Capítulo 2-Projeto Lógico Combinacional D

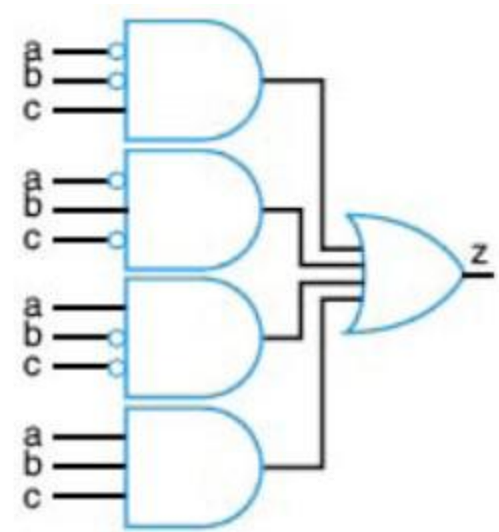
Profa. Eliete Caldeira

Projeto Lógico Combinacional

- Passo 1: Capturar a função
  - Tabela-verdade ou equacao que descreva o problema
- Passo 2: Converter para equações
  - Usar a forma canônica de soma de mintermos ou de produto de maxtermos
- Passo 3: Implementar um circuito
  - Criar um circuito correspondente a equação da saída em questão

Implementação lógica de dois níveis

- Estrutura em soma de produtos:
  - Uma coluna de portas lógicas AND
  - Saídas das portas alimentando uma porta OR
- Estrutura em produto de somas:
  - Uma coluna de portas lógicas OR
  - Saídas das portas alimentando uma porta AND



- Supondo que as variáveis estão disponíveis em forma normal e complementada rende dois níveis entre a entrada e a saída
- A forma de soma de produtos (Figura) é a mais usada

Exercício:

- Queremos implementar um circuito que possa detectar se um padrão de pelo menos três uns adjacentes ocorrerem em qualquer lugar em uma entrada de 8 bits e produzir uma saída 1 neste caso. As entradas são **a**, **b**, **c**, **d**, **e**, **f**, **g** e **h** e a saída é **y.**
- Assim:
- Se **abcdefgh**=00011101 y=1 porque **d**=1, **e**=1 e **f**=1
- Se **abcdefgh**=10101011 y=0 porque não tem três 1s em sequência
- Se **abcdefgh**=11110101 y=1 porque tem três 1s em sequência duas vezes **abc** e **bcd**

Exercício:

- Queremos implementar um circuito que possa detectar se um padrão de pelo menos três uns adjacentes ocorrerem em qualquer lugar em uma entrada de 8 bits e produzir uma saída 1 neste caso. As entradas são **a**, **b**, **c**, **d**, **e**, **f**, **g** e **h** e a saída é **y**.
- Passo1: **y**=**abc**+**bcd**+**cde**+**def**+**efg**+**fgh**
- Passo2: já temos uma equação

Exercício:

- Passo3:

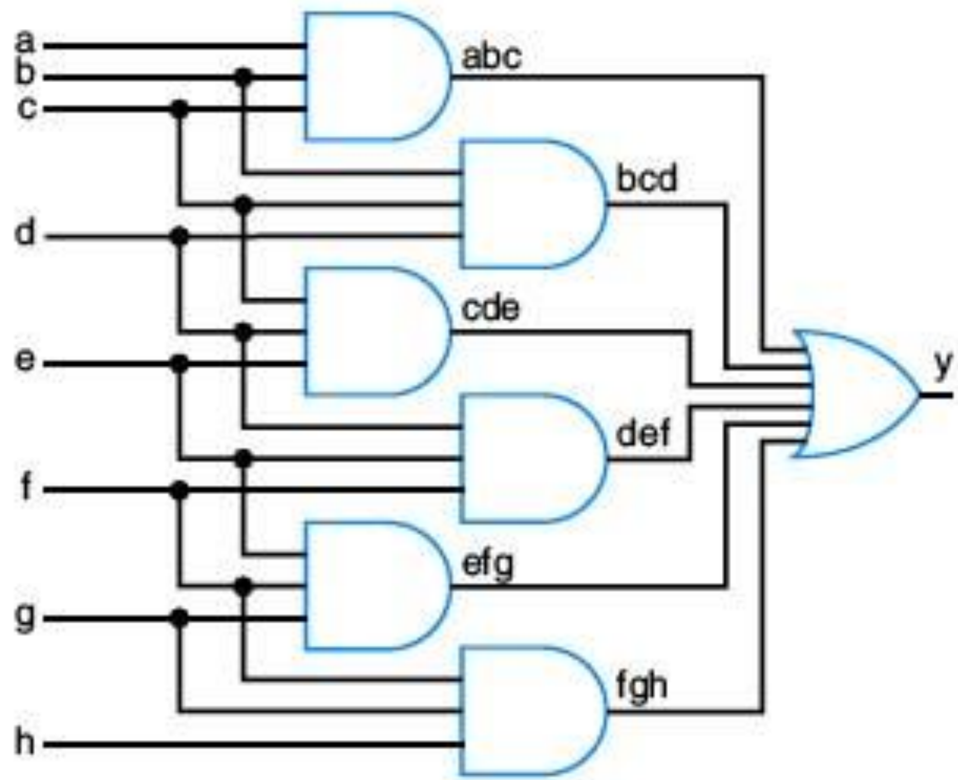


Figure 2.40 Three 1s pattern detector.

Exercício

- Nós queremos projetar um circuito que conta o número de 1s presente em uma entrada de 3 bits abc, gerando uma saída binária de dois bits y e z.

Exercício

- Nós queremos projetar um circuito que conta o número de 1s presente em uma entrada de 3 bits abc, gerando uma saída binária de dois bits y e z.
- Passo1:

a	b	c	y	z	Número de 1s
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	2
1	0	0	0	1	1
1	0	1	1	0	2
1	1	0	1	0	2
1	1	1	1	1	3

- Passo2: $y = a'bc+ab'c+abc'+abc$
  e $z=a'b'c+a'bc'+ab'c'+abc$

Exercício

- Passo3:

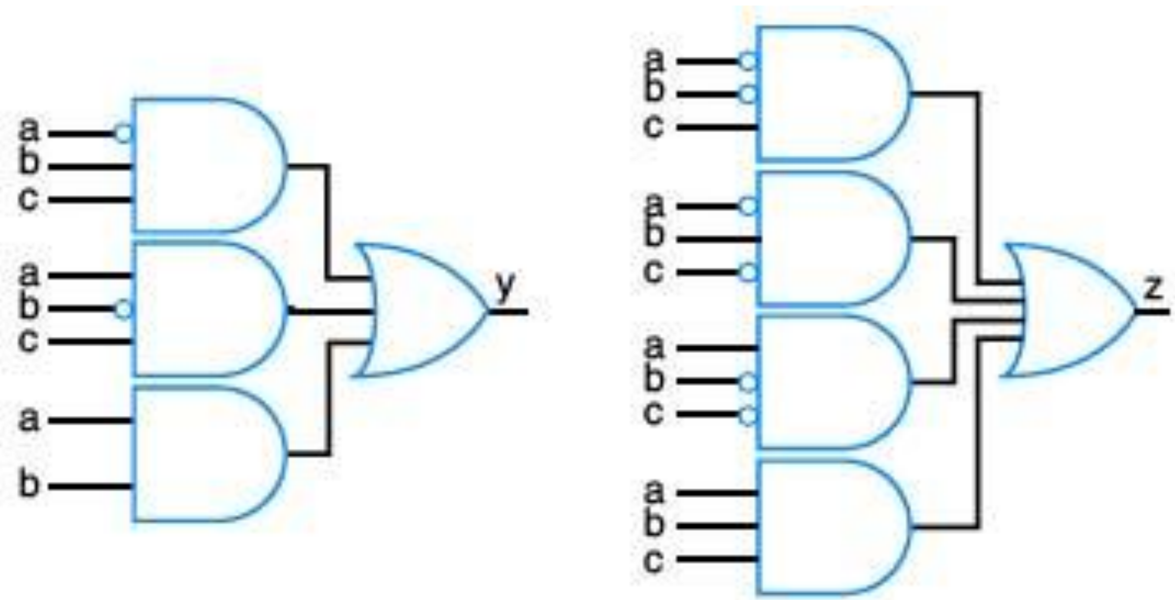


Figure 2.41 Number-of-1s counter gate-based circuit.

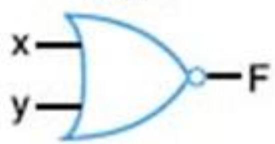
Exercício

- Projete um circuito com quatro entradas a, b, c e d e uma saída z que seja 1 quando o número na entrada for composto só de 0s.

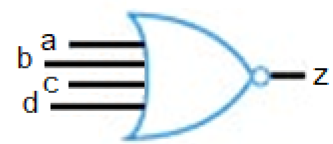
Exercício

- Projete um circuito com quatro entradas a, b, c e d e uma saída z que seja 1 quando o número na entrada for composto só de 0s.
- A função NOR produz uma saída 1 quando as entradas são 0, assim...

NOR



x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

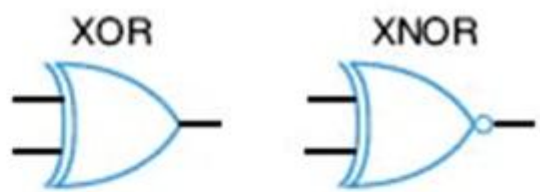


Exercício

- Projete um circuito que gera uma saída E = 1 quando A igual a B, onde A e B são números de 3 bits ($A_2A_1A_0$ e $B_2B_1B_0$).

