UNIVERSIDADE FEDERAL DA FRONTEIRA SUL

CAMPUS DE CHAPECÓ

CURSO DE CIÊNCIA DA COMPUTAÇÃO

MATHEUS VINÍCIUS TODESCATO

MICHEL FELIPE WELTER

WILLIAN BORDIGNON GENERO

RELATÓRIO DA ULA (UNIDADE LÓGICA ARITMÉTICA)

DISCIPLINA DE CIRCUITOS DIGITAIS

CHAPECÓ

2017

RESUMO

Este trabalho traz as informações e métodos de como foi encontrado uma solução para que um circuito digital seja capaz de trabalhar com 4 bits, no valor de 0 a 15, desempenhando várias funções aritméticas e lógicas escolhidas pelo usuário.

SUMÁRIO

1. LISTA DE FIGURAS……………………………………………..4
2. LISTA DE TABELAS…………………………………………….5
3. OBJETIVO……………………………………………………...6
4. INTRODUÇÃO………………………………………………....7
5. METODOLOGIA………………………………………………8
5.1. F = A AND (NOT B)..9
5.2. F = A XOR B...9
5.3. F = NOT A...10
5.4. F = NOT (A OR B)..11
5.5. F = A1A0 * B1B0...12
5.6. F = A + B...15
5.7. F = A - B...17
5.8. F = A + 1...20
6. RESULTADOS……………………………………..…….22
7. DISCUSSÃO…………………………………………….....23
8. CONCLUSÃO………………………………………………24
REFERÊNCIAS……………………………………………..26

1.0 - LISTA DE FIGURAS

Imagem 1: circuito representante de F = A AND (NOT B)………..………9

Imagem 2: circuito representante de F = A XOR B…………..…..………10

Imagem 3: circuito representante de F = NOT A……………..…………...11

Imagem 4: circuito representante de NOT (A OR B)...............…………...12

Imagem 5: circuito representante de F = A1A0 * B1B0………..……....14

Imagem 6: circuito representante de um somador de 1 bit……..………...16

Imagem 7: circuito representante de F = A + B……………………………17

Imagem 8: circuito representante de um subtrator de 1 bit………………....19

Imagem 9: circuito representante de F = A - B……………………………..20

Imagem 10: circuito representante de F = A + 1…………………………....21

Imagem 11: circuito representante da pré-ULA…………………………….22

Imagem 12: circuito representante da ULA………………………………...23

2.0 - LISTA DE TABELAS

Tabela 1: tabela verdade para F=A1A0 * B1B0……………………….…12

Tabela 2: tabela verdade de F = A + B……………………………….....15

Tabela 3: tabela verdade de F = A - B………………………………...17

3.0 - OBJETIVO

Realizar a construção de um circuito digital que seja capaz de desempenhar diversas funções aritméticas e lógicas determinadas pelo usuário.

4.0 - INTRODUÇÃO

Circuitos Digitais são circuitos eletrônicos que utilizam de sinais elétricos em dois níveis de corrente (ou tensão) para definir a representação de valores binários. É essencial a fixação desse assunto para aprendizado sobre os diversos tipos de máquinas e circuitos existentes assim possibilitando o trabalho relacionado ao tema.

Nosso trabalho tem como objetivo criar uma ULA (Unidade Lógica e Aritmética), onde são desempenhadas diversas funções aritméticas. Para tudo foi necessário conhecer todos os fundamentos da lógica digital, utilização de multiplicadores, somadores, subtratores, multiplexadores entre outros.

Primeiramente foi necessário entender como funcionaria cada parte da nossa ULA para que todas as operações pudessem ser feitas. Foi necessário todo o conhecimento adquirido durante o semestre e principalmente sobre circuitos combinacionais e circuitos aritméticos.

5.0 - METODOLOGIA

Para encontrar a solução do problema proposto primeiramente separamos as operações solicitadas para o grupo 6 e pensamos exclusivamente para cada uma delas.

5.1 - F = A AND (NOT B)

Começando por F = A AND (NOT B), criamos um circuito adicional no software de simulação de circuitos lógicos digitais Logisim. Com o circuito criado adicionamos 8 pinos para que trabalhem com os valores de entrada, 4 pinos para o valor de A que corresponde de 0 até 15 e 4 para o valor de B que por sua vez também vai de 0 a 15. Após, barramos os quatro valores binários de B usando portas NOT e por fim cada valor barrado de B passa em uma porta AND com o mesmo bit de A. Saindo assim a solução da operação. Para as flags o valor da saída Overflow será sempre igual a 0 para isso usaremos a conexão terra. E para satisfazer a saída ZERO, que deve ser 1 se F = 0, teremos os 4 bits da solução unidos por uma porta NOR assim se o valor de F for 0 a saída será 1.

