**UNIVERSIDADE ESTADUAL DE MARINGÁ**

**INFORMÁTICA**

OLGA MARIA DOS SANTOS R.A. 130002

LEONARDO OTA KUDO R.A. 130266

**PROJETO ULA 4 BITS**

MARINGÁ

2022

Olga Maria dos Santos Gonçalves R.A. 130002

Leonardo Ota Kudo R.A. 130266

**PROJETO ULA 4 BITS**

Curso: Informática,

Disciplina Circuitos Digitais (9889-2022-T32)

Professor: Maurilio Martins Campano Junior

MARINGÁ

2022

**SUMÁRIO**

**Introdução**...............................................................................................................4

O que é uma ULA?........................................................................................4

Elementos utilizados......................................................................................5

**Desenvolvimento**....................................................................................................9

**Justificativa**..................................................................................................9

**Decisões de projeto para a simulação**.....................................................10

**Funcionamento da ULA**.............................................................................20

**Conclusão**..............................................................................................................25

**Referências**............................................................................................................26

**Introdução**

Este projeto consiste em desenvolver e simular o projeto de um circuito combinacional de uma ULA (Unidade Lógico e Aritmética) de 4bits.

Primeiros vamos definir:

**O que é uma ULA**?

A unidade lógica e aritmética (ULA), executa as principais operações lógicas e aritméticas do computador, podendo ser somas, multiplicações..., comparações, decisões lógicas. Normalmente, uma ULA recebe os operandos de entrada, e uma entrada de controle, que neste projeto chamaremos de seletores, que permite especificar qual operação deverá ser realizada. Por esse motivo, a construção de uma ULA se baseia em dois fundamentos principais: o fluxo de dados e a construção de circuitos que implementam operações.

Neste projeto, a ULA terá foco em operações aritméticas de soma, subtração, incremento e decremento, e nossos operandos serão restringidos aos binários, os cálculos serão todos baseados nos comportamentos binários de 4 bits, sendo que para ser 4 bits devemos ter 8 entradas para os operandos e estes serão convertidos e mostrados aos usuários por método de displays de 7 segmentos para que haja melhor entendimento dos valores informados e dos resultados.

Para deixar mais claro o entendimento de todo o projeto, será esclarecido o nome, comportamento dos componentes que iremos utilizar.

Utilizaremos o programa **logisim**, que é um simulador lógico que permite o desenho e a simulação de circuitos através de uma interface gráfica,

**Elementos utilizados:**

**Pinos de entrada**: recebem os valores 0 ou 1, que vão compor as entradas em binário.



**Constante**: são pinos de entrada, sempre ficaram com o valor 0 ou 1, que já aparecerá nele:



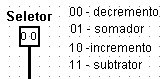
**Display de 7 segmentos**: Display que mostrará os valores de binários, em decimal (0 até 9), será utilizado tanto em entradas, quanto na saída de dados



**Multiplexador:** abreviado por MUX, responsável por receber a saídas das operações de soma, subtração, incremento e decremento, e pelo seletor poderemos escolher qual operação terá o resultado mostrado na tela.



**Seletor**: é um pino de entrada com dois bits, será responsável por proporcionar a escolha de qual saída será apresentada, será conectado ao MUX.



**Pino de saída**: este vai mostrar o valor de 5 bits em binário do resultado da operação que escolhermos anteriormente no seletor conectado ao MUX.



**Distribuidores**: Eles recebem os valores e distribuem na quantidade de bits que precisamos, que nos é de 4 ou 5 bits, servirão para deixar os circuitos mais limpo e uma distribuição mais eficiente:



**Chips**: São “caixinhas” que compõe toda a lógica do que foi programado dentro delas, são maneiras resumidas, iremos utilizar estas em soma, subtração, incremento, decremento e nos displays, assim o circuito da nossa ULA vai ficar visualmente mais eficiente e de melhor entendimento, pois nomearemos os chips com qual sua função, e seu visual pode mudar dependendo do numero de entradas e saídas:

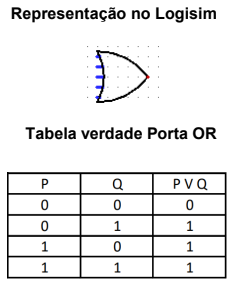


Estes elementos não estão visualmente no circuito da ULA de 4 bits, mas são utilizados dentro da lógica dos somadores, substrator, displays, incremento e decremento.

