Introdução aos Sistemas Digitais

Algebra de Boole
Postulados
Teoremas e expressões
Simplificação algébrica

Circuitos combinatórios

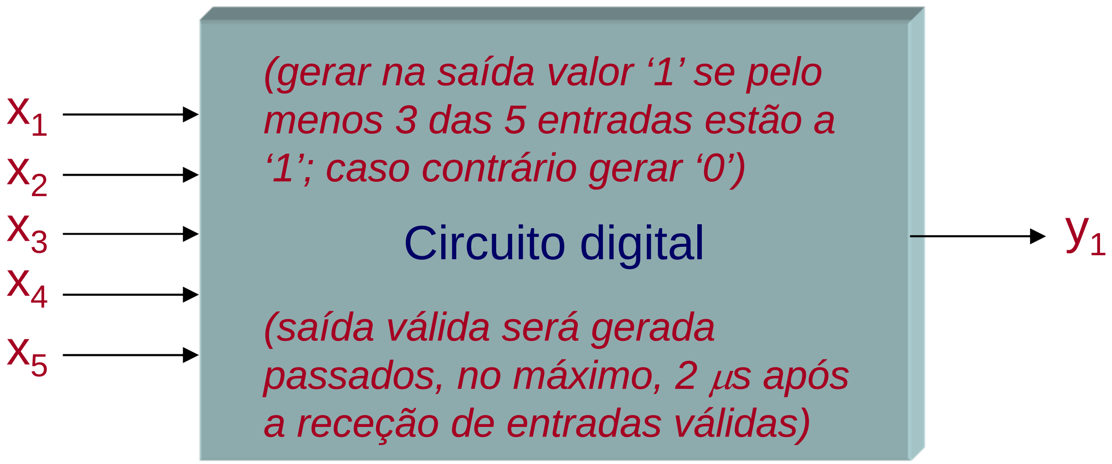
Um circuito digital combinatório possui:

- uma ou mais entradas digitais;

- uma ou mais saídas digitais;

- uma especificação funcional que descreve cada saída em função dos valores das entradas;

- uma especificação temporal que inclui, pelo menos, o tempo máximo que o circuito vai demorar para produzir valores de saída a partir de um conjunto arbitrário de valores de entrada (válidos e estáveis) -> tempo de propagação.



Álgebra de Boole

Álgebra de Boole binária - é um instrumento matemático que permite descrever relações funcionais entre as entradas e saídas de um circuito digital.

Álgebra de Boole é uma estrutura matemática baseada num conjunto {B,+, ·}, satisfazendo o seguinte conjunto de postulados:

1. Fecho (as operações são fechadas em B)
2. Comutatividade
3. Elementos neutros
4. Distributividade mútua
5. Complementação
6. Cardinalidade

Postulados

P1 - fecho: ambas as operações são fechadas em B:

$$\forall b_1, b_2 \in B \quad \begin{cases} b_1 + b_2 \in B \\ b_1 \cdot b_2 \in B \end{cases}$$

P2 - comutatividade

$$\forall b_1, b_2 \in B \quad \begin{cases} b_1 + b_2 = b_2 + b_1 \\ b_1 \cdot b_2 = b_2 \cdot b_1 \end{cases}$$

P3 – elementos neutros

$$\exists 0 \ \forall b \in B: \quad b + 0 = b$$

$$\exists 1 \ \forall b \in B: \quad b \cdot 1 = b$$

Postulados (cont.)

P4 – distributividade mútua

$$\forall_{b_1, b_2, b_3 \in B} \quad b_1 + (b_2 \cdot b_3) = (b_1 + b_2) \cdot (b_1 + b_3)$$

$$\forall_{b_1, b_2, b_3 \in B} \quad b_1 \cdot (b_2 + b_3) = (b_1 \cdot b_2) + (b_1 \cdot b_3)$$

P5 - complementação

$$\forall b \in B \exists \overline{b} \in B \quad \begin{cases} b + \overline{b} = 1 \\ b \cdot \overline{b} = 0 \end{cases}$$

P6 – cardinalidade

$$\forall_{b_1 \in B} \; \exists_{b_2 \in B} \; b_1 \neq b_2 \Leftrightarrow \# B \geq 2$$

Álgebra de Boole binária

Se #B = 2, temos álgebra de Boole a dois valores (B={0,1}).

Operadores:

soma lógica **OR**	produto **AND**	complementação **NOT**
$0 + 0 = 0$ $0 + 1 = 1$ $1 + 0 = 1$ $1 + 1 = 1$	$0 \cdot 0 = 0$ $0 \cdot 1 = 0$ $1 \cdot 0 = 0$ $1 \cdot 1 = 1$	$\overline{0} = 1$ $\overline{1} = 0$

Expressões - conjunto de variáveis e/ou constantes 0 e 1 associadas por operadores

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

Teoremas

Idempotência $\forall b \in B \quad b \cdot b = b$ e $b + b = b$

$$b \cdot b \overset{P3}{=} b \cdot b + 0 \overset{P5}{=} b \cdot b + b \cdot \overline{b} \overset{P4}{=} b \cdot (b + \overline{b}) \overset{P5}{=} b \cdot 1 \overset{P3}{=} b$$

Unicidade do elemento neutro

Sejam 0_a e 0_b tal que $b + 0_a = b + 0_b = b$

$0_a + 0_b = 0_a$ P3

$0_b + 0_a = 0_b$ P3 $\Rightarrow 0_a = 0_b$

$0_a + 0_b = 0_b$ P2

Teoremas (cont.)

Unicidade do complemento

Elemento absorvente $\forall b \in B \quad b \cdot 0 = 0$

$$b + 1 = 1$$

$$b \cdot 0 \overset{P3}{=} b \cdot 0 + 0 \overset{P5}{=} b \cdot 0 + b \cdot \overline{b} \overset{P4}{=} b \cdot (0 + \overline{b}) \overset{P3}{=} b \cdot \overline{b} \overset{P5}{=} 0$$

Absorção $\forall x,y \in B \quad x + x \cdot y = x$

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

$$x + x \cdot y \overset{P3}{=} x \cdot 1 + x \cdot y \overset{P4}{=} x \cdot (1 + y) = x \cdot 1 \overset{P3}{=} x$$

Simplificação $\forall x,y \in B \quad x + \overline{x} \cdot y = x + y$

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

Teoremas (cont.)

Adjacência $\forall x,y \in B$ $x \cdot y + x \cdot \overline{y} = x$

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

$$x \cdot y + x \cdot \overline{y} \overset{P4}{=} x \cdot (y + \overline{y}) \overset{P5}{=} x \cdot 1 \overset{P3}{=} x$$

Involução $\forall x \in B$ $\overline{\overline{x}} = x$

indução perfeita:

x	$\overline{x}$	$\overline{\overline{x}}$
0	1	0
1	0	1

Consenso $\forall x,y,z \in B$ $x \cdot y + \overline{x} \cdot z + y \cdot z = x \cdot y + \overline{x} \cdot z$

$(x + y) \cdot (\overline{x} + z) \cdot (y + z) = (x + y) \cdot (\overline{x} + z)$

Associatividade $\forall x,y,z \in B$ $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

$(x + y) + z = x + (y + z)$

Leis de DeMorgan

$\forall x,y \in B \quad \overline{x+y} = \bar{x} \cdot \bar{y} \quad$ e $\quad \overline{x \cdot y} = \bar{x} + \bar{y}$

P4

P5, elemento absorvente, idempotência

$(x+y) \cdot (\bar{x} \cdot \bar{y}) = x \cdot \bar{x} \cdot \bar{y} + \bar{x} \cdot y \cdot \bar{y} = 0$

Generalização para *n* variáveis:

$$\overline{\sum_{i=1}^{n} x_i} = \prod_{i=1}^{n} \overline{x_i} \qquad \overline{\prod_{i=1}^{n} x_i} = \sum_{i=1}^{n} \overline{x_i}$$

$$\overline{F(x_1, x_2, \ldots, x_n, +, \bullet)} = F(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n, \bullet, +)$$

Princípio da dualidade

Todo o teorema ou identidade algébrica dedutível a partir dos postulados da álgebra de Boole conserva a validade se as operações (+) e (.) e os elementos neutros forem trocados.

$$F^D(x_1, x_2, \ldots, x_n, +, \cdot, 0, 1) = F(x_1, x_2, \ldots, x_n, \cdot, +, 1, 0)$$

$$\overline{F(x_1, x_2, \ldots, x_n)} = F^D(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$$

Exemplos:

$[\,(x + \bar{y}) \cdot (z + 1)\,]^D = (x \cdot \bar{y}) + (z \cdot 0)$

$$\overline{(x_1 + x_2) \cdot \bar{x}_3} = \bar{x}_1 \cdot \bar{x}_2 + x_3$$

Conjunto completo de operadores

- conjunto de operadores a partir dos quais se pode representar qualquer função booleana.

{ AND, OR, NOT }
{ AND, NOT }
{ OR, NOT }
{ NAND }
{ NOR }
...

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

x	y	x NAND y
0	0	1
0	1	1
1	0	1
1	1	0

$$\overline{x + y}$$

x	y	x NOR y
0	0	1
0	1	0
1	0	0
1	1	0

Operadores NAND e NOR

Para escrever uma expressão booleana apenas com operadores NAND deve-se primeiro colocá-la na forma da soma de produtos e a seguir aplicar o teorema de involução e as leis de DeMorgan

Para escrever uma expressão booleana apenas com operadores NOR deve-se primeiro colocá-la na forma do produto de somas e a seguir aplicar o teorema de involução e as leis de DeMorgan

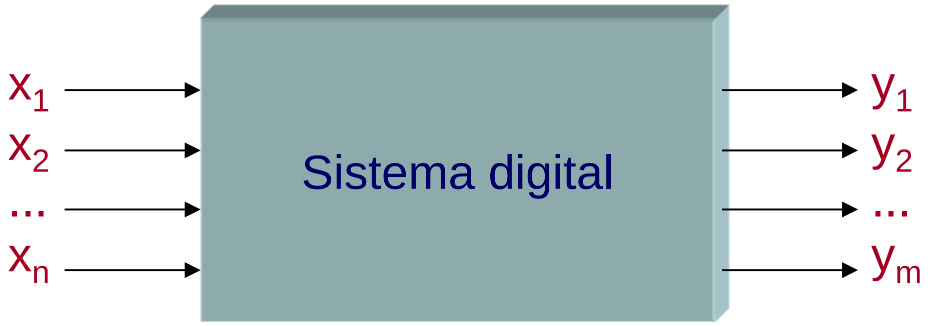
Exemplos:

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

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

Funções booleanas

Uma função booleana é uma correspondência que associa um elemento do conjunto B={0,1} a cada uma das 2^n combinações possíveis que as variáveis podem assumir.



