

UNIVERSIDADE ESTADUAL DE MARINGÁ
INFORMÁTICA

OLGA MARIA DOS SANTOS R.A. 130002
LEONARDO OTA KUDO R.A. 130266

PROJETO ULA 1 BIT

MARINGÁ
2022

Olga Maria dos Santos Gonçalves R.A. 130002

Leonardo Ota Kudo R.A. 130266

PROJETO ULA 1 BIT

Curso: Informática,
Disciplina Circuitos Digitais (9889-2022-T32)
Professor: Maurilio Martins Campano Junior

MARINGÁ
2022

Sumário

Introdução	04
• Componentes utilizados	04
Desenvolvimento	07
• Objetivos/Justificativa	07
▪ Definindo o que é uma ULA	07
▪ Diagramas da ULA	08
• Funcionamento da ULA	09
• Decisões de projeto para a simulação	11
▪ Circuito final do projeto	11
▪ Circuito com partes destacadas	11
▪ Circuito do somador	12
▪ Circuito do subtrator	12
▪ Verificação da funcionalidade da ULA	13
▪ Resultados Obtidos	14
Conclusão	17
Bibliografia	18

Introdução

O objetivo do projeto é compreender o desenvolvimento e simular o funcionamento de uma ULA de 1 bit (Unidade Lógica e Aritmética).

Componentes Utilizados

Para o desenvolvimento do projeto, utilizamos o programa Logisim, que é um simulador lógico que permite o desenho e a simulação de circuitos através de uma interface gráfica, nele utilizamos conceitos como:

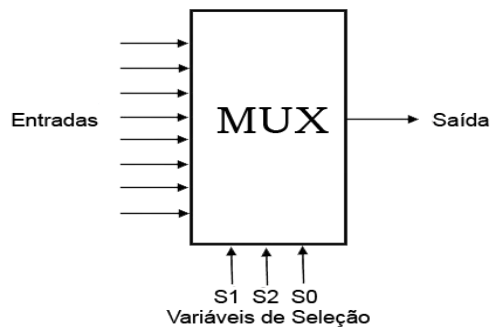
- Multiplexador 8 entradas para 1 saída;
- Porta Lógica AND;
- Porta Lógica OR;
- Porta Lógica XOR;
- Porta NOT;
- Pinos de entradas e saídas;

Descrição dos Componentes Utilizados:

- Tomamos:
 - 1 para verdadeiro;
 - 0 para falso;

Multiplexador: Um multiplexador é um componente que seleciona as informações de duas ou mais fontes de dados para conduzi-las por um único canal de saída. Um MUX pode ter até 2 elevado nas entradas, necessárias para a seleção do sinal que irá à saída, nas entradas envolvidas na seleção. Portanto, para dois sinais, é necessário um sinal de seleção, para quatro sinais, são necessários dois sinais para seleção, e assim por diante, a saída varia de acordo com a seleção escolhida.

Multiplexador 8x1



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.

Representação no Logisim

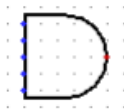


Tabela verdade da Porta AND

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

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.

Representação no Logisim



Tabela verdade Porta OR

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

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.

Representação no Logisim

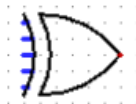


Tabela Verdade XOR

P	Q	$P \oplus Q$
0	0	0
0	1	1
1	0	1
1	1	0

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.

Representação no Logisim

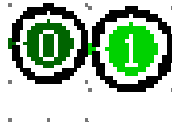


Tabela verdade NOT

A	A'
0	1
1	0

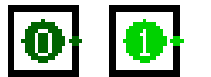
Pino de Saída: Este componente é possível ver o resultados obtidos pelas operações, sendo seus únicos possíveis valores, 0 e 1.

Representação no Logisim



Pino de Entrada: os pinos de entrada são espécies de seletores que podem estar com os valores 0 e 1, permitindo a escolha entre as duas condições, são importantes pois conduzem todo o experimento.

