



Informática
Circuitos Digitais(9889-2020-T32)

Trabalho teórico/prático referente ao “Projeto - Parte I”

Data: 26/11/2000

Professor: Nardênio Almeida Martins

Discentes

R.A.	NOME
116843	Gabriel Goncalves de Matos
118237	John William Vicente
99069	Vinicius Felipe Congeski Nunes



SUMÁRIO

Introdução e objetivos	3
1.1 Identificação da Experiência	3
Fundamentação Teórica	3
2.1 Componentes Utilizados	3
2.2 Descrição dos Componentes Utilizados	3
Procedimentos Executados	7
Descrição dos Procedimentos Executados	7
Definindo o que é uma ULA	7
Análise dos diagramas de uma ULA de um bit	7
Experiência 1: Desenvolvimento e simulação de circuito combinatório	9
Experiência 2: Verificação do funcionamento de ULA de 1 bit	12
Resultados Obtidos	13
Operações lógicas	13
Operações aritméticas	15
Conclusão	17
Bibliografia	17



1. Introdução e objetivos

1.1 Identificação da Experiência

Experiência 1: Desenvolvimento e simulação do circuito;

Experiência 2: Verificação do funcionamento de ULA de 1 bit;

1.2 Objetivos da Experiência

- Compreender o desenvolvimento e simulação do projeto de circuito combinacional: ULA de 1 bit (Unidade Lógica e Aritmética de 1 bit)

2. Fundamentação Teórica

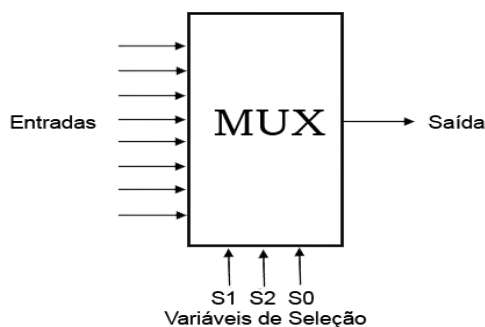
2.1 Componentes Utilizados

- 1 Multiplexador 8-para-1 (74LS151)
- 6 Porta Lógica AND (74LS08)
- 5 Porta Lógica OR (74LS32)
- 2 Porta Lógica XOR (74LS86)
- 5 inverter (74LS04)
- 2 Logic Display
- 7 Logic Switch 5V

2.2 Descrição dos Componentes Utilizados

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” no português, faz referência a uma operação lógica que aceita dois ou mais operandos, que sempre resulta em um valor lógico verdadeiro(1) 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(0).

Representação simbólica

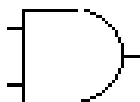


Tabela verdade da Porta AND

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

Porta Lógica OR: A porta 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

ser verdadeiro, ou seja, se existir um valor verdadeiro entre o conjunto das entradas, logicamente o valor resultante da saída dessa operação vai ser um valor verdadeiro.

Representação simbólica



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 operandos, 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 e vice-versa a saída resultará em um valor verdadeiro, porém caso ambos sejam verdadeiros(1) ou ambos forem falsos(0) o valor resultante será falso.

Representação simbólica



Tabela Verdade XOR

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

Inverter: A porta **inversora** tem apenas a função de trocar o valor passado por ela, por exemplo, se um valor verdadeiro é passado por uma porta inversora a saída do mesmo

será falso e a recíproca é verdadeira, quando o valor passado seja falso. A ideia em si é sempre trocar os valores que são passados pela porta inversora.

Representação simbólica

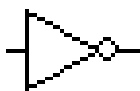
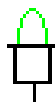


Tabela verdade NOT ou inversora

A	A'
0	1
1	0

Logic Display: Através deste componente é possível ver o resultados obtidos pelas portas lógicas, sendo 1(true) lâmpada acesa e 0(false) lâmpada apagada.

Representação simbólica no Circuit Maker



Logic Switch: As logic switches são espécies de chaves que podem estar ligadas ou fechadas, permitindo assim a livre escolha entre as duas condições aberta ou fechada. Uma chave quando está aberta ela passa um valor alto e quando fechada é passado um valor baixo, que seria no caso respectivamente um valor verdadeiro e um valor falso.