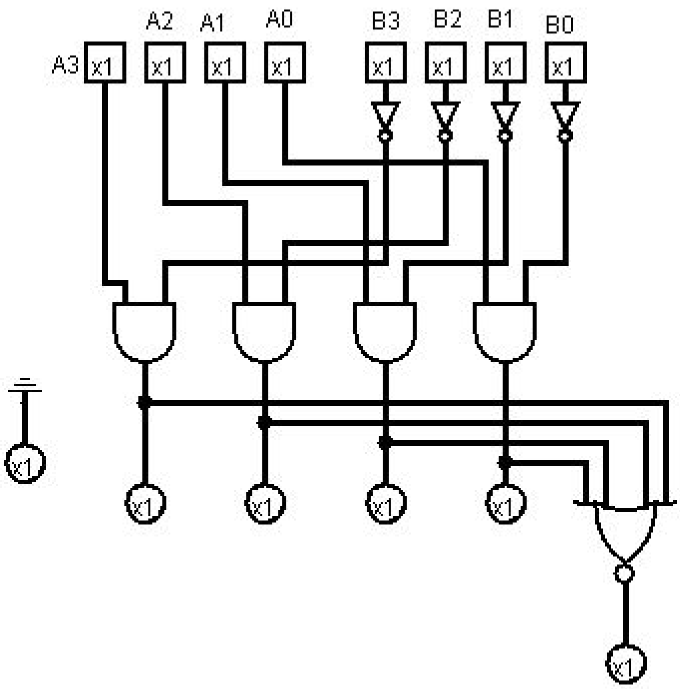


Imagem 1: circuito representante de F = A AND (NOT B).

5.2 - F = A XOR B

Para a operação lógica A XOR B, trabalhamos com o mesmo princípio a entrada será feita através de 4 pinos tanto para A como para B. Cada bit passará por uma porta XOR unindo A e B e a saída de F será 4 bits indo do valor de 0 a 15 na transformação de binário para decimal. As flags serão satisfeitas após os 4 bits da solução passarem por uma porta NOR solucionando a saída ZERO. E o Overflow será sempre 0 então usaremos a conexão terra.

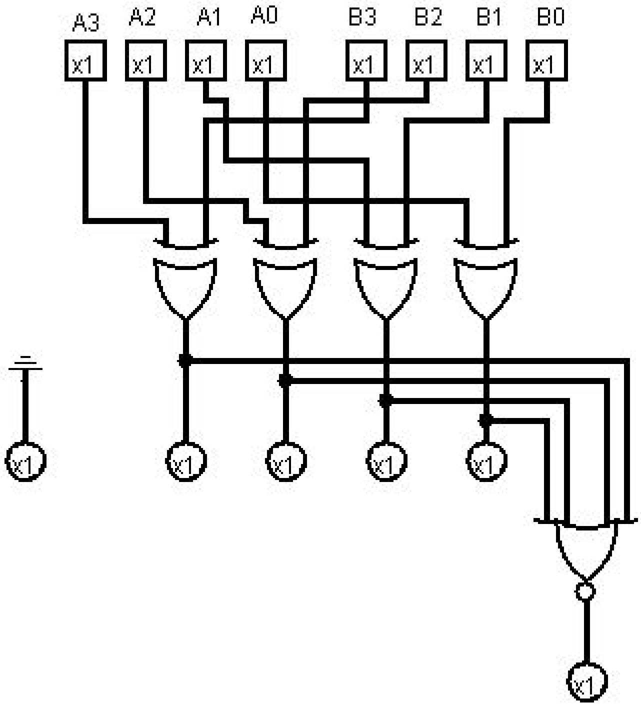


Imagem 2: circuito representante de F = A XOR B.

5.3 - F = NOT A

A operação mais simples. Só são necessários 4 bits de entrada e por sua vez 4 pinos para trabalhar com os valores 0 e 1, cada bit passará por uma inversora, porta NOT, e aí teremos a solução. A saída ZERO só ativará se a entrada for o valor 7 ou 1111, para isso usaremos uma porta NOR que une os bits da solução. E overflow será sempre 0 então será ligado na conexão terra.

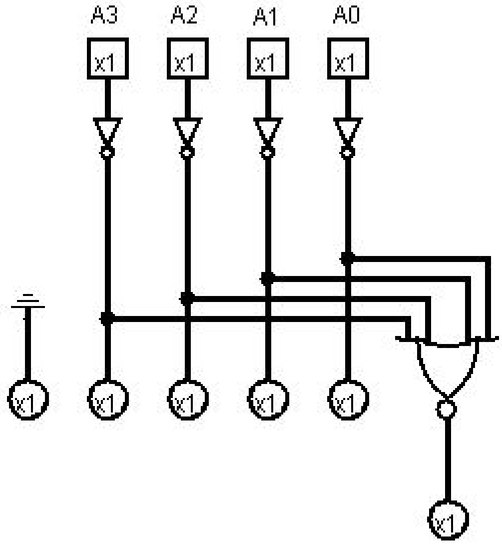


Imagem 3: circuito representante de F = NOT A.

