

FUNÇÕES LÓGICAS

FUNÇÕES LÓGICAS - 2

- **SUMÁRIO:**
 - FUNÇÕES LÓGICAS
 - CIRCUITOS COM PORTAS NAND
 - CIRCUITOS COM PORTAS NOR
 - REPRESENTAÇÕES NORMALIZADAS
 - SOMA DE PRODUTOS
 - MINTERMOS
 - PRODUTO DE SOMAS
 - MAXTERMS
 - FUNÇÕES INCOMPLETAMENTE ESPECIFICADAS

FUNÇÕES LÓGICAS - 3

● 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

$$f_0(x, y) = 0$$

$$f_3(x, y) = \bar{x}$$

$$f_5(x, y) = \bar{y}$$

$$f_8(x, y) = x \cdot y \quad \text{AND}$$

$$f_{10}(x, y) = y$$

$$f_{12}(x, y) = x$$

$$f_{14}(x, y) = x + y \quad \text{OR}$$

$$f_{15}(x, y) = 1$$

FUNÇÕES LÓGICAS - 4

● FUNÇÕES NOR e NAND

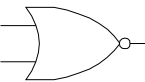
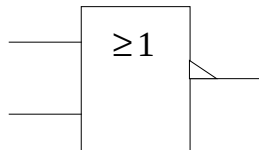
x	y	f_1	f_7
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	0

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

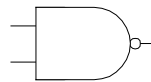
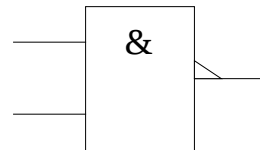
$$f_7(x, y) = \bar{x} + \bar{y} = \overline{x \cdot y} \quad \text{NAND}$$

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

► Simbologia



NOR



NAND

Nas tecnologias mais comuns (p.ex. CMOS), as portas NOR e NAND (portas inversoras) requerem menos transistores que as portas OR e AND (portas não inversoras).

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

FUNÇÕES LÓGICAS - 5

● FUNÇÕES OU-EXCLUSIVO

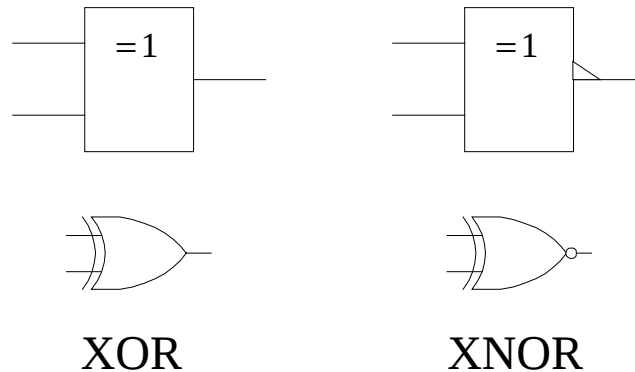
x	y	f_6	f_9
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

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

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

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

► Simbologia



FUNÇÕES LÓGICAS - 6

● FUNÇÕES BASEADAS NO OPERADOR BOOLEANO *IMPLICAÇÃO*

x	y	f_2	f_4	f_{11}	f_{13}
0	0	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	1	0	0	1	1

$$f_{11}(x, y) = \bar{x} + x \cdot y = \bar{x} + y = x \rightarrow y \quad : x \text{ implica } y$$

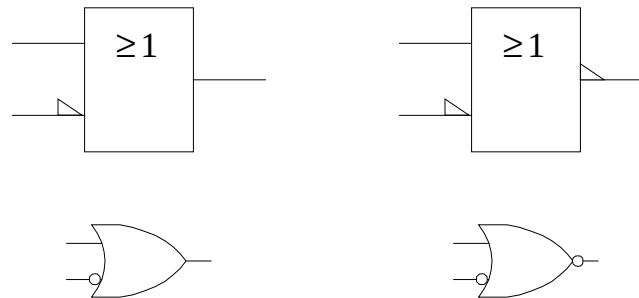
$$f_{13}(x, y) = \bar{y} + x \cdot y = y \rightarrow x$$

$$f_2(x, y) = \overline{y \rightarrow x}$$

$$f_4(x, y) = \overline{x \rightarrow y}$$

► Simbologia

Estas funções não estão habitualmente disponíveis como portas lógicas básicas para realização de sistemas digitais.



