# Circuitos Lógicos e Organização de Computadores

## Capítulo 4 – Implementações Otimizadas de Funções Lógicas

Ricardo Pannain
pannain@unicamp.br

# Mapa de Karnaugh

Seja a função $f = \Sigma\, m(0, 2, 4, 5, 6)$, olhando para a tabela verdade,
Outra maneira de se representar uma função → Mapa de Karnaugh

| **Número da linha** | **x1** | **x2** | **x3** | **f** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 0 |

# Mapa de Karnaugh

É uma aplicação sistemática das propriedades 14a e 14b

14a) $x \cdot y + x \cdot \overline{y} = x$ 14b) $(x + y) \cdot (x + \overline{y}) = x$

Exemplo da tabela do slide anterior f = $(m_0, m_2, m_4, m_5, m_6)$

$f = \overline{x1}\ \overline{x2}\ \overline{x3} + \overline{x1}\ x2\ \overline{x3} + x1\ \overline{x2}\ \overline{x3} + x1\ \overline{x2}\ x3 + x1\ x2\ \overline{x3}$

Aplicando 14a com $(m_0, m_2)$ e $(m_4, m_6)$, temos:

1) $\overline{x1}(\overline{x2}\ \overline{x3} + x2\ \overline{x3}) = \overline{x1}\ \overline{x3}$ e 2) $x1(\overline{x2}\ \overline{x3} + x2\ \overline{x3}) = x1\ \overline{x3}$

Aplicando novamente em 1) e 2) → $\overline{x1}\ \overline{x3} + x1\ \overline{x3} = \overline{x3}$

$m_0, m_2, m_4$ e $m_6$ foram trocados por $\overline{x3}$ → os mintermos estão incluídos em $\overline{x3}$

$m_5$ pode ser combinado com $m_4$ → $x1(\overline{x2}\ \overline{x3} + \overline{x2}\ x3) = x1\ \overline{x2}$

→ $f = \overline{x3} + x1\ \overline{x2}$ → (expressão de custo mínimo)

# Mapa de Karnaugh – 2 variáveis

| $x_1$ | $x_2$ | |
|---|---|---|
| 0 | 0 | $m_0$ |
| 0 | 1 | $m_1$ |
| 1 | 0 | $m_2$ |
| 1 | 1 | $m_3$ |

(a) Tabela Verdade

| $x_2$ \ $x_1$ | 0 | 1 |
|---|---|---|
| 0 | $m_0$ | $m_2$ |
| 1 | $m_1$ | $m_3$ |

(b) Mapa de Karnaugh

# Mapa de Karnaugh

Estratégia de minimização → encontrar sempre que possível, os maiores grupos de 1s - SOP (ou 0s - POS), que cubram todos os casos onde o valor de f = 1 ($\overline{f}$ = 1)

TERMINOLOGIA

LITERAL → xi ou $\overline{x}$i
IMPLICANTE → termo produto onde f = 1 (SOP)
PRIMO IMPLICANTE → um implicante que não pode se combinado com outro, i. é, nenhum literal deste implicante pode ser suprimido.
COBERTURA → conjunto de implicantes que determina o valor 1 para a função
CUSTO → Σ no. de portas + Σ no. de entradas (assumir que entradas e entradas barradas estão disponíveis)
MENOR CUSTO → quando a cobertura de uma função consiste de primos implicantes (essenciais ou não)
Se um primo implicante é um mintermo, que não está incluído em outro primo implicante, então ele deve ser incluído na cobertura e é chamado de primo implicante essencial

# Mapa de Karnaugh

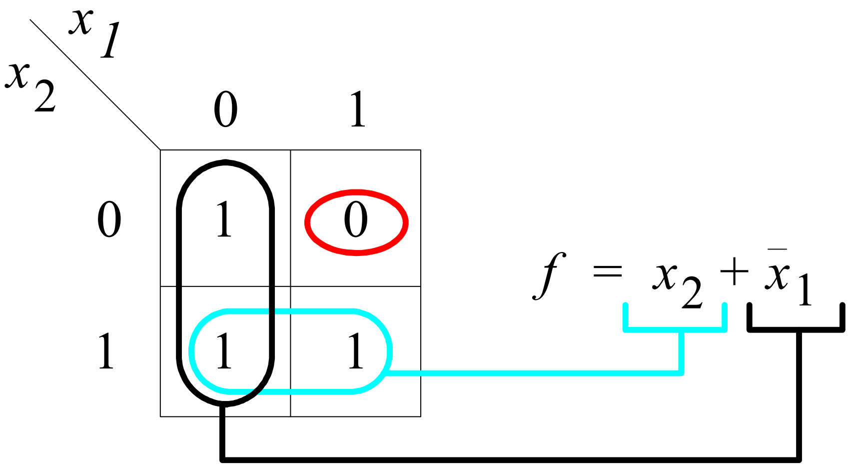
Processo para encontrar circuito de menor custo:

- Gerar todos os primos implicantes de uma dada função f.

- Encontrar o conjunto de primos implicantes essenciais.

- Se o conjunto cobrir todas as possibilidades para f = 1, temos o circuito de menor custo.
  Caso contrário, determinar a(s) primo(s) implicante(s) não essenciais, que seriam adicionados para completar a cobertura de custo mínimo. (esta escolha geralmente não é obvia → usar heurística para escolher e melhor solução.

# Minimização de uma função lógica de 2 variáveis usando Mapa de Karnaugh




| x1 | x2 | f |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| x1 | x2 | f |
|---|---|---|
| 0 | 0 | m0 |
| 0 | 1 | m1 |
| 1 | 0 | m2 |
| 1 | 1 | m3 |

SOP

f = x1'x'2 + x'1x2 + x1x2 = f = $\Sigma$ m ( 0,1,3)

f = x'1 (x'2 + x2) + x2 (x'1 +x1) = x'1 + x2

POS

f = $\Pi$ M (2) = (x1' + x2)

# Exercício – simplifique usando mapa de Karnaugh

| x1 | x2 | f |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# Exercício – simplifique usando mapa de Karnaugh

| x2 \ x1 | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

| x1 | x2 | f |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

1 → x1 / 2 → x2

f = x1 + x2

SOP – f = x1’x2+x1x2’+x1x2 = x2(x1’+x1) +x1 (x2’+x2) =

=x2 + x1

# Mapa de Karnaugh – 3 variáveis

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

(a) Tabela Verdade

| $x_3$ \ $x_1 x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $m_0$ | $m_2$ | $m_6$ | $m_4$ |
| 1 | $m_1$ | $m_3$ | $m_7$ | $m_5$ |

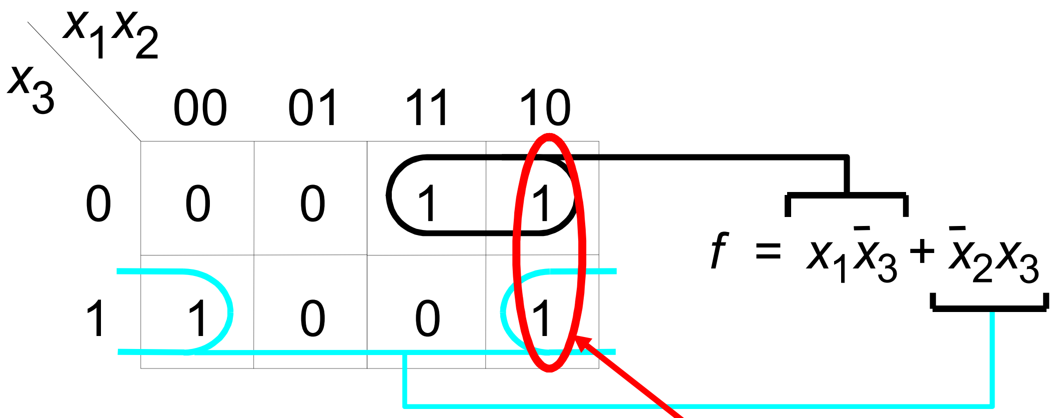
(b) Mapa de Karnaugh

Exercícios

1. $f = \Sigma m(1,4,5,6)$

2. $f = \Sigma m(0,2,4,5,6)$

# Minimização de uma função lógica de 3 variáveis usando Mapa de Karnaugh



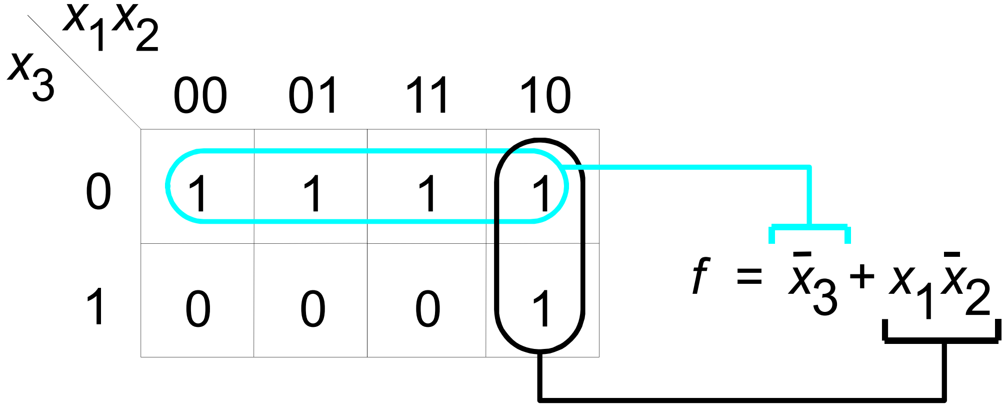

(a) Exemplo de simplificação

redundante

1. f = Σm(1,4,5,6)

# Minimização de uma função lógica de 3 variáveis usando Mapa de Karnaugh

2. $f = \Sigma m(0,2,4,5,6)$




(b) Função do slide 2

# Mapa de Karnaugh – 4 variáveis

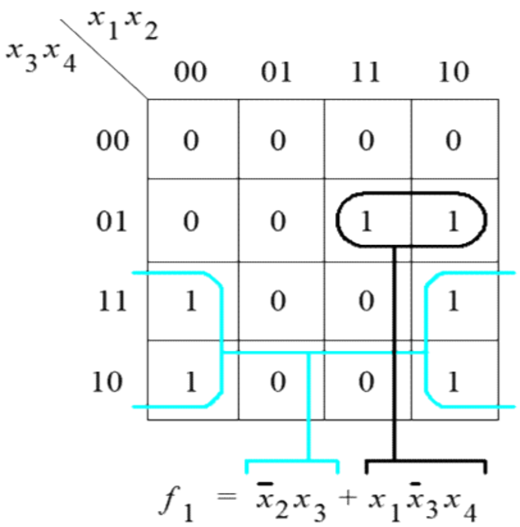
| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $m_0$ | $m_4$ | $m_{12}$ | $m_8$ |
| 01 | $m_1$ | $m_5$ | $m_{13}$ | $m_9$ |
| 11 | $m_3$ | $m_7$ | $m_{15}$ | $m_{11}$ |
| 10 | $m_2$ | $m_6$ | $m_{14}$ | $m_{10}$ |

# Exercícios 1

$$f = \sum m\,(2,3,\,9,10,11,\,13)$$

| $x_3x_4$ \ $x_1x_2$ | 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 | $m_2$ | $m_6$ | $m_{14}$ | $m_{10}$ |

# Exercícios 1



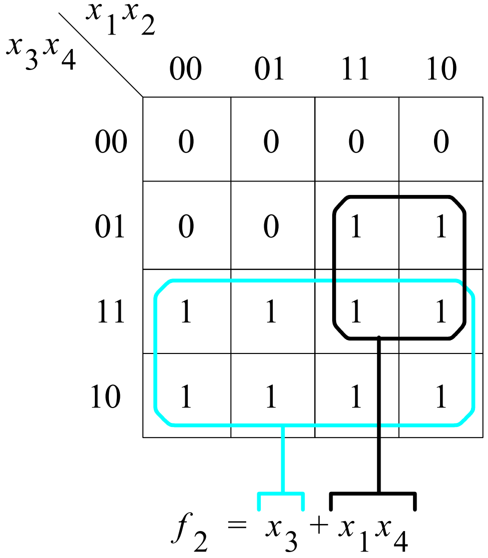

$$f = \sum m\ (2,3,\ 9,10,11,\ 13)$$

# Exercícios 2

$$f = \sum m\,(2,3,6,7,9,10,11,13,14,15)$$

# Exercícios 2

$$f = \sum m\,(2,3,6,7,9,10,11,13,14,15)$$

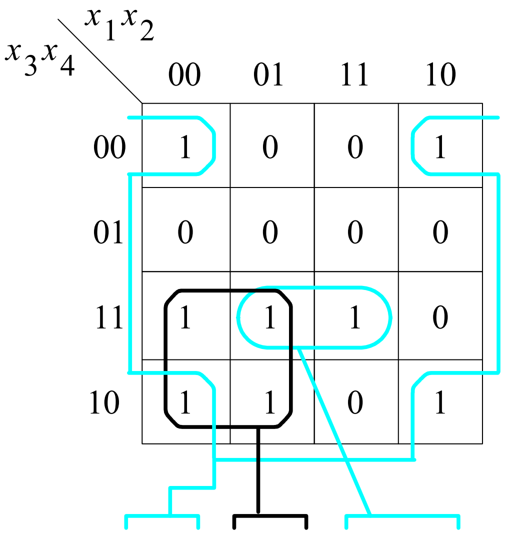


# Exercícios 3

$$f = \sum m\,(0,2,3,6,7,8,10,15)$$

# Exercícios 3

f = $\Sigma$ m (0,2,3, 6 ,7, 8, 10, 16)




$$f_3 = \bar{x}_2\bar{x}_4 + \bar{x}_1x_3 + x_2x_3x_4$$

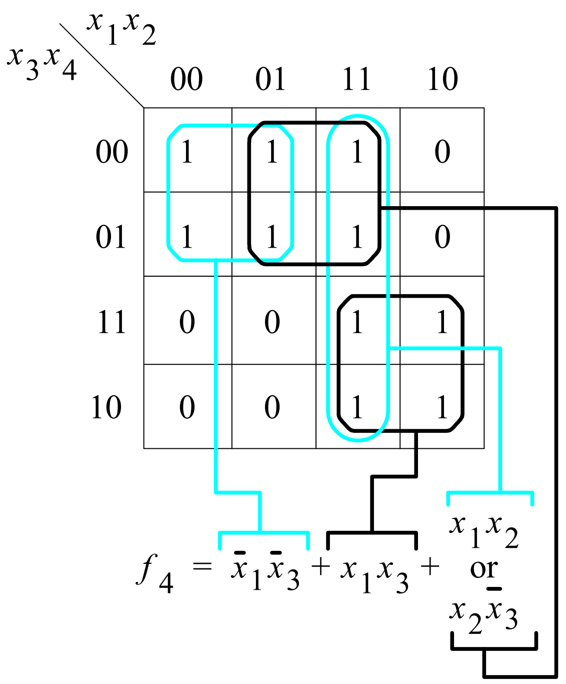
# Exercícios 4

$f = \sum m\,(0,1,4,5,10,11,12,13,14,15)$

# Exercícios 4

$f = \sum m\,(0,1,4,5,10,11,12,13,14,15)$

f = x1'x3'+x1x2+x1x3
ou
f = x1'x3'+x1x3+x2x3'