5.4 - F = NOT (A OR B)

A última operação lógica representada pela seleção 011, será solucionada respeitando a prioridade dos parênteses então os 8 bits 4 de cada entrada separados por bit passarão por uma porta OR e em sequência serão barrados por uma porta NOT e teremos a solução. A saída ZERO será os 4 bits da solução passando por uma porta NOR e overflow será sempre 0.

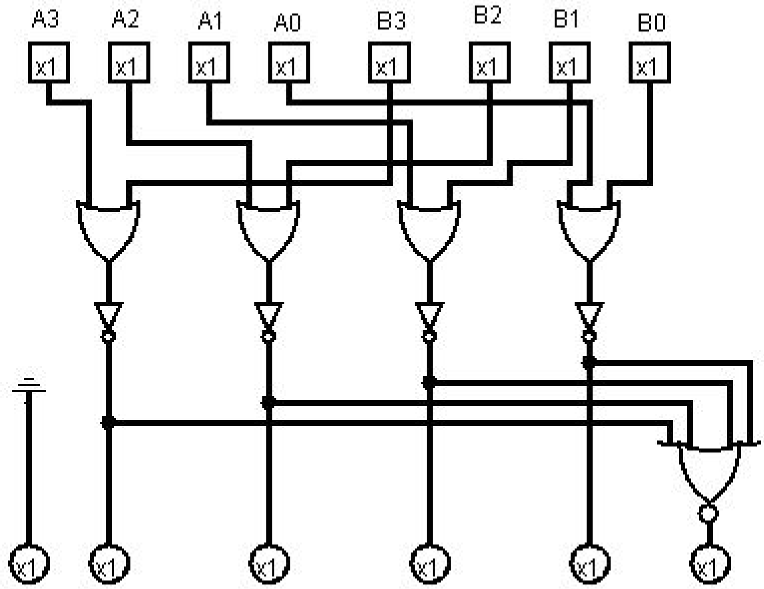


Imagem 4: circuito representante de NOT (A OR B).

5.5 - F = A1A0 * B1B0

A operação selecionada pelo valor 100 é a primeira operação aritmética e também a mais difícil. Para ser solucionada devemos encontrar as possibilidades através da tabela verdade.

A1	A0	B1	B0	S3	S2	S1	S0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0

0	**1**	**1**	**1**	**0**	**0**	**1**	**1**
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**0**	**1**	**0**	**0**	**1**	**0**
1	**0**	**1**	**0**	**0**	**1**	**0**	**0**
1	**0**	**1**	**1**	**0**	**1**	**1**	**0**
1	**1**	**0**	**0**	**0**	**0**	**0**	**0**
1	**1**	**0**	**1**	**0**	**0**	**1**	**1**
1	**1**	**1**	**0**	**0**	**1**	**1**	**0**
1	**1**	**1**	**1**	**1**	**0**	**0**	**1**

Tabela 1: tabela verdade para F=A1A0 * B1B0.

A equação resultante desta tabela verdade é enorme para ter maior facilidade de trabalho usaremos o software Karma que possibilita a simplificação algébrica de uma expressão. Após, teremos uma expressão simplificada ao menor tamanho possível. Abaixo encontramos o circuito representante.

