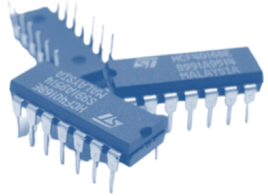
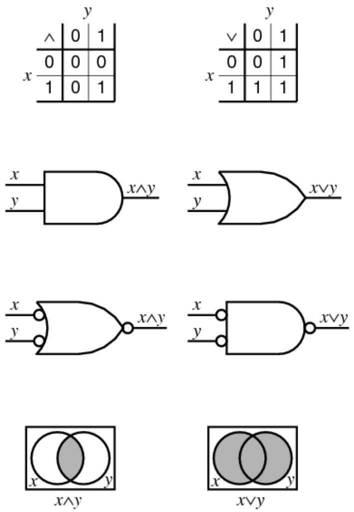
TÉCNICO LISBOA

TÉCNICO LISBOA

# Sistemas Digitais



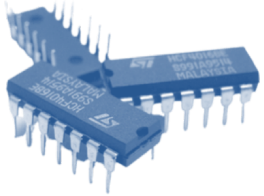



## Síntese de Funções Booleanas

Prof. António Grilo

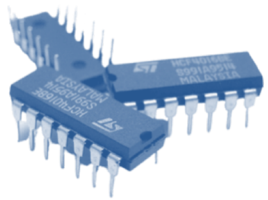
TÉCNICO LISBOA

# Sumário

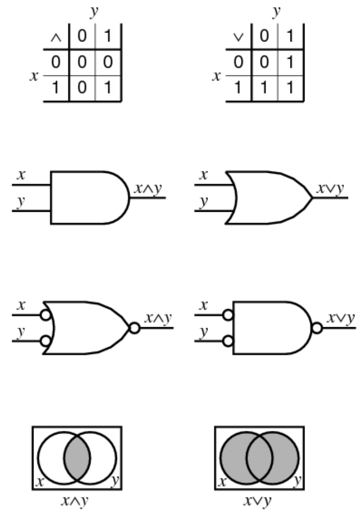


- Álgebra de Boole
  - Operações básicas
  - Propriedades
- Portas Lógicas
  - Portas OR, AND e NOT
  - Portas NOR e NAND
  - Portas XOR e XNOR
  - Saídas tri-state
- Circuitos com portas NAND e NOR
  - Lei de DeMorgan
  - Circuitos com portas NAND
  - Circuitos com portas NOR
- Funções lógicas
  - Representações normalizadas:
    - Mintermos e Soma de produtos
    - Maxtermos e Produto de somas
  - Funções incompletamente especificadas.
- Simplificação e minimização de funções lógicas
  - Teorema da adjacência
  - Representação de funções no quadro de Karnaugh
  - Agrupamentos de uns e zeros
  - Método de minimização de Karnaugh

TÉCNICO LISBOA



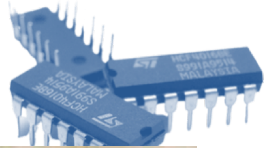
# Álgebra de Boole

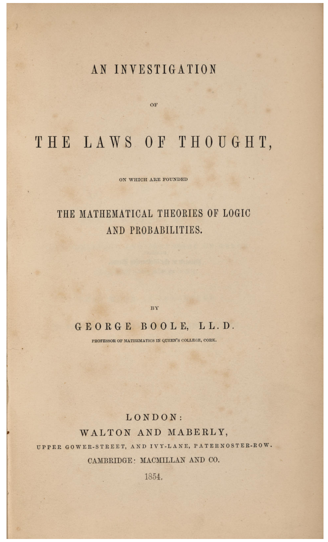



**Sumário:**

- Operações básicas
- Propriedades

# Álgebra de Boole





AN INVESTIGATION

OF

THE LAWS OF THOUGHT,

ON WHICH ARE FOUNDED

THE MATHEMATICAL THEORIES OF LOGIC AND PROBABILITIES.

BY

GEORGE BOOLE, LL.D.

LONDON:
WALTON AND MABERLY,
UPPER GOWER-STREET, AND IVY-LANE, PATERNOSTER-ROW.
CAMBRIDGE: MACMILLAN AND CO.
1854.

- A lógica como um sistema binário:
  - Em 1854, George Boole, Professor de Matemática da Universidade de Cork (Irlanda), publicou o livro:

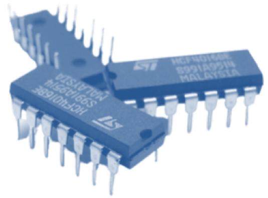
    *"An Investigation on The Laws of Thought, on which are founded the Mathematical Theories of Logic and Probabilities"*

  - Este trabalho, mais tarde refinado por Jevons (1869, 1890), Peirce (1880), Schröder (1890) e Huntingdon (1904), considera um sistema lógico <u>binário</u>, i.e., com dois objetos que se podem designar por:

**sim-não**, **verdadeiro-falso**, ou ainda **1-0**

TÉCNICO LISBOA

# Álgebra de Boole



## • Operações Básicas:

- Boole define ainda três operações básicas: AND, OR, NOT.
- Considere-se duas variáveis booleanas: x,y $\in$ {0,1},
  i.e., x,y $\in$ {falso,verdadeiro}

| AND (Produto lógico) | | |
|---|---|---|
| **X** | **Y** | **X · Y** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

O resultado é verdadeiro se X for verdadeiro **E** **(AND)** Y for verdadeiro

| OR (Soma lógica) | | |
|---|---|---|
| **X** | **Y** | **X+Y** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

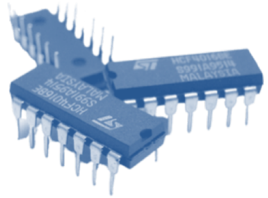
O resultado é verdadeiro se X for verdadeiro **OU** **(OR)** Y for verdadeiro

| NOT (Complemento) | |
|---|---|
| **X** | $\overline{\mathbf{X}}$ |
| 0 | 1 |
| 1 | 0 |

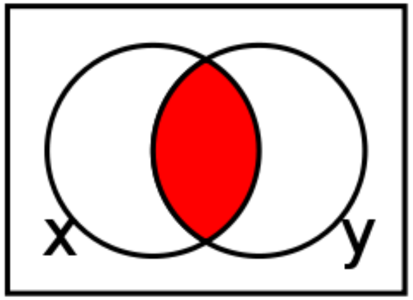
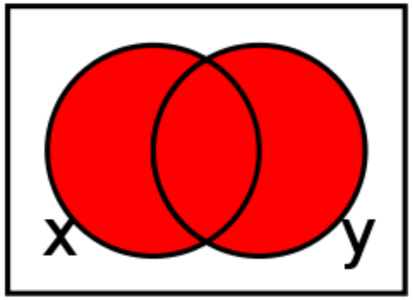
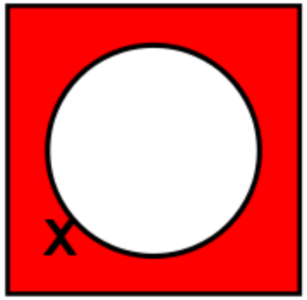
Negação **(NOT)** da afirmação.

TÉCNICO LISBOA

# Álgebra de Boole

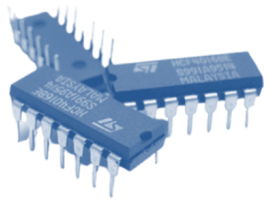


- Operações Básicas:
  - Boole define ainda três operações básicas: AND, OR, NOT.
  - Considere-se duas variáveis booleanas: $x,y \in \{0,1\}$,
    i.e., $x,y \in$ {falso,verdadeiro}

| AND<br>**(Produto lógico)** | OR<br>**(Soma lógica)** | NOT<br>**(Complemento)** |
|---|---|---|
|  |  |  |
| O resultado é verdadeiro se X for verdadeiro **E (AND)** Y for verdadeiro | O resultado é verdadeiro se X for verdadeiro **OU (OR)** Y for verdadeiro | Negação **(NOT)** da afirmação. |

TÉCNICO LISBOA

# Álgebra de Boole



- Álgebra de Boole binária:
  - A extensão ao trabalho de George Boole por Jevons (1869, 1890), Peirce (1880), Schröder (1890) e Huntingdon (1904), define:

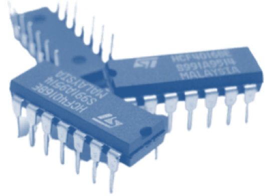
Uma **Álgebra de Boole binária** é um sistema algébrico $B_2$ = (A={0,1}, . ,+, ¯) formado por um conjunto gerador A e por duas operações binárias, . , +, designadas por produto lógico e soma lógica, e por uma operação designada por complemento, ¯ , tal que:

**Propriedade de Fecho:**

$$\forall_{x,y \in A}\ (x \cdot y \in A) \land (x + y \in A) \land (\bar{x} \in A)$$

*O resultado da aplicação de uma ou mais operações básicas sobre o conjunto gerador A, é um valor binário pertencente ao espaço do conjunto gerador A.*

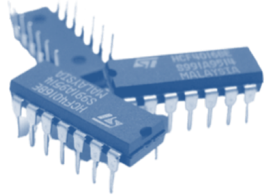
# Álgebra de Boole



- Propriedades básicas:
  - Considere-se as variáveis booleanas: x,y,z $\in$ A

| | | |
|---|---|---|
| **Identidade** | $x + 0 = x$ | $x \cdot 1 = x$ |
| **Idempotência** | $x + x = x$ | $x \cdot x = x$ |
| **Aniquilação** | $x + 1 = 1$ | $x \cdot 0 = 0$ |
| **Opostos** | $x + \overline{x} = 1$ | $x \cdot \overline{x} = 0$ |
| **Dupla negação** | $\overline{\overline{x}} = x$ | |
| **Comutatividade** | $x + y = y + x$ | $x \cdot y = y \cdot x$ |
| **Associatividade** | $x + (y + z) = (x + y) + z$ | $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ |
| **Distributividade** | $x \cdot (y + z) = x \cdot y + x \cdot z$ | $x + (y \cdot z) = (x + y) \cdot (x + z)$ |
| **DeMorgan** | $\overline{x + y} = \overline{x} \cdot \overline{y}$ | $\overline{x \cdot y} = \overline{x} + \overline{y}$ |
| **Adjacência** | $x \cdot y + x \cdot \overline{y} = x$ | $(x + y) \cdot (x + \overline{y}) = x$ |

# Álgebra de Boole

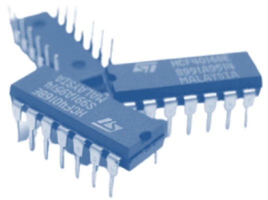


- Princípio da dualidade:
  - Qualquer expressão válida numa álgebra de Boole tem uma expressão dual, também válida nessa álgebra, que se obtém por troca do símbolo operatório + com o símbolo operatório • e do limite universal 0 com o limite universal 1.

Exemplo:

$x \bullet 1 = x$ é a expressão dual de $x + 0 = x$

# Álgebra de Boole



- Exemplo de aplicação das propriedades básicas (Propriedade da Absorção)

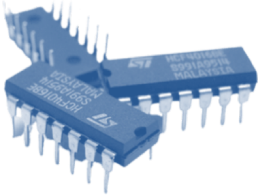
$$x \cdot (x + y) = x$$

$$\begin{aligned} x \cdot (x + y) &= (x + 0) \cdot (x + y) \\ &= x + (0 \cdot y) \\ &= x \end{aligned}$$

$$x + (x \cdot y) = x$$

$$\begin{aligned} x + (x \cdot y) &= (x \cdot 1) + (x \cdot y) \\ &= x \cdot (1 + y) \\ &= x \end{aligned}$$

# Álgebra de Boole



- Exemplo de aplicação das propriedades básicas (Propriedade do Consenso)

$$xy + yz + \bar{x}z = xy + \bar{x}z$$

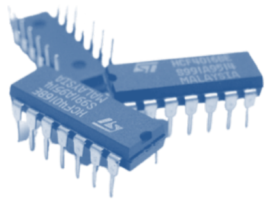
$$\begin{aligned} xy + yz + \bar{x}z &= xy + (x + \bar{x})yz + \bar{x}z \\ &= \underbrace{xy + xyz}_{\textit{Prop. Absorção}} + \underbrace{\bar{x}yz + \bar{x}z}_{\textit{Prop. Absorção}} \\ &= \quad xy \quad + \quad \bar{x}z \\ &= xy + \bar{x}z \end{aligned}$$

$$(x + y).(y + z).(\bar{x} + z) = (x + y).(\bar{x} + z)$$

$$\begin{aligned} &(x + y).(y + z).(\bar{x} + z) \\ &= (x + y).[(x.\bar{x}) + y + z].(\bar{x} + z) \\ &= \underbrace{(x + y).(x + y + z)}_{\textit{Prop. Absorção}}.\underbrace{(\bar{x} + y + z).(\bar{x} + z)}_{\textit{Prop. Absorção}} \\ &= \quad (x + y) \quad . \quad (\bar{x} + z) \\ &= (x + y).(\bar{x} + z) \end{aligned}$$

TÉCNICO LISBOA

# Álgebra de Boole



- Demonstração das leis de DeMorgan

$$\overline{x+y} = \overline{x} \cdot \overline{y}$$

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

Verificação por Tabelas da Verdade

| $x$ | $y$ | $x+y$ | $\overline{x+y}$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

