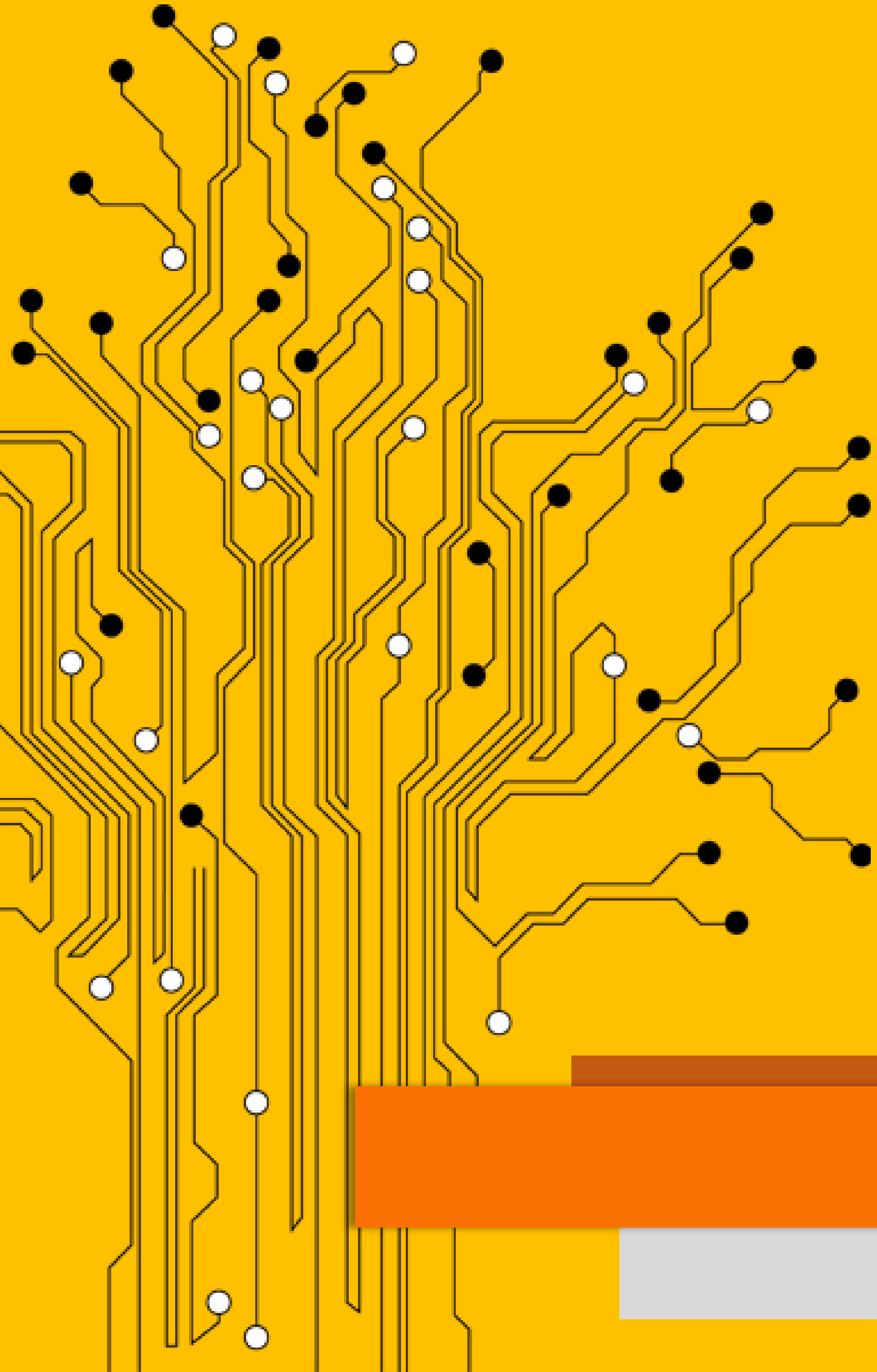


Arquitetura de computadores

CIRCUITOS DIGITAIS

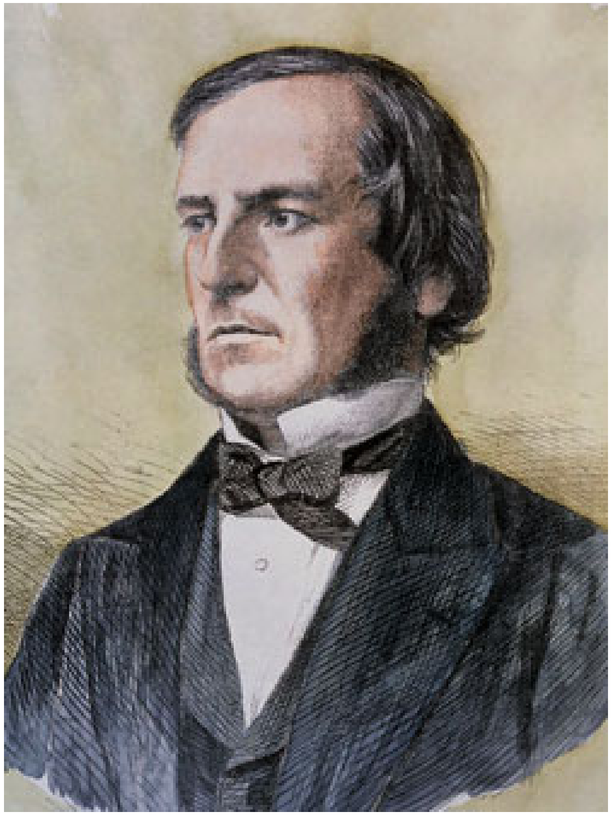
FELIPE G. TORRES

ESSA APRESENTAÇÃO POSSUI QRCODE PARA ACESSAR INFORMAÇÕES ADICIONAIS AOS SLIDES.

Código QR Reader

Código QR

INTRODUÇÃO A CIRCUITOS LÓGICOS



- No nosso dia-a-dia estamos repletos de circunstâncias em que somente dois estados são possíveis : luz apagada ou acesa, pessoa morta ou viva, porta fechada ou aberta, etc.

- Em 1854 o matemático George Boole descreveu um conjunto de regras capaz de relacionar estas circunstâncias (entradas) de maneira a permitir a tomada de decisões (saídas).

- Este conjunto de regras foi denominado de **álgebra booleana**.

FUNÇÕES E VARIÁVEIS LÓGICAS

Seguem algumas definições importantes :

- **Variável booleana:** é uma quantidade que pode ser, em diferentes momentos, igual a 0 ou 1.
- **Função booleana:** associa a cada n variáveis de entrada uma única saída.

Podemos descrever uma função booleana utilizando:

- tabela verdade
- portas lógicas
- equações
- formas de onda

Diferente da álgebra comum, a álgebra booleana possui somente três operações básicas : OR, AND e NOT, conhecidas como operações lógicas.

TABELA VERDADE

Seja uma função $f(A_1,\cdots,A_n)$ com n entradas. A tabela verdade expressa o estado da saída para todas as combinações possíveis dos estados de entrada $\{A_1,\cdots,A_n\}$. Segue m exemplo para duas entradas.

A_1	A_2	$f(A_1,A_2)$
0	0	1
0	1	1
1	0	1
1	1	0

Além de 0s e 1s a função $f(\cdot)$ pode ser igual ao caracter x , chamado de *don't care*. Este caracter serve para indicar que para uma dada combinação de entradas, x pode ser tanto 0 como 1.

