

# Experimento 5

## Circuitos Combinacionais: Conversão de Códigos

Isaac Lopes, 12/0120801

Lucas Mafra Chagas, 12/0126443

Marcelo Giordano Martins Costa de Oliveira, 12/0037301

<sup>1</sup>Dep. Ciência da Computação – Universidade de Brasília (UnB)  
CiC 116351 - Circuitos Digitais - Turma C

{giordano.marcelo, chagas.lucas.mafra, isaaclopinho}@gmail.com

**Abstract.** *In this report, it was designed an BCD code converter with 5bit using Quartus 13.01 software.*

**Resumo.** *Nesse experimento, foi projetado um conversor de código de BCD com 5bits utilizando o software Quartus 13.01.*

### Objetivos

Estudo e projeto de conversores de código. O código de entrada utilizado como exemplo é o BCD e o de saída é um código de 5 bits. São apresentados, ainda, os códigos de Gray, excesso-3, biquinário e outros.

### Materiais

- software Quartus II versão 13.0
- kit de desenvolvimento em FPGA DE2 Altera

### Introdução

A necessidade de realizar a comunicação entre dois sistemas digitais diferentes leva à criação de sistemas de interligação, que possui como função básica a conversão de códigos de um sistema para o outro. Temos um código quando uma palavra  $X = y_1, y_2, \dots, y_n$ , que é interpretada de certa maneira, é escrita de maneira diferente,  $X = z_1, z_2, \dots, z_n$  e essa nova forma de escrita só é compreendida se a entidade que lê sabe como funciona o código. Um conversor de códigos é um tipo de circuito que pega a palavra  $X$  em sua forma original, escrita em um determinado código e a transforma. A palavra é a mesma mas a sua constituição é diferente, estando de acordo com um novo código. É dito que um conversor é um codificador quando sua entrada representa um código que é ativado uma e somente uma linha de entrada por vez. Analogamente, é dito que um conversor é um decodificador quando sua saída representa um código que é ativado uma e somente uma linha de saída por vez. Por exemplo, um conversor é codificador e decodificador para códigos decimais.

O código BCD é um código muito comum. Este código é um código ponderado, ou seja, a cada bit é atribuído um peso. Portanto, números BDC podem ser convertidos para decimal da seguinte forma:

$$N = 8B_3 + 4B_2 + 2B_1 + 1B_0$$

Onde  $B_i$  é igual ao valor 0 ou 1,  $i \in 0, 1, 2, 3$ . O código BCD é usado para representar apenas valores de 0 a 9. O código 2 4 2 1 também é ponderado, como o BCD, mas os valores dos seus pesos são diferentes. O código Excesso de 3 não é ponderado pois temos que a representação dos valor é equivalente à sua representação somada de três em binário. O código biquinário é ponderado, com os pesos sendo 5043210. O código de Gray é o código conhecido como código binário refletido. Ele é vantajoso pois dois valores consecutivos diferem em apenas 1 bit. Esse código é utilizado no mapa de Karnaugh. O experimento relatado aqui trabalha com o código BCD e analisa o funcionamento de um conversor deste código.

## Procedimentos

Nesse experimento tivemos que:

- Obter as funções booleanas através do mapa de Karnaugh;
- Desenhar o diagrama lógico total usando o número mínimo de portas lógicas e realizar a simulação funcional e temporal;
- Desenhar o diagrama lógico total usando somente portas NAND de 2 e 4 entradas e realizar a simulação funcional e temporal;
- Sintetizar o conversor de códigos projetado com o número mínimo de portas lógicas.

## Funções booleanas através do mapa de Karnaugh

As funções booleanas obtidas para V, W, X, Y e Z foram obtidas através da análise dos minitermos de cada função, já que, para a implementação com portas NAND a utilização de minitermos é mais favorável. Temos portanto que:

$$V = \overline{A}\overline{B}\overline{C}D + \overline{A}B\overline{C}D + \overline{A}BCD + A\overline{B}\overline{C}D + A\overline{B}CD;$$

$$W = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D};$$

$$X = \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}CD;$$

$$Y = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D};$$

$$Z = \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}CD + A\overline{B}C\overline{D};$$

Como estamos codificando valores em código BCD, temos que os valores de 10 a 15 são don't cares. Portanto, em todos os mapas de Karnaugh apresentados abaixo, temos um 'X' marcado nas casas do mapa referentes a estes valores.

