**Jaqueline, eu e o henrique terminamos sábado de manhã e eu vou mandar relatório quando estiver aberto lá! Até sexta. Bjs**

**Circuitos Digitais - 2º Experimento - Portas Lógicas: NAND, NOR e XOR**

**Data de Realização: 27/03/2015**

**Camila Imbuzeiro Camargo (13/0104868)**

**Henrique Medrado de Faria (12/0032121)**

**Jaqueline Couto Moreira (10/0088040)**

**Objetivo**

Analisar experimentalmente as portas lógicas NAND, NOR e XOR, mediante o estudo de suas respectivas tabelas da verdade e equivalências lógicas. Além disso, pretende-se verificar o caráter universal das portas NAND e NOR. Por fim, são observados e discutidos o teorema de De Morgan e os conceitos de *fan-in* e *fan-out*.

**Materiais**

• Painel digital;

• Protoboard;

• Ponta lógica;

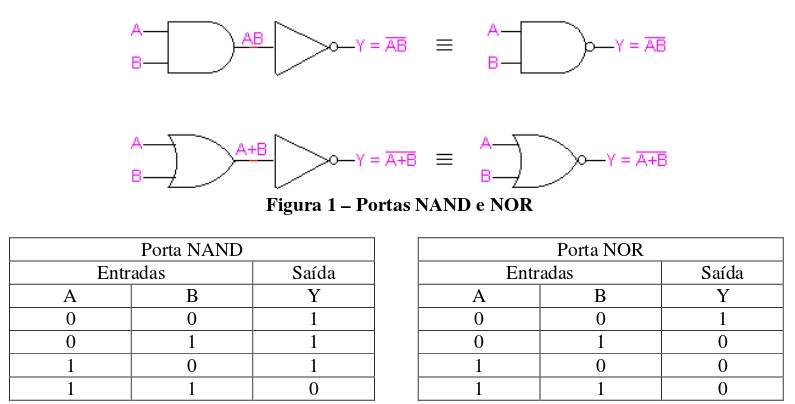
• Fios conectores;

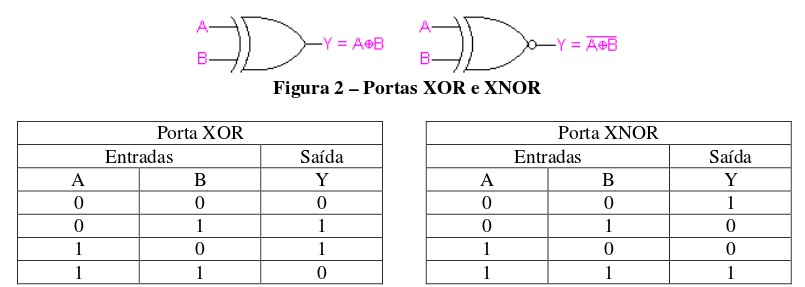
• Portas NAND e XOR.

**Introdução**

**1.1. Portas NAND, NOR e XOR:**

Quando implementamos circuitos, existem certos conjuntos de portas que são universais, ou seja, eles são capazes de representar qualquer expressão lógica sozinhos. As portas NAND e as portas NOR são dois exemplos de conjuntos de portas universais. A porta NAND representa a negação da expressão lógica AND, e portanto, ela tem a tabela verdade com saídas inversas à da tabela verdade da expressão AND. O mesmo ocorre para a porta lógica NOR, que é a negação da expressão lógica OR.



Além das portas NAND e NOR, existem outras portas que, apesar de não serem universais tem aplicação muito útil: as portas XOR e XNOR. A primeira porta é conhecida como OU-exclusivo. Ela compara dois bits e a saída será 1 se e somente se os bits forem diferentes. Se houver mais de duas entradas, aí será 1 se houver um numero ímpar de valores verdadeiros com entrada. Já a porta XNOR, que produz a tabela verdae complementar da porta XOR, irá ter saída 1 quando as duas entradas forem iguais, ou, para o caso de mais de duas entradas, quando o número de entradas 1 for par. 

