

# Experimento 1

## Portas Lógicas AND, OR e NOT

Matheus Cardoso de Souza, 202033507

Ualiton Ventura da Silva, 202033580

Grupo G42

<sup>1</sup>Dep. Ciência da Computação – Universidade de Brasília (UnB)  
CIC0231 - Laboratório de Circuitos Lógicos

matheus-cardoso.mc@aluno.unb.br, 202033580@aluno.unb.br

**Abstract.** *This report corresponds to the Experiment 1 on “Logical Gates NAND, NOR and XOR”.*

**Resumo.** *O presente relatório corresponde ao Experimento 1 sobre “Portas Lógicas NAND, NOR e XOR”.*

### 1. Introdução

Neste experimento temos como objetivo abordar conceitos referentes ao uso de portas lógicas que conceitualmente derivam de portas primitivas como **AND**, **OR** e **NOT**.

#### 1.1. Objetivos

Os experimentos realizados visam abordar assuntos como a universalidade de portas, assim como o comportamento ocorrido no encadeamento de portas lógicas. Também tem como objetivo descrever como circuitos lógicos **XOR** de N entradas se comportam de maneira generalizada.

#### 1.2. Materiais

Em função da natureza do ensino a distância, os presentes experimentos não foram realizados usando-se materiais e equipamentos físicos, mas sim emulados por meio da simulador online Tinkercad.

A seguir estão enumerados os materiais simulados:

- Painel Digital
- *Protoboard*
- Fios
- Seletores de estado lógico
- LEDs
- Resistores
- Multímetros
- Portas Lógicas **NAND**, **NOR** e **XOR**

### 2. Procedimentos

Passaremos a apresentar os experimentos requeridos.

## 2.1. Implementação de uma porta NAND de 3 entradas

O atual experimento visa implementar uma porta **NAND** com três entradas com o uso apenas de portas **NAND**, sendo também requisitado a medição dos valores lógicos em pontos específicos do circuito.

Abaixo sua implementação:

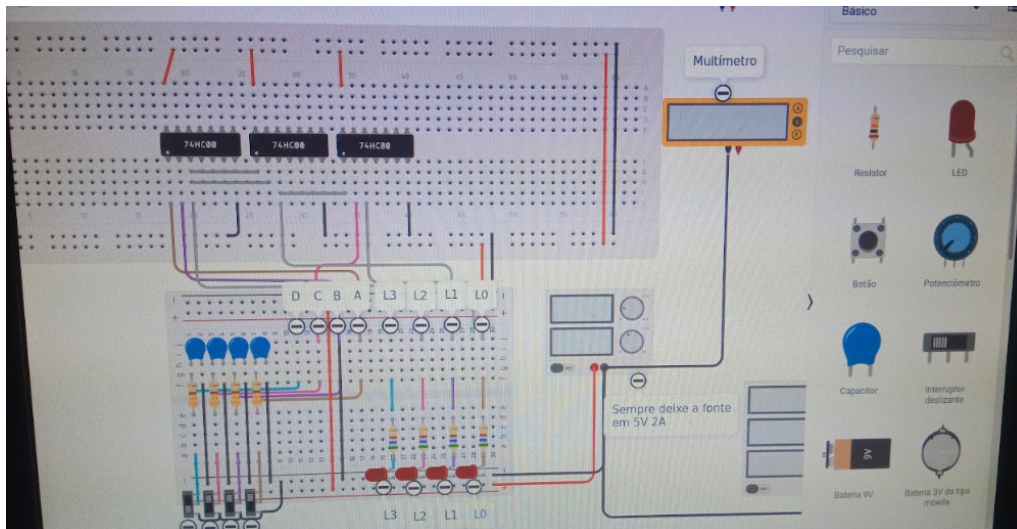


Figura 1. Circuito NAND de 3 entradas

Para conferir o vídeo deste experimento, acesse o seguinte link:  
<https://youtu.be/lwS4AKuvgps>.

Abaixo temos a tabela dos valores lógicos obtidos ao realizar o experimento:

Tabela 1. Tabela para a porta lógica NAND 3 entradas

Entradas			Saídas		
C	B	A	$L3 = \overline{A.B}$	$L2 = \overline{A.B}$	$L1 = \overline{A.B.C}$
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	0	1	0

É possível concluir através da tabela e vídeo que de fato descreveu uma porta **NAND** de três entradas, sendo esta equivalente a  $\overline{A.B.C}$ .