Representação simbólica no Circuit Maker



3. Procedimentos Executados

Descrição dos Procedimentos Executados

Definindo o que é uma ULA

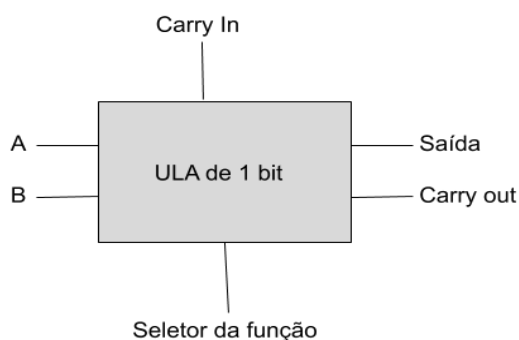
O processo de desenvolvimento da ULA de um bit ocorreu em alguns estágios e de uma forma progressiva foi elaborado, construído, testado e finalizado o projeto. Dessa forma, desde a busca por conteúdo, tanto o sugerido na aula quanto o que foi encontrado em outras fontes, foi utilizado para definir os conceitos e funcionalidades que a ULA possui e, conseqüentemente, ser possível 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; multiplicação e divisão. A ULA se encontra em componentes que envolvem processamento de dados, como o processador do computador, se configurando, dessa maneira, como sendo um dos componentes fundamentais à computação de dados.

Como visto a ULA possui várias funcionalidades e, por consequência, um arranjo de vários componentes eletrônicos, que permitem o seu bom funcionamento.

Análise dos diagramas de uma ULA de um bit

Anterior a construção da ULA de um bit, em um simulador, foi necessário fazer a análise de diagramas que remetem a mesma, como esse logo abaixo:

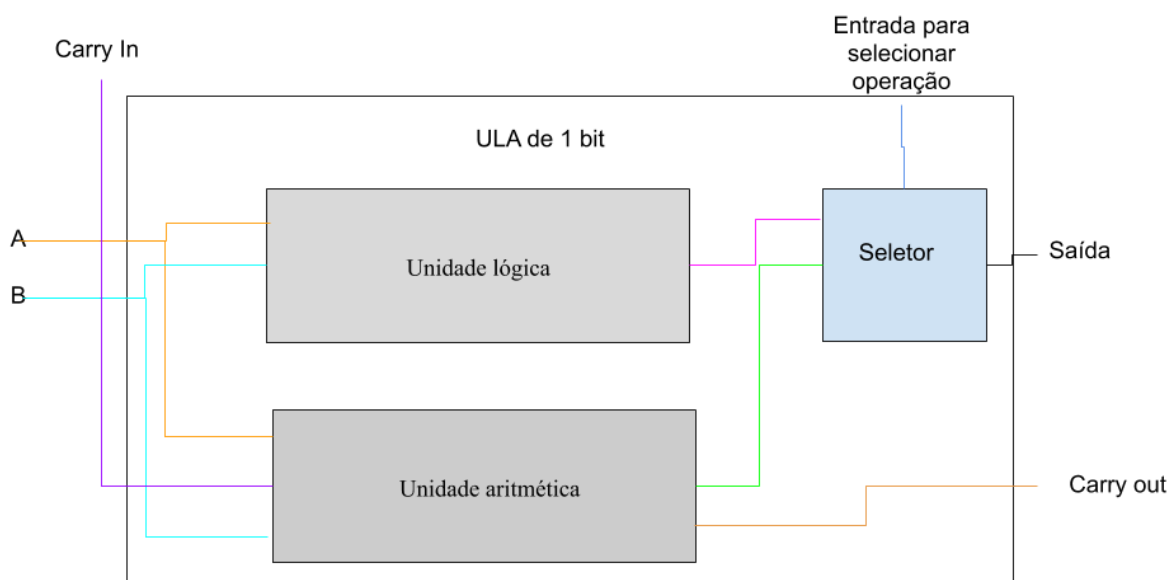


Representação simbólica de uma ULA de 1 bit

No diagrama podemos observar que há um conjunto de entradas e no final temos duas saídas uma referente a saída do processamento da ULA, por isso “1” bit, e outra referente ao “carry out” da unidade aritmética.

As entradas A e B serão os valores que passarão por algum processo que compõe as unidades da ULA, são dois pois como nossa lógica matemática é composta por operações binárias. A entrada “carry in” é necessária apenas pela parte aritmética e, totalmente, desconsiderada pelo núcleo lógico.

