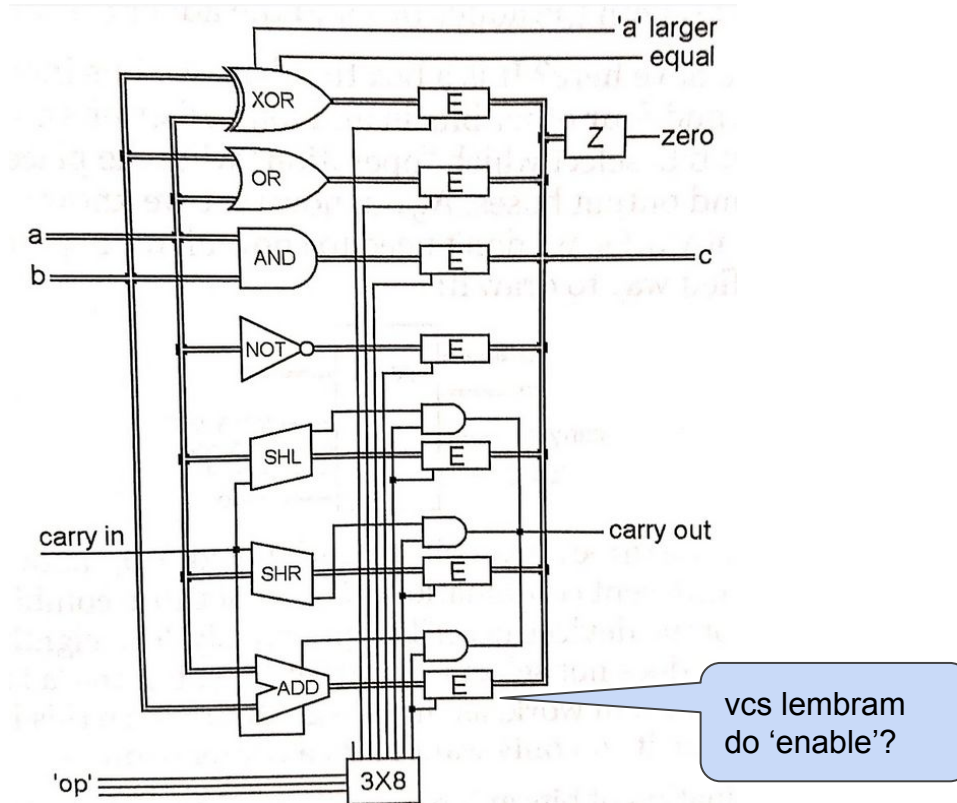


Comparador geral

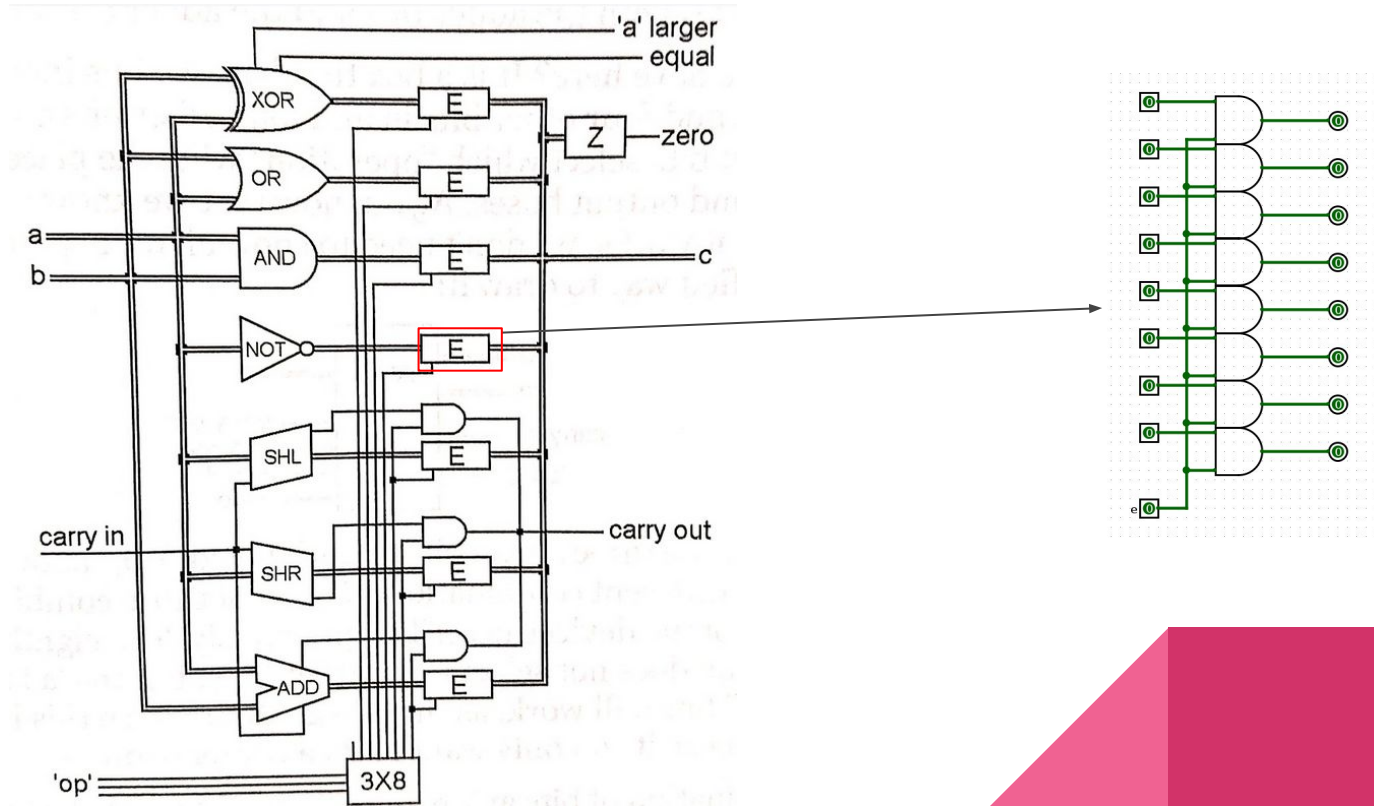


Unidade Lógica e Aritmética

ULA

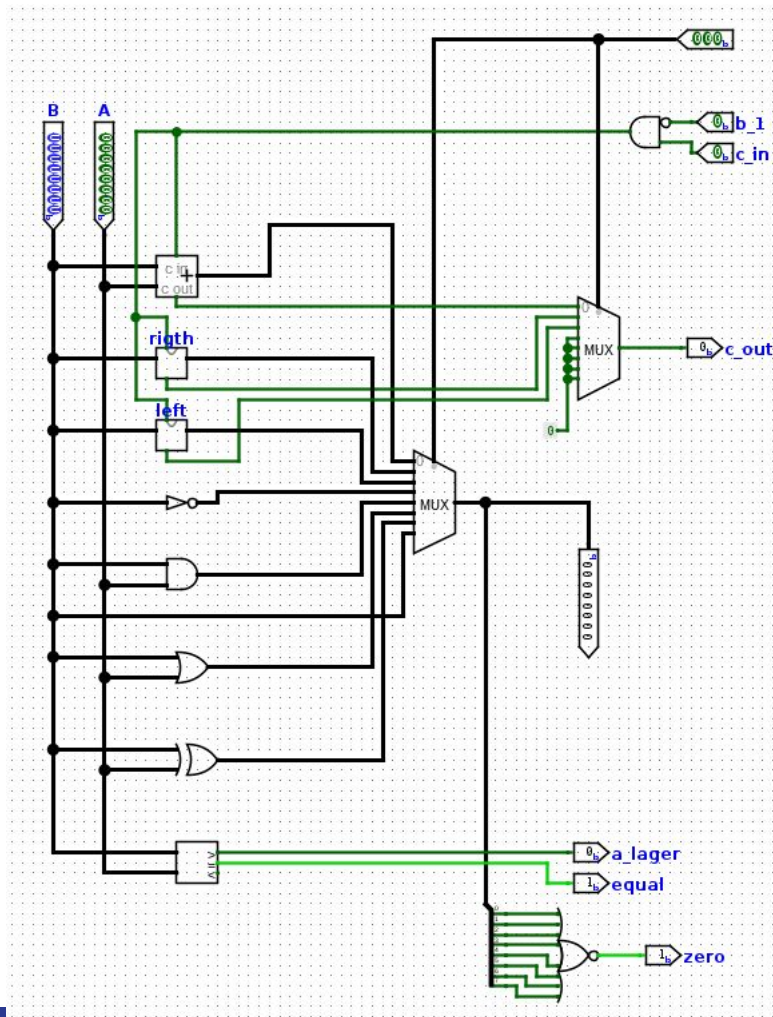


ULA



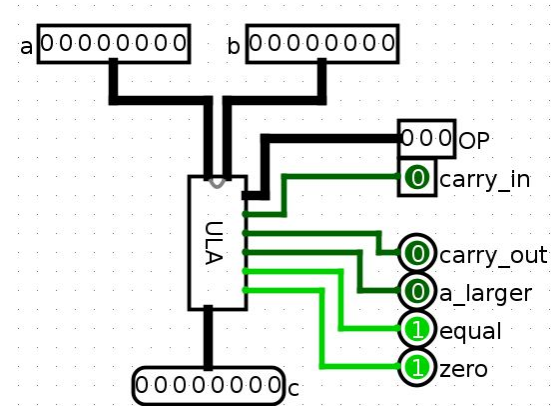
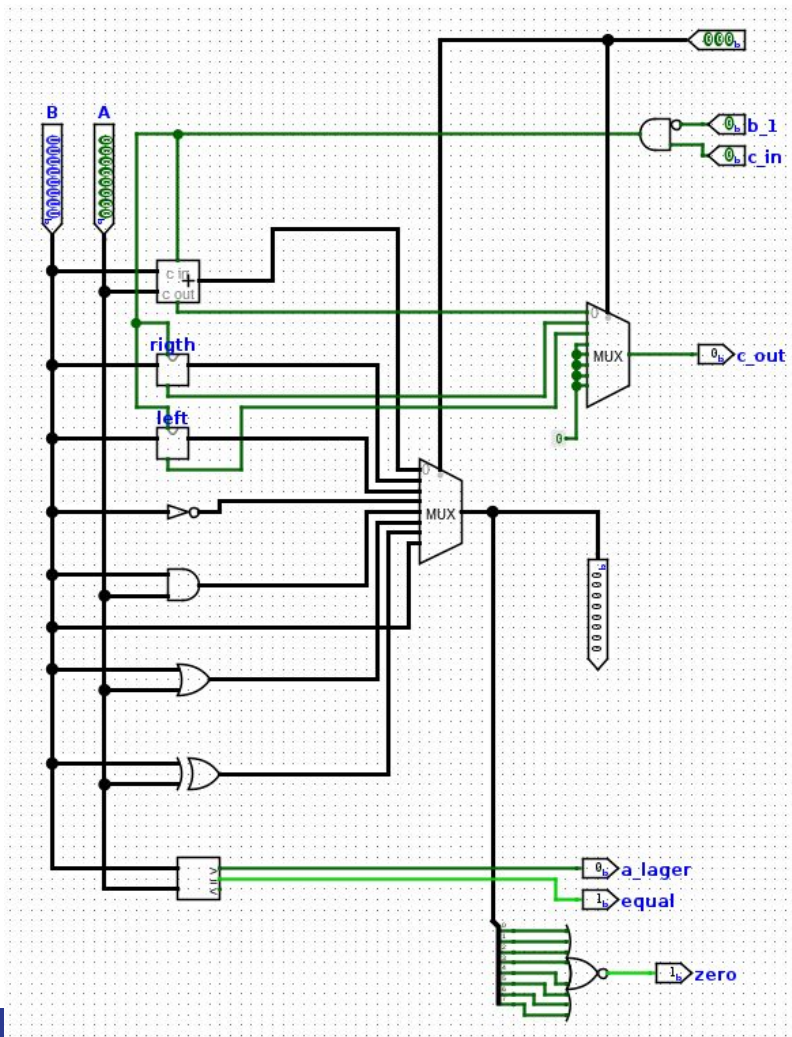