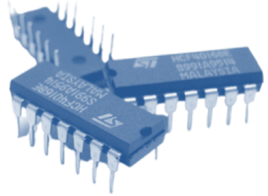
| $x$ | $y$ | $\overline{x}$ | $\overline{y}$ | $\overline{x} \cdot \overline{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |

- Generalização para n variáveis:

$$\overline{x_1 + x_2 + \ldots + x_n} = \overline{x}_1 . \overline{x}_2 \ldots \overline{x}_n$$

$$\overline{x_1 . x_2 \ldots x_n} = \overline{x}_1 + \overline{x}_2 + \ldots + \overline{x}_n$$

# Álgebra de Boole



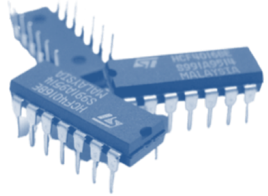
- Aplicação sucessiva das leis de DeMorgan

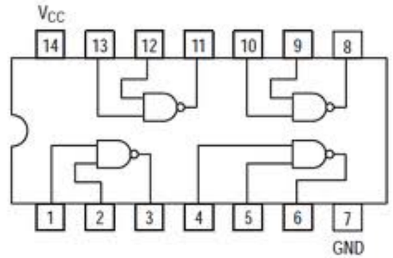
  - Exemplo:

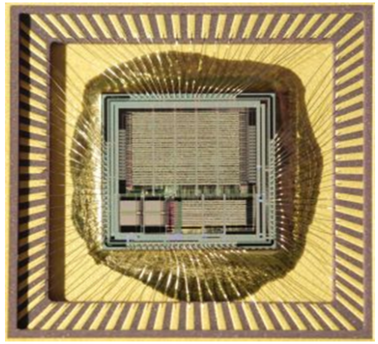
$$
\begin{aligned}
\overline{a\,.\,(b+z\,.\,(x+\overline{a}))} &= \overline{a}+\overline{(b+z\,.\,(x+\overline{a}))}\\
&= \overline{a}+\left(\overline{b}\,.\,\overline{(z\,.\,(x+\overline{a}))}\right)\\
&= \overline{a}+\left(\overline{b}\,.\,\left(\overline{z}+\overline{(x+\overline{a})}\right)\right)\\
&= \overline{a}+\left(\overline{b}\,.\,\left(\overline{z}+\left(\overline{x}\,.\,a\right)\right)\right)\\
&= \overline{a}+\overline{b}\,.\,\left(\overline{z}+\overline{x}\,.\,a\right)
\end{aligned}
$$

TÉCNICO LISBOA

# Portas Lógicas



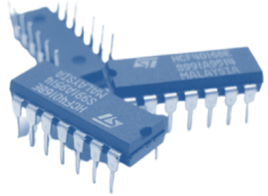



**Sumário:**

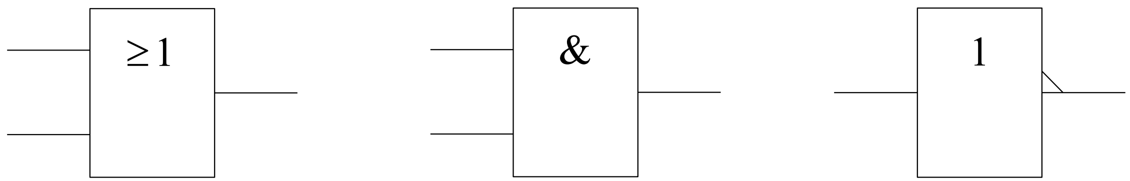
- Portas OR, AND e NOT
- Portas NOR e NAND
- Portas XOR e XNOR
- Saídas tri-state

# Portas Lógicas



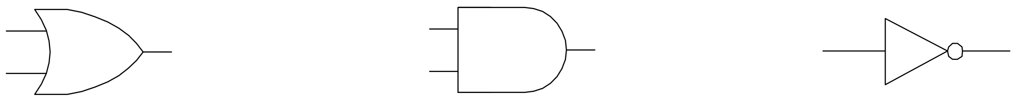
- Portas Lógicas:
  - Na prática, os circuitos digitais baseiam-se na Álgebra de Boole, sendo implementados a partir de um conjunto de portas lógicas base.

Simbologia (IEC 617):

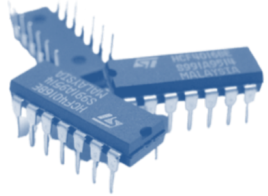



OR AND NOT

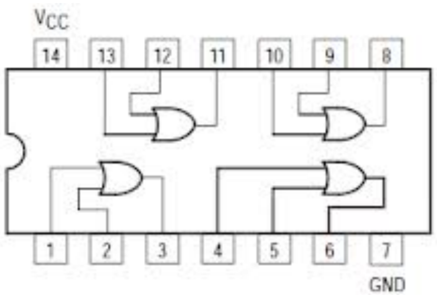
Simbologia (ANSI/IEEE):



# Portas Lógicas



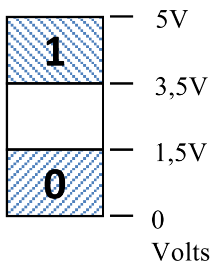
- Componentes Discretos:
  - Exemplos:



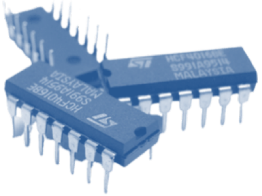



| Componentes Discretos | |
|---|---|
| Dispositivo | Função |
| ’00 | 4 NAND2 |
| ’02 | 4 NOR2 |
| ’04 | 6 NOT |
| ’08 | 4 AND2 |
| ’20 | 2 NAND4 |
| ’21 | 2 AND4 |
| ’27 | 3 NOR3 |
| ’30 | 1 NAND8 |
| ’32 | 4 OR2 |
| ‘126 | 4 Buffers Tri-State |
| ‘136 | 4 XOR2 |

  - Nas tecnologias mais comuns, o circuito lógico distingue 2 intervalos distintos de tensão, os quais são interpretados como ‘um’ ou ‘zero’



# Portas Lógicas



- Função Booleana (exemplo):

$$f = \overline{a}.b + c$$

$\bar{a}.b$ e **c** são os **termos** da função.

$\bar{a}$, $b$ e $c$ são os **literais**.

- Circuito Lógico:

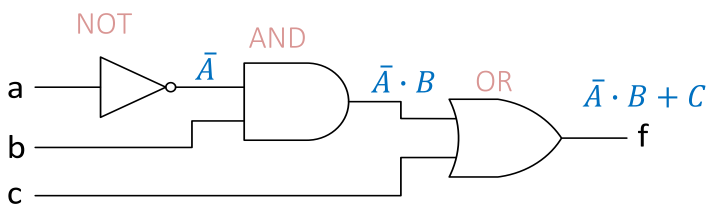
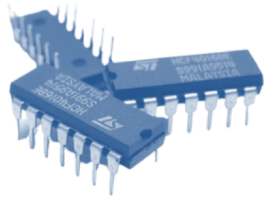



Tabela da Verdade

| a | b | c | ā b | f |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |

TÉCNICO LISBOA

# Portas Lógicas



- Existem 16 funções de 2 variáveis Booleanas:

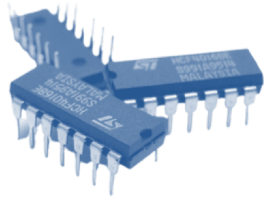
| x | y | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

- Funções já conhecidas:

$$f_0(x,y) = 0$$
$$f_3(x,y) = \bar{x}$$
$$f_5(x,y) = \bar{y}$$
$$f_8(x,y) = x\,.\,y \quad \textbf{AND}$$

$$f_{10}(x,y) = y$$
$$f_{12}(x,y) = x$$
$$f_{14}(x,y) = x + y \quad \textbf{OR}$$
$$f_{15}(x,y) = 1$$

# Portas Lógicas



- Funções NOR e NAND:

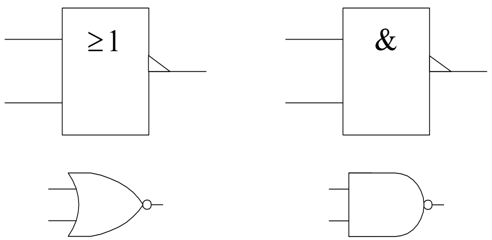
$$f_1(x,y) = \bar{x} . \bar{y} = \overline{x+y} \quad \textbf{NOR}$$

$$f_7(x,y) = \bar{x} + \bar{y} = \overline{x . y} \quad \textbf{NAND}$$

Funcionam como uma porta OR (ou uma porta AND) seguida de uma porta NOT

| x | y | $f_1$ | $f_7$ |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

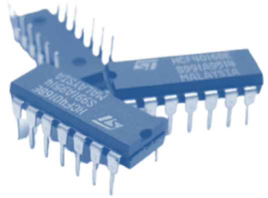
- Simbologia:




NOR NAND

Nas tecnologias mais comuns (p.ex. CMOS), as portas NOR e NAND (portas inversoras) requerem <u>menos</u> transístores que as portas OR e AND (portas não inversoras).

De facto, as portas OR e AND é que são habitualmente realizadas com uma porta NOR ou NAND seguida de uma porta NOT.



# Portas Lógicas

- Funções OU-EXCLUSIVO:

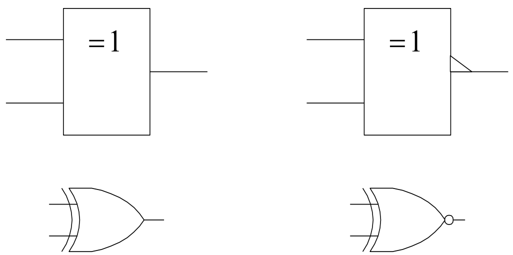
XOR: $f_6(x,y) = x \oplus y = \bar{x} \cdot y + x \cdot \bar{y}$

XNOR: $f_9(x,y) = x \odot y = x \cdot y + \bar{x} \cdot \bar{y} = \overline{x \oplus y}$

XOR é verdadeira se uma e apenas uma das 2 entradas for verdadeira.

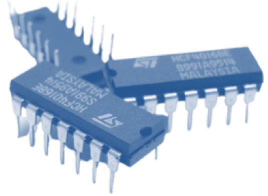
| x | y | $f_6$ | $f_9$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

- Simbologia:

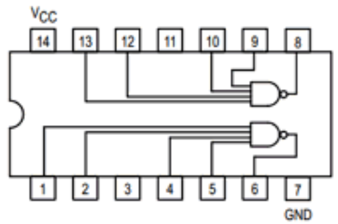



XOR XNOR

# Portas Lógicas

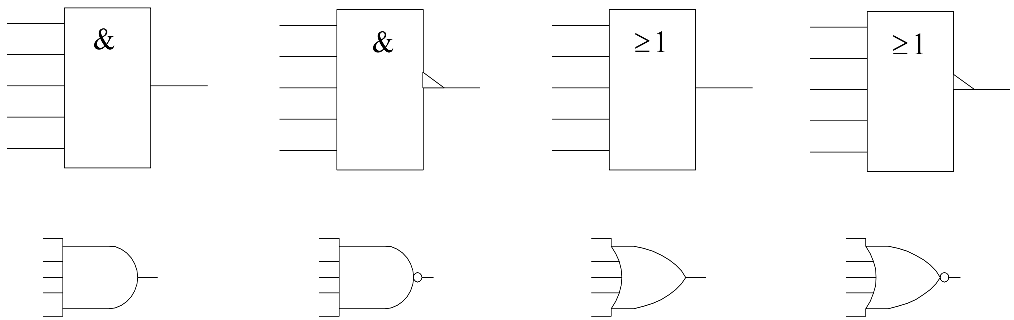


- Portas com mais de 2 entradas:
  - As operações AND e OR (e consequentemente as portas NAND e NOR) são facilmente generalizáveis para N-entradas.
  - Uma porta AND de N entradas tem a saída a 1 sse todas as entradas estiverem a 1.
  - Uma porta OR de N entradas tem a saída a 1 se pelo menos uma entrada estiver a 1.

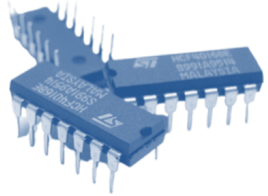



74LS20

  - Simbologia:

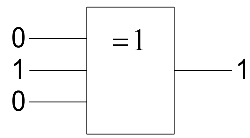


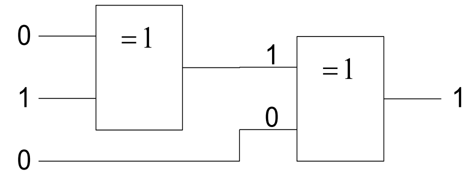
# Portas Lógicas



- Caso particular: OU-EXCLUSIVO com mais de 2 entradas:

$$x \oplus y \oplus z = (x \oplus y) \oplus z$$

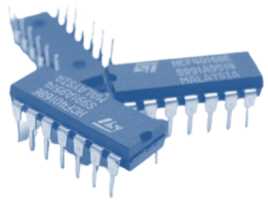



  - A porta XOR de 3 entradas é verdadeira se uma e apenas uma das 3 entradas for igual a 1, ou se as 3 entradas forem iguais a 1.

$$x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_N = (((x_1 \oplus x_2) \oplus x_3) \oplus \cdots) \oplus x_N$$

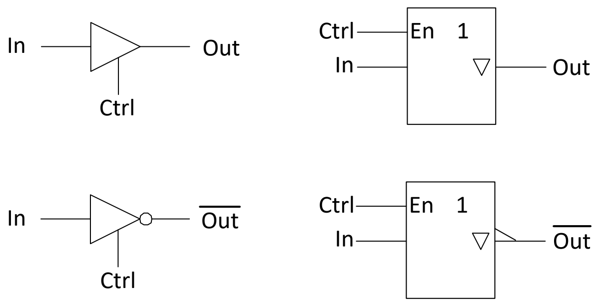
  - A porta XOR de <u>N entradas</u> é verdadeira se o número de entradas iguais a 1 for <u>ímpar</u>.
  - De facto, e embora usada genericamente, a designação OU-exclusivo só é estritamente correta para a função de 2 variáveis.