Representação no Logisim



Desenvolvimento

Objetivos e Justificativa

Definindo o que é uma ULA

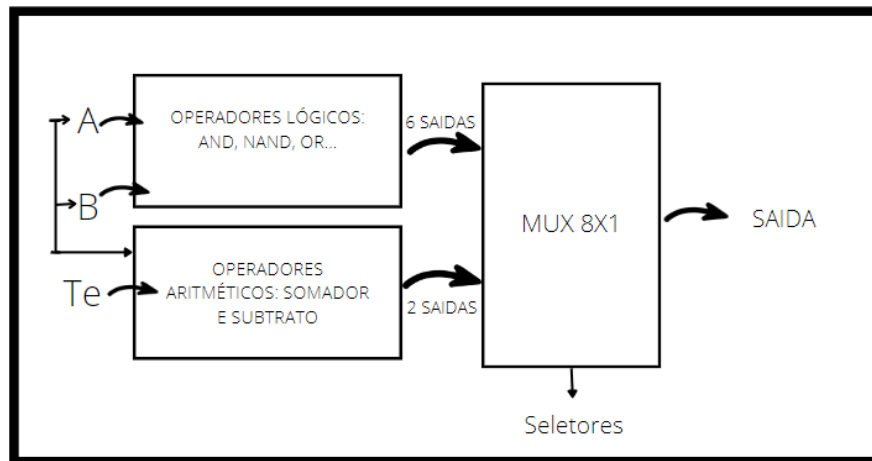
Uma Unidade Lógica e Aritmética (ULA) é um dispositivo que realiza operações lógicas e aritméticas sobre números representados em circuitos lógicos. Normalmente, uma ULA recebe dois operandos como entrada, e uma entrada de controle, que neste projeto chamaremos de seletores, permite especificar qual 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.

O processo de desenvolvimento da ULA de um bit ocorreu em estágios e de uma forma progressiva foi elaborado, construído, testado e finalizado o projeto. Dessa forma, desde a busca por conteúdo, tanto os ensinamentos na aula quanto o que foi encontrado em outras fontes, foi utilizado para definir os conceitos e funcionalidades que a ULA possui e a construção de um projeto que simule e faça as operações advindas da mesma, que no caso requerido pelo trabalho consiste de uma ULA de um bit.

A ULA, unidade lógico e aritmética, tem como função realizar operações lógicas, como and; or; nand; nor; xor; xnor, e, também, realizar operações aritméticas, como soma; subtração de números binários. A ULA se encontra em componentes que envolvem processamento de dados, como o processador do computador, e podemos dizer como sendo um dos componentes fundamentais à computação de dados, e tem várias funcionalidades, seus valores saídas são diversos, principalmente quando o número de entradas aumenta.

Diagramas de uma ULA de um bit

Anterior a construção da ULA de um bit, em um simulador, foi necessário fazer a análise de diagrama, como este logo abaixo:



Representação simbólica de uma ULA de 1 bit

No diagrama podemos observar que há 2 entradas e uma saída principal, referente a saída do processamento da ULA, por isso “1” bit, e vão haver mais duas saídas em nosso circuito referente ao “Ts” (transporte de saída), muito conhecido como “carry out” da unidade aritmética.

As entradas A e B serão os valores de entradas, que passarão por processos que compõe as unidades da ULA, e ainda teremos a entrada “Te” (transporte de entrada), ou conhecido como “carry in”, teremos que utilizar tanto o “carry in” quanto “carry out” para conseguirmos concluir a lógica aritmética dos somador completo e subtrator completo e esta entrada e saída “Te” e “Ts” são totalmente, desconsideradas pelo núcleo lógico.

Vimos os tipos de entradas e saídas, lembrando que os valores possíveis são sempre 0 e 1, estes valores irão para os dois núcleos da ULA, o lógico e o aritmético, e dentro deles teremos as operações apresentadas anteriormente, mas para escolhermos qual delas vai ser mostrado o resultado, ou saída, utilizaremos um seletor, dentro do programa logisim, é um pino de entrada com 3 bit de dados sendo seus valores possíveis valores que vão determinar qual operação a ULA realizará.