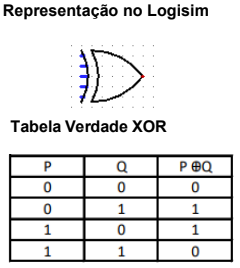
**Porta AND**: A porta lógica AND, “e” na língua portuguesa, faz referência a uma operação lógica que aceita dois ou mais operandos, que sempre resultem em um valor lógico verdadeiro, se somente se todos os valores passados terem seu valor sendo verdadeiro, caso o contrário se existir um valor ou mais que sejam falsos a saída resultante é falsa.



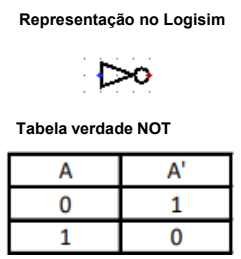
**Porta Lógica OR**: na tradução para o português fica “ou”, faz referência a uma operação lógica que permite a entrada de um ou mais valores e que sempre retorna um valor verdadeiro se um desses valores forem verdadeiros.



**Porta Lógica XOR**: Também conhecido como “ou exclusivo”, esse componente faz referência a uma operação lógica entre apenas dois operadores, que resulta em um valor verdadeiro se e somente se os dois valores passados forem distintos entre si. Ou seja, se um for verdadeiro e o outro for falso, a saída resultará em um valor verdadeiro, porém caso ambos sejam verdadeiros ou forem falsos o valor resultante será falso.



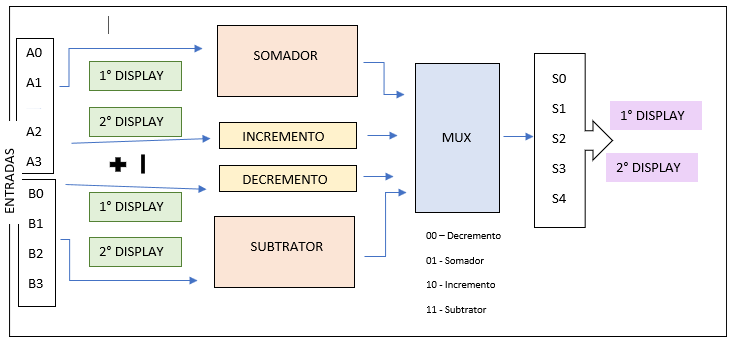
**Porta NOT**: A porta not, em português “não”, tem a função de trocar o valor passado por ela, por exemplo, se um valor verdadeiro é passado por uma porta not a saída será falsa, se a entrada é falsa, a saída será verdadeira. A ideia em si é sempre trocar os valores.



**Desenvolvimento:**

**Justificativa**

O planejamento do funcionamento da ULA foi baseado no diagrama abaixo:



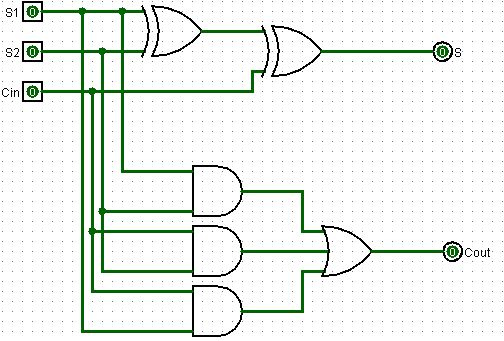
Sendo as 8 entradas necessárias para os processos, e para ser considerado 4 bits, estas informado pelo usuário em forma de binários serão convertidas pela lógica do display, a um valor decimal que pode ir de 0 até 15, tanto nas entradas “A”, quanto na “B”, logo teremos dois displays para A, sendo um que vai somente de 0 a 1 e o segundo que vai de 0 a 9, compondo assim quaisquer valores de 0 até 15, e vai ser a mesma regra para as entradas B.

