



**UNIVERSIDADE FEDERAL DA FRONTEIRA SUL  
CAMPUS CHAPECÓ  
CIÊNCIA DA COMPUTAÇÃO**

**IGOR LUIZ GONÇALVES TOMAZ  
FABIO ANTÔNIO RIBEIRO DOS SANTOS  
DANIEL BRUM**

**CIRCUITOS DIGITAIS  
BATALHA NAVAL - RELATÓRIO AULA PRÁTICA**

**CHAPECÓ  
2016**

## RESUMO

Seguindo as regras do jogo "Batalha Naval" e utilizando 6 possíveis entradas, sendo 3 disponibilizadas para cada um dos jogadores. Faz-se necessário a criação de 2 módulos, onde as 3 primeiras entradas que pertencerá ao jogador J1 dará início ao jogo no primeiro módulo e posteriormente as 3 entradas do jogador J2 que adentrará o segundo módulo. Para o primeiro módulo foi construído a tabela verdade utilizando as entrada A, B, C, dando origem a 8 possíveis resultados, onde serão utilizadas apenas 6 saídas. Destas 6 saídas, utilizando a codificação escolhida pelos desenvolvedores foram geradas 3 resultados, para cada uma, gerando assim, as saídas S1, S2 e S3 onde serão as 3 primeiras entradas do segundo módulo.

O segundo módulo é composto por 6 entradas, sendo as 3 primeiras S1, S2 e S3, saídas resultantes do primeiro módulo e as 3 jogadas do segundo jogador, no caso A2, B2 e C2. Este segundo módulo utiliza de um circuito comparador, que através de portas XNOR realiza a comparação com as 3 primeiras jogadas e caso sejam iguais, o jogador 2 terá ganhado aquela rodada.

## SUMÁRIO

<b>RESUMO.....</b>	<b>2</b>
<b>LISTA DE FIGURAS.....</b>	<b>4</b>
<b>LISTA DE TABELAS.....</b>	<b>5</b>
<b>OBJETIVO.....</b>	<b>6</b>
<b>INTRODUÇÃO.....</b>	<b>7</b>
<b>MATERIAIS.....</b>	<b>8</b>
<b>METODOLOGIA.....</b>	<b>9</b>
CODIFICAÇÃO DE ENTRADAS E SAIDAS.....	9
ELABORAÇÃO DA TABELA VERDADE.....	10
CONSTRUÇÃO DO CIRCUITO LÓGICO NO LOGISIM A PARTIR DA TABELA VERDADE.....	11
SIMPLIFICAÇÃO DAS EXPRESSÕES ALGÉBRICAS.....	15
UTILIZANDO NETLIST PARA IMPLANTAÇÃO NA BREADBORD.....	17
CÁLCULO DO CAMINHO CRÍTICO.....	19
UTILIZAÇÃO DO SOFTWARE PARA SIMULAÇÃO DA BREADBORD.....	20
<b>RESULTADO.....</b>	<b>22</b>
<b>DISCUSSÃO.....</b>	<b>23</b>
<b>CONCLUSÃO.....</b>	<b>24</b>
<b>REFERÊNCIAS.....</b>	<b>25</b>

LISTA DE FIGURAS

FIGURA 1.....11

FIGURA 2.....12

FIGURA 3.....14

FIGURA 4.....15

FIGURA 5.....16

FIGURA 6.....16

FIGURA 7.....16

FIGURA 8

FIGURA 9

FIGURA 10

FIGURA 11

FIGURA 12

FIGURA 13

FIGURA 14

LISTA DE TABELAS

TABELA 1.....9

TABELA 2.....10

TABELA 3.....10

TABELA 4.....13

TABELA 5.....14

## **OBJETIVO**

Utilizando o conhecimento adquirido em aula e extra classe, construir um circuito digital simulador do jogo "Batalha Naval", onde a partir das regras do jogo, dê origem a um possível vencedor.