Exercício

- Projete um circuito que gera uma saída E = 1 quando A igual a B, onde A e B são números de 3 bits ($A_2A_1A_0$ e $B_2B_1B_0$).
- As funções XOR e XNOR de duas variáveis

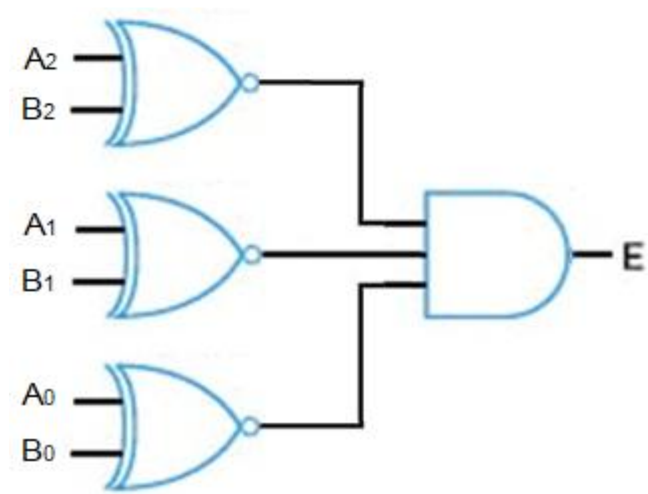


x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

Exercício

- Projete um circuito que gera uma saída E = 1 quando A igual a B, onde A e B são números de 3 bits ($A_2A_1A_0$ e $B_2B_1B_0$).



Exercícios

- Projete um circuito que gere a paridade par de um número de três bits

- A paridade ímpar pode ser gerada com a XNOR

Implementação em redes de dois níveis

- Todas as entradas estão disponíveis na forma normal e complementada
- Rede AND – OR
  - Corresponde a uma soma de produtos
  - Facilmente transformável em uma rede NAND – NAND
- Rede OR – AND
  - Corresponde a um produto de somas
  - Facilmente transformável em uma rede NOR – NOR

Redes de dois níveis mínimas

- Considerações:
  - Entradas disponíveis nas formas x e x’
  - Não há limitação quanto ao número de entradas
  - Possuem somente 1 saída
  - A métrica de minimização consiste em minimizar o numero de portas com o mínimo numero de entradas
  - Não estamos preocupados em manter a forma canônica

Redes de dois níveis mínimas

- Métodos de minimização:
  - Algébrico
  - Mapas de Karnaugh
  - Quine-McCluskey

Otimização versus *tradeoff*

- Otimização: melhoramos um ou mais critérios sem prejudicar outros
- *Tradeoffs*: melhoramos um critério às custas de outro

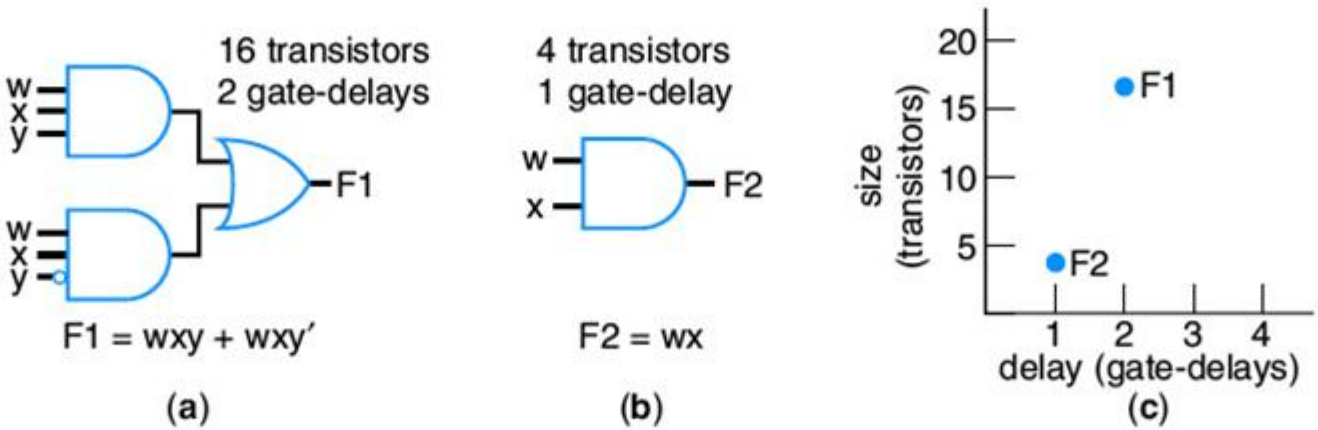


Figure 6.1 A circuit transformation that improves both size and delay, that is, an *optimization*: (a) original circuit, (b) optimized circuit, (c) plot of size and delay of each circuit.

Otimização versus *tradeoff*

- Otimização: melhoramos um ou mais critérios sem prejudicar outros
- *Tradeoffs*: melhoramos um critério às custas de outro

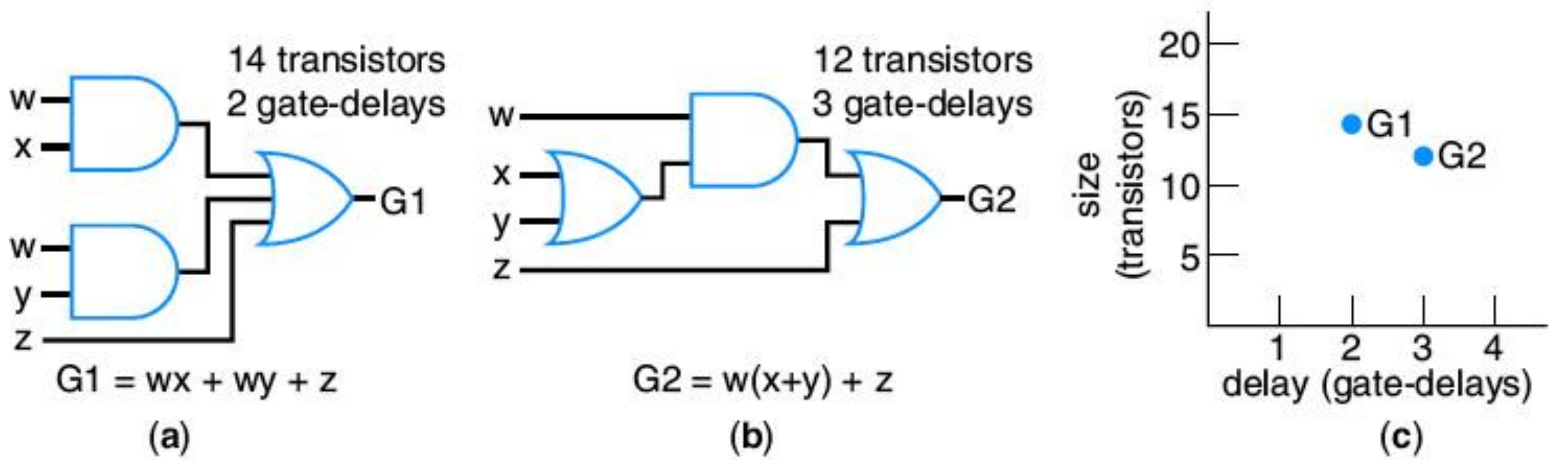


Figure 6.2 A circuit transformation that improves size but worsens delay, that is, a *tradeoff*: (a) original circuit, (b) transformed circuit, (c) plot of size and delay of each circuit.

Otimização versus *tradeoff*

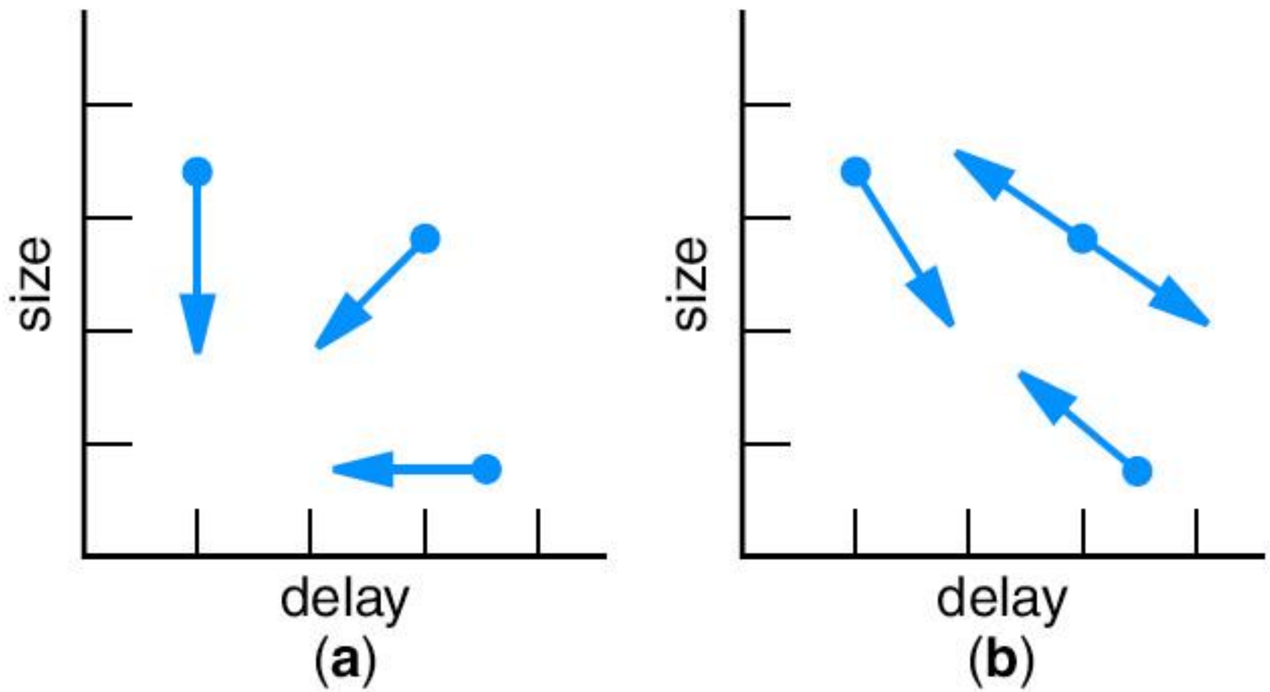


Figure 6.3 (a) Optimizations, versus (b) tradeoffs.