Após isso o circuito vai processar automaticamente todas as operações, soma, subtração, incremento e decremento, sendo que o incremento, executará sobre as entradas A, o decremento sobre as entradas B, mas ao chegar os resultados no MUX, o usuário deve escolher qual operação irá ser mostrado o resultado, escolhendo entre as opções listadas no diagrama (00|01|10|11) e após a escolha, a saída do MUX mostrará o resultado em binário de 5 bits, e este também será mostrado em um display de 7 segmentos, que terá dois displays um de 0 até 3, e outro de 0 até 9, compondo todas as combinações de 0 até 31.

**Decisões de projeto para a simulação**

Vamos apresentar por partes a lógica de soma, subtração, incremento, decremento e dos displays, após explicar a lógica por tabelas e expressões, mostraremos o resultado final da ULA.

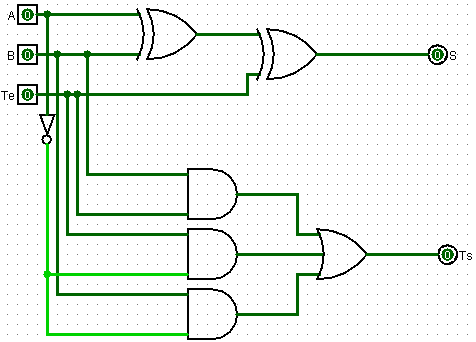
**Somador:**

Chegamos neste circuito, seguindo a tabela verdade do somador, retirando expressão pelo mapa de karnaugh.



**Subtrator:**

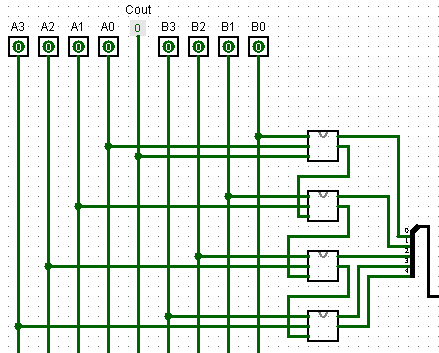
 

Chegamos neste circuito, seguindo a tabela verdade do somador, retirando expressão pelo mapa de karnaugh.



Para chegarmos no somador e subtrator que estará visivel no circuito da ULA de 4 bits, foi necessário utilizar o meio dos chips, para que a saida Ts ou Cout fosse uma entrada na próxima operação e o ultimo teria duas saídas, que daria os possiveis 5 bits de saídas.

Resultado do somador e subtrator na ULA:

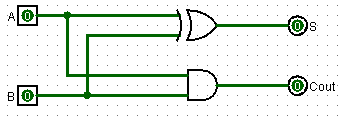


O primeiro chip, receberá as entradas A0 e B0, e o Cout sempre será 0, será feito a lógica da tabela verdade do somador e subtrator, e terá a saída e outra cout, que será entrada para o próximo chip, e o ultimo chip terá a saída S e o cout como saídas, e o resultado disso estará conectado a uma distribuidor que está conectado ao MUX.

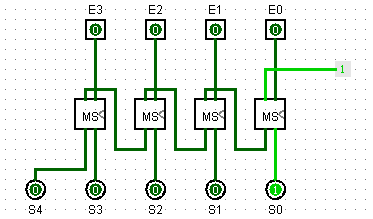
**Incremento:**

Para ser feito o incremento é utilizado a lógica do meio somador:





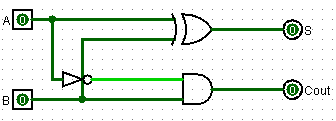
Utilizando as caixinhas ligando a saída cout nas próximas entradas, a estrutura final do incremento ficou:



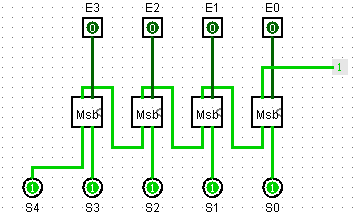
**Decremento:**

Para ser feito o decremento é utilizado a lógica do meio subtrator

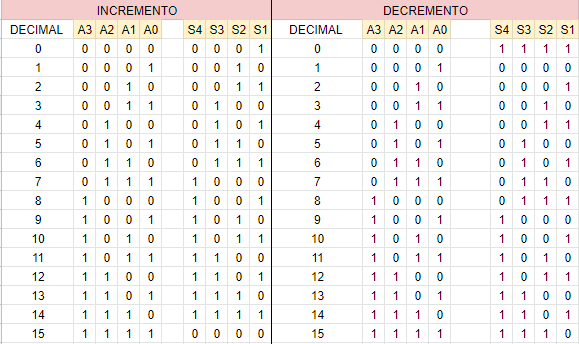




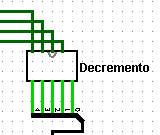
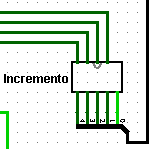
Utilizando as caixinhas ligando a saída cout nas próximas entradas, a estrutura final do decremento ficou:



A lógica utilizada para montar incremento e o decremento, com os meios subtrator e meios somadores, veio advinda das tabelas verdades e da lógicas dos mesmos, e eles vão receber 4 entradas o incremento aumentará esta entrada em um o decremento diminuirá em 1, seguindo a tabela de 0000 até 1111, se for incrementado 0000, ele se tornará 0001, se 1111 for decrementado resultará em 1110, como mostra as possibilidades:



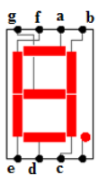
Para utilizar o incremento e o decremento no circuito principal da ULA, foi também utilizando por meio dos chips

* O decremento receber os valores de B0, B1, B2 e B3, retorna os valores decrementados em um distribuidor ligado ao MUX;
* O Incremento receber os valores de A0, A1, A2 e A3, retorna os valores incrementados em um distribuidor ligado ao MUX;

**Displays:**

O display utilizado como base para todas as tabelas dos display foi:

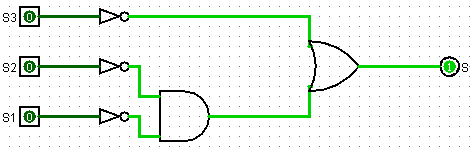


**Display das entradas A e B**

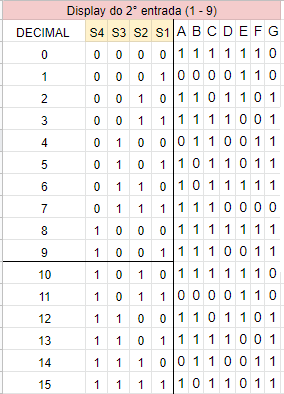
Para utilizarmos os displays nas entradas foi necessário fazer a seguindo tabela, nela os valores 1 de A até F significa quais segmentos do display irá acender, ou no caso do logisim, ficar em vermelho, os que acendem sincronizadamente, mostravam as formas dos números decimais respectivos as entradas binárias que o usuário informou tanto em A quanto B, as possibilidades dos números com as combinações binárias estão na tabela, ou seja nas linhas 0 das tabelas das entradas ou saídas, mostra qual é o decimal e seu respectivo valor de entrada nos campos de A e B, e logo ao lado quais segmentos serão acendidos.

Para o primeiro display de entrada, que mostraria valor 0 ou 1, notamos que seria somente nos binários 1010 até 1111, então a lógica do circuito utilizado foi:

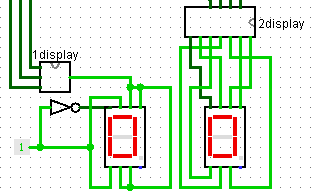
S = S3 + ( S2 . S1 )



Para o segundo display de entrada, foi criado a tabela:



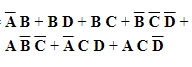
Foi retirado as expressões por karnaugh da tabela, e com as expressões de A até F, foram criados 7 circuitos e em um circuito nomeado “2display” na pasta displayEntrada, foi inserido todos os 7 circuitos em forma de chips, e o circuito “2display” e “1display” foram utilizados na ULA, para as entradas, ficando:



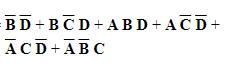
**Expressões retiradas da tabela:**

A= 

B = CD + AC + ACD + AD + BD



C =

D = 

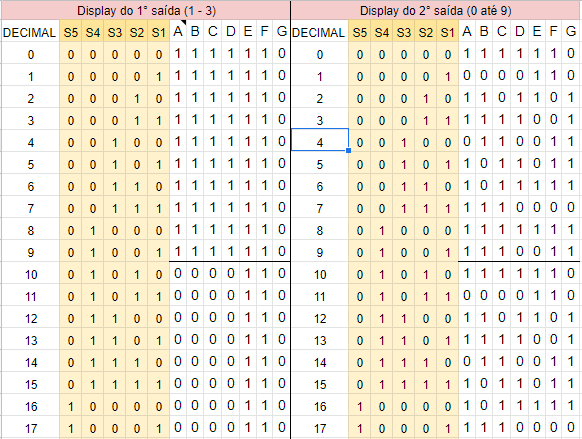
E = 

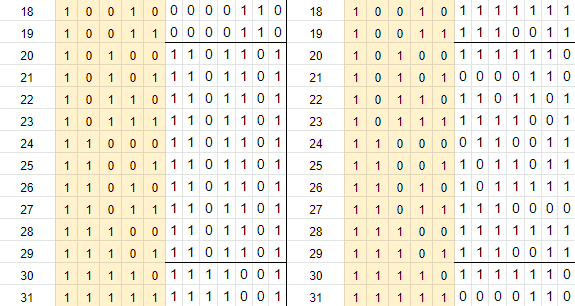
F =  => (A  C)

G = 

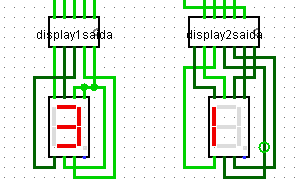
**Display dos resultados:**

Os resultados das 4 operação que nossa ULA fará, foram encaminhadas por meio do distribuidor ao MUX, o usuário irá selecionar qual o resultado quer que seja mostrado, e este será ligado a 2 distribuidores, que os valores binários do resultado serão entradas nos displays de 7 segmentos, estes serão dois, um que pode mostrará os valores de 0 a 3, e outro de 0 a 9, foram construidos por meio da tabela:

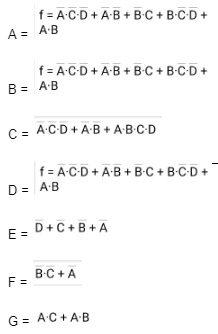




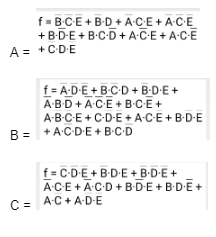
Definimos as tabelas, foi retirado as expressões por mapa de karnaugh e estas expressões foram feitos todos os circuitos de A até F das duas tabelas, no primeiro display e do segundo, estes formaram dois circuitos, o primeiro os 7 chips, cada um representando uma letra de A-F, e na segunda tabela a mesma ideia. Depois de montarmos somente dois circuitos com todas as letras dos segmentos, este denominados “display1saida” e “display2saida”, se tornaram chips dentro do circuitos da ULA, e assim representa a seguinte saída, pode ser de 00 até 31 em decimal.