## INTRODUÇÃO

Este relatório, por meio de exemplos, demonstra o processo da construção de um circuito lógico baseado no jogo Batalha Naval, utilizando ci's como interpretador das entradas, aplicando as regras do jogo e por meio da simplificação das expressões algébricas e o mapa de Karnaugh, determinar as saídas e posteriormente os circuitos. Após esta etapa, será realizado os testes no programa Logisim, que demonstrará se o mesmo apresenta-se correto e então implementado na protoboard.

## MATERIAIS

Para a construção do circuito foi necessário os seguintes itens:

Fios de cobre;

1 Breadboard;

1 LED;

1 Resistor 1k;

6 Pushbuttons;

Circuitos Integrados utilizados:

1 74LS04 - Portas NOT;

1 74LS08 - Portas AND;

1 74LS32 - Portas OR;

1 74LS86 - Portas XOR;

Fonte de alimentação simétrica digital SKILL-TEC 0-32V (SKFA-03D) para alimentação do circuito;

Logisim para ilustração e simulação do circuito digital;

Autodesk circuits para a simulação da breadboard e do circuito pré-implantado.



## METODOLOGIA

### CODIFICAÇÃO DE ENTRADAS E SAIDAS

Baseado nas regras do jogo, junto ao conhecimento sobre a construção de circuitos lógicos, foi definido que cada jogador entraria com 3 valores binários, dando assim origem a cada rodada. Sendo nomeados como J1 e J2, cada jogador terá de entrar com o valor binário referente a sua jogada seguindo a tabela abaixo:

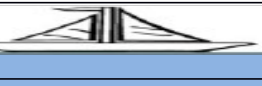






Entrada	Jogada Equivalente		
000			
010			
011			
101			
110			
111			

Tabela 1. Codificação das entradas do circuito.

Esta codificação foi escolhida, devido a testes realizados com outras possíveis entradas e encontrando nesta, a possibilidade de uma maior simplificação do circuito, ocasionando um menor custo para o circuito.

Através destas entradas, acontece o processamento no circuito lógico, dando origem a duas possíveis saídas (0 ou 1), informando se o jogador J2, acertou ou não a jogada.

## ELABORAÇÃO DA TABELA VERDADE

Para a construção do circuito do primeiro módulo, foi montada a tabela verdade e a partir da codificação das entradas, criou-se três saídas, nomeadas como **S1**, **S2** e **S3** a partir dos maxitermos. Assim como ilustrado na tabela verdade abaixo:

Entrada			Saídas		
A	B	C	S1	S2	S3
0	0	0	0	0	1
0	0	1	X	X	X
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	X	X	X
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	0	1

Tabela 2: Tabela verdade com as saídas codificadas

O segundo módulo trata-se de um circuito comparador, das saídas do primeiro com as 3 entradas diretamente no circuito, resultando nas possíveis saídas:

Saídas do primeiro módulo			Entradas no segundo módulo			São Iguais?	
S1	S2	S3	A2	B2	C2	Sim	Não
0	0	1	0	0	1	1	0
0	1	0	0	1	0	1	0
1	0	0	1	0	0	1	0
0	1	1	0	1	1	1	0
1	1	0	1	1	0	1	0
1	0	1	1	0	1	1	0

Tabela 3: Tabela verdade com as possíveis jogadas do jogador J2 e resultados

## CONSTRUÇÃO DO CIRCUITO LÓGICO NO LOGISIM A PARTIR DA TABELA VERDADE

O Logisim é uma ferramenta educacional utilizada para a concepção e a simulação digital de circuitos lógicos e por meio desta, utilizando a tabela verdade, construímos o seguinte circuito:

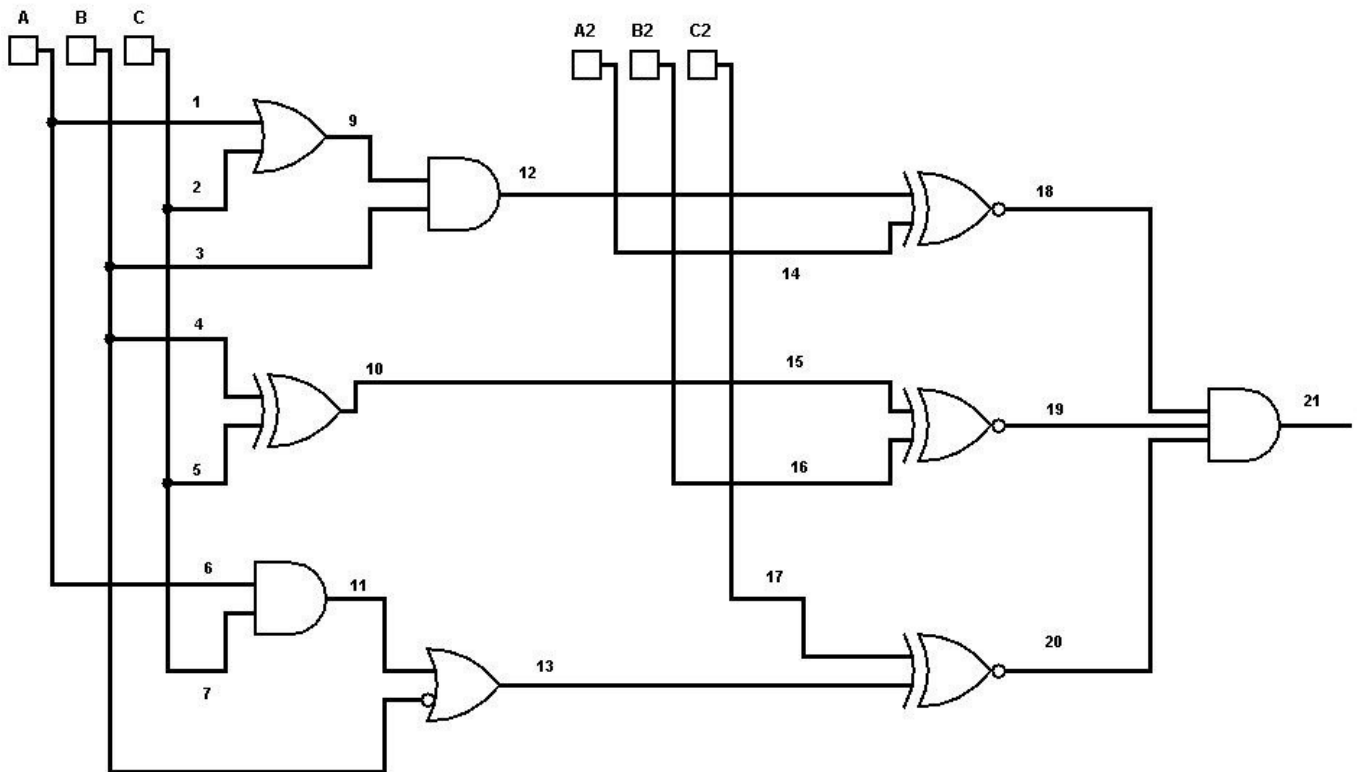


Figura 1: Circuito lógico completo obtido a partir da tabela verdade

O primeiro módulo consiste no seguinte circuito:

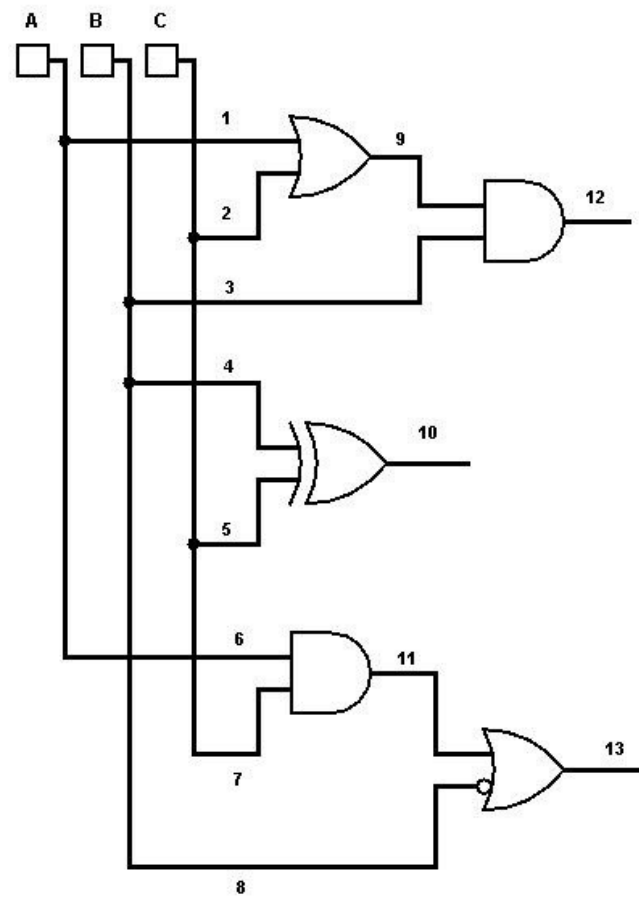


Figura 2: Circuito lógico do primeiro módulo

Cada número neste circuito representa uma expressão algébrica ou variável, assim como demonstrado na tabela a seguir:

Número	Valor algébrico
1	A
2	C
3	B
4	B
5	C
6	A
7	C
8	B!
9	$A + C$
10	$B \oplus C$
11	$A \cdot C$
12	$(A + C) \cdot B$
13	$B! \cdot (A \cdot C)$

Tabela 4: Expressões algébricas referente a cada entrada/saída do módulo 1

O segundo módulo foi criado com a ideia de realizar uma comparação entre as 3 saídas do primeiro módulo e as 3 entradas representadas por A2, B2 e C2, assim como representado na figura abaixo:

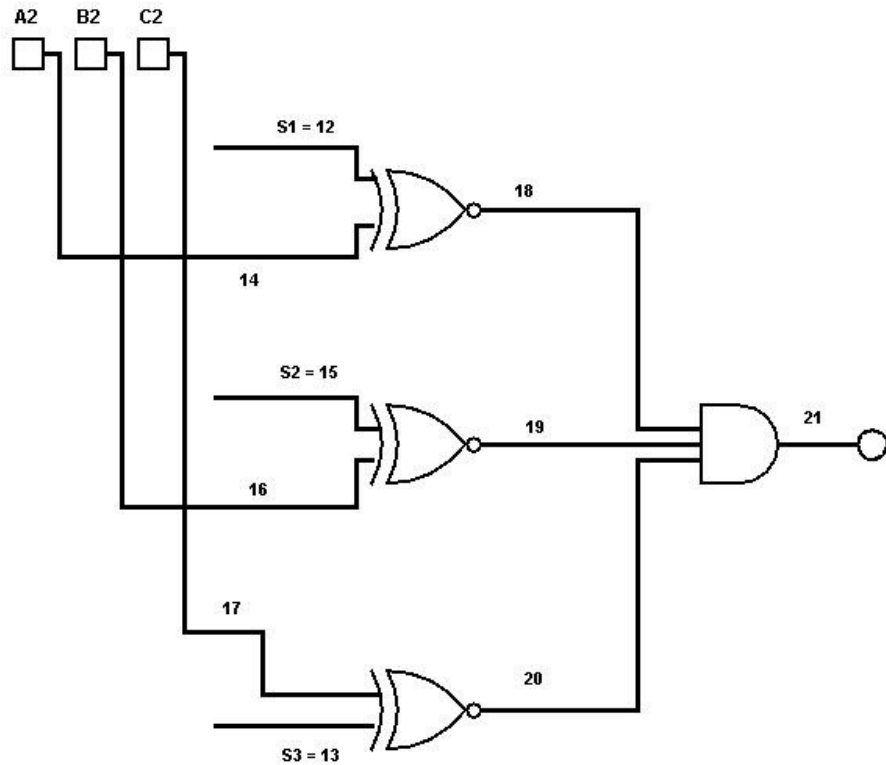


Figura 3: Circuito lógico do segundo módulo

Assim como no primeiro circuito, cada um destes números representam uma expressão algébrica, de acordo com a tabela a seguir:

Número	Valor algébrico
12	S1
13	S3
14	A2
15	S2
16	B2
17	C2
18	$(S1 \oplus A2)!$
19	$(S2 \oplus B2)!$
20	$(S3 \oplus C2)!$
21	$(S1 \oplus A2)! \cdot (S2 \oplus B2)! \cdot (S3 \oplus C2)!$

Tabela 5: Expressões algébricas referente a cada entrada/saída do módulo 2

A partir da tabela verdade e da construção desse primeiro circuito, obtivemos as seguintes expressões algébricas de acordo com os maxitermos:

- $S1 = (A+B+C).(A+B'+C).(A'+B+C')$
- $S2 = (A+B+C).(A+B'+C').(A'+B'+C')$
- $S3 = (A+B'+C).(A+B'+C').(A'+B'+C)$

O segundo circuito gerou de origem a seguinte expressão:

- $(S1 \oplus A2)! \cdot (S2 \oplus B2)! \cdot (S3 \oplus C2)!$

## SIMPLIFICAÇÃO DAS EXPRESSÕES ALGÉBRICAS

Para realizar a simplificação das expressões algébricas, foi utilizado o mapa de Karnaugh, um método gráfico e eficiente, facilitando o processo de simplificação e diminuindo o uso de circuitos integrados.

Foram construídos 3 mapas de Karnaugh, um para cada saída codificada, assim como nas imagens abaixo:

MAPA DE KARNAUGH				
S1	BC	BC'	B'C'	B'C
A	0	X	1	0
A'	X	0	1	1

S2	BC	BC'	B'C'	B'C
A	0	X	0	1
A'	X	1	0	1

S3	BC	BC'	B'C'	B'C
A	1	X	0	0
A'	X	1	1	0

Figura 4: Mapa de Karnaugh das saídas S1, S2 e S3.

Esses três mapas resultaram nas seguintes simplificações:

	Saídas simplificadas
S1	$B.(A + C)$
S2	$(B+C).(B'+C')$
S3	$B' + A.C$

Figura 5: Saídas simplificadas

Afim de minimizar o máximo possível o circuito, concluímos que ainda sim existia possibilidade de simplificação, a saída S2 consistem em:  $(B+C).(B'+C')$ , porém, se aplicarmos o teorema de De Morgan teremos a seguinte expressão:  $(B'.C')+(B.C)$ , se olhar pelo postulado de equivalência, percebemos que essa expressão é equivalente a  $(B \oplus C)!$ , logo aplicaremos novamente o teorema de De Morgan obtendo a expressão:  $(B \oplus C)$ , as expressões finais:

	Saídas simplificadas
S1	$B.(A + C)$
S2	$B \oplus C$
S3	$B' + A.C$

Figura 6: Saídas simplificadas após o teorema de De Morgan

O circuito final, realizado com a junção dos dois módulos, foi construído a partir da expressão algébrica gerada de acordo com a junção destas saídas simplificadas e as expressões algébricas referentes ao módulo dois, resultando na seguinte expressão:

Expressão Algébrica do Circuito
$\overline{((B.(A + C)) \oplus A2)} . \overline{((B \oplus C) \oplus B2)} . \overline{((B'+A.C) \oplus C2)}$

Figura 7: Expressão algébrica do jogo Batalha Naval



## UTILIZANDO NETLIST PARA IMPLANTAÇÃO NA BREADBORD

A Netlist é um arquivo gerado por um programa simulador de circuitos lógicos, onde contém as informações das situações de entrada e saída dos circuitos integrados. Abaixo é demonstrado uma sequência de tabelas onde foi registrado cada entrada e saída de seus respectivos circuitos integrados a fim de auxiliar na implantação da breadbord:

Netlist		
CI - 74LS04 - NOT		
Porta	Entrada/Saída	Valor
1	Entrada	$(B.(A+C)) \oplus A2$
2	Saída	$\sim((B.(A+C)) \oplus A2)$
3	Entrada	
4	Saída	
5	Entrada	
6	Saída	
7	GND	GND
8	Saída	$\sim((B \oplus C) \oplus B2)$
9	Entrada	$(B \oplus C) \oplus B2$
10	Saída	$\sim((B'.(A.C)) \oplus C2)$
11	Entrada	$(B'.(A.C)) \oplus C2$
12	Saída	$B'$
13	Entrada	$B$
14	VCC	VCC