A expressão dasportas XOR e XNOR podem ser escritas em termos das portas AND, OR e NOT:

A⊕B = (NOT A . B) + (NOT B . A)

NOT(A ⊕ B) = (AB) + (NOT A . NOT B)

**1.2. Fatores de Carga (*fan-in*, *fan-out*):**

O tempo de chaveamento de uma porta lógica depende do número de portas alimentadas pela saída. O **Fan-out** de uma porta é o número de portas que pode ser alimentado na saída e depende de como a porta é utilizada na sequência lógica. Ele representa o número máximo de entradas lógicas que uma saída pode acionar com segurança. Se o valor estabelecido pelo FAN-OUT for excedido, a tensão de nível lógico de saída não poderá ser mais garantida. Este conceito se aplica quando ocorre o consumo de energia das portas ligadas na saída. Seu valor depende da tecnologia empregada:

• TTL: 2 a 10

• CMOS: 50 a 100

Além disso, o termo **Fan-in** é utilizado para representar o número máximo de entradas que uma porta lógica possui. Para a para a série TTL 74XX, utilizada nos experimentos da matéria, tem-se:

1 unidade de carga TTL = 40 µA, no nível lógico 1.

= 1,6 mA, no nível lógico 0.

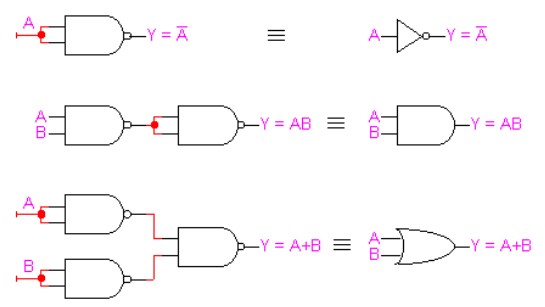
Em outras palavras, uma porta 7400 que necessite de uma corrente de entrada máxima de IIL = 1,6 mA para o nível lógico 0 e uma corrente de entrada máxima IIH = 40 µA para o nível lógico 1 é especificada como tendo um fator de carga unitário. Isto é, possui um fan-in de 1. Por outro lado, a saída de uma porta 7400 absorverá 16 mA no nível lógico 0 e fornecerá 800 µA no nível lógico 1. Portanto, ela tem capacidade de acionar 10 portas no nível lógico 0 (pois 16 mA / 1,6 mA = 10). Isto é, possui um fan-out de 10 para o nível lógico 0. Da mesma forma, o fan-out para o nível lógico 1 é 800 µA / 40 µA = 20.

**1.3. Teorema de De Morgan:**

Um teorema muito útil e que comprova a universalidade de certos conjuntos de portas é o teorema de DeMorgan. Este teorema estabelece que:

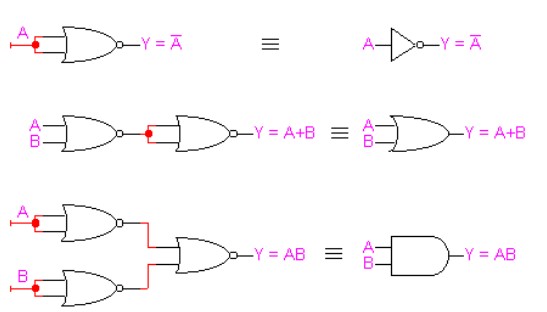


Isso, em outras palavras, quer dizer que, para retirar a negação de uma expressão devemos negar as partes e intercambiar a expressão entre elas. A partir disso é possível concluir o porque da universalidade das portas NAND e NOR. Nessas portas temos uma expressão negada, ou seja, é possivel criar variáveis negadas e obter a expressão lógica não originalmente testada pela porta. Para as portas NAND temos a seguinte relação:



**Figura 3 - Universalidade da porta NAND**

Já para a porta NOR, temos:

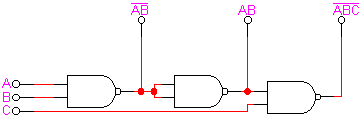


**Figura 4 - Universalidade da porta NOR**

Neste experimento iremos provar que esta universalidade é de fato verdadeira, e iremos explorar o funcionamento das portas NAND, NOR e XOR.