Existem $2^{m\times 2^n}$ funções booleanas diferentes que podem ser implementadas num sistema digital com *n* entradas e *m* saídas.

Exemplos:

Para n=1, m=1: $2^{1\times 2^1} = 4$

x	constante '0'	x	$\overline{x}$	constante '1'
0	0	0	1	1
1	0	1	0	1

Para n=4, m=3: $2^{3\times 2^4} = 2^{48}$= 281 474 976 710 656

Funções duais

Para obter a função dual de *f*, deve-se aplicar o princípio de dualidade a *f*.

$$f^D(x_1, x_2, \ldots, x_n, +, \cdot, 0, 1) = f(x_1, x_2, \ldots, x_n, \cdot, +, 1, 0)$$

Uma função *f* é auto-dual se *f* = *f*D => $f(x_1, x_2, \ldots, x_n) = \overline{f(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)}$

Exemplo:

$$f(x, y, z) = x \cdot y + x \cdot \overline{z} + y \cdot \overline{z}$$

$$f^D(x, y, z) = (x + y) \cdot (x + \overline{z}) \cdot (y + \overline{z}) =$$

$$= (x + y \cdot \overline{z}) \cdot (y + \overline{z}) =$$

$$= x \cdot y + x \cdot \overline{z} + y \cdot \overline{z} = f(x, y, z)$$

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Representação de funções

- tabular (tabela de verdade)
- algébrica
- esquemática (circuitos lógicos)

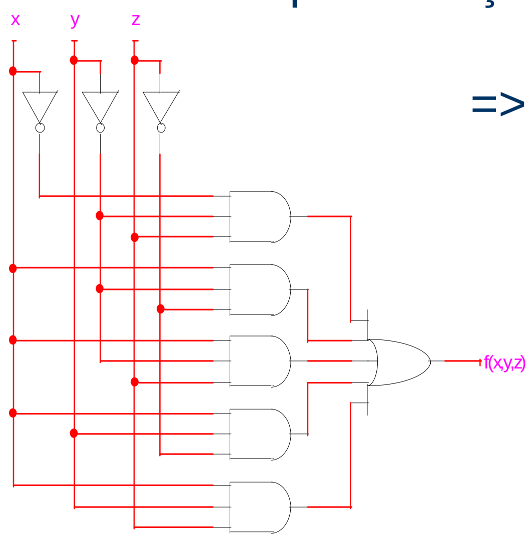
Representação tabular é única:

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Representação algébrica inclui frequentemente termos redundantes:

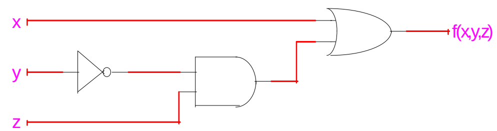
$$f(x,y,z) = \bar{x}\cdot\bar{y}\cdot z + x\cdot\bar{y}\cdot\bar{z} + x\cdot\bar{y}\cdot z + x\cdot y\cdot\bar{z} + x\cdot y\cdot z$$

Representação esquemática:



=> necessidade de simplificação

$$f(x,y,z) = x + \bar{y}\cdot z$$



Exercícios

Prove o teorema de simplificação: $\forall x,y \in B \quad x + \overline{x} \cdot y = x + y$

Prove o teorema de consenso: $\forall x,y,z \in B \quad x \cdot y + \overline{x} \cdot z + y \cdot z = x \cdot y + \overline{x} \cdot z$

Prove que o operador NAND é funcionalmente completo

Quantas funções booleanas diferentes se pode implementar num sistema digital com *2* entradas e *1* saída

As expressões seguintes estão corretas?

$$[x + y \cdot z]^D = x \cdot y + z \qquad\qquad [x + y \cdot z]^D = \overline{x} \cdot (\overline{y} + \overline{z})$$

Verifique se as funções seguintes são auto-duais:

$$f_1(x, y, z) = \overline{x} \cdot y + \overline{x} \cdot z + y \cdot z$$

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

Exercícios (cont.)

Exprima a função *y* na forma mais simples recorrendo a operadores NAND.

$$y = x_1 \cdot (x_2 + \bar{x}_3 \cdot x_4) + x_2$$

$$y = x_1 \cdot x_2 + x_1 \cdot \bar{x}_3 \cdot x_4 + x_2 = x_2 + x_1 \cdot \bar{x}_3 \cdot x_4$$

$$y = \overline{\overline{x_2 + x_1 \cdot \bar{x}_3 \cdot x_4}} = \overline{\bar{x}_2 \cdot \overline{x_1 \cdot \bar{x}_3 \cdot x_4}}$$

Exprima a função *y* na forma mais simples recorrendo a operadores NOR.

$$y = x_1 \cdot (x_2 + \bar{x}_3 \cdot x_4) + x_2$$

$$y = (x_1 + x_2) \cdot (x_2 + \bar{x}_3 \cdot x_4 + x_2) = (x_1 + x_2) \cdot (x_2 + \bar{x}_3 \cdot x_4)$$

$$y = (x_1 + x_2) \cdot (x_2 + \bar{x}_3) \cdot (x_2 + x_4)$$

$$y = \overline{\overline{(x_1 + x_2) \cdot (x_2 + \bar{x}_3) \cdot (x_2 + x_4)}} = \overline{\overline{(x_1 + x_2)} + \overline{(x_2 + \bar{x}_3)} + \overline{(x_2 + x_4)}}$$

Introdução aos Sistemas Digitais

Teorema de Shannon
Formas canónicas
Irrelevâncias
Minimização algébrica de funções Booleanas

ieeta

Termos mínimo e máximo

Termo mínimo de ordem *i* (m_i) - produto lógico de *n* variáveis booleanas independentes, em que cada uma delas aparece uma e uma só vez, não complementada ou complementada consoante toma valores 1 ou 0, respetivamente, na i-ésima combinação das variáveis independentes.

Termo máximo de ordem *i* (M_i) - soma lógica de *n* variáveis booleanas independentes, em que cada uma delas aparece uma e uma só vez, não complementada ou complementada consoante toma valores 0 ou 1, respetivamente, na i-ésima combinação das variáveis independentes.

	x	y	z	f(x,y,z)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$$m_0 = \bar{x} \cdot \bar{y} \cdot \bar{z} \qquad M_0 = x + y + z$$

$$m_5 = x \cdot \bar{y} \cdot z \qquad M_5 = \bar{x} + y + \bar{z}$$

$$m_i = \overline{M_i} \qquad i = 0,1,\ldots,2^n - 1$$

Expansão de Shannon

Qualquer função $f(x_0, x_1,...,x_{n-1})$ pode ser representada na forma seguinte:

$$f(x_0, x_1,..., x_{n-1}) = \overline{x}_0 \cdot f(0, x_1,..., x_{n-1}) + x_0 \cdot f(1, x_1,..., x_{n-1})$$

Indução perfeita:

Se $x_0 = 0$ temos: $f(0, x_1,..., x_{n-1}) = 1 \cdot f(0, x_1,..., x_{n-1}) + 0 \cdot f(1, x_1,..., x_{n-1})$

Se $x_0 = 1$ temos: $f(1, x_1,..., x_{n-1}) = 0 \cdot f(0, x_1,..., x_{n-1}) + 1 \cdot f(1, x_1,..., x_{n-1})$

1ª forma canónica

Estendendo para 2 variáveis:

$$f(x_0, x_1, \dots, x_{n-1}) = \bar{x}_0 \cdot f(0, x_1, \dots, x_{n-1}) + x_0 \cdot f(1, x_1, \dots, x_{n-1}) =$$

$$= \bar{x}_0 \cdot \bar{x}_1 \cdot f(0,0, x_2, \dots, x_{n-1}) + \bar{x}_0 \cdot x_1 \cdot f(0,1, x_2, \dots, x_{n-1}) +$$

$$+ x_0 \cdot \bar{x}_1 \cdot f(1,0, x_2, \dots, x_{n-1}) + x_0 \cdot x_1 \cdot f(1,1, x_2, \dots, x_{n-1})$$

Continuando a expansão até x_{n-1} pode-se obter a 1ª forma canónica:

$$f(x_0, x_1, \dots, x_{n-1}) = \sum_{i=0}^{2^n-1} m_i \cdot f_i \qquad f_i = f((x_0, x_1, \dots, x_{n-1}) = i)$$

Forma normal disjuntiva DNF – *disjunctive normal form*

Formas canónicas

2ª forma canónica:

$$f(x_0, x_1, \dots, x_{n-1}) = \overline{\overline{f(x_0, x_1, \dots, x_{n-1})}} = \overline{\sum_{i=0}^{2^n-1} \overline{f}_i \cdot m_i} = \prod_{i=0}^{2^n-1} (f_i + M_i)$$

Forma normal conjuntiva CNF – *conjunctive normal form*

3ª forma canónica:

$$f(x_0, x_1, \dots, x_{n-1}) = \overline{\overline{f(x_0, x_1, \dots, x_{n-1})}} = \overline{\overline{\sum_{i=0}^{2^n-1} f_i \cdot m_i}} = \overline{\prod_{i=0}^{2^n-1} \overline{f_i \cdot m_i}}$$

4ª forma canónica:

$$f(x_0, x_1, \dots, x_{n-1}) = \overline{\overline{f(x_0, x_1, \dots, x_{n-1})}} = \overline{\overline{\prod_{i=0}^{2^n-1} f_i + M_i}} = \overline{\sum_{i=0}^{2^n-1} \overline{f_i + M_i}}$$

Implementação de funções

1ª forma canónica: $$f(x_0, x_1, \ldots, x_{n-1}) = \sum_{i=0}^{2^n-1} m_i \cdot f_i$$ AND-OR

2ª forma canónica: $$f(x_0, x_1, \ldots, x_{n-1}) = \prod_{i=0}^{2^n-1} (f_i + M_i)$$ OR-AND

3ª forma canónica: $$f(x_0, x_1, \ldots, x_{n-1}) = \overline{\prod_{i=0}^{2^n-1} \overline{f_i \cdot m_i}}$$ NAND-NAND

4ª forma canónica: $$f(x_0, x_1, \ldots, x_{n-1}) = \overline{\sum_{i=0}^{2^n-1} \overline{f_i + M_i}}$$ NOR-NOR

Formas canónicas (cont.)

