MC 602

Circuitos Lógicos e Organização de Computadores

IC/Unicamp

Prof Mario Côrtes

Capítulo 4

Síntese e minimização de circuitos combinacionais

Tópicos

- Mapas de Karnaugh
- Minimização
  – terminologia e procedimento
  – SOP e POS
- Funções incompletamente especificadas
- Circuitos multi-saída
- Circuitos multi-nível
- Método tabular
- Técnica de minimização por cubos lógicos

IC-UNICAMP

Síntese manual de um circuito (sec 2.6)

Row number	x_1	x_2	x_3	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

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

Minimização algébrica

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

$m_0 + m_2 + m_4 + m_5 + m_6$

$f = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_2 x_3 + x_1 x_2 \bar{x}_3$

$A = m_0 + m_2 = \bar{x}_1\bar{x}_3$

$B = m_4 + m_5 = x_1\bar{x}_2$

$C = m_4 + m_6 = x_1\bar{x}_3$

$D = A + C = \bar{x}_3$

$f = B + D = \bar{x}_3 + x_1\bar{x}_2$

OBS: notem que m_4 foi "duplicado" ($m_4+m_4=m_4$)

e que m_4 foi combinado duas vezes: com m_5 e com m_6

IC-UNICAMP

Simplificações típicas

- Na manipulação algébrica, usamos sucessivas vezes
  - dois termos que diferiam de apenas um literal eram combinados para gerar um termo com um literal a menos
- Mapa de Karnaugh:
  - rearranjo da tabela verdade para facilitar a identificação de termos “vizinhos”
- Função com n variáveis
  - Tabela verdade → 2^n linhas
  - Mapa de Karnaugh→ 2^n células

Mapa de Karnaugh de 2 variáveis

$(m_0,m_2)=(\bar{x}_1\bar{x}_2,\ x_1\bar{x}_2)$

$(m_1,m_3)=(\bar{x}_1 x_2,\ x_1 x_2)$

$(m_0,m_1)=(\bar{x}_1\bar{x}_2,\ \bar{x}_1 x_2)$

$(m_2,m_3)=(x_1\bar{x}_2,\ x_1 x_2)$

x_1	x_2	
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

(a) Tabela verdade

$x_2 \backslash x_1$	0	1
0	m_0	m_2
1	m_1	m_3

(b) Mapa de Karnaugh

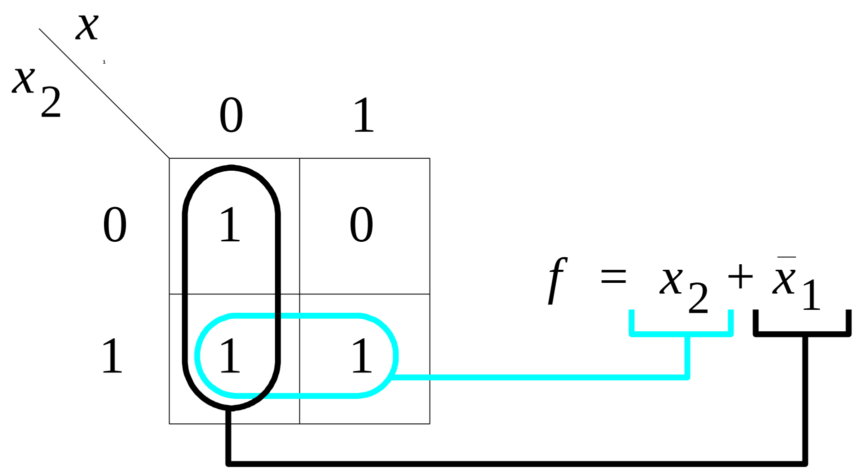
- Mintermos vizinhos no Mapa de Karnaugh (horizontal e vertical) diferem de apenas um literal

IC-UNICAMP

Uma função simples (fig 2.15)

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

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1



- Possibilidades para cobrir todos os mintermos

  $x_2 + \overline{x}_1.\overline{x}_2$ $\quad \overline{x}_1 + x_1.x_2$ $\quad \overline{x}_1 + x_2$

- Qual tem o menor custo?

  custo = nº total de gates + nº total de entradas

IC-UNICAMP

Mapa de Karnaugh de 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

$x_3 \backslash x_1x_2$	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

- Mintermos vizinhos no Mapa de Karnaugh (horizontal e vertical) diferem de apenas um literal
- Observar que também são vizinhos (m_0,m_4) e (m_1,m_5)

IC-UNICAMP

Exemplos de funções de 3 variáveis

Função da Fig. 2.18

$x_3 \backslash x_1x_2$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$$f = x_1\bar{x}_3 + \bar{x}_2 x_3$$

Função da Fig. 4.1

$x_3 \backslash x_1x_2$	00	01	11	10
0	1	1	1	1
1	0	0	0	1

$$f = \bar{x}_3 + x_1\bar{x}_2$$

Mapa de Karnaugh de 4 variáveis

- Mintermos vizinhos no Mapa de Karnaugh (horizontal e vertical) diferem de apenas um literal
- Observar que também são vizinhos os mintermos das colunas (00,10) e das linhas (00,10)

x_1

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}

x_2

x_3

x_4

Exemplos de M.K. de 4 variáveis

IC-UNICAMP

x_3x_4 \ x_1x_2	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	0	0	1
10	1	0	0	1

$$f_1 = \bar{x}_2x_3 + x_1\bar{x}_3x_4$$

x_3x_4 \ x_1x_2	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	1	1	1
10	1	1	1	1

$$f_2 = x_3 + x_1x_4$$

x_3x_4 \ x_1x_2	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	1	1	1	0
10	1	1	0	1