**Procedimentos**

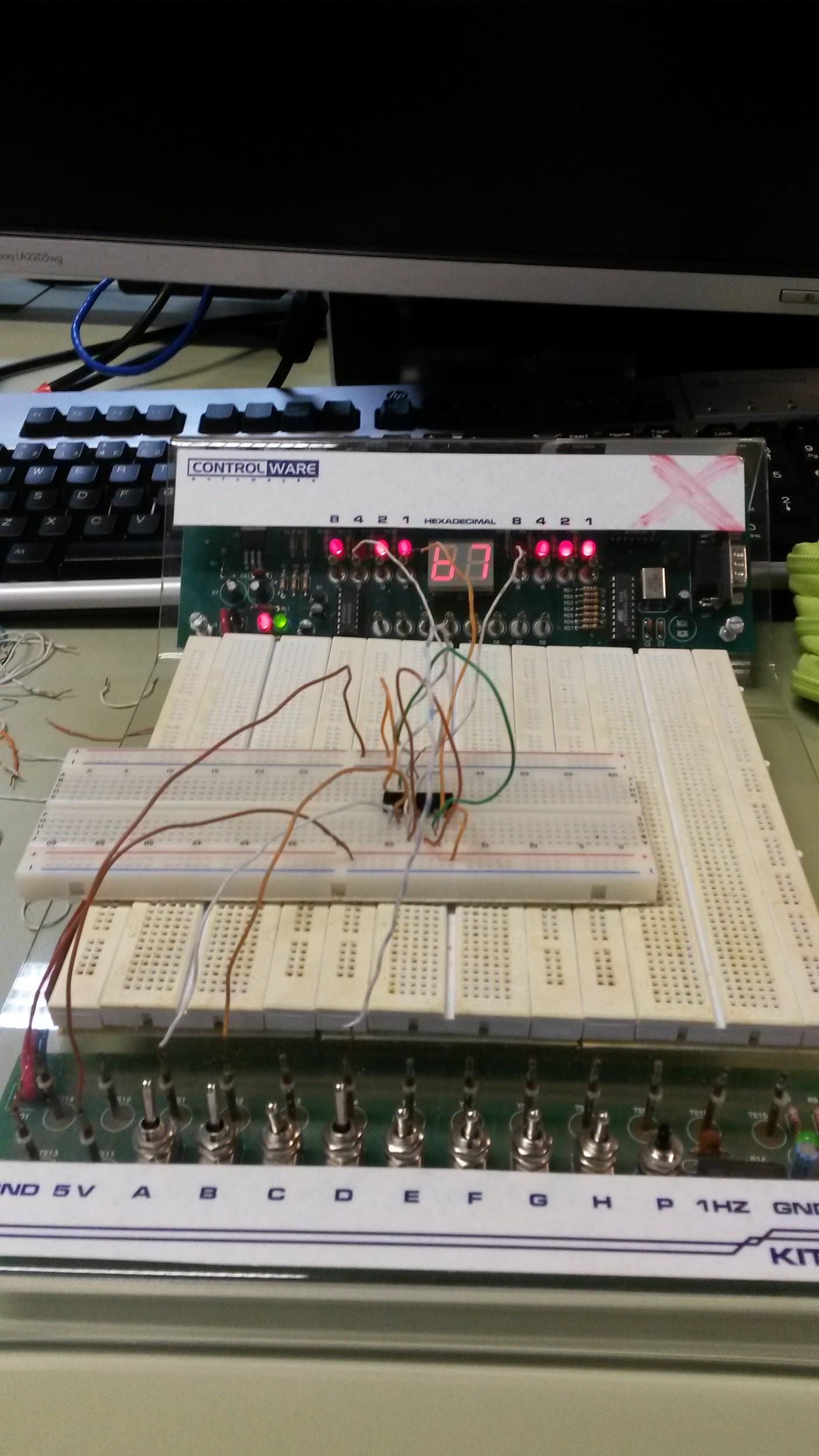
**2.1.** Na primeira parte do experimento, foi implementado uma porta NAND de três entradas, utilizando-se três portas NAND de duas entradas:



**Figura 5 : Esquema da porta NAND de três entradas**

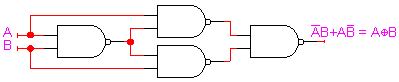
Após a devida implementação do circuito, foi preenchida a tabela da verdade para as saídas

Captura de Tela 2015-03-27 às 11.12.47.png, **A.B** e Captura de Tela 2015-03-27 às 11.13.52.png. A imagem abaixo apresenta o circuito montado:



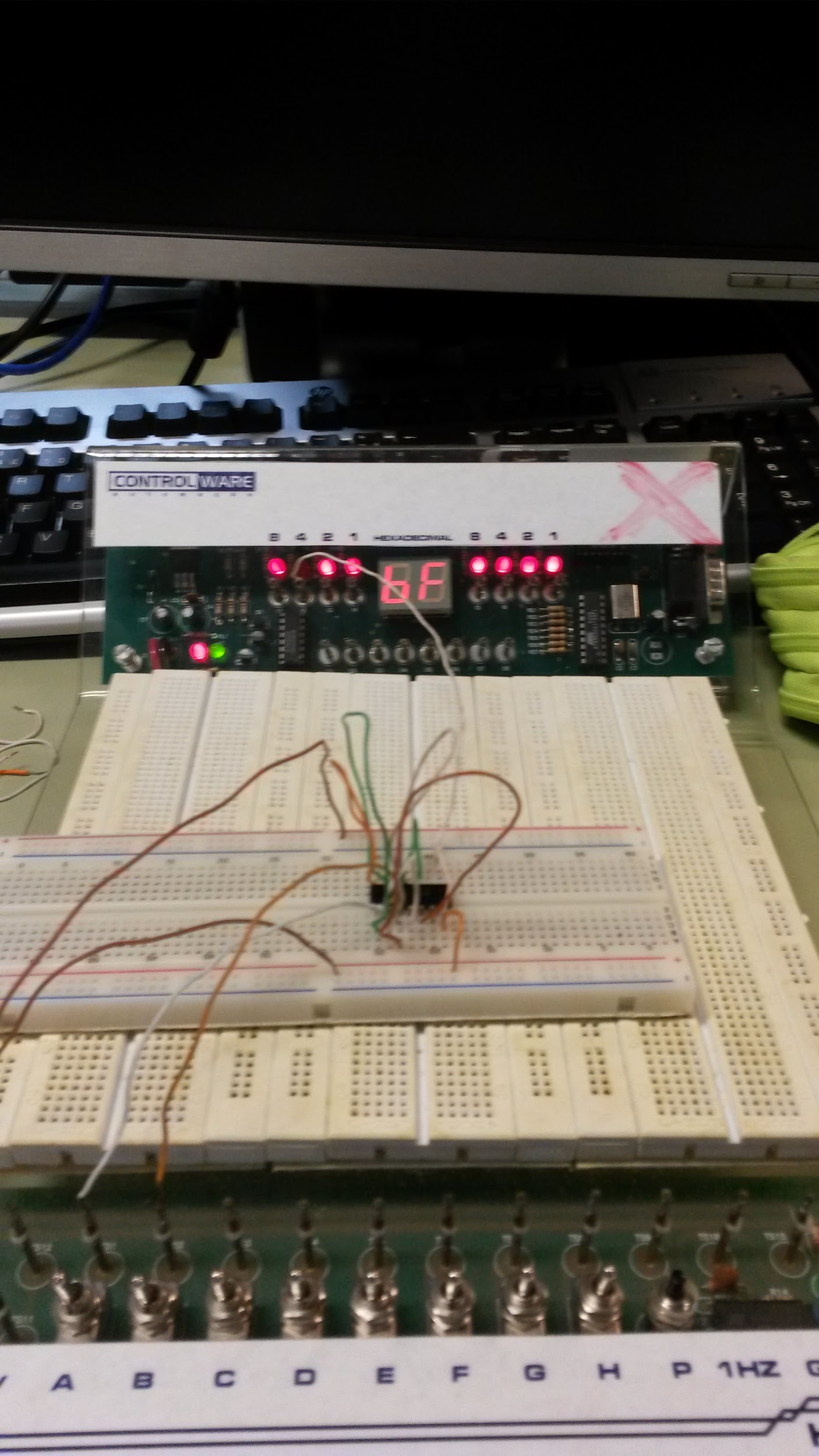
**Figura 6 : Porta NAND de três entradas**