Exemplo: Determinar as formas canónicas da função f(x,y,z)

$$f(x,y,z) = x \cdot y + \bar{z}$$

1ª:

$$f(x,y,z) = x \cdot y \cdot (z + \bar{z}) + \bar{z} \cdot (x + \bar{x}) \cdot (y + \bar{y}) =$$
$$= x \cdot y \cdot z + x \cdot y \cdot \bar{z} + x \cdot y \cdot \bar{z} + x \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot \bar{y} \cdot \bar{z} =$$
$$= x \cdot y \cdot z + x \cdot y \cdot \bar{z} + x \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot \bar{y} \cdot \bar{z}$$

2ª:

$$f(x,y,z) = (x + \bar{z}) \cdot (y + \bar{z}) =$$
$$= (x + y + \bar{z}) \cdot (x + \bar{y} + \bar{z}) \cdot (x + y + \bar{z}) \cdot (\bar{x} + y + \bar{z}) =$$
$$= (x + y + \bar{z}) \cdot (x + \bar{y} + \bar{z}) \cdot (\bar{x} + y + \bar{z})$$

3ª:

$$f(x,y,z) = \overline{\overline{x \cdot y \cdot z + x \cdot y \cdot \bar{z} + x \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot \bar{y} \cdot \bar{z}}} =$$
$$= \overline{\overline{(x \cdot y \cdot z)} \cdot \overline{(x \cdot y \cdot \bar{z})} \cdot \overline{(x \cdot \bar{y} \cdot \bar{z})} \cdot \overline{(\bar{x} \cdot y \cdot \bar{z})} \cdot \overline{(\bar{x} \cdot \bar{y} \cdot \bar{z})}}$$

4ª:

$$f(x,y,z) = \overline{\overline{(x + y + \bar{z}) \cdot (x + \bar{y} + \bar{z}) \cdot (\bar{x} + y + \bar{z})}} =$$
$$= \overline{\overline{(x + y + \bar{z})} + \overline{(x + \bar{y} + \bar{z})} + \overline{(\bar{x} + y + \bar{z})}}$$

Formas canónicas (cont.)

Exemplo: Determinar as formas canónicas da função f(x,y,z)

$$f(x, y, z) = x \cdot y + \overline{z}$$

	x	y	z	f(x,y,z)
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	0
6	1	1	0	1
7	1	1	1	1

1ª: $f(x, y, z) = \sum m(0,2,4,6,7)$

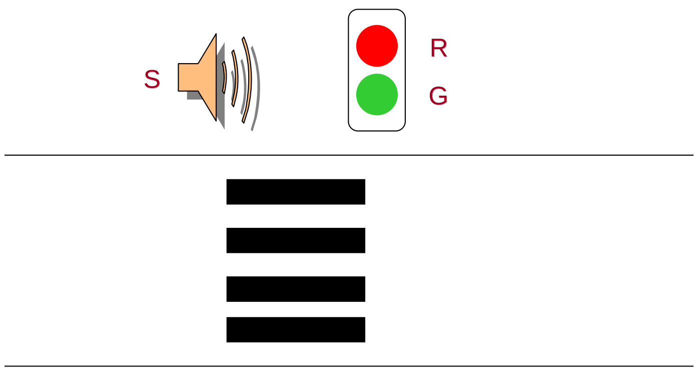
2ª: $f(x, y, z) = \prod M(1,3,5)$

3ª: $f(x, y, z) = \overline{\prod \overline{m(0,2,4,6,7)}}$

4ª: $f(x, y, z) = \overline{\sum \overline{M(1,3,5)}}$

Irrelevâncias

Condições irrelevantes – combinações das variáveis de entrada para as quais a saída não se conhece ou é irrelevante.



G (verde)	R (vermelho)	S (som)
0	0	0
0	1	0
1	0	1
1	1	x

x – *don't care*

O circuito real para todas as condições irrelevantes vai produzir nas saídas quaisquer valores válidos (0 ou 1).

O projetista tem a liberdade de atribuir valores 0 ou 1 às saídas respetivas.

Condições irrelevantes e formas canónicas

G (verde)	R (vermelho)	S (som)
0	0	0
0	1	0
1	0	1
1	1	x

1ª: $$S(G,R) = \sum m(2) + \sum d(3) = G \cdot \overline{R} + G \cdot R$$

2ª: $$S(G,R) = \prod M(0,1) \cdot \prod D(3) = (G + R) \cdot (G + \overline{R}) \cdot (\overline{G} + \overline{R})$$

3ª: $$S(G,R) = \overline{\prod \overline{m(2)} \cdot \overline{d(3)}} = \overline{\overline{G \cdot \overline{R}} \cdot \overline{G \cdot R}}$$

4ª: $$S(G,R) = \overline{\sum \overline{M(0,1)} + \overline{D(3)}} = \overline{\overline{(G + R)} + \overline{(G + \overline{R})} + \overline{(\overline{G} + \overline{R})}}$$

Análise de circuitos

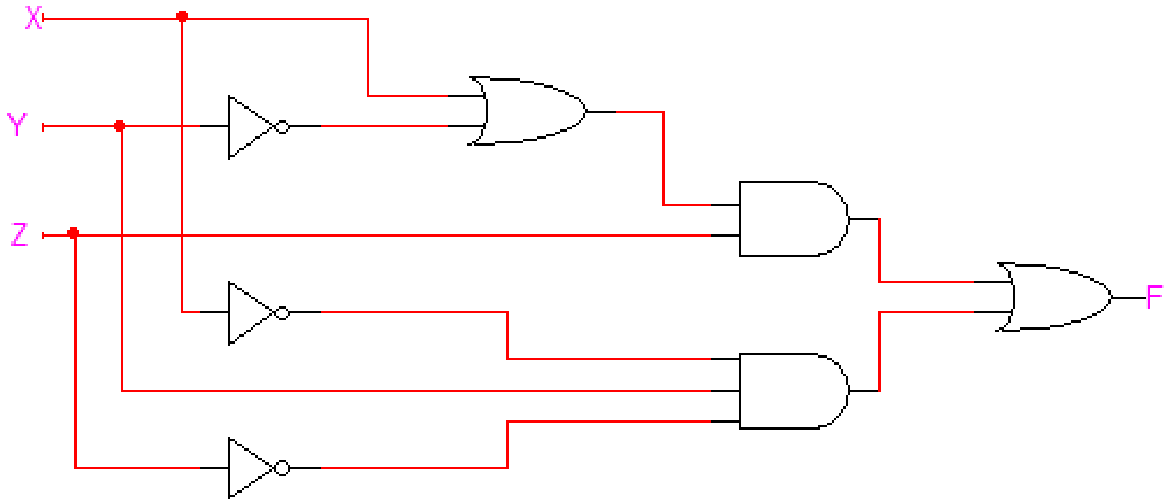
Quê função é realizada no circuito seguinte?

É possível reduzir o número de componentes?

Quantos níveis de atraso tem o circuito?

É possível reduzir o número de níveis de atraso?

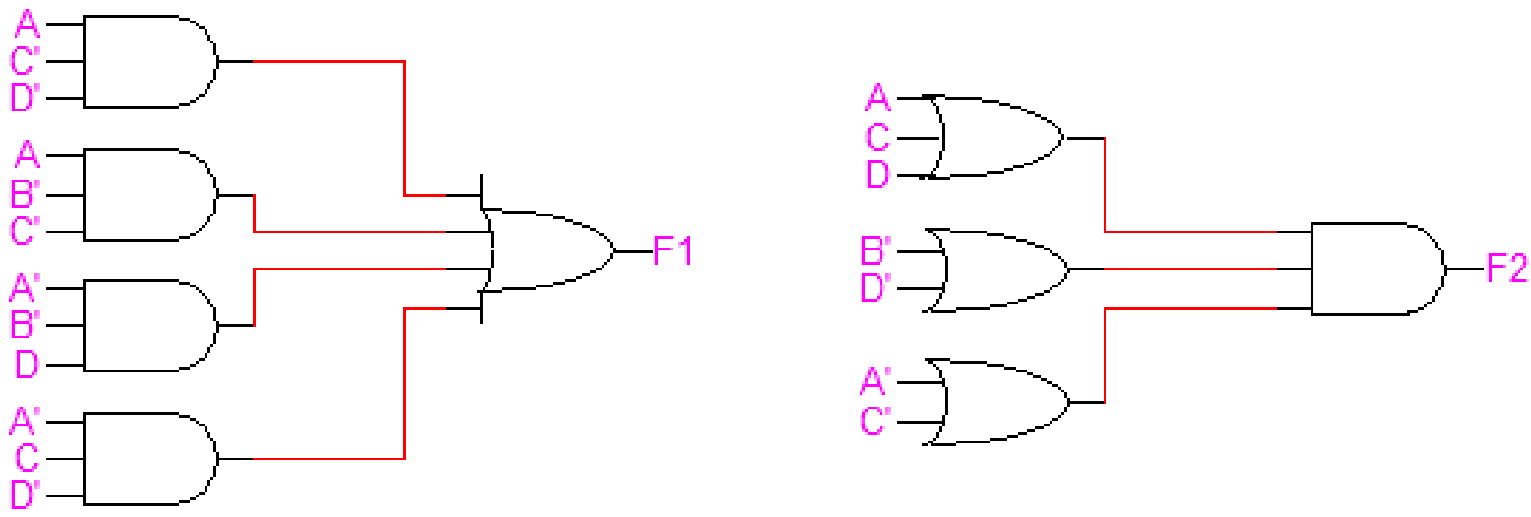
É possível implementar o circuito só com portas NAND?



Análise de circuitos

Compare os circuitos seguintes em termos de custos de implementação.