## EXERCÍCIO 5

$f = \Pi M(0,1,6,7,8,9,10,14,15)$

EXERCÍCIO 5

$$f = \Pi M(0,1,6,7,8,9,10,14,15) = (x2 + x3) \,.\, (x2' + x3') \,.\, (x1' + x3' + x4)$$

ou

$$= (x2 + x3) \,.\, (x2' + x3') \,.\, (x1' + x2 + x4)$$

| x3x4\|x1x2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 1 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 0 | 0 |

Custo = 3 or + 7 ent. + 1 and+ 3 ent. = 14

# Simplificação de uma função lógica de 5 variáveis usando Mapa de Karnaugh

| x3x4 \ x1x2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 8 | 24 | 16 |
| 01 | 2 | 10 | 26 | 18 |
| 11 | 6 | 14 | 30 | 22 |
| 10 | 4 | 12 | 28 | 20 |

x5 = 0

| x3x4 \ x1x2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 9 | 25 | 17 |
| 01 | 3 | 11 | 27 | 19 |
| 11 | 7 | 15 | 31 | 23 |
| 10 | 5 | 13 | 29 | 21 |

x5 = 1

f(x1,x2,x3,x4,x5)

# Simplificação de uma função lógica de 5 variáveis usando Mapa de Karnaugh – EXERCÍCIO 6

$$f = \Sigma m\ (4,5,6,7,12,13,14,15,17,18,19,26,27)$$

| x3x4 \ x1x2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 8 | 24 | 16 |
| 01 | 2 | 10 | 26 | 18 |
| 11 | 6 | 14 | 30 | 22 |
| 10 | 4 | 12 | 28 | 20 |

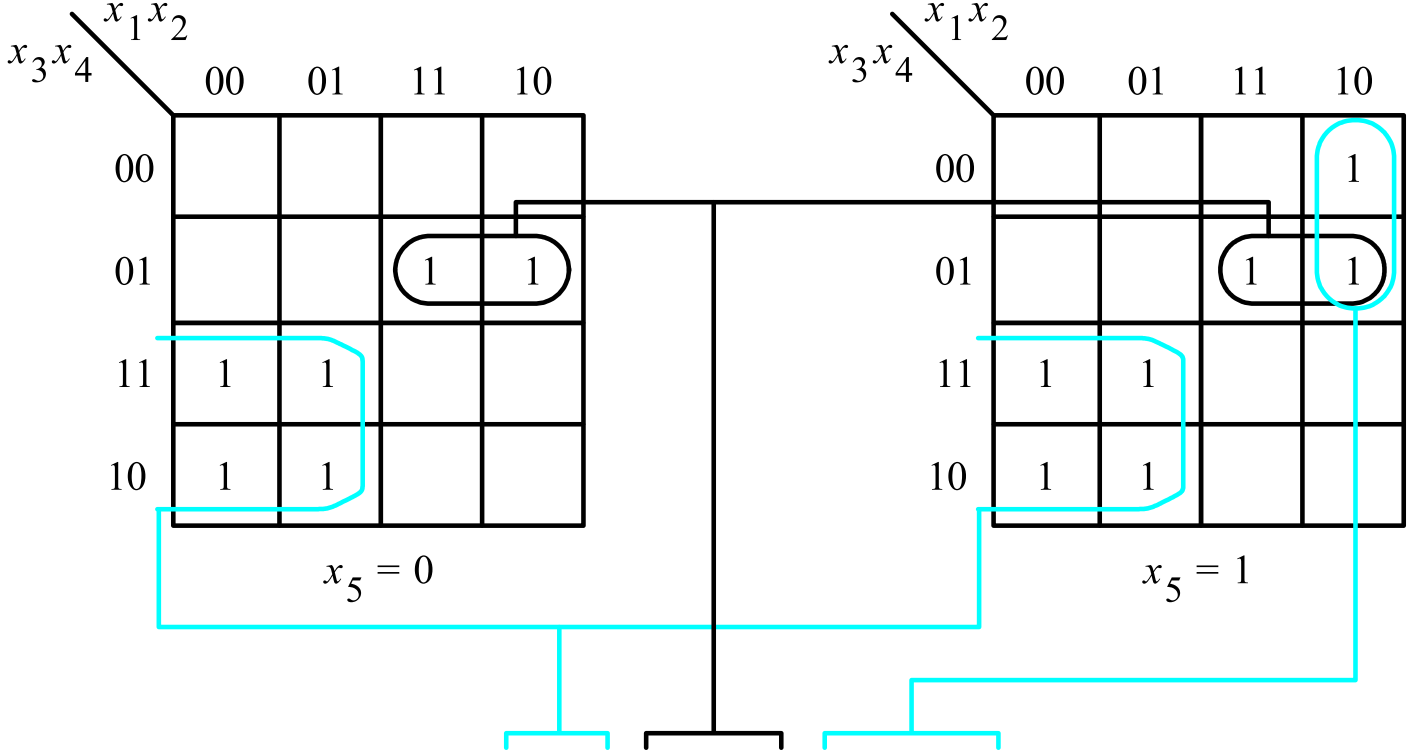
x5 = 0

| x3x4 \ x1x2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 9 | 25 | 17 |
| 01 | 3 | 11 | 27 | 19 |
| 11 | 7 | 15 | 31 | 23 |
| 10 | 5 | 13 | 29 | 21 |

x5 = 1

# Simplificação de uma função lógica de 5 variáveis usando Mapa de Karnaugh – EXERCÍCIO 6

$$f = \Sigma m\,(4,5,6,7,12,13,14,15,17,18,19,26,27)$$




$$f_1 = \bar{x}_1 x_3 + x_1 \bar{x}_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_5$$

# Simplificação de uma função lógica de 5 variáveis usando Mapa de Karnaugh – EXERCÍCIO 7

$$f = \Sigma m\,(1,2,3,4,10,11,12,13,18,19,27)$$

# Simplificação de uma função lógica de 5 variáveis usando Mapa de Karnaugh – EXERCÍCIO 8

f = Σm (1,2,3,4,10,11,12,13,18,19,27)    Custo = 11+55+1+11 = 78

| x3x4 \ x1x2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | 1 | 1 | | 1 |
| 11 | | | | |
| 10 | 1 | 1 | | |

x5 = 0

| x3x4 \ x1x2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | | |
| 01 | 1 | 1 | 1 | 1 |
| 11 | | | | |
| 10 | | 1 | | |

x5 = 1

f = x3’x4x5 + x1’x3’x4 + x2’x3’x4+ x1’x2’x3’x5 + x1’x2x3x4’ + x1’x3x4’x5’

Custo = 34

## Funções não completamente especificadas

Em sistemas digitais, frequentemente temos certas entradas que nunca ocorrem. Por exemplo: 2 chaves x1 e x2 interligadas que nunca podem ser fechadas ao mesmo tempo → (x1,x2) = (0,0) , (0,1) ou (1, 0). A combinação (1,1) nunca vai ocorrer. Chamamos esta combinação de don’t care conditions.

| x1 | x2 | f |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | d |

| x2 \ x1 | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | d |

Sem don’t care = x1x2’+ x1’x2

Com don’t care = x1 + x2

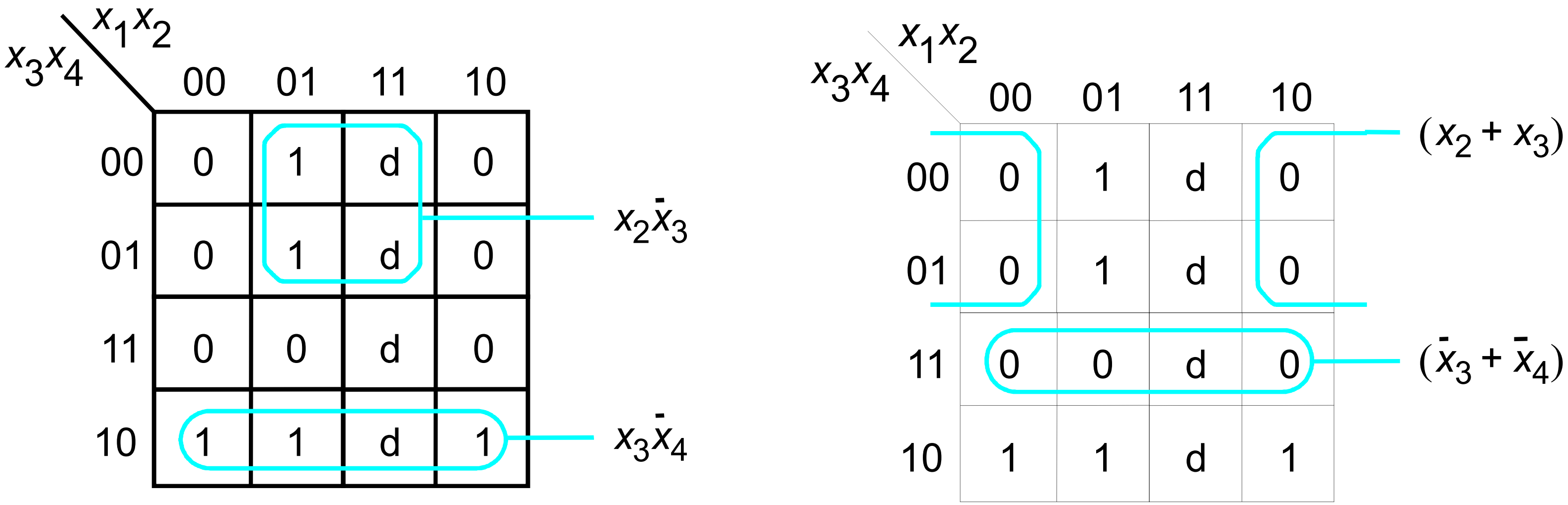
# Funções especificadas não completamente – EXERCÍCIO 9 –SOP e POS

$f = \Sigma\, m(2, 4, 5, 6, 10) + D(12, 13, 14, 15)$

OBS. – Escrever a função sem levar em conta o don’t care e comparar

# Funções especificadas não completamente

$f = \Sigma\, m(2, 4, 5, 6, 10) + D(12, 13, 14, 15)$




(a) SOP implementation (b) POS implementation

OBS. – Escrever a função sem levar em conta o don’t care e comparar

# Mapa de Karnaugh – Exercício 10

$$f = \Sigma\, m(0, 1, 2, 3, 7) + D\ (4, 5,6, 9,10)$$

# Mapa de Karnaugh – Exercício 10

$$f = \Sigma\, m(0, 1, 2, 3, 7) + D\,(4, 5, 6, 9,10)$$

| **x1x2 \x3x4** | **00** | **01** | **11** | **10** |
|---|---|---|---|---|
| 00 | 1 | d | 0 | 0 |
| 01 | 1 | d | 0 | d |
| 11 | 1 | 1 | 0 | 0 |
| 10 | 1 | d | 0 | d |

$f$ = x1’x2’ + x1’x3x4 sem don’t care

$f$ = x1’ com don’t care (em verde)

# Mapa de Karnaugh – Exercício 11

$$f = \Pi\, M(0, 4, 8, 10, 11, 14) + D\,(12, 13, 15)$$

## Mapa de Karnaugh – Exercício 11

$f = \Pi\, M(0, 4, 8, 10, 11,14) + D\ (12, 13, 15)$

| **x1x2 \x3x4** | **00** | **01** | **11** | **10** |
|---|---|---|---|---|
| 00 | 0 | 0 | d | 0 |
| 01 | 1 | 1 | d | 1 |
| 11 | 1 | 1 | d | 0 |
| 10 | 1 | 1 | 0 | 0 |

$f = (x1 + x3 + x4) \, . \, (x1'+x3'+x4) \, . \, (x1' + x2 + x3').(x1' + x2 + x4)$ sem don't care

$f = (x3 + x4) \, . \, (x1' + x3')$ don't care ( marron ou magenta)

# Método de Quine McCluskey

O método de Quine-McCluskey para encontrar primos implicantes de uma função booleana, usa um procedimento sistemático para tabulá-los, iniciando com o mintermos/maxtermos e usando a expressão AB + AB´ / (A+B).(A+B’) repetidamente. Os passos básicos para este método são:

1. Listar todos os mintermos/maxtermos e agrupá-los pelo número de 1s que eles contenham.
2. Formar pares de mintermos/maxtermos que diferem por uma variável e criar um novo termo com uma variável a menos (a variável que difere vai ser absorvida).
3. Repetir o passo 2 até não existir um novo termo a ser formado. O resultado é um conjunto de primos implicantes da função.
4. Formar uma tabela onde os mintermos/maxtermos originais definem as colunas e os primos implicantes definem as linhas. A relação entre cada mintermo/maxtermos e um dado primo implicante é indicado por um *X* no cruzamento da linha e coluna, respectivamente, referentes a ambos.
5. Usando a tabela, determinar o primo implicante essencial a um conjunto adicional de primos implicantes que cobrem toda a função.