- Buffer de três estados (tri-state):
  - Dispositivo que, para além de uma entrada e uma saída de dados, dispõe ainda de uma entrada de controlo que define o comportamento da saída:

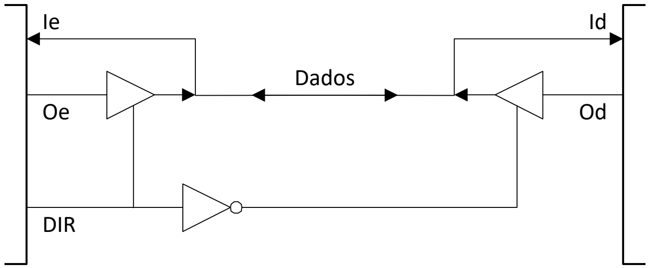
    Controlo = H → o valor da saída é igual ao valor que se apresenta na entrada de dados;

    Controlo = L → o porto de saída fica em alta impedância, i.e., a saída fica (eletricamente) desligada.

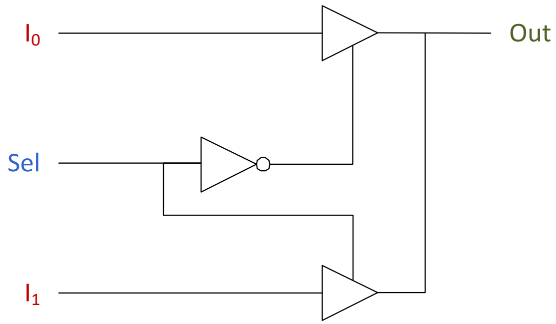


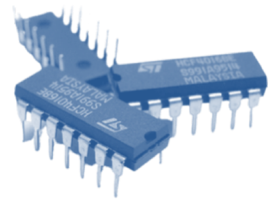

| Ctrl | In | Out | $\overline{\text{Out}}$ |
|---|---|---|---|
| L | X | Desligada | Desligada |
| H | L | L | H |
| H | H | H | L |

- Buffer de três estados: exemplos de aplicação
  - Linha Bidirecional:




  - Seleção de Sinais:



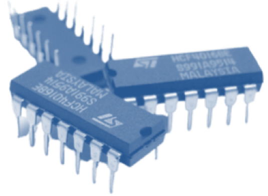


# Circuitos com portas NAND e NOR

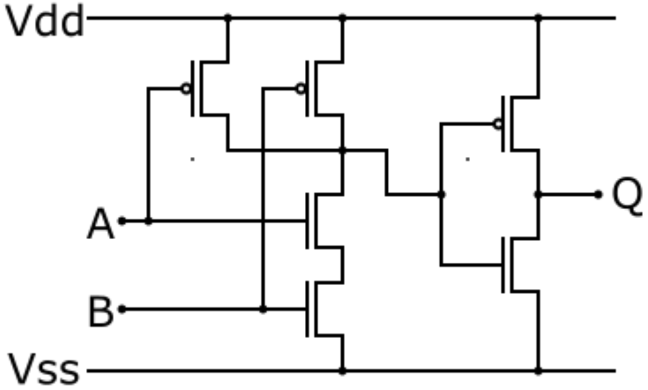
**Sumário:**

- Lei de DeMorgan
- Circuitos com portas NAND
- Circuitos com portas NOR

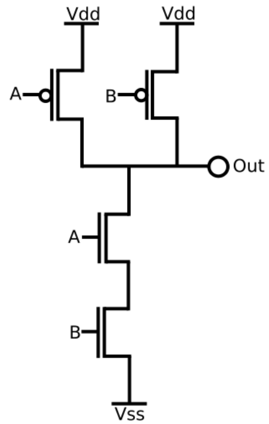
# Circuitos com portas NAND e NOR



- Circuitos com portas NAND:
  - Nalgumas tecnologias (p.ex. TTL e CMOS) as portas NAND são as portas mais simples (portanto mais baratas), pelo que é vantajosa a realização de circuitos só com NANDs.



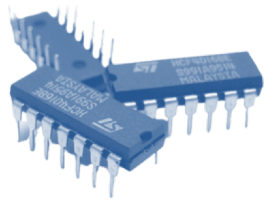

Porta AND (CMOS)



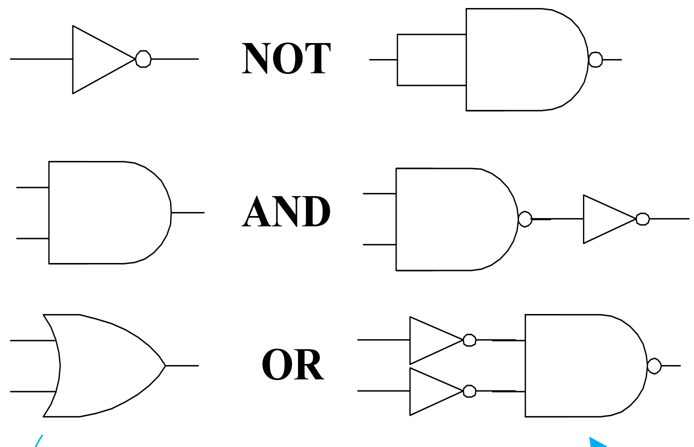

Porta NAND (CMOS)

TÉCNICO LISBOA

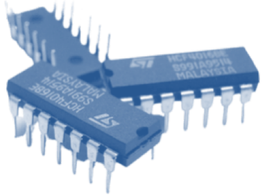
# Circuitos com portas NAND e NOR



- Circuitos com portas NAND:
  - A porta NAND é considerada uma porta universal porque qualquer circuito digital pode ser realizado apenas com portas NAND.
  - Qualquer função booleana é realizável apenas com portas NAND por substituição direta das operações NOT, AND e OR.
  - A operação NOT é normalmente considerada em sentido lato, como uma NAND de 1 entrada.




$$A + B = \overline{\overline{A + B}} = \overline{\bar{A} \cdot \bar{B}}$$



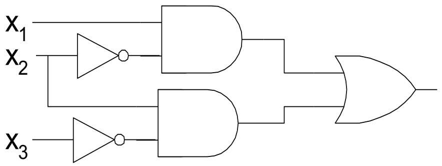
# Circuitos com portas NAND e NOR

- Circuitos com portas NAND (cont.):
  - Uma função representada na forma de uma soma de produtos pode ser transformada numa forma diretamente realizável apenas com portas NAND por simples aplicação da lei de DeMorgan.

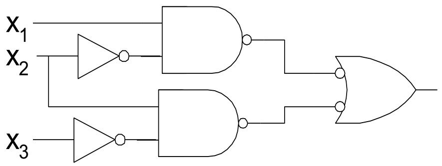
    Exemplo:

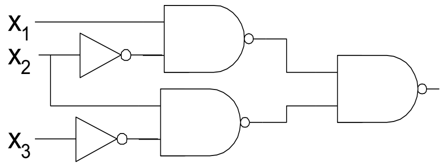
$$f = x_1\bar{x}_2 + \bar{x}_3x_2 = \overline{\overline{x_1\bar{x}_2 + \bar{x}_3x_2}} = \overline{\overline{x_1\bar{x}_2} \cdot \overline{\bar{x}_3x_2}}$$

$$= \left(x_1 \ nand \ \bar{x}_2\right) nand \left(\bar{x}_3 \ nand \ x_2\right)$$

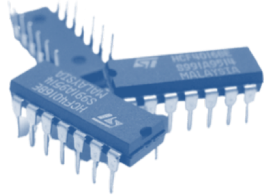



A estrutura do circuito mantém-se inalterada.



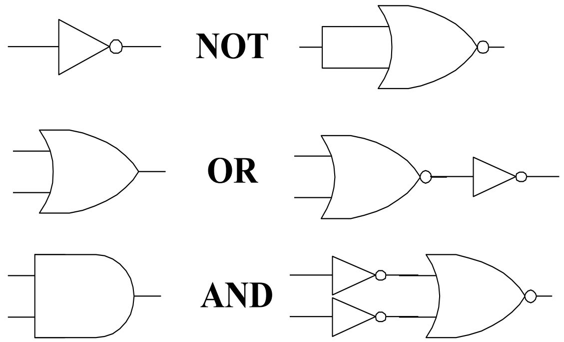
# Circuitos com portas NAND e NOR



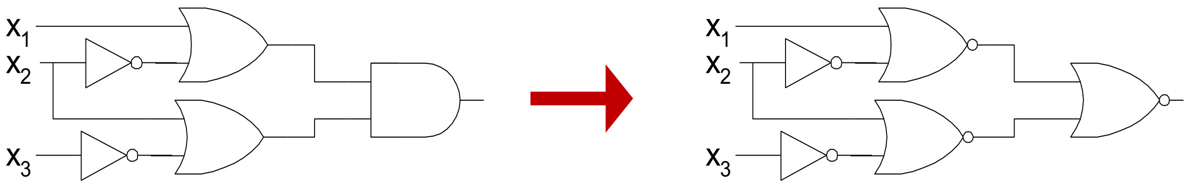
- Circuitos com portas NOR:

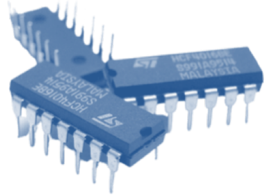
  Dual:

  - Qualquer circuito pode ser realizado apenas com portas NOR.
  - No caso de a função estar representada como um produto de somas, a transformação mantém a estrutura.

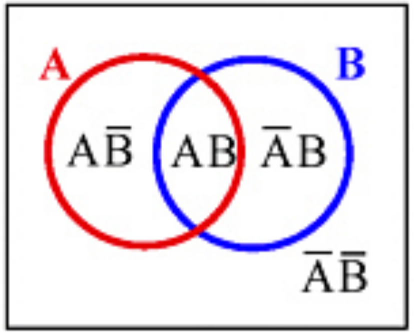



$$g=(x_1+\bar{x}_2)\cdot(\bar{x}_3+x_2)=\overline{\overline{(x_1+\bar{x}_2)\cdot(\bar{x}_3+x_2)}}=\overline{\overline{(x_1+\bar{x}_2)}+\overline{(\bar{x}_3+x_2)}}$$
$$=(x_1\ nor\ \bar{x}_2)\ nor\ (\bar{x}_3\ nor\ x_2)$$





# Funções Lógicas

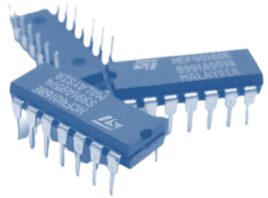



| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**Sumário:**

- Representações normalizadas:
  - Mintermos e Soma de produtos
  - Maxtermos e Produto de somas
- Funções incompletamente especificadas.

# Funções Lógicas



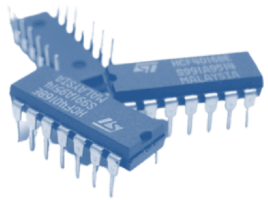
- **Representação Normalizada: Soma de Produtos**
  - Designa-se por forma normal **disjuntiva** de uma função booleana simples completamente especificada, $y=f(x_1,x_2,\ldots,x_n)$, uma expressão lógica representativa da função com a estrutura de uma **soma de produtos**.
  - Se cada parcela for constituída por um produto lógico envolvendo **n** literais distintos, diz-se que a função se encontra representada na primeira forma canónica ou **forma canónica disjuntiva**.

Exemplos:

$$f(x_1,x_2,x_3) = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$$ ← Forma não canónica

$$f(x_1,x_2,x_3) = x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$$ ← Forma canónica

# Funções Lógicas



## • Mintermos:

- Designa-se por **mintermo** (também produto canónico, implicante canónico ou termo minimal) um termo de produto em que <u>todas as variáveis aparecem exactamente uma vez</u>, complementadas ou não.