- Operação NOT: Para qualquer entrada A, ela é definida como:

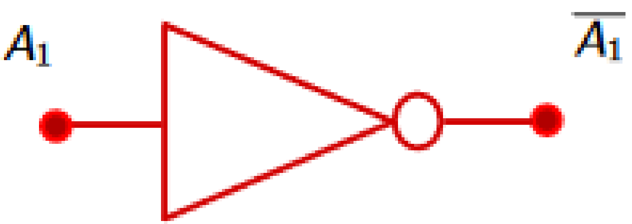
$$f(A) = \bar{A}$$

ou seja, é a entrada negada (barrada). Para uma entrada A_1, por exemplo temos:

Tabela verdade

A_1	$f(A_1)$
0	1
1	0

Porta lógica



- **Operação OR:** Para qualquer entrada $\{A_1, \cdots, A_n\}$, ela é definida como:

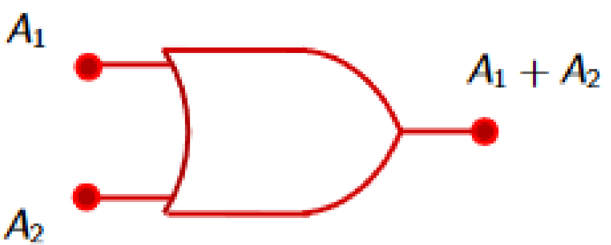
$$f(A_1, \cdots, A_n) = \sum_{i=1}^{n} A_i$$

E vale 1 se qualquer uma das entradas for igual a 1. Para duas entradas temos:

Tabela verdade

A_1	A_2	$f(A_1, A_2)$
0	0	0
0	1	1
1	0	1
1	1	1

Porta lógica



OPERAÇÕES E PORTAS LÓGICAS

- Operação AND: Para qualquer entrada $\{A_1, \cdots, A_n\}$, ela é definida como:

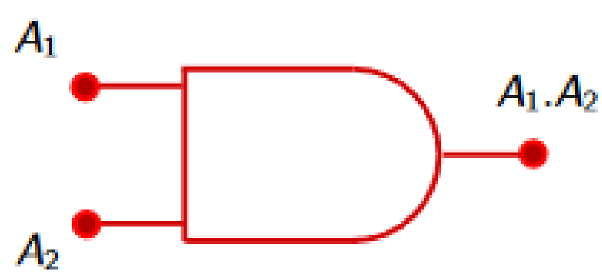
$$f(A_1, \cdots, A_n) = \prod_{i=1}^{n} A_i$$

E vale 1 apenas se todas as entradas forem iguais a 1. Para duas entradas temos:

Tabela verdade

A_1	A_2	$f(A_1, A_2)$
0	0	0
0	1	0
1	0	0
1	1	1

Porta lógica



- Operação NOR: É a operação OR negada. Para duas entradas $\{A_1, A_2\}$, ela é definida como:

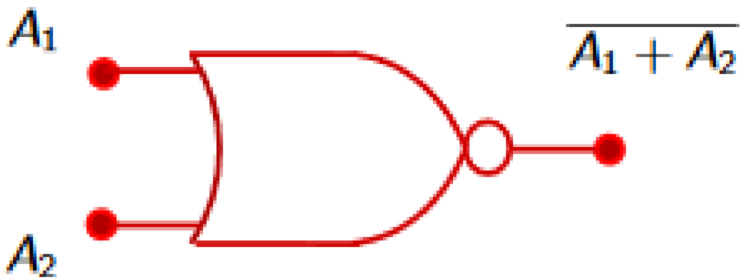
$$f(A_1, A_2) = \overline{A_1 + A_2}$$

E vale 1 apenas se todas as entradas forem iguais a 0. Para duas entradas temos:

Tabela verdade

A_1	A_2	$f(A_1, A_2)$
0	0	1
0	1	0
1	0	0
1	1	0

Porta lógica



OPERAÇÕES E PORTAS LÓGICAS

- Sobre a porta NOR, podemos realizar os seguintes comentários:
  - Utilizando a tabela verdade podemos verificar que:

$$\overline{A_1 + A_2} = \overline{A_1} \,.\, \overline{A_2}$$

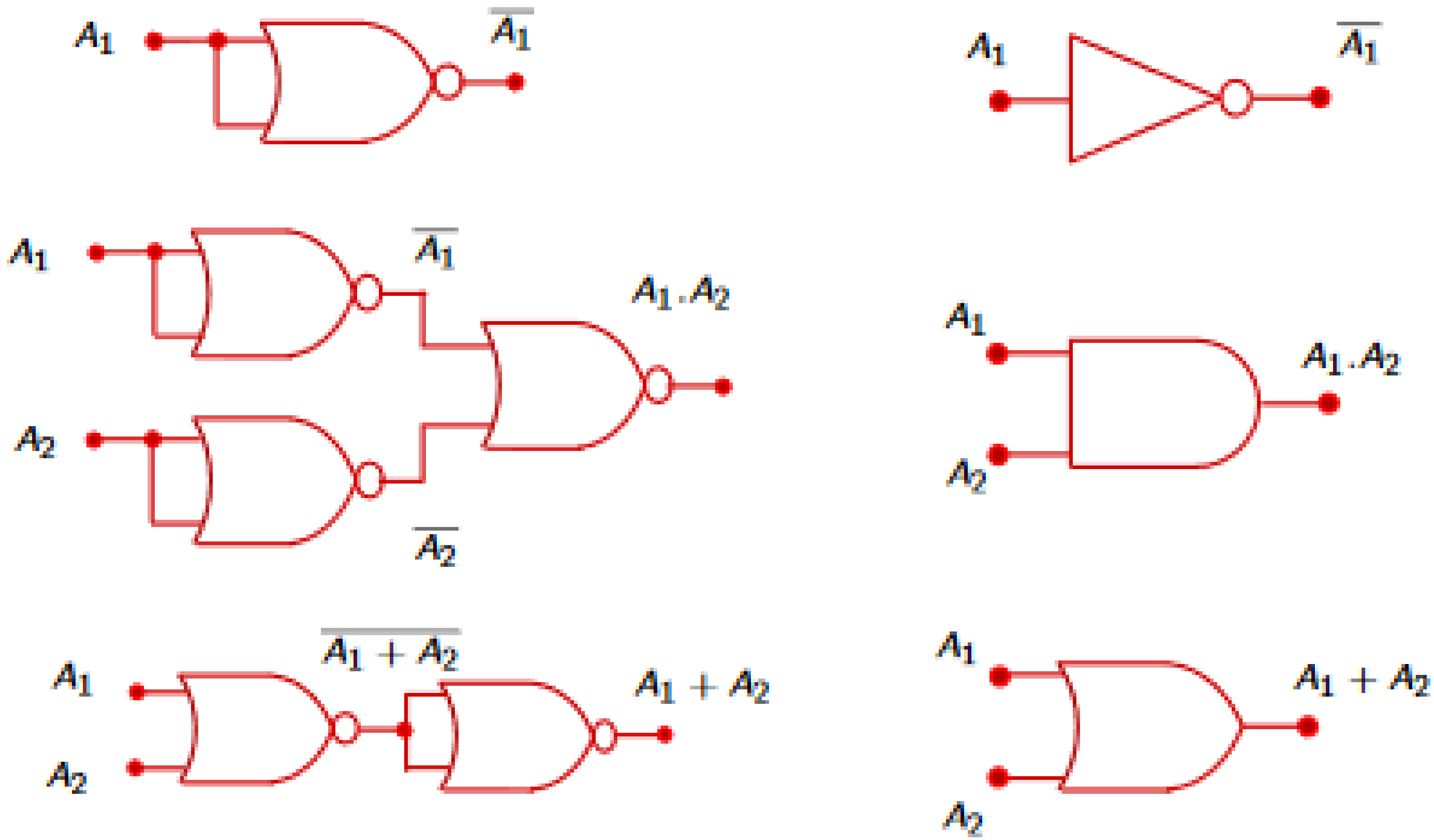
Que é um dos resultados do teorema de Morgan que veremos a seguir...