Minimização algébrica

- A minimização usando este método pode ser
- vista como:
- Minimizar o número de termos em uma equação booleana

Exercício

- 1-Minimizar a expressão
  $F=xyz+xyz'+x'y'z'+x'y'z$

Exercício

- 2-Minimizar a expressão
  $F=x'y'z'+x'y'z+x'yz$

Exercício

- 3-Minimizar a expressão
  $F=xy'z'+xy'z+xyz+xyz'$

Problema do método algébrico

- Dependendo da ordem dos termos é mais fácil ou não de chegar à forma mínima simples

- É difícil visualizar que o mesmo termo deve ser associado a mais de um elemento

Mapas de Karnaugh

- Método visual com o objetivo de ajudar a minimização
- Não são mais usados na prática, mas ajudam a entender os métodos básicos de otimização
- Nada mais do que uma maneira gráfica de representar uma função

Mapas de Karnaugh

- Cada célula do mapa corresponde a uma linha na tabela-verdade
- A numeração não é binária
- Células adjacentes têm apenas um bit diferente
- Adjacência-quatro
- Cilindro: primeira coluna adjacente à última

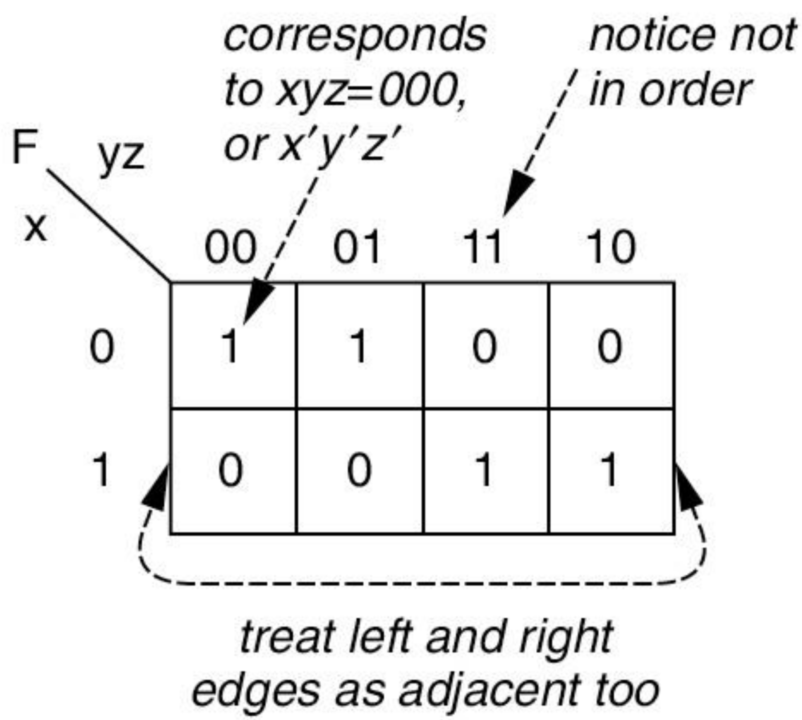


Figure 6.4 Three-variable K-map.

Mapas de Karnaugh

- Adjacente significa na horizontal ou vertical e NUNCA na diagonal
- Usam-se 1's para uma notação em soma de mintermos (implicantes)

  ou 0's para produto de maxtermos (implicados)

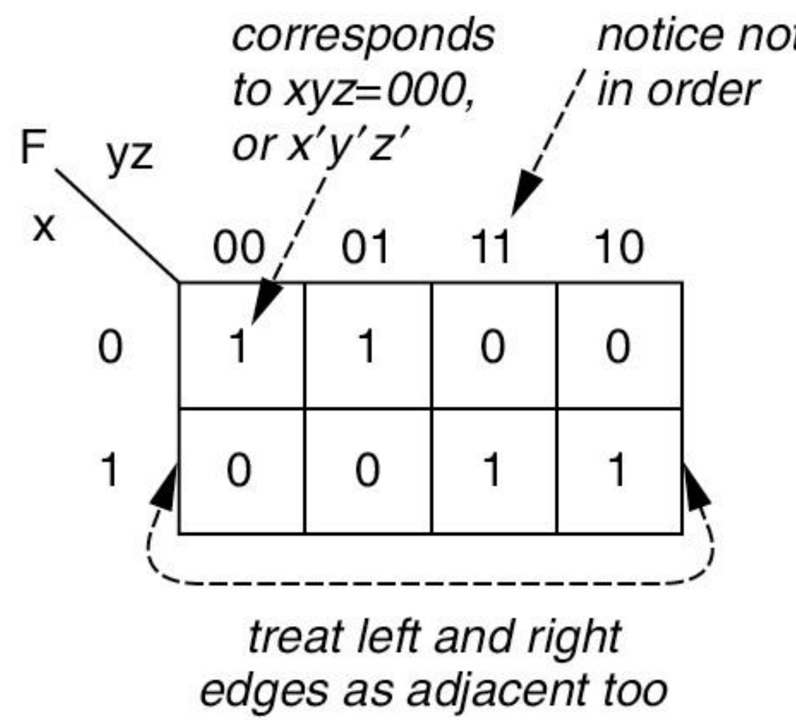


Figure 6.4 Three-variable K-map.

Mapas de Karnaugh

- Unir dois 1's adjacentes significa agrupar dois mintermos e eliminar uma variável!
- Então podemos agrupar 1's e simplificar a expressão da função

  binária representada

  pelo mapa.

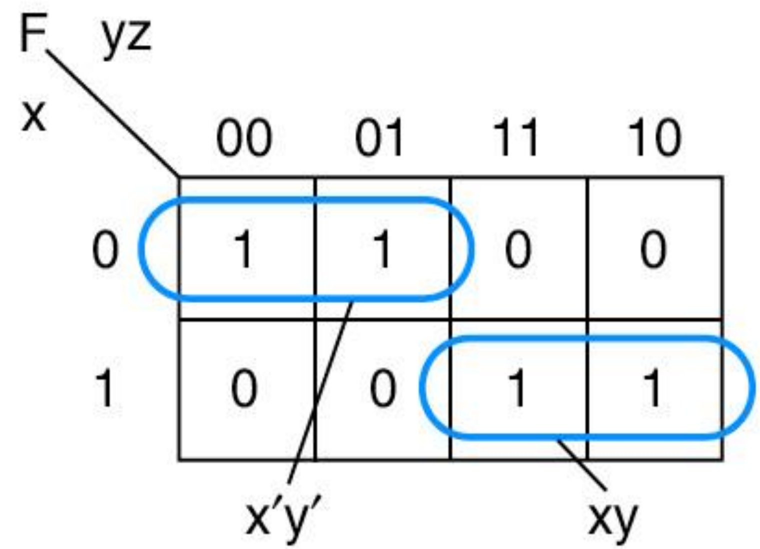


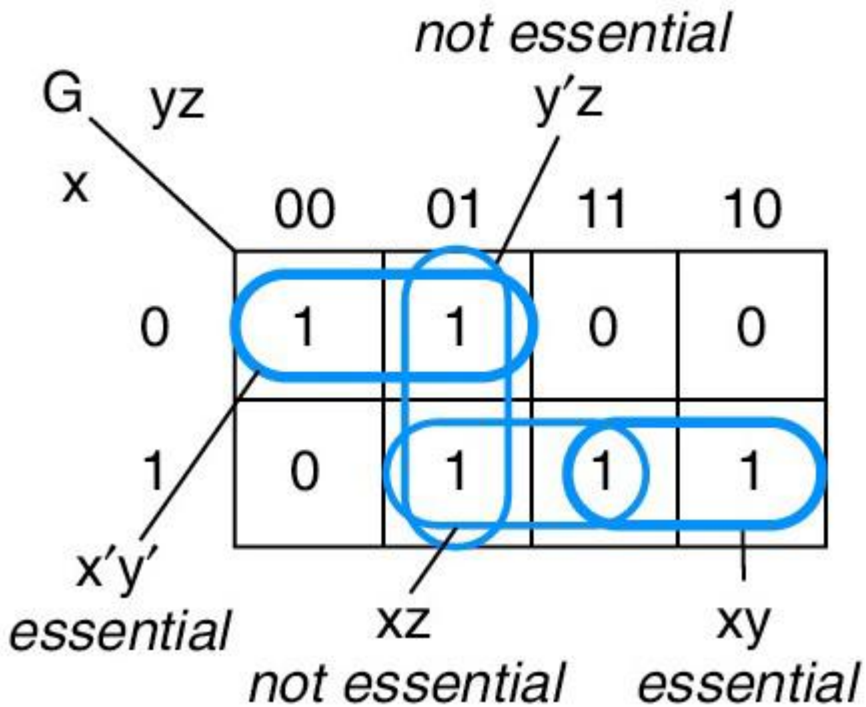
Figure 6.5 Minimizing a three-variable function using a K-map.

Mapa de Karnaugh

- Algumas definições:
- Implicante: qualquer termo produto que faz F=1 na tabela (mintermo ou grupo de 1's)
- Implicante primo: maior termo produto que faz F=1, maior grupo de 1's que se consegue fazer em uma determinada vizinhança. Se for aumentado irá cobrir um 0 (zero) na tabela.
- Implicante primo essencial: o ÚNICO implicante primo que cobre um dado mintermo pertencente ao conjunto-1 da função.

Mapa de Karnaugh

- Exemplo:



Mapa de Karnaugh

- Como fazer grupos de 1's?
- Devemos juntar 1, 2, 4 ou 8 células.
- Por que? Grupos de 2^n células adjacentes incluem n variáveis em todas as combinações possíveis (cada variável no modo normal e complementado).

- Agrupar 3, 5, 6 ou 7 células não elimina variáveis, já que não se combinam todos os possíveis valores!!!!

Mapa de Karnaugh

- Como fazer grupos de 1's?

