

|                                    |                                     |                      |                        |
|------------------------------------|-------------------------------------|----------------------|------------------------|
| Nome:                              | <b>Pedro Gabriel G. R. Balestra</b> | Curso: <b>GEC</b>    | Matrícula: <b>1551</b> |
| Nome:                              | <b>Wesley Marcos Borges</b>         | Curso: <b>GEC</b>    | Matrícula: <b>1651</b> |
| <b>INATEL – 2º SEMESTRE - 2022</b> |                                     | Matéria: <b>C208</b> | Turma: <b>L5</b>       |

## Projeto ALU

Nesse projeto, construímos uma ALU (Arithmetic Logic Unit), que através da sequência de OP CODES definida, consegue fazer algumas operações lógicas e aritméticas, como esperado.

Esses OP CODES (códigos de operação) são sequências de números que são usados para escolher qual operação será realizada, visto que a ALU, nesse projeto, faz 8 operações.

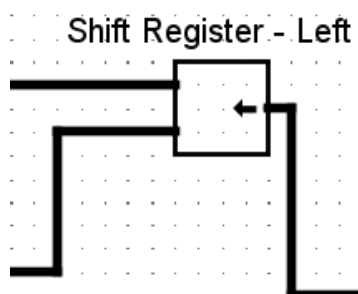
A seguir, será apresentada a definição de cada operação e seu respectivo circuito, quando houver.

### 1) Deslocamento à esquerda

O shift register (registrador de deslocamento) pega um dado de entrada e tem a função de fazer o deslocamento bit à bit desse dado. O deslocamento à esquerda é feito, matematicamente da seguinte forma:

$D * 2^N$ , onde D é o dado de entrada e N o número de bit que será deslocado.

Usamos um registrador de deslocamento do próprio Logisim, como o nome **“Shift Register – Left”**. Seu OP CODE é 0.

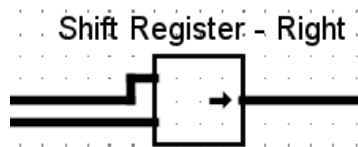


### 2) Deslocamento à direita

Já o deslocamento à direita é feito, matematicamente da seguinte forma:

$\frac{D}{2^N}$ , onde D é o dado de entrada e N o número de bit que será deslocado.

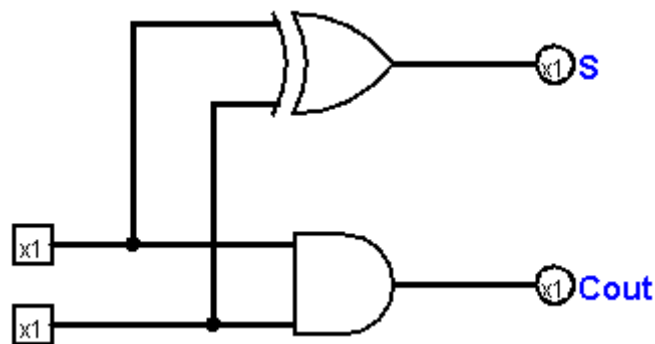
Usamos um registrador de deslocamento do próprio Logisim, como o nome **“Shift Register – Right”**. Seu OP CODE é 5.



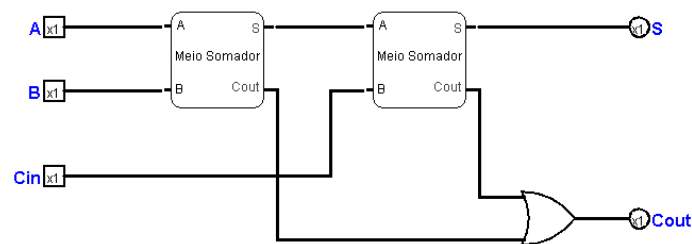
### 3) Adição e Subtração

A subtração foi feita usando um somador completo de 8 bits. O circuito se encontra no arquivo “**Somador\_8**”. Ele foi feito usando 8 somadores completos (que são feitos usando meio somadores), no qual temos uma entrada chamada **Cin**, que seleciona se vamos **somar** (**Cin = 0**) ou **subtrair** (**Cin = 1**). No circuito final do projeto, tem o nome “**Somador/Subtrator Completo**”. Os OPCODEs são: Adição = 7 e Subtração = 10. Seu circuito ficou assim:

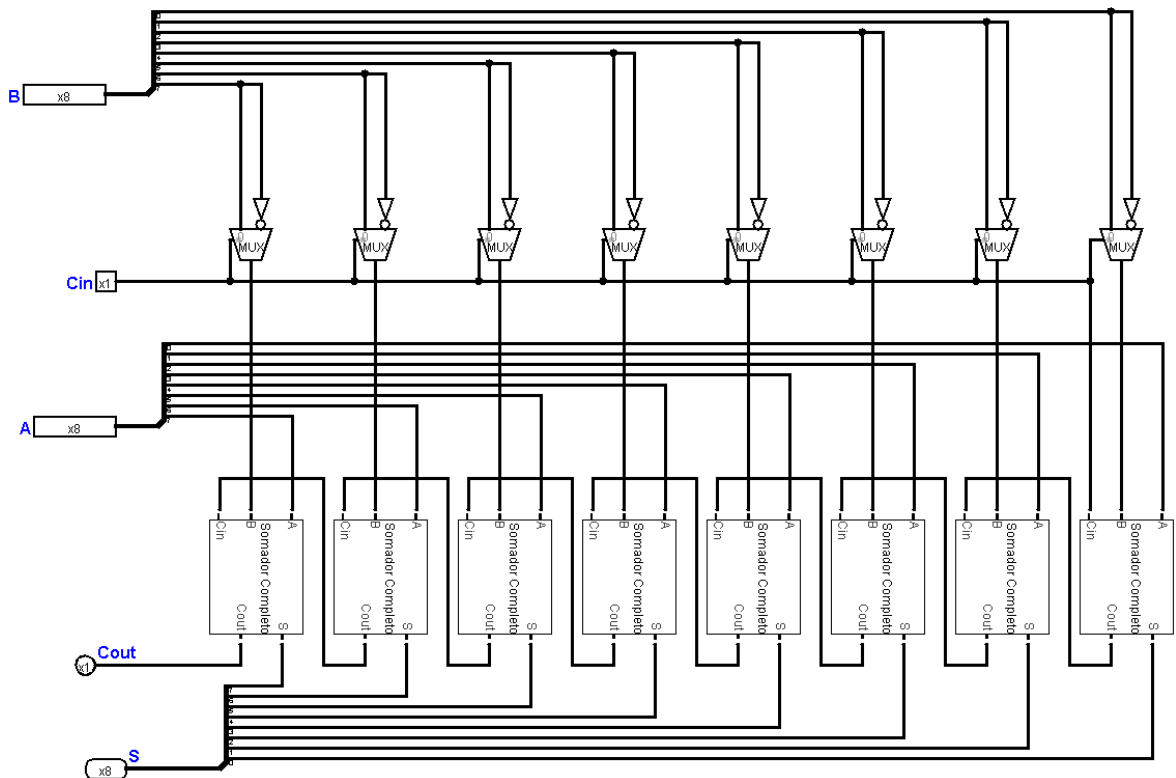
#### Meio Somador:



#### Somador Completo:

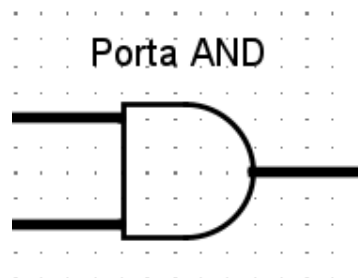


## Somador com 8 bits:



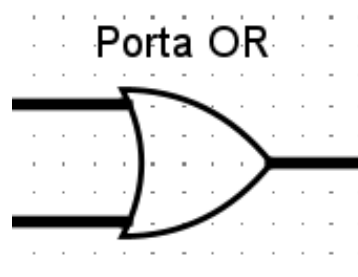
## 4) AND

Usamos uma porta lógica AND do próprio Logisim, como o nome **“Porta AND”**. Seu OPCODE é 9.



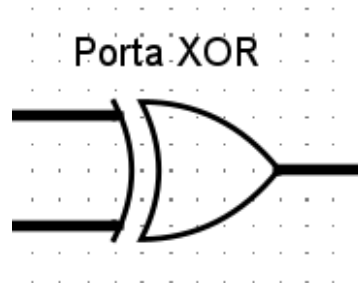
## 5) OR

Usamos uma porta lógica OR do próprio Logisim, como o nome **“Porta OR”**. Seu OPCODE é 15.



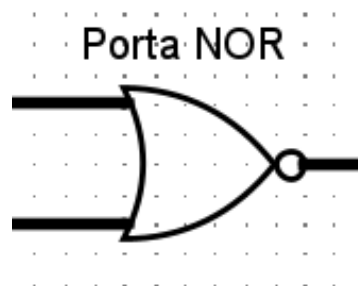
## 6) XOR

Usamos uma porta lógica XOR do próprio Logisim, como o nome ***“Porta XOR”***. Seu OPCODE é 4.



## 7) NOR

Usamos uma porta lógica NOR do próprio Logisim, como o nome ***“Porta NOR”***. Seu OPCODE é 2.



## 8) Circuito Completo da ALU

Após juntar todas essas operações em um MUX, temos o circuito completo da nossa ALU. As entradas do MUX que não estão sendo usadas, as ligamos no terra (ground). Na saída, conectamos um ponto de saída no sistema **Hexadecimal (16)** com o nome ***“S\_16”*** e um ponto no sistema **Decimal (10)** com o nome ***“S\_10”***. Já para seleção do OPCODE, colocamos um **Display Hexadecimal** para facilitar a seleção das operações.

Dessa forma, concluímos a criação da nossa ALU.