## 2.2. Implementação da função XOR usando portas NAND.

Sendo portas **NAND** universais, são capazes de descrever não somente **NAND** de três entradas como qualquer outro tipo, assim, este experimento realiza a implementação de uma porta **XOR** utilizando **NAND**.

Abaixo a implementação do circuito:

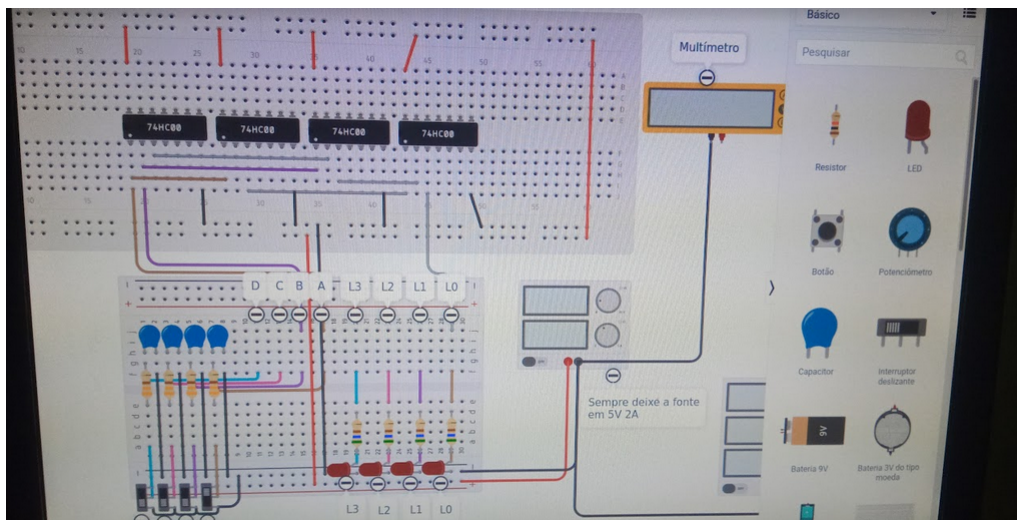


Figura 2. Circuito XOR com NAND

Para conferir o vídeo deste experimento, acesse o seguinte link:  
<https://youtu.be/QhaJeFVEhhw>.

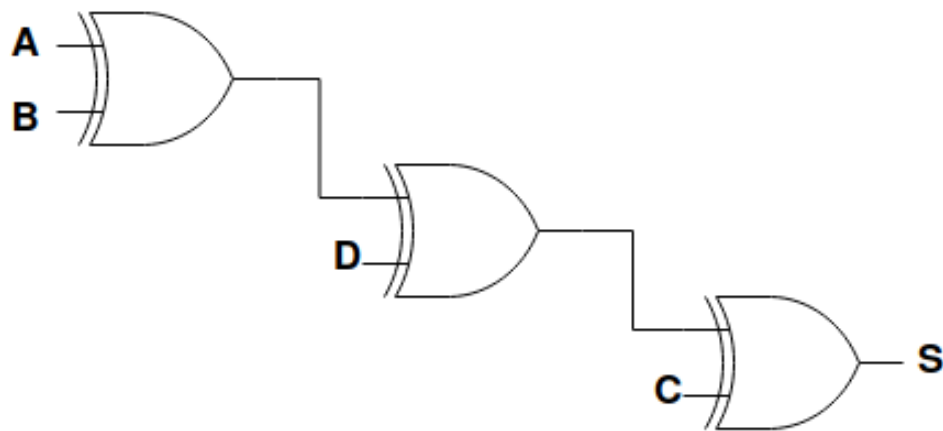
Tabela 2. Tabela para a porta lógica AND

Entradas		Saídas
B	A	$L0 = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Portanto, verifica-se através tanto do vídeo quanto tabela que de fato foi possível descrever a porta lógica **XOR** através de portas **NAND**.