# Método de Quine McCluskey

Exemplo: $f = \Sigma\, m(0,2,3,7,8,10,11,13)$

f (x1,x2,x3,x4)

0 (0,0,0,0) !

--------------

2 (0,0,1,0) !

8 (1,0,0,0) !

--------------

3 (0,0,1,1) !

10 (1,0,1,0) !

--------------

7 (0,1,1,1) !

11(1,0,1,1) !

13(1,1,0,1) *

(0,2) (0,0,-,0) !

(0,8) (-,0,0,0) !

------------------

(2,3) (0,0,1,-) !

(2,10) (-,0,1,0) !

(8,10) (1,0,-,0) !

------------------

(3,7) (0,-,1,1) *

(3,11) (-,0,1,1) !

(10,11) (1,0,1,-) !

(0,2,8,10) (-,0,-,0) *

------------------------

(2,3,10,11) (-,0,1,-) *

Obs:

| x3x4\|x1<br>x2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | 1 | 1 | 0 | 1 |
| 10 | 1 | 0 | 0 | 1 |

# Método de Quine McCluskey

Exemplo: $f = \Sigma\, m(0,2,3,7,8,10,11,13)$

| **Mintermos** | **0** | **2** | **3** | **7** | **8** | **10** | **11** | **13** |
|---|---|---|---|---|---|---|---|---|
| **Primos Implicantes** | | | | | | | | |
| **(0,2,8,10)** * | **X** | **X** | | | **X** | **X** | | |
| **(2,3,10,11)** * | | **X** | **X** | | | **X** | **X** | |
| **(3,7)** * | | | **X** | **X** | | | | |
| **(13)** * | | | | | | | | **X** |
| | | | | | | | | |

**f= (13) + (3,7) + (2,3,10,11) + (0,2,8,10) =** $x1\ x2\ \overline{x3}\ x4 + \overline{x1}\ x3\ x4 + \overline{x2}\ x3 + \overline{x2}\ \overline{x4}$

# Quine McCluskey - Exercícios

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

# Quine McCluskey - Exercícios

1) $f = \Sigma\, m(2, 3, 5, 6, 7, 10, 11, 13, 14)$   f = (x1,x2,x3,x4)

2 0010
-----------
3 0011
5 0101
6 0110
10 1010
------------
7 0111
11 1011
13 1101
14 1110

(2,3) 001_
(2,6) 0_10
(2,10) _010
--------------
(3,7) 0_11
(3,11) _011
(5,7) 01_1
(5,13) _101
(6,7) 011_
(6,14) _110
(10,11) 101_
(10,14) 1_10

(2,3,6,7) 0_1_
(2,3,10,11) _01_
(2,6,10,14) __10
---------------------

x1’x3
x2’x3
x3x4’

# Método de Quine McCluskey

$f = \Sigma\, m(2, 3, 5, 6, 7, 10, 11, 13, 14)$ f = (x1,x2,x3,x4)

| Mintermos | 2 | 3 | 5 | 6 | 7 | 10 | 11 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|
| (2,6,10,14) | X | | | X | | X | | | X |
| (2,3,10,11) | X | X | | | | X | X | | |
| (2,3,6,7) | X | X | | X | X | | | | |
| (5,7) | | | X | | X | | | | |
| (5,13) | | | X | | | | | X | |

**f= (2,6,10,14) + (2,3,10,11) + (2,3,6,7) + (5,13) = x3x4’ + x2’x3 + x1’x3 + x2x3’x4**

# Quine McCluskey - Exercícios

*2) f* = Π *M*(0, 2, 4, 5, 10, 11, 13, 15)

0    0000
-----------
2    0010
4    0100
-----------
5    0101
10   1010
-----------
11   1011
13   1101
-----------
15   1111

(0,2)   00_0
(0,4)   0_00
---------------
(2,10) _010
(4,5)   010_
---------------
(5,13)   _101
(10,11) 101_
---------------
(11,15)  1_11
(13,15) 11_1

| Maxtermos | 0c | 2c | 4c | 5c | 10c | 11c | 13c | 15c |
|---|---|---|---|---|---|---|---|---|
| 0,2 | x | x | | | | | | |
| 0,4 | x | | x | | | | | |
| 2,10 | | x | | | x | | | |
| 4,5 | | | x | x | | | | |
| 5,13 | | | | x | | | x | |
| 10,11 | | | | | x | x | | |
| 11,15 | | | | | | x | | x |
| 13,15 | | | | | | | x | x |

f = (x1+x2+x4) . (x1+x2'+x3). (x1'+x2+x3') . (x1'+x2'+x4')
ou
f = (x1+x3+x4) . (x2+x3'+x4). (x2'+x3+x4') . (x1'+x3'+x4')

# Quine McCluskey - Exercícios

3\. $f = \Sigma\, m(0, 1, 2, 3, 7) + D\,(4, 5, 6, 9, 10)$

| | |
|---|---|
| 0 | (0,0,0,0) |
| ------------- | |
| 1 | (0,0,0,1) |
| 2 | (0,0,1,0) |
| 4 | (0,1,0,0) |
| --------------- | |
| 3 | (0,0,1,1) |
| 5 | (0,1,0,1) |
| 6 | ( 0,1,1,0) |
| 9 | (1,0,0,1) |
| 10 | ( 1,0,1,0) |
| --------------- | |
| 7 | (0,1,1,1) |

| | |
|---|---|
| (0,1) | (0,0,0,-) |
| (0,2) | (0,0,-,0) |
| (0,4) | (0,-,0,0) |
| ------------------- | |
| (1,3) | (0,0,-,1) |
| (1,5) | (0,-,0,1) |
| (1,9) | (-,0,0,1) |
| (2,3) | (0,0,1,-) |
| (2,6) | (0,-,1,0) |
| (2,10) | (-,0,1,0) |
| (4,5) | (0,1,0,-) |
| (4,6) | (0,1,-,0) |
| ------------------- | |
| (3,7) | (0,-,1,1) |
| (5,7) | (0,1,-,1) |
| (6,7) | (0,1,1,-) |

| | |
|---|---|
| (0,1,2,3) | (0,0,-,-) |
| (0,1,4,5) | (0,-,0,-) |
| (0,2,4,6) | (0,-,-,0) |
| ---------------------- | |
| (1,3,5,7) | (0,-,-,1) |
| (2,3,6,7) | (0,-,1,-) |
| (4,5,6,7) | (0,1,-,-) |

| | |
|---|---|
| (0,1,2,3,4,5,6,7) | (0,-,-,-) |

| | 0 | 1 | 2 | 3 | 7 |
|---|---|---|---|---|---|
| (0,1,2,3,4,5,6,7) | X | X | X | X | X |
| (1,9) | | X | | | |
| (2,10) | | | X | | |

f = x1’

# Quine McCluskey - Exercícios

$4\ .\ f = \Pi\ M(0, 4, 8, 10, 11, 14) + D\ (12, 13, 15)$

| | | | | | |
|---|---|---|---|---|---|
| 0 | (0,0,0,0) | (0,4) | (0,-,0,0) | (0,4,8,12) | (-,-,0,0) |
| 4 | (0,1,0,0) | (0,8) | (-,0,0,0) | (8,10,12,14) | (1,-,-,0) |
| 8 | (1,0,0,0) | (4,12) | (-,1,0,0) | (10,11,14,15) | (1,-,1,-) |
| 10 | (1,0,1,0) | (8,10) | (1,0,-,0) | (12,13,14,15) | (1,1,-,-) |
| 12 | (1,1,0,0) | (8,12) | (1,-,0,0) | | |
| 11 | (1,0,1,1) | (10,11) | (1,0,1,-) | | |
| 13 | (1,1,0,1) | (10,14) | (1,-,1,0) | | |
| 14 | (1,1,1,0) | (12,13) | (1,1,0,-) | | |
| 15 | (1,1,1,1) | (12,14) | (1,1,-,0) | | |
| | | (11,15) | (1,-,1,1) | | |
| | | (13,15) | (1,1,-,1) | | |
| | | (14,15) | (1,1,1,-) | | |

$4\ .\ f = \Pi\ M(0, 4, 8, 10, 11, 14) + D\ (12, 13, 15)$

| | 0 | 4 | 8 | 10 | 11 | 14 |
|---|---|---|---|---|---|---|
| (0,4,8,12) | X | X | X | | | |
| (8,10,12,14) | | | X | X | | X |
| (10,11,14,15) | | | | X | X | X |
| (12,13,14,15) | | | | | | X |
| | | | | | | |

$$F = (x3 + x4)\ .(x1' + x3')$$

# Circuitos de múltiplas saídas

Dê o menor circuito definido pelas funções abaixo:

*1* - $f_1 = \Sigma\, m(2,3,5,6,7,8,9,12,13)$

$f_2 = \Sigma\, m(2,3,6,7,8,9,12,13,15)$

2- $f_3 = \Sigma\, m(1,3,5,6,7,13,15)$

$f_4 = \Sigma\, m(1,3,6, 9,11,13,15)$

OBS. – cada circuito (de 1 e 2) tem 4 entradas e duas saídas

# Circuitos de múltiplas saídas

*1 -* $f_1 = \Sigma\, m(2,3,5,6,7,8,9,12,13)$

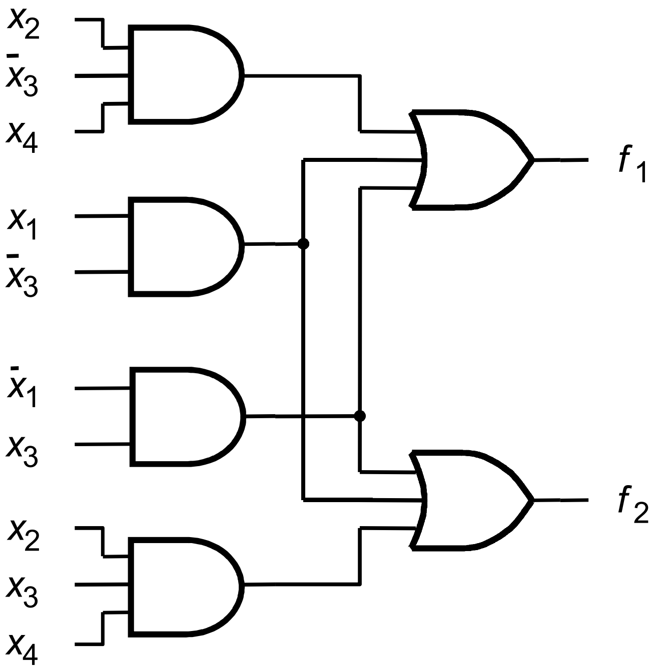
$f_2 = \Sigma\, m(2,3,6,7,8,9,12,13,15)$

| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | | 1 | 1 | 1 |
| 11 | 1 | 1 | | |
| 10 | 1 | 1 | | |

(a) Função $f_1$

| $x_3x_4$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | | | 1 | 1 |
| 11 | 1 | 1 | 1 | |
| 10 | 1 | 1 | | |

(b) Função $f_2$

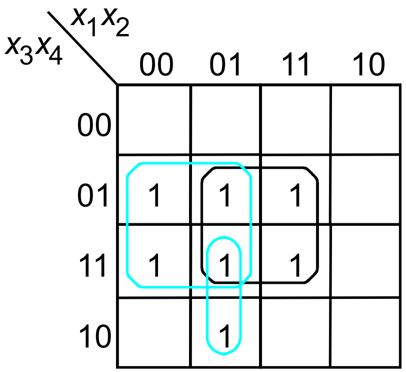



(c) Circuito combinando $f_1$ and $f_2$

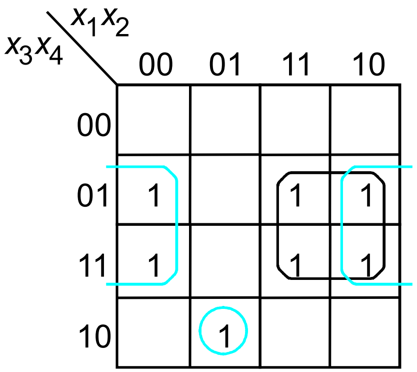
# Circuitos de múltiplas saídas

2- $f_3 = \Sigma\, m(1,3,5,6,7,13,15)$

$f_4 = \Sigma\, m(1,3,6, 9,11,13,15)$




(a) Funçao otimizada → $f_3$



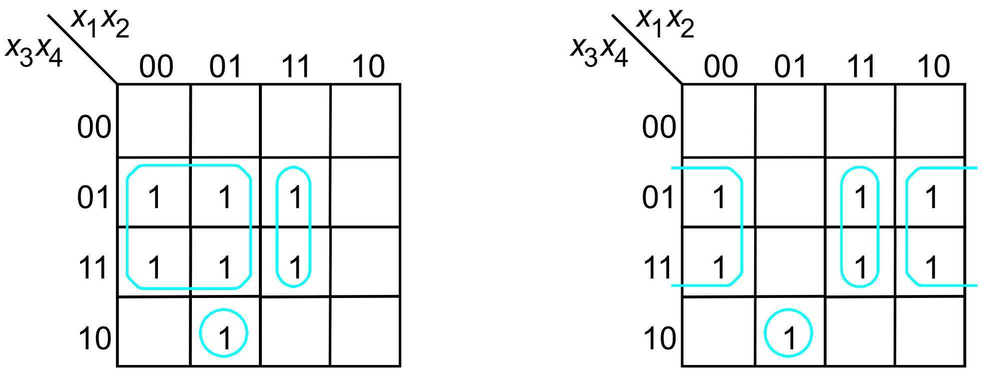

(b) Função otimizada → $f_4$

f3

custo = [3 ANDs + 7 entradas +1 OR
+ 3 entradas] +

f4

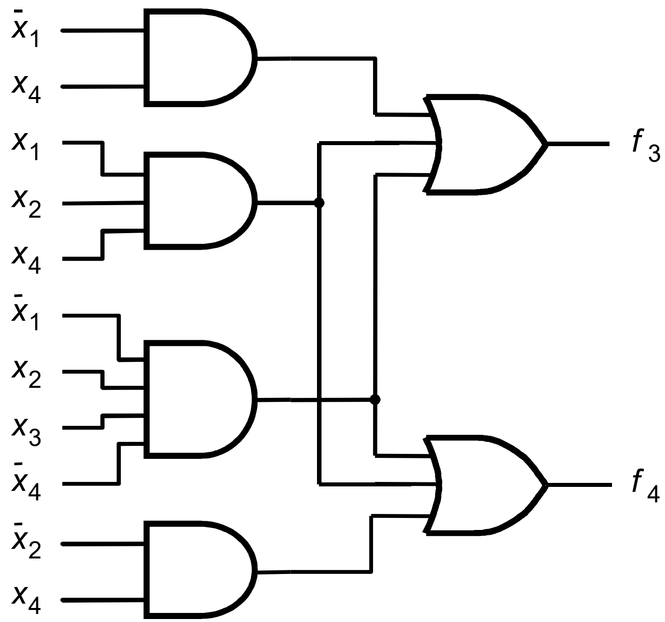
+ [ 4 ANDs + 8 entradas + 1 OR
+ 3 entradas] = 14+15 = 29