Será que os circuitos realizam a mesma função lógica?



Exercícios

Represente nas formas canónicas a função *f*: $f(x,y,z) = x \cdot y + \bar{x} \cdot \bar{z} + y \cdot z$

$$f(x,y,z) = \sum m(0,2,3,6,7) = \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot y \cdot \bar{z} + x \cdot y \cdot z$$

$$f(x,y,z) = \prod M(1,4,5) = (x + y + \bar{z}) \cdot (\bar{x} + y + z) \cdot (\bar{x} + y + \bar{z})$$

$$f(x,y,z) = \overline{\prod \overline{m(0,2,3,6,7)}} = \overline{\overline{(\bar{x} \cdot \bar{y} \cdot \bar{z})} \cdot \overline{(\bar{x} \cdot y \cdot \bar{z})} \cdot \overline{(\bar{x} \cdot y \cdot z)} \cdot \overline{(x \cdot y \cdot \bar{z})} \cdot \overline{(x \cdot y \cdot z)}}$$

$$f(x,y,z) = \overline{\sum \overline{M(1,4,5)}} = \overline{\overline{(x + y + \bar{z})} + \overline{(\bar{x} + y + z)} + \overline{(\bar{x} + y + \bar{z})}}$$

Minimize esta função aplicando métodos de minimização algébrica que conhece.

Exercícios (cont.)

Minimize funções seguintes aplicando métodos de minimização algébrica:

$$f(a,b,c) = \overline{a} \cdot b + \overline{a} \cdot \overline{c} + a \cdot c + a \cdot \overline{b} + b + c$$

$$f(a,b,c) = \overline{a} \cdot \overline{b} \cdot \overline{c} + \overline{a} \cdot b \cdot c + a \cdot b \cdot \overline{c} + a \cdot \overline{b} \cdot c$$

Introdução aos Sistemas Digitais

Minimização de funções Booleanas
Método de Karnaugh

Minimização de funções

$$f(x,y,z)=\overline{x}\cdot y\cdot\overline{z}+\overline{x}\cdot y\cdot z+x\cdot z+\overline{y}\cdot z+y\cdot\overline{z}$$

Aplicando o teorema de adjacência a dois primeiros termos:

$$f(x,y,z)=\overline{x}\cdot y+x\cdot z+\overline{y}\cdot z+y\cdot\overline{z}$$

4 termos, 8 literais

$$f(x,y,z)=\overline{x}\cdot y\cdot\overline{z}+y\cdot\overline{z}+z\cdot(\overline{x}\cdot y+x+\overline{y})=$$

$$=y\cdot\overline{z}+z\cdot(y+x+\overline{y})=$$

(absorção, simplificação)

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

(complementação, elemento absorvente)

$$=y+z$$

2 literais

(simplificação)

- Expressões irredutíveis podem não ser mínimas
- Pode existir mais que uma expressão mínima
- O processo de simplificações está sujeito a erros

Critérios de minimização

Para circuitos a dois níveis pode-se estabelecer seguintes critérios de minimização:

- Minimizar o número de termos (número de portas do 1º nível do circuito e número de entradas no 2º nível do circuito).
- Minimizar o número de literais (número de entradas nas portas do 1º nível do circuito).
- Os métodos de minimização não consideram o custo de inversão das variáveis de entrada.

Exemplos:

$f(a,b,c,d) = \overline{a} \cdot \overline{c} \cdot \overline{d} + b \cdot d + a \cdot c$ 3 termos, 7 literais

$f(a,b,c,d) = (\overline{a} + c + d) \cdot (b + c + \overline{d}) \cdot (a + b + \overline{c}) \cdot (a + \overline{c} + d)$ 4 termos, 12 literais

$g(a,b,c) = \overline{a} \cdot b + \overline{a} \cdot \overline{c}$ 2 termos, 4 literais

$g(a,b,c) = \overline{a} \cdot (b + \overline{c})$ 2 termos, 3 literais

Minimização de circuitos AND-OR

A soma de produtos mínima da função *f* é a soma de produtos que tem o número mínimo de produtos e o número mínimo de literais (comparando com todas as outras somas de produtos que possam existir com o mesmo número de produtos).

Exemplo:

$$f(a,b,c,d) = \overline{a}\cdot b\cdot \overline{c} + \overline{b}\cdot \overline{c}\cdot \overline{d} + a\cdot \overline{c}\cdot d + \overline{a}\cdot \overline{c}\cdot \overline{d}$$ 4 termos, 12 literais

$$f(a,b,c,d) = \overline{a}\cdot b\cdot \overline{c} + \overline{b}\cdot \overline{c}\cdot \overline{d} + a\cdot \overline{c}\cdot d$$ 3 termos, 9 literais

$$f(a,b,c,d) = \overline{a}\cdot \overline{c}\cdot \overline{d} + b\cdot \overline{c}\cdot d + a\cdot \overline{b}\cdot \overline{c}$$ 3 termos, 9 literais

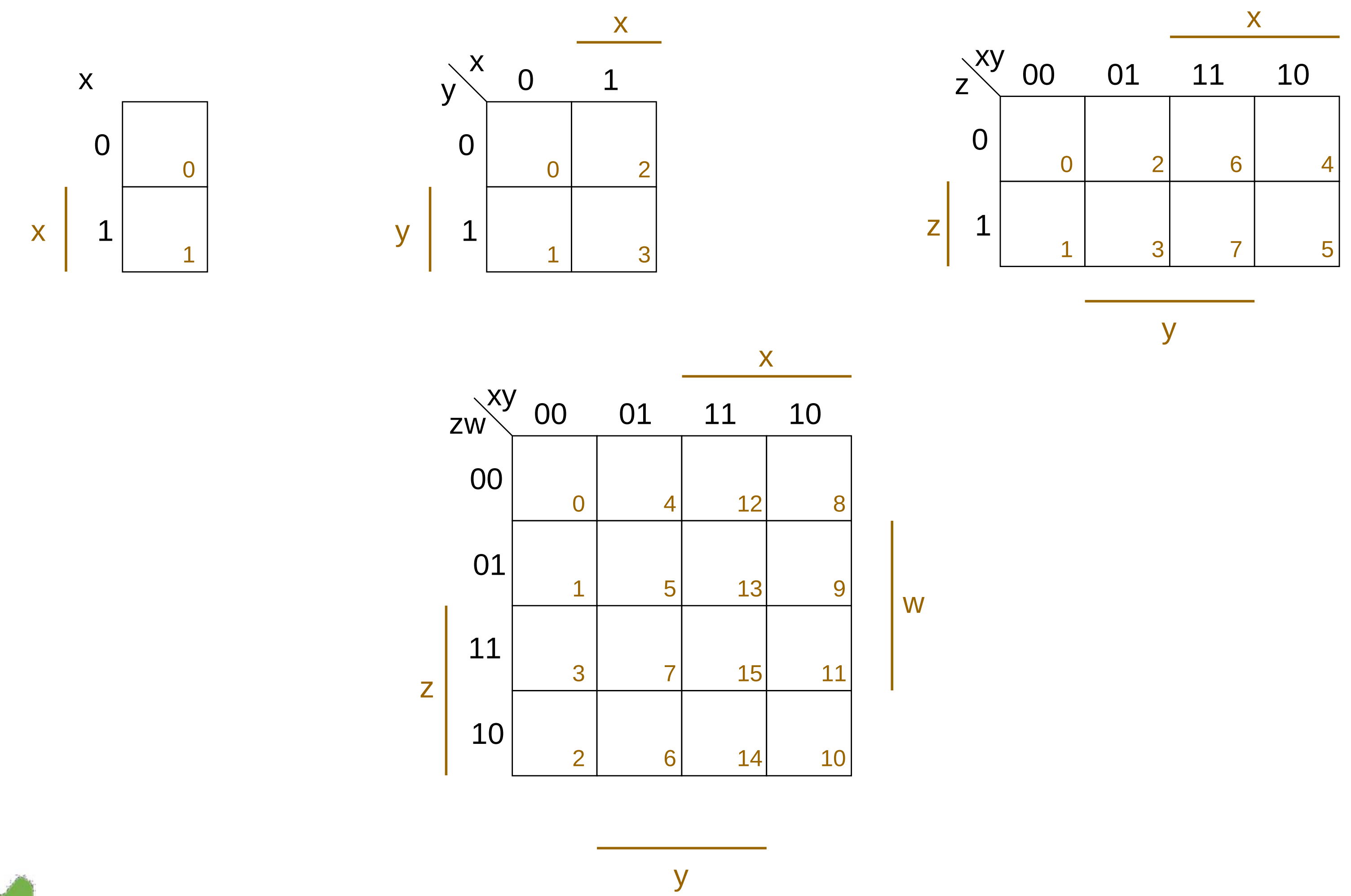
Todos os métodos de minimização são baseados na aplicação do teorema de adjacência:

$$\forall x,y \in B \quad x\cdot y + x\cdot \overline{y} = x$$

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

Mapas de Karnaugh

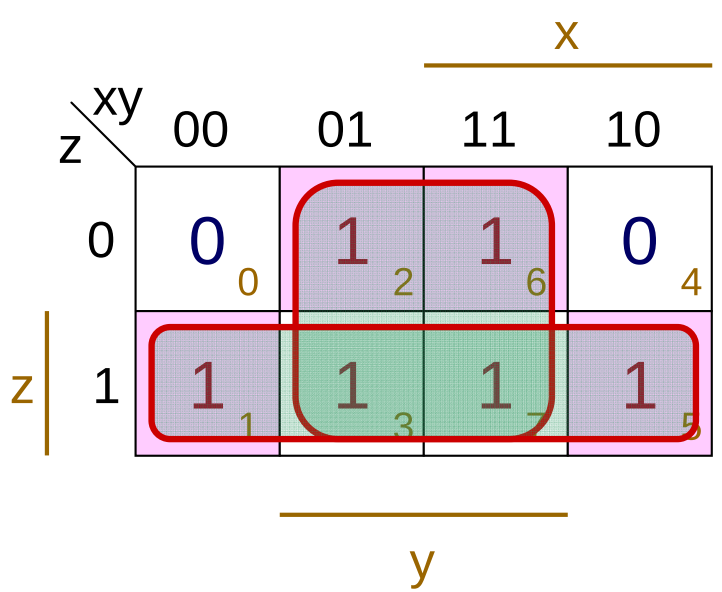
Um mapa de Karnaugh é a representação gráfica da tabela de verdade de uma função lógica.