Figura 8: Netlist referente ao CI - 74LS04

Netlist		
CI - 74LS08 - AND		
Porta	Entrada/Saida	Valor
1	Entrada	$(A+C)$
2	Entrada	B
3	Saida	$B.(A+C)$
4	Entrada	$\sim((B.(A+C)) \oplus A2)$
5	Entrada	$\sim((B \oplus C) \oplus B2). \sim((B'.(A.C)) \oplus C2)$
6	Saida	$\sim((B \oplus C) \oplus B2). \sim((B'.(A.C)) \oplus C2). \sim((B.(A+C)) \oplus A2)$
7	GND	GND
8	Saida	$\sim((B \oplus C) \oplus B2). \sim((B'.(A.C)) \oplus C2)$
9	Entrada	$\sim((B \oplus C) \oplus B2)$
10	Entrada	$\sim((B'.(A.C)) \oplus C2)$
11	Saida	$(A.C)$
12	Entrada	C
13	Entrada	A
14	VCC	VCC

Figura 9: Netlist referente ao CI - 74LS08

Netlist		
CI - 74LS32 - OR		
Porta	Entrada/Saida	Valor
1	Entrada	A
2	Entrada	C
3	Saida	$(A+C)$
4	Entrada	
5	Entrada	
6	Saida	
7	GND	GND
8	Saida	
9	Entrada	
10	Entrada	
11	Saida	$B'+(A.C)$
12	Entrada	$B'$
13	Entrada	$(A.C)$
14	VCC	VCC

Figura 10: Netlist referente ao CI - 74LS32

Netlist		
CI - 74LS86 - XOR		
Porta	Entrada/Saida	Valor
1	Entrada	$B.(A+C)$
2	Entrada	A2
3	Saida	$(B.(A+C)) \oplus A2$
4	Entrada	B
5	Entrada	C
6	Saida	$B \oplus C$
7	GND	GND
8	Saida	$(B \oplus C) \oplus B2$
9	Entrada	B2
10	Entrada	$B \oplus C$
11	Saida	$(B' + (A.C)) \oplus C2$
12	Entrada	C2
13	Entrada	$B' + (A.C)$
14	VCC	VCC

Figura 11: Netlist

referente ao CI - 74LS86

## CÁLCULO DO CAMINHO CRÍTICO

O caminho crítico do circuito passa por:

1x CI 74LS04, cujo  $T_{pHL}$  é igual a 15

1x CI 74LS08, cujo  $T_{pHL}$  é igual a 18

1x CI 74LS32, cujo  $T_{pHL}$  é igual a 15

1x CI 74LS86, cujo  $T_{pHL}$  é igual a 23

Assim, a fórmula para o tempo de atraso do circuito pode ser definida por:

$$T = (15 * 2) + 18 + 23$$

Logo,

$$T = 71\text{ns.}$$

Para calcularmos a frequência, utilizamos a fórmula:

$$f = 1 / 2T$$

$$2 * 71 = 142\text{ns.}$$

Para o cálculo de frequência, o tempo deve ser dado em segundos, assim:

$$142\text{ns} = 142 * 10^{-7} = 0,000000142\text{s}$$

Então,

$$f = 1 / 0,000000142 \text{ é aproximadamente } 7042253,53 \text{ Hz} = 7,04225353 \text{ MHz}$$

### UTILIZAÇÃO DO SOFTWARE PARA SIMULAÇÃO DA BREADBORD

Utilizando o web app Autodesk Circuits e a Netlist previamente apresentada, simulamos a implantação na breadbord a fim de nos orientar sobre como poderia ser implantado na breadbord, abaixo o resultado de como ficou a implantação no software:

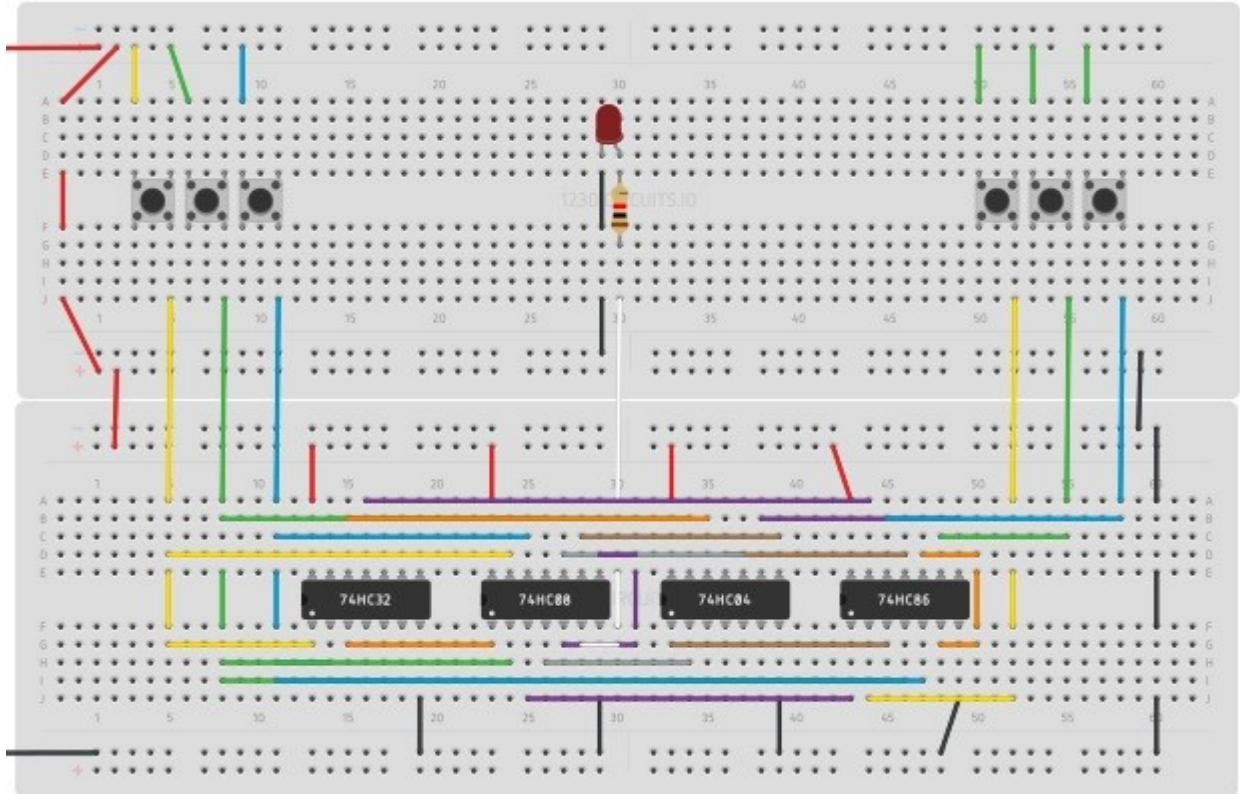


Figura 12: Breadbord simulada no Autodesk Circuits

Para termos uma melhor organização foi reservado as utilizando uma breadboard onde a parte de baixo acontece o processamento das entradas e a parte de cima ocorre as entradas (via botões) e é demonstrado se o jogador J2 acertou ou não sua jogada.