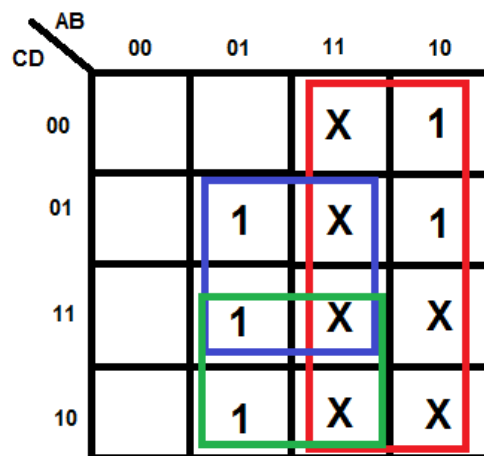


Figure 1. Mapa de Karnaugh da função V

$$V = A + BC + BD$$

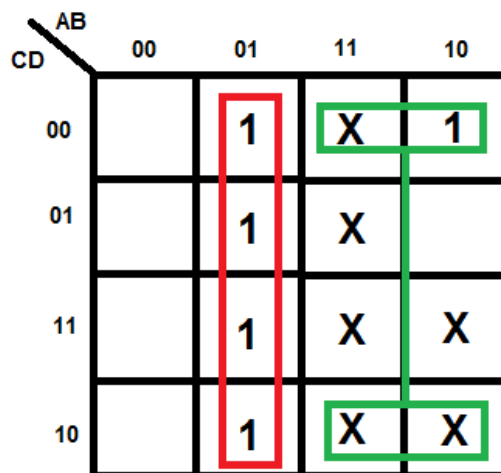


Figure 2. Mapa de Karnaugh da função W

$$W = B + A\overline{D}$$

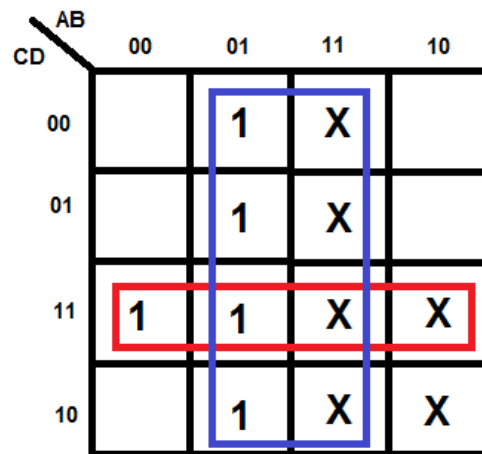


Figure 3. Mapa de Karnaugh da função X

$$X = B + CD$$

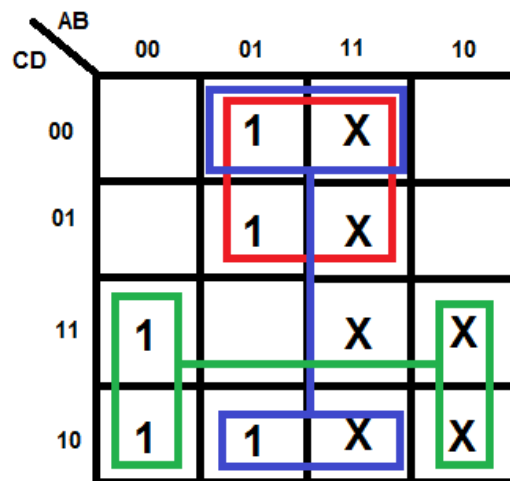


Figure 4. Mapa de Karnaugh da função Y

$$Y = \overline{B}C + B\overline{C} + C\overline{D}$$