ULA

| code | OP |
|------|-----|
| 000 | ADD |
| 001 | SHR |
| 010 | SHL |
| 011 | NOT |
| 100 | AND |
| 101 | OR |
| 110 | XOR |
| 111 | CMP |



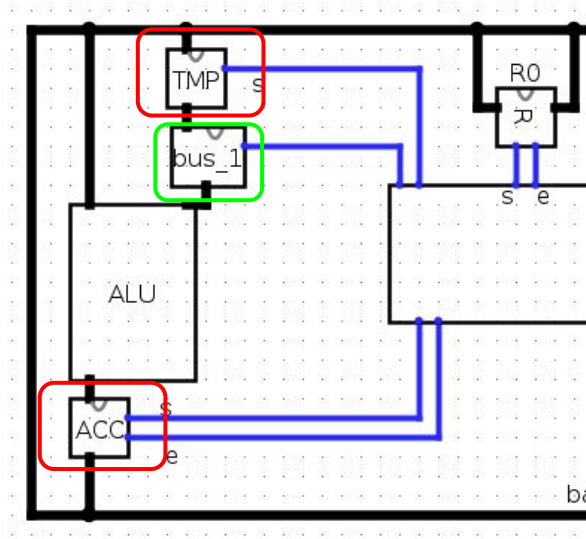
ULA

| code | OP |
|------|-----|
| 000 | ADD |
| 001 | SHR |
| 010 | SHL |
| 011 | NOT |
| 100 | AND |
| 101 | OR |
| 110 | XOR |
| 111 | CMP |



Conexões da ULA

Elementos adicionais



Dois registradores, um na entrada e outro na saída.



Um seletor que gera a constante 1.

Atividades

- Enviar o circuito da ULA simplificado e testado