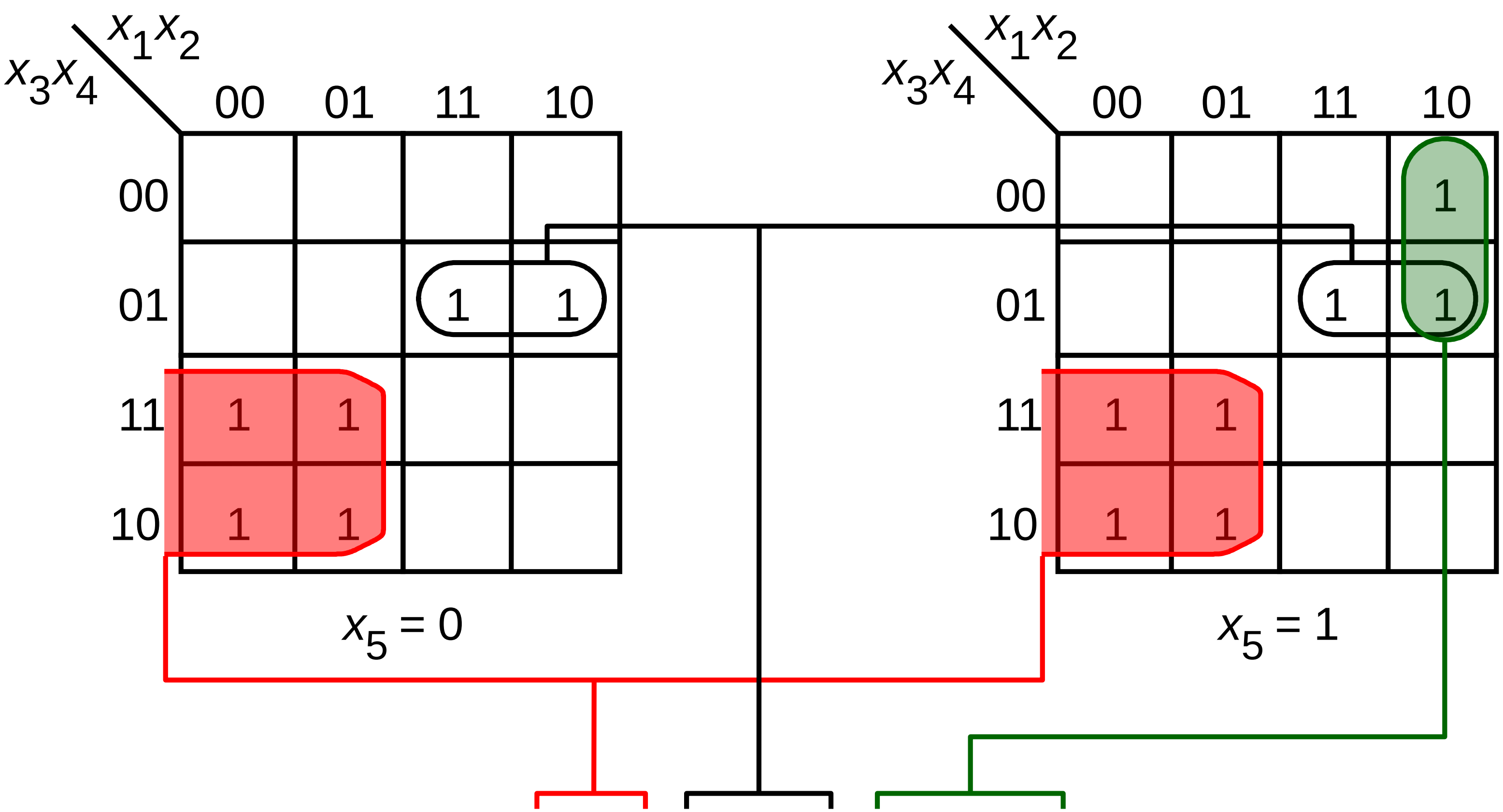
$$f_3 = \bar{x}_2\bar{x}_4 + \bar{x}_1x_3 + x_2x_3x_4$$

x_3x_4 \ x_1x_2	00	01	11	10
00	1	1	1	0
01	1	1	1	0
11	0	0	1	1
10	0	0	1	1

$$f_4 = \bar{x}_1\bar{x}_3 + x_1x_3 + \begin{array}{c} x_1x_2 \\ \text{or} \\ x_2\bar{x}_3 \end{array}$$

IC-UNICAMP

Mapa de Karnaugh de 5 variáveis



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

Mapas de Karnaugh: observações

- E para 6 variáveis? (tentar)
- E para 7? Ou mais?

- Exercício: calcular o número possível de funções lógicas de n variáveis e uma saída
  - dica: raciocinar com a tabela verdade ou o Mapa de Karnaugh

Estratégia para minimização

- Exemplos vistos de Mapas de Karnaugh:
  - minimização feita intuitivamente, por tentativa e erro
  - buscar maiores grupos de mintermos vizinhos que pudessem cobrir todos os 1s
- Veremos agora uma estratégia estruturada

  → terminologia

Terminologia para f $(x_1 \ldots x_n)$

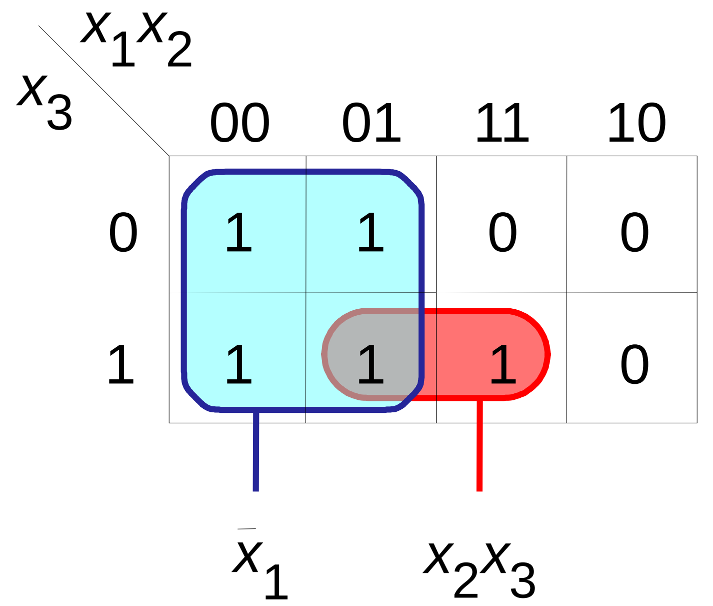
- **Mintermo**: produto que contém todas as variáveis (complementadas ou não)→ já visto
- **Literal**: cada ocorrência de uma variável de entrada em um produto, complementada ou não
  - um mintermo de uma função de n variáveis tem n literais
  - $x_1\bar{x}_3\bar{x}_5$ tem 3 literais e $x_1\bar{x}_3\bar{x}_5x_6\bar{x}_8\bar{x}_9$ tem 6 literais

Terminologia para f $(x_1 \ldots x_n)$ cont.

- **Implicante**: produto de qualquer número de variáveis para o qual f=1
  - o implicante mais básico é o mintermo, que tem n literais e é representado por 1 célula no M.K.
  - um implicante de (n-1) literais → 2 células vizinhas no M.K.
  - implicante de (n-2) literais → 4 células vizinhas
  - .... (n-k) literais → 2^k células vizinhas
  - implicante de 1 literal → $2^{(n-1)}$ = metade do M.K.
  - implicante de 0 literais (f=1 ou f=0) → M.K. inteiro

Implicantes em f = Σ m(0, 1, 2, 3, 7)

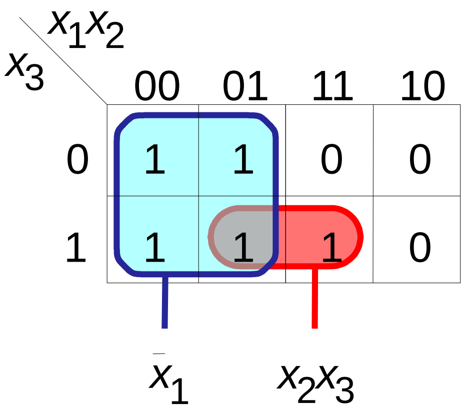
- Há 11 implicantes
  - De 1 célula → mintermos = 5
  - De 2 células → pares vizinhos de mintermos = 5
  - De 4 células → “quadrados” de mintermos = 1



Implicantes principais

- Implicante que
  - não pode ser combinado com outro implicante, ou
  - não está contido em outro implicante com menos literais, ou
  - não pode ter qualquer de seus literais removido e manter-se como implicante principal

- Na figura, temos dois implicantes principais: $\bar{x}_1$ e x_2x_3



IC-UNICAMP

Cobertura (*cover*)