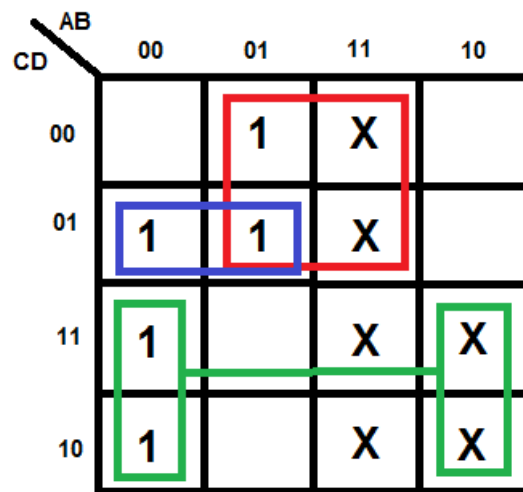


Figure 5. Mapa de Karnaugh da função Z

$$Z = \overline{B}C + B\overline{C} + \overline{A}BD$$

### Diagrama lógico total usando o número mínimo de portas lógicas

A partir das fórmulas obtidas pelo mapa de Karnaugh, obtivemos o seguinte diagrama lógico no software Quartus II:

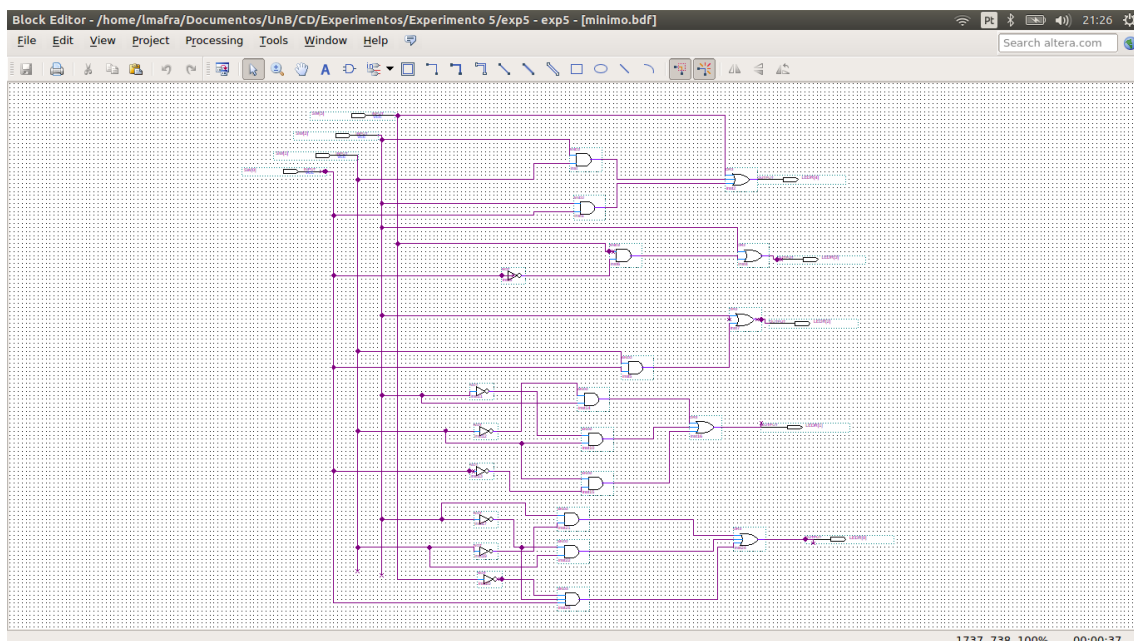
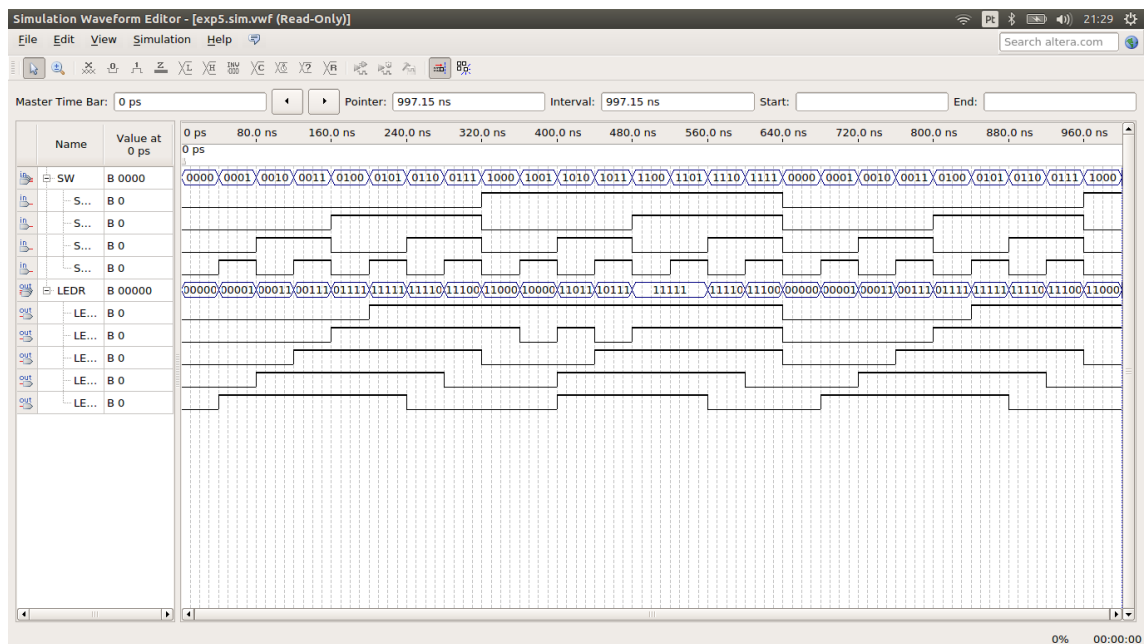