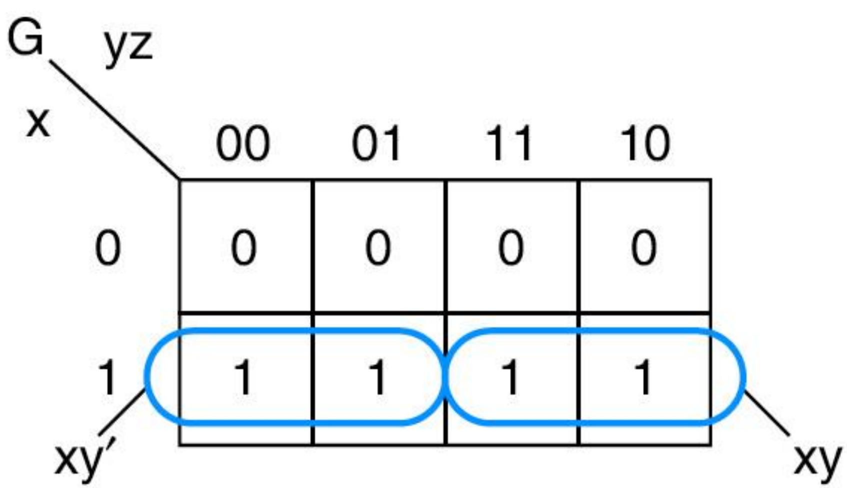


Figure 6.7 Nonoptimal circles.

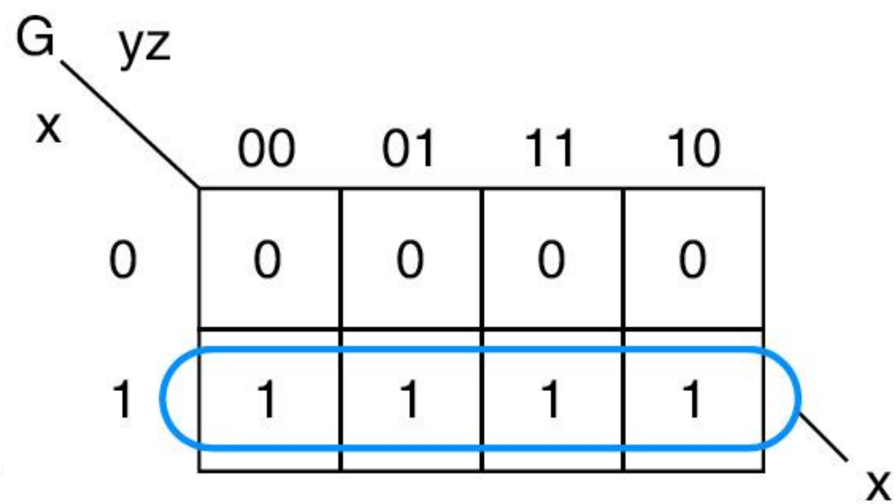


Figure 6.6 Four adjacent 1s.

Mapa de Karnaugh

- Como fazer grupos de 1's?

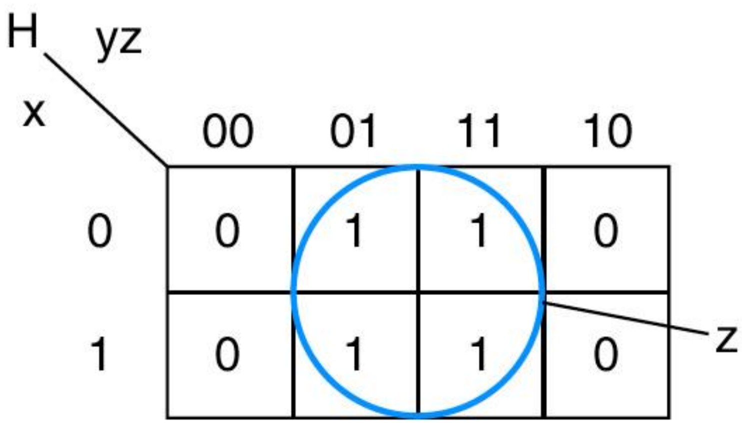


Figure 6.8 Four adjacent 1s.

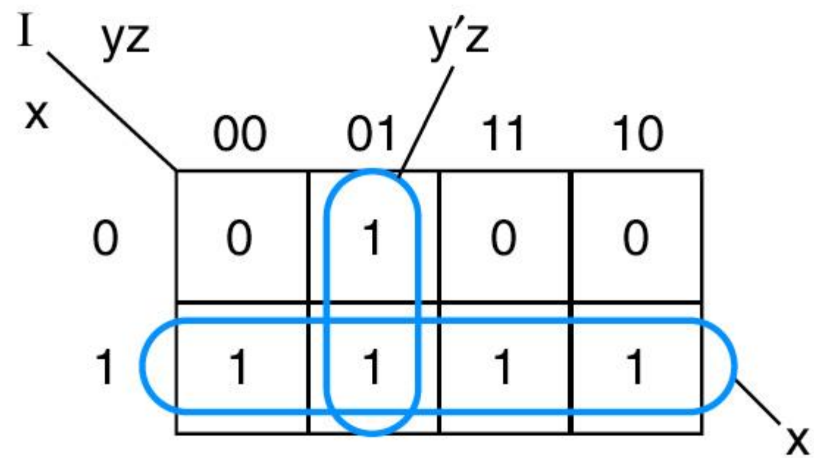


Figure 6.9 Circling a 1 twice.

Mapa de Karnaugh

- Como fazer grupos de 1's?

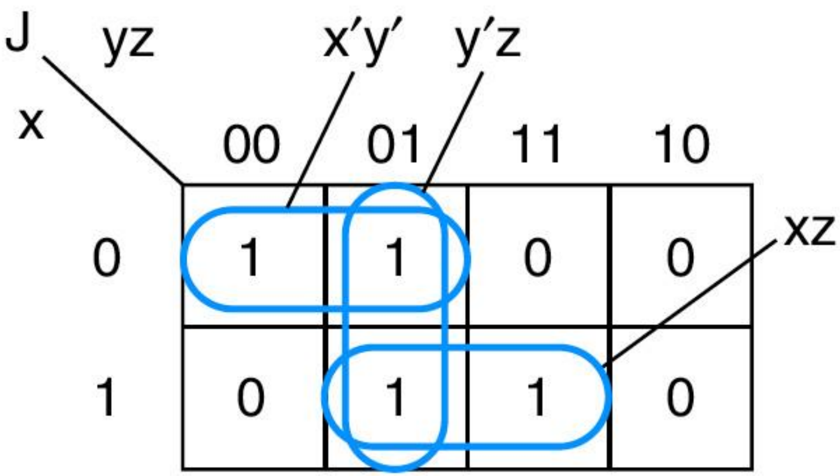


Figure 6.10 An unnecessary term.

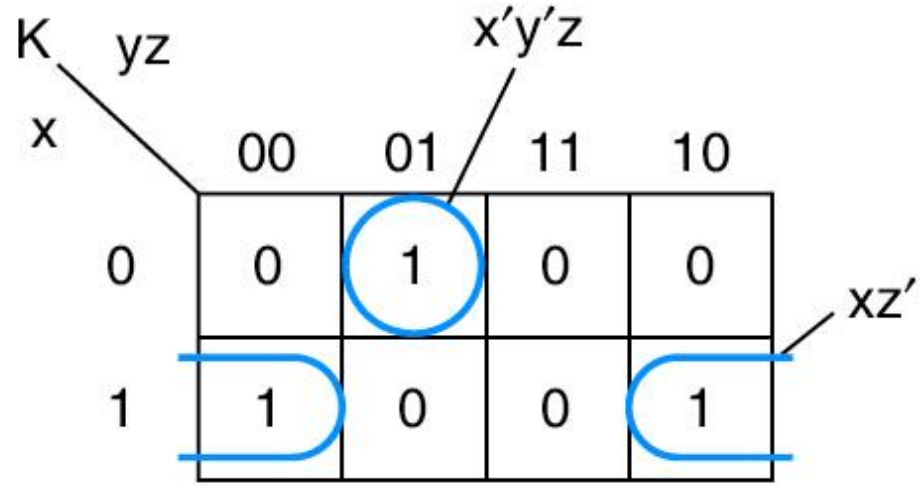


Figure 6.11 Sides are adjacent.

Mapa de Karnaugh

- Como fazer grupos de 1's?

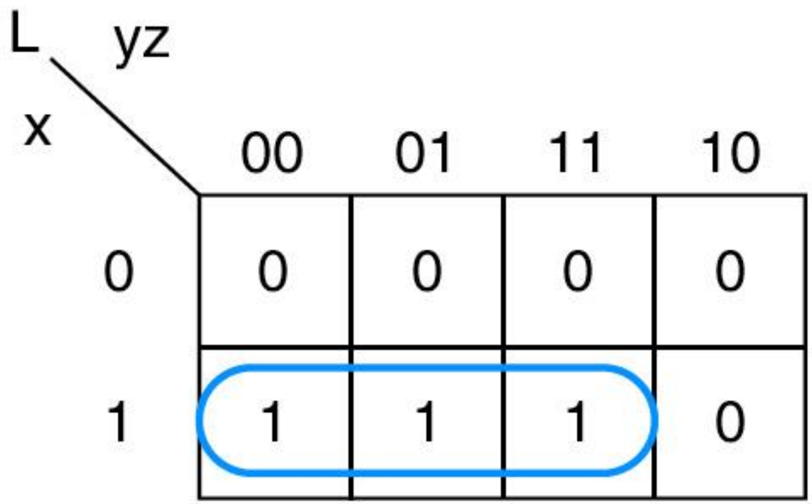


Figure 6.12 Invalid circle.

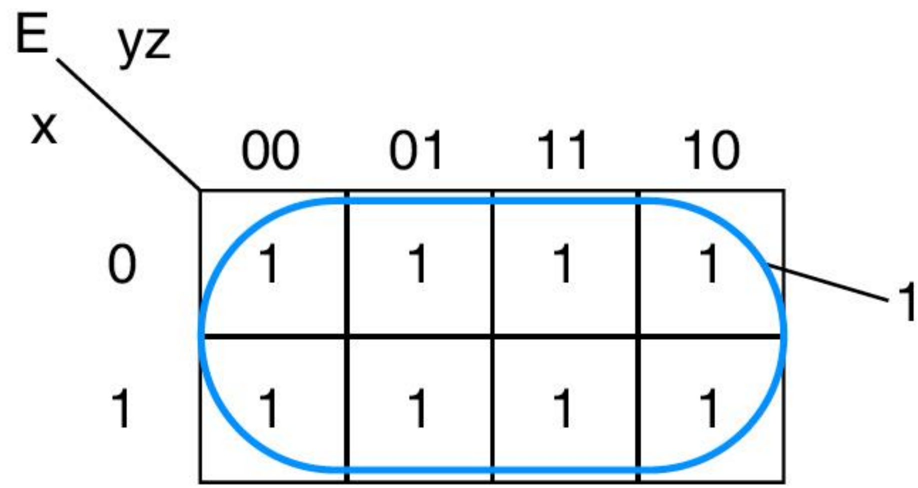


Figure 6.13 Four adjacent 1s.