Método de Karnaugh (3 variáveis)

$$f(x,y,z) = \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot z + \bar{y} \cdot z + y \cdot \bar{z}$$

	x	y	z	f(x,y,z)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1



$$f(x,y,z) = y + z$$

2 implicantes primos
4 células distintas
2 implicantes primos essenciais

Método de Karnaugh (4 variáveis)

	x	y	z	w	f(x,y,z,w)
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

zw \ xy	00	01	11	10
00	0_{0}	0_{4}	1_{12}	0_{8}
01	1_{1}	0_{5}	1_{13}	1_{9}
11	1_{3}	0_{7}	0_{15}	1_{11}
10	0_{2}	0_{6}	1_{14}	0_{10}

x

y

z

w

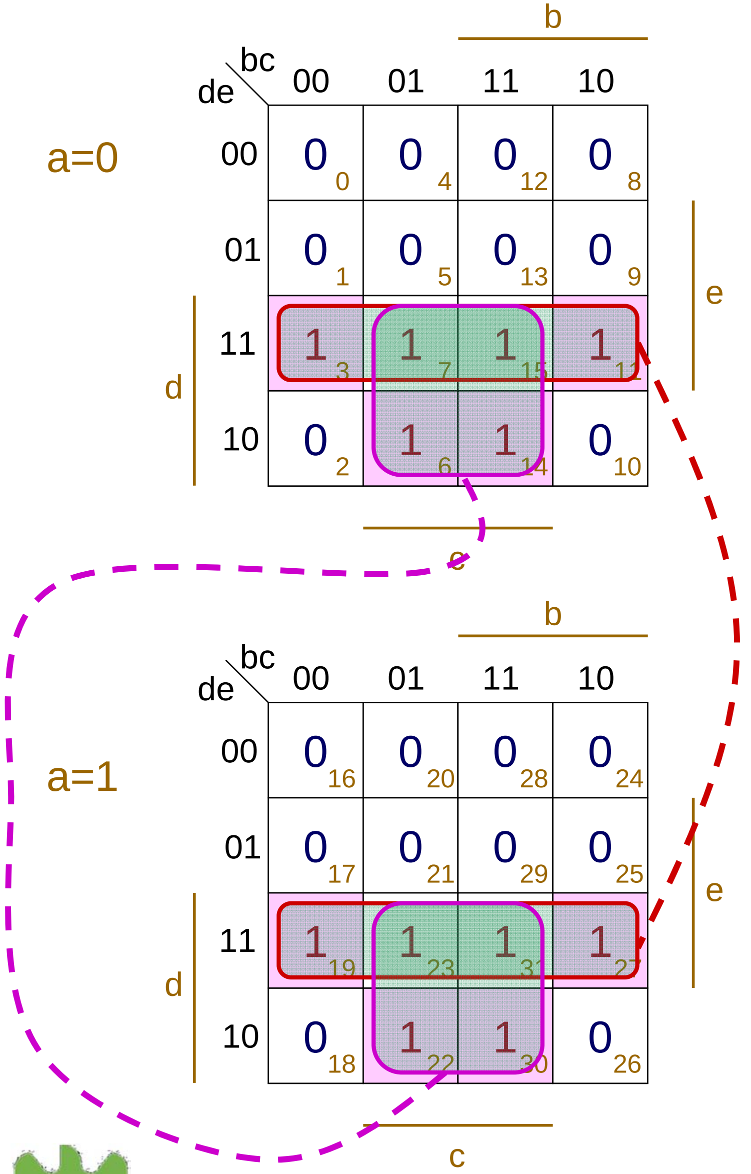
4 implicantes primos

4 células distintas

2 implicantes primos essenciais

$$f(x,y,z,w) = \overline{y} \cdot w + x \cdot y \cdot \overline{w} + x \cdot y \cdot \overline{z}$$

Método de Karnaugh (5 variáveis)



$$f(a,b,c,d,e) = \bar{a}\cdot\bar{b}\cdot d\cdot e + \bar{a}\cdot c\cdot d + \\ + \bar{a}\cdot b\cdot d\cdot e + a\cdot d\cdot e + a\cdot d\cdot c\cdot \bar{e}$$

2 implicantes primos

8 células distintas

2 implicantes primos essenciais

$$f(a,b,c,d,e) = d\cdot e + c\cdot d$$

Método de Karnaugh

1. Preencher o mapa de Karnaugh.
2. Encontrar todos os implicantes primos.

   Uma função *p* implica a função lógica *f* se em todos os casos quando *p* = ‘1’, *f* também é igual a ‘1’.

   O implicante primo da função *f* é um produto *p* que implica *f*, tal que se removêssemos um literal do *p* o produto resultante já não vai implicar *f*.

   Num mapa de Karnaugh, o implicante primo é um conjunto de 2^n células adjacentes que contêm valores ‘1’ e tais que se tentássemos expandir o conjunto até 2^{n+1} células este passará a incluir células com ‘0’.

3. Marcar no mapa todas as células distintas – células que só são cobertas por um único implicante primo.
4. Encontrar todos os implicantes primos essenciais – implicantes primos que cobrem uma ou mais células distintas.
5. Incluir na soma mínima todos os implicantes primos essenciais e dos restantes implicantes primos escolher o menor número daqueles que contêm menos literais.

Minimização de circuitos OR-AND

O produto de somas mínimo da função *f* é o produto de somas que tem o número mínimo de somas e o número mínimo de literais (comparando com todos os outros produtos de somas que possam existir com o mesmo número de somas).

A minimização pode ser feita aplicando o princípio da dualidade e analisando células com '0' no mapa de Karnaugh.

$$f(x,y,z) = \overline{x}\cdot y\cdot \overline{z} + \overline{x}\cdot y\cdot z + x\cdot z + \overline{y}\cdot z + y\cdot \overline{z}$$

	x	y	z	f(x,y,z)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

z \ xy	00	01	11 (x)	10 (x)
0	0_{0}	1_{2}	1_{6}	0_{4}
1 (z)	1_{1}	1_{3}	1_{7}	1_{5}

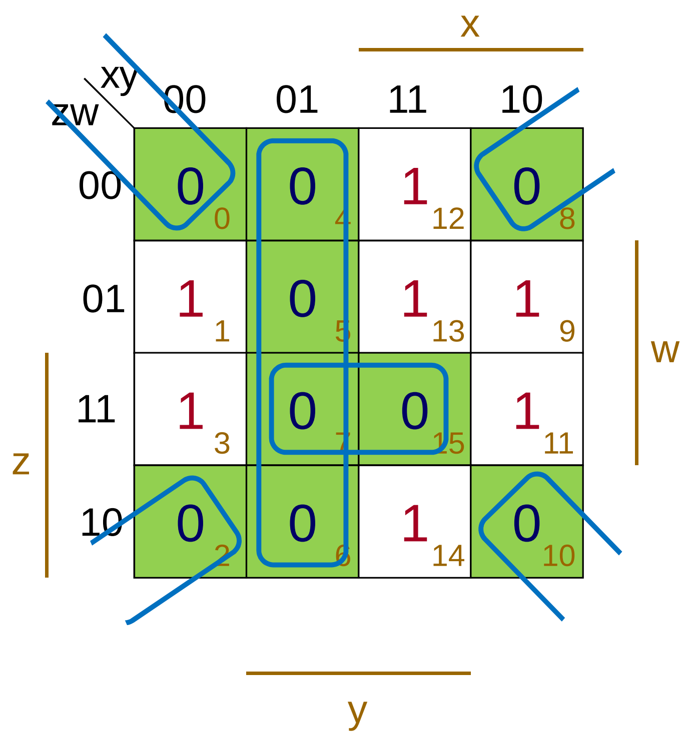
x

y

$$f(x,y,z) = y + z$$

Circuitos OR-AND (4 variáveis)

	x	y	z	w	f(x,y,z,w)
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0



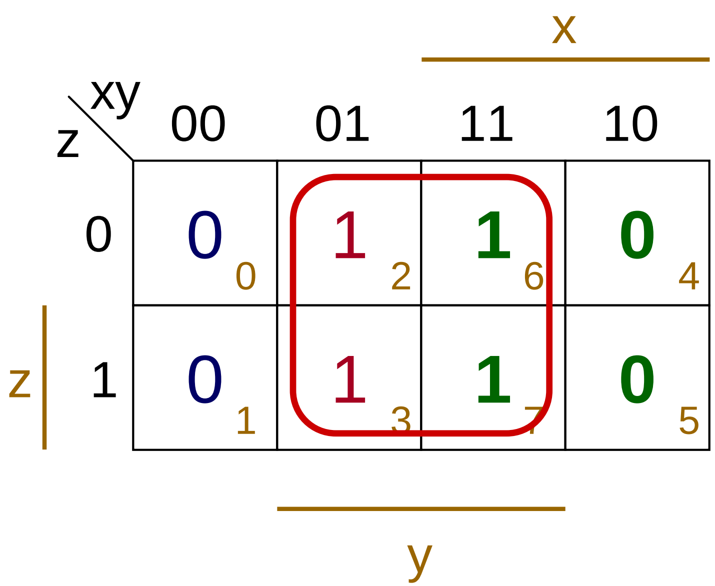
3 implicados primos
8 células distintas
3 implicados primos essenciais

$$f(x,y,z,w) = (y+w)\cdot(x+\overline{y})\cdot(\overline{y}+\overline{z}+\overline{w})$$

Condições irrelevantes

No mapa de Karnaugh as combinações irrelevantes deverão assumir valores que permitem reduzir o número de literais em cada um dos implicantes (implicados) primos (i.e. permitem aumentar as dimensões de cada conjunto de 2^n células).

	x	y	z	f(x,y,z)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	x
5	1	0	1	x
6	1	1	0	x
7	1	1	1	x



$$f(x, y, z) = y$$

Exercícios

Obtenha a forma mínima da função *f* na soma de produtos e produto de somas com o método de Karnaugh.

$$f(x,y,z) = x \cdot y + \bar{x} \cdot \bar{z} + y \cdot z$$

$$f(x,y,z) = y + \bar{x} \cdot \bar{z} = (\bar{x} + y) \cdot (y + \bar{z})$$

Determine a forma mínima das funções representadas nos seguintes mapas de Karnaugh (na soma de produtos e produto de somas):

cd \ ab	00	01	11	10
00	1 $_{0}$	0 $_{4}$	0 $_{12}$	0 $_{8}$
01	0 $_{1}$	1 $_{5}$	1 $_{13}$	0 $_{9}$
11	0 $_{3}$	1 $_{7}$	1 $_{15}$	0 $_{11}$
10	1 $_{2}$	1 $_{6}$	1 $_{14}$	1 $_{10}$

a (ab = 11, 10); b (ab = 01, 11); c (cd = 11, 10); d (cd = 01, 11)

cd \ ab	00	01	11	10
00	0 $_{0}$	0 $_{4}$	0 $_{12}$	0 $_{8}$
01	x $_{1}$	1 $_{5}$	1 $_{13}$	x $_{9}$
11	0 $_{3}$	x $_{7}$	x $_{15}$	0 $_{11}$
10	1 $_{2}$	1 $_{6}$	1 $_{14}$	1 $_{10}$

a (ab = 11, 10); b (ab = 01, 11); c (cd = 11, 10); d (cd = 01, 11)

Exercícios (cont.)