- Cobertura = coleção de implicantes que contém todas as células iguais a um no mapa de Karnaugh
  - – a coleção de todos os implicantes principais é uma cobertura
- Algumas coberturas para o exemplo:
  - – todos os mintermos:

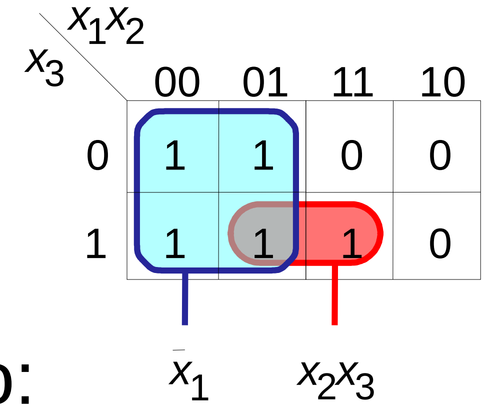
    $f = \Sigma\, m(0, 1, 2, 3, 7)$

  - – alguns implicantes:

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

  - – implicantes principais:

    $f = \overline{x}_1 + x_2 x_3$



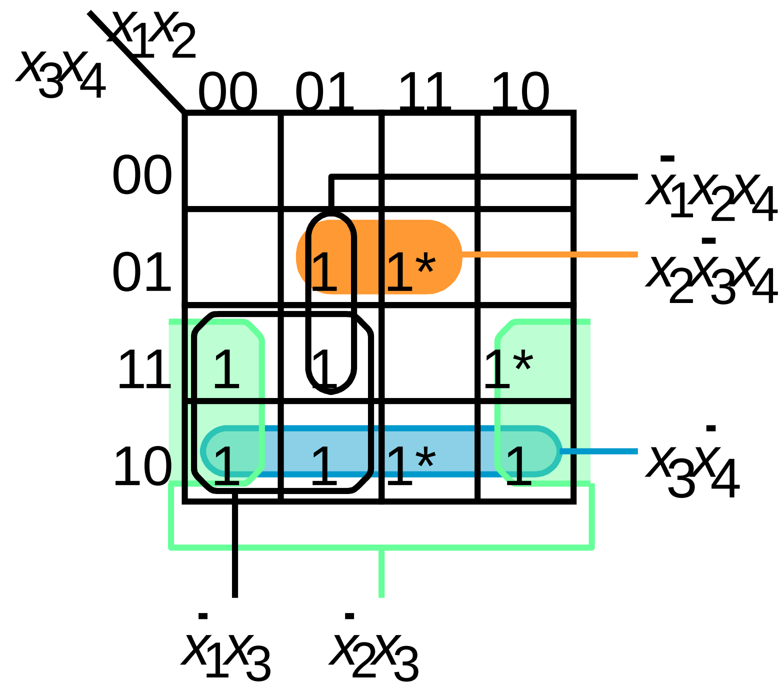
Custo da implementação

- No livro texto, custo = nº de gates + nº de entradas
- Mas, será assumido que os complementos das entradas primárias também estão disponíveis → não considerar inversores nas entradas
- custo para f= $x_1\overline{x}_2 + x_3\overline{x}_4$
  - 2 (AND)+1(OR)+4(entr. AND)+2(entr. OR)=9
- custo para f= $\overline{(x_1\overline{x}_2+x_3)}(\overline{x}_4+x_5)$
  - 2 (AND)+2(OR)+1(NOT)+9(entradas)=14

Implicante principal essencial

IC-UNICAMP

- Implicante principal é essencial se for o único a cobrir algum mintermo
- Exemplo: $f = \Sigma\, m(2, 3, 5, 6, 7, 10, 11, 13, 14)$
  - 5 implicantes principais
  - somente 3 são essenciais*
    - $\bar{x}_2 x_3$ devido a m11
    - $x_3 \bar{x}_4$ devido a m14
    - $x_2 \bar{x}_3 x_4$ devido a m13
  - faltou somente cobrir m7, e há 2 impl princ → escolher menor custo
  - $f = \bar{x}_2 x_3 + x_3 \bar{x}_4 + x_2 \bar{x}_3 x_4 + \bar{x}_1 x_3$



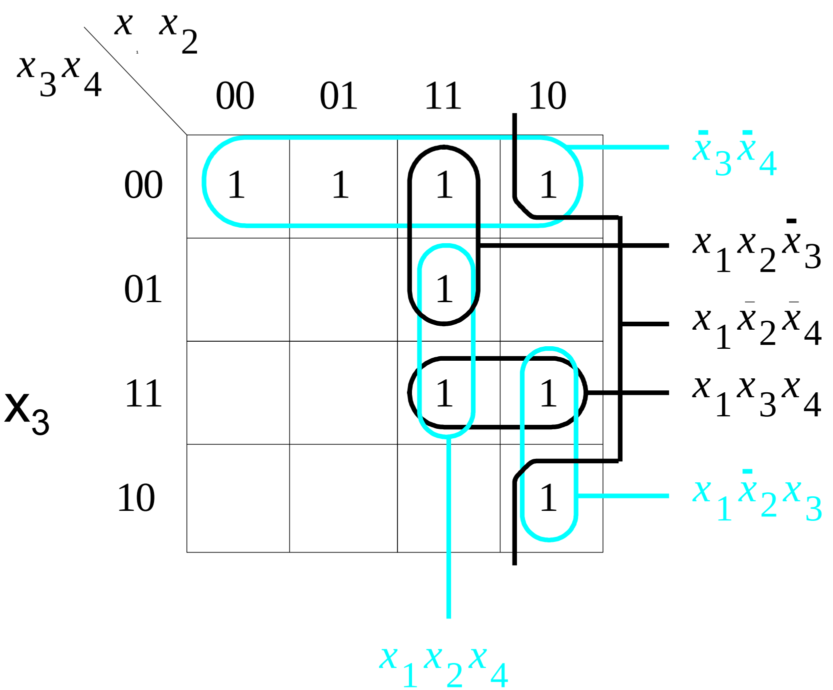
Procedimento de minimização

- Procedimento
  1. identificar todos os implicantes principais
  2. identificar quais são essenciais
  3. selecionar outros implicantes principais de modo a completar a cobertura, com custo mínimo

IC-UNICAMP

Exemplo de aplicação

- $f = \Sigma\, m(0, 4, 8, 10, 11, 12, 13, 15)$
- Há 6 impl. principais
- Só um essencial: $\bar{x}_3 \bar{x}_4$
- Considerar $x_1 x_2 \bar{x}_3$
  - alt 1: incluí-lo
    - para cobrir m10, m11, m15
    - usar $x_1 x_3 x_4$ e $x_1 \bar{x}_2 x_3$
    - $f = \bar{x}_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 + x_1 x_3 x_4 + x_1 \bar{x}_2 x_3$
  - alt 2: não incluí-lo
    - $x_1 x_2 x_4$ se torna essencial para cobrir m13
    - $f = \bar{x}_3 \bar{x}_4 + x_1 x_2 x_4 + x_1 \bar{x}_2 x_3$
    - MENOR CUSTO