FUNÇÕES LÓGICAS - 7

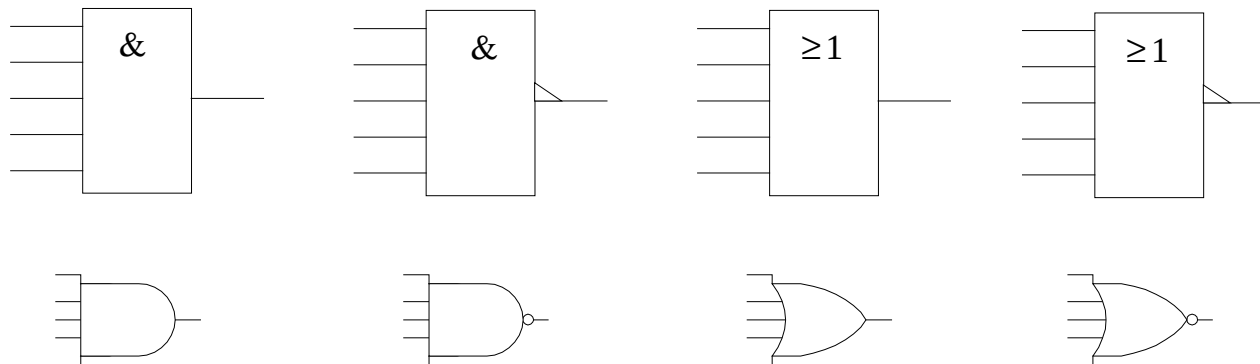
● PORTAS COM MAIS DE 2 ENTRADAS

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 estiverem a 1.

► Simbologia

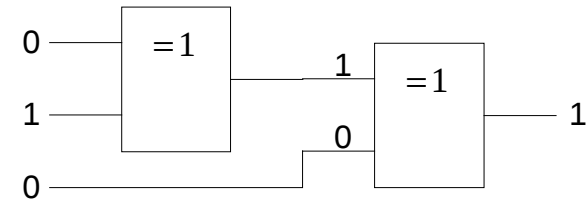
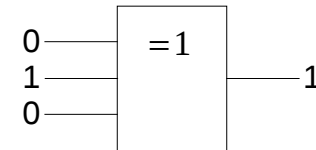


FUNÇÕES LÓGICAS - 8

● FUNÇÃO OU-EXCLUSIVO COM MAIS DE 2 ENTRADAS

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

A 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.



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

A XOR de N entradas é verdadeira se o número de entradas iguais a 1 for ímpar.

De facto e embora usada genericamente, a designação OU-exclusivo só é estritamente correcta para a função de 2 variáveis.

As funções de paridade são muito utilizadas em sistemas de comunicação que requerem detecção de erros: um bit de paridade é habitualmente usado para detectar erros de transmissão.

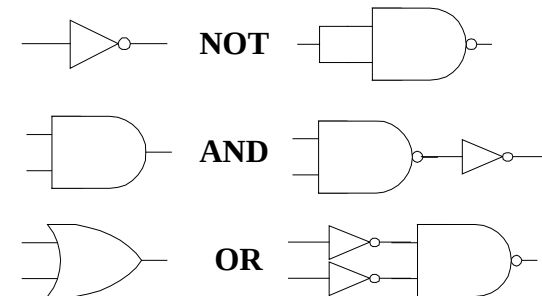
● CIRCUITOS COM PORTAS NAND

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 directa das operações NOT, AND e OR.

A operação NOT é normalmente considerada, em sentido lato, como uma NAND de 1 entrada.

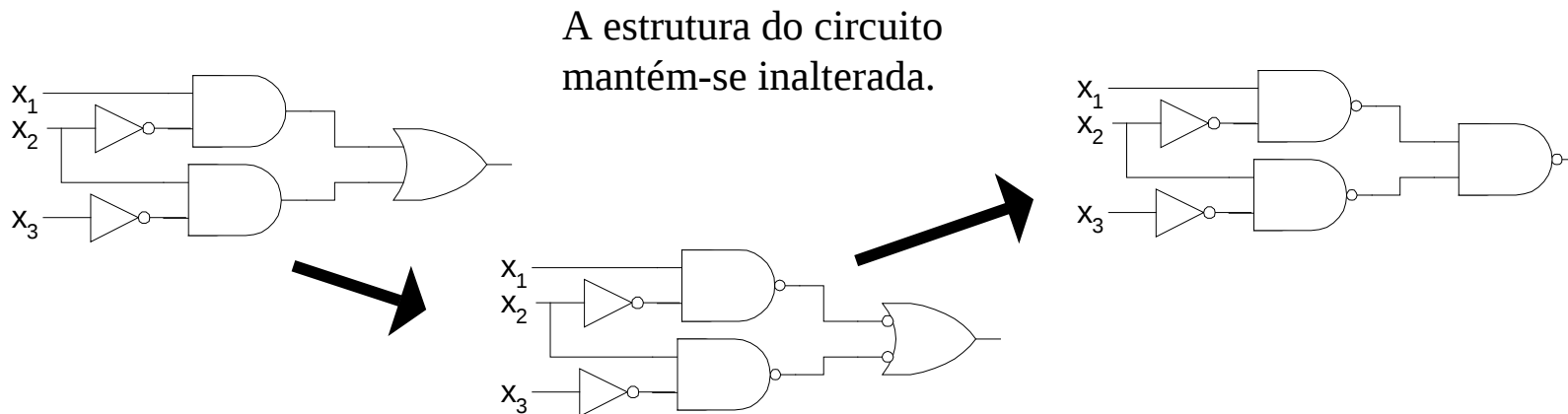
Nalgumas tecnologias (p.ex. TTL) as portas NAND são as portas mais simples (portanto mais baratas), pelo que é vantajosa a realização de circuitos só com NANDs.