**2.2.** Na segunda parte do experimento, implementou-se uma função XOR mediante a utilização de quatro portas NAND:



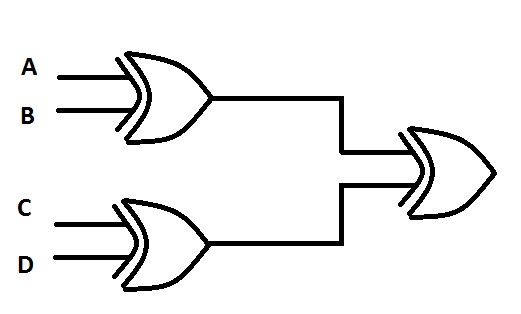
**Figura 7: Esquema da função XOR com portas NAND.**

Após a devida implementação do circuito, foi preenchida a tabela verdade correspondente à função XOR.



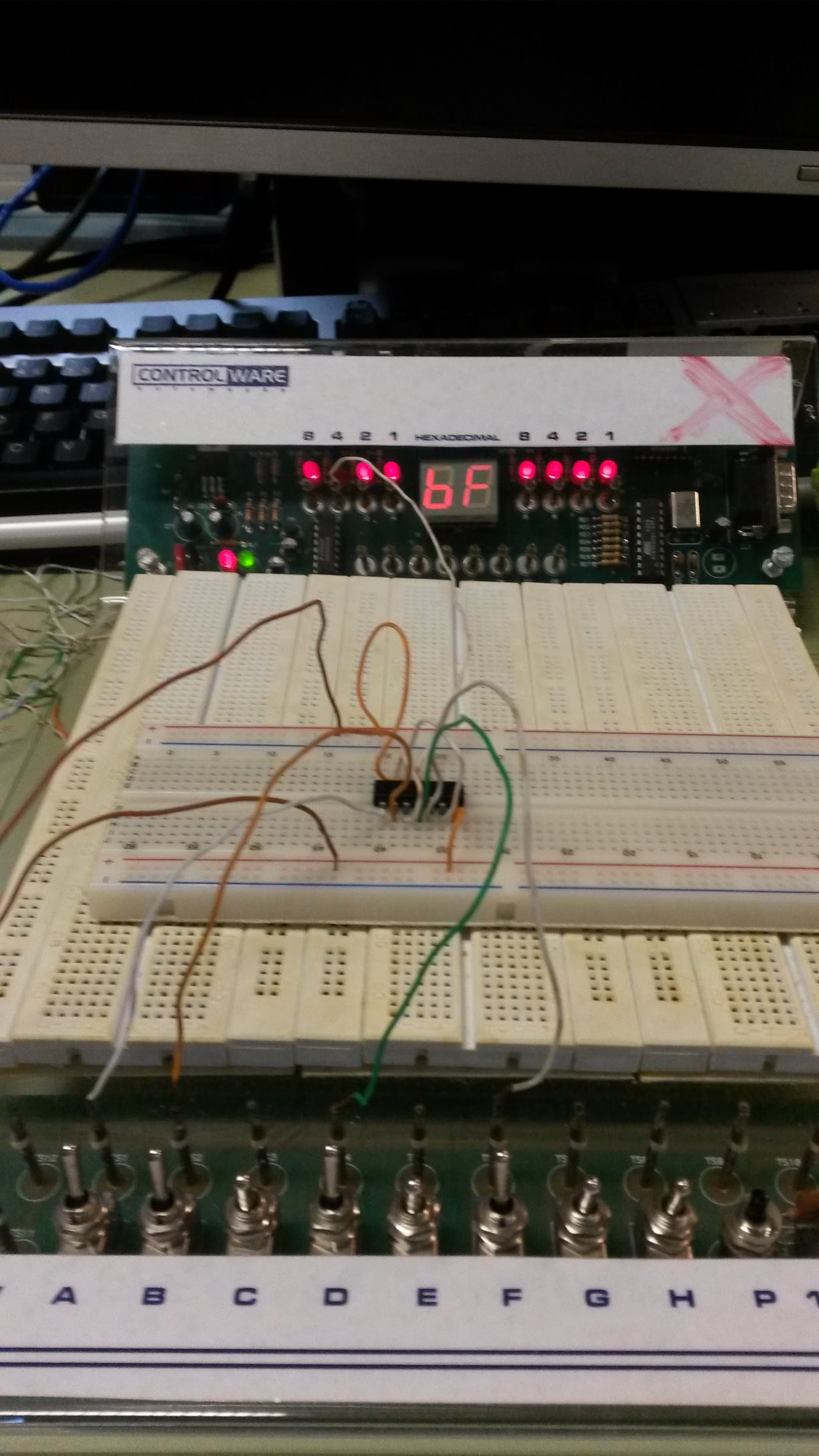
**Figura 8: Função XOR com portas NAND.**