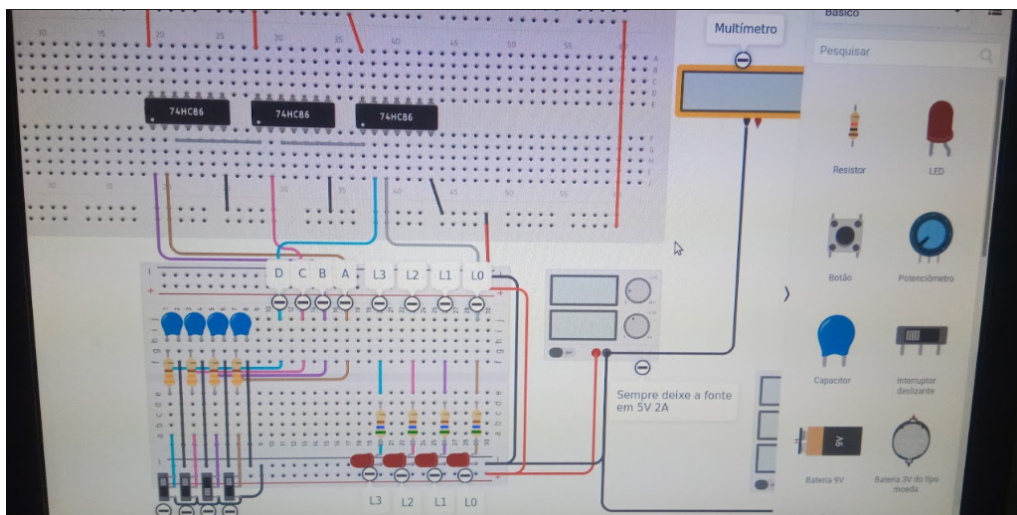
## 2.3. Implementação de uma porta XOR de 4 entradas usando portas XOR de 2 entradas

O atual experimento realiza a implementação de uma porta **XOR** de quatro entradas com o uso de portas **XOR** de duas entradas, assim, esquematicamente temos que poderá ser descrito como:



**Figura 3. Esquema de uma porta XOR de 4 entradas**

Expressando através do Thinkercad e com o uso do circuito 74HC86 teremos:



**Figura 4. Circuito XOR de 4 entradas**

Para conferir o vídeo deste experimento, acesse o seguinte link:  
<https://youtu.be/CHNdltWfVFs>.  
 Assim, sua tabela verdade será:

**Tabela 3. Tabela para a porta lógica XOR 4 Entradas**

Entradas				Saídas
D	C	B	A	L0
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Através da tabela e vídeo apresentado é possível notar que uma porta **XOR** de 4 entradas possui valor lógico 1 quando o número de bits é ímpar, caso contrário será 0, este comportamento não é constante, mas de maneira geral tem-se que para uma porta **XOR** de N entradas, se N é par, então quando o seu número de bits de entrada que assumem valor lógico 1 for ímpar, seu resultado será 1, contrariamente temos que se N é ímpar, então quando o número de bits de entrada assumir valor lógico 1 for par, assim teremos valor lógico 1.

### 3. Análise dos Resultados

Através dos experimentos realizados foi possível comprovar a universalidade de portas **NAND** e também como portas **XOR** de N entradas possuem um regra geral que é capaz de descrever suas saídas.

### 4. Conclusão

Utilizando conceitos referentes a universalidade de portas temos uma ferramenta que possibilita maior versatilidade no dia a dia, a princípio poderia parecer ilógico do porquê é útil este conceito, contudo, deve-se notar que nem sempre possuímos todas as portas desejadas disponíveis, assim, ter recursos alternativos para lidar com situações como estas é importante. Referente ao uso de portas **XOR** temos que seus conceitos de paridade são muito propícios em análise de bits, exemplo a ser citado é através da transmissão de dados, e um exemplo de código que utiliza de tal característica é o código de Hamming, que através da análise de paridade de bits é capaz de realizar a detecção de erros em transmissão de dados, criando então uma camada de segurança.

## **Referências**

- [Harris 2012] Harris, D. (2012). *Digital Design and Computer Architecture*. Morgan Kaufmann, 2nd edition.
- [Lamar 2021] Lamar, M. V. (2021). Laboratório de circuitos lógicos.
- [Mandelli 2021] Mandelli, M. (2021). Circuitos lógicos.
- [Patt 2003] Patt, Y. N. (2003). *Introduction to Computing Systems: From Bits and Gates to C and Beyond*. McGraw-Hill Education, 2nd edition.

**Auto-Avaliação**

Respostas:

A	B
1	c
2	a
3	d
4	c
5	b