A ULA desenvolvida para esse trabalho apresenta as seguintes operações:

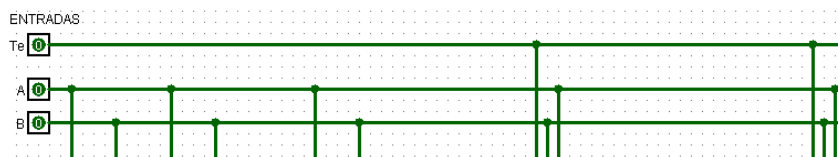
- Lógicas:
 - AND, NAND, OR, NOR, XOR e XNOR;
- Aritméticas:
 - Somador e subtrator

Funcionamento da ULA

O projeto da ULA de um bit foi todo realizado através do simulador **Logisim** este possui uma vasta quantidade de componentes que podem ser utilizados em circuitos lógicos. A seguir irá ser mostrado o passo a passo da construção do projeto:

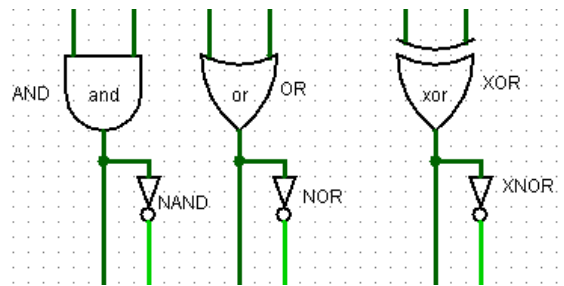
1. Posicionou-se as respectivas entradas, que são os componentes os **pinos de entrada**:

- A, B e “Te” com seus bits de entrada;



2. Através das tabelas verdades, mapa de Karnaugh do somador e subtrator completo, que serão apresentadas mais adiante na explicação, foi possível montar os circuitos que formam todas as operações lógicas e aritméticas;
3. Foi posicionado os componentes para formarem a parte lógica da ULA:
 - AND;
 - OR;
 - XOR;

Como as portas NAND, NOR e XNOR são respectivamente os valores contrários das portas AND, OR e XOR negadas, foi conectados **portas not** nas saídas destas últimas. Dessa forma, não foi necessário a utilização de outras portas na parte lógica, com isso o sistema ficou mais organizado, e com o mesmo potencial de realizar todas as operações.

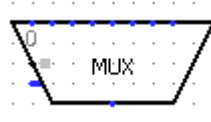


4. Foi realizado o circuito do somador e do subtrator em circuitos diferentes, e atribuídos eles ao circuito principal por métodos de “caixinhas”, que recebem a mesma quantidade de entradas, operam a lógica do circuito dentro dela e respondem com a saída, o objetivo de utilizar estas caixinhas foi de deixar o circuito mais “limpo”;



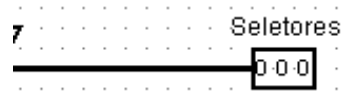
5. Para finalizar a parte lógica todas as portas foram conectadas com as respectivas entradas A e B, no caso do somador e subtrator teve a entrada “Te” envolvida, porém as saídas das portas lógicas, que forneceriam os resultados das operações, foram conectadas ao multiplexador, também abreviado por “MUX”;

Representação no logisim



6. Conectado ao MUX, um pino de entrada com 3 bit de seleção, sendo suas combinações que possibilitam a seleção de qual operação irá mostrar a saída.

- **Seletor**, do multiplexador 8x1:



- Sendo os possíveis valores lógicos do seletor

```

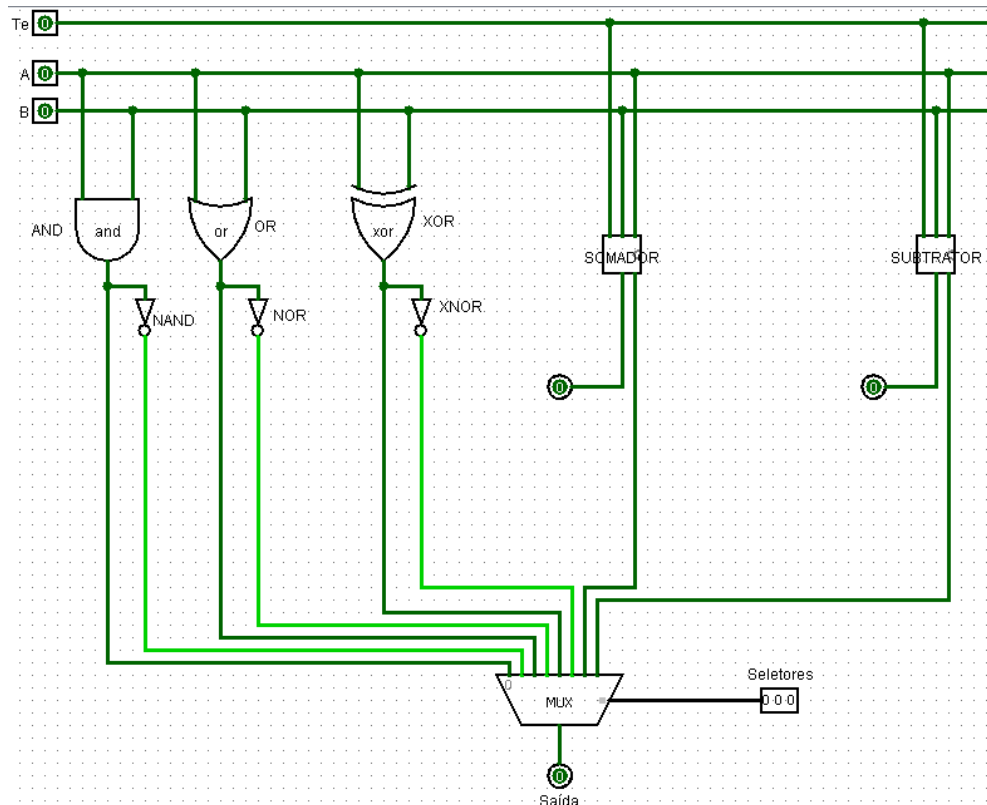
000 AND
001 NAND
010 OR
011 NOR
100 XOR
101 XNOR
110 Adição
111 Subtração

```

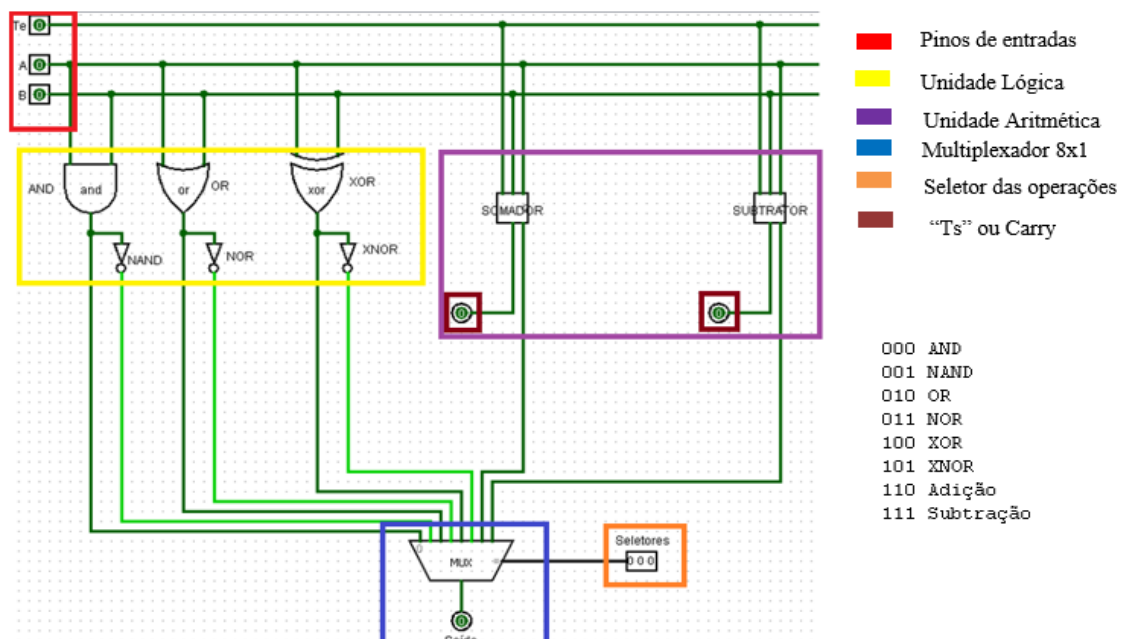
7. Na saída do multiplexador e na saída dos “Ts” foi colocado um pino de saída para monitorar os valores finais;
8. Dessa maneira foi possível testar e utilizar a ULA.