(c) Otimização combinando $f_3$ e $f_4$

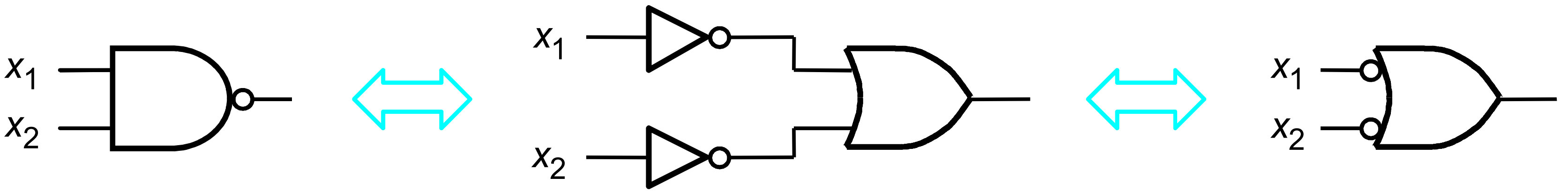
custo =4 ANDs + 11 entradas
+ 2 OR+ 6 = 23

# Circuitos de múltiplas saídas

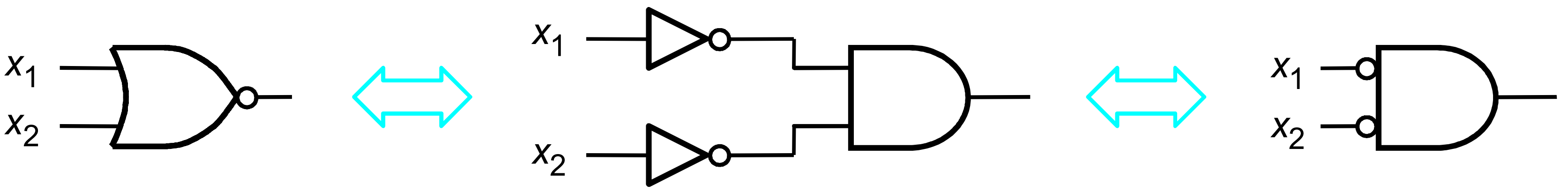



(d) Circuito combinado $f_3$ e $f_4$

# Teorema de DeMorgan → circuitos com NANDs e NORs




(a) $\overline{x_1 x_2} = \bar{x}_1 + \bar{x}_2$




(b) $\overline{x_1 + x_2} = \bar{x}_1 \bar{x}_2$

# Teorema de DeMorgan: Observação

$f = x1'x2 + x3x4'$

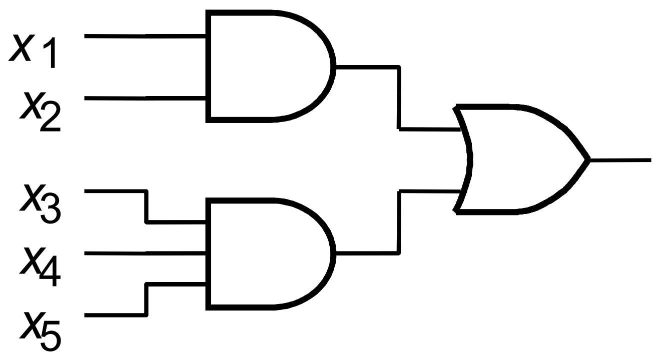
$f' = \overline{x1'x2 + x3x4'} = \overline{(x1' . x2)} . \overline{(x3 .x4')}$