● CIRCUITOS COM PORTAS NAND (cont.)

Uma função representada na forma de uma soma de produtos, pode ser transformada numa forma directamente realizável apenas com portas NAND por simples aplicação da lei de DeMorgan.

$$\begin{aligned} f &= x_1 \bar{x}_2 + \bar{x}_3 x_2 = \overline{\overline{x_1 \bar{x}_2 + \bar{x}_3 x_2}} = \overline{\overline{x_1 \bar{x}_2} \cdot \overline{\bar{x}_3 x_2}} \\ &= (x_1 \text{ nand } \bar{x}_2) \text{ nand } (\bar{x}_3 \text{ nand } x_2) \end{aligned}$$



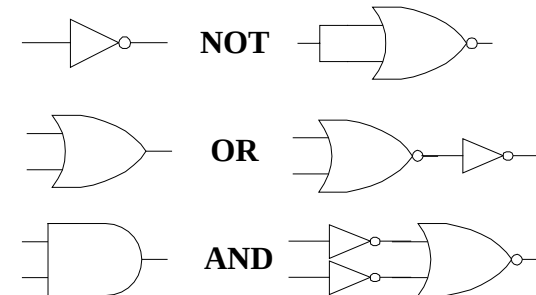
FUNÇÕES LÓGICAS - 11

● CIRCUITOS COM PORTAS NOR

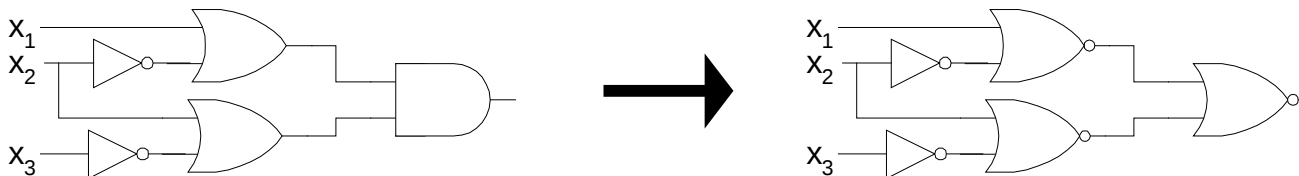
Dual:

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.



$$\begin{aligned} g &= (x_1 + \bar{x}_2) \cdot (\bar{x}_3 + x_2) = \overline{\overline{(x_1 + \bar{x}_2)} \cdot \overline{(\bar{x}_3 + x_2)}} = \overline{\overline{(x_1 + \bar{x}_2)} + \overline{(\bar{x}_3 + x_2)}} \\ &= (x_1 \text{ nor } \bar{x}_2) \text{ nor } (\bar{x}_3 \text{ nor } x_2) \end{aligned}$$



● REPRESENTAÇÃO NORMALIZADA: SOMA DE PRODUTOS

Designa-se por **forma normal disjuntiva** de uma função booleana simples completamente especificada, $y=f(x_1, x_2, \dots, x_N)$, uma expressão lógica representativa da função com a estrutura de uma soma de produtos.

Por esta razão designa-se habitualmente uma forma normal disjuntiva simplesmente por **soma de produtos**.

Se cada parcela for constituída por um produto lógico envolvendo n literais distintos, diz-se que a função se encontra representada em **primeira forma canónica** ou **forma canónica disjuntiva**.

► Exemplos

$$f(x_1, x_2, x_3) = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \quad \leftarrow \text{forma não canónica}$$

$$f(x_1, x_2, x_3) = x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \quad \leftarrow \text{forma canónica}$$

● MINTERMOS

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

Mintermos para 3 variáveis

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

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

Uma função 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.