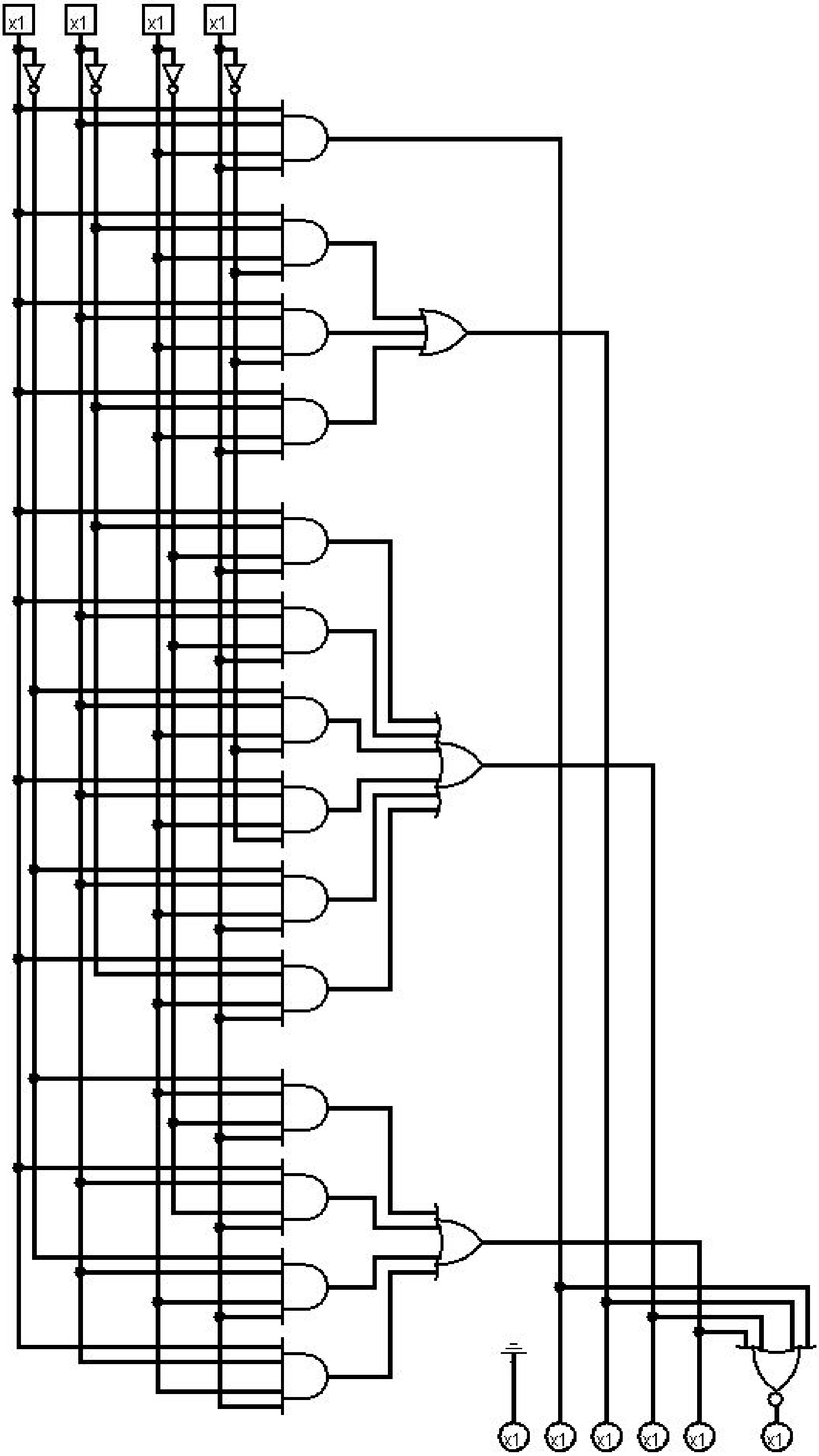


Imagem 5: circuito representante de F = A1A0 * B1B0.

5.6 - F = A + B

A operação de somar A com B só é possível ser solucionada pela tabela de possibilidades de soma abaixo:

A	B	C	S	Carry
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Tabela 2: tabela verdade de F = A + B.

A e B são valores quaisquer em binário que serão somados e C é a sobra de quando ultrapassa o valor e ele será acrescentado na próxima sentença a esquerda. S é o valor de saída menos importante após a adição e carry tem a mesma função de C, porém, é uma saída. Após a construção da tabela conseguimos retirar uma equação que ajudará para a obtenção da solução.

Abaixo a equação:

S = (~A * ~B * C) + (~A * B * ~C) + (A * ~B * ~C) + (A * B * C).

Carry = (~A * ~B * C) + (A * ~B * C) + (A * B * ~C) + (A * B * C).

Com essa equação conseguimos construir um pequeno somador de 1 bit.

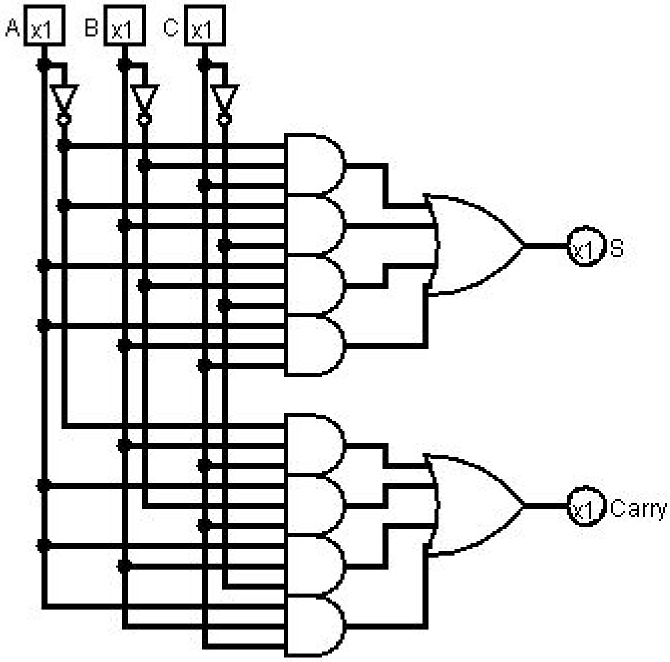


Imagem 6: circuito representante de um somador de 1 bit.