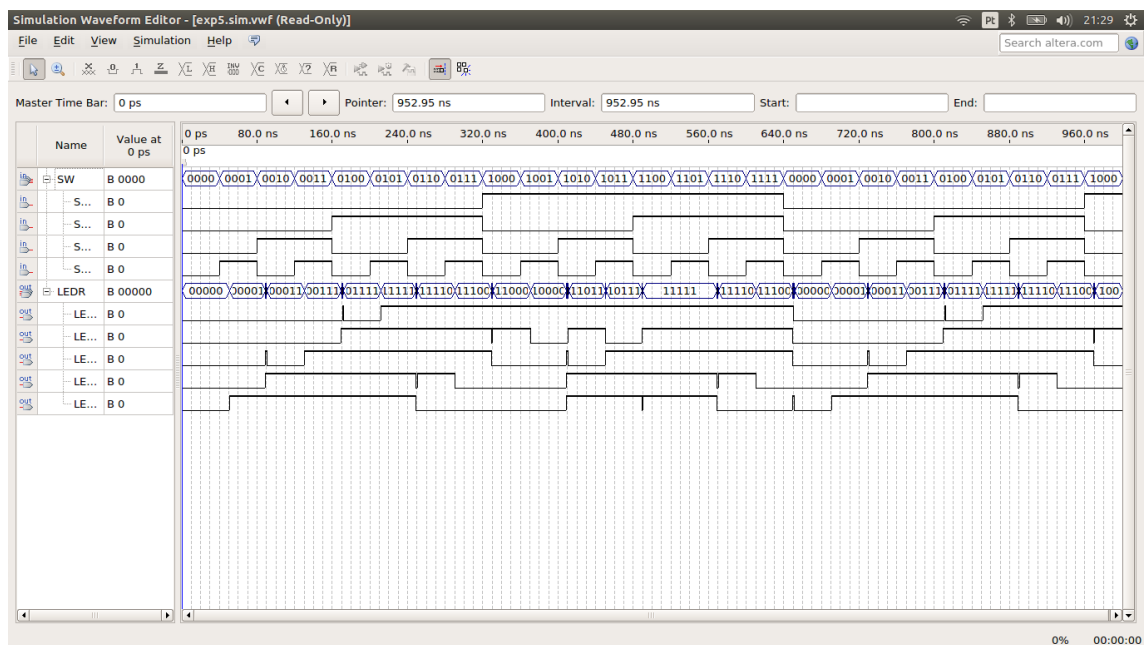
Figure 6. Diagrama no Quartus II.