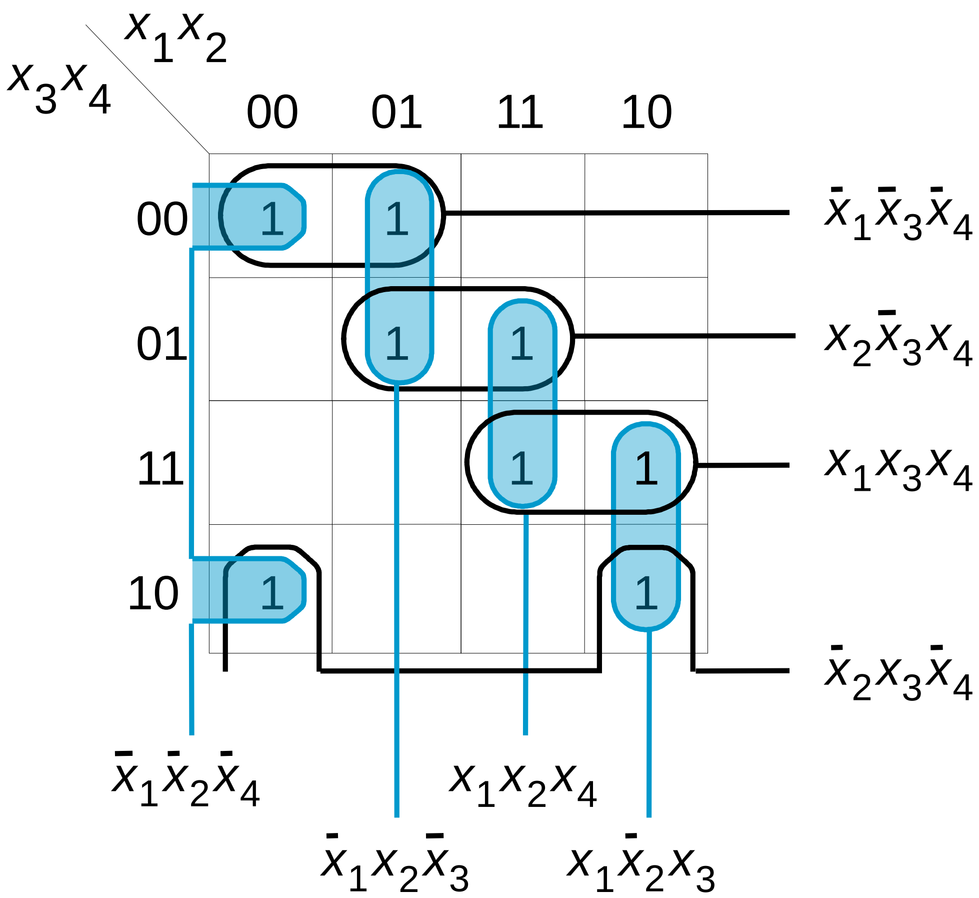


IC-UNICAMP

Exemplo sem implicantes essenciais

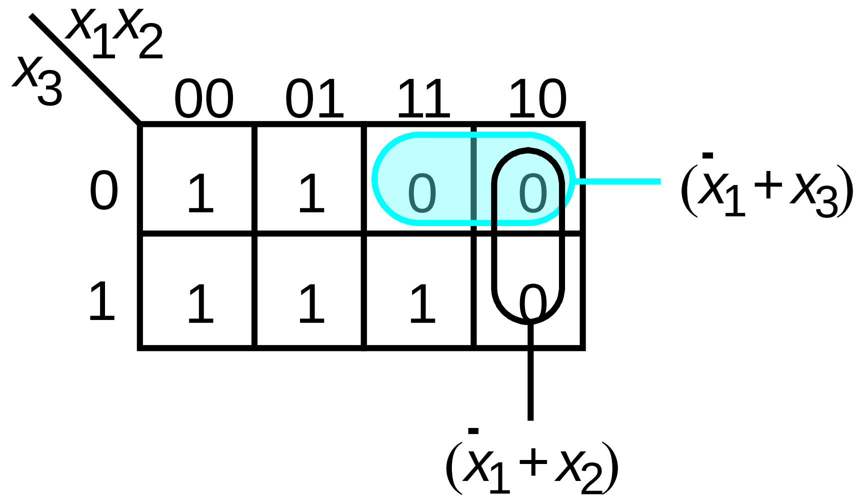
- $f = \Sigma\, m(0, 2, 4, 5, 10, 11, 13, 15)$

- Neste caso, escolher um implicante para considerar dentro ou fora
- No exemplo, duas soluções de mesmo custo
  - implicantes "horizontais"
  - implicantes "verticais"



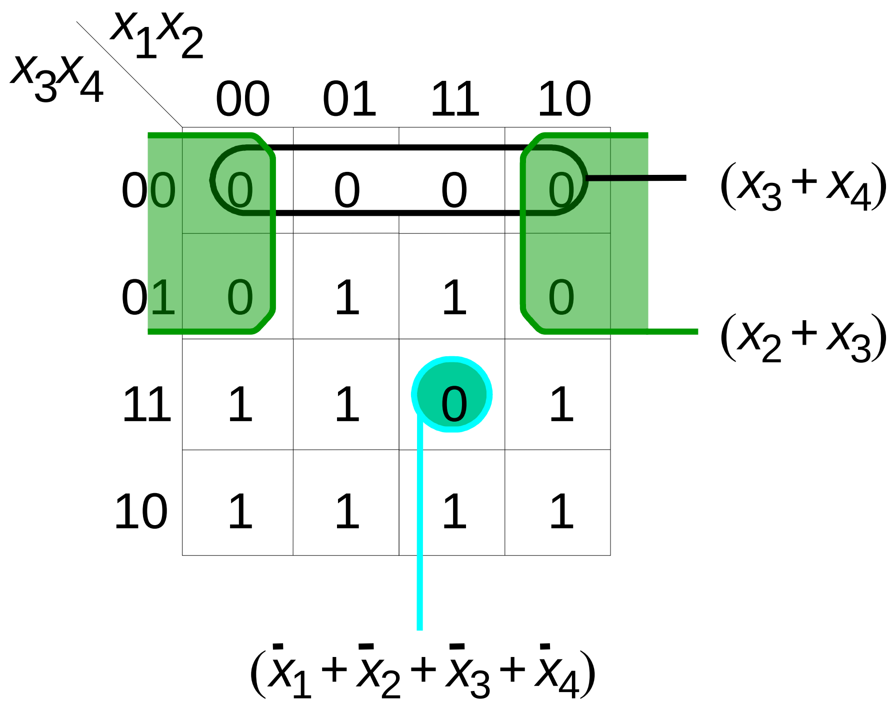
Minimização de POS

- Procedimento dual
- $f = \Pi\, M(4, 5, 6)$ (ver slide 17)
- $f = (\overline{x}_1 + x_2) \cdot (\overline{x}_1 + x_3)$
- custo ficou maior que o SOP



Outro exemplo POS

- $f = \Pi\, M(0, 1, 4, 8, 9, 12, 15)$
- $f=(x_2+x_3)(x_3+x_4)(\bar{x}_1+\bar{x}_2+\bar{x}_3+\bar{x}_4)$
- custo =15
- comparar com SOP slide 21
- custo SOP = 18