Mapa de Karnaugh

Mapa de 4 variáveis: a primeira e a última

lun

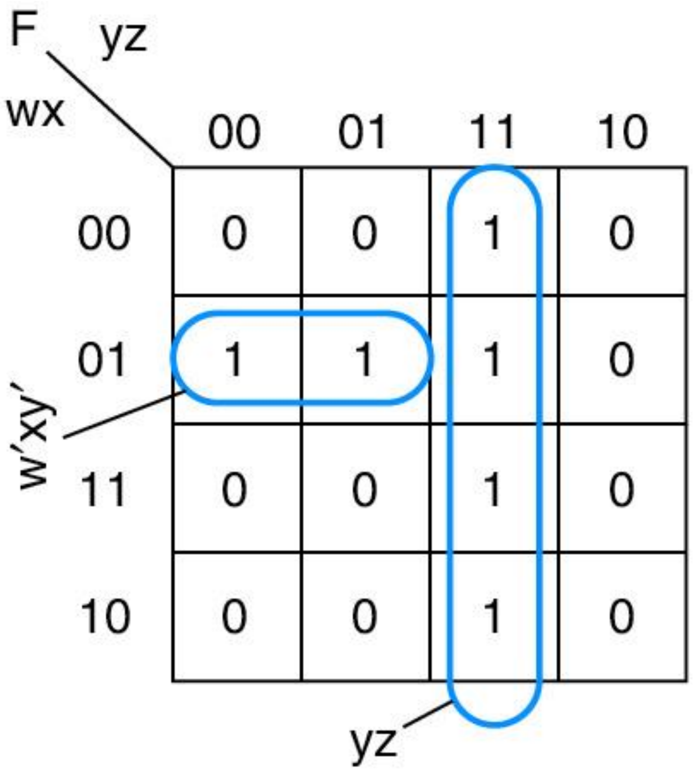


Figure 6.14 Four-variable K-map.

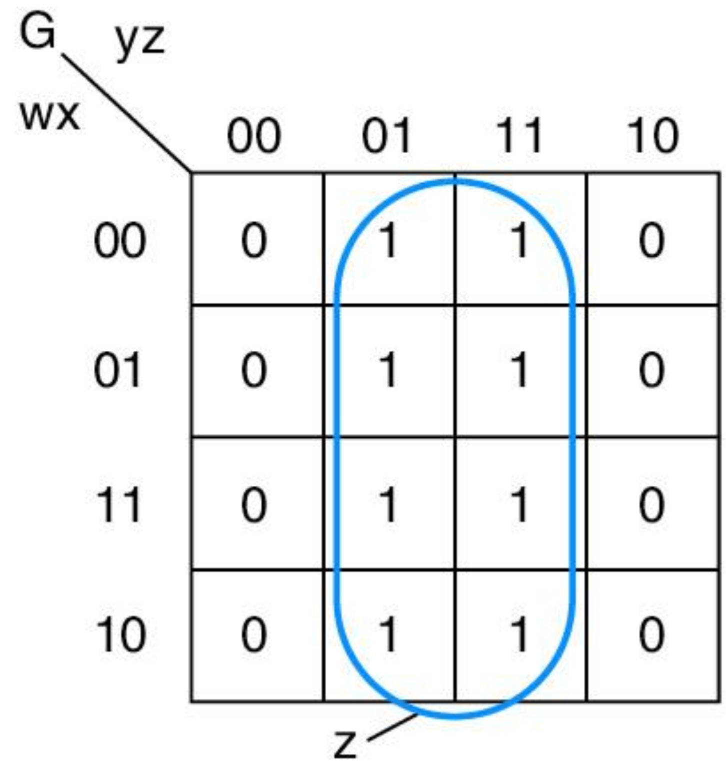


Figure 6.15 Eight adjacent cells.

Mapa de Karnaugh

- Mapa de 4 variáveis:
- A primeira e a última linhas são vizinhas
- A primeira e a última colunas são vizinhas
- É um toróide em 3D

Mapa de Karnaugh

Mapa de 2 variáveis

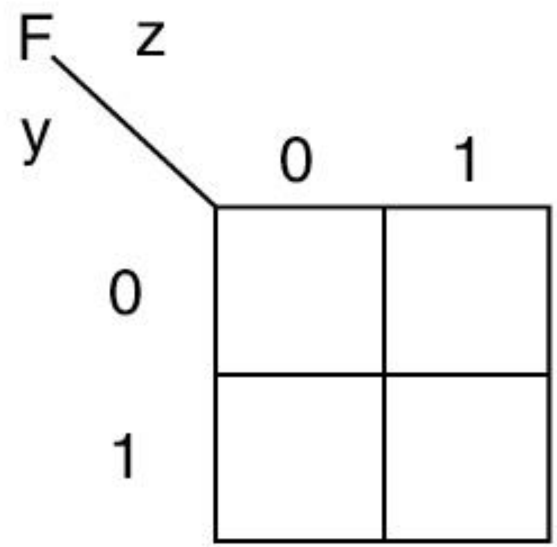


Figure 6.16 Two-variable K-map.

Mapas de Karnaugh

- Para mais de 4 variáveis são muito complicados para serem aplicados na prática.
- Para 5 variáveis, precisamos de 2 mapas de Karnaugh.
  - Ex: F(A,B,C,D,E) → um mapa para A' e outro mapa para A
  - Um mapa é paralelo ao outro no espaço
- Para 6, já são 4!!!
  - Ex: F(A,B,C,D,E,F) → mapas para A'B', A'B, AB' e AB
  - Difícil visualizar vizinhança...

Mapa de Karnaugh

- Como obter a função booleana simplificada?
  - Formar os maiores grupos possíveis e
  - Usar o menor número de grupos de 1s (ou 0's, conforme desejado)

- Procedimento para Soma de Produtos:
  1. Converta a função da equação para uma soma de mintermos ou faça a tabela-verdade
  2. Coloque um 1 na célula apropriada
  3. Cubra todos os 1's agrupando-os em número mínimo de maiores grupos possíveis
  4. Faça o OR de todos os termos resultantes

Mapa de Karnaugh

- **Procedimento prático**
- Agrupar implicantes primos essenciais:
  1. Comece agrupando os elementos que não têm nenhum vizinho 1. Este formará um grupo com 1 elemento
  2. Agrupe em seguida o elemento que só pode ser agrupado em um par, formando assim um implicante primo com 2 elementos
  3. Em seguida, agrupe os 1s que só pode se agrupar em uma quadra
  4. Assim por diante até que todos os implicantes primos essenciais sejam formados.
- Se restar algum 1 descoberto, use o maior implicante primo não essencial para cobri-lo.
- Escreva uma expressão de produto representando cada grupo formado
- Faça a soma de todos os termos resultantes

Mapa de Karnaugh

- Um grupo com 1 elemento não elimina variável
- Um grupo com 2 elementos elimina 1 variável
- Um grupo com 4 elementos elimina 2 variáveis
- Um grupo com 8 elementos elimina 3 variáveis
- Um grupo com 2^n elementos elimina n variáveis
- A expressão de produto representando cada grupo é formada associando-se a variável normal, se seu valor no grupo for 1, e a variável complementada, se seu valor for 0
- A(s) variável(is) eliminada(s) muda(m) de valor no grupo

Mapa de Karnaugh–Exemplos

- Minimize a função F=w’xz+yz+w’xy’z’
  - Para preencher o mapa você pode observar que o primeiro termo engloba as duas células em azul claro na figura, o segundo termo corresponde à quadra vertical em azul e o último termo corresponde à célula com 1 à esquerda

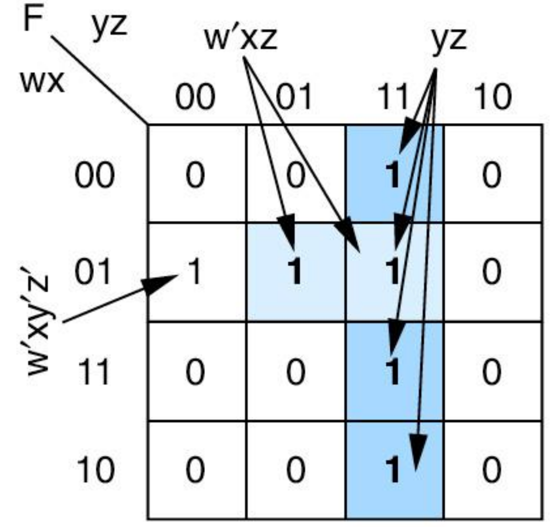


Figure 6.17 w ' xz and yz terms.

Mapa de Karnaugh–Exemplos

- Minimize a função F=w’xz+yz+w’xy’z’

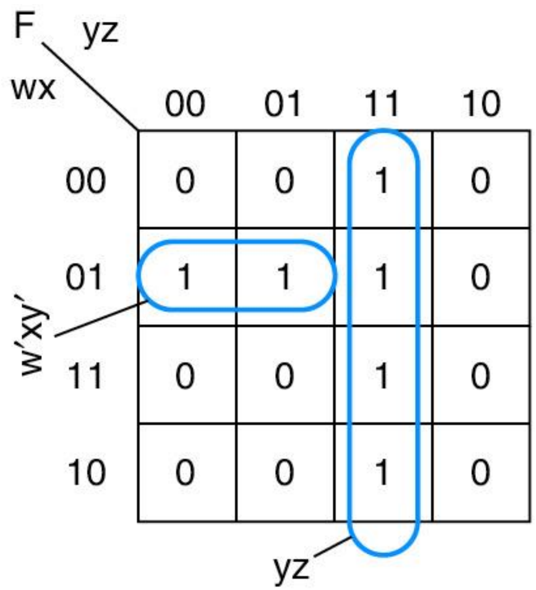


Figure 6.14 Four-variable K-map.