$f' = (x1+ x2') . (x3' + x4) = g$

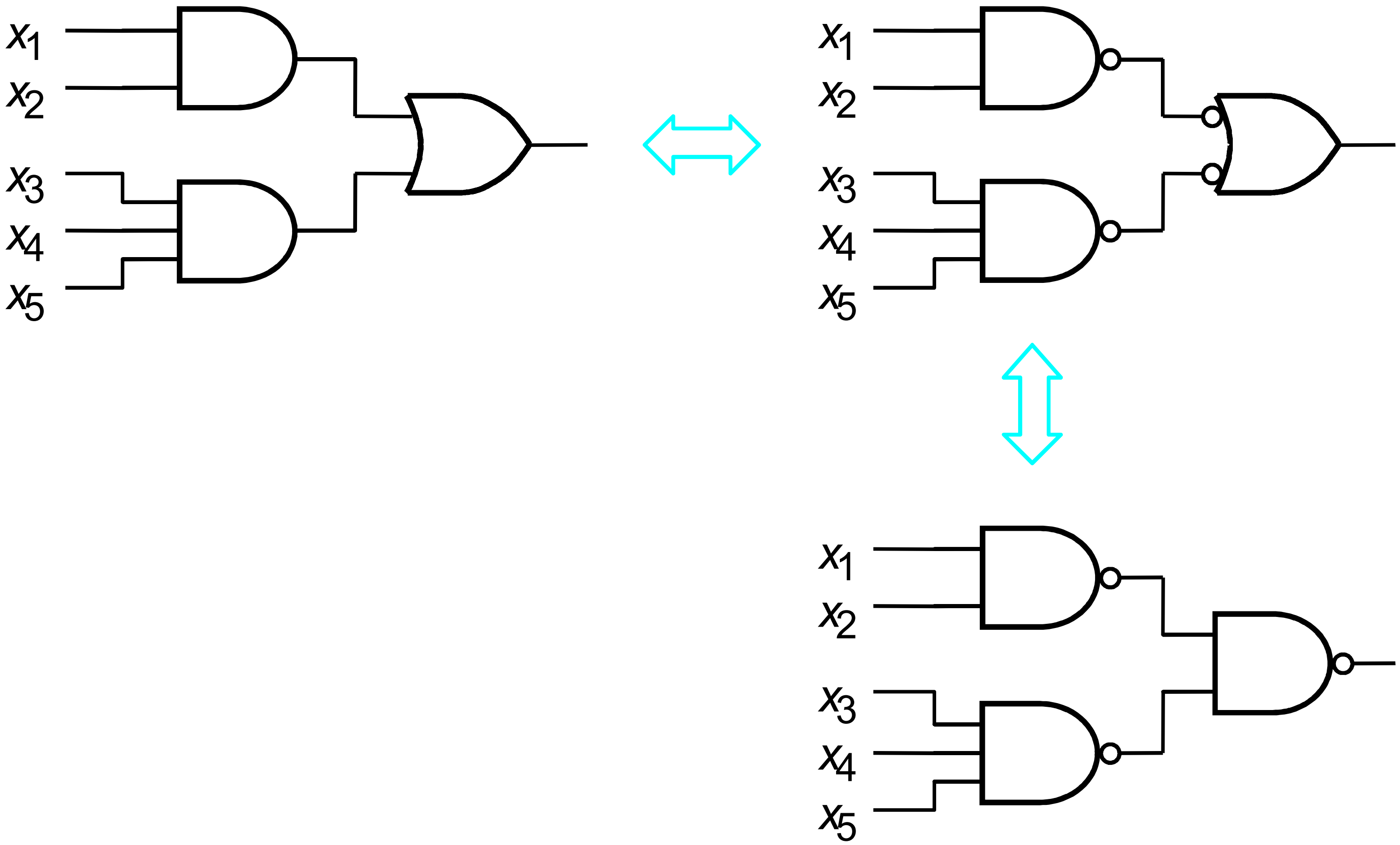
$\mathbf{f <> g}$

# Circuitos com NANDs

Transforme o circuito abaixo em um equivalente, somente com NANDs

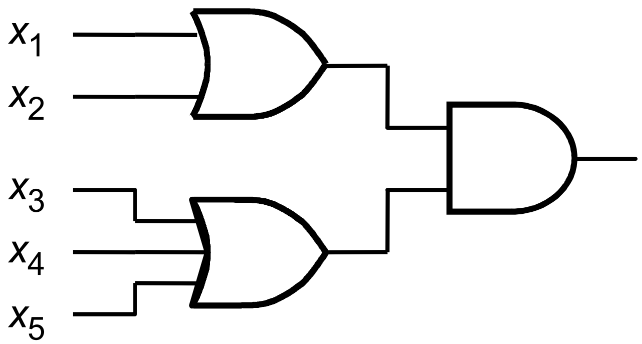


# Circuitos com NANDs

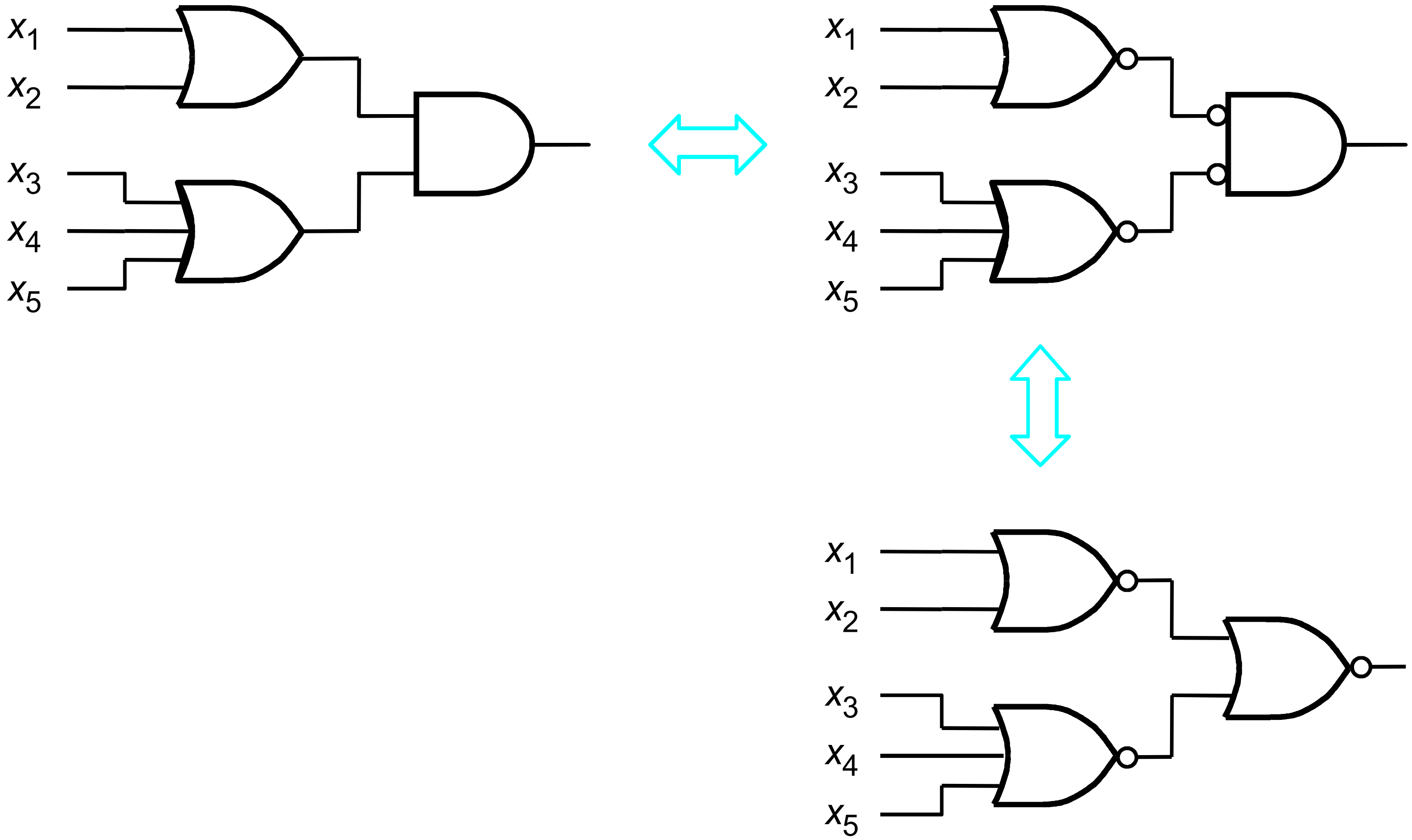


# Circuitos com NORs

Transforme o circuito abaixo em um equivalente, somente com NORs

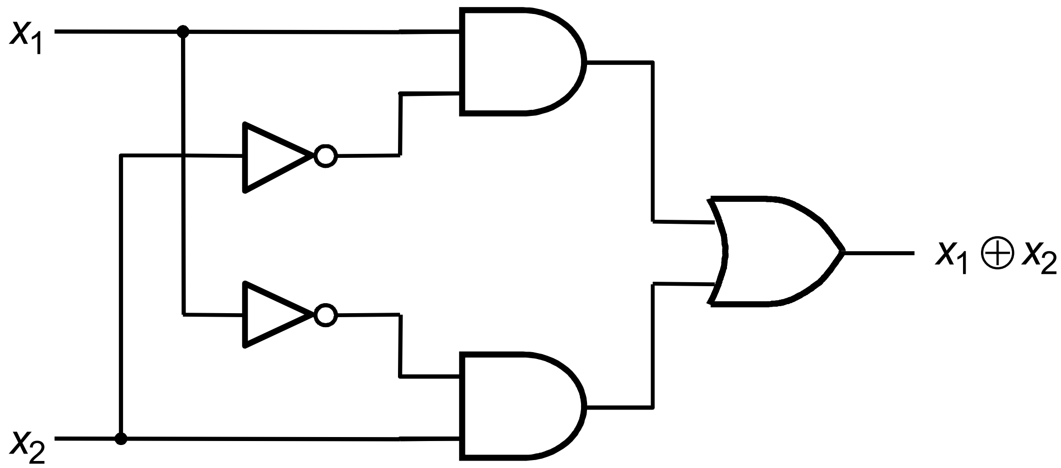


# Circuitos com NORs



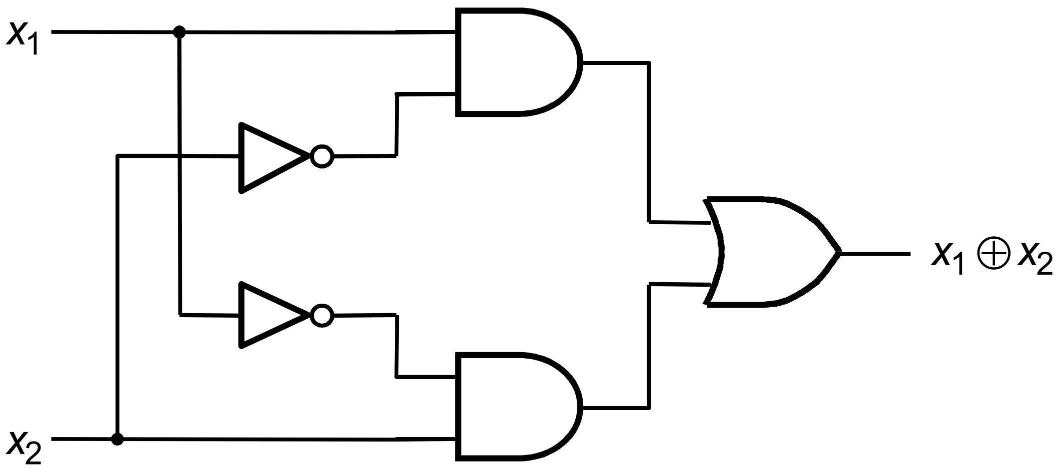
Implemente um XOR só com NANDs

# Implementação de um XOR com NANDs

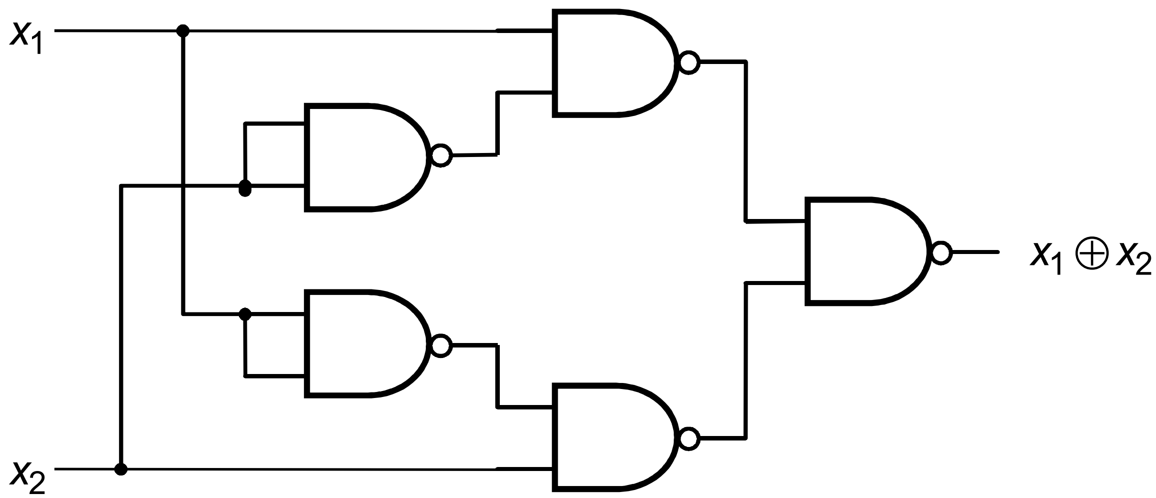



(a) Implementação SOP

# Implementação de um XOR com NANDs




(a) Implementação SOP




(b) Impemantação com NAND

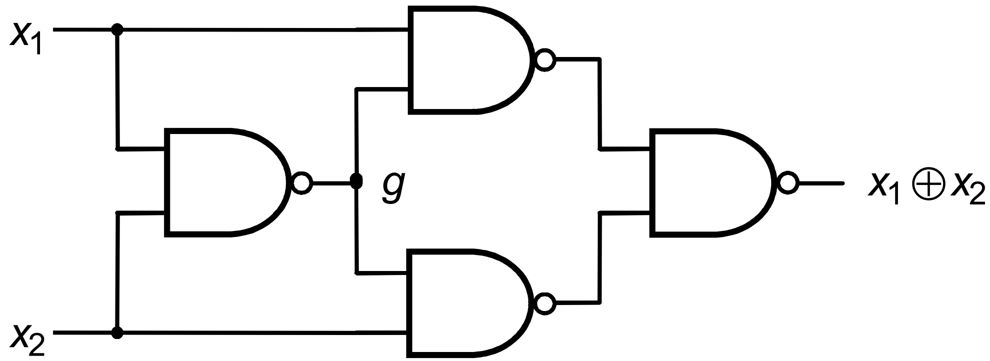
Implementação otimizada de um XOR com NANDs

$$f = x_1 \oplus x_2 = x_1\overline{x}_2 + \overline{x}_1 x_2$$

Figure 4.29 b Implementation of XOR

# Implementação otimizada de um XOR com NANDs

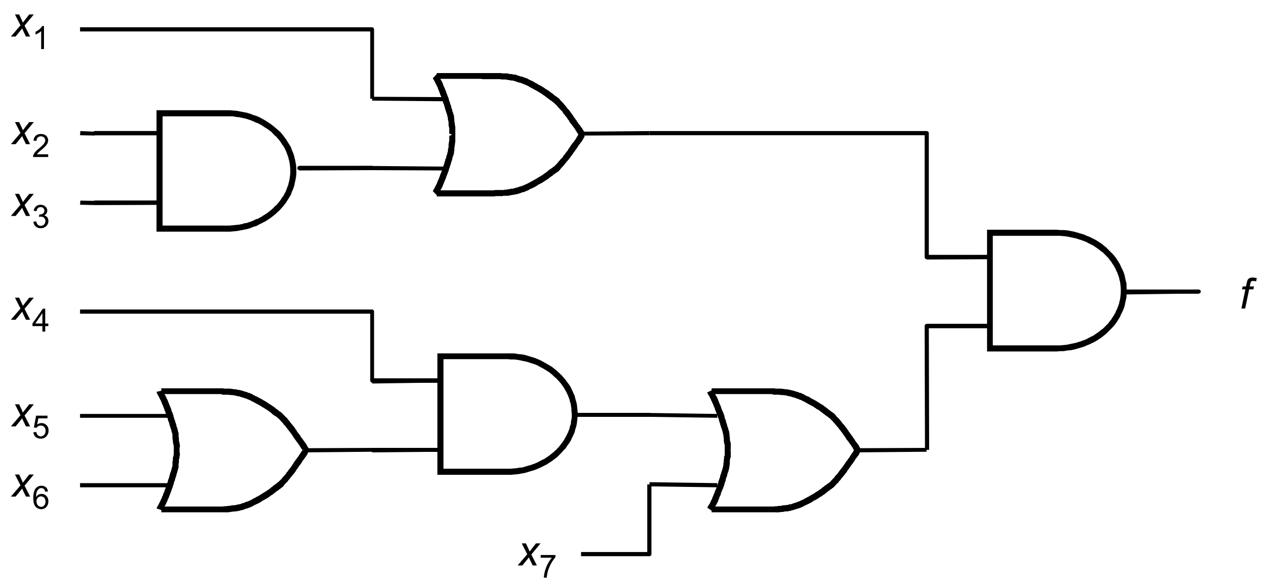
$$f = x_1 \oplus x_2 = x_1\bar{x}_2 + \bar{x}_1 x_2 = x_1(\bar{x}_1 + \bar{x}_2) + x_2(\bar{x}_1 + \bar{x}_2)$$