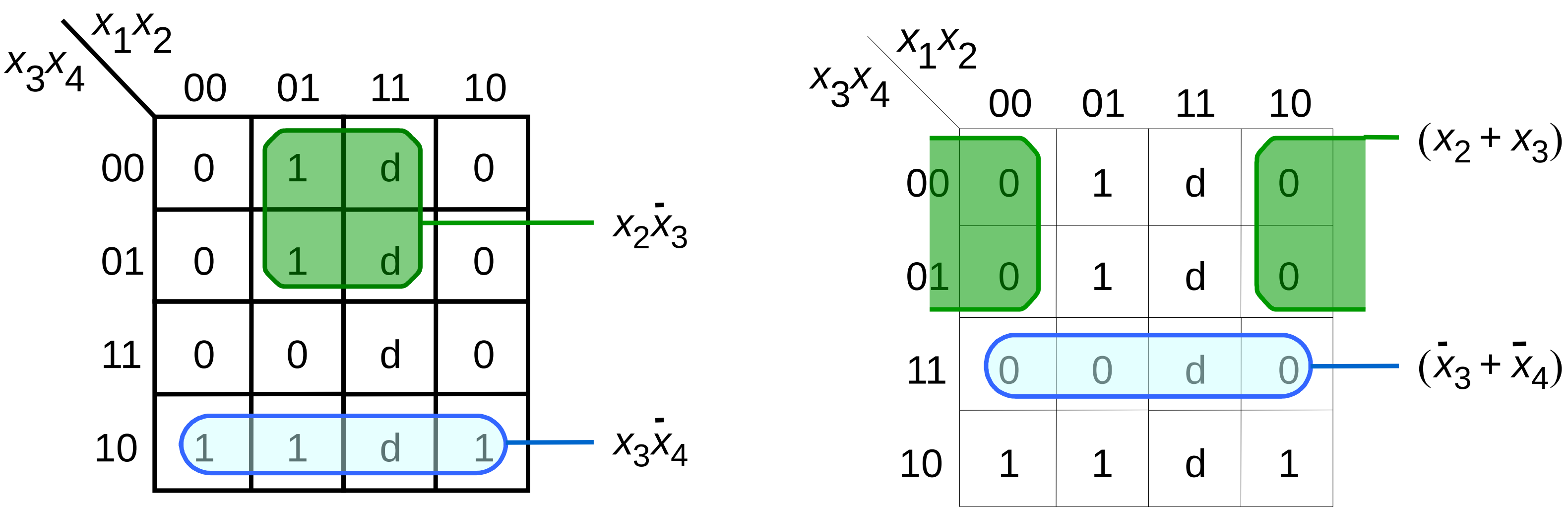


Funções especificadas incompletamente

- Às vezes certas combinações de entrada nunca acontecem
  - ex, se x_1 controla uma escolha do modo de operação em um sistema e x_2 outra escolha, então x_1 e x_2 =1 nunca podem acontecer
  - entradas possíveis: x_1x_2 = (00,01,10)
  - entrada impossível: x_1x_2=(11)
- As células do Mapa de Karnaugh impossíveis de acontecer são "don't care", representadas por "d"
- Exemplo de representação
  $f = \Sigma\, m(2, 4, 5, 6, 10) + D(12, 13, 14, 15)$

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

- Incluir “d” nos implicantes principais, quando oportuno para minimização

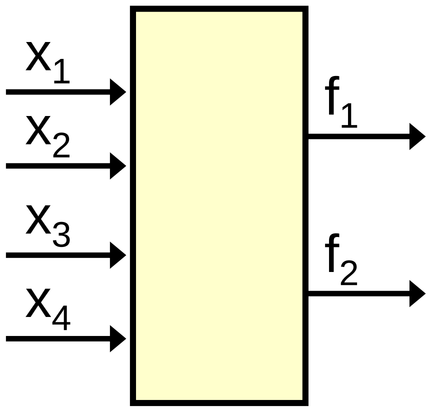


(a) SOP implementation

(b) POS implementation

IC-UNICAMP

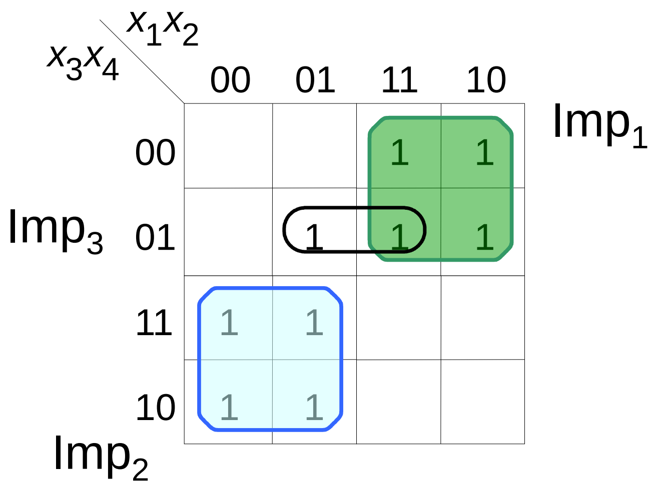
Circuitos com múltiplas saídas



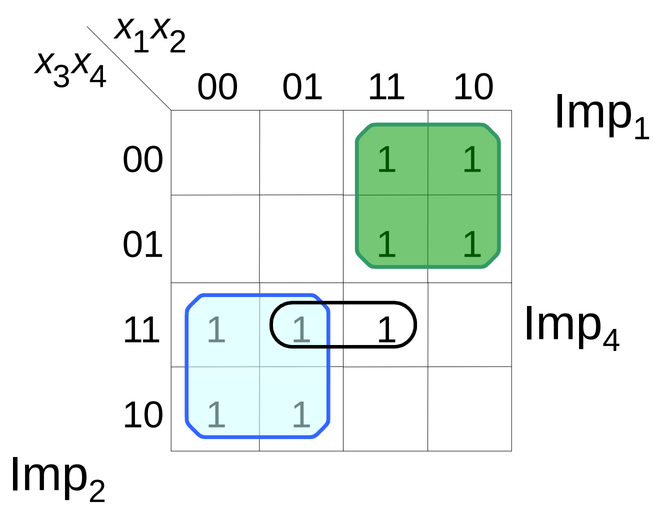
- Implementação convencional:
  - mínimo para f1 e mínimo para f2
  - implementados separadamente
- Implementação resultante pode não ser mínima
- Procedimento para obter mínimo
  - tentar compartilhar implicantes principais entre f_1 e f_2
- Compartilhamento pode ser natural ou forçado (exemplos a seguir)