**2.4.** Na última parte do experimento, uma porta XOR de quatro entradas foi projetada e implementada a partir da utilização de três portas XOR de duas entradas.



**Figura X: Função XOR de quatro entradas.**

Após a devida projeção e implementação do circuito, foram verificados os casos nos quais a saída do circuito corresponde a 1.



**Figura 8: Função XOR com 4 entradas.**

**Dados**

**2.1.** Implementação de uma porta NAND de 3 entradas com 3 níveis de portas.

A tabela verdade obtida após a construção do circuito encontra-se abaixo:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | Captura de Tela 2015-03-27 às 11.12.47.png | **A.B** | Captura de Tela 2015-03-27 às 11.13.52.png |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 |

**Tabela 1 - Tabela verdade de uma porta NAND de 2 entradas, de uma porta NAND negada e te uma porta NAND de 3 entradas**

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

A tabela verdade obtida após a construção do circuito encontra-se abaixo:

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A(XOR)B** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Tabela 2 - Tabela verdade de uma porta XOR de 2 entradas**

**2.3.** Verificação da função XOR usando a porta XOR (CI7486)

A tabela verdade obtida para esta parte do experimento foi a mesma obtida na parte 2.2 (acima).

**2.4.** Implementação da porta XOR de 4 entradas utilizando portas XOR de 2 entradas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **f(A,B,C,D)** |
| 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 | 1 |
| 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 |

**Tabela 3 - Tabela verdade de uma porta XOR de 4 entradas**

**Análise dos dados**

Para todas as partes deste experimento temos que os dados obtidos corresponderam ao esperado, sendo iguais aos resultados teóricos.

Para ambas a parte 2.1 e a parte 2.2 vemos a aplicação direta do Teorema de DeMorgan, já que é possível obter uma função AND e uma função OR (que compoe a função XOR) a partir de uma negação da função AND.

Para a parte 2.1 foi provado como a porta AND pode ser substituída por 2 portas NAND gerando a mesma tabela verdade. Temos:

f(A,B) = NOT(AB)

Aplicando outra porta NAND, usando f(A,B) para as duas entradas, ficamos com:

f(A,B) = NOT(NOT(AB) . NOT(AB) = NOT(NOT(AB))

De acordo com DeMorgan:

f(A,B) = NOT (NOT A + NOT B) = NOT NOT A . NOT NOT B

f(A,B) = AB

Chegando a esse ponto, foi possivel implementar uma porta NAND de três entradas, incluindo f(A,B) e uma nova chave C, finalmente chegando à:

g(A,B,C) = NOT(f(A,B) . C) = NOT(ABC)

