


Karnaugh  
Simplificações:  
Dont care  
Nao E  
Não OU



# Considerações

Quando uma variável aparece nas formas complementada e não-complementada em um agrupamento, tal variável é eliminada da expressão. As variáveis que não se alteram para todos os quadros do agrupamento têm de permanecer na expressão final.

## • Procedimento passo-a-passo

- Passo 1** Construa o mapa K e coloque os 1s nos quadros que correspondem aos 1s na tabela-verdade. Coloque 0s nos outros quadros.
- Passo 2** Analise o mapa quanto aos 1s adjacentes e agrupe os 1s que *não* sejam adjacentes a quaisquer outros 1s. Esses são denominados 1s *isolados*.
- Passo 3** Em seguida, procure os 1s que são adjacentes a somente um outro 1. Agrupe *todo* par que contém tal 1.
- Passo 4** Agrupe qualquer octeto, mesmo que ele contenha alguns 1s que já tenham sido agrupados.
- Passo 5** Agrupe qualquer quarteto que contenha um ou mais 1s que ainda não tenham sido agrupados, *certificando-se de usar o menor número de agrupamentos*.
- Passo 6** Agrupe quaisquer pares necessários para incluir quaisquer 1s que ainda não tenham sido agrupados, *certificando-se de usar o menor número de agrupamentos*.
- Passo 7** Forme a soma OR de todos os termos gerados por cada grupo.



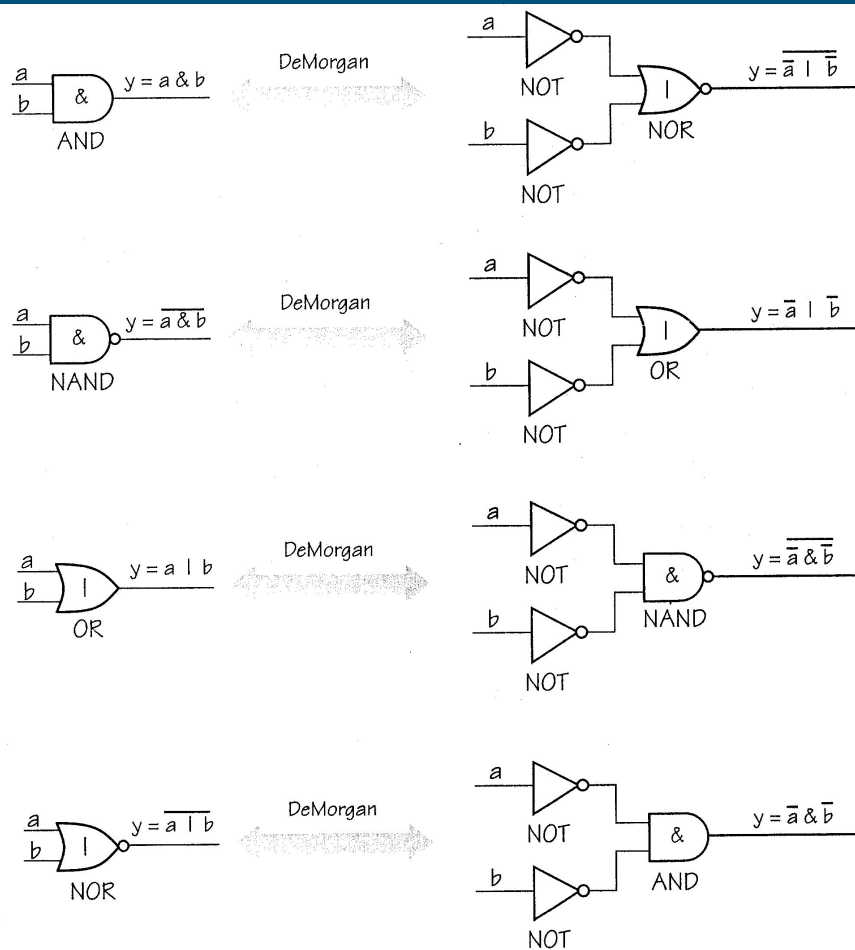
# Método Nao-E e Nao-OU



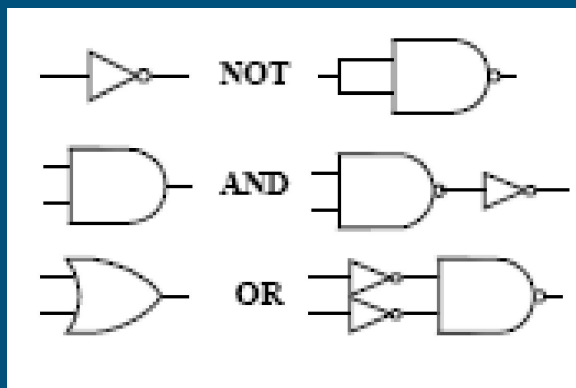
Relembrar Morgan



# Simplificações de Funções Booleanas – Exemplo Morgan



# Simplificações de Funções Booleanas – Simplificações com Portas NAND

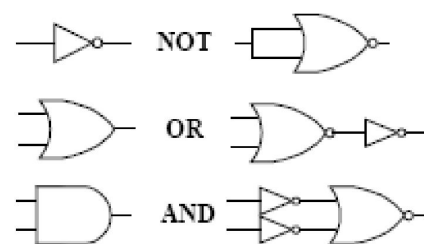


## Simplificações com Portas NOR

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.