IC-UNICAMP

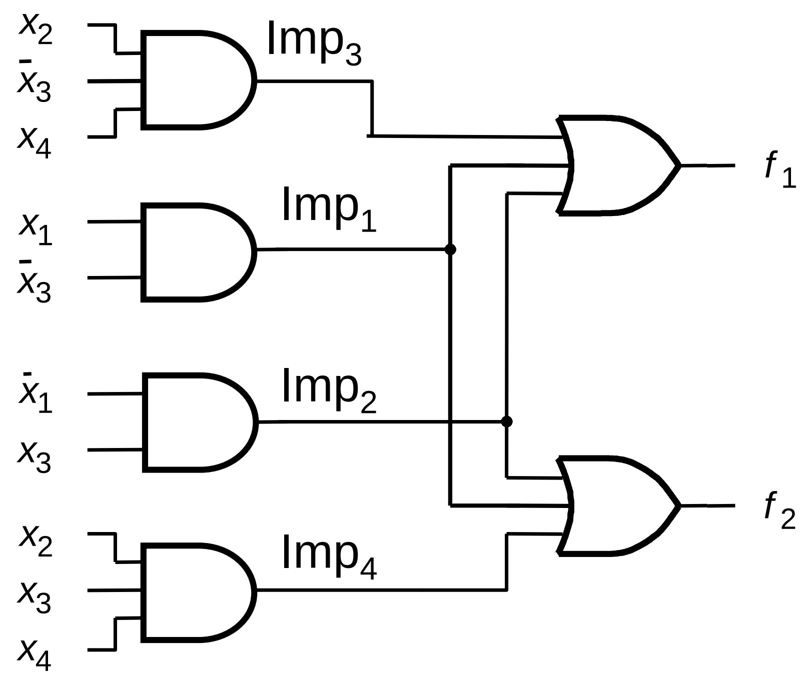
Ex1: circuito de 2 saídas



f_1



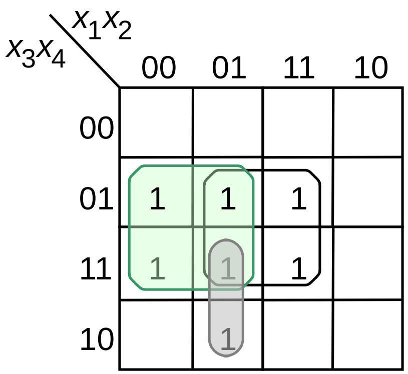
f_2



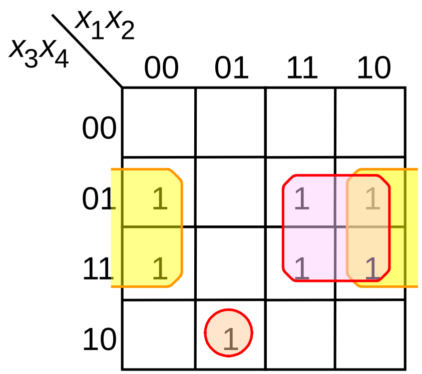
Circuito resultante para f_1 e f_2

IC-UNICAMP

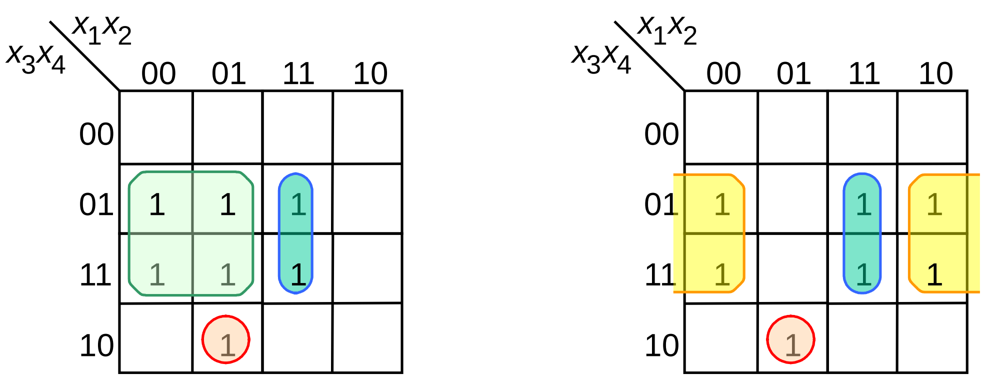
Ex1: circuito de 2 saídas



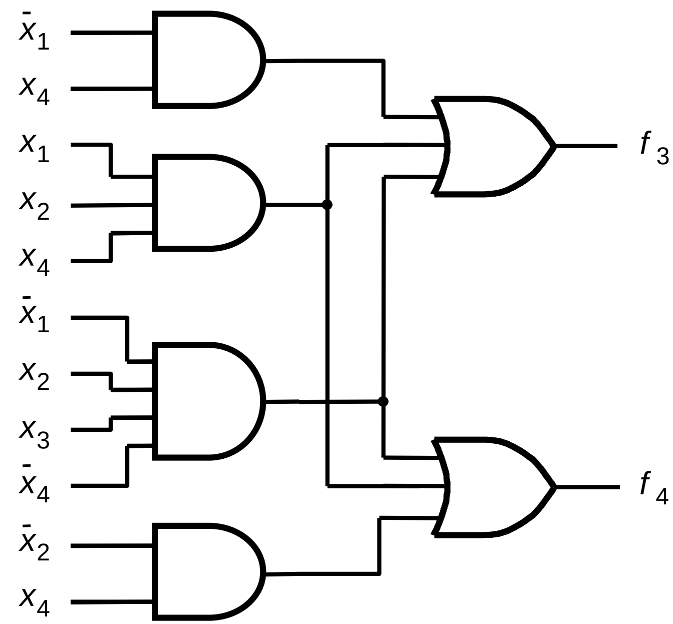
Minimização de f_3



Minimização de f_4



Minimização conjunta de f_3 e f_4



Minimização conjunta para f_3 e f_4

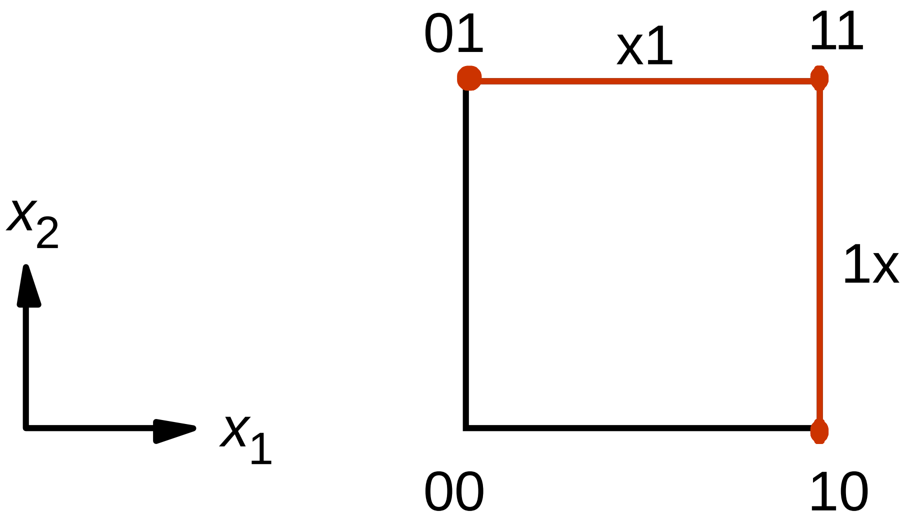
Síntese multi-nível

- Visto até agora: síntese visando custo mínimo → SOP ou POS de 2 níveis
- OK para circuitos de porte pequeno/médio
- Circuitos muito grandes podem ter problema de fan-in excessivo (portas com muitas entradas)
- Para sintetizar circuitos com outras restrições (ex fan-in), ferramentas de SW implementam circuitos multi-níveis
- Detalhes na seção 4.6 e 4.7 do livro texto