Como a ULA possui várias operações é necessário uma espécie de interface para que seja escolhida a operação que se deseja realizar em cima dos valores A, B e “carry in”, este último apenas em operações de natureza aritmética. Dessa forma se monta um conjunto de entradas de dados com a única função de selecionar as operações que serão realizadas. Um paralelo que se pode tirar é considerar a ULA como um menu com opções que podem ser escolhidas, no caso de sistemas digitais terá diversas operações que ao ser selecionadas irão gerar um resultado esperado.



Esta imagem revela de forma ilustrativa a essência da ULA

A ULA desenvolvida para esse trabalho apresenta as seguintes operações:

- Lógicas:
 - AND;
 - NAND;
 - OR;
 - NOR;
 - XOR;
 - XNOR;
- Aritméticas:
 - SOMADOR COMPLETO;
 - SUBTRATOR COMPLETO.

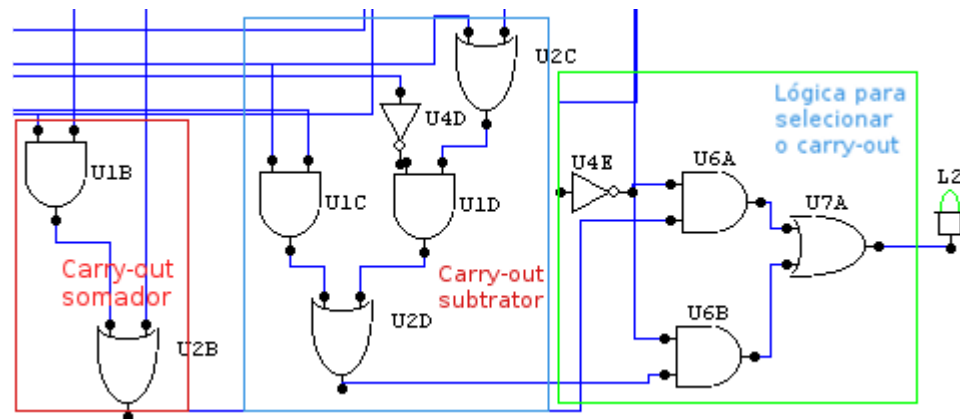


Experiência 1: Desenvolvimento e simulação de circuito combinatório

O projeto da ULA de um bit foi todo realizado através do simulador **Circuit Maker** 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 **logic switch**:
 - A, B e “carry In” os bits de entrada;
 - **Três** logic switches que juntos compõem a interface para a seleção da operação, desde o começo do projeto imaginava-se que a seleção seria através de um **multiplexador 8x1**;
 - **“Enable”** esse logic switch foi colocado devido a uma exigência de uma entrada do multiplexador, esse logic switch é quem habilita o funcionamento do multiplexador. Porém, este foi colocado por uma exigência da simulação e não deve ter seu valor alterado, que no caso é “0”.
2. 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 foi conectados **inversores** nas saídas destas últimas. Dessa forma, não foi necessário a utilização de outras portas na parte lógica, com isso o sistema ficou mais enxuto e organizado, porém com potencial de realizar todas as operações pedidas.
3. Para finalizar a parte lógica todas as portas foram conectadas com as respectivas entradas A e B, porém as saídas das portas lógicas, que forneceriam os resultados das operações, não foram conectadas a nenhum lugar;
4. Com o uso da tabela verdade e do mapa de Karnaugh foi esquematizado a parte aritmética da ULA;
5. Como o somador e subtrator usam o mesmo esquema de circuitos, não foi necessário a elaboração de dois para cada. Como esse esquema utiliza combinações de componentes que se encontram na unidade aritmética, no caso a porta XOR, não foi necessário a repetição dos mesmos. Com isso a parte aritmética economizou a utilização de componentes e tornou o projeto mais eficiente no que tange a utilização e reaproveitamento de partes do mesmo;

6. Através do mapa de Karnaugh e das tabelas verdades do somador e subtrator completo foi possível montar os circuitos que formam o “carry out” das respectivas operações;
7. Foi montado uma lógica, através de um esquema de seletor, para que seja selecionado o “carry out” correspondente a determinada operação aritmética, levando em consideração o último bit da seleção.



