Lógica Digital (1001351)

Mapas de Karnaugh

Prof. Edilson Kato
kato@ufscar.br

Prof. Maurício Figueiredo
mauricio@ufscar.br

Prof. Ricardo Menotti
menotti@ufscar.br

Prof. Roberto Inoue
rsinoue@ufscar.br

Departamento de Computação
Universidade Federal de São Carlos

Atualizado em: 6 de março de 2019

Departamento de Computação

Mapas de Karnaugh: produto das somas

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

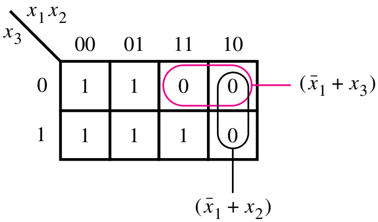


Figure 2.60 POS minimization of $f(x_1, x_2, x_3) = \Pi M(4, 5, 6)$.

Mapas de Karnaugh: produto das somas

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

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

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

$$f = \overline{x_1\overline{x}_2}.\overline{x_1\overline{x}_3}$$

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

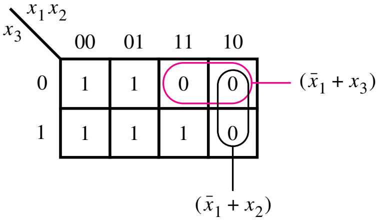
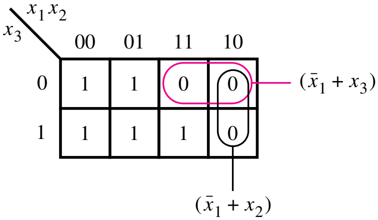


Figure 2.60 POS minimization of $f(x_1, x_2, x_3) = \Pi M(4, 5, 6)$.

Mapas de Karnaugh: produto das somas



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

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

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

$$f = \overline{x_1\overline{x}_2}.\overline{x_1\overline{x}_3}$$

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

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

Figure 2.60 POS minimization of $f(x_1, x_2, x_3) = \Pi M(4, 5, 6)$.

Mapas de Karnaugh: produto das somas

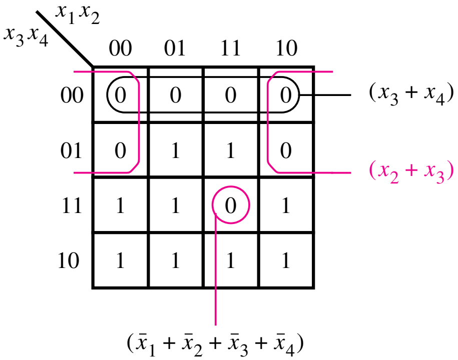
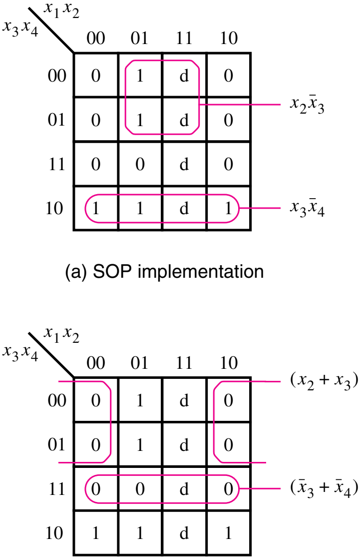


Figure 2.61 POS minimization of $f(x_1, \ldots, x_4) = \Pi M(0, 1, 4, 8, 9, 12, 15)$.

Especificação incompleta (don’t care)

- Nos circuitos digitais, há certas situações onde algumas entradas para uma função nunca acontecem. Ex:
  - Um sensor para detectar se uma porta está aberta e outro para detectar se a mesma porta está fechada;
  - Um sensor para detectar se um objeto é muito pesado e outro se ele é muito leve; etc.
- Em funções deste tipo, as entradas que nunca ocorrem são chamadas de ***indiferenças*** *(don’t care conditions)*;
  - Tanto faz qual será a saída da função nesses casos, já que a entrada nunca ocorre;
  - Isso pode ser usada para otimizar a função, adotando 0 ou 1 na saída de acordo com a conveniência.

Especificação incompleta (don't care)



(b) POS implementation

Figure 2.62 Two implementations of the function $f(x_1, \ldots, x_4) =$

Especificação incompleta (don't care)

BCD	b_3	b_2	b_1	b_0	f
0	0	0	0	0	0
1	0	0	0	1	0
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	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
A	1	0	1	0	–
b	1	0	1	1	–
C	1	1	0	0	–
d	1	1	0	1	–
E	1	1	1	0	–
F	1	1	1	1	–

x_3x_2 \ x_1x_0	00	01	11	10
00	0	0	1	0
01	0	0	0	1
11	D	D	D	D
10	0	1	D	D

Implementar $f(b_3, b_2, b_1, b_0) = \Sigma m_{(3,6,9)} + D_{(10,11,12,13,14,15)}$

Especificação incompleta (don’t care)

BCD	b_3	b_2	b_1	b_0	f
0	0	0	0	0	0
1	0	0	0	1	0
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	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
A	1	0	1	0	–
b	1	0	1	1	–
C	1	1	0	0	–
d	1	1	0	1	–
E	1	1	1	0	–
F	1	1	1	1	–

x_3x_2 \ x_1x_0	00	01	11	10
00	0	0	1	0
01	0	0	0	1
11	D	D	D	D
10	0	1	D	D

Implementar $f(b_3, b_2, b_1, b_0) = \Sigma m_{(3,6,9)} + D_{(10,11,12,13,14,15)}$