**Mintermos para 3 variáveis**

| $x_3$ | $x_2$ | $x_1$ | mintermo | |
|---|---|---|---|---|
| 0 | 0 | 0 | $\bar{x}_3.\bar{x}_2.\bar{x}_1$ | $m_0$ |
| 0 | 0 | 1 | $\bar{x}_3.\bar{x}_2.x_1$ | $m_1$ |
| 0 | 1 | 0 | $\bar{x}_3.x_2.\bar{x}_1$ | $m_2$ |
| 0 | 1 | 1 | $\bar{x}_3.x_2.x_1$ | $m_3$ |
| 1 | 0 | 0 | $x_3.\bar{x}_2.\bar{x}_1$ | $m_4$ |
| 1 | 0 | 1 | $x_3.\bar{x}_2.x_1$ | $m_5$ |
| 1 | 1 | 0 | $x_3.x_2.\bar{x}_1$ | $m_6$ |
| 1 | 1 | 1 | $x_3.x_2.x_1$ | $m_7$ |

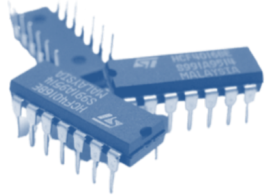
Um **mintermo** representa exactamente uma combinação das variáveis binárias na tabela de verdade da função.

Uma tabela de n variáveis tem $2^n$ mintermos.

Cada mintermo é também designado por $m_i$ em que o índice *i* indica o número decimal equivalente à combinação binária por ele representada.

O mintermo vale 1 para a combinação representada, e 0 para todas as outras.

# Funções Lógicas



## • Representação Normalizada: Produto de Somas

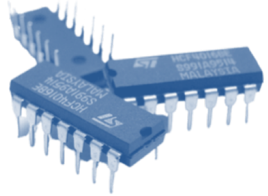
- Designa-se por forma normal **conjuntiva** de uma função booleana simples completamente especificada, $y=f(x_1,x_2,...,x_n)$, uma expressão lógica representativa da função com a estrutura de um **produto de somas**.
- Se cada parcela for constituída por uma soma lógica envolvendo **n** literais distintos, diz-se que a função se encontra representada na segunda forma canónica ou **forma canónica conjuntiva**.

Exemplos:

$$f(x_1,x_2,x_3)=(x_1+x_2)\cdot(\overline{x_1}+\overline{x_2}+\overline{x_3})$$ ← Forma <u>não</u> canónica

$$f(x_1,x_2,x_3)=(x_1+x_2+x_3)\cdot(x_1+x_2+\overline{x_3})\cdot(\overline{x_1}+\overline{x_2}+\overline{x_3})$$ ← Forma canónica

# Funções Lógicas



- ## Maxtermos:
  - Designa-se por maxtermo (também soma canónica, implicado canónico ou termo maximal) um termo de soma em que todas as variáveis aparecem exactamente uma vez, complementadas ou não.

**Maxtermos para 3 variáveis**

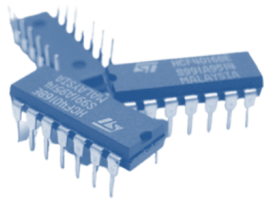
| $x_3$ | $x_2$ | $x_1$ | maxtermo | |
|---|---|---|---|---|
| 0 | 0 | 0 | $x_3 + x_2 + x_1$ | $M_0$ |
| 0 | 0 | 1 | $x_3 + x_2 + \bar{x}_1$ | $M_1$ |
| 0 | 1 | 0 | $x_3 + \bar{x}_2 + x_1$ | $M_2$ |
| 0 | 1 | 1 | $x_3 + \bar{x}_2 + \bar{x}_1$ | $M_3$ |
| 1 | 0 | 0 | $\bar{x}_3 + x_2 + x_1$ | $M_4$ |
| 1 | 0 | 1 | $\bar{x}_3 + x_2 + \bar{x}_1$ | $M_5$ |
| 1 | 1 | 0 | $\bar{x}_3 + \bar{x}_2 + x_1$ | $M_6$ |
| 1 | 1 | 1 | $\bar{x}_3 + \bar{x}_2 + \bar{x}_1$ | $M_7$ |

Um **maxtermo** representa exactamente uma combinação das variáveis binárias na tabela de verdade da função.

Uma tabela de n variáveis tem $2^n$ maxtermos.

Cada maxtermo é também designado por $M_i$ em que o índice *i* indica o número decimal equivalente à combinação binária por ele representada.

O maxtermo vale 0 para a combinação representada, e 1 para todas as outras.



## • Mintermos e Maxtermos:

- O **mintermo** corresponde a uma função ≠ 0 com o número mínimo de 1's na tabela da verdade.

Exemplo:

$$f = \bar{x}_3 . x_2 . x_1$$

| $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

# Funções Lógicas

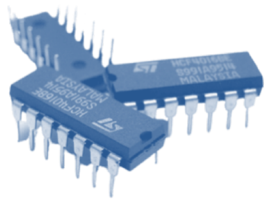
- **Mintermos e Maxtermos:**
  - O **maxtermo** corresponde a uma função ≠ 1 com o número máximo de 1's na tabela da verdade.

Exemplo:

$$f = x_3 + \overline{x}_2 + \overline{x}_1$$

| $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

# Funções Lógicas



## • Mintermos e Maxtermos:

- Um mintermo e um maxtermo com o mesmo índice são complementos um do outro:

$$m_j = \overline{M}_j$$

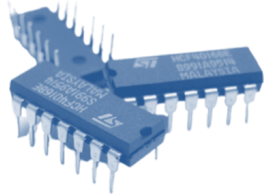
| $x_3$ | $x_2$ | $x_1$ | mintermo | | maxtermo | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | $\bar{x}_3.\bar{x}_2.\bar{x}_1$ | $m_0$ | $x_3 + x_2 + x_1$ | $M_0$ |
| 0 | 0 | 1 | $\bar{x}_3.\bar{x}_2.x_1$ | $m_1$ | $x_3 + x_2 + \bar{x}_1$ | $M_1$ |
| 0 | 1 | 0 | $\bar{x}_3.x_2.\bar{x}_1$ | $m_2$ | $x_3 + \bar{x}_2 + x_1$ | $M_2$ |
| 0 | 1 | 1 | $\bar{x}_3.x_2.x_1$ | $m_3$ | $x_3 + \bar{x}_2 + \bar{x}_1$ | $M_3$ |
| 1 | 0 | 0 | $x_3.\bar{x}_2.\bar{x}_1$ | $m_4$ | $\bar{x}_3 + x_2 + x_1$ | $M_4$ |
| 1 | 0 | 1 | $x_3.\bar{x}_2.x_1$ | $m_5$ | $\bar{x}_3 + x_2 + \bar{x}_1$ | $M_5$ |
| 1 | 1 | 0 | $x_3.x_2.\bar{x}_1$ | $m_6$ | $\bar{x}_3 + \bar{x}_2 + x_1$ | $M_6$ |
| 1 | 1 | 1 | $x_3.x_2.x_1$ | $m_7$ | $\bar{x}_3 + \bar{x}_2 + \bar{x}_1$ | $M_7$ |

Exemplo:

$$\begin{aligned} m_3 &= \bar{x}_3.x_2.x_1 \\ &= \overline{\overline{\bar{x}_3.x_2.x_1}} \\ &= \overline{x_3 + \bar{x}_2 + \bar{x}_1} \\ &= \overline{M}_3 \end{aligned}$$

TÉCNICO LISBOA

# Funções Lógicas



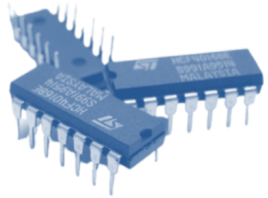
- **Tabela de Verdade ↔ Soma de Produtos**
  - Uma função booleana pode ser expressa algebricamente como uma <u>soma de produtos</u> directamente a partir da tabela de verdade.
  - A soma inclui todos os mintermos para os quais a função vale 1.

Exemplo:

$$f(x_3, x_2, x_1) = \sum m(0,1,3,5,7)$$

$$\begin{aligned} f(x_3, x_2, x_1) = {} & \bar{x}_3.\bar{x}_2.\bar{x}_1 & \leftarrow m_0 \\ & + \bar{x}_3.\bar{x}_2.x_1 & \leftarrow m_1 \\ & + \bar{x}_3.x_2.x_1 & \leftarrow m_3 \\ & + x_3.\bar{x}_2.x_1 & \leftarrow m_5 \\ & + x_3.x_2.x_1 & \leftarrow m_7 \end{aligned}$$

| $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |



# Soma de Produtos ↔ Produto de Somas

- Conversão entre formas canónicas: o produto de somas utiliza os maxtermos correspondentes aos mintermos não utilizados na soma de produtos.
- É equivalente a aplicar a lei de DeMorgan ao complemento da função.

Exemplo:

| $x_3$ | $x_2$ | $x_1$ | $f$ | $\overline{f}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

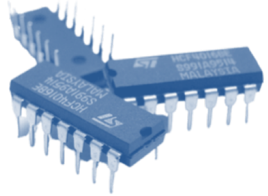
$$f(x_3,x_2,x_1) = m_0 + m_1 + m_3 + m_5 + m_7$$

$$\overline{f(x_3,x_2,x_1)} = m_2 + m_4 + m_6$$

$$f(x_3,x_2,x_1) = \overline{m_2 + m_4 + m_6} = \overline{m}_2.\overline{m}_4.\overline{m}_6$$
$$= M_2.M_4.M_6$$

$$f = \overline{\overline{x}_3.x_2.\overline{x}_1 + x_3.\overline{x}_2.\overline{x}_1 + x_3.x_2.\overline{x}_1}$$
$$= \left(\overline{\overline{x}_3.x_2.\overline{x}_1}\right).\left(\overline{x_3.\overline{x}_2.\overline{x}_1}\right).\left(\overline{x_3.x_2.\overline{x}_1}\right)$$
$$= \left(x_3 + \overline{x}_2 + x_1\right).\left(\overline{x}_3 + x_2 + x_1\right).\left(\overline{x}_3 + \overline{x}_2 + x_1\right)$$

# Funções Lógicas



## • Tabela de Verdade ↔ Produto de Somas

- Uma função booleana pode ser expressa algebricamente, como um produto de somas, directamente a partir da tabela de verdade.
- O produto inclui todos os maxtermos para os quais a função vale 0.

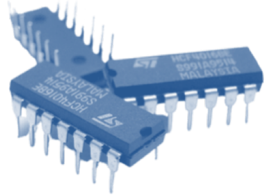
Exemplo:

$$f(x_3,x_2,x_1)=\prod M(2,4,6)$$

$$\begin{aligned} f(x_3,x_2,x_1)=&(x_3+\bar{x}_2+x_1) && \leftarrow M_2 \\ &.(\bar{x}_3+x_2+x_1) && \leftarrow M_4 \\ &.(\bar{x}_3+\bar{x}_2+x_1) && \leftarrow M_6 \end{aligned}$$

| $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

# Funções Lógicas



- ## Funções Incompletamente Especificadas

**Exemplo:** Função que detecta se o resultado de um lançamento de um dado (número no intervalo [1,6]) é múltiplo de 3.

| $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | X |

A função toma o valor 'X' (às vezes também representado por '-') para cada uma das combinações das entradas que nunca ocorrerão.

**Realidade Física:** 'X' não existe, apenas existem '0' ou '1'.

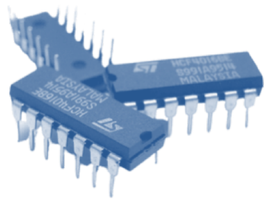
X – *"don´t care"*: não nos preocupamos com o comportamento do circuito para os valores fora do intervalo, portanto podemos <u>escolher</u> para cada 'X' o valor mais adequado entre '0' ou '1'.

Representação:

$$f(x_3, x_2, x_1) = \sum m(3,6) + \sum m_d(0,7) = m_3 + m_6 + m_{d0} + m_{d7}$$

$$f(x_3, x_2, x_1) = \prod M(1,2,4,5) \cdot \prod M_d(0,7) = M_1 \cdot M_2 \cdot M_4 \cdot M_5 \cdot M_{d0} \cdot M_{d7}$$

# Funções Lógicas



- **Funções Incompletamente Especificadas (cont.)**

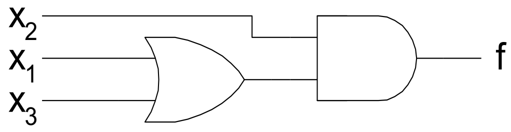
## Exemplo:

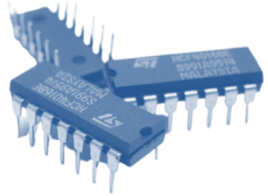
| $x_3$ | $x_2$ | $x_1$ | $f$ | $\to g$ |
|---|---|---|---|---|
| 0 | 0 | 0 | X | $\to$ 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | X | $\to$ 1 |

**Estratégia**: para cada 'X' escolhemos '0' ou '1' de acordo com os objectivos do projecto (habitualmente, maior simplificação).