Parte que envolve o carry-out do subtrator e do somador

8. Com as partes aritméticas e lógicas montadas foi necessário a adição de um multiplexador, 8x1, com o intuito de conectar todas as operações e a interface de seleção, que no caso são três logic switches;
9. Dessa maneira para selecionar cada operação deve-se combinar as chaves de seleção como na imagem abaixo:

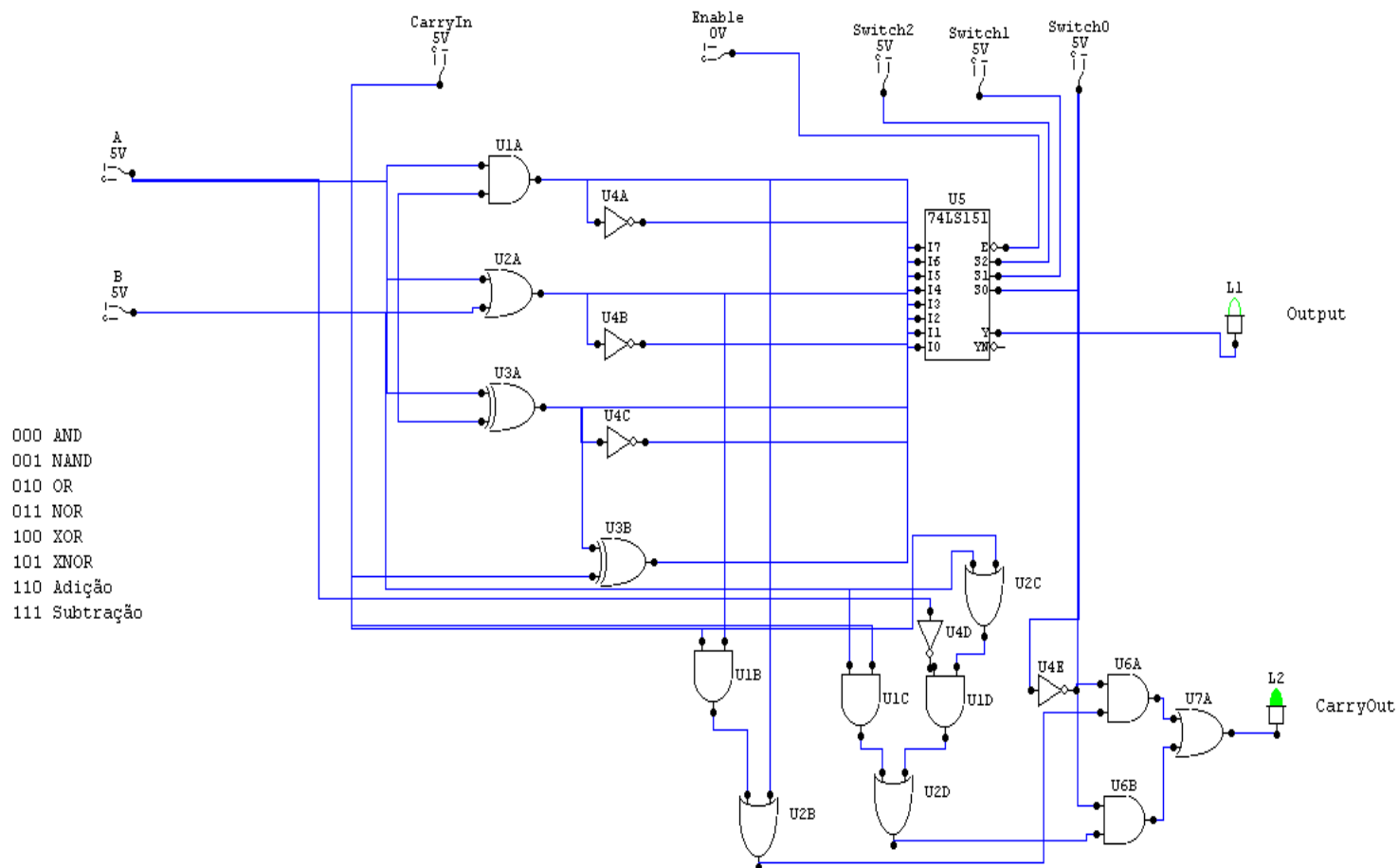
```

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

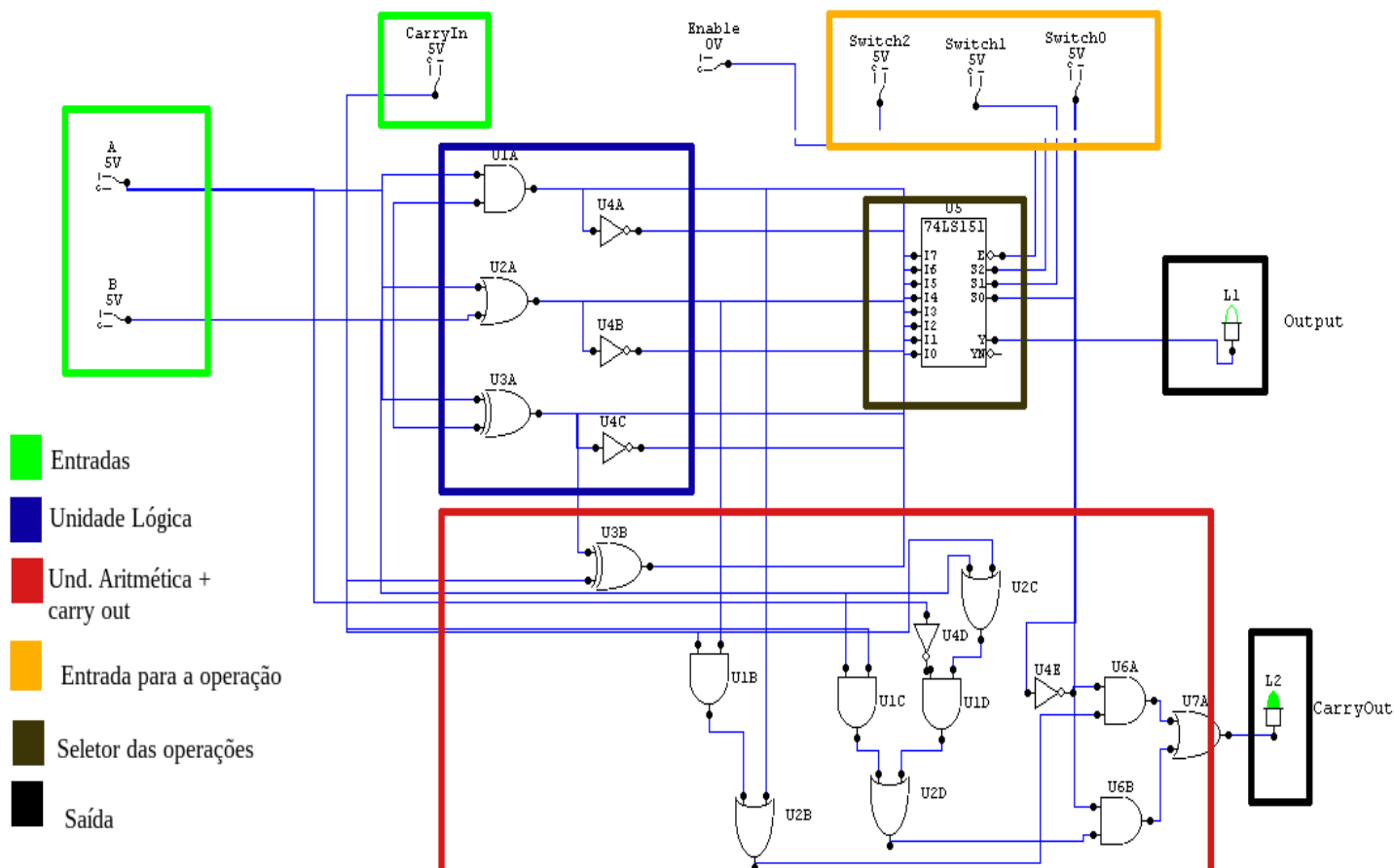
```