Depois de estruturar o diagrama lógico, realizamos a:  
Simulação Funcional:



**Figure 7. Simulação Funcional de Portas Mínimas.**

E a Simulação Temporal:



**Figure 8. Simulação Temporal de Portas Mínimas.**

## Diagrama lógico total usando somente portas NAND de 2 e 4 entradas

A partir da simplificação das fórmulas obtidas pelo mapa de Karnaugh, obtivemos o seguinte diagrama lógico no software Quartus II:

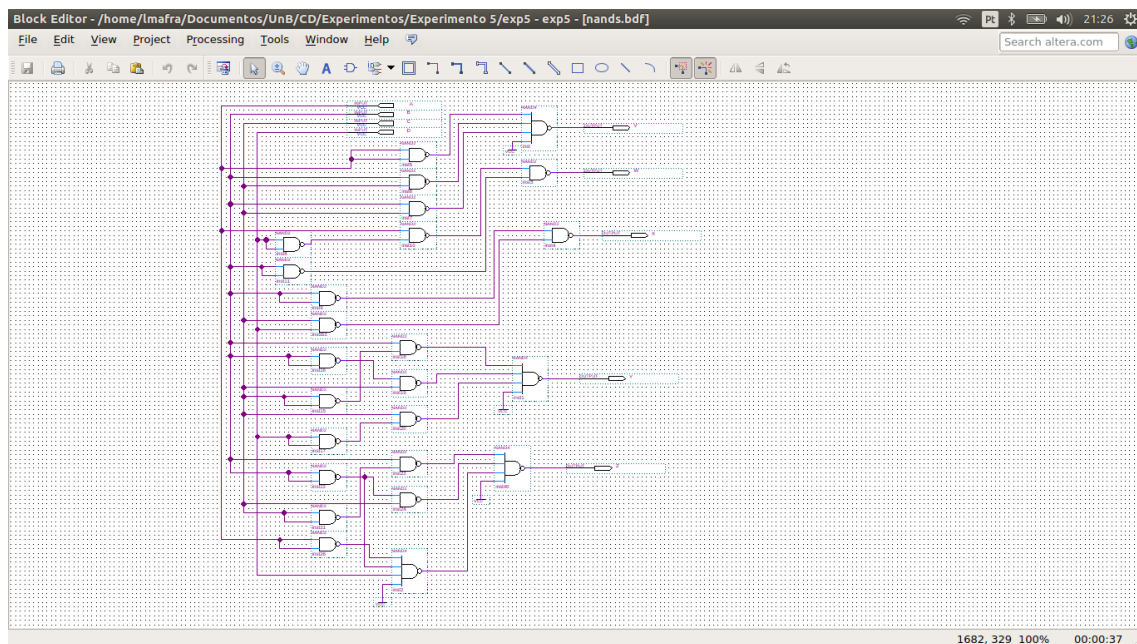


Figure 9. Diagrama no Quartus II.

Depois de estruturar o diagrama lógico, realizamos a:  
Simulação Funcional:

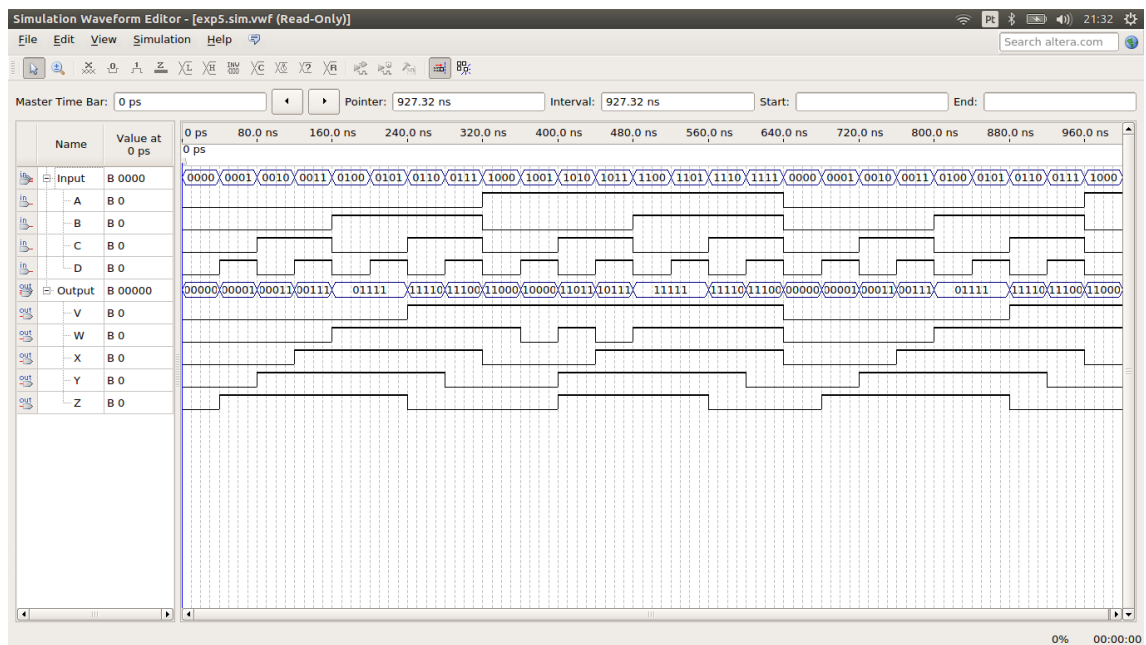
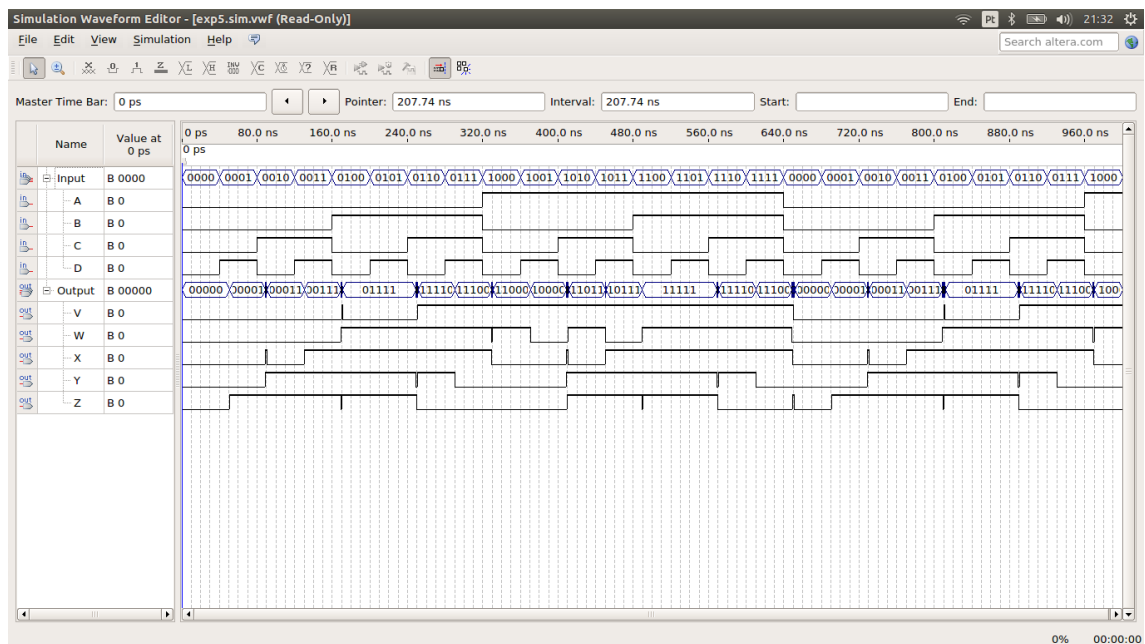


Figure 10. Simulação Funcional com Portas Nands.

E a Simulação Temporal:



**Figure 11. Simulação Temporal com Portas Nands.**

## Conversor de códigos projetado com o número mínimo de portas lógicas

Desenvolvemos no kit de desenvolvimento em FPGA DE2 Altera o código projetado com o número mínimo de portas lógicas. É possível ver o resultado no seguinte link: Vídeo no Youtube

## Análise dos Resultados

## Conclusão

O experimento tornou possível perceber a utilidade de conversores na interface entre sistemas digitais. A implementação desses conversores utilizando circuitos combinacionais foi muito interessante de ser feita, além de ter sido concluída com sucesso. Também foi possível ver, mais uma vez, que as portas NAND são de extrema importância na construção de circuitos digitais, inclusive os conversores, já pela rapidez oferecida por essa porta.



### **Auto-Avaliação**

1. b
2. a
3. c
4. d
5. a
6. a