A fim de evitarmos também fios cruzados na diagonal, foram reservados também 6 colunas referentes as entradas do jogador J1 e três para o jogador J2 assim como nas duas figuras a seguir:

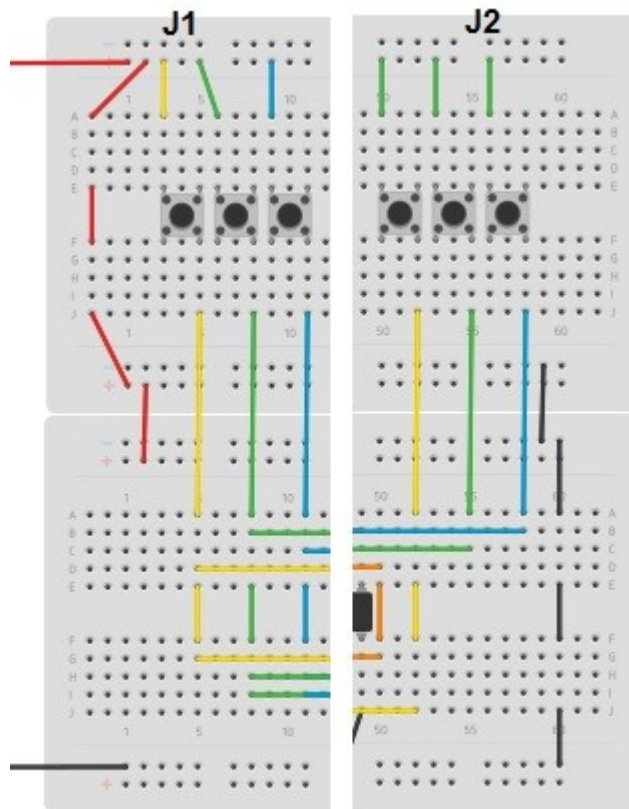


Figura 13: Caminhos reservados para as 6 entradas

Legenda de cores:

■ VCC

■ GND

■ Entrada A e A2 (A2 = J2)

■ Entrada B e B2 (B2 = J2)

■ Entrada C e C2 (C2 = J2)

## RESULTADO

As expressões algébricas unificadas e simplificadas foram adaptadas de acordo com os ci's e apresentam-se da seguinte forma:

$$\overline{((B.(A + C)) \oplus A2)} . \overline{((B \oplus C) \oplus B2)} . \overline{((B' + A.C) \oplus C2)}$$

Após a implantação na breadboard foi colocada a led para demonstração do resultado do jogo assim como demonstrado na imagem abaixo:

## DISCUSSÃO

Com o principal objetivo de se desenvolver um trabalho de boa qualidade, desde o principio foi discutido qual abordagem era realmente necessária para chegar o resultado esperado, desde qual codificação utilizar para as entradas e como ficariam as saídas, as modificações das portas e qual programa para realizar os testes. Os pontos de maior atenção foram:

- Uma forma de se registrar cada entrada e saída dos circuitos integrados, a fim de que houvesse uma forma de não se perder ao operar na breadboard. Dentre os meios que pensamos, percebemos que a netlist é a melhor forma para se registrar o mesmo.
- É de extrema importância que houve uma forma de que os fios não se cruzassem, então foi pensado a questão de reservar as primeiras colunas da breadboard para cada entrada, distanciando e deixando evidente cada entrada.
- Optamos por utilizar a breadboard ao invés da protoboard por conta de um maior espaço e uma melhor organização. Também foi utilizado um arduíno mega como alimentação para testes em casa.

## CONCLUSÃO

O circuito lógico demonstrou os resultados de acordo com o esperado, sendo capaz de reproduzir o proposto inicialmente. Não houve necessidade de se fazer qualquer mudança significativa na estrutura do trabalho. A utilização de botões proporcionou uma melhor entrada das tensões, evitando a troca de fios e eventuais problemas com a fiação.

Conclui-se também que a dimensão de expansão deste circuito é imensurável, como por exemplo, utilizar de outras peças e equipamentos para tratar a saída de forma que não interferisse o conceito em si, podendo desde utilizar speakers até um servidor de resposta para que recebesse a mensagem do vencedor. Estes tipos de implementações podem sempre acarretar um aumento de conhecimento de todos os integrantes do grupo.



## REFERÊNCIAS

1. **Autodesk Circuits** - <<https://circuits.io/>> Acessado todos os dias durante a construção deste trabalho.
2. **Futurlec** - <<http://www.futurlec.com/>> Acessado todos os dias durante a construção deste trabalho.