Esse somador será utilizado para fazermos o somador de A com B. Para isso usaremos uma ferramenta do software Logisim que permite adicionarmos um circuito dentro do outro, assim criamos um somador completo de 4 bits. Este somador funciona de forma simples, possui 3 entradas e 2 saídas, sendo que uma das saídas será usada como entrada para a seguinte, cada somador de bit único tem como entrada o valor do mesmo bit de A e B, por exemplo: a0 e b0, e o carry que é a sobra da adição. Por isso, o carry só resulta em um valor ao final de todas os bits somados, e se o carry estiver com o valor de 1 significa que F é maior que 15 e assim o overflow será ligado toda vez que aconteça esse excesso. A solução é cada bit menos significativo resultante do somador. A saída ZERO é cada bit da solução e também do overflow passando por uma porta NOR que será ativada somente quando F for igual a 0.

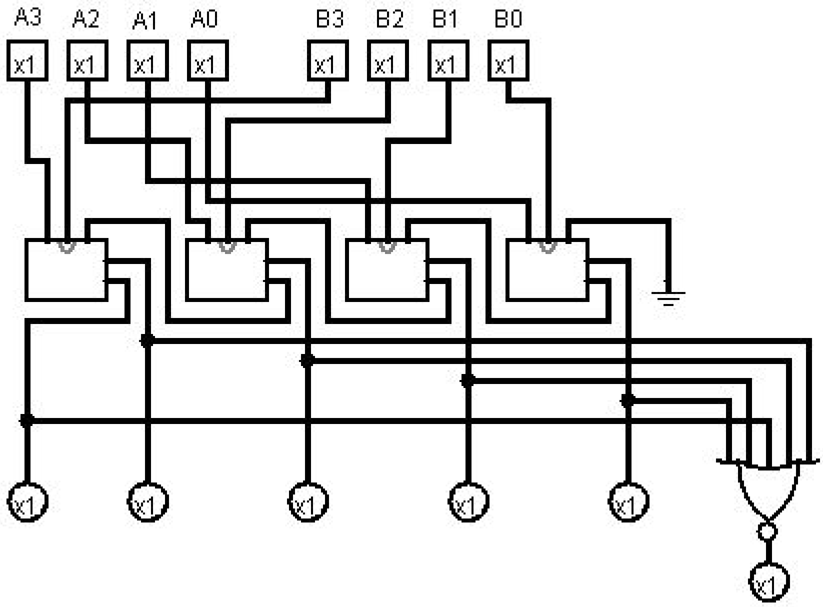


Imagem 7: circuito representante de F = A + B.

5.7 - F = A - B

A operação de subtração funciona de forma semelhante a de adição. Com a tabela abaixo é possível entender melhor como foi encontrada a solução para a operação.

A	B	C	S	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0

1	**0**	**1**	**0**	**0**
1	**1**	**0**	**0**	**0**
1	**1**	**1**	**1**	**1**

Tabela 3: tabela verdade de F = A - B.

Possui os 3 valores que são usados em uma subtração, A e B são valores quaisquer e C é o ato de "pedir emprestado", ou borrow em inglês, S será a solução que é dada pelos bits menos significativos e Borrow tem a mesma função do carry que foi usado nos somadores. Com isso conseguimos encontrar um circuito que conseguirá solucionar a operação.

S = (A (+) B) (+) C.

Borrow = (~A * C) + (~A * B) + (B * C).

A partir dessa equação conseguimos chegar em um circuito que funciona como um subtrator de 1 bit.