Tabela verdade

A_1	A_2	$\overline{A_1 + A_2}$	$\overline{A_1} \,.\, \overline{A_2}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

OPERAÇÕES E PORTAS LÓGICAS

- Utilizando apenas a porta NOR, podemos obter as outras três portas básicas:



- Operação NAND: É a operação AND negada. Para duas entradas $\{A_1, A_2\}$, ela é definida como:

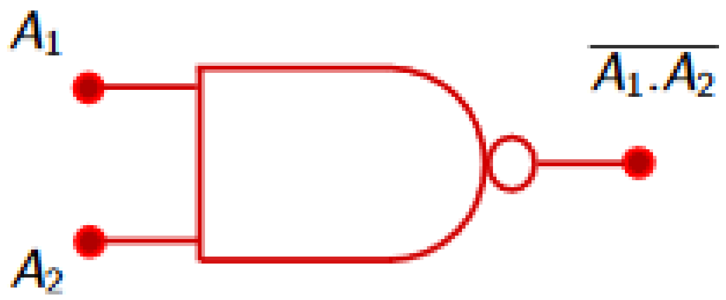
$$f(A_1, A_2) = \overline{A_1 \,.\, A_2}$$

E vale 0 apenas se todas as entradas forem iguais a 1. Para duas entradas temos:

Tabela verdade

A_1	A_2	$f(A_1, A_2)$
0	0	1
0	1	1
1	0	1
1	1	0

Porta lógica



- Sobre a porta NAND, podemos realizar os seguintes comentários:

  - Utilizando a tabela verdade podemos verificar que:

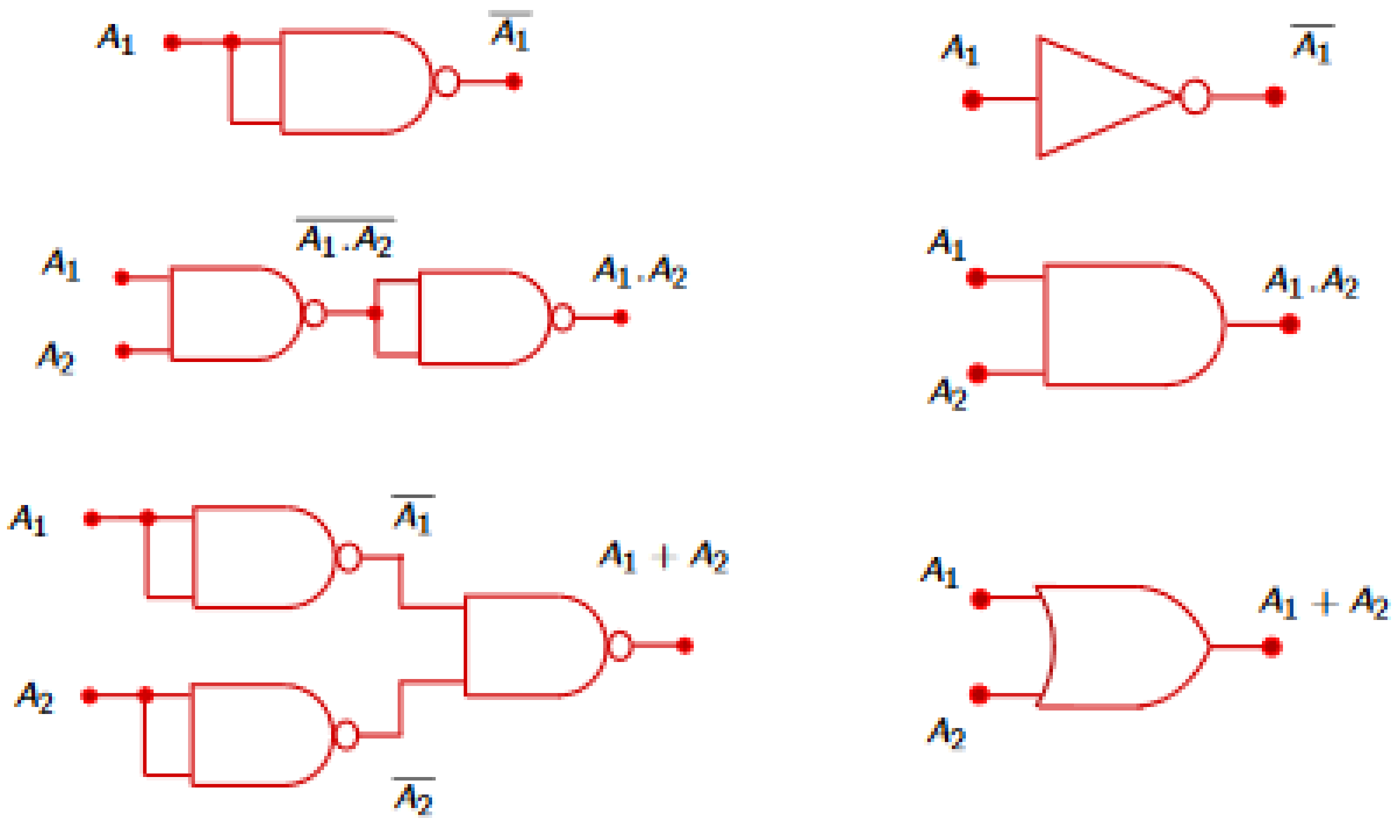
$$\overline{A_1}\,.\,\overline{A_2} = \overline{A_1 + \,A_2}$$

Que é um dos resultados do teorema de Morgan que veremos a seguir...

Tabela verdade

A_1	A_2	$\overline{A_1 . A_2}$	$\overline{A_1} + \overline{A_2}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

- Utilizando apenas a porta NAND, podemos obter as outras três portas básicas:



- Operação XOR (ou exclusivo): Definida para duas entradas $\{A_1, A_2\}$, ela é definida como:

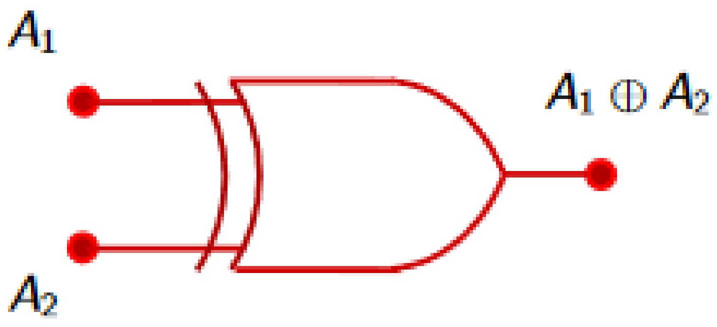
$$f(A_1, A_2) = \overline{A_1}\,.A_2 + \overline{A_2}\,.A_1 = A_1 \oplus A_2$$

E vale 1 apenas se as entradas forem diferentes. Para duas entradas temos:

Tabela verdade

A_1	A_2	$f(A_1, A_2)$
0	0	0
0	1	1
1	0	1
1	1	0

Porta lógica



- Operação XNOR (coincidência): Definida para duas entradas $\{A_1, A_2\}$, ela é definida como:

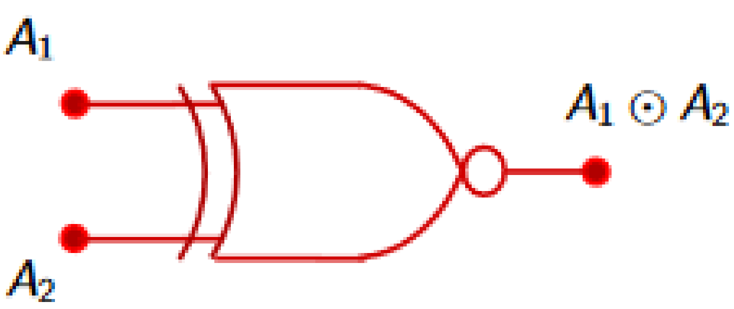
$$f(A_1, A_2) = \overline{A_1} \,.\, \overline{A_2} + A_1 . A_2 = A_1 \odot A_2$$

E vale 1 apenas se as entradas forem iguais. Para duas entradas temos:

Tabela verdade

A_1	A_2	$f(A_1, A_2)$
0	0	1
0	1	0
1	0	0
1	1	1

Porta lógica



- As regras operacionais de minimização utilizando a álgebra de Boole decorrem dos postulados e propriedades a segui:

  - Postulados da complementação

$$\bar{\bar{A}} = A$$

  - Postulados da adição

$$A + 0 \; = A, A + 1 = 1, A + A = A, A + \bar{A} = 1$$

  - Postulados da multiplicação

$$A.0 \; = 0, A.1 = A, A.A = A, A.\bar{A} = 0$$

  - Propriedades: Comutativa, associativa e distributiva são válidas para a adição e a multiplicação.