Encontre exemplos de funções de 4 variáveis que obedeçam aos seguintes critérios:

- têm duas formas mínimas;

- a soma de produtos mínima e o produto de somas mínimo têm o mesmo número de termos e literais;

- a soma de produtos mínima tem menos termos e literais que o produto de somas mínimo;

- a soma de produtos mínima tem mais termos e literais que o produto de somas mínimo.

Exercícios (cont.)

Preencha diretamente o mapa de Karnaugh da seguinte função e obtenha a soma de produtos mínima:

$$f(x,y,z) = (x \oplus y)\cdot z + \overline{x}\cdot(y \oplus z)$$

Encontre um exemplo de função de 3 variáveis para a qual a soma de produtos mínima tem <u>menos termos e literais</u> que o produto de somas mínimo.

Quantos implicantes primos e implicantes primos essenciais tem a função seguinte?

$$f(a,b,c,d) = \sum m(2,6,7,8,9,10,13,15)$$

Introdução aos Sistemas Digitais

Síntese de circuitos combinatórios
Manipulação de circuitos

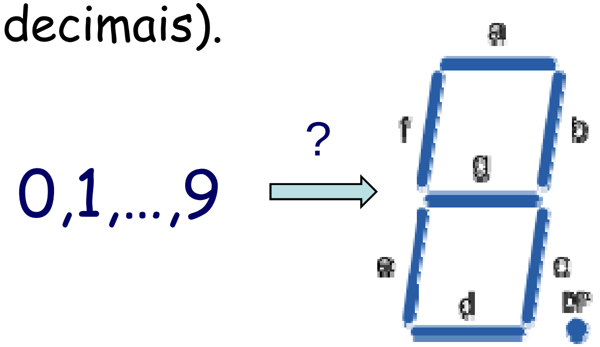
Síntese de circuitos

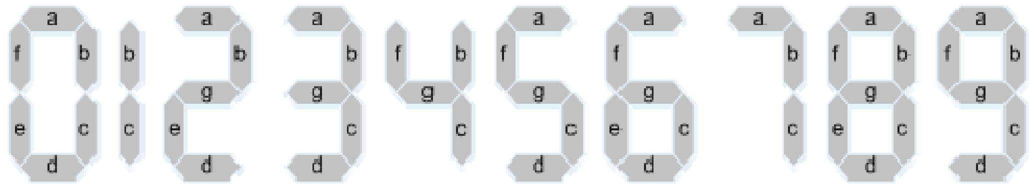
Partir de uma descrição da função pretendida e sintetizar o circuito que implementa esta função.

1. Identificar entradas e saídas.
2. Construir a tabela de verdade.
3. Obter expressão mínima para a(s) saída(s).
4. Desenhar o circuito.

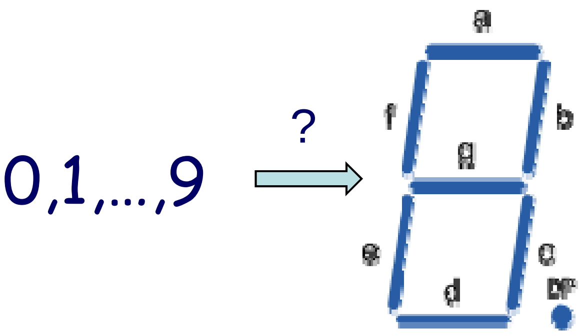
Exemplo:

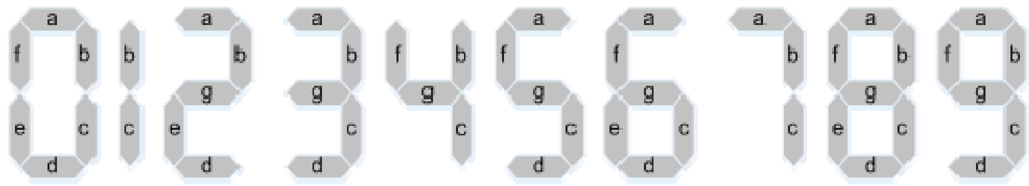
Construa um controlador de display de 7 segmentos (para representar dígitos decimais).

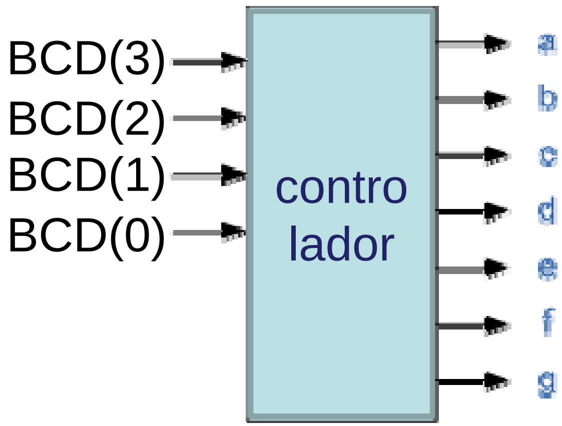




Controlador de *display* de 7 segmentos

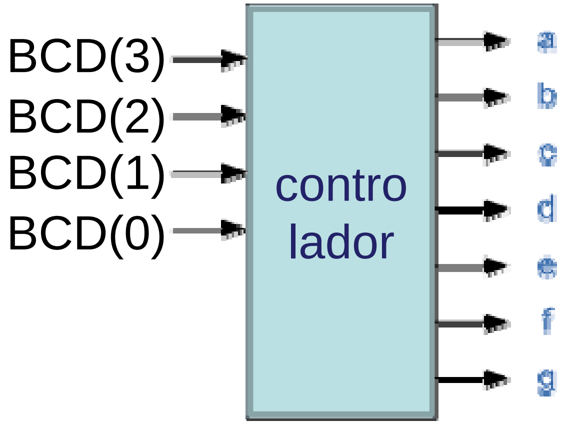






BCD	número	segmentos individuais						
		a	b	c	d	e	f	g
0000	0	1	1	1	1	1	1	
0001	1		1	1				
0010	2	1	1		1	1		1
0011	3	1	1	1	1			1
0100	4		1	1			1	1
0101	5	1		1	1		1	1
0110	6	1		1	1	1	1	1
0111	7	1	1	1				
1000	8	1	1	1	1	1	1	1
1001	9	1	1	1	1		1	1
101x	x	x	x	x	x	x	x	x
11xx	x	x	x	x	x	x	x	x

Controlador de *display* de 7 segmentos



Obter expressões mínimas para as saídas *a*..*g* (por exemplo com o método de Karnaugh).

Realizar as funções obtidas

a(BCD(3), BCD(2), BCD(1), BCD(0))

..

g(BCD(3), BCD(2), BCD(1), BCD(0))

com portas lógicas elementares (ou outros blocos mais complexos – matéria das próximas aulas).

Nomes de sinais

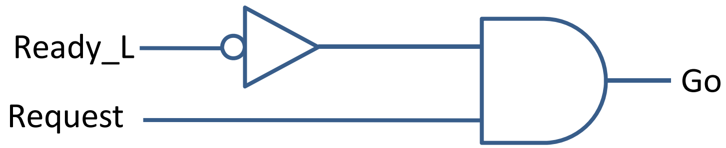
Um sinal está ativo a *High* se este força uma ação ou denota uma condição quando está a *1*.

Um sinal está ativo a *Low* se este força uma ação ou denota uma condição quando está a *0*.

Para clarificar a interpretação de circuitos deve-se indicar explicitamente que sinais são ativos *Low* (`nome_de_sinal_L`) e que sinais são ativos *High* (`nome_de_sinal`).

Exemplo:

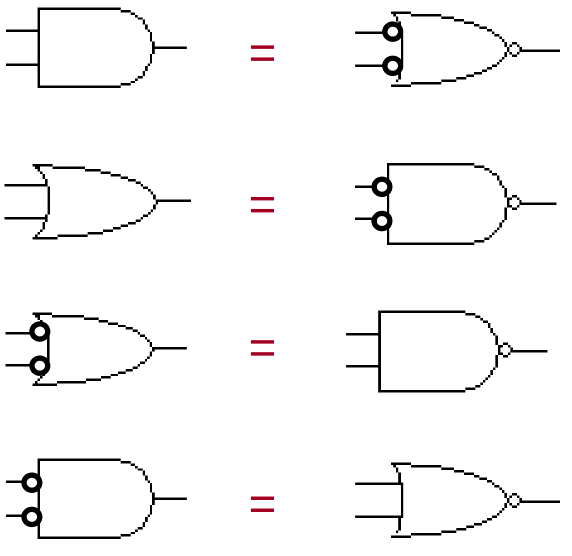
Realizar ação `Go` se os sinais `Ready` (ativo a Low) e `Request` (ativo a High) forem ativados.