Especificação incompleta (don't care)

BCD	b_3	b_2	b_1	b_0	f
0	0	0	0	0	0
1	0	0	0	1	0
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	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
A	1	0	1	0	–
b	1	0	1	1	–
C	1	1	0	0	–
d	1	1	0	1	–
E	1	1	1	0	–
F	1	1	1	1	–

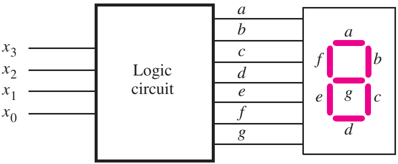
x_3x_2 \ x_1x_0	00	01	11	10
00	0	0	1	0
01	0	0	0	1
11	D	D	D	D
10	0	1	D	D

Implementar $f(b_3, b_2, b_1, b_0) = \Sigma m_{(3,6,9)} + D_{(10,11,12,13,14,15)}$

Circuitos com múltiplas saídas

- Frequentemente é necessário implementar funções que são parte de um sistema maior;
- Pode ser possível compartilhar algumas das portas necessárias na implementação de funções individuais;
- Essa estratégia nem sempre funciona da melhor maneira, como veremos a seguir;
- Em vez de derivar as expressões individualmente, podemos procurar implicantes que possam ser compartilhados com vantagem na realização combinada das funções.

Circuitos com múltiplas saídas

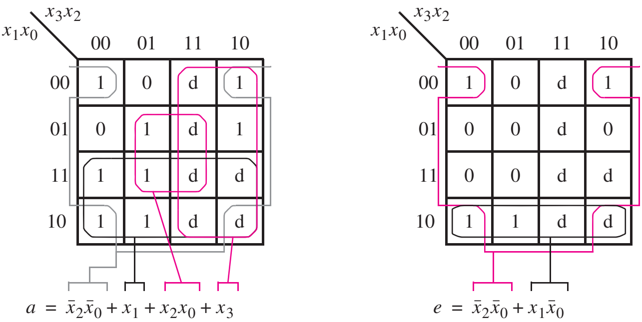


(a) Logic circuit and 7-segment display

	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

(b) Truth table

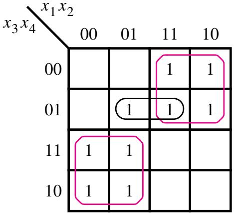
Circuitos com múltiplas saídas



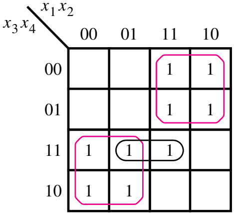
(c) The Karnaugh maps for outputs a and e.

Figure 2.63 Using don't-care minterms when displaying BCD numbers.

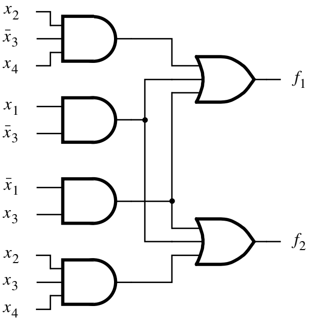
Circuitos com múltiplas saídas



(a) Function f_1

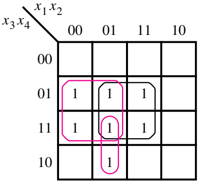


(b) Function f_2

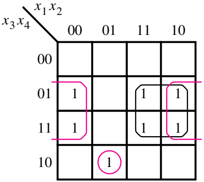


(c) Combined circuit for f_1 and f_2

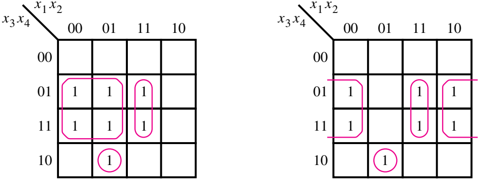
Circuitos com múltiplas saídas



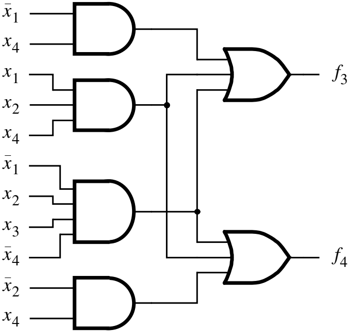
(a) Optimal realization of f_3



(b) Optimal realization of f_4



(c) Optimal realization of f_3 and f_4 together



(d) Combined circuit for f_3 and f_4

Bibliografia

- Brown, S. & Vranesic, Z. - Fundamentals of Digital Logic with Verilog Design, 3rd Ed., Mc Graw Hill, 2009

Lógica Digital (1001351)

Mapas de Karnaugh

Prof. Edilson Kato
kato@ufscar.br

Prof. Maurício Figueiredo
mauricio@ufscar.br

Prof. Ricardo Menotti
menotti@ufscar.br

Prof. Roberto Inoue
rsinoue@ufscar.br

Departamento de Computação
Universidade Federal de São Carlos

Atualizado em: 6 de março de 2019

Departamento de Computação