Representação cúbica

IC-UNICAMP

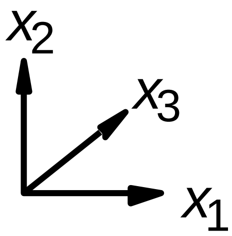
- Mapa de Karnaugh é ótimo para ilustrar conceitos.
  - automatização da síntese em SW precisa de outra ferramenta → exemplo repres. cúbica
- Exemplo: $f = \Sigma\, m(1, 2, 3) = x_1 + x_2$



x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	1

IC-UNICAMP

Exemplo de cubo 3D



- $f = \Sigma\, m(0, 2, 4, 5, 6)$
- algumas possibilidades de representação

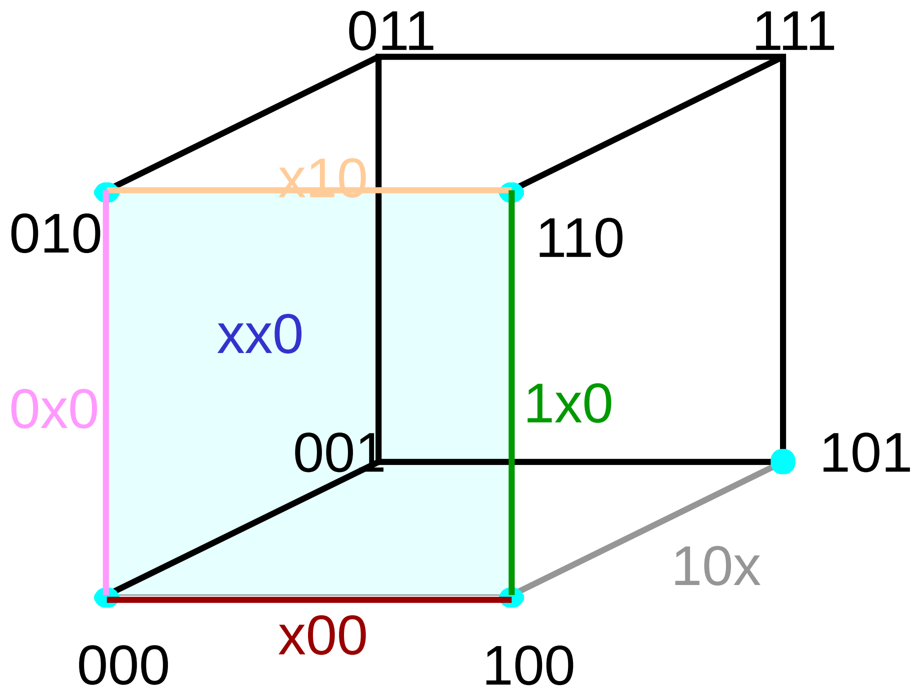
  (000,010,100,101,110)

  (0x0,1x0,101)

  (x00,x10,101)

  (x00,x10,10x)

  (xx0,10x)



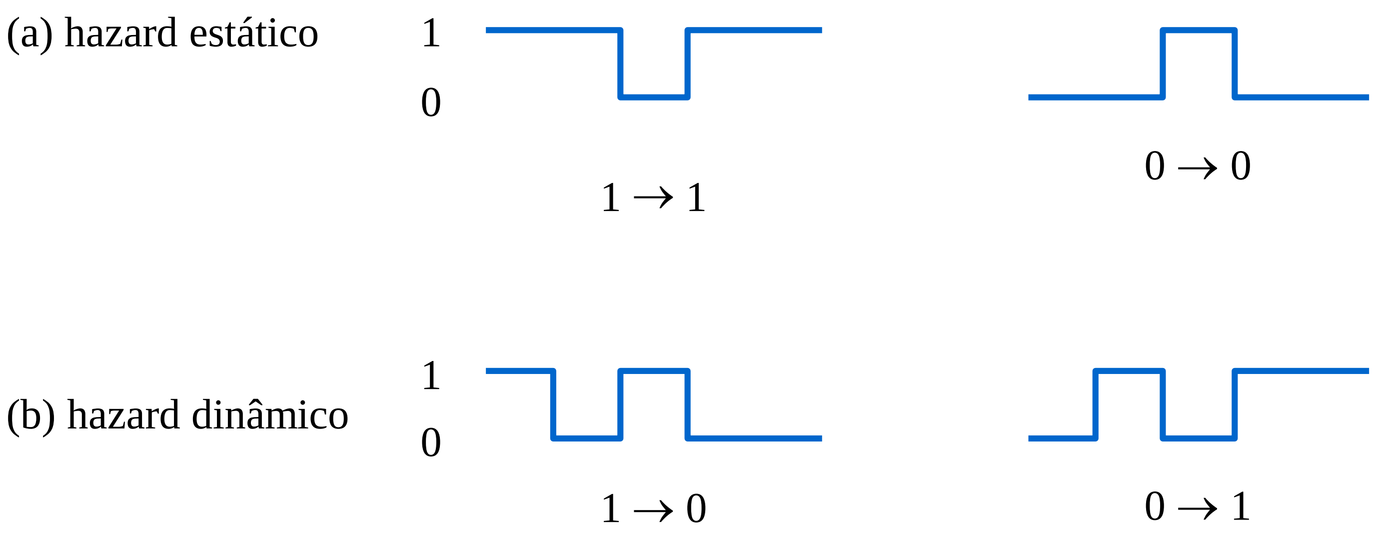
- (xx0,10x) → $\bar{x}_3 + x_1 \bar{x}_2$ é a ótima

Método tabular

- Algoritmo de Quine e McCluskey (1959)
- Início: tabela de mintermos
- A partir do mintermos (0-cubo) gera-se 1-cubos
- Mintermos cobertos são marcados
- A partir dos 1-cubos gera-se 2-cubos
- Termos cobertos são marcados
- Assim por diante
- Detalhes na seção 4.9

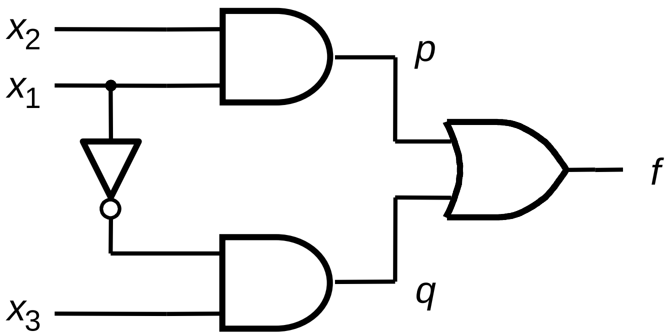
Hazards, spikes e glitches

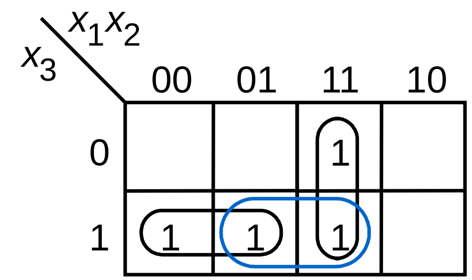
- Hazards e glitches: sinais temporários espúrios que ocorrem antes do sinal de estabilizar
- Podem causar problemas