Para a parte 2.2 foi comprovada novamente essa universalidade da porta NAND, com a construção de uma porta XOR sendo feita apenas com a utilização da negação de AND. Temos, incialmente:

f(A,B) = NOT(AB)

Por DeMorgan, f(A,B) = NOT A + NOT B

Pegando este resultado e adicionando-o a uma nova porta NAND junto com a chave A:

g(A, f(A,B)) = NOT ((NOT A + NOT B) . A) = NOT(A . NOT B) = NOT A + B

Para a outra porta, onde o resultado é inserido em uma porta NAND com a chave B, temos:

h(B, f(A,B)) = NOT ((NOT A + NOT B) . B) = NOT(B . NOT A) = NOT B + A

Finalmente, juntando h e g em uma última porta NAND, ficaremos com:

F(A,B) = NOT((NOT A + B)(NOT B + A)) = NOT(NOT A + B) + NOT(NOT B + A)

= (NOT NOT A . NOT B) + (NOT NOT B . NOT A) = (A . NOT B) + (B . NOT A)

Que nada mais é do que a função XOR.

Finalmente, na parte 2.4 vemos como a presença de apenas duas portas não deve nos limitar a utilizar apenas duas portas, pois é possível utilizar vários circuitos de duas portas para reproduzir circuitos com mais portas. Com 3 portas XOR foi possível reproduzir o resultado de caso a porta tivesse 4 entradas. Temos que, para duas portas XOR:

f(A,B) = (A . NOT B) + (B . NOT A)

g(C,D) = (C . NOT D) + (C . NOT D)

Juntando f e g em uma última porta XOR, ficaremos com:

h(A,B,C,D) = (NOT((A . NOT B) + (B . NOT A)) . (C . NOT D) + (C . NOT D)) + (NOT (C . NOT D) + (C . NOT D) . (A . NOT B) + (B . NOT A))

Simplificando esta expressão, ficaremos com a soma dos minitermos indicando os oito casos em que a saída do circuito será 1:

h(A,B,C,D) = ABD(NOT C) + ABC(NOT D) + (NOT A)(NOT B)(NOT C)D + (NOT A)(NOT B)(NOT D)C + CD(NOT A)B + CD(NOT B)A + (NOT C)(NOT D)(NOT A)B + (NOT C)(NOT D)(NOT B)A

É possível notar que, nesta simplificação, serão verdadeiros apenas os casos em que temos um número ímpar de entradas 1.

**Conclusão**

Formos apresentados às portas NAND e XOR, pudemos fazer de ambas as partes a implementaçao das tabelas verdade. Além disso, verificamos o Teorema DeMorgan e a fundamentabilidade da porta NAND.

**Auto-Avaliação**

**1.** Se uma porta NAND de 3 entradas tiver duas de suas entradas ligadas a 5 V e a terceira entrada for A, então a saída será:

a) A

**b)** Captura de Tela 2015-03-27 às 11.15.36.png

c) 1

d) 0

**2.** Se uma entrada de uma porta NOR de 3 entradas for 1 e as outras entradas não forem conhecidas, então a saída será:

**a) 0**

b) 1

c) Indeterminada

d) NDA

3. Pelo teorema de De Morgan a função f = [ Captura de Tela 2015-03-27 às 11.06.26.png] é igual a:

a) [ Captura de Tela 2015-03-27 às 11.07.58.png]

b) [Captura de Tela 2015-03-27 às 11.09.50.png ]

c) [ Captura de Tela 2015-03-27 às 11.10.52.png]

**d) [** Captura de Tela 2015-03-27 às 11.11.45.png**]**

**4.** Para usar uma porta XOR como NOT:

a) ambas as entradas devem ser 1

b) ambas as entradas devem ser 0

c) uma das entradas deve ser aterrada

**d) uma das entradas deve ser ligada a 5V**

**5**. Se as entradas de uma porta XOR forem iguais, a saída será 1?

a) Certo.

**b) Errado.**

c) Depende do valor das entradas