(c) Implementação otimizada com NAND

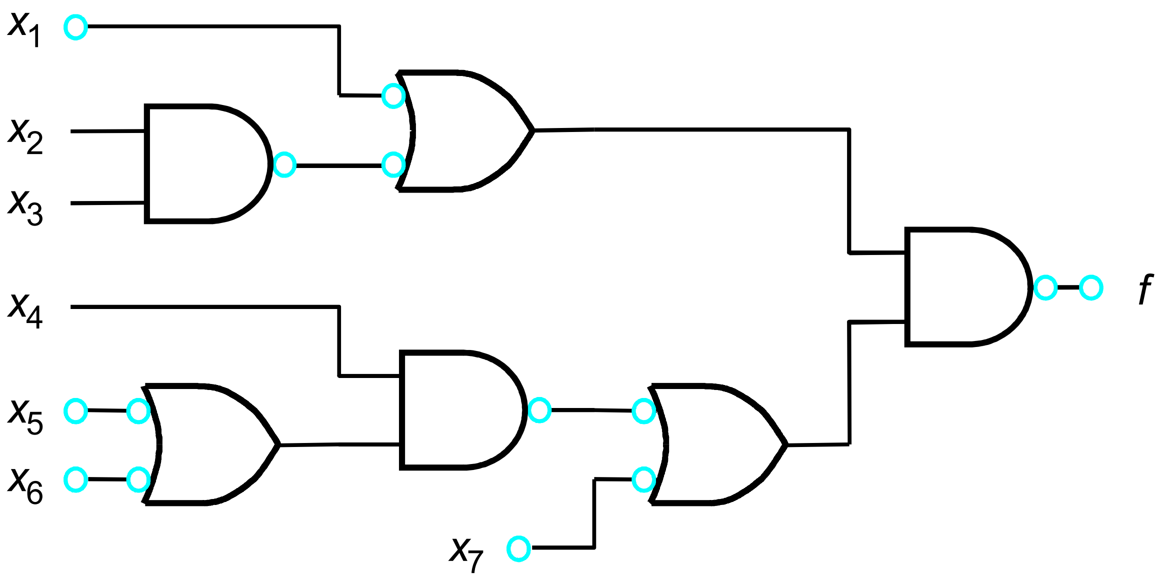
Figure 4.29 b Implementation of XOR

Conversão para um circuito com ANDs e ORs para NANDs



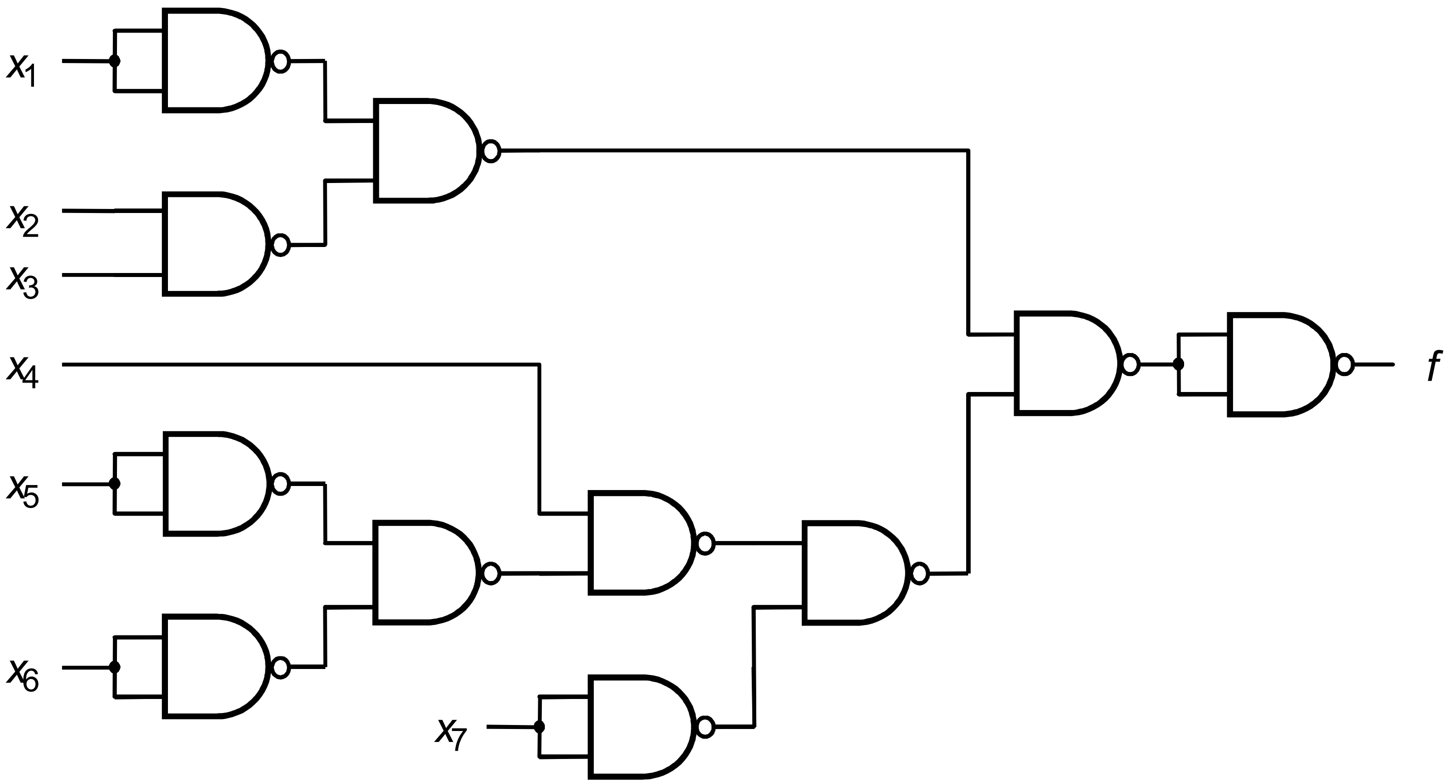

(a) Circuito com portas AND e OR

# Conversão para um circuito com ANDs e ORs para  NANDs

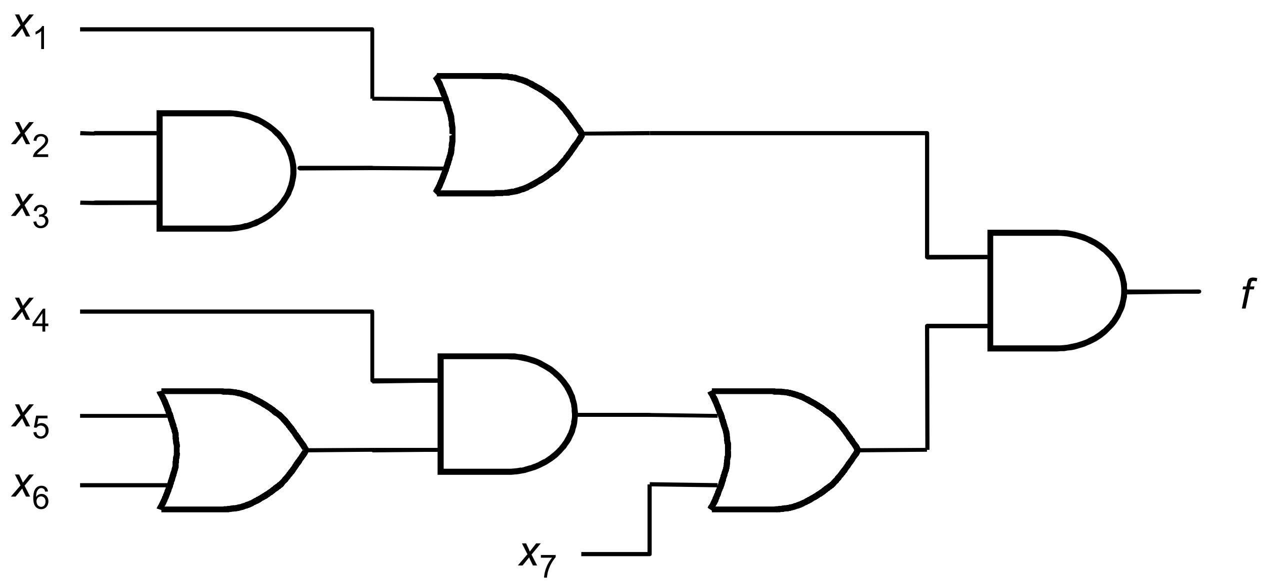



(b) Inversões necessárias para converter em NANDs

Conversão para um circuito com NANDs

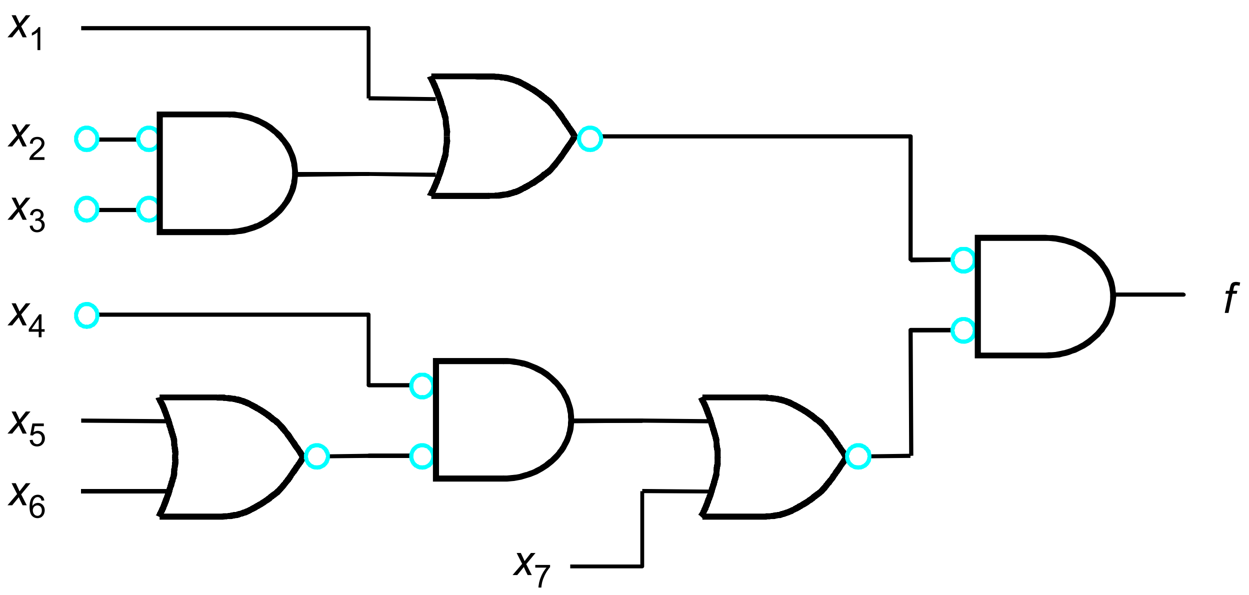


# Conversão para um circuito com ANDs e ORs para NORs



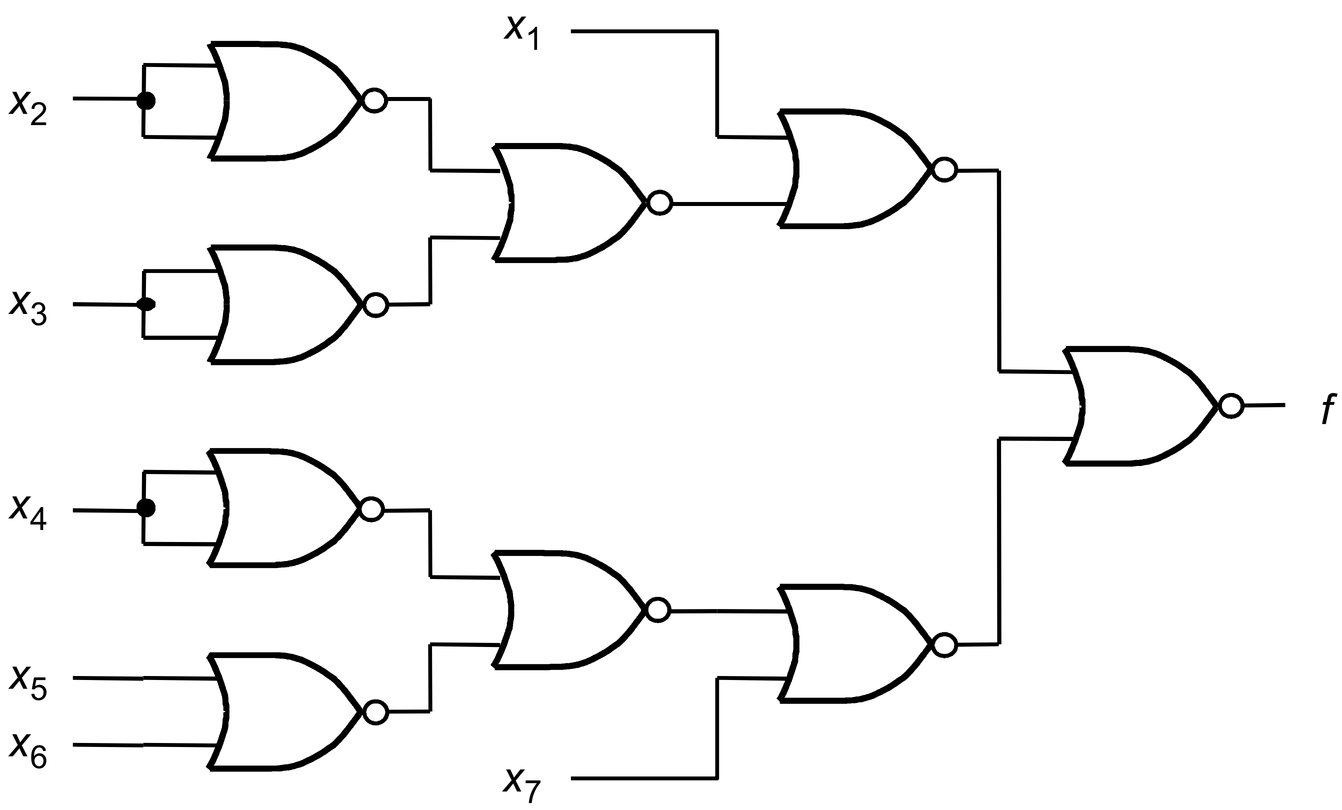

(a) Circuito com portas AND e OR

# Conversão para um circuito com ANDs e ORs para NORs



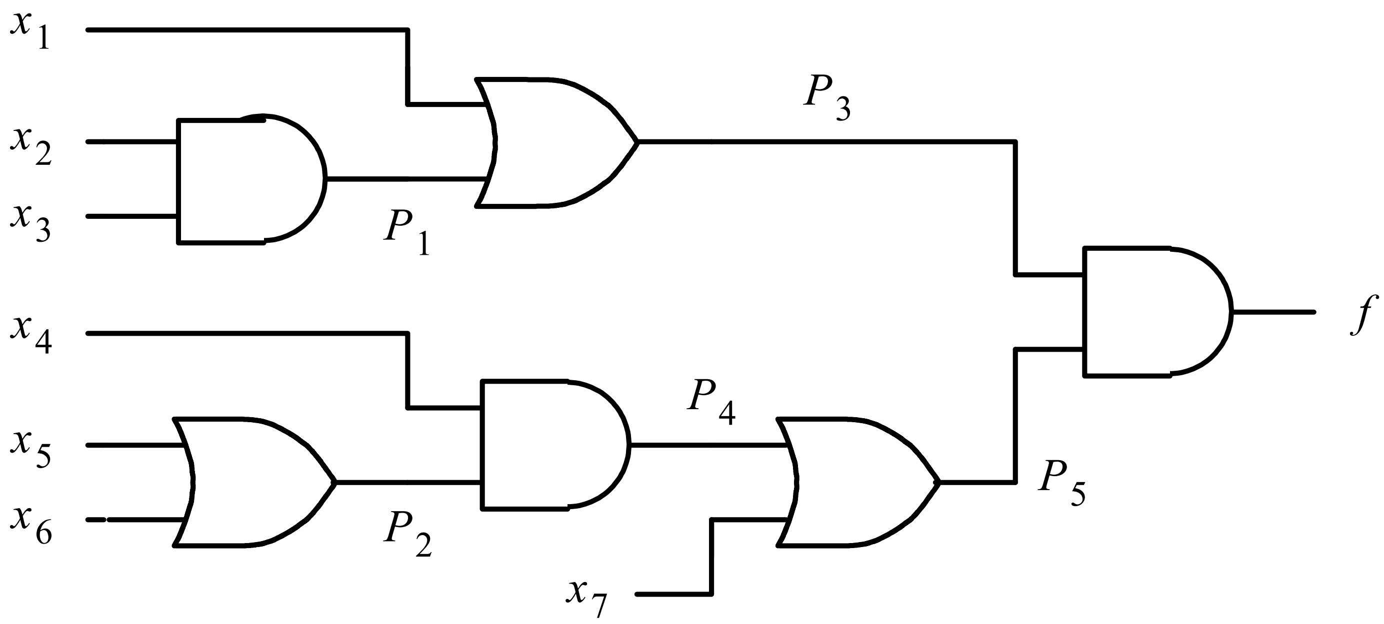

(a) Inversões necessárias para converter em NORs

# Conversão para um circuito com ANDs e ORs para NORs



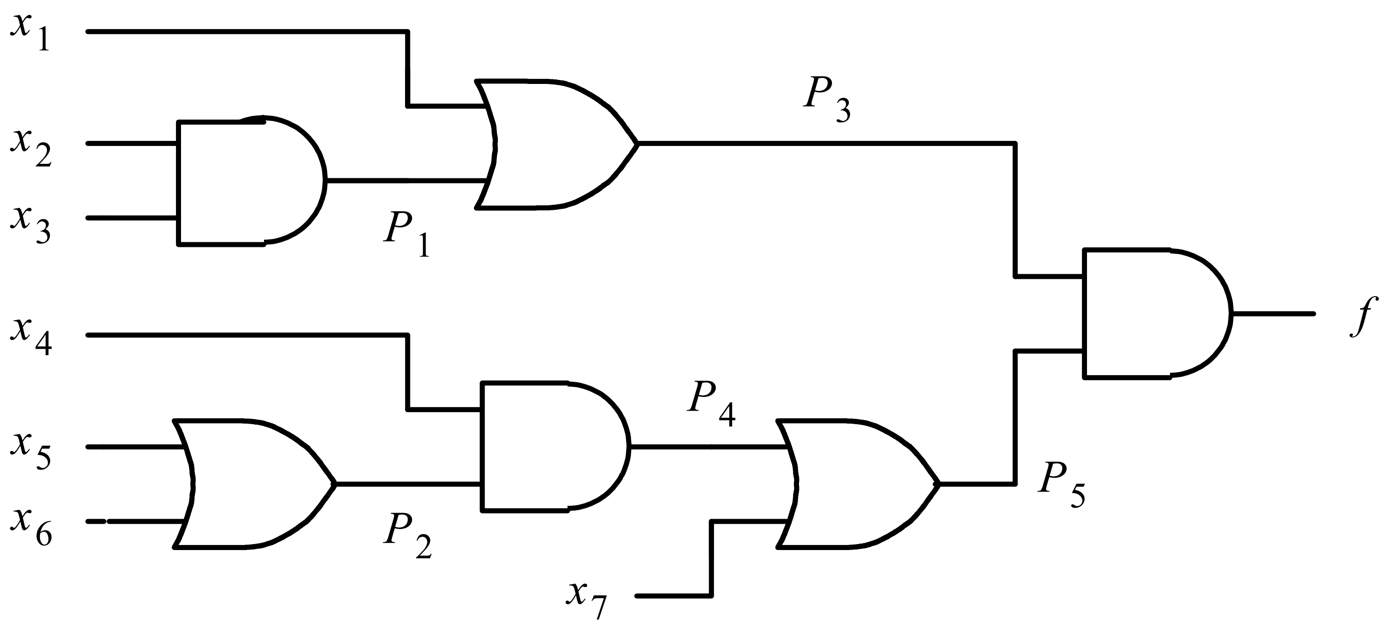
# Análise de circuitos multi-níveis

Exemplo: Escrever a função f



# Análise de circuitos multi-níveis

Exemplo: Escrever a função f




P1 = x2 x3

P3 = P1 + x1 = x2 x3 + x1

P2 = x5 + x6

P4 = P2 . x4 = x5 x4 + x6 x4

P5 = P4 + x7 = x5x4 + x6 x4 + x7

f = P3 . P5 = (x2 x3 + x1 ) . (x5x4 + x6 x4 + x7)

Exemplo de código VHDL para a função $f = \Sigma\, m(1, 4, 5, 6)$

```
ENTITY func1 IS
      PORT ( x1, x2, x3   : IN     BIT ;
                   f                : OUT  BIT ) ;
END func1 ;

ARCHITECTURE LogicFunc OF func1 IS
BEGIN
      f <= (NOT x1 AND NOT x2 AND x3) OR
             (x1 AND NOT x2 AND NOT x3) OR
             (x1 AND NOT x2 AND x3) OR
             (x1 AND x2 AND NOT x3) ;
END LogicFunc ;
```

Código VHDL usando STD_LOGIC

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY func2 IS
	PORT ( x1, x2, x3	: IN	STD_LOGIC ;
	       f		: OUT	STD_LOGIC ) ;
END func2 ;

ARCHITECTURE LogicFunc OF func2 IS
BEGIN
	f <= (NOT x1 AND NOT x2 AND x3) OR
	     (x1 AND NOT x2 AND NOT x3) OR
	     (x1 AND NOT x2 AND x3) OR
	     (x1 AND x2 AND NOT x3) ;
END LogicFunc ;
```

Exemplo de código VHDL para a função $f = \Sigma\, m(0, 2, 4, 5, 6)$

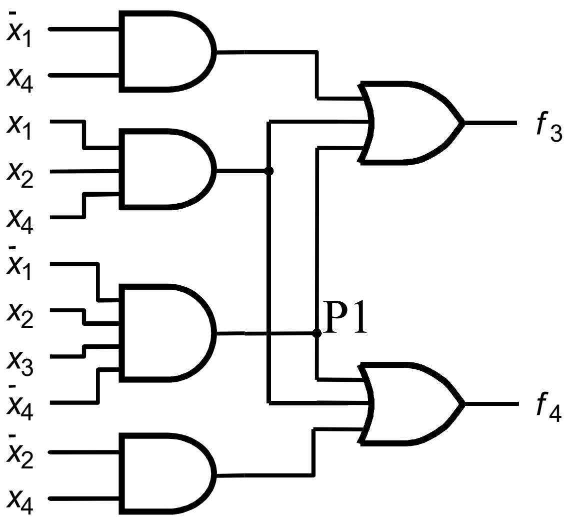
```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY func3 IS
    PORT ( x1, x2, x3  : IN    STD_LOGIC ;
           f           : OUT   STD_LOGIC ) ;
END func3 ;

ARCHITECTURE LogicFunc OF func3 IS
BEGIN
    f <= (NOT x1 AND NOT x2 AND NOT x3) OR