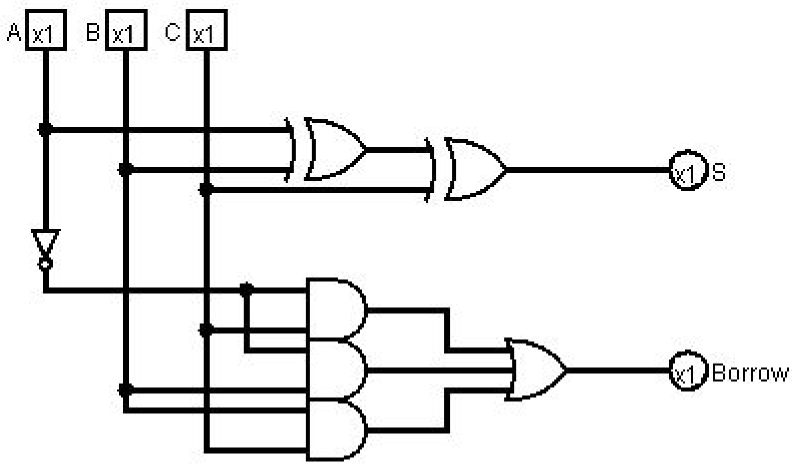


Imagem 8: circuito representante de um subtrator de 1 bit.

Com esse subtrator de 1 bit conseguimos montar um circuito que satisfaça a operação aritmética solicitada.

Exatamente distribuído como no somador cada bit de A é subtraído pelo de mesma importância de B, por exemplo A2 e B2 e o borrow inicia sempre valendo 0 para não ocasionar uma mudança no resultado e após passar pelo primeiro circuito do subtrator de 1 bit ele passará a ter um valor dependente da operação em si. Quando o valor introduzido para B for maior que o de A o overflow ativará e o resultado será em complemento de 2 pois em binário não há valores negativos, porém com sinal e magnitude, complemento de 1 e outros é possível trabalhar com valores negativos, no presente trabalho será em complemento de 2. Se o de A e de B for igual então teremos a saída ZERO ativada.

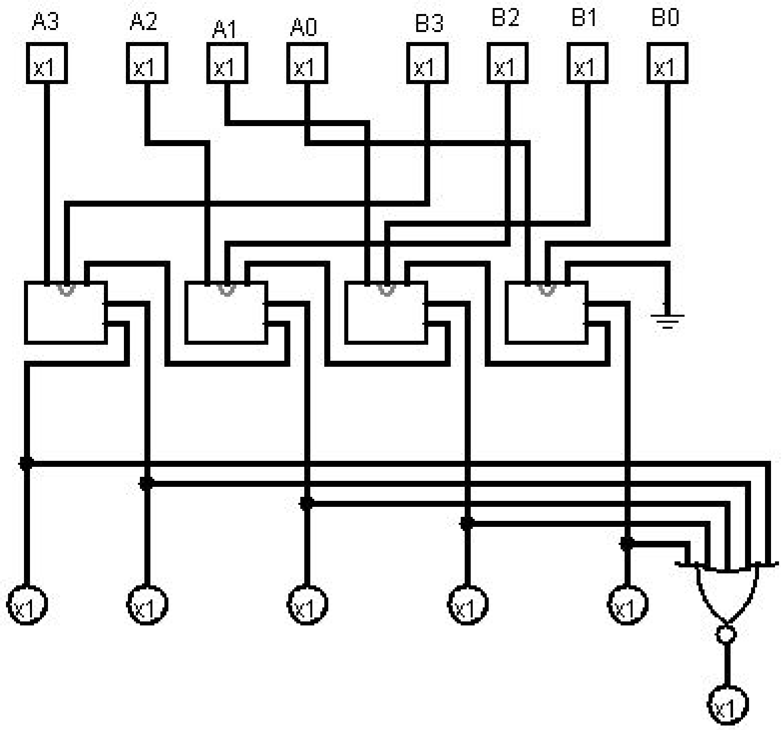


Imagem 9: circuito representante de F = A - B.

5.8 - F = A + 1

A última opção aritmética que pode ser selecionada por 111 é muito simples e foi solucionada a partir da operação F = A + B a única mudança é que não mais trabalhamos com um valor de B então ele será ligado no circuito terra e o carry inicial será ligado na fonte possuindo assim o valor de 1. A saída ZERO estará sempre desativada pois um valor somando 1 será maior que 0. E caso F maior que 15 então a saída overflow será ativada.