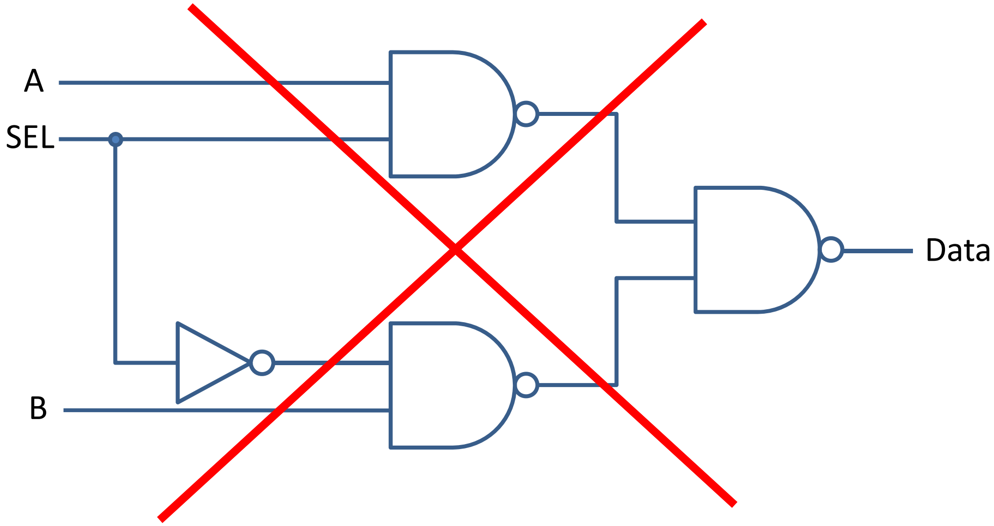
Projeto *bubble-to-bubble*

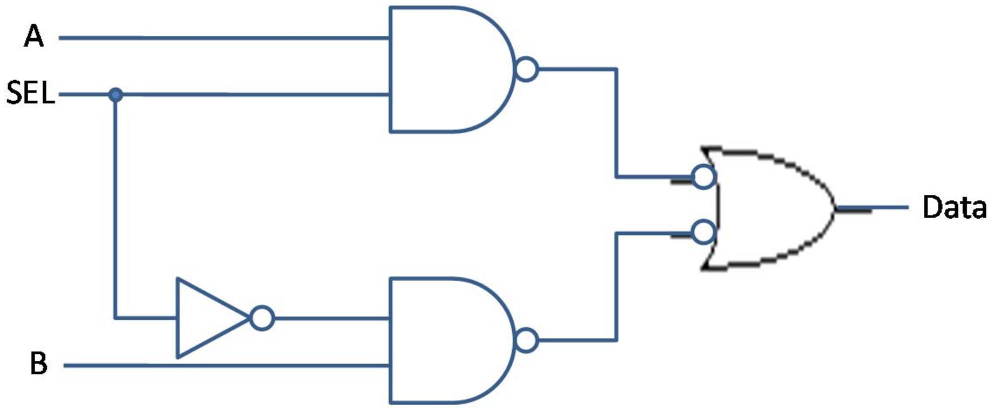
Convém, sempre que possível, praticar *bubble-to-bubble design* que permite simplificar o entendimento da função dum circuito desenhando-o de modo que as inversões existentes cancelam umas outras.

De acordo com a lei de DeMorgan:



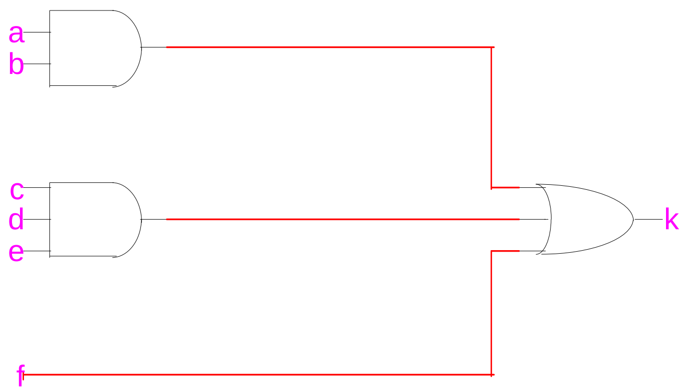
Projeto *bubble-to-bubble* (cont.)

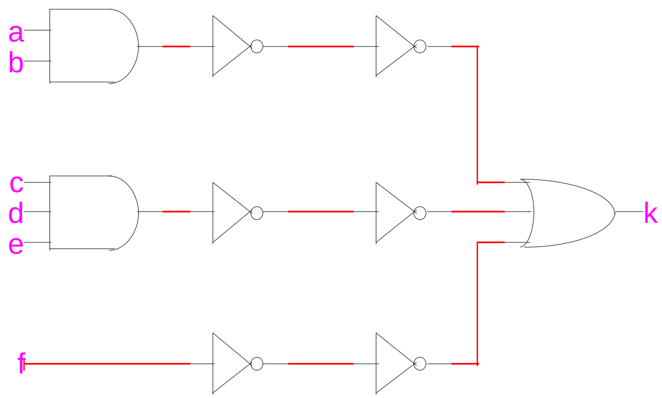


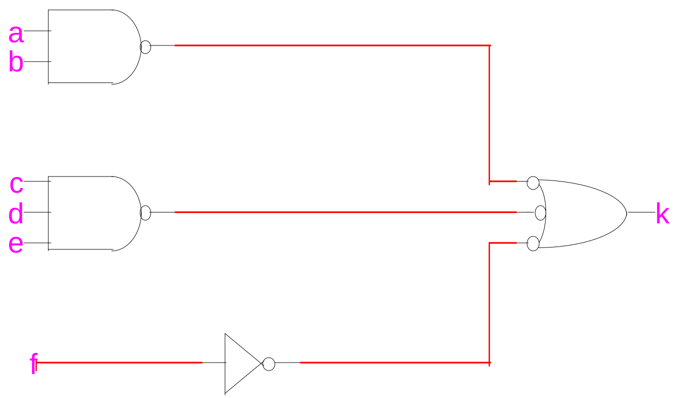


Conversão de formas (AND-OR -> NAND-NAND)

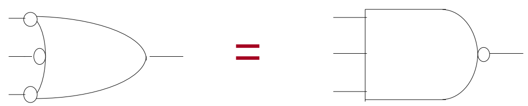
Um circuito AND-OR a 2 níveis pode ser convertido num circuito NAND-NAND a 2 níveis invertendo as saídas das portas do 1º nível e as entradas das portas do 2º nível.





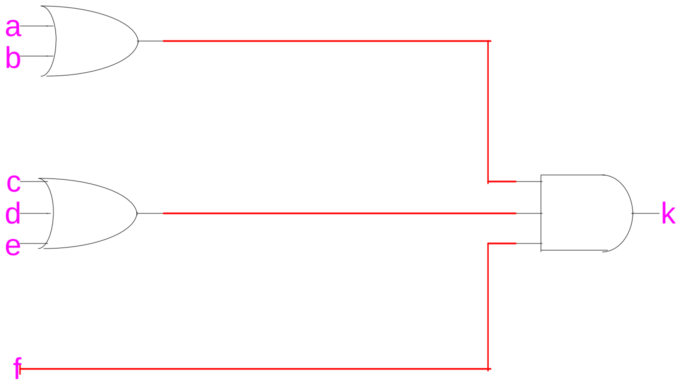


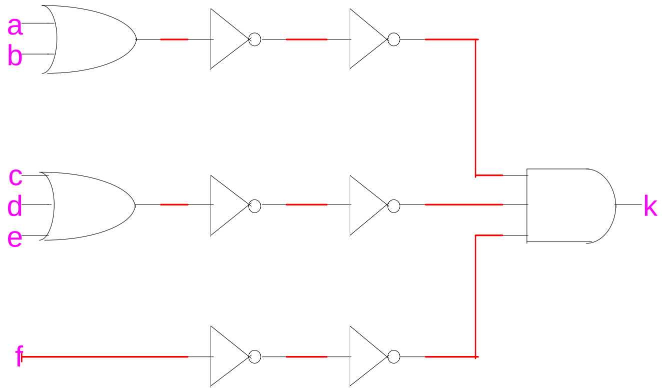
De acordo com a lei de DeMorgan:

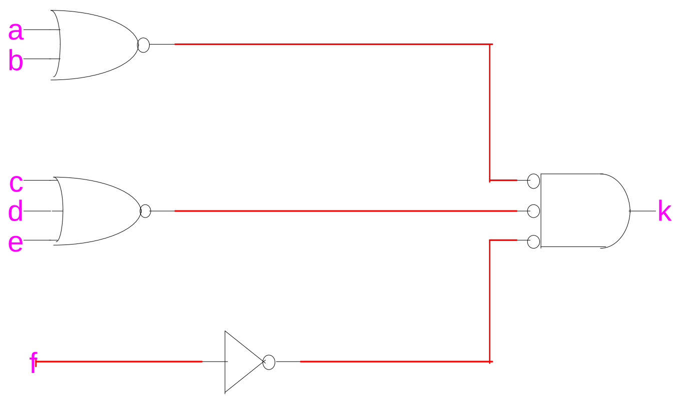


Conversão de formas (OR-AND -> NOR-NOR)

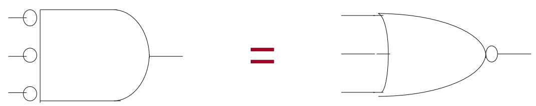
Um circuito OR-AND a 2 níveis pode ser convertido num circuito NOR-NOR a 2 níveis invertendo as saídas das portas do 1º nível e as entradas das portas do 2º nível.