         (NOT x1 AND x2 AND NOT x3) OR
         (x1 AND NOT x2 AND NOT x3) OR
         (x1 AND NOT x2 AND x3) OR
         (x1 AND x2 AND NOT x3) ;
END LogicFunc ;
```

Exemplo de código VHDL para a função $f = \Sigma\, m(2, 3, 9, 10, 11, 13)$

```
LIBRARY ieee ;
USE ieee.std logic 1164.all ;

ENTITY func4 IS
      PORT ( x1, x2, x3, x4   : IN    STD LOGIC ;
                 f                        : OUT  STD LOGIC ) ;
END func4 ;

ARCHITECTURE LogicFunc OF func4 IS
BEGIN
      f <= (NOT x1 AND NOT x2 AND x3 AND NOT x4) OR
              (NOT x1 AND NOT x2 AND x3 AND x4) OR
              (x1 AND NOT x2 AND NOT x3 AND x4) OR
              (x1 AND NOT x2 AND x3 AND NOT x4) OR
              (x1 AND NOT x2 AND x3 AND x4) OR
              (x1 AND x2 AND NOT x3 AND x4) ;
END LogicFunc ;
```

Exemplo de código VHDL para uma função com 7 variáveis

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY func5 IS
    PORT ( x1, x2, x3, x4, x5, x6, x7  : IN   STD_LOGIC ;
           f                          : OUT  STD_LOGIC ) ;
END func5 ;

ARCHITECTURE LogicFunc OF func5 IS
BEGIN
    f <= (x1 AND x3 AND NOT x6) OR
         (x1 AND x4 AND x5 AND NOT x6) OR
         (x2 AND x3 AND x7) OR
         (x2 AND x4 AND x5 AND x7) ;
END LogicFunc ;
```




```
LIBRARY ieee
USES ieee.std_logic_1164.all
ENTITY exemplo IS
  PORT ( x1,x2,x3,x4   : IN std_logic;
              f3,f4              : OUT std_logic);
  END  exemplo;
ARCHITECTURE função OF exemplo IS
         signal P1 : std_logic;
BEGIN
      P1 <= NOT x1 AND x2 AND x3 AND NOT x4;
      f3 <=  (NOT x1 AND X4) OR (x1 AND x2 AND X4)
                 OR P1;
       f4 <= (NOT x2 AND X4) OR (x1 AND x2 AND X4)
                 OR P1;
END funcao;
```

Exercícios para fazer fora do horário de aula

Exercícios para fazer fora do horário de aula

## Mapa de Karnaugh - Exercícios

*1)* $f = \Sigma\, m(0, 1, 2, 3, 7)$

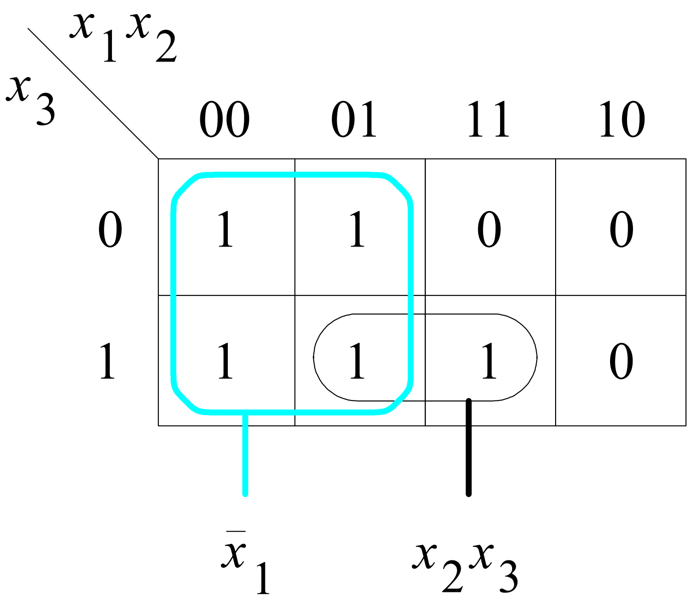
*2)* $f = \Sigma\, m(2, 3, 5, 6, 7, 10, 11, 13, 14)$

*3)* $f = \Sigma\, m(0, 4, 8, 10, 11, 12, 13, 15)$

*4)* $f = \Sigma\, m(0, 2, 4, 5, 10, 11, 13, 15)$

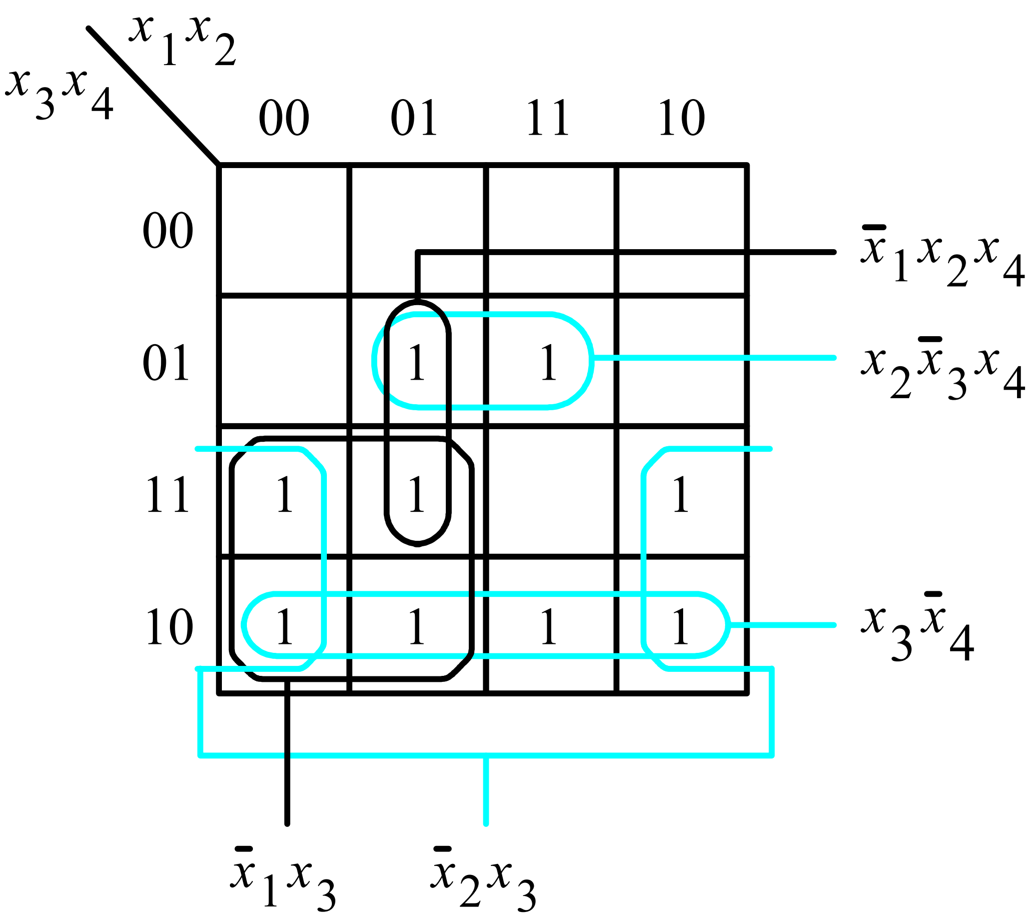
# Mapa de Karnaugh - Exercícios

1) $f = \Sigma\, m(0, 1, 2, 3, 7)$ = x1' +x2x3



# Mapa de Karnaugh - Exercícios

2) $f = \Sigma\, m(2, 3, 5, 6, 7, 10, 11, 13, 14)$ = x1'x3 + x3 x4' +x2'x3 +x2x3'x4