- O seguinte teorema ́e importante pois permite simplificar expressões booleanas ⇒minimização

Teorema de De Morgan

As seguintes igualdades são verdadeiras :

- $\overline{A \,.\, B \,.\, C \,.\, \cdots \,.\, N} = \overline{A} + \overline{B} + \cdots + \overline{N}$
- $\overline{A + B + C + \cdots + N} = \overline{A} \,.\, \overline{B} \,.\, \cdots \,.\, \overline{N}$

- Exemplo 1 : Minimize a expressão sem utilizar o teorema.

$$
\begin{aligned}
\bar{A}\bar{B} + \bar{A}B + A\bar{B} &= \bar{A}(B + \bar{B}) + A\bar{B} \\
&= \bar{A}(1 + \bar{B}) + A\bar{B} \\
&= \bar{A} + (A + \bar{A})\bar{B} \\
&= \bar{A} + \bar{B}
\end{aligned}
$$

TEOREMA DE DE MORGAN

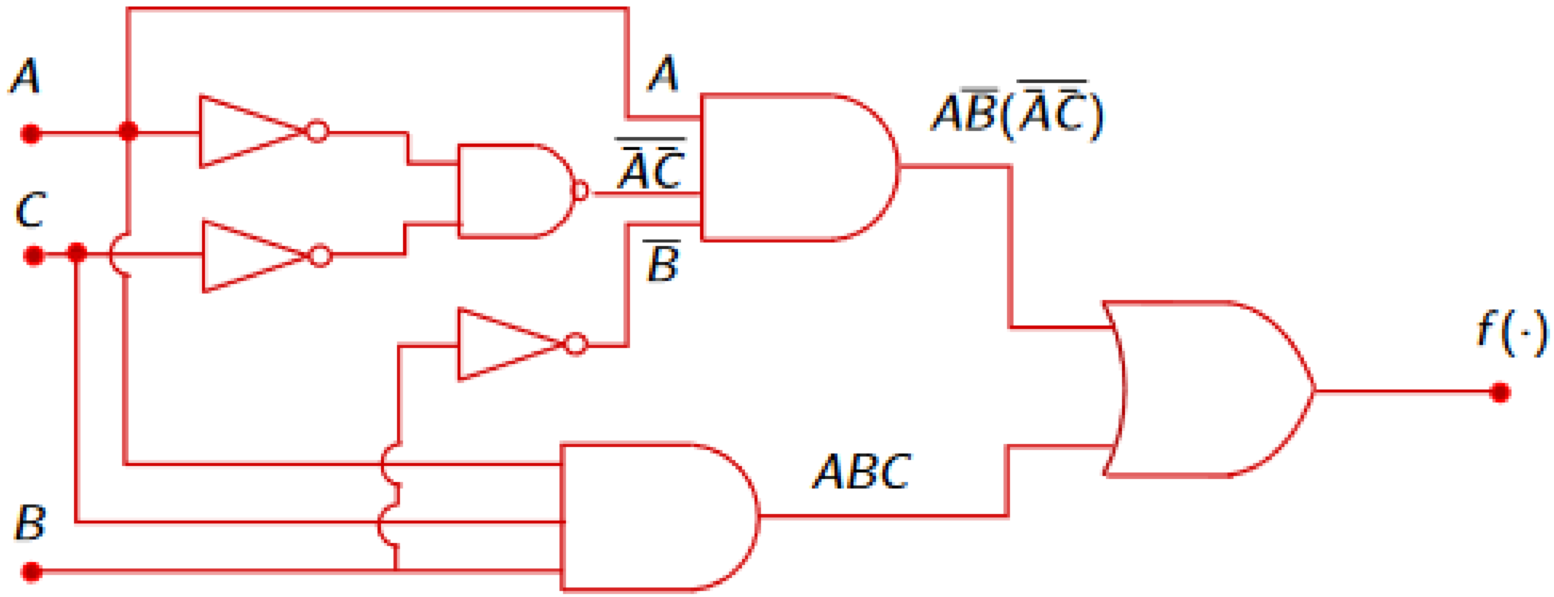
- Exemplo 2 : Minimize a mesma expressão utilizando o teorema de De Morgan.

$$
\begin{aligned}
\bar{A}\bar{B} + \bar{A}B + A\bar{B} &= \bar{A}(B + \bar{B}) + A\bar{B} \\
&= \overline{\overline{\bar{A} + A\bar{B}}} \\
&= \overline{A.(\bar{A} + B)} \\
&= \overline{AB} \\
&= \overline{A} + \overline{B}
\end{aligned}
$$

- Exemplo 3 : Minimize a seguinte expressão

$$
\begin{aligned}
ABC + A\overline{B} + A\overline{C} &= A(BC + \overline{B} + \overline{C}) \\
&= A(BC + (\overline{\overline{\overline{B} + \overline{C}}})) \\
&= A(BC + \overline{BC}) \\
&= A
\end{aligned}
$$

- Descreva a expressão lógica que representa o circuito a seguir



- A expressão lógica é dada por

$$f(A,B,C) = A \,.\, B \,.\, C + A \,.\, \overline{B} \,.\, (\overline{\overline{A} \,.\, \overline{C}})$$

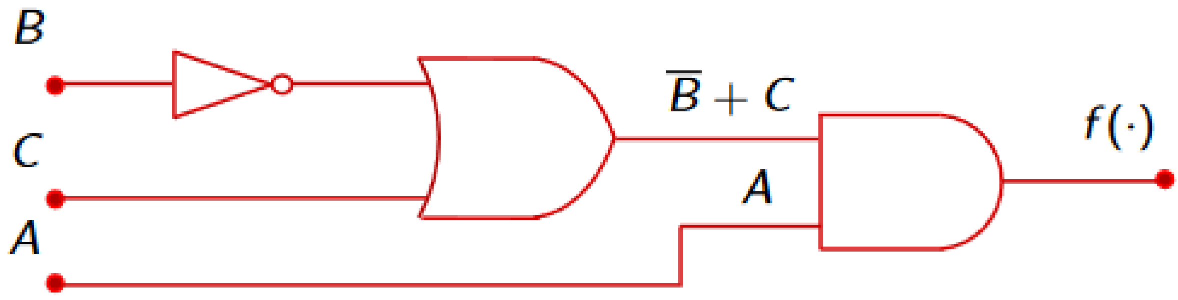
Minimizar a expressão de um circuito lógico, significa obter uma outra equivalente com menos termos e operações. Isto implica em menos portas lógicas e conexões.

- Como vimos, podemos usar a álgebra de Boole para realizar a minimização.
- Neste caso, a simplificação nem sempre é óbvia.
- Geralmente, podemos seguir dois passos essenciais :
  - colocar a expressão na forma de soma de produtos
  - identificar fatores comuns e realizar a fatoração
- Algumas vezes devemos contar com habilidade e experiência para obter uma boa simplificação.

- Utilizando a álgebra de Boole, podemos minimizar a expressão da função do exercício anterior

$$
\begin{aligned}
f(A,B,C) &= A\,.\,B\,.\,C + A\,.\,\overline{B}\,.\,(\overline{\overline{A}\,.\,\overline{C}}) \\
&= A\,.\,B\,.\,C + A\,.\,\overline{B}\,.\,(A + C) \\
&= A\,.\,C\,.\,(B + \overline{B}) + A\,.\,\overline{B} \\
&= A\,.\,(C + \overline{B})
\end{aligned}
$$

- O circuito lógico simplificado é dado por.



A quantidade de portas lógicas foi reduzida de 7 para 3 ! ! !

- A partir do circuito apresentado anteriormente, obtenha a sua tabela verdade e, a partir dela, obtenha a expressão lógica.