Decisões de projeto para a simulação

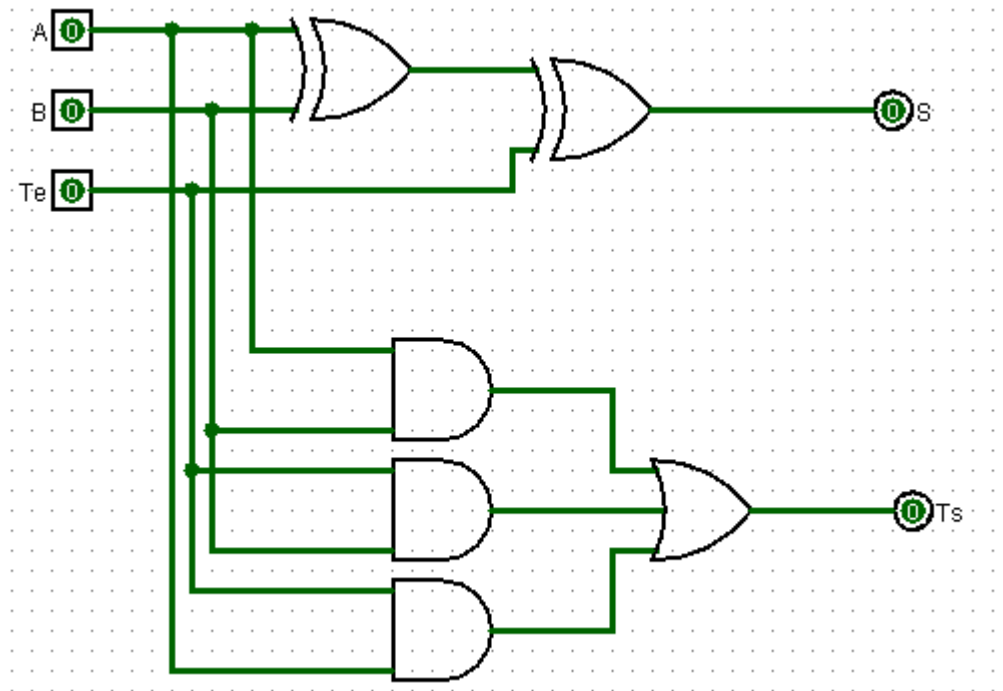
Circuito final do projeto



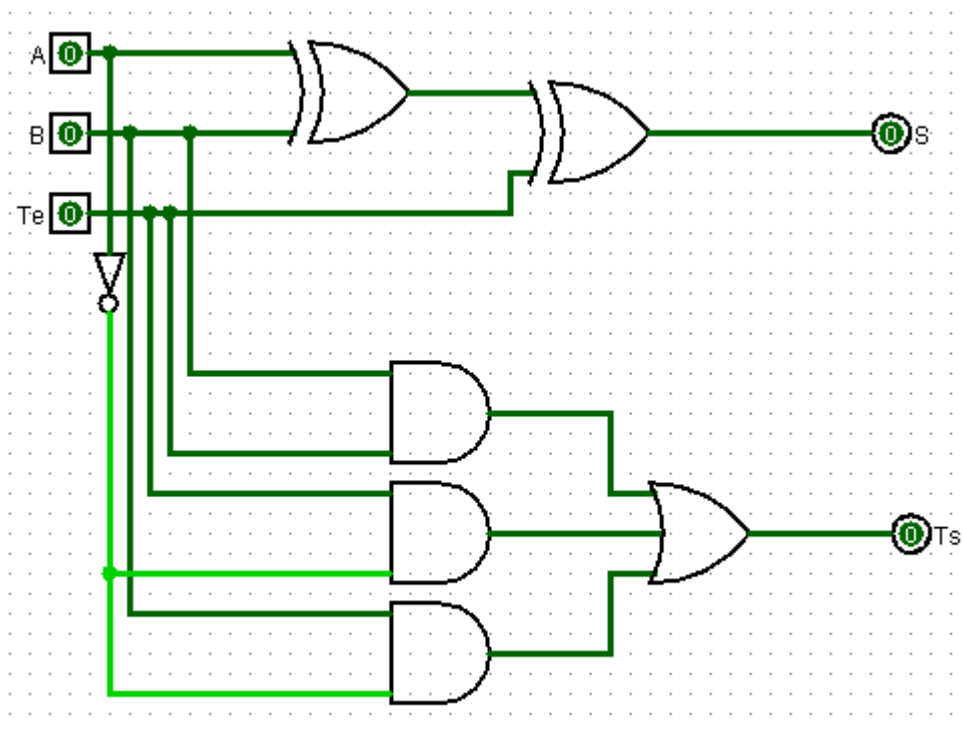
Circuito com as partes destacadas



Circuito do somador completo



Circuito do subtrator completo



Verificação do funcionamento de ULA de 1 bit

Como foi utilizado o Logisim para fazer a simulação e dessa maneira ter a verificação do funcionamento do projeto muitos dos passos tomados foram relacionados ao programa. Nessa fase de verificação foi feito os seguintes passos:

1. Foi feito os testes, no circuito denominado MAIN, alterando os valores de entrada, A, B e “Te”, e verificando com as tabelas verdades das operações se as mesmas selecionadas na ULA estavam em seu estado correto de funcionamento;
2. Foi visto que o “Ts” sempre retorna um valor mesmo que as operações não sejam da natureza aritmética, porém visto que todos as fontes consultadas não apresentavam nenhuma atenção à isso acabou por deixar a saída do mesmo como se encontrava.

Resultados Obtidos

Os resultados foram obtidos alterando os valores das entradas e comparando com os valores das tabelas verdades das operações. Primeiramente se testou as operações lógicas e, posteriormente, as aritméticas.

Operações lógicas

As operações lógicas correspondem as seguintes operações:

- AND;
- NAND;
- OR;
- NOR;
- XOR

- XNOR.

Por se tratar de funções binárias a entrada “Te” pode ser desconsiderada, pois esta não interfere na saída, apenas as entradas A, B e os seletores, serão levadas em consideração.