10. Na saída do multiplexador e na saída do seletor do “carry out” fora colocado um display de saída para monitorar os valores finais;
11. Dessa maneira foi possível testar e utilizar a ULA.

Circuito final do projeto



Circuito com as partes destacadas



Experiência 2: Verificação do funcionamento de ULA de 1 bit

Como foi utilizado o Circuit Maker 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 modificado o estado da simulação trocando “**analog mode**” para “**digital mode**”;
2. Para simular o circuito foi necessário clicar em run/stop e dessa forma o circuito começou a funcionar;
3. Foi feito os testes alterando os valores de entrada, A, B e “carryIn”, e verificando com as tabelas verdades das operações se as mesmas selecionadas na ULA estavam em seu estado correto de funcionamento;
4. Foi visto que o “carry-out” 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 “CarryIn” pode ser desconsiderada, pois esta não interfere na saída, apenas as entradas A, B e os seletores, que serão denominados sw0, sw1 e sw2, serão levadas em consideração.

Para ficar mais apresentável será passada a tabela verdade esperada de cada operação e a obtida na simulação.

AND - opção nos seletores 000

AND			Resultado da ULA					
A	B	S	A	B	SW0	SW1	SW2	S
0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0
1	1	1	1	1	0	0	0	1



NAND - opção nos seletores 001

NAND			Resultado da ULA					
A	B	S	A	B	SW0	SW1	SW2	S
0	0	1	0	0	0	0	1	1
0	1	1	0	1	0	0	1	1
1	0	1	1	0	0	0	1	1
1	1	0	1	1	0	0	1	0

OR - opção nos seletores 010

OR			Resultado da ULA					
A	B	S	A	B	SW0	SW1	SW2	S
0	0	0	0	0	0	1	0	0
0	1	1	0	1	0	1	0	1
1	0	1	1	0	0	1	0	1
1	1	1	1	1	0	1	0	1

NOR - opção nos seletores 011

NOR			Resultado da ULA					
A	B	S	A	B	SW0	SW1	SW2	S
0	0	1	0	0	0	1	1	1
0	1	0	0	1	0	1	1	0
1	0	0	1	0	0	1	1	0
1	1	0	1	1	0	1	1	0



XOR - opção nos seletores 100

XOR			Resultado da ULA					
A	B	S	A	B	SW0	SW1	SW2	S
0	0	0	0	0	1	0	0	0
0	1	1	0	1	1	0	0	1
1	0	1	1	0	1	0	0	1
1	1	0	1	1	1	0	0	0

XNOR - opção nos seletores 101

XNOR			Resultado da ULA					
A	B	S	A	B	SW0	SW1	SW2	S
0	0	1	0	0	1	0	1	1
0	1	0	0	1	1	0	1	0
1	0	0	1	0	1	0	1	0
1	1	1	1	1	1	0	1	1

Operações aritméticas

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

SOMADOR COMPLETO - opção nos seletores 110

SOMADOR COMPLETO					Resultado da ULA							
A	B	Cin	S	Cout	A	B	Cin	SW0	SW1	SW2	S	Cout
0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	1	1	1	0	1	0
0	1	0	1	0	0	1	0	1	1	0	1	0
0	1	1	0	1	0	1	1	1	1	0	0	1



1	0	0	1	0	1	0	0	1	1	0	1	0
1	0	1	0	1	1	0	1	1	1	0	0	1
1	1	0	0	1	1	1	0	1	1	0	0	1
1	1	1	1	1	1	1	1	1	1	0	1	1

SUBTRATOR COMPLETO - opção nos seletores 111

SUBTRATOR COMPLETO					Resultado da ULA							
A	B	Cin	S	Cout	A	B	Cin	SW0	SW1	SW2	S	Cout
0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	1	1	1	0	0	1	1	1	1	1	1
0	1	0	1	1	0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	0	1
1	0	0	1	0	1	0	0	1	1	1	1	0
1	0	1	0	0	1	0	1	1	1	1	0	0
1	1	0	0	0	1	1	0	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1



4. Conclusão

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

A experiência 1 teve como objetivo o desenvolvimento da ULA de 1 bit propriamente dita, tal como sua parte lógica e sua parte aritmética. Os logic switches A, B, C in serviram como a entrada de valores e os logic switches “switch0”, “switch1” e “switch2”, também, como entrada, porém esses de forma especial têm como responsabilidade a seleção das operações que a ULA possui. Todas as partes da ULA foram conectadas a um Multiplexador 8x1 que serviu como seletor dos valores das operações.

A experiência 2 foi focada no funcionamento e nos testes que a ULA deveria passar. Dessa maneira, se testou todas as operações, tanto lógicas quanto aritméticas, e comparou-se com os valores contidos nas tabelas verdades das respectivas operações. Como todos os testes resultaram nos valores esperados o bom funcionamento da ULA foi retificado.

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=x6jGmcLwquI>>. Acesso em: 25 nov. 2020;
- FLOYD, THOMAS. SISTEMAS DIGITAIS: FUNDAMENTOS E APLICAÇÕES. 9ª EDIÇÃO. BOOKMAN - COMPANHIA EDITORA.