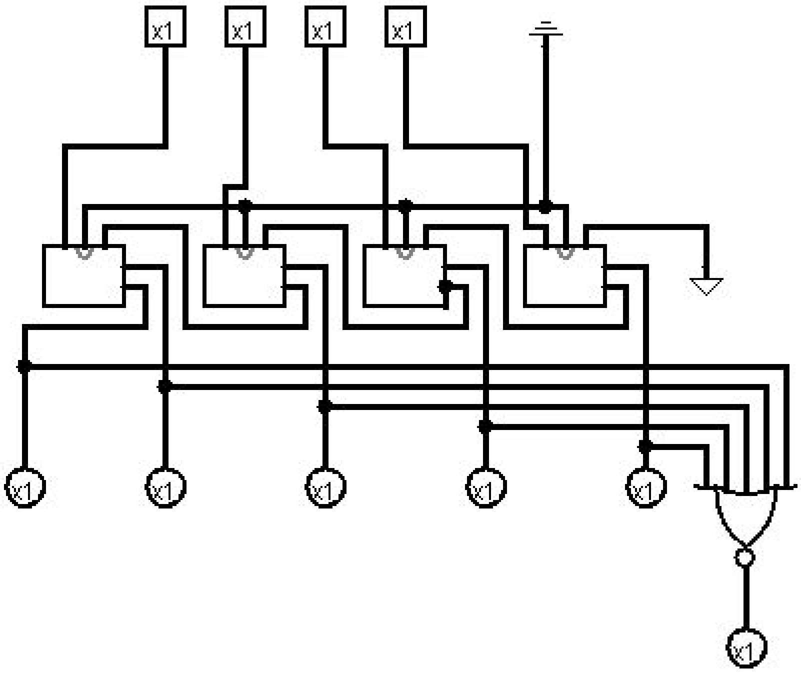


Imagem 10: circuito representante de F = A + 1.

Com todo essas operações resolvidas devemos encontrar um método para agrupar esses pequenos circuitos em um funcional. Para isso teremos uma nova entrada que será de 3 bits e irá controlar a seleção de 6 multiplexadores.

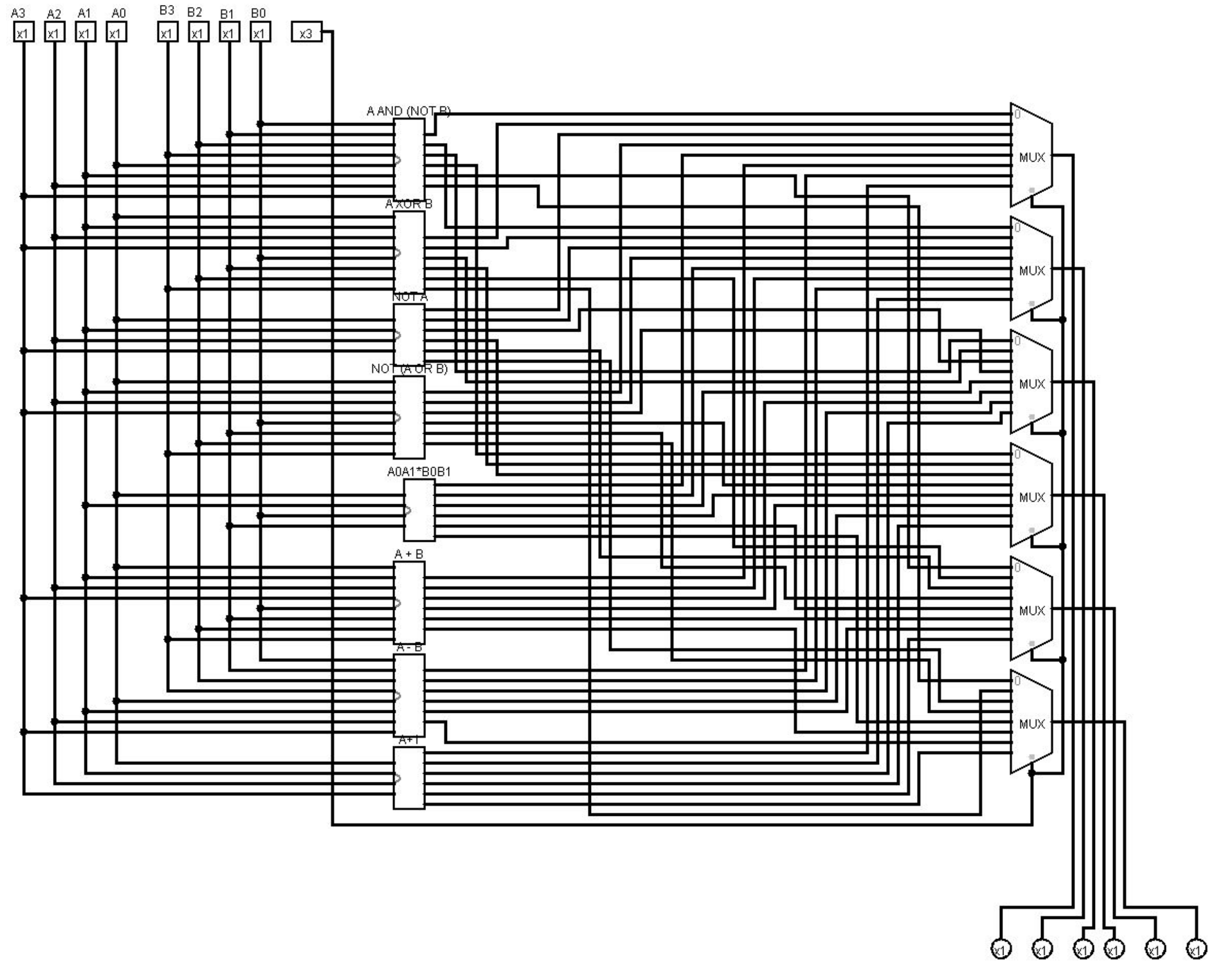


Imagem 11: circuito representante da pré-ULA.