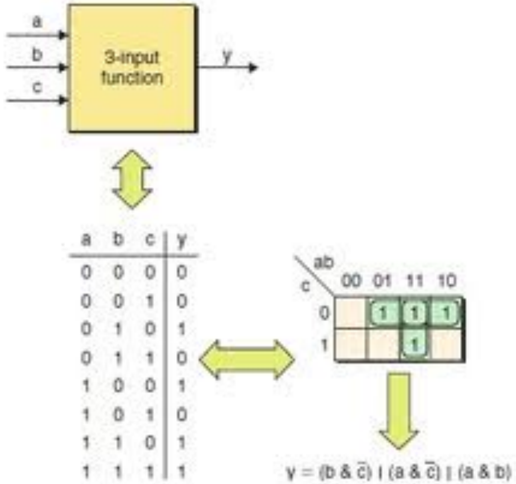
Neste caso, a solução mais simples corresponde a substituir o primeiro 'X' por '0' e o segundo por '1'   (veremos depois porquê…)

$$
\begin{aligned}
f \to g(x_3, x_2, x_1) &= \sum m(3,6,7) = \prod M(0,1,2,4,5) \\
&= x_2 x_1 + x_3 x_2 \\
&= x_2 (x_1 + x_3)
\end{aligned}
$$





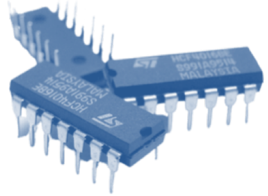
# Simplificação e minimização de funções lógicas




**Sumário:**

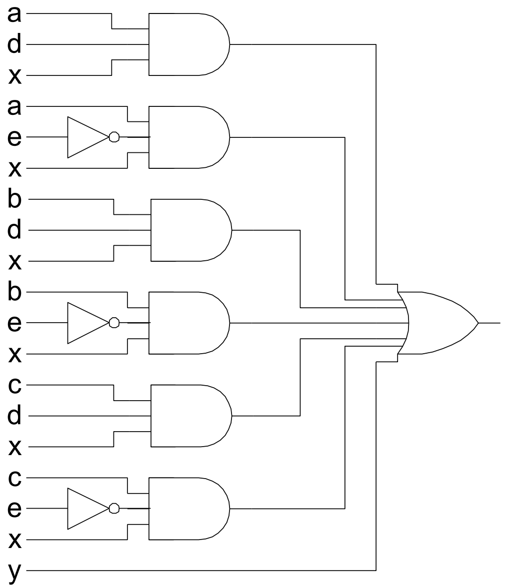
- Teorema da adjacência
- Representação de funções no quadro de Karnaugh
- Agrupamentos de uns e zeros
- Método de minimização de Karnaugh

# Funções Lógicas

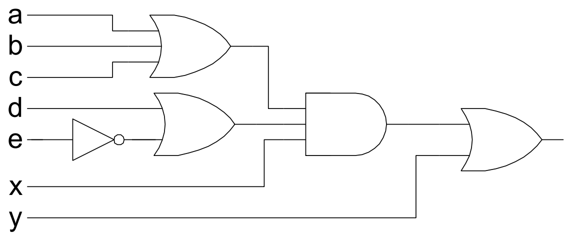


- Simplificação algébrica
  - Exemplo 1:

$$
\begin{aligned}
f &= adx + a\bar{e}x + bdx + b\bar{e}x + cdx + c\bar{e}x + y \\
&= \left(ad + a\bar{e} + bd + b\bar{e} + cd + c\bar{e}\right)x + y \\
&= \left(\left(a+b+c\right)d + \left(a+b+c\right)\bar{e}\right)x + y \\
&= \left(\left(a+b+c\right)\left(d+\bar{e}\right)\right)x + y \\
&= \left(a+b+c\right)\cdot\left(d+\bar{e}\right)\cdot x + y
\end{aligned}
$$

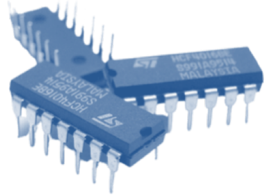



**Realização a 2 níveis**
**(soma de produtos)**



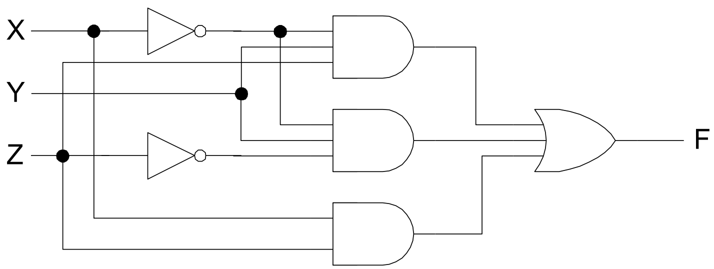

**Realização Multinível**

# Funções Lógicas

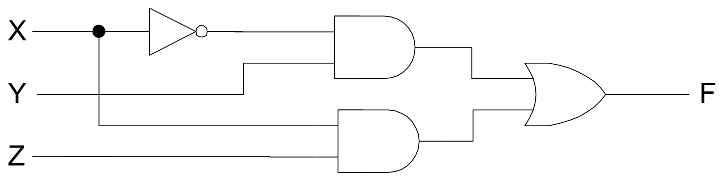


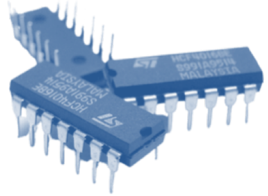
- Simplificação algébrica
  - Exemplo 2:

$$f = \overline{x}yz + \overline{x}y\overline{z} + xz$$



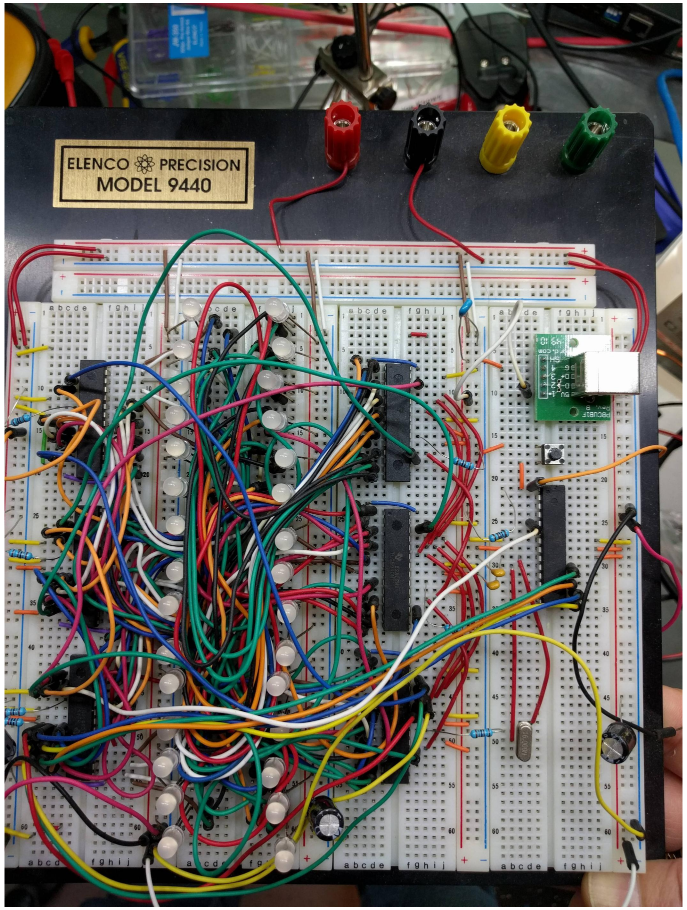

$$\begin{aligned} f &= \overline{x}yz + \overline{x}y\overline{z} + xz \\ &= \overline{x}y\left(z + \overline{z}\right) + xz \\ &= \overline{x}y.1 + xz \\ &= \overline{x}y + xz \end{aligned}$$



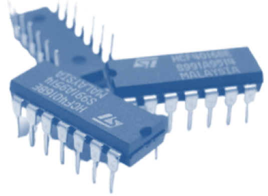


# Necessidade de simplificação

- Simplificação algébrica:
  - A simplificação e manipulação algébrica das funções lógicas tem vários benefícios:
    - Permite reduzir a complexidade de circuitos, o que leva a uma redução no número de erros na montagem do circuito.
    - Permite reduzir o tempo de propagação dos sinais ao longo do circuito de cálculo (ex.: processadores mais rápidos)
    - Permite reduzir a potência consumida (ex: processadores energeticamente mais eficientes)



# Minimização Algébrica



- Simplificação Algébrica pelo Teorema da Adjacência
  - Um termo com n literais tem n adjacentes possíveis
  - Exemplo:

| $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

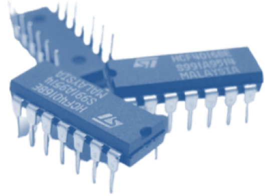
$m_0$ tem 3 adjacentes possíveis, mas neste exemplo apenas $m_1$ também vale 1.

$$m_0 = \bar{x}_3.\bar{x}_2.\bar{x}_1 \rightarrow \begin{cases} \bar{x}_3.\bar{x}_2.x_1 = m_1 \\ \bar{x}_3.x_2.\bar{x}_1 = m_2 \\ x_3.\bar{x}_2.\bar{x}_1 = m_4 \end{cases}$$

$m_0$ apenas pode ser simplificado com $m_1$.

$$m_0 + m_1 = \bar{x}_3.\bar{x}_2.\left(x_1 + \bar{x}_1\right) = \bar{x}_3.\bar{x}_2$$

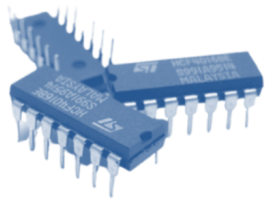
# Minimização Algébrica



- Simplificação Algébrica pelo Teorema da Adjacência
  - Exemplo:

| $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

| | **Adjacentes** | **Obs.** |
|---|---|---|
| $m_0$ | $m_1$ | $m_0$ só pode ser simplificado com $m_1$ |
| $m_1$ | $m_0$ , $m_3$ , $m_5$ | $m_1$ pode ser simplificado com $m_0$ ou com $m_3$ ou com $m_5$ |
| $m_3$ | $m_1$ , $m_7$ | $m_3$ pode ser simplificado com $m_1$ ou com $m_7$ |
| $m_5$ | $m_1$ , $m_7$ | $m_5$ pode ser simplificado com $m_1$ ou com $m_7$ |
| $m_7$ | $m_3$ , $m_5$ | $m_7$ pode ser simplificado com $m_3$ ou com $m_5$ |



# Minimização Algébrica

- Simplificação Algébrica pelo Teorema da Adjacência
  - Exemplo:

| $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

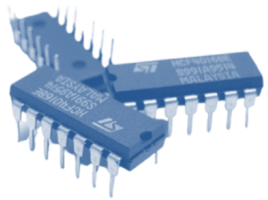
$$\begin{aligned} f = & \left(m_0 + m_1\right) \quad \leftarrow \text{essencial} \\ & + \left(m_3 + m_7\right) \\ & + \left(m_1 + m_5\right) \end{aligned}$$

$$\begin{aligned} f = & \, \bar{x}_3 \cdot \bar{x}_2 \\ & \left.\begin{aligned} + x_2 \cdot x_1 \\ + \bar{x}_2 \cdot x_1 \end{aligned}\right\} \quad \textit{adjacentes} \end{aligned}$$

$$f = \bar{x}_3 \cdot \bar{x}_2 + x_1$$

TÉCNICO LISBOA

# Minimização Algébrica



- Re-ordenação da Tabela
  - Exemplo:

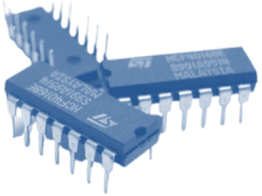
| | $x_3$ | $x_2$ | $x_1$ | $f$ |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 1 |
| $m_1$ | 0 | 0 | 1 | 1 |
| $m_3$ | 0 | 1 | 1 | 1 |
| $m_2$ | 0 | 1 | 0 | 0 |
| $m_6$ | 1 | 1 | 0 | 0 |
| $m_7$ | 1 | 1 | 1 | 1 |
| $m_5$ | 1 | 0 | 1 | 1 |
| $m_4$ | 1 | 0 | 0 | 0 |

Os termos em linhas consecutivas diferem apenas de 1 bit – **código de Gray**

Deste modo, grande parte dos termos adjacentes ficam representados em linhas contíguas, o que facilita a identificação de adjacências.

(Não é habitualmente usada, porque se preferem os quadros a 2 dimensões → ver a seguir…)

# Quadro de Karnaugh



- Quadro de Karnaugh
  - Reordenação da tabela de verdade em 2 dimensões.

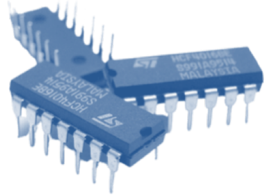
  Exemplo:

| | $x_3$ | $x_2$ | $x_1$ | f |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 1 |
| $m_1$ | 0 | 0 | 1 | 1 |
| $m_3$ | 0 | 1 | 1 | 1 |
| $m_2$ | 0 | 1 | 0 | 0 |
| $m_6$ | 1 | 1 | 0 | 0 |
| $m_7$ | 1 | 1 | 1 | 1 |
| $m_5$ | 1 | 0 | 1 | 1 |
| $m_4$ | 1 | 0 | 0 | 0 |

| $x_3$ \ $x_2 x_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $^{0}$ 1 | $^{1}$ 1 | $^{3}$ 1 | $^{2}$ 0 |
| 1 | $^{4}$ 0 | $^{5}$ 1 | $^{7}$ 1 | $^{6}$ 0 |

***Maurice Karnaugh***
4/Out/1924,NY

Os **termos adjacentes** ficam representados em linhas/colunas contíguas.