A	B	C	$f(A,B,C)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

- Utilizando a tabela, sua expressão lógica é dada por

$$f(A,B,C) = A\,.\,\overline{B}\,.\,\overline{C} + A\,.\,\overline{B}\,.\,C + A\,.\,B\,.\,C$$

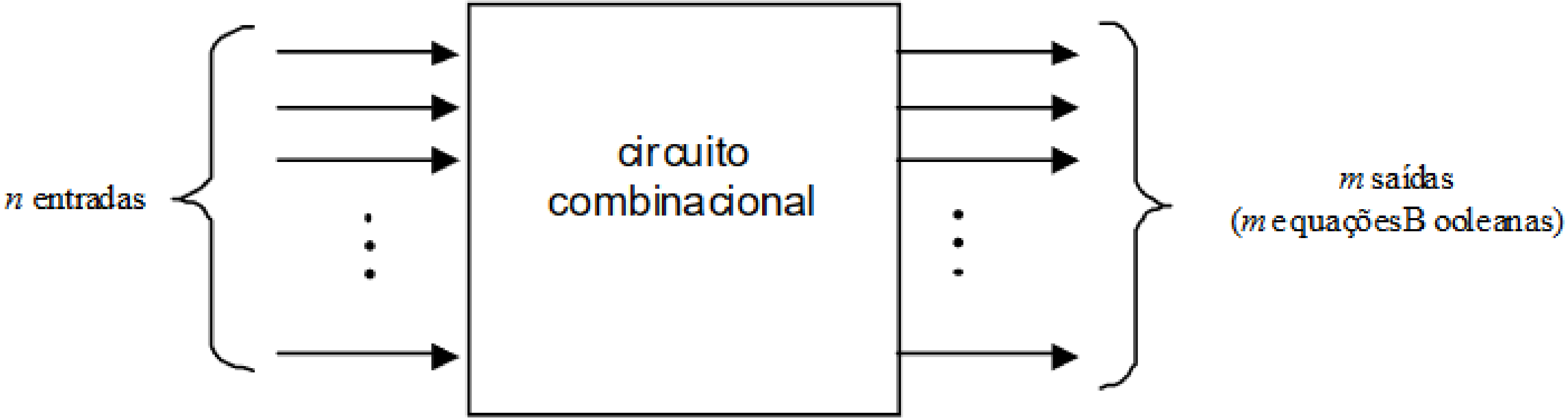
embora seja equivalente à função obtida através do circuito, ela possui um número maior de termos.

INTRODUÇÃO A CIRCUITOS LÓGICOS

Os circuitos lógicos dos sistemas digitais podem ser de dois tipos:

- **Circuitos combinacionais**
  - Um circuito combinacional é constituído por um conjunto de portas lógicas as quais determinam os valores das saídas diretamente a partir dos valores atuais das entradas.
- **Circuitos sequenciais**
  - Um circuito sequencial, por sua vez, emprega elementos de armazenamento denominados latches e flip-flops, além de portas lógicas.

INTRODUÇÃO A CIRCUITOS COMBINACIONAIS



ANÁLISES DE CIRCUITOS COMBINACIONAIS

O objetivo da análise de um circuito combinacional é determinar seu comportamento.

Então, dado o diagrama de um circuito, deseja-se encontrar as equações que descrevem suas saídas.

Uma vez encontradas tais equações, pode-se obter a tabela verdade, caso esta seja necessária.

É importante certificar-se que o circuito é combinacional e não sequencial.

Um modo prático é verificar se existe algum caminho (ou ligação) entre saída e entrada do circuito. Caso não exista, o circuito é combinacional.

ANÁLISES DE CIRCUITOS COMBINACIONAIS

- Vamos agora utilizar os conceitos iniciais apresentados para realizar a síntese de alguns circuitos combinacionais importantes :
    - meio somadores e somadores completos
    - comparadores
    - codificadores e decodificadores
    - multiplexadores e demultiplexadores

ANÁLISES DE CIRCUITOS COMBINACIONAIS

- Vamos agora utilizar os conceitos iniciais apresentados para realizar a síntese de alguns circuitos combinacionais importantes :
  - meio somadores e somadores completos
  - comparadores
  - codificadores e decodificadores
  - multiplexadores e demultiplexadores

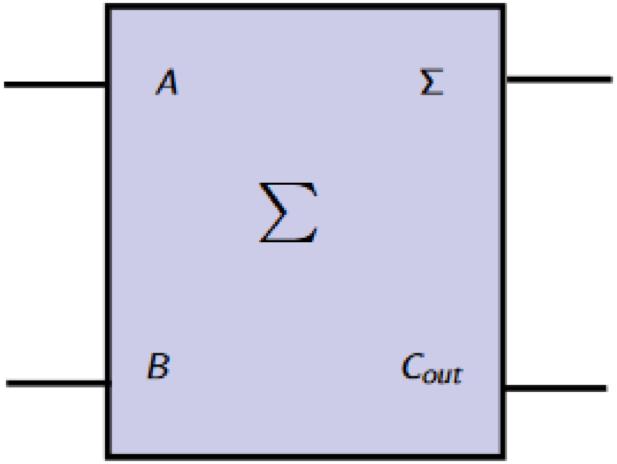
MEIO SOMADORES

- O meio somador aceita duas variáveis de entrada A e B e possui como saídas a soma Σ e o carry out C_{out}.

Tabela verdade

A	B	Σ	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Meio somador



MEIO SOMADORES

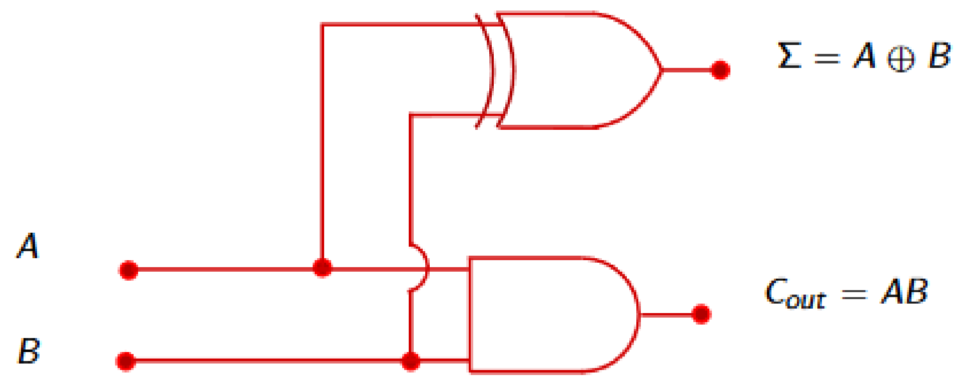
- Não é difícil verificar que

$$\begin{aligned}\Sigma &= \bar{A}B + A\bar{B} \\ &= A \oplus B\end{aligned}$$

e que

$$C_{out} = AB$$

- Seu circuito é dado por



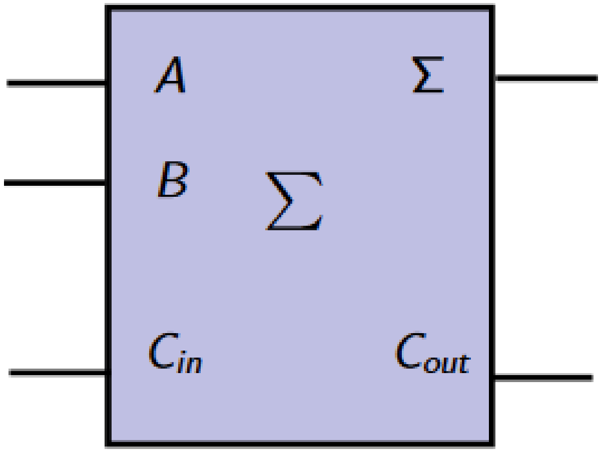
SOMADOR COMPLETO

- O somador completo possui como variáveis de entrada A, B e o carry in C_{in} e como variáveis de saída a soma Σ e o carry out C_{out}.