  - Solução: F=w’xy’+yz

Mapa de Karnaugh – Exemplos

- Minimize a função G = a+a’b’c’+b(c’+bc’)
  - Aplicando a propriedade x+xy = x (lei da absorção) no termo entre parênteses temos

    G = a+a’b’c’+bc’
  - Preenchendo o mapa

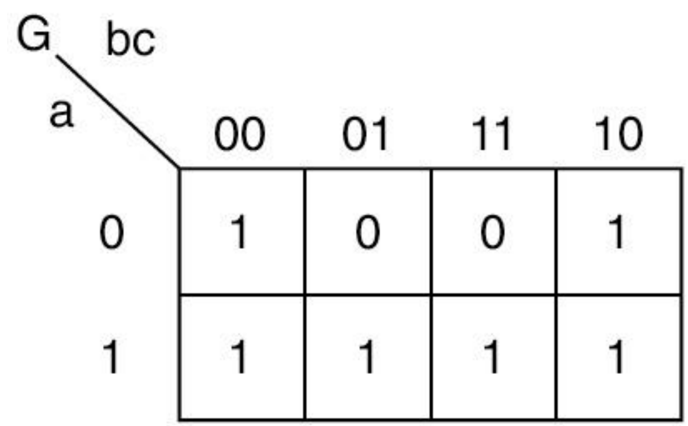


Figure 6.18 Terms on the K-map.

Mapa de Karnaugh – Exemplos

- Minimize a função G = a+a'b'c'+b(c'+bc')
  - Agrupando os 1s

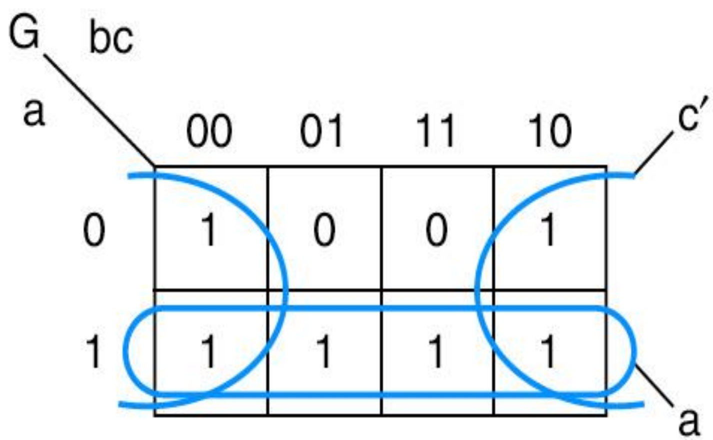


Figure 6.19 A cover.

  - G = a+c'

Mapa de Karnaugh – Exemplos

- Minimize a função

$H=a'b'(cd'+c'd')+ab'c'd'+ab'cd'+a'bd+a'bcd'$

  - Aplicando a distributiva e montando o mapa

Mapa de Karnaugh – Exemplos

- Minimize a função

H=a’b’(cd’+c’d’)+ab’c’d’+ab’cd’+a’bd+a’bcd’

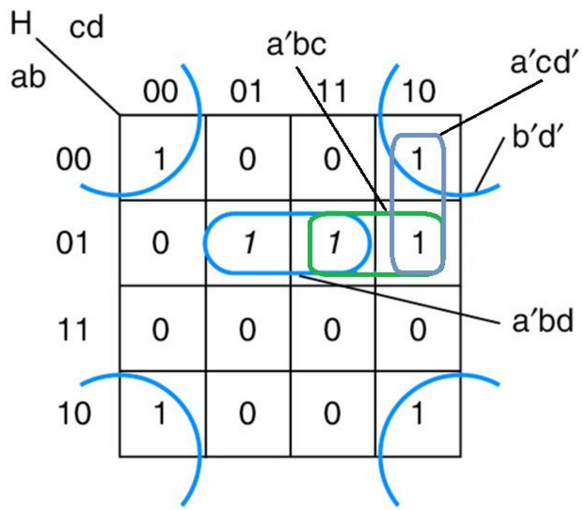


Figure 6.20 K-map example.

H=a’bd+b’d’+a’bc ou H=a’bd+b’d’+a’cd’

Combinações don’t care de entrada

- Algumas vezes temos a garantia de que algumas combinações de entrada nunca poderão ocorrer.
- Para estas combinações não importa se a saída é 0 ou 1, sendo chamada don’t care
- Podemos escolher a saída 0 ou 1 de forma a obter a melhor otimização possível
- Representamos don’t care como x no mapa de Karnaugh e nas tabelas-verdade

Mapas de Karnaugh com don’t care

- Fazemos o termo don’t care igual a 1 para aumentar grupos de 1s

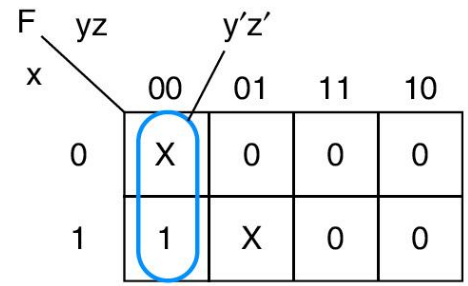


Figure 6.21 Map with don’t cares.

- Não fazemos grupos para cobrir saídas don’t care. Neste caso, melhor considerar o don’t care igual a 0

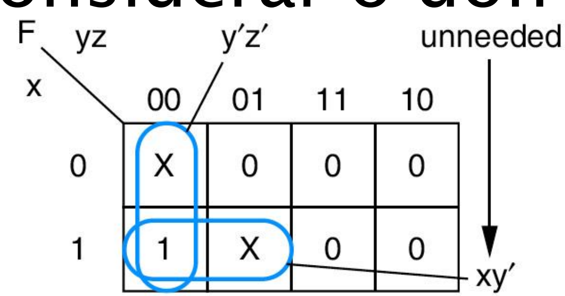


Figure 6.22 Wasteful use of Xs.

Mapas de Karnaugh com don't care

- Minimize a expressão F=a'bc'+abc'+a'b'c sabendo que a'bc e abc são don't care

Mapas de Karnaugh com don't care

- Minimize a expressão F=a'bc'+abc'+a'b'c sabendo que a'bc e abc são don't care

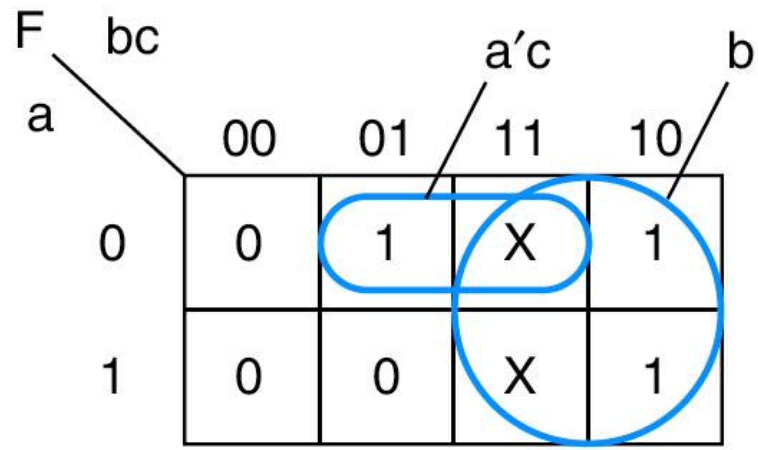


Figure 6.23 Using don't cares.

Mapas de Karnaugh com don't care

- Considere a chave deslizante da figura que tem 3 bits de saída (x, y e z) e que pode estar em uma de cinco posições (xyz=001, 010, 011, 100, 101). As outras combinações (000, 110 e 111) não são possíveis e a saída é don't care.
- Projete um sistema mínimo que com as entradas x, y e z forneça uma saída G=1 se a chave estiver nas posições 2, 3 ou 4.

Mapas de Karnaugh com don't care

- Considere a chave deslizante da figura que tem 3 bits de saída (x, y e z) e que pode estar em uma de cinco posições (xyz=001, 010, 011, 100, 101). As outras combinações (000, 110 e 111) não são possíveis e a saída é don't care.
- Projete um sistema mínimo que com as entradas x, y e z forneça uma saída G=1 se a chave estiver nas posições 2, 3 ou 4.

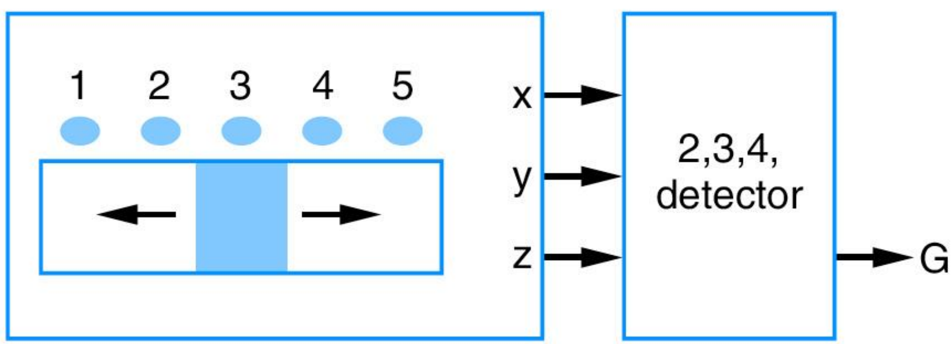


Figure 6.24 Sliding switch example.