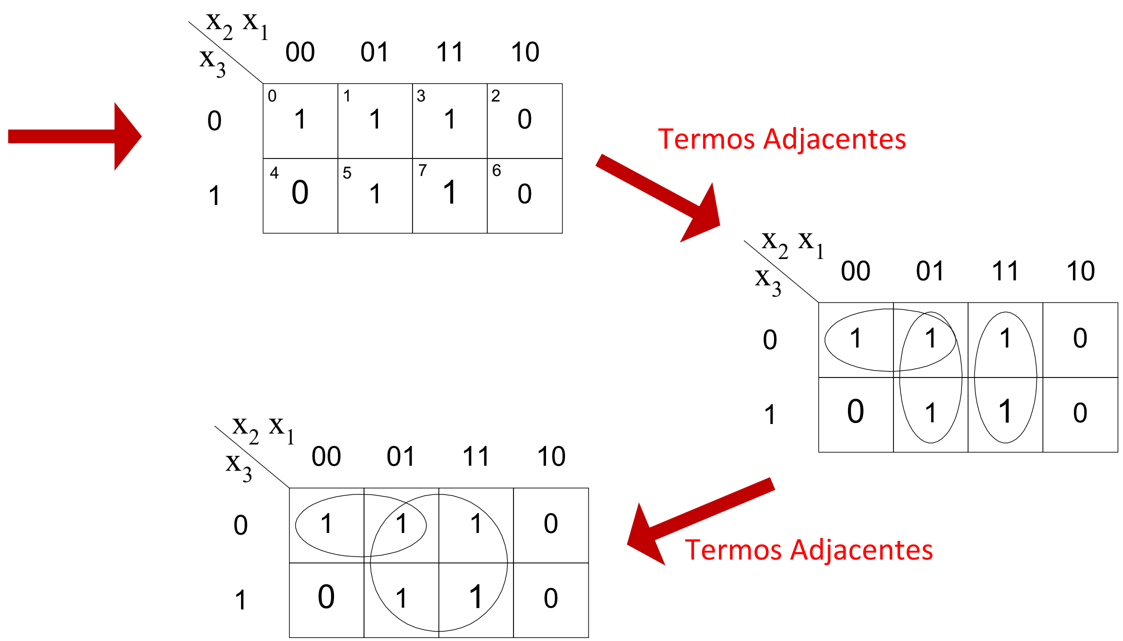


# Quadro de Karnaugh

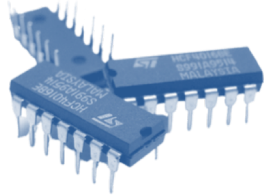
- Quadro de Karnaugh
  - Os termos adjacentes ficam representados em linhas/colunas contíguas.

Exemplo:

| | $x_3$ | $x_2$ | $x_1$ | $f$ |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 1 |
| $m_1$ | 0 | 0 | 1 | 1 |
| $m_3$ | 0 | 1 | 1 | 1 |
| $m_2$ | 0 | 1 | 0 | 0 |
| $m_6$ | 1 | 1 | 0 | 0 |
| $m_7$ | 1 | 1 | 1 | 1 |
| $m_5$ | 1 | 0 | 1 | 1 |
| $m_4$ | 1 | 0 | 0 | 0 |



# Quadro de Karnaugh



- Identificação dos Termos no Quadro de Karnaugh
  - Exemplos:

| $x_3$ \ $x_2 x_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |

O termo é 1 quando: $x_3=0$; e $x_2=0$; e ($x_1=0$ ou $x_1=1$)

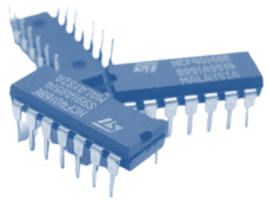
ou seja: $\bar{x}_3.\bar{x}_2.\underbrace{\left(\bar{x}_1+x_1\right)}_{simplificados} \rightarrow \bar{x}_3.\bar{x}_2$

| $x_3$ \ $x_2 x_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |

O termo é 1 quando: ($x_3=0$ ou $x_3=1$); e ($x_2=0$ ou $x_2=1$); e $x_1=1$

ou seja: $\underbrace{\left(\left(\bar{x}_3+x_3\right).\left(\bar{x}_2+x_2\right)\right)}_{simplificados}.x_1 \rightarrow x_1$

# Quadro de Karnaugh



- Representação de Funções – Q. de Karnaugh
  - Quadros de 3 Variáveis

| X \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $\bar{X}\bar{Y}\bar{Z}$ | $\bar{X}\bar{Y}Z$ | $\bar{X}YZ$ | $\bar{X}Y\bar{Z}$ |
| 1 | $X\bar{Y}\bar{Z}$ | $X\bar{Y}Z$ | $XYZ$ | $XY\bar{Z}$ |

Y

X

Z

f(X,Y,Z)

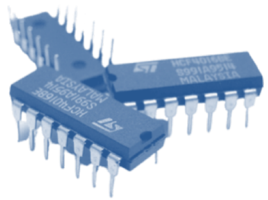
| X \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 1 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |

Exemplo:

$f(X,Y,Z) = \Sigma m(0,3,5,6)$

| X \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |

# Quadro de Karnaugh



- Representação de Funções – Q. de Karnaugh (Cont.)
  - Quadros de 4 Variáveis:
    - A mesma função pode ter representações diferentes, mas equivalentes, num Quadro de Karnaugh, pela simples alteração da localização das variáveis

f(W,X,Y,Z)

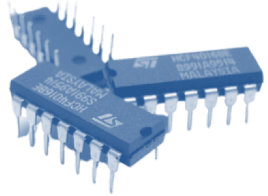
| WX \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 01 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |
| 11 | $m_{12}$ | $m_{13}$ | $m_{15}$ | $m_{14}$ |
| 10 | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ |

f(W,X,Y,Z)

| YZ \ WX | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $m_0$ | $m_4$ | $m_{12}$ | $m_8$ |
| 01 | $m_1$ | $m_5$ | $m_{13}$ | $m_9$ |
| 11 | $m_3$ | $m_7$ | $m_{15}$ | $m_{11}$ |
| 10 | $m_2$ | $m_6$ | $m_{14}$ | $m_{10}$ |

TÉCNICO LISBOA

# Quadro de Karnaugh



- Representação de Funções – Q. de Karnaugh (Cont.)
  - Quadros de N Variáveis
    - Um Quadro de Karnaugh de N variáveis é obtido pela duplicação de quadro de N-1 variáveis, devendo ser acrescentada a N-ésima variável e o correspondente eixo de simetria de modo a manter a representação das variáveis de forma reflectida.

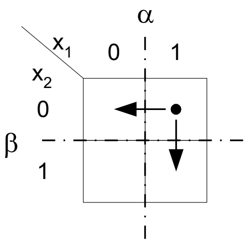
f(V,W,X,Y,Z)

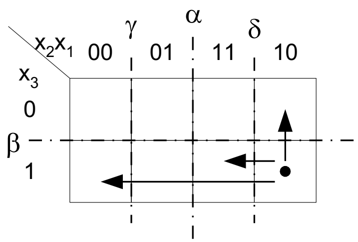
| V W \ XYZ | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | $m_0$ | $m_1$ | $m_3$ | $m_2$ | $m_6$ | $m_7$ | $m_5$ | $m_4$ |
| 01 | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ | $m_{14}$ | $m_{15}$ | $m_{13}$ | $m_{12}$ |
| 11 | $m_{24}$ | $m_{25}$ | $m_{27}$ | $m_{26}$ | $m_{30}$ | $m_{31}$ | $m_{29}$ | $m_{28}$ |
| 10 | $m_{16}$ | $m_{17}$ | $m_{19}$ | $m_{18}$ | $m_{22}$ | $m_{23}$ | $m_{21}$ | $m_{20}$ |

# AGRUPAMENTO DE MINTERMOS E MAXTERMOS

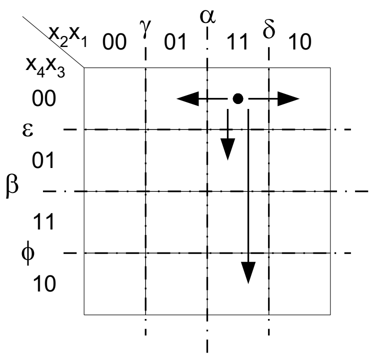
# Agrupamento de Mintermos e Maxtermos

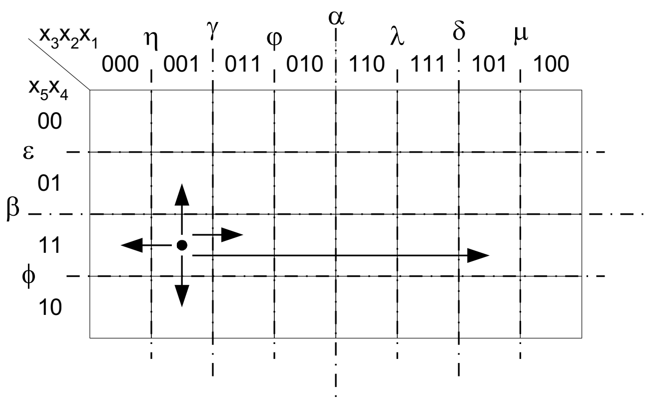
- Agrupamento de Mintermos e Maxtermos
  - Eixos de Simetria:



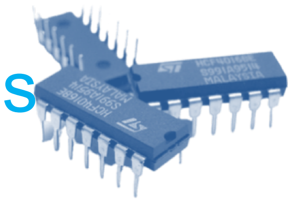




2 **quadrados** dizem-se **adjacentes** em termos lógicos quando apenas uma variável lógica altera o seu valor na representação desses quadrados.



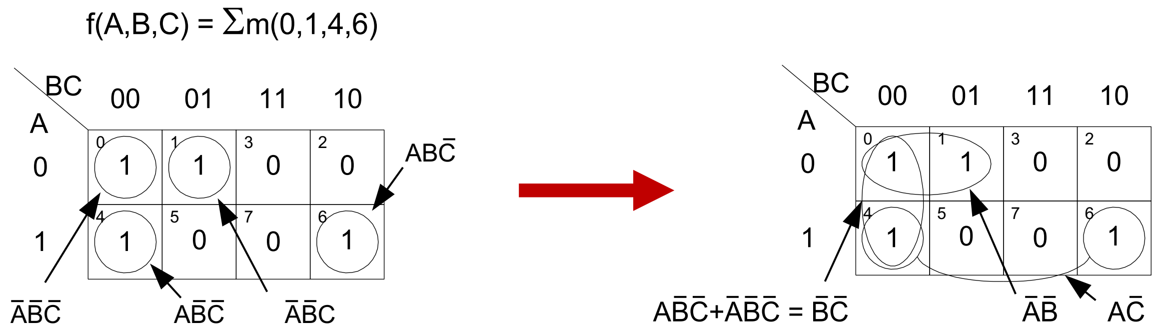

Num quadro de N variáveis, para cada quadrado existem <u>sempre</u> **N** outros adjacentes



# Agrupamento de Mintermos e Maxtermos

- Agrupamento de Mintermos e Maxtermos (cont.)
  - Um termo de produto diz-se um implicante da função sse essa função assume 1 para todos os mintermos que o constituem.

  Exemplos:

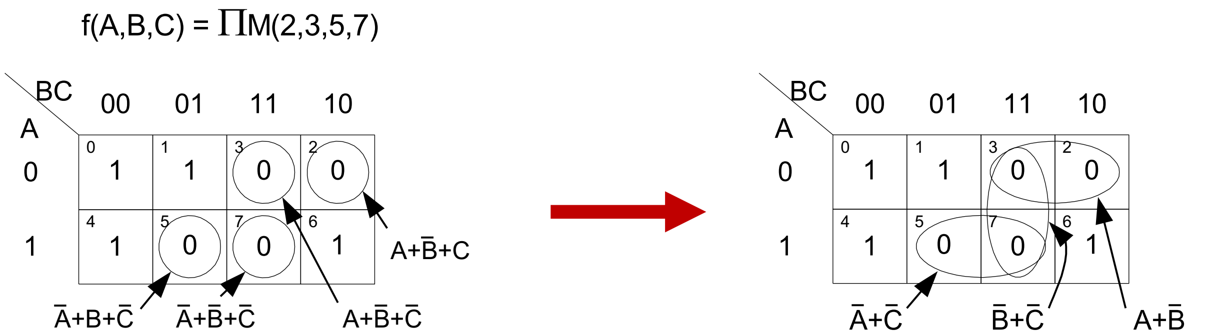



Agrupamentos de $2^n$ quadrados correspondem à eliminação de n literais

# Agrupamento de Mintermos e Maxtermos

- Agrupamento de Mintermos e Maxtermos (cont.)
  - Um termo de soma diz-se um implicado da função sse essa função assume 0 para todos os maxtermos que o constituem.