# Exemplo:

---

Considere uma votação de 4 juízes (A, B, C e D). O juiz A tem direito a voto de qualidade valendo 3 votos simples enquanto os restantes apenas têm direito a um voto simples cada. Determine a tabela de verdade das funções que representam uma decisão a favor por unanimidade (F0), uma decisão a favor por maioria ( $> 50\%$ ) (F1) e uma decisão contra (F2). Utilizando Mapas de Karnaugh, simplifique as funções de saída. Implemente as funções de saída usando unicamente portas lógicas NOR.

# DONT CARE



## Condições irrelevantes ou don't care

- Em alguns projetos, a condição de saída pode ser irrelevante, porque certas condições de entrada nunca ocorrerão
- Essa condição de saída pode assumir o estado *ALTO* ou *BAIXO*, de acordo com a escolha do projetista, e é sinalizada na tabela verdade por um **x**
- Pode-se então escolher a saída como 0 ou 1, de forma a simplificar o circuito o máximo possível



# Exemplo de don't care

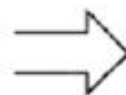
A	B	C	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	x
1	0	0	x
1	0	1	1
1	1	0	1
1	1	1	1

(a)

} don't  
care

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	0	x
$AB$	1	1
$A\bar{B}$	x	1

(b)



	$\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	0	0
$AB$	1	1
$A\bar{B}$	1	1

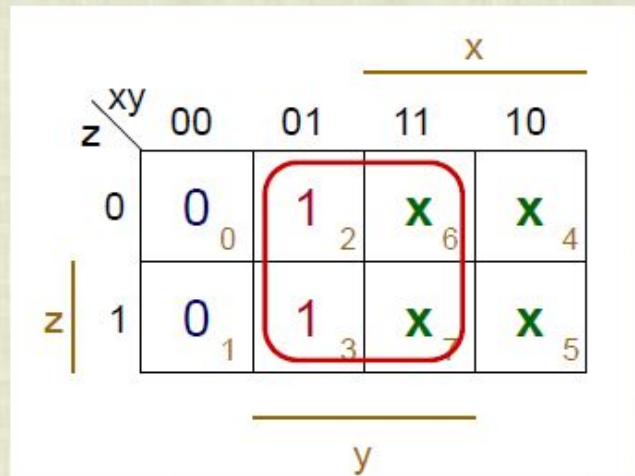
(c)

z = A

## Condições irrelevantes (Don't care)

No mapa de Karnaugh as combinações irrelevantes deverão assumir valores que permitem reduzir o número de literais em cada um dos implicantes 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



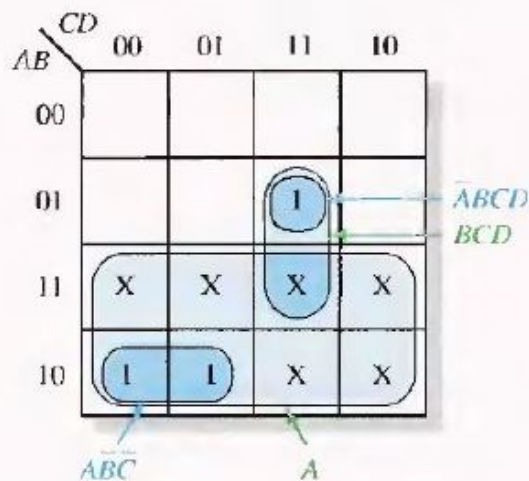
$$f(x,y,z) = y$$

## Condições irrelevantes (Don't care)

Inputs <i>ABCD</i>	Output <i>Y</i>
0 0 0 0	0
0 0 0 1	0
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	0
0 1 1 1	1
1 0 0 0	1
1 0 0 1	1
1 0 1 0	X
1 0 1 1	X
1 1 0 0	X
1 1 0 1	X
1 1 1 0	X
1 1 1 1	X

(a) Truth table

Don't cares



- (b) Without "don't cares"  $Y = \overline{A}\overline{B}\overline{C} + \overline{A}BCD$   
 With "don't cares"  $Y = A + BCD$

1) Determine, utilizando mapas de Karnaugh, a forma mínima do produto de somas das seguintes funções, atendendo às condições indiferente.

a)  $F(W,X,Y,Z) = \sum m(0,1,2,3,7,8,10)$  ;

$d(W,X,Y,Z) = \sum m(5,6,11,15)$

b)  $F(A,B,C,D) = \sum m(3,4,13,15)$  ;  $d(A,B,C,D) = \sum m(1,2,5,6,8,10,12,14)$

# Desafio:

- **Projetar um circuito simplificado que caracterize um elevador da seguinte forma:**
  - M sinaliza que o elevador está em movimento (1) ou parado (0)
  - O prédio possuir 3 andares (A1, A2 e A3)
    - O sistema deve reconhecer a presença do elevador no andar (1) ou não (0)
  - A saída P deve indicar que a porta pode ser aberta (1) sempre que elevador estiver parado em um dado andar.