Mapas de Karnaugh com don't care

- G=y+z'

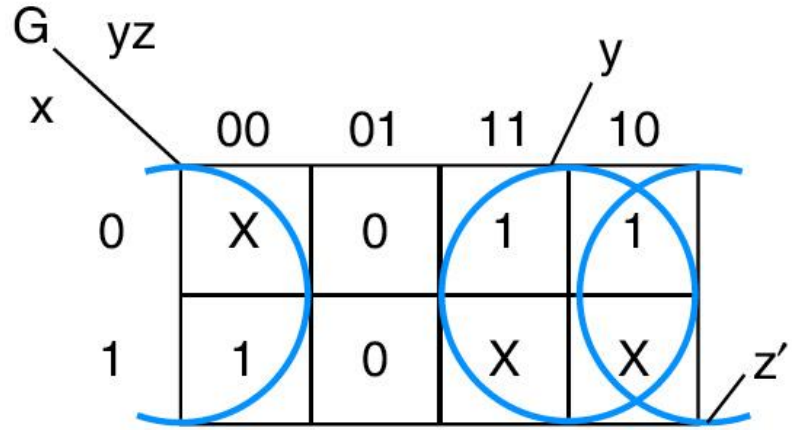


Figure 6.26 With don't cares.

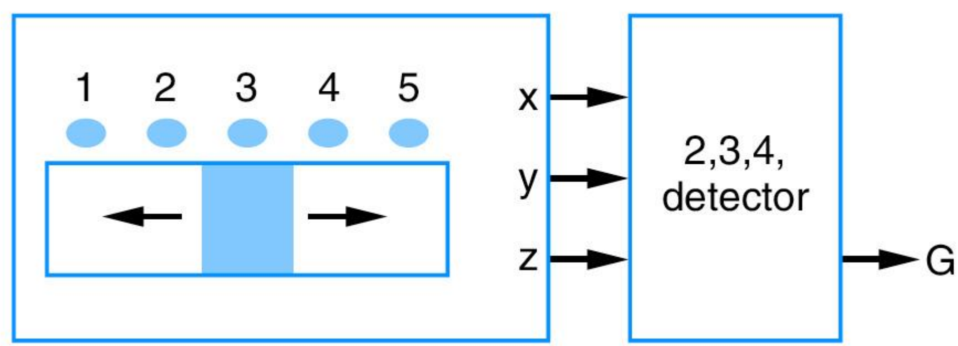


Figure 6.24 Sliding switch example.

Mapas de Karnaugh com don’t care

- $G = xy'z' + x'y$

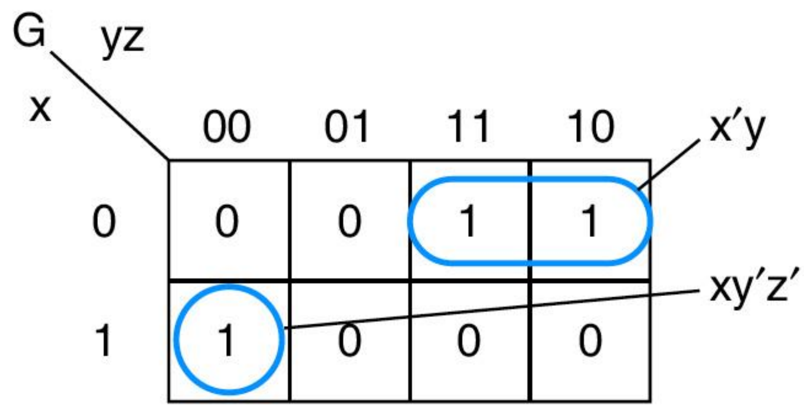


Figure 6.25 Without don’t cares.

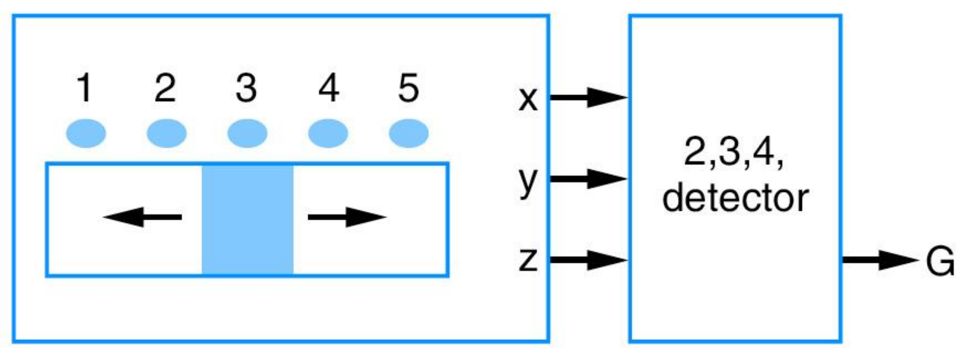


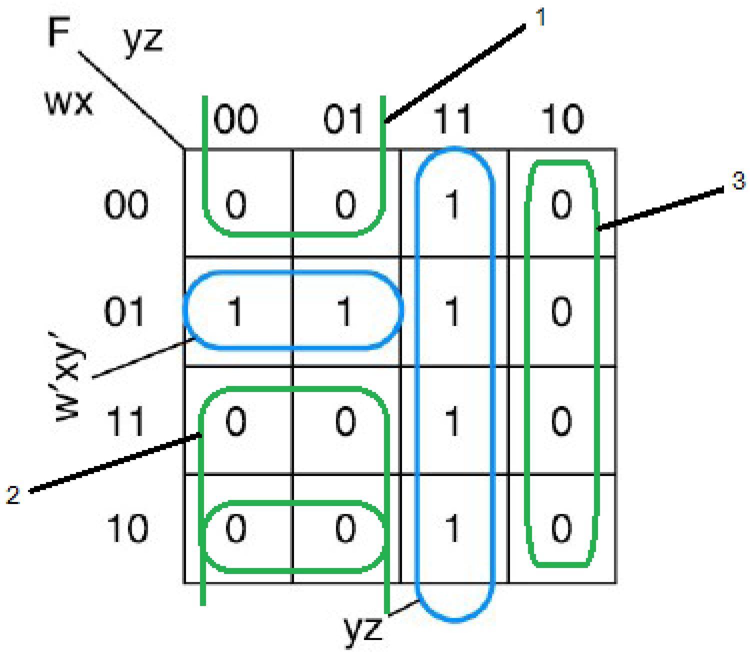
Figure 6.24 Sliding switch example.

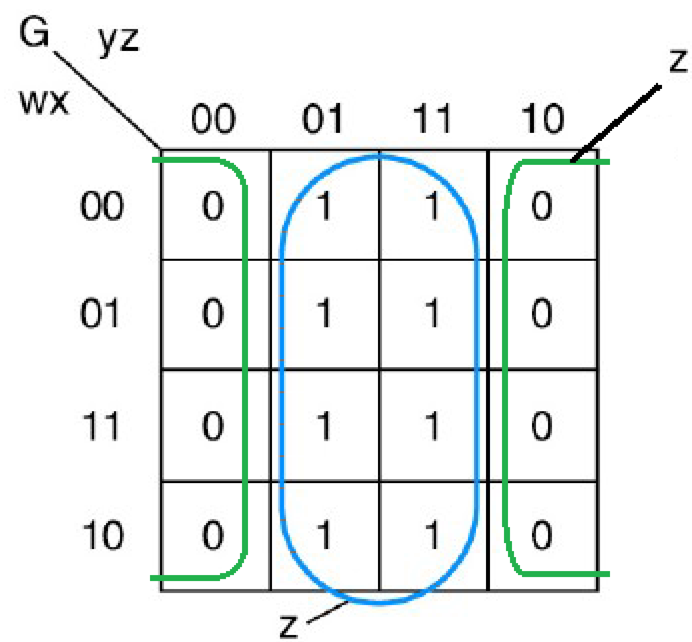
Mapas de Karnaugh – grupos de 0’s

- Algumas definições:
- Implicado: qualquer termo de soma que faz F=0 na tabela (maxtermo ou grupo de 0’s)
- Implicado primo: maior termo de soma que faz F=0, maior grupo de 0’s que se consegue fazer em uma determinada vizinhança. Se for aumentado irá cobrir um 1 (um) na tabela.
- Implicado primo essencial: o ÚNICO implicado primo que cobre um dado maxtermo pertencente ao conjunto-0 da função.

Mapas de Karnaugh – grupos de 0's

- F=(x+y)(w'+y)(y'+z) G=z





Problemas com mapas de Karnaugh

- Fica difícil visualizar efetivamente funções de mais de 5 ou 6 variáveis
- É suscetível a falhas humanas
  - A ordem pela qual o projetista começa a cobrir os 1's pode levar a funções com mais termos

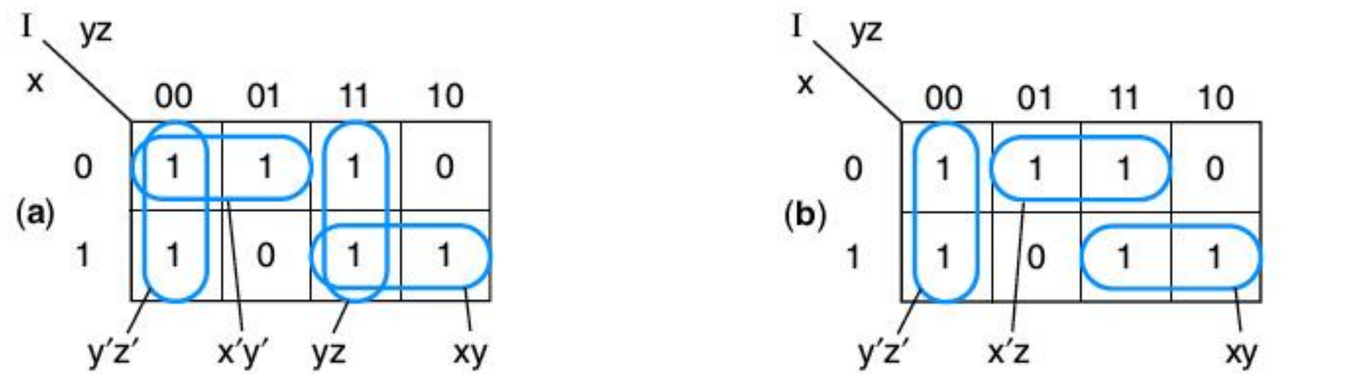


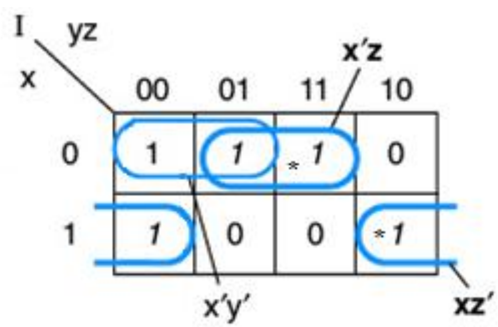
Figure 6.27 A cover is not necessarily optimal: (a) a four-term cover, and (b) a three-term cover of the same function.