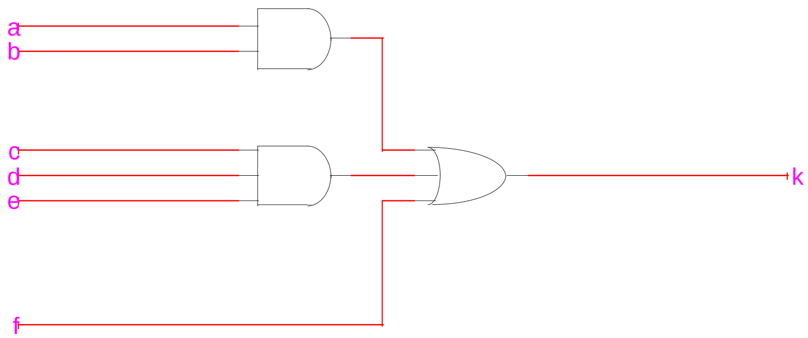


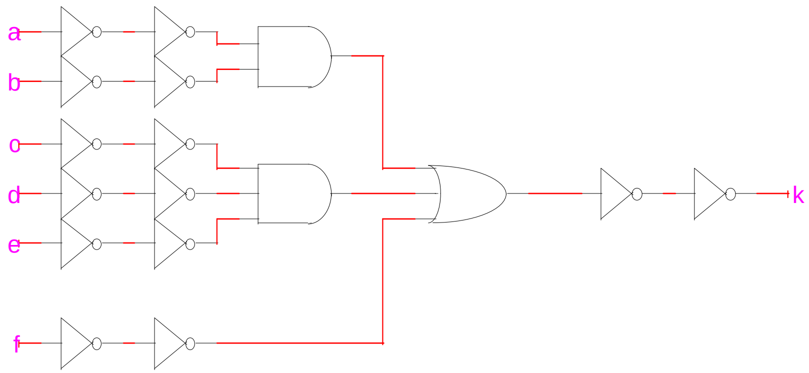
De acordo com a lei de DeMorgan:

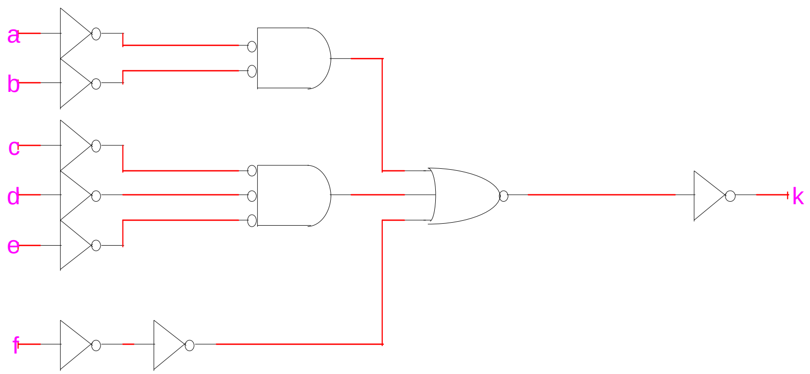


Conversão de formas (AND-OR -> NOR-NOR)

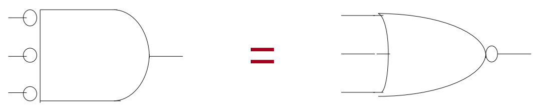
Um circuito AND-OR a 2 níveis pode ser convertido num circuito NOR-NOR a 2 níveis invertendo duas vezes as entradas das portas do 1º nível e as saídas das portas do 2º nível.





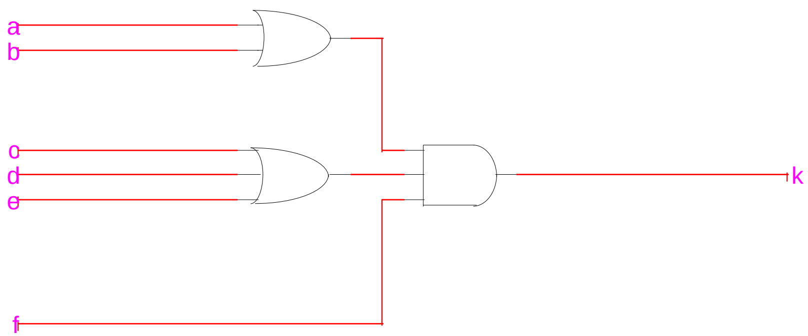


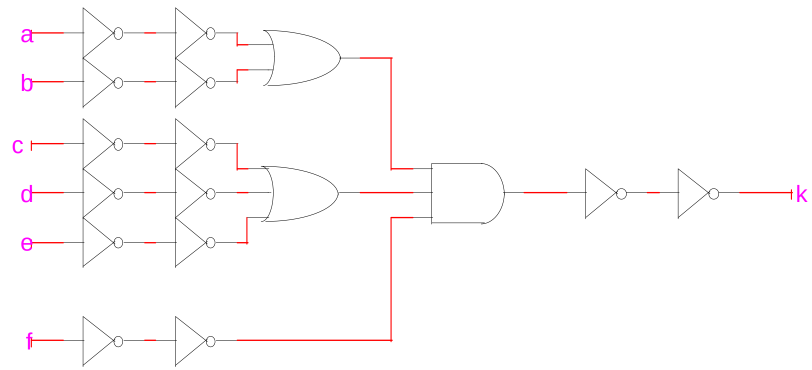
De acordo com a lei de DeMorgan:

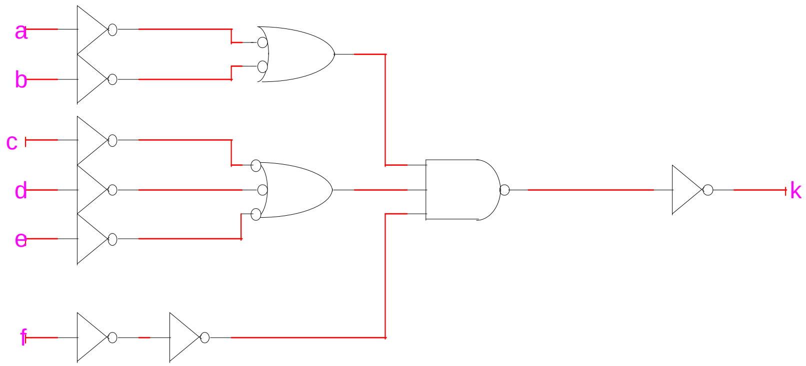


Conversão de formas (OR-AND -> NAND-NAND)

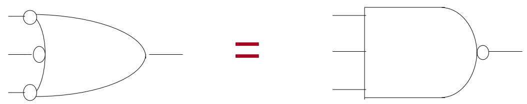
Um circuito OR-AND a 2 níveis pode ser convertido num circuito NAND-NAND a 2 níveis invertendo duas vezes as entradas das portas do 1º nível e as saídas das portas do 2º nível.







De acordo com a lei de DeMorgan:



Conversão de circuitos multi-nível

Um circuito AND-OR multi-nível pode ser convertido num circuito NAND-NAND multi-nível invertendo as saídas das portas dos níveis ímpares e as entradas das portas dos níveis pares.

Um circuito OR-AND multi-nível pode ser convertido num circuito NOR-NOR multi-nível invertendo as saídas das portas dos níveis ímpares e as entradas das portas dos níveis pares.

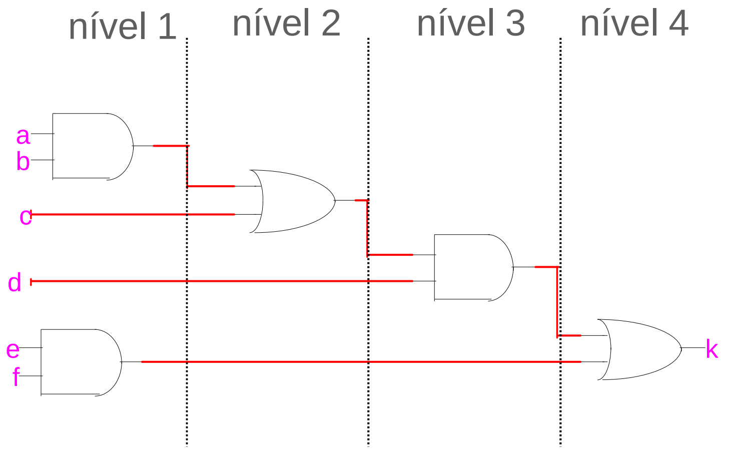
Um circuito AND-OR multi-nível pode ser convertido num circuito NOR-NOR multi-nível invertendo duas vezes as entradas das portas dos níveis ímpares e as saídas das portas dos níveis pares.

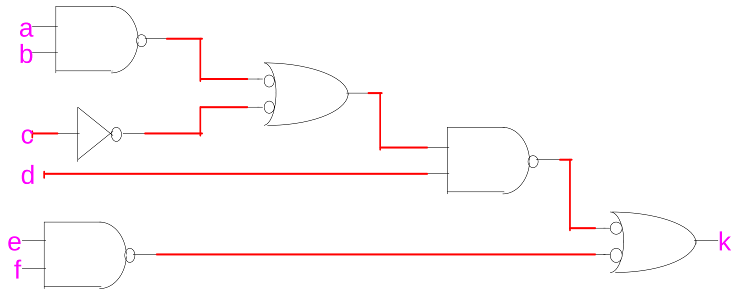
Um circuito OR-AND multi-nível pode ser convertido num circuito NAND-NAND multi-nível invertendo duas vezes as entradas das portas dos níveis ímpares e as saídas das portas dos níveis pares.

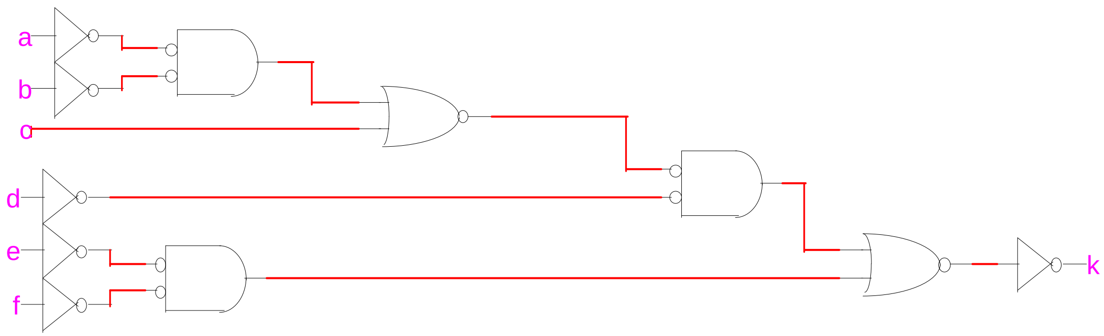
Conversão de circuitos multi-nível (cont.)

Exemplo:

Converta o circuito seguinte em circuitos NAND-NAND e NOR-NOR.







Exercícios

Represente o circuito nas formas NOR-NOR e NAND-NAND.