Para ficar mais apresentável será passada a tabela verdade esperada de cada operação e a obtida na simulação, as expressões foram tiradas para conseguirmos chegar no circuito.

AND - opção no seletor 000

AND			Resultado da ULA			
A	B	S	A	B	Seletor	S
0	0	0	0	0	000	0
0	1	0	0	1	000	0
1	0	0	1	0	000	0
1	1	1	1	1	000	1

Expressão retirada pelo método de mapa de karnaugh:

$$S = A \cdot B$$

NAND - opção no seletor 001

NAND			Resultado da ULA			
A	B	S	A	B	Seletor	S
0	0	1	0	0	001	1
0	1	1	0	1	001	1
1	0	1	1	0	001	1
1	1	0	1	1	001	0

Expressão retirada pelo método de mapa de karnaugh:

$$S = \overline{A \cdot B}$$

OR – opção no seletor 010

OR			Resultado da ULA			
A	B	S	A	B	Seletor	S
0	0	0	0	0	010	0
0	1	1	0	1	010	1
1	0	1	1	0	010	1
1	1	1	1	1	010	1

Expressão retirada pelo método de mapa de karnaugh:

$$S = A + B$$

NOR - opção no seletor 011

NOR			Resultado da ULA			
A	B	S	A	B	Seletor	S
0	0	1	0	0	010	1
0	1	0	0	1	010	0
1	0	0	1	0	010	0
1	1	0	1	1	010	0

Expressão retirada pelo método de mapa de karnaugh:

$$S = \overline{A + B}$$

XOR - opção no seletor 100

XOR			Resultado da ULA			
A	B	S	A	B	Seletor	S
0	0	0	0	0	100	0
0	1	1	0	1	100	1
1	0	1	1	0	100	1
1	1	0	1	1	100	0

Expressão retirada pelo método de mapa de karnaugh:

$$S = A \oplus B$$

XNOR - opção no seletor 101

XNOR			Resultado da ULA			
A	B	S	A	B	Seletor	S
0	0	1	0	0	101	1
0	1	0	0	1	101	0
1	0	0	1	0	101	0
1	1	1	1	1	101	1

Expressão retirada pelo método de mapa de karnaugh:

$$S = \overline{A \oplus B}$$

Operações aritméticas

Diferente das operações lógicas aqui o “Te” será levado em consideração o mesmo vale para a saída “Ts”. São apenas duas operações aritméticas, somador e subtrator.