Otimização automatizada

1. Determine os implicantes primos
2. Encontrar todos os implicantes primos essenciais
3. Cobrir todos os demais mintermos com implicantes não essenciais (usando o mínimo de implicantes primos)

Exemplo

- Na letra b os asteriscos (*) marcam os implicantes primos essenciais
- Na letra c o implicante primo vertical foi escolhido
- F=x'z+xz'+y'z'
- Mas poderia ter sido o horizontal
- F=x'z+xz'+x'y'



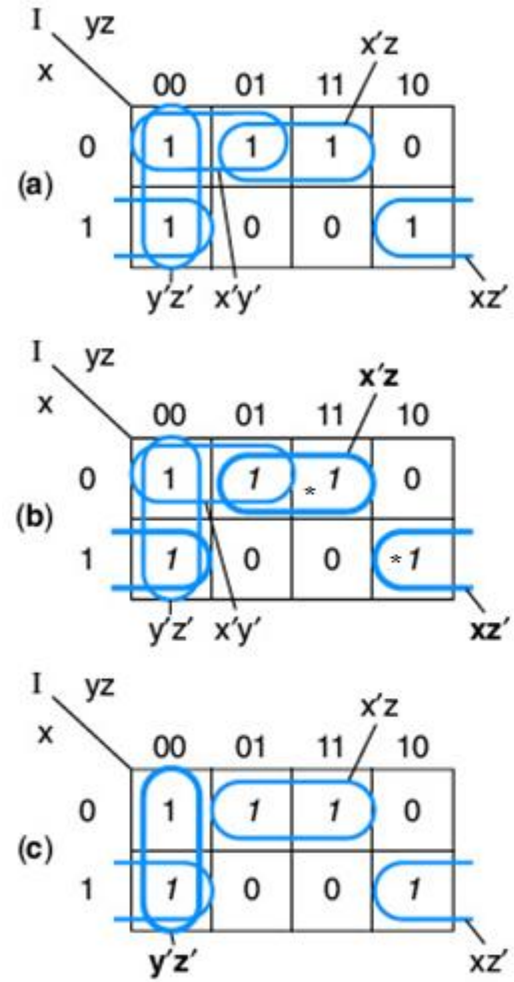


Figure 6.30 Illustration of two-level optimization: (a) all prime implicants, (b) including essential prime implicants in the cover, (c) covering remaining 1s.

Otimizacao automatizada usando Quine McCluskey

- Método Tabular
- Dividido em duas etapas:
  - Tabela de Implicantes primos
  - Tabela de Implicantes primos essenciais

Exemplo do Método de Quine McCluskey

- Considere a função

$f(A,B,C,D) = \Sigma m(4,5,6,8,9,10,13) + \Sigma d(0,7,15)$

- Estágio 1: encontrar todos os implicantes primos
- Passo 1.1: preencha a Coluna 1 com o conjunto de mintermos (1's) e don't cares (X's) da função agrupados pelo número de 1's

Tabela de Implicantes	
Coluna 1	
0000	
0100	
1000	
0101	
0110	
1001	
1010	
0111	
1101	
1111	

Exemplo do Método de Quine McCluskey

- Passo 1.2: aplique o teorema da união
  - Compare elementos de um grupo com N 1's com os elementos com N+1 1's
  - Se diferirem em apenas um bit então são adjacentes.
  - Elimine a variável que muda e coloque na coluna seguinte.
  - Exemplos:
    - 0000 vs. 0100 da 0-00
    - 0000 vs. 1000 da -000
  - Quando combinados marcar com um visto (√).
  - Se não puder ser combinado marcar com um asterisco (*).
  - Os asteriscos identificam os implicantes primos.

Tabela de Implicantes	
Coluna 1	Coluna 2
0000√	0_00
0100√	_000
1000√	010_
0101√	01_0
0110√	100_
1001√	10_0
1010√	01_1
0111√	_101
1101√	011_
1111√	1_01
	_111
	11_1

Exemplo do Método de Quine McCluskey

- Passo 1.3: Repetir combinando elementos da coluna 2 de grupos adjacentes gerando a coluna 3.
  - Para combinar elementos de colunas com implicantes com "_" eles têm de diferir em um único bit e terem o(s) "_" na mesma posição.
  - Na coluna 3 aparecem grupos repetidos. Considere uma vez apenas.
- Passo 1.4: Repetir combinando elementos da coluna 3.
  - Como não é possível combinar, o primeiro estágio acabou.

Tabelas de implicantes		
Coluna 1	Coluna 2	Coluna 3
0000√	0_00*	
0100√	_000*	
1000√	010_√	01_ _*
0101√	01_0√	~~01_ _~~
0110√	100_*	
1001√	10_0*	
1010√	01_1√	_1_1*
0111√	_101√	~~_1_1~~
1101√	011_√	
1111√	1_01*	
	_111√	
	11_1√	

Exemplo do Método de Quine McCluskey

- Tabela de implicantes primos essenciais

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

linhas = implicantes primos
colunas = elementos do conjunto de "1's"
coloque um "X" se um elemento do conjunto de "1's" é coberto pelo implicante primo

Exemplo do Método de Quine McCluskey

- Tabela de implicantes primos essenciais

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

Se a coluna tiver apenas um X, o implicante associado com a linha respectiva é essencial. Tem de aparecer na cobertura mínima.

Exemplo do Método de Quine McCluskey

- Tabela de implicantes primos essenciais

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

Eliminar todas as colunas cobertas por implicantes primos essenciais

Exemplo do Método de Quine McCluskey

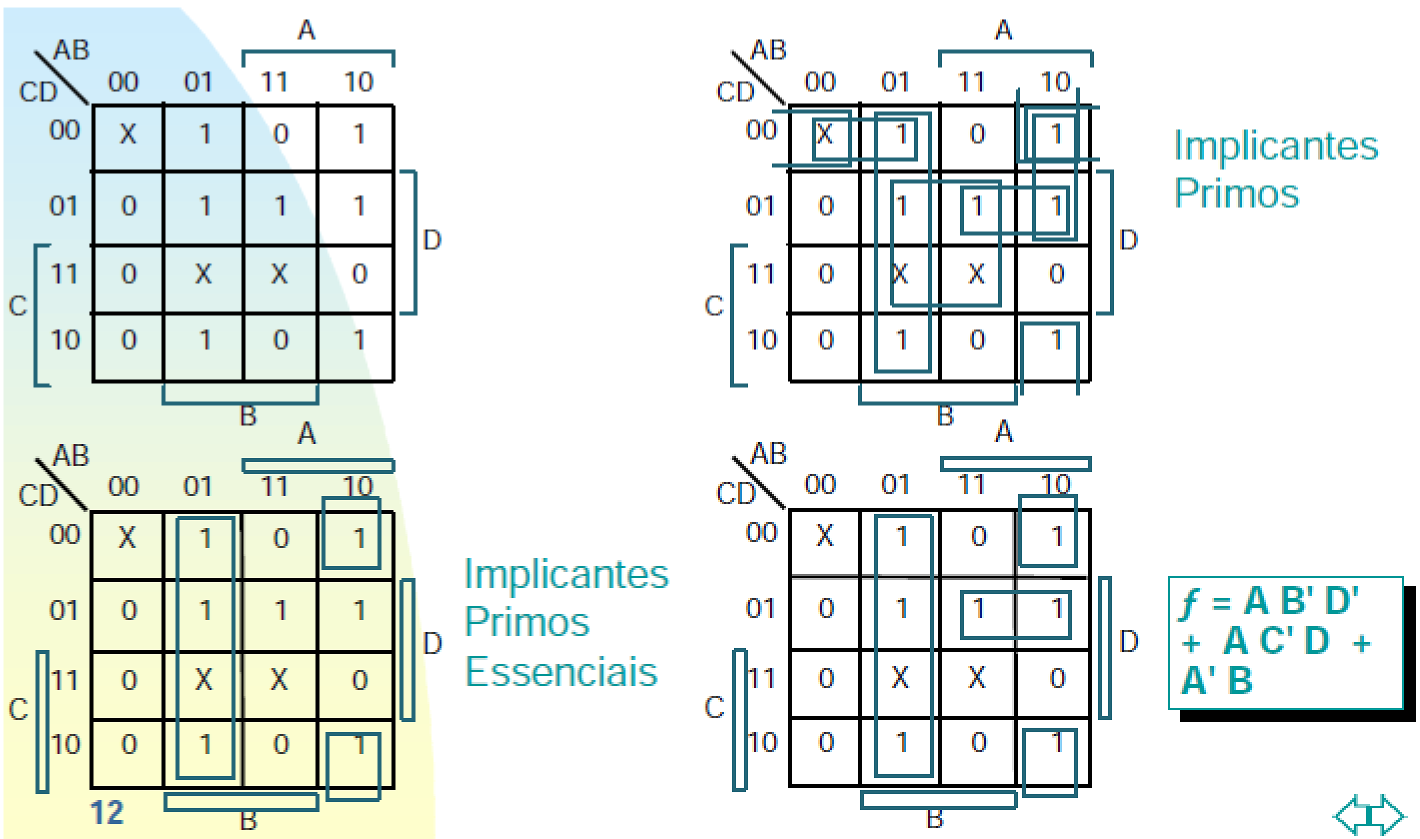
- Tabela de implicantes primos essenciais

	4	5	6	8	9	10	13
0,4 (0-00)	X						
0,8 (-000)				X			
8,9 (100-)				X	X		
8,10 (10-0)				X		X	
9,13 (1-01)					X		X
4,5,6,7 (01--)	X	X	X				
5,7,13,15 (-1-1)		X					X

Encontrar o conjunto mínimo de linhas que cobre as colunas remanescentes

$f = A B' D' + A C' D + A' B$

Mesmo exemplo com mapa de Karnaugh



Para ser continuado...