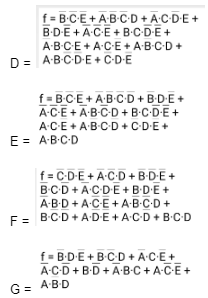


**Expressões do 1° display retiradas da tabela:**



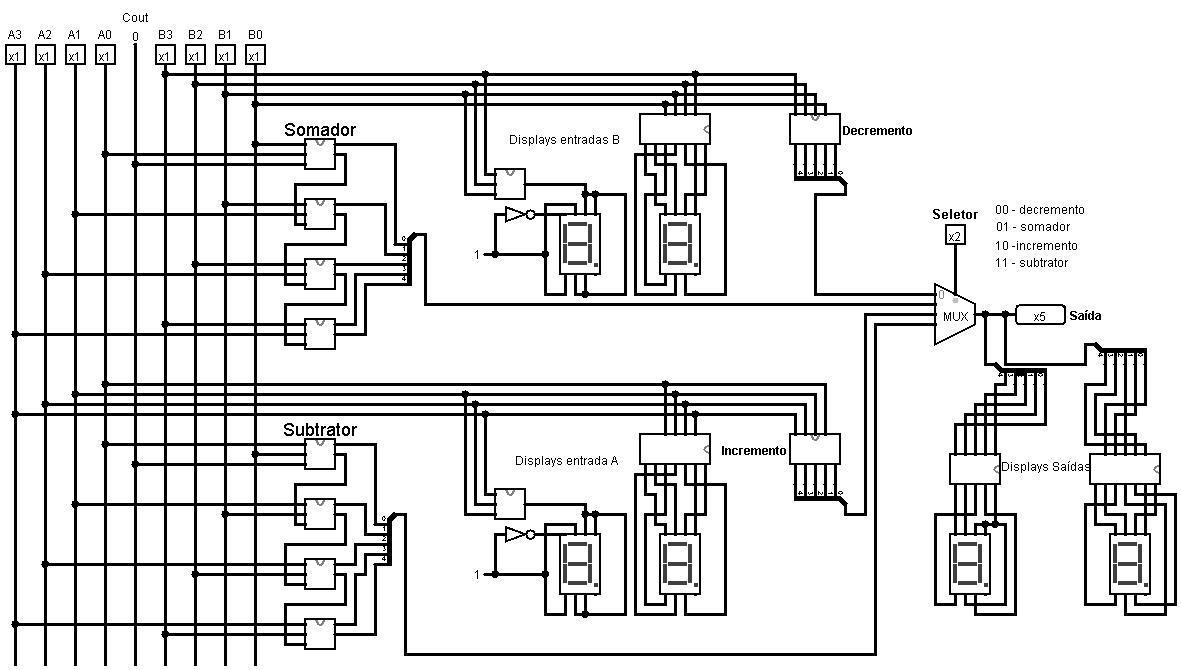
**Expressões do 2° display retiradas da tabela:**



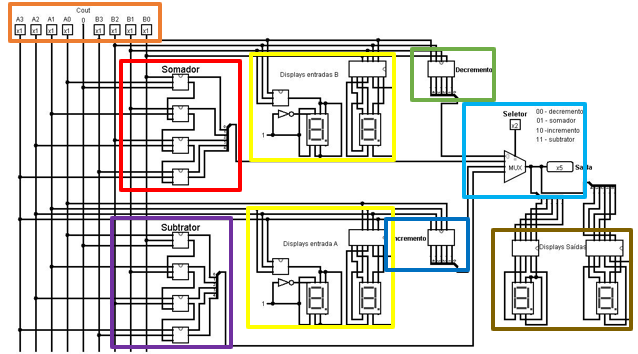


**Funcionamento da ULA**

Projeto final:



Circuito com as partes destacadas:



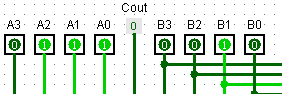
Descrição das partes:



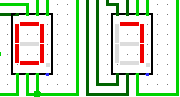
Para a simulação da ULA é necessário, informar um valor binário nas quatro entradas A e nas entradas B, que será mostrado em decimal nos displays e depois escolher no seletor, qual operação deseja ver o resultado (soma, subtração, incremento ou decremento), e será mostrado em binário e em decimal o resultado.

Demonstração:

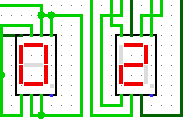
* Nas entradas A informaremos o valor 0111 e na B será o 0010, respectivamente estes valores ficam 07 e 02 em decimal



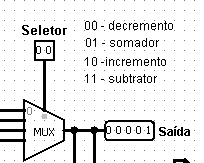
* Os valores decimais, no display da entrada A, na opção 0111



* Os valores decimais, no display da entrada B, na opção 0010

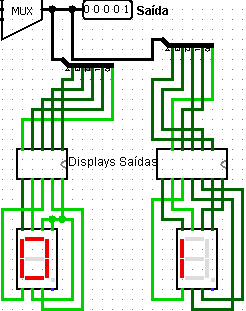


* Após informar os valores de entrada, todas as opções irão processar, e no seletor do MUX, precisará escolher qual delas quer a saída



* Se seletor for 00:

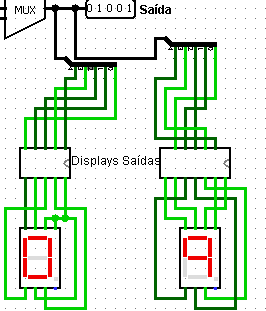
A opção escolhida fica o DECREMENTO, e ele está relacionada a entrada B, se a entrada B é 0010, o decremento é 0001, seguindo a tabela verdade do decremento, que está na seção acima, está correto, e o resultado no display ficou o decimal 01, que é antecessor de 02.