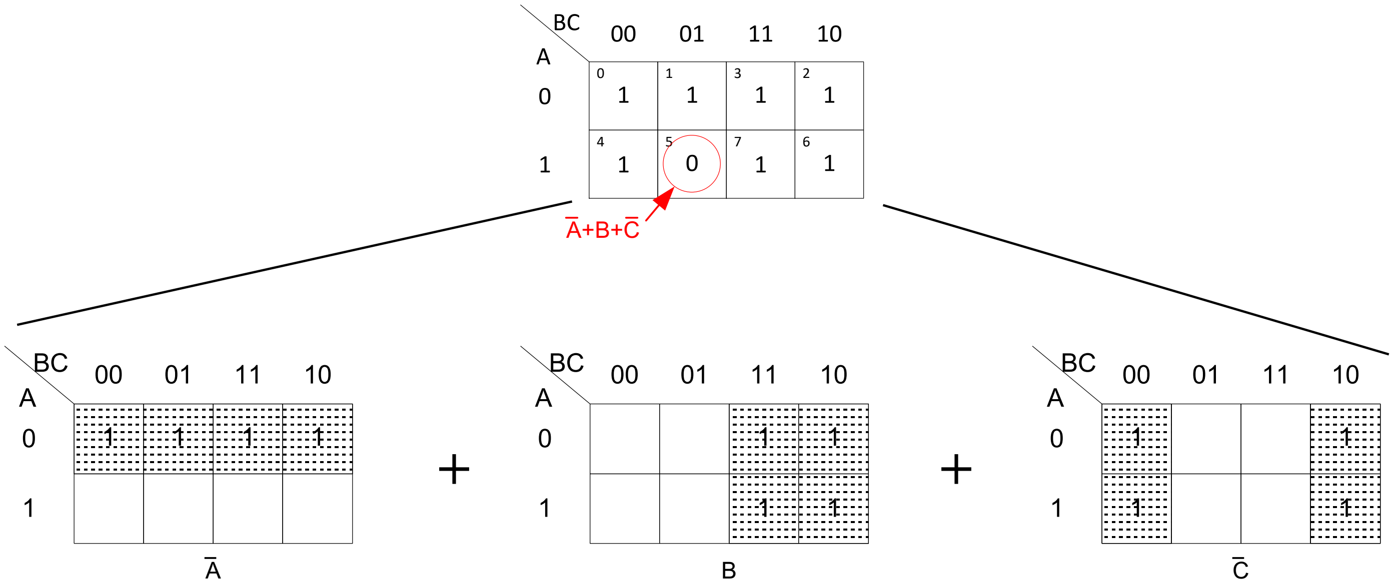
  Exemplos:



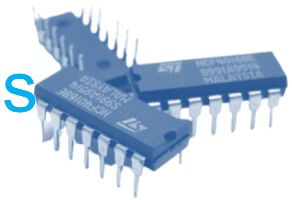

Agrupamentos de $2^n$ quadrados correspondem à eliminação de n literais

# Agrupamento de Mintermos e Maxtermos

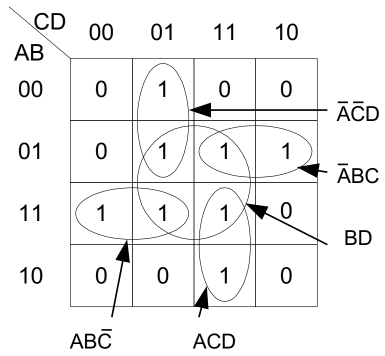
- Agrupamento de Mintermos e Maxtermos (cont.)
  - Exemplos da representação de Maxtermos:



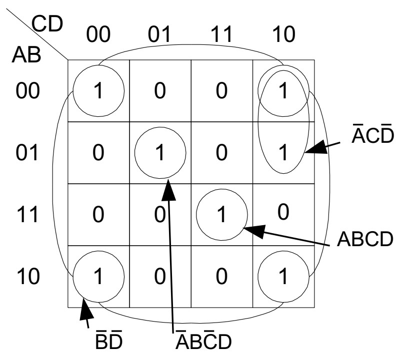
# Agrupamento de Mintermos e Maxtermos

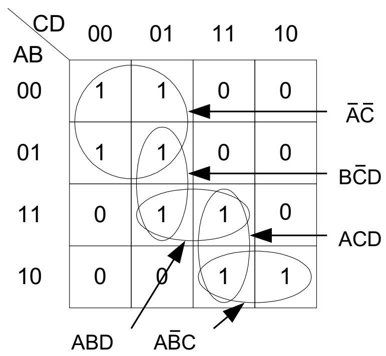


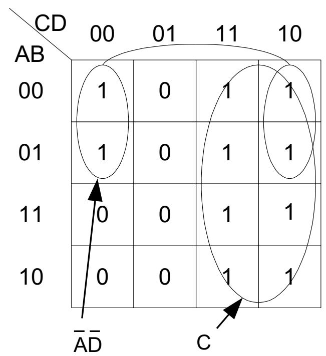
- Agrupamento de Mintermos e Maxtermos (cont.)
  - Um termo de produto diz-se um implicante primo se a remoção de um qualquer literal, desse termo de produto, resulta num termo de produto que não é um implicante da função.
  - Ou seja: implicante primo = maior agrupamento possível naquela região do quadro




Exemplos:

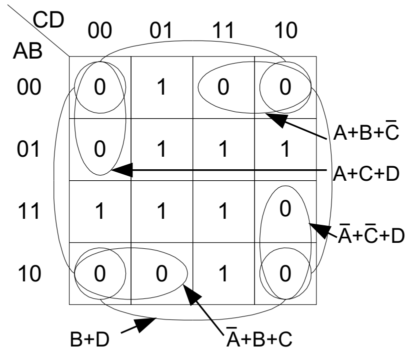


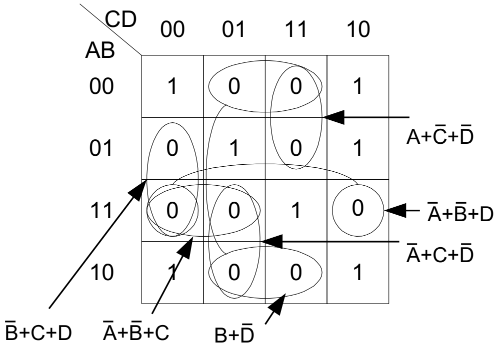



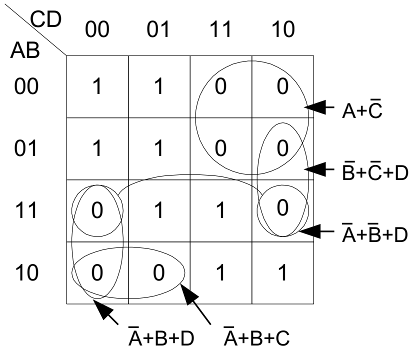
# Agrupamento de Mintermos e Maxtermos

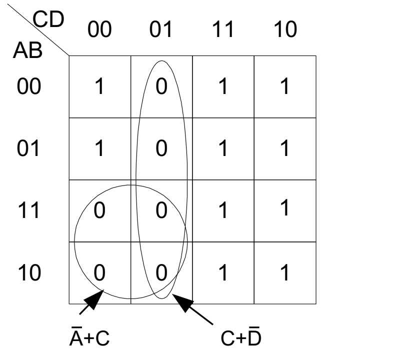
- Agrupamento de Mintermos e Maxtermos (cont.)
  - Um termo de soma diz-se um implicado primo se a remoção de um qualquer literal, desse termo de soma, resulta num termo de soma que não é um implicado da função.
  - <u>Ou seja</u>: implicado primo = maior agrupamento possível naquela região do quadro



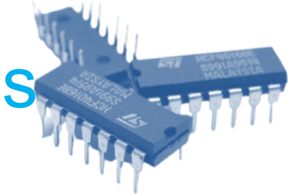

Exemplos:



# Agrupamento de Mintermos e Maxtermos



- Agrupamento de Mintermos e Maxtermos (cont.)
  - Um implicante primo de uma função diz-se implicante primo essencial se contém pelo menos um mintermo não contido em nenhum outro implicante primo.
  - Ou seja: este implicante é *obrigatório* para garantir a incorporação desse mintermo.

Exemplos:

**Implicantes Primos**

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 0 | 0 | 1 | 0 |

$\bar{A}\bar{C}D$, $\bar{A}BC$, $BD$, $AB\bar{C}$, $ACD$

**Implicantes Primos Essenciais**

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 0 | 0 | 1 | 0 |

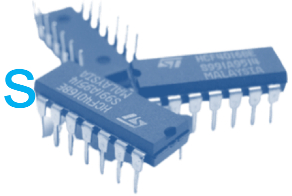
**Implicantes Primos**

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 0 | 1 |

**Implicantes Primos Essenciais**

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 0 | 1 |

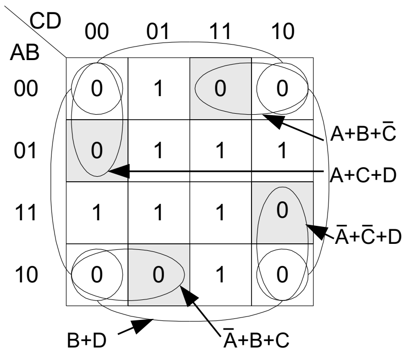
# Agrupamento de Mintermos e Maxtermos



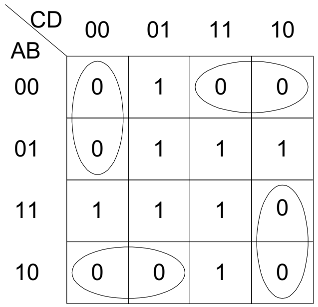
- Agrupamento de Mintermos e Maxtermos (cont.)
  - Um implicado primo de uma função diz-se implicado primo <u>essencial</u> se contém pelo menos um maxtermo não contido em nenhum outro implicado primo.
  - <u>Ou seja</u>: este implicado é *obrigatório* para garantir a incorporação desse maxtermo.

Exemplos:

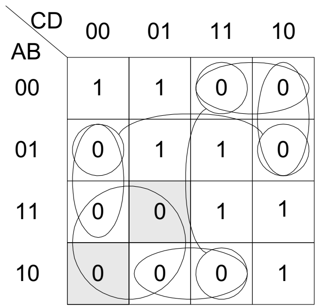
**Implicados Primos**



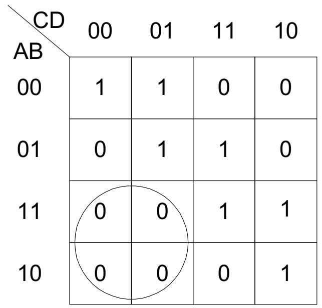

**Implicados Primos Essenciais**



**Implicados Primos**



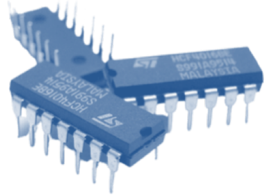
**Implicados Primos Essenciais**



# MÉTODO DE MINIMIZAÇÃO DE KARNAUGH

TÉCNICO LISBOA

# Minimização de Karnaugh



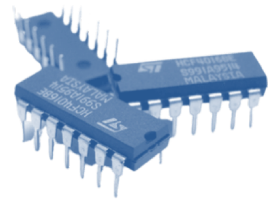
- Método de Minimização de Karnaugh
  - Algoritmo de Minimização:
    - O procedimento sistemático para a obtenção da expressão simplificada de uma função representada num quadro de Karnaugh corresponde à execução dos seguintes passos:

**Passo 1**: Identificação de todos os **implicantes/implicados primos <u>essenciais</u>**.

**Passo 2**: Determinação do **menor conjunto de implicantes/implicados <u>primos</u>** que contenham os mintermos/maxtermos não incluídos nos implicantes/implicados primos essenciais identificados no passo anterior.

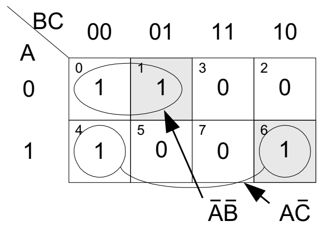
**Passo 3**: **Escrita da expressão simplificada** como soma/produto de todos os termos de produto/soma seleccionados nos passos 1 e 2.

# Minimização de Karnaugh



- Método de Minimização de Karnaugh (cont.)
  - Forma Normal/Mínima Disjuntiva: funções cuja simplificação utiliza apenas implicantes primos essenciais.

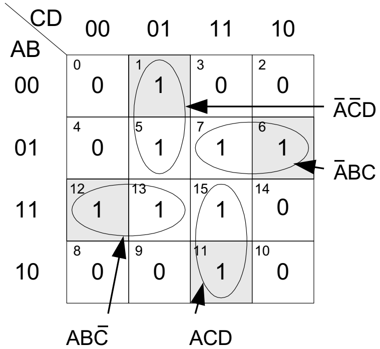
  Exemplos:




$$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C}$$

$$\Downarrow$$

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

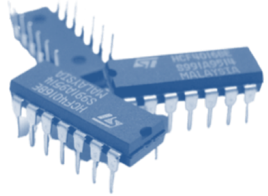



$$f(A,B,C,D) = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + \bar{A}BCD + \bar{A}BC\bar{D} + AB\bar{C}\bar{D} + AB\bar{C}D + ABCD + A\bar{B}CD$$

$$\Downarrow$$

$$f(A,B,C,D) = AB\bar{C} + ACD + \bar{A}BC + \bar{A}\bar{C}D$$

# Minimização de Karnaugh



- Método de Minimização de Karnaugh (cont.)
  - Exemplos:

$$f(A,B,C,D) = \Sigma m(0,1,5,7,10,14,15)$$
$$= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + \bar{A}BCD + A\bar{B}C\bar{D} + ABC\bar{D} + ABCD$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | (0) 1 | (1) 1 | (3) 0 | (2) 0 |
| 01 | (4) 0 | (5) 1 | (7) 1 | (6) 0 |
| 11 | (12) 0 | (13) 0 | (15) 1 | (14) 1 |
| 10 | (8) 0 | (9) 0 | (11) 0 | (10) 1 |

O conjunto de implicantes primos essenciais é único.