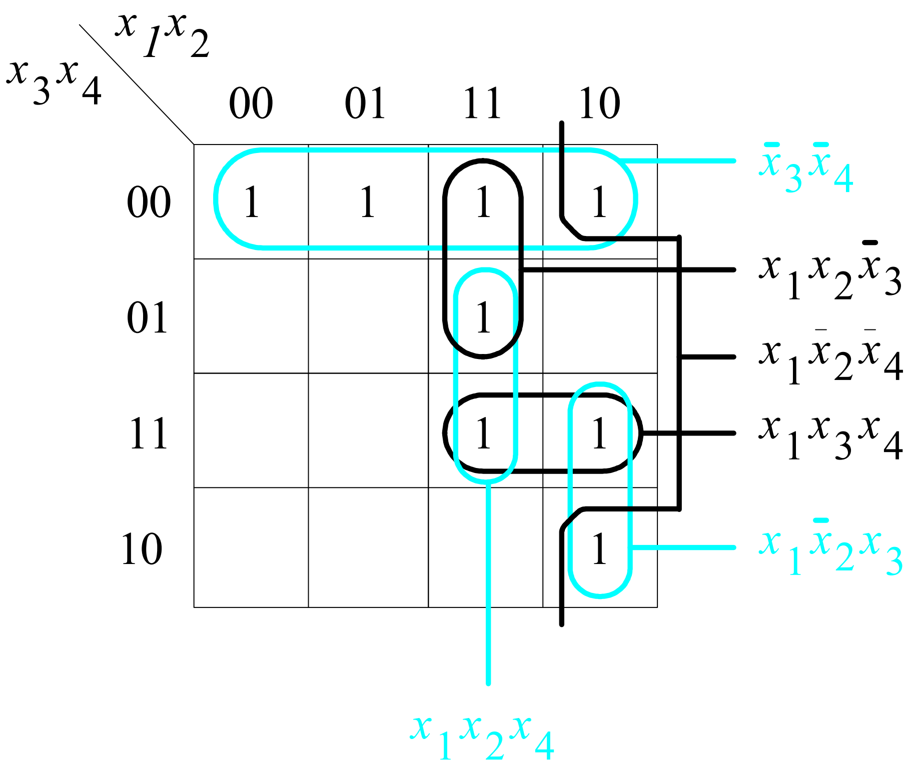


| | | | |
|---|---|---|---|
| | | | |
| | 1 | 1 | |
| 1 | 1 | | 1 |
| | | | |

# Mapa de Karnaugh - Exercícios

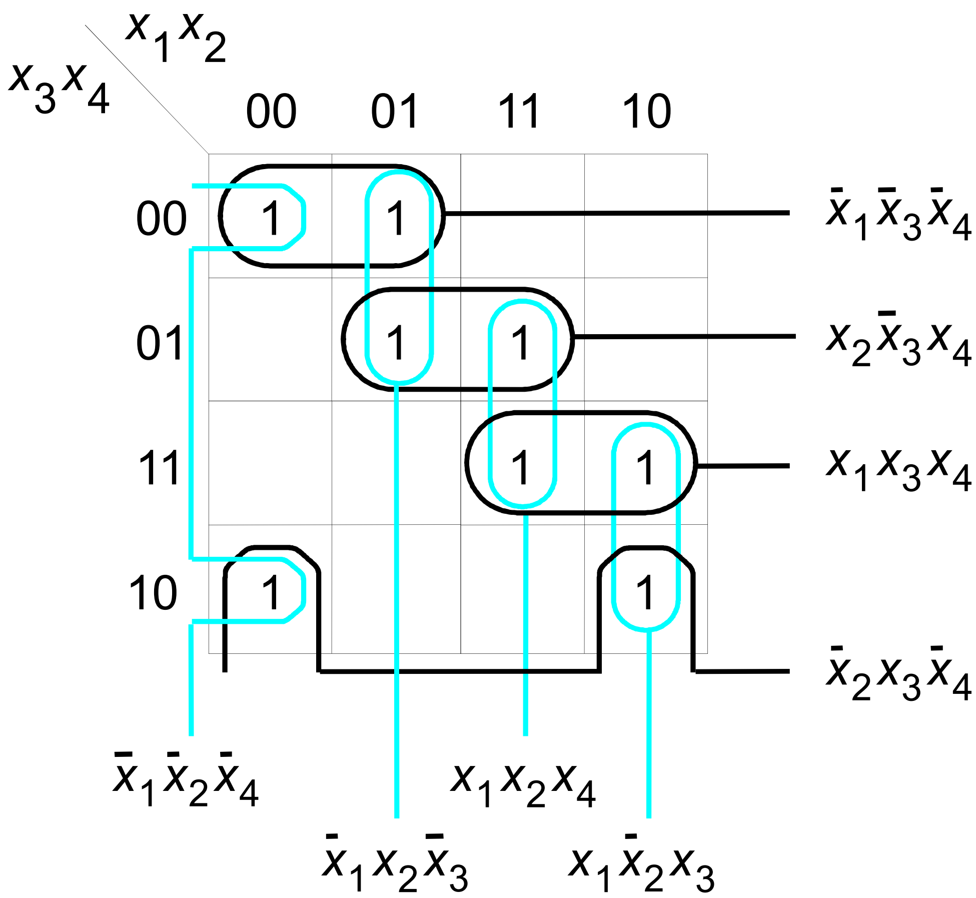
3) $f = \Sigma\, m(0, 4, 8, 10, 11, 12, 13, 15)$ = x3'x4' + x1x2x4 + x1x2'x3

=x3'x4' + x1x2x3'+x1x3x3+x1x2'x3



# Mapa de Karnaugh - Exercícios

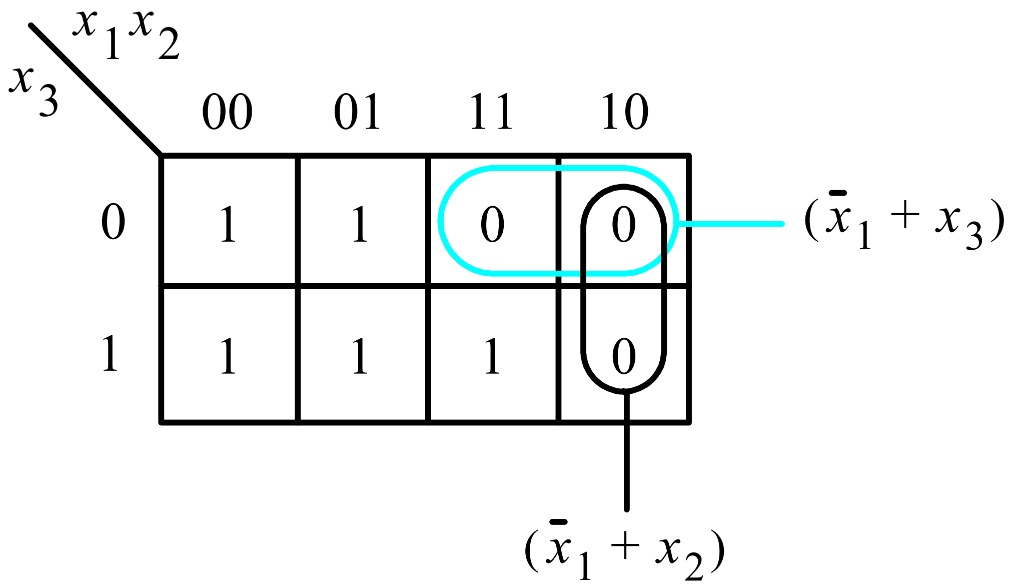
4) $f = \Sigma\, m(0, 2, 4, 5, 10, 11, 13, 15)$




.

# Mapa de Karnaugh – Minimização usando POS

Função $f = \Pi\ M(4, 5, 6)$




OBS. -

Para se obter a implementação POS de custo mínimo, a partir do SOP → fazer $f = \overline{\overline{f}}$

# Mapa de Karnaugh - Exercícios

Minimizar a função $f = \Pi\, M(0, 1, 4, 8, 9, 12, 15)$ e obter a implementação POS de custo mínimo.

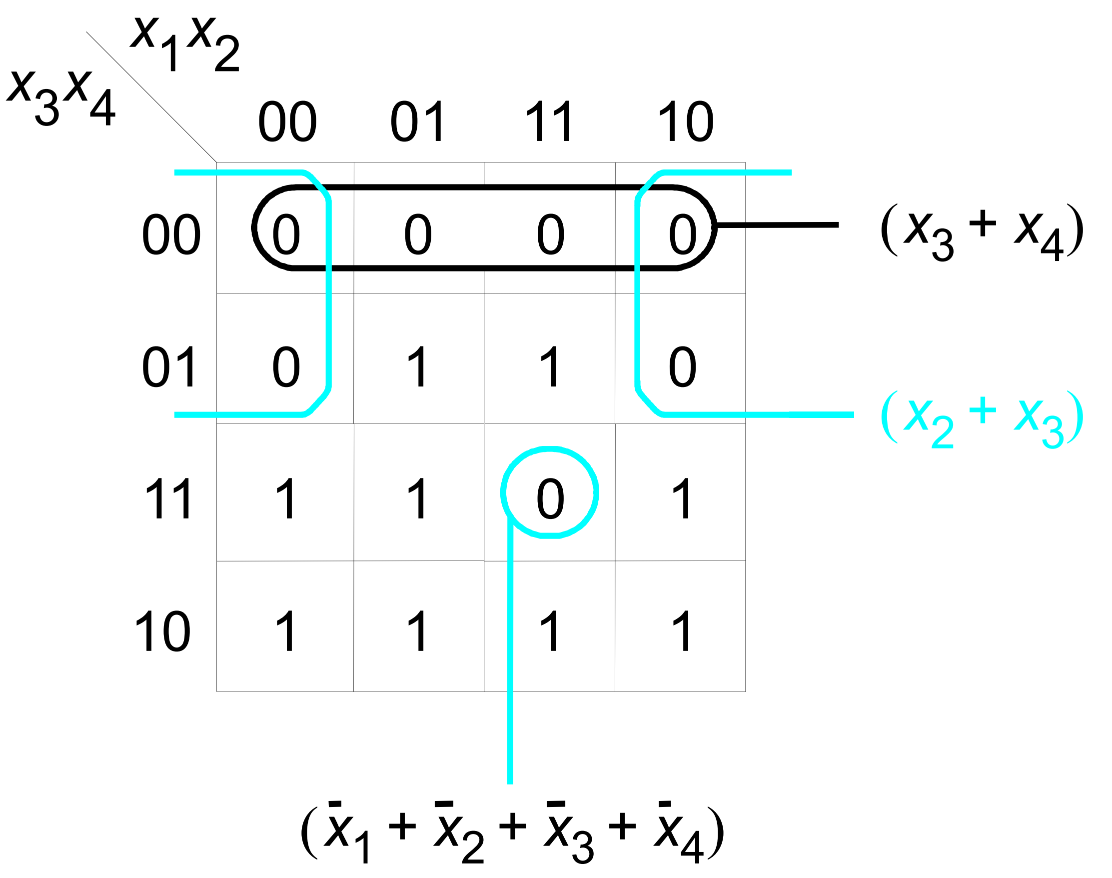
# Mapa de Karnaugh - Exercícios

$f = \Pi\, M(0, 1, 4, 8, 9, 12, 15) = (x3 + x4) \,.\, (x2 + x3) \,.\, (x1' + x2' + x3' + x4')$

Custo = 3 OR + 8 + 1 AND + 3 = 15

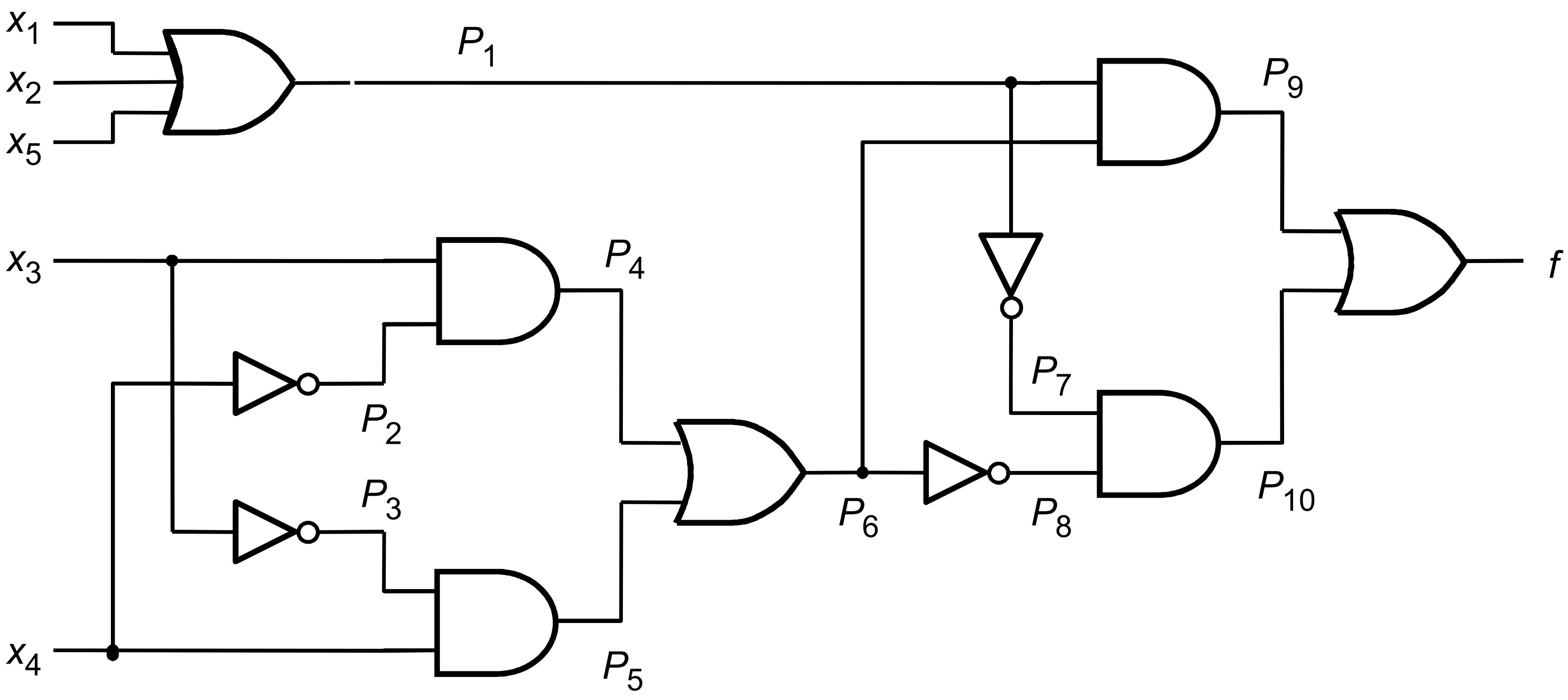
$f = \Sigma\, m(2,3,5,6,7,10,11,13,14) = x3x4' + x1'x3 + x2'x3 + x2x3'x4$

Custo = 4 AND + 9 + 1 OR + 4 = 18



# Exercícios para fazer fora do horário de aula

Exemplo: Escrever a função f



Exemplo: Escrever a função f