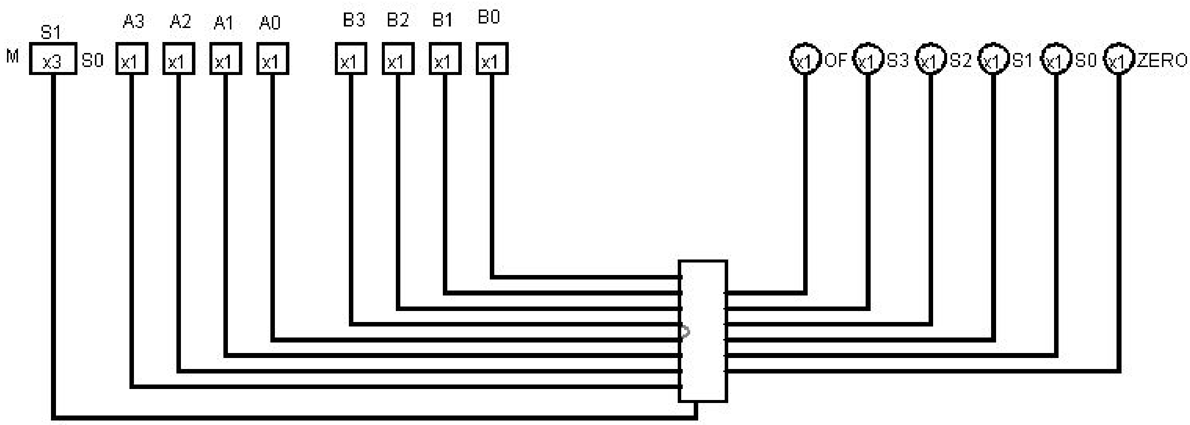
6.0 - RESULTADOS

Considerando que os tempos de propagação das portas lógicas do maior CI encontrado no projeto seja:

- 54LS11 - 13ns(médio) e 18ns(crítico);
- CD4072B-125ns e 250ns.

Tendo uma quantidade de 32 54LS11 e 9 CD4072B temos um tempo de propagação médio de 1541ns e crítico de 2826ns.

E com a pré Unidade Lógica e Aritmética com uma pequena simplificação chegamos ao circuito final. Abaixo imagem:



000	F = A AND (NOT B)	100	F = a1a0 * b1b0
001	F = A XOR B	101	F = A + B
010	F = NOT A	110	F = A - B
011	F = NOT (A OR B)	111	F = A + 1

Imagem 12: circuito representante da ULA.

7.0 - DISCUSSÃO

Levando em consideração tudo o projeto desenvolvido ganhamos uma "skill" a mais em construir algo parecido, por exemplo, com uma calculadora.

É bom sabermos como todo esse conhecimento pode nos ajudar futuramente e como também nos ajudou a entender como certos equipamentos funcionam.

Interessante ressaltarmos que o funcionamento da ULA se assemelha a uma calculadora. Se implementarmos mais funções e mais bits para ela trabalhar ela poderia servir como uma calculadora funcional que realiza todas as operações básicas.

8.0 - CONCLUSÃO

Com a construção da ULA desenvolvemos e utilizamos muito bem o que foi aprendido durante o semestre em Circuitos Digitais. A utilização de programas como o LogiSim também foi essencial e com certeza nos ajudará em próximos projetos.

Utilizando desses conhecimentos conseguimos desenvolver a ULA com 100% de funcionalidade em todas as funções aritméticas e lógicas necessárias. Com esse trabalho podemos concluir que o conhecimento adquirido foi muito bem utilizado e que com certeza tivemos uma noção muito ampla sobre o que podemos fazer com todo esse aprendizado.

REFERÊNCIAS

https://moodle-academico.uffs.edu.br/pluginfile.php/164051/mod_resource/content/1/circuitos%20combinacionais_guntzel%20ufpel.pdf.
Arquivo deixado para consulta.

http://www.datasheetcatalog.com/datasheets_pdf/5/4/L/S/54LS11.shtml

http://www.datasheetcatalog.com/datasheets_pdf/C/D/4/0/CD4072BME4.shtml