Soluções alternativas

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | (0) 1 | (1) 1 | (3) 0 | (2) 0 |
| 01 | (4) 0 | (5) 1 | (7) 1 | (6) 0 |
| 11 | (12) 0 | (13) 0 | (15) 1 | (14) 1 |
| 10 | (8) 0 | (9) 0 | (11) 0 | (10) 1 |

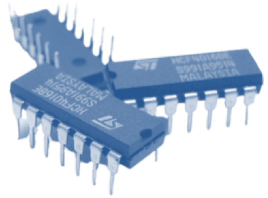
$$f(A,B,C,D) = \bar{A}\bar{B}\bar{C} + \bar{A}BD + ABC + AC\bar{D}$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | (0) 1 | (1) 1 | (3) 0 | (2) 0 |
| 01 | (4) 0 | (5) 1 | (7) 1 | (6) 0 |
| 11 | (12) 0 | (13) 0 | (15) 1 | (14) 1 |
| 10 | (8) 0 | (9) 0 | (11) 0 | (10) 1 |

$$f(A,B,C,D) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C}D + BCD + AC\bar{D}$$

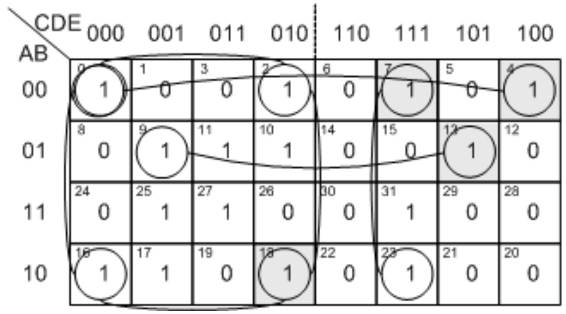
O conjunto de implicantes primos não essenciais que completam a expressão simplificada oferece várias alternativas.

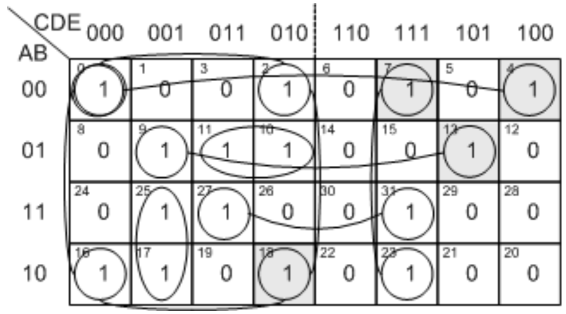
# Minimização de Karnaugh



- Método de Minimização de Karnaugh (cont.)
  - Exemplos:

| AB \ CDE | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | (0) 1 | (1) 0 | (3) 0 | (2) 1 | (6) 0 | (7) 1 | (5) 0 | (4) 1 |
| 01 | (8) 0 | (9) 1 | (11) 1 | (10) 1 | (14) 0 | (15) 0 | (13) 1 | (12) 0 |
| 11 | (24) 0 | (25) 1 | (27) 1 | (26) 0 | (30) 0 | (31) 1 | (29) 0 | (28) 0 |
| 10 | (16) 1 | (17) 1 | (19) 0 | (18) 1 | (22) 0 | (23) 1 | (21) 0 | (20) 0 |

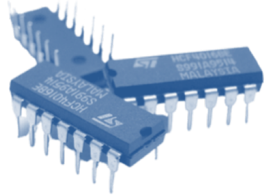




**Expressão Original:**
- 15 Termos de Produto de 5 Literais

**Expressão Simplificada:**
- 6 Termos de Produto de 4 Literais
- 1 Termo de Produto de 3 Literais

# Minimização de Karnaugh



- Método de Minimização de Karnaugh (cont.)
  - Forma Normal/Mínima Conjuntiva:

$$f(A,B,C,D) = \prod M(2,3,4,6,8,9,11,12,13)$$
$$= (A+B+\bar{C}+D).\ (A+B+\bar{C}+\bar{D}).\ (A+\bar{B}+C+D).$$
$$(A+\bar{B}+\bar{C}+D).\ (\bar{A}+B+C+D).\ (\bar{A}+B+C+\bar{D}).$$
$$(\bar{A}+B+\bar{C}+\bar{D}).\ (\bar{A}+\bar{B}+C+D).\ (\bar{A}+\bar{B}+C+\bar{D})$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | (0) 1 | (1) 1 | (3) 0 | (2) 0 |
| 01 | (4) 0 | (5) 1 | (7) 1 | (6) 0 |
| 11 | (12) 0 | (13) 0 | (15) 1 | (14) 1 |
| 10 | (8) 0 | (9) 0 | (11) 0 | (10) 1 |

Soluções alternativas

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | (0) 1 | (1) 1 | (3) 0 | (2) 0 |
| 01 | (4) 0 | (5) 1 | (7) 1 | (6) 0 |
| 11 | (12) 0 | (13) 0 | (15) 1 | (14) 1 |
| 10 | (8) 0 | (9) 0 | (11) 0 | (10) 1 |

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

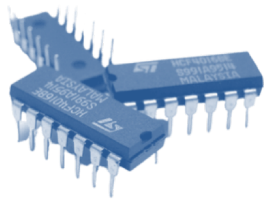
| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | (0) 1 | (1) 1 | (3) 0 | (2) 0 |
| 01 | (4) 0 | (5) 1 | (7) 1 | (6) 0 |
| 11 | (12) 0 | (13) 0 | (15) 1 | (14) 1 |
| 10 | (8) 0 | (9) 0 | (11) 0 | (10) 1 |

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

O conjunto de implicados primos essenciais é único.

O conjunto de implicados primos não essenciais que completam a expressão simplificada oferece várias alternativas.

# Minimização de Karnaugh



- Método de Minimização de Karnaugh (cont.)
  - Algoritmo de Minimização (Funções Incompletamente Especificadas):
    - O procedimento sistemático para a obtenção da expressão simplificada de uma função representada num quadro de Karnaugh corresponde execução dos seguintes passos:

**Passo 1**: Identificação de todos os **implicantes/implicados primos <u>essenciais</u>**.*

**Passo 2**: Determinação do **menor conjunto de implicantes/implicados <u>primos</u>** que contenham os mintermos/maxtermos não incluídos nos implicantes/implicados primos essenciais identificados no passo anterior.*

**Passo 3**: **Escrita da expressão simplificada** como soma/produto de todos os termos de produto/soma seleccionados nos passos 1 e 2.

* **<u>Incluindo as indeterminações</u>** sempre que isso permita reduzir o número de literais presentes nesse implicante/implicado.

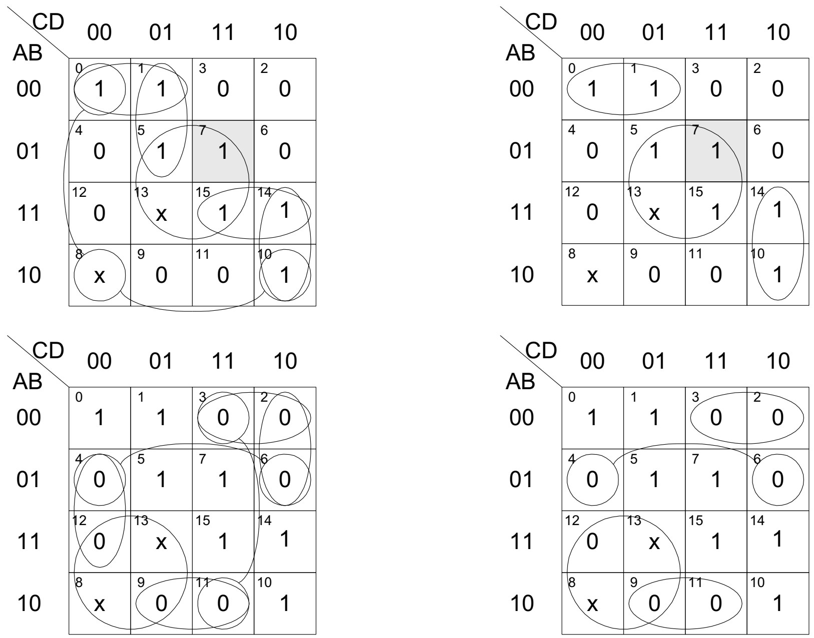
# Minimização de Karnaugh

- Método de Minimização de Karnaugh (cont.)
  - Funções Incompletamente Especificadas:

  Exemplos:

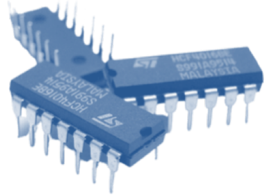
$$f(A,B,C,D) = \Sigma m(0,1,5,7,10,14,15)+\Sigma m_d(8,13)$$
$$=\Pi M(2,3,4,6,9,11,12).\Pi M_d(8,13)$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | [0] 1 | [1] 1 | [3] 0 | [2] 0 |
| 01 | [4] 0 | [5] 1 | [7] 1 | [6] 0 |
| 11 | [12] 0 | [13] x | [15] 1 | [14] 1 |
| 10 | [8] x | [9] 0 | [11] 0 | [10] 1 |



TÉCNICO LISBOA

# Minimização de Karnaugh



- Método de Minimização de Karnaugh (cont.)
  - Exemplos:

| AB \ CDE | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | $^{0}$ x | $^{1}$ 0 | $^{3}$ 0 | $^{2}$ 1 | $^{6}$ 0 | $^{7}$ 1 | $^{5}$ 0 | $^{4}$ 0 |
| 01 | $^{8}$ x | $^{9}$ 1 | $^{11}$ 1 | $^{10}$ x | $^{14}$ x | $^{15}$ 1 | $^{13}$ x | $^{12}$ x |
| 11 | $^{24}$ 0 | $^{25}$ x | $^{27}$ x | $^{26}$ 0 | $^{30}$ 0 | $^{31}$ 1 | $^{29}$ x | $^{28}$ 0 |
| 10 | $^{16}$ 1 | $^{17}$ 0 | $^{19}$ 0 | $^{18}$ 1 | $^{22}$ 0 | $^{23}$ 1 | $^{21}$ 0 | $^{20}$ 0 |

| AB \ CDE | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | $^{0}$ x | $^{1}$ 0 | $^{3}$ 0 | $^{2}$ 1 | $^{6}$ 0 | $^{7}$ 1 | $^{5}$ 0 | $^{4}$ 0 |
| 01 | $^{8}$ x | $^{9}$ 1 | $^{11}$ 1 | $^{10}$ x | $^{14}$ x | $^{15}$ 1 | $^{13}$ x | $^{12}$ x |
| 11 | $^{24}$ 0 | $^{25}$ x | $^{27}$ x | $^{26}$ 0 | $^{30}$ 0 | $^{31}$ 1 | $^{29}$ x | $^{28}$ 0 |
| 10 | $^{16}$ 1 | $^{17}$ 0 | $^{19}$ 0 | $^{18}$ 1 | $^{22}$ 0 | $^{23}$ 1 | $^{21}$ 0 | $^{20}$ 0 |

**Implicantes primos essenciais**

| AB \ CDE | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | $^{0}$ x | $^{1}$ 0 | $^{3}$ 0 | $^{2}$ 1 | $^{6}$ 0 | $^{7}$ 1 | $^{5}$ 0 | $^{4}$ 0 |
| 01 | $^{8}$ x | $^{9}$ 1 | $^{11}$ 1 | $^{10}$ x | $^{14}$ x | $^{15}$ 1 | $^{13}$ x | $^{12}$ x |
| 11 | $^{24}$ 0 | $^{25}$ x | $^{27}$ x | $^{26}$ 0 | $^{30}$ 0 | $^{31}$ 1 | $^{29}$ x | $^{28}$ 0 |
| 10 | $^{16}$ 1 | $^{17}$ 0 | $^{19}$ 0 | $^{18}$ 1 | $^{22}$ 0 | $^{23}$ 1 | $^{21}$ 0 | $^{20}$ 0 |

**ou**

| AB \ CDE | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | $^{0}$ x | $^{1}$ 0 | $^{3}$ 0 | $^{2}$ 1 | $^{6}$ 0 | $^{7}$ 1 | $^{5}$ 0 | $^{4}$ 0 |
| 01 | $^{8}$ x | $^{9}$ 1 | $^{11}$ 1 | $^{10}$ x | $^{14}$ x | $^{15}$ 1 | $^{13}$ x | $^{12}$ x |
| 11 | $^{24}$ 0 | $^{25}$ x | $^{27}$ x | $^{26}$ 0 | $^{30}$ 0 | $^{31}$ 1 | $^{29}$ x | $^{28}$ 0 |
| 10 | $^{16}$ 1 | $^{17}$ 0 | $^{19}$ 0 | $^{18}$ 1 | $^{22}$ 0 | $^{23}$ 1 | $^{21}$ 0 | $^{20}$ 0 |

TÉCNICO LISBOA