* Se seletor for 01:

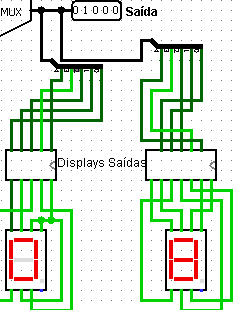
A opção escolhida é a soma dos valores de A e B, sendo então:

, 1001 é exatamente o que mostra na saída binária, e 01001 na tabela dos displays de saída representa o 09, se somar também 07+02 = 09 (valor de A em decimal + valor de B em decimal = resultado em decimal)



* Se seletor for 10:

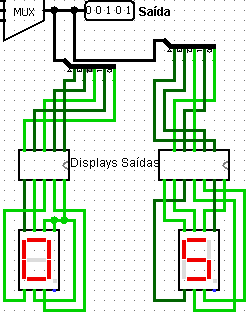
A opção é o incremento, ou seja, aumentar o valor de A, logo se A é 0111 incrementando seu valor, ficaria 1000, o mesmo que está descrito na tabela de incremento, e A em decimal está como 07, um a mais seria 08:



* Se seletor for 11:

A opção escolhida é a subtração dos valores de A e B, sendo então:

0101 é o resultado, e é exatamente o que mostra na saída binária, e 00101 na tabela dos displays de saída representa o 05, se subtrair 07-02 = 05 (valor de A em decimal - valor de B em decimal = resultado em decimal)



**Conclusão**

O objetivo deste projeto, ULA de 4 bits, foi compreender e desenvolver o funcionamento da ULA, somente uma parte neste projeto, a aritmética, composta pela soma e subtração e incremento e decremento, que de certa forma são somas e subtrações, sendo que um é soma um e outra menos 1 ao valor original.

Nosso primeiro passo foi criar todos os componentes isolados e testar seu funcionamento, em primeiro foi o somador e o subtrator, e estes foram criados pelos conceitos estudados em sala de aula, após foi realizado o incremento e decremento, que o conceito da aula mais as pesquisas deram o resultado, e por fim os displays, que por mais que fossem de fácil entendimento, sua criação foi longa, definimos com base em qual display criaríamos a tabela, depois de criar os de entradas, retirar expressões e fazer circuitos tivemos que testar sua funcionalidade e o mesmo para os displays de saída.

Por fim a união de todos os componentes, de um modo esteticamente compreensível, e neste momento que unimos os resultados usando distribuidores ao MUX, e fizemos os testes finais.

O projeto por si só foi uma demonstração de usabilidade dos conceitos de circuitos digitais, como eles se integram, e como podemos reutilizar os componentes, em como conceitos iniciais como portas lógica e tabelas verdades se comportam dentro de grandes funcionalidades.

**Referências:**

Moraes Tannus, Alexandre, Unidade Lógica Aritmética, Anápolis 2019, disponível em:

<http://repositorio.aee.edu.br/bitstream/aee/1853/1/Unidade%20L%C3%B3gica%20Aritm%C3%A9tica.pdf>

Cantu, Evandro, simulador de circuitos lógicos, sistemas digitais 2016, disponível em:wiki.foz.ifpr.edu.br/wiki/index.php/Simulador\_de\_Circuitos\_L%C3%B3gicos\_-\_Logisim

Galdino, Jean, Organização de computadores, 2016, disponível: <https://docente.ifrn.edu.br/jeangaldino/disciplinas/2016.1/organizacao-de-computadores/aula-14-logisim-interligando-as-partes>

The academician, Incrementer (using Half Adder) and Decrementer circuit, disponível em: <https://www.youtube.com/watch?v=aJPUJl47DPg>

Increment e Decrement, disponível em:

<https://www.youtube.com/watch?v=-1WQ8B0_i-Y>

Multiplexador, Fabiana Paixão disponível em:

https://www.youtube.com/watch?v=YQg9sRNw3aU, acesso em 20, out

2022