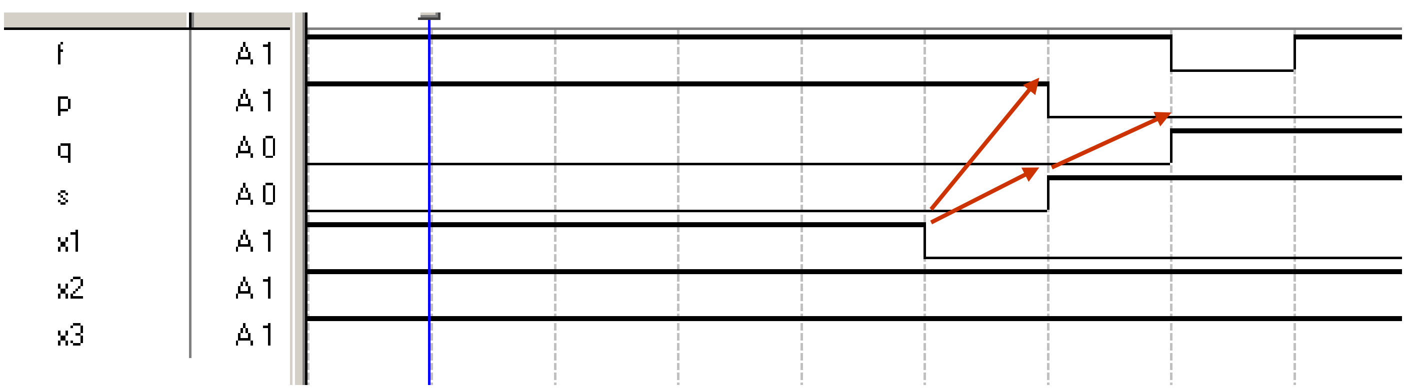


Exemplo de hazard estático

- Efeito de x1(1→0) faz p(1→0) antes que q (0→1)
- Existe momento temporário em que p+q=0

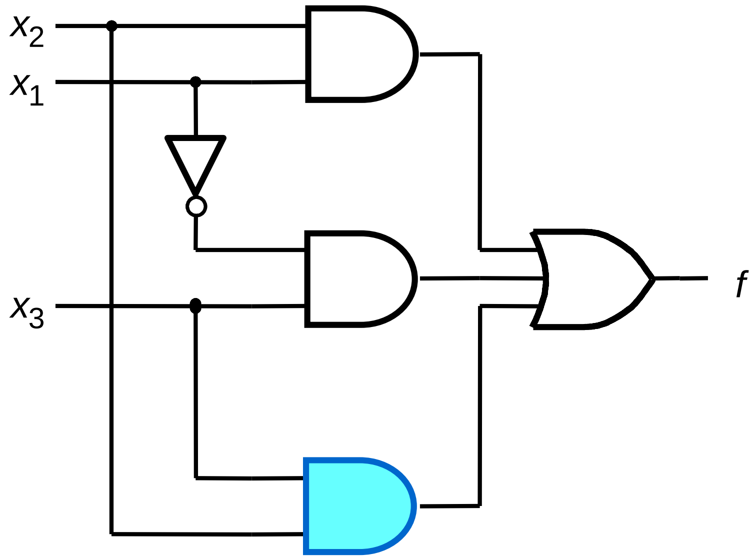


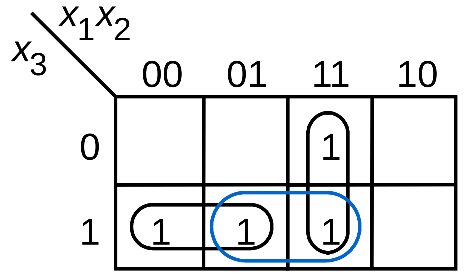




Solução do hazard estático

- Existe risco de hazard sempre que dois 1s adjacentes no mapa de Karnaugh não são cobertos por um único implicante
- Solução: criar novo implicante (redundante) para cobrir a transição

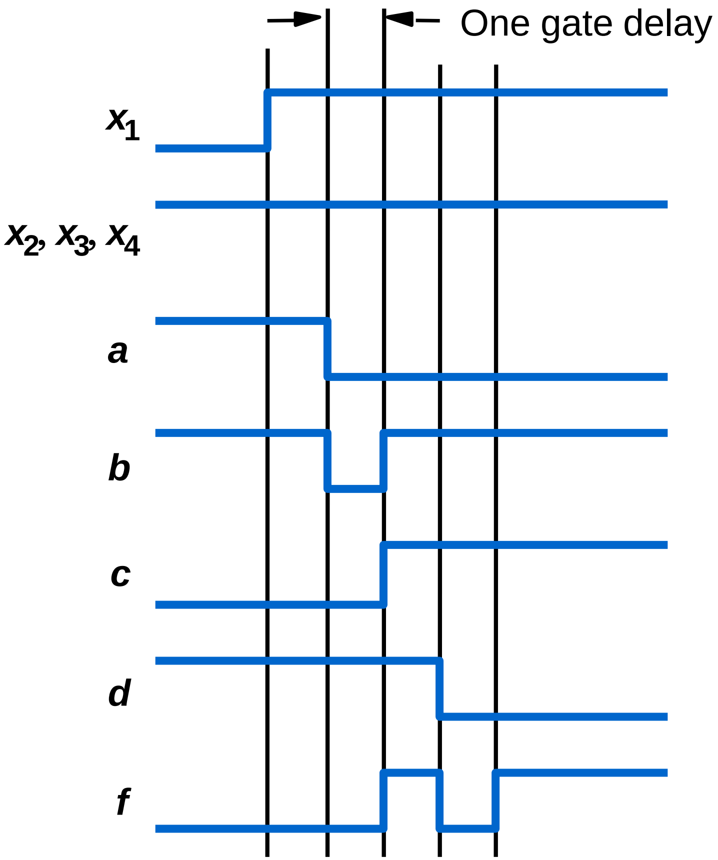


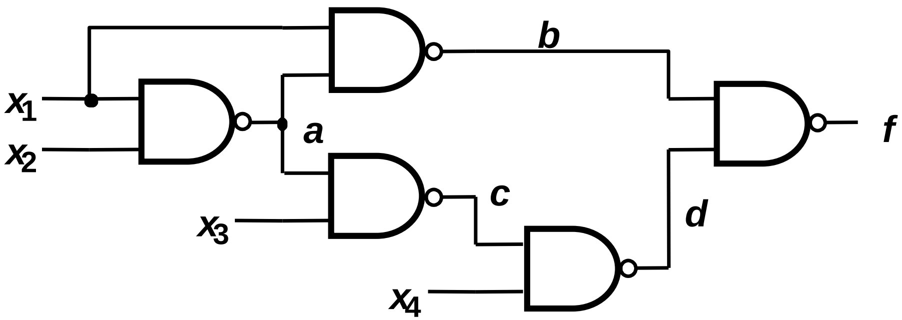


IC-UNICAMP

Exemplo de hazard dinâmico

- Quando 0→1→0 ou 1→0→1
- Aparece em circuitos > 2 níveis
- Observar hazard estático interno





Efeitos dos hazards

- Em circuitos combinacionais
  - apenas causa oscilações temporárias sem afetar o valor final da saída
  - poucos efeitos nocivos

- Em circuitos sequenciais (cap. 7)
  - pode levar a estado incorreto (→erro) se
    - hazard na entrada de dados dentro do tempo de setup e hold
    - ou no sinal de clock
    - ou em entradas de controle assíncronas: load, preset, clear