Tabela verdade

A	B	C_{in}	Σ	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Somador completo



SOMADOR COMPLETO

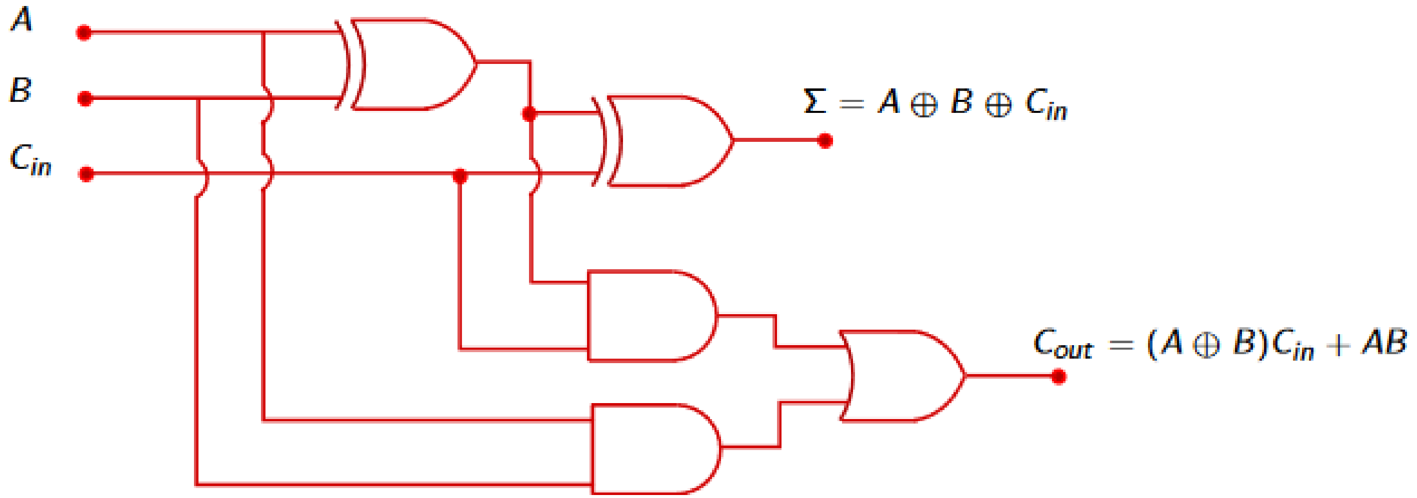
- Podemos verificar que

$$\Sigma = A \oplus B \oplus C_{in}$$

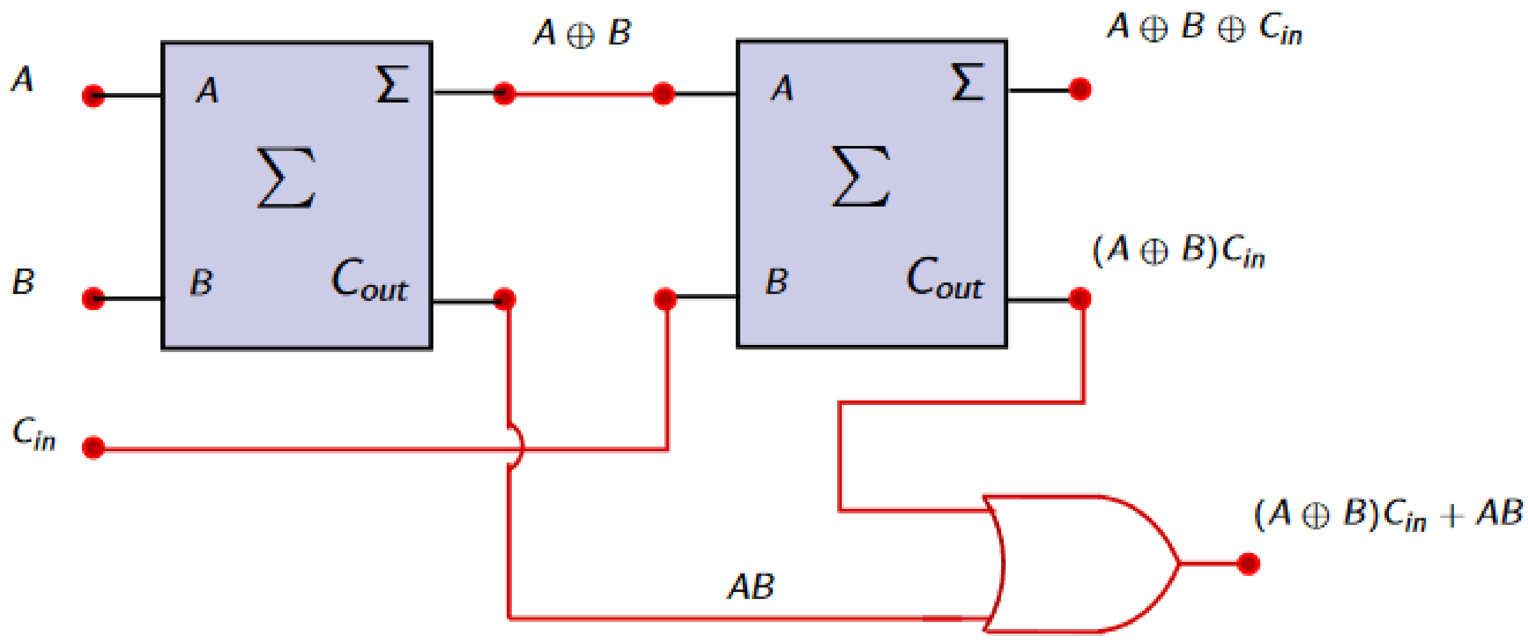
e que

$$\begin{aligned} C_{out} &= \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in} \\ &= (A \oplus B)C_{in} + AB \end{aligned}$$

- Seu circuito é dado por

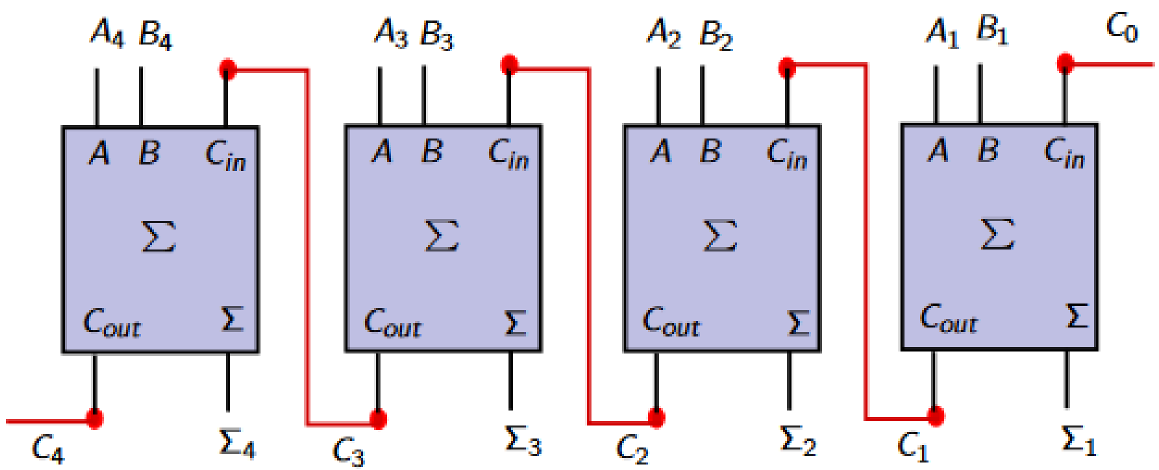


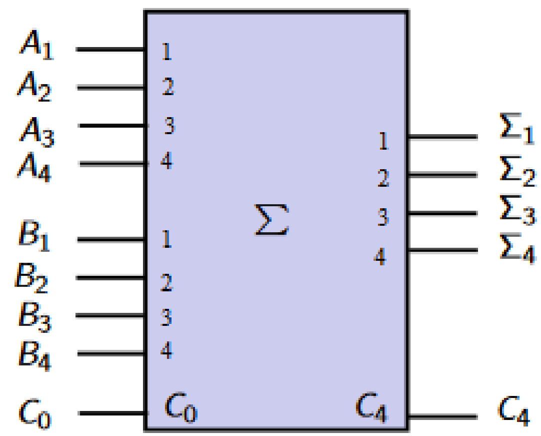
- O somador completo pode ser construído a partir de dois meio somadores.



- Para números de 4 bits, um somador paralelo básico está apresentado a seguir.

$$\begin{array}{r} A_4\ A_3\ A_2\ A_1 \\ +B_4\ B_3\ B_2\ B_1 \\ \hline {\color{red}C_4}\ S_4\ S_3\ S_2\ S_1 \end{array}$$





- Podemos cascatear os somadores de maneira a considerar palavras maiores.