SOMADOR – opção no seletor 110

SOMADOR COMPLETO					Resultado da ULA					
A	B	Te	S	Ts	A	B	Te	Seletor	S	Ts
0	0	0	0	0	0	0	0	110	0	0
0	0	1	1	0	0	0	1	110	1	0
0	1	0	1	0	0	1	0	110	1	0
0	1	1	0	1	0	1	1	110	0	1
1	0	0	1	0	1	0	0	110	1	0
1	0	1	0	1	1	0	1	110	0	1
1	1	0	0	1	1	1	0	110	0	1
1	1	1	1	1	1	1	1	110	1	1

Expressão retirada pelo método de mapa de karnaugh:

$$S = A \oplus B \oplus Te \quad Ts = BTe + ATe + AB$$

SUBTRATOR – opção no seletor 111

SUBTRATOR COMPLETO					Resultado da ULA					
A	B	Te	S	Ts	A	B	Te	Seletor	S	Ts
0	0	0	0	0	0	0	0	111	0	0
0	0	1	1	1	0	0	1	111	1	1
0	1	0	1	1	0	1	0	111	1	1
0	1	1	0	1	0	1	1	111	0	1
1	0	0	1	0	1	0	0	111	1	0
1	0	1	0	0	1	0	1	111	0	0
1	1	0	0	0	1	1	0	111	0	0
1	1	1	1	1	1	1	1	111	1	1

Expressão retirada pelo método de mapa de karnaugh:

$$S = A \oplus B \oplus Te \quad Ts = \bar{A}B + \bar{A}Te + BTe$$

Conclusão

O objetivo deste experimento foi apresentar conceitos básicos sobre a ULA de 1 bit e desenvolvê-la no Logisim, programa de simulação, e testar o seu funcionamento.

A primeira etapa, foi necessário a pesquisa e entendimento do que é uma ULA, após chegar a definição, foi necessário determinar qual a ordem de processamento da ULA, quais as entradas, quais as saídas, definimos a ordem, e ficou da maneira que foi apresentada nas tabelas na seção resultados. Após isso foi feita as tabelas, retirado as expressões por mapa de karnaugh, e realizados testes no circuito para verificarmos se funcionavam, testado cada parte do circuito separadamente para depois juntar todas.

Depois das portas lógicas estarem feitas e funcionando, foi feito na mesma lógica o somador e o subtrator, mas estes são diferentes, eles foram criados em circuitos externos ao circuito principal, depois trouxéssemos eles ao circuito main, por meio de “caixinhas”, que é uma ferramenta disponibilizada pelo logisim, nela recebe o mesmo número de entrada e retorna a mesma quantidade de saídas e dentro da caixinha tem a lógica do circuito, estes também foram feitos os testes e as saídas e os Ts foram compatíveis com a tabela verdade do somador e subtrator.

Chegamos quase no fim, onde todos os operadores lógicos e aritméticos funcionavam, agora era necessário colocados em um multiplexador 8x1, pois o número de saídas dos operadores lógicos e mais os aritméticos eram 8, mas o multiplexador iria retornar somente 1 por vez, e para escolhermos qual, foi utilizado um pino de entrada de 3 bit, pois precisávamos ter 8 escolhas uma para cada operador da nossa ULA. Após unirmos o MUX, as saídas dos operadores e ao seletor e confirmarmos o funcionamento baseado nas tabelas verdades, podemos concluir que o projeto foi executado de maneira eficiente.

Bibliografia

Edgington, Padraic. 9. Building a 1-bit ALU. 2020. (5m56s). Disponível em:<<https://www.youtube.com/watch?v=y9EA-XBbRgg>>. Acesso em: 25 nov. 2020;

“ALU in Detail.” Tutorials, exploreembedded.com/wiki/ALU_in_Detail;

Organization of Computer Systems: Computer Arithmetic, www.cise.ufl.edu/~mssz/CompOrg/CDA-arith.html;

Education, University Of. “Lec 2 Digital Basics.” SlideShare, 24 Aug. 2014, www.slideshare.net/inam12/lec-2-digital-basics;

Hasan, Rakib. 9. One Bit ALU design. 2020. (2m21s). Disponível em:<<https://www.youtube.com/watch?v=x6jGmcLwqul>>. Acesso em: 25 nov. 2020;

Multiplexador, Fabiana Paixão disponível em: <https://www.youtube.com/watch?v=YQg9sRNw3aU>, acesso em 20, out 2022