● REPRESENTAÇÃO NORMALIZADA: PRODUTO DE SOMAS

Designa-se por **forma normal conjuntiva** de uma função booleana simples completamente especificada, $y=f(x_1, x_2, \dots, x_N)$, uma expressão lógica representativa da função com a estrutura de um produto de somas.

Por esta razão designa-se habitualmente uma forma normal conjuntiva simplesmente por **produto de somas**.

Se cada parcela for constituída por uma soma lógica envolvendo n literais distintos, diz-se que a função se encontra representada em **segunda forma canónica** ou **forma canónica conjuntiva**.

► Exemplos

$$f(x_1, x_2, x_3) = (x_1 + x_2) \cdot (\overline{x_1} + \overline{x_2} + \overline{x_3}) \quad \leftarrow \text{forma não canónica}$$

$$f(x_1, x_2, x_3) = (x_1 + x_2 + x_3) \cdot (x_1 + x_2 + \overline{x_3}) \cdot (\overline{x_1} + \overline{x_2} + \overline{x_3}) \quad \leftarrow \text{forma canónica}$$

● 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 função 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 $\neq 0$ com o número mínimo de 1's na tabela da verdade.

O maxtermo corresponde a uma função $\neq 1$ com o número máximo de 1's na tabela da verdade.

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

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

► Exemplo

$$m_3 = \overline{x}_3 . x_2 . x_1 = \overline{\overline{\overline{x}_3 . x_2 . x_1}} = \overline{x_3 + \overline{x}_2 + \overline{x}_1} = \overline{M_3}$$

● TABELA DA VERDADE ↔ SOMA DE PRODUTOS

Uma função booleana pode ser expressa algebricamente, como uma soma de produtos, directamente a partir da tabela de verdade.

A soma inclui todos os mintermos para os quais a função vale 1.

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

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

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

● SOMA DE PRODUTOS $\leftrightarrow$ PRODUTO DE SOMAS

Conversão entre formas canônicas: 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.

x_3	x_2	x_1	f	$\bar{f}$
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

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

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

$$f(x_3, x_2, x_1) = \overline{m_2 + m_4 + m_6} = \bar{m}_2 \cdot \bar{m}_4 \cdot \bar{m}_6$$

$$= M_2 \cdot M_4 \cdot M_6$$

$$f = \overline{\bar{x}_3 \cdot x_2 \cdot \bar{x}_1 + x_3 \cdot \bar{x}_2 \cdot \bar{x}_1 + x_3 \cdot x_2 \cdot \bar{x}_1}$$

$$= (\bar{x}_3 \cdot x_2 \cdot \bar{x}_1) \cdot (x_3 \cdot \bar{x}_2 \cdot \bar{x}_1) \cdot (x_3 \cdot x_2 \cdot \bar{x}_1)$$

$$= (x_3 + \bar{x}_2 + x_1) \cdot (\bar{x}_3 + x_2 + x_1) \cdot (\bar{x}_3 + \bar{x}_2 + x_1)$$

● TABELA DA VERDADE $\leftrightarrow$ 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.

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

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

$$f(x_3, x_2, x_1) = (x_3 + \bar{x}_2 + x_1) \quad \leftarrow M_2$$

$$\cdot (\bar{x}_3 + x_2 + x_1) \quad \leftarrow M_4$$

$$\cdot (\bar{x}_3 + \bar{x}_2 + x_1) \quad \leftarrow M_6$$

● FUNÇÕES INCOMPLETAMENTE ESPECIFICADAS

► **Exemplo** Função que detecta se um 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 escolher para cada 'X' o mais adequado entre '0' ou '1'.

$$\begin{aligned} \text{Representação: } f(x_3, x_2, x_1) &= \sum m(3,6) + \sum m_d(0,7) \\ &= m_3 + m_6 + m_{d0} + m_{d7} \end{aligned}$$

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

● FUNÇÕES INCOMPLETAMENTE ESPECIFICADAS (cont.)

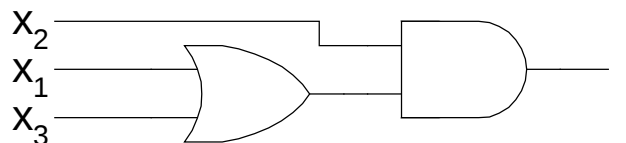
► Exemplo

x_3	x_2	x_1	f	$\rightarrow g$
0	0	0	X	$\rightarrow 0$
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	$\rightarrow 1$

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'.

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



BIBLIOGRAFIA

[1] M. Morris Mano, Charles R. Kime, “Logic and Computer Design Fundamentals”, Prentice-Hall International, Inc. (Capítulo 2, Secções 2.3, 2.6-2.7)