CODIFICADOR E DECODIFICADOR

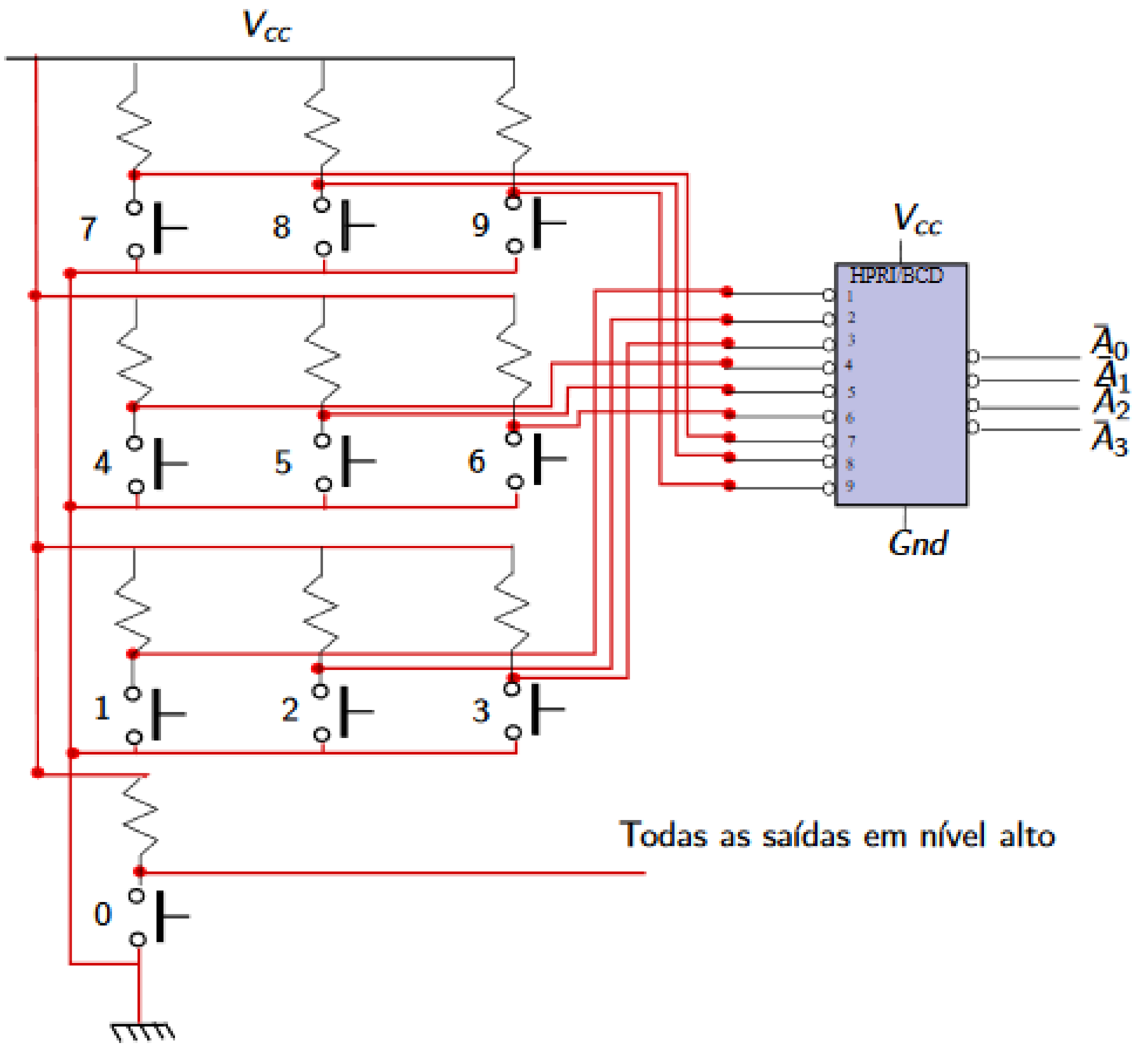
Os circuitos codificadores e decodificadores são aqueles que efetuam a passagem de um código para outro

O circuito codificador torna possível a passagem de um número decimal para um número binário.

Exemplo : o circuito inicial de uma calculadora que transforma decimal (nossa linguagem) para binário (linguagem da máquina).

O circuito decodificador faz o inverso, ou seja, transforma um código desconhecido em outro conhecido. É claro, que o termo codificador ou decodificador depende do referencial que estamos considerando.

- Exemplo de um teclado. Entradas e saídas ativas em nível baixo.



- Segue um exemplo de decodificador BCD para decimal. Ele possui 4 variáveis de entrada e 10 variáveis de saída relacionadas como na tabela a seguir.

A	B	C	D	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0

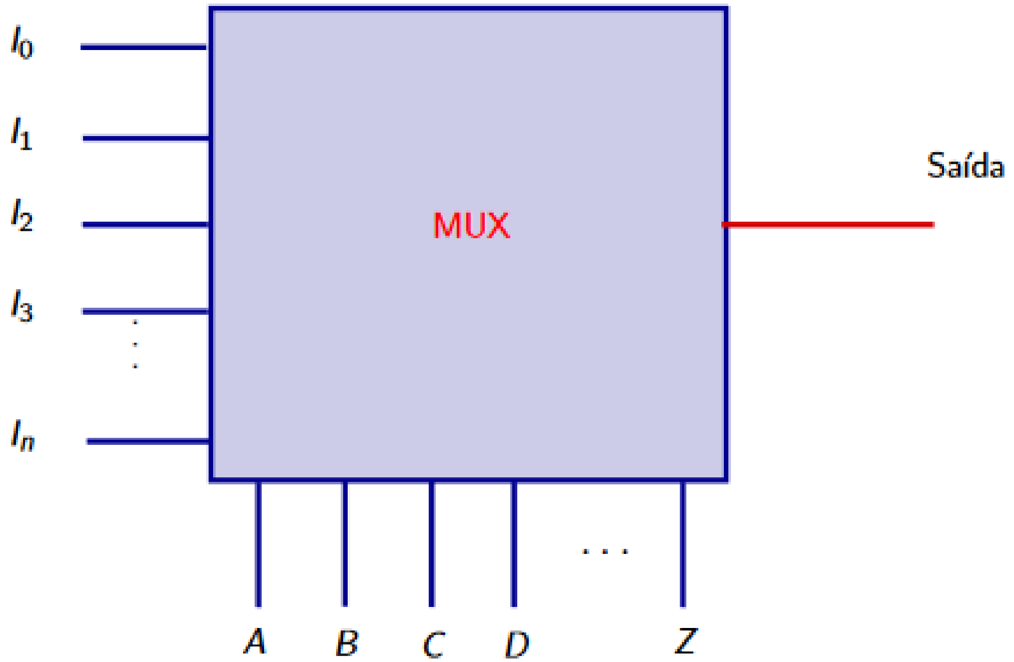
- Procedendo com a simplificação para os demais dígitos, obtemos o seguinte resultado.

$$
\begin{aligned}
D_8 &= A \,.\, \bar{D} \\
D_7 &= B \,.\, C \,.\, D \\
D_6 &= B \,.\, C \,.\, \bar{D} \\
D_5 &= B \,.\, \bar{C} \,.\, D \\
D_4 &= B \,.\, \bar{C} \,.\, \bar{D} \\
D_3 &= \bar{B} \,.\, C \,.\, D \\
D_2 &= \bar{B} \,.\, C \,.\, \bar{D} \\
D_1 &= \bar{A} \,.\, \bar{B} \,.\, \bar{C} \,.\, D \\
D_0 &= \bar{A} \,.\, \bar{B} \,.\, \bar{C} \,.\, \bar{D}
\end{aligned}
$$

Verifique !

MULTIPLEXADOR

- Os multiplexadores são circuitos que permitem passar uma informação digital proveniente de diversos canais em um só canal. Eles também são chamados de selecionadores de dados.
- A Figura a seguir apresenta o esquema de um multiplexador.



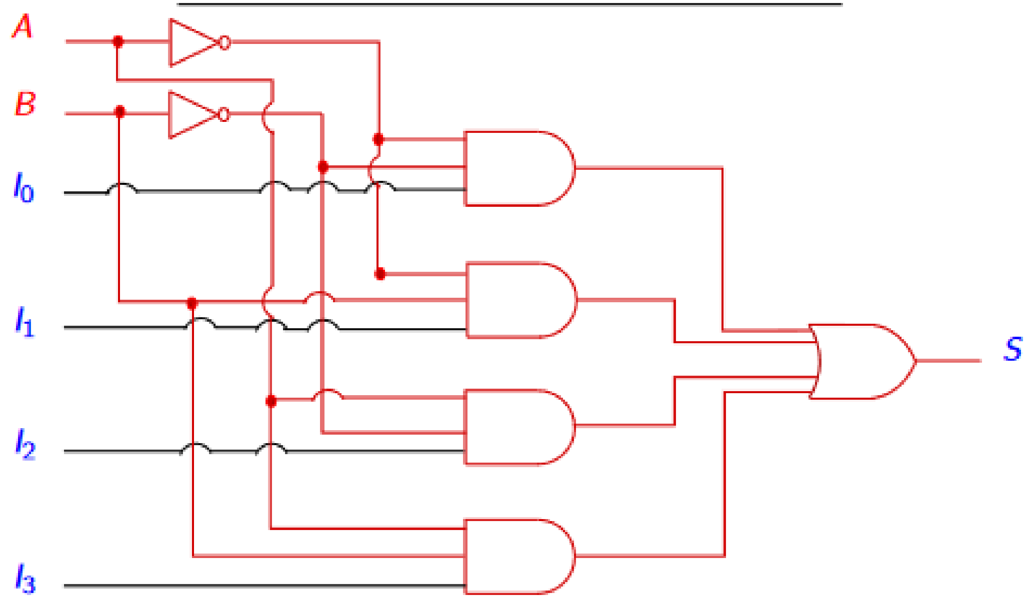
MULTIPLEXADOR

- Vamos supor que temos 4 linhas de informações e apenas uma linha de transmissão. Neste caso o selecionador possui 2 bits e seu circuito está apresentado a seguir.

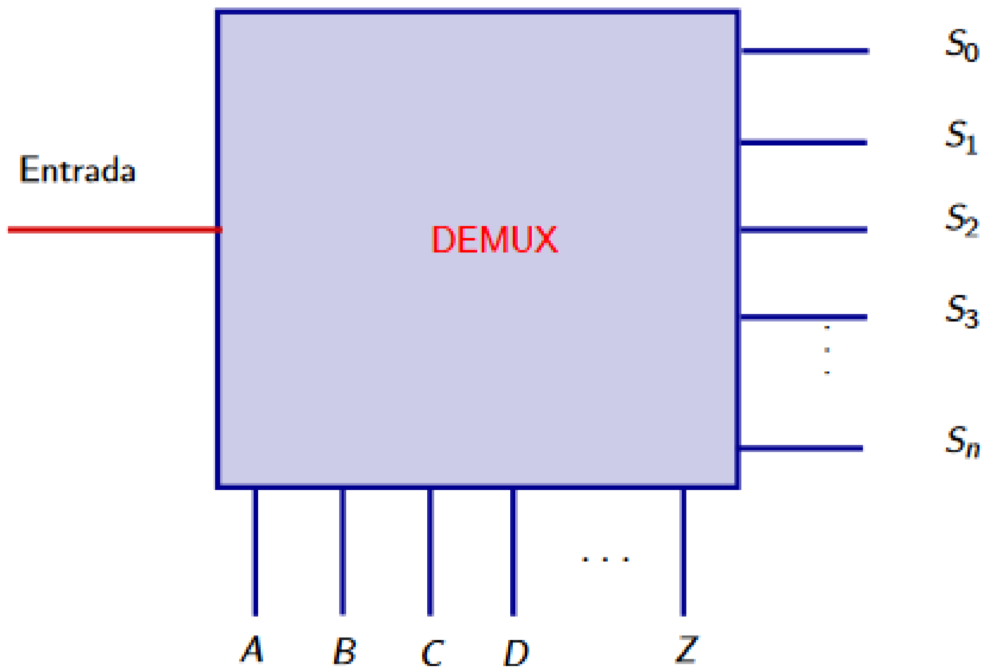
Tabela verdade

A	B	S
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Multiplexador de 4 entradas



- Os demultiplexadores são circuitos capazes de enviar informações contidas em um único canal de entrada à vários canais de saída.
- A Figura a seguir apresenta o esquema de um demultiplexador.

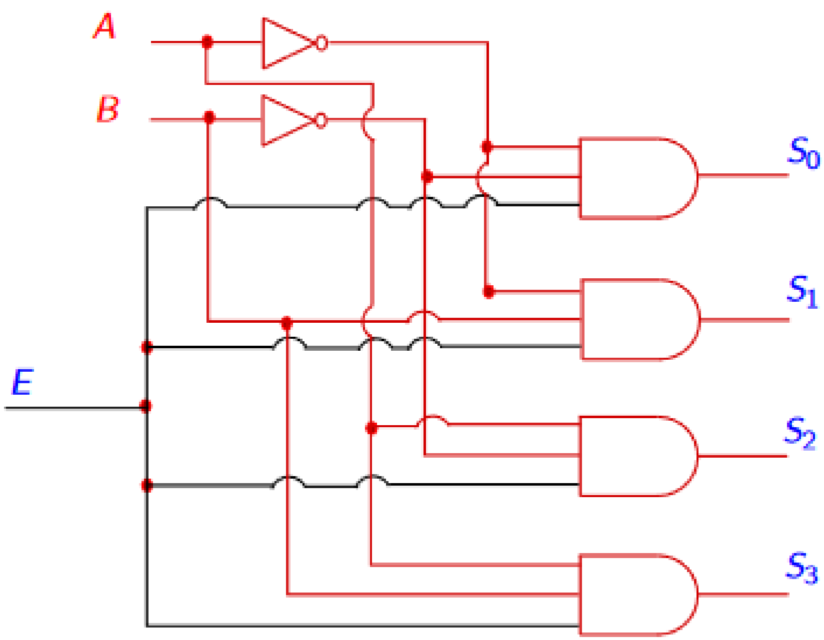


- Vamos supor que temos 1 linha de informação e 4 linhas de transmissão. Neste caso o selecionador possui 2 bits e seu circuito está apresentado a seguir.

Tabela verdade

A	B	S_0	S_1	S_2	S_3
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

Demultiplexador de 4 entradas



- A função do comparador é comparar a magnitude de números binários.
- Para comparar a igualdade de dois bits, basta utilizar a porta lógica XNOR, que fornecerá nível lógico alto apenas na igualdade.
- Desta maneira, para comparar se dois números binários, por exemplo, $A = A_3A_2A_1A_0$ e $B = B_3B_2B_1B_0$ são iguais basta agrupar os bits dois a dois da forma $\{A_3, B_3\}$, $\{A_2, B_2\}$, $\{A_1, B_1\}$ e $\{A_0, B_0\}$ e conectá-los, respectivamente, à quatro portas XNORs. As saídas destas portas são conectadas à uma porta AND de quatro entradas. A saída da porta AND terá nível alto somente se os números forem iguais.

- Para comparar se dois números são diferentes e detectar qual deles é o maior, basta analisá-los começando com o bit mais significativo. Por exemplo, para dois números binários $A = A_3A_2A_1A_0$ e $B = B_3B_2B_1B_0$, o procedimento a seguir é realizado.

  - Se $A_3 = 1$ e $B_3 = 0$ então $A > B$.
  - Se $A_3 = 0$ e $B_3 = 1$ então $A < B$.
  - Se $A_3 = B_3$ realizam-se as verificações anteriores para o bit consecutivo menos significativo.

REFERÊNCIAS

STALLINGS, William. **Arquitetura e organização de computadores: projeto para o desempenho**. 8 ed. São Paulo: Prentice Hall : Person Education, 2010. 624 p. ISBN 9788576055648.

TANENBAUM, Andrew S. **Organização estruturada de computadores**. 5. ed São Paulo: Pearson Prentice Hall, 2007. 449 p. ISBN 9788576050674.

TOCCI & WIDMER **Sistemas Digitais: Princípios e Aplicações**, 10ª Edição. Editora LTC, 2007

http://www.fem.unicamp.br/~grace/circuitos_combinacionais.pdf

Arquitetura de computadores

CIRCUITOS DIGITAIS

FELIPE G. TORRES