Sistemas Digitais

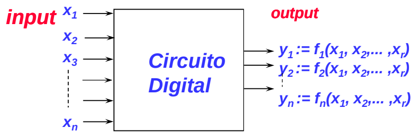
Mapa de Karnaugh

Aula 5

Prof. Abel Guilhermino

Mapa de Karnaugh

- O passo final no projeto de um rede combinacional envolve a implementação da rede em termos de elementos lógicos padrão, o qual pode ser feito em uma variedades de caminhos dependendo dos tipos de elementos a serem usados e restrições de projeto. Esta rede pode ser simplificada reduzindo sua lógica de implementação.
- Considere que desejamos implementar a lógica combinacional abaixo, adotando lógica de dois níveis.



Mapa de Karnaugh

- Funções Completamente especificadas e não completamente especificadas
- Funções completamente especificadas
  - Uma função é dita completamente especificada se $f_1(x_1, x_2,\ldots ,x_r)$ recebe um valor para todos os possíveis valores da r-tupla $[x_1, x_2,\ldots ,x_r]$.
  - Exemplo:
    - Representação dos números na base 8 (0 a 7). Neste caso todos os valores de entrada da 3-tuplas $[x_1, x_2, x_3]$ que representam um número octal seriam especificadas.

x_1	x_2	x_3	$f(x_1, x_2,\ldots ,x_r)$
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Mapa de Karnaugh

- Funções não completamente especificadas
  - Uma função é dita não completamente especificada se existem valores de $[x_1, x_2,\ldots ,x_r]$ que nunca ocorrerão devido ao caminho no qual as variáveis são geradas, resultando numa função $f_1(x_1, x_2,\ldots ,x_r)$ que não precisa ser definida para estes valores.
  - **Esta situação é chamada condição "don't care" e a função é dita ser não completamente especificada.**
  - Exemplo:
    - Representação dos números decimais no código BCD. Neste caso apenas 10 valores de entrada da 4-tuplas $[x_1,x_2,x_3, x_4]$ seriam especificados e os outros 6 corresponderiam a condições don't care.
    - Don't care é geralmente representado por d, x, ou -

x_1	x_2	x_3	x_4	$f_1(x_1, x_2,\ldots ,x_r)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	d
1	0	1	1	d
1	1	0	0	d
1	1	0	1	d
1	1	1	0	d

Mapa de Karnaugh

- Exemplo de minimização com don't cares

Considere a função abaixo:

x_1	x_2	x_3	$f_1(x_1, x_2,\ldots ,x_r)$
0	0	0	1
0	0	1	1
0	1	0	d
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	d
1	1	1	0

Observando a figura ao lado vemos que as 3-tuplas [0,1,0] e [1,1,0] nunca ocorrerão como combinações de entrada. Assim podemos assumir qualquer valor que quisermos para g(0,1,0) e g(1,1,0).

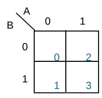
1. $g(0,1,0) = g(1,1,0) = 0$
   $g_1(x_1, x_2, x_3) = \bar{x}_1.\bar{x}_2.\bar{x}_3 + \bar{x}_1.\bar{x}_2.x_3 + \bar{x}_1.x_2.x_3$
   $= \bar{x}_1.\bar{x}_2 + \bar{x}_1.x_3$
2. $g(0,1,0) = 1$ e $g(1,1,0) = 0$
   $g_2(x_1, x_2, x_3) = \bar{x}_1.\bar{x}_2.\bar{x}_3 + \bar{x}_1.\bar{x}_2.x_3 + \bar{x}_1.x_2.\bar{x}_3 + \bar{x}_1.x_2.x_3$
   $= \bar{x}_1$
3. $g(0,1,0) = 0$ e $g(1,1,0) = 1$
   $g_3(x_1, x_2, x_3) = \bar{x}_1.\bar{x}_2.\bar{x}_3 + \bar{x}_1.\bar{x}_2.x_3 + \bar{x}_1.x_2.x_3 + x_1.x_2.\bar{x}_3$
   $= \bar{x}_1.\bar{x}_2 + \bar{x}_1.x_3 + x_1.x_2.\bar{x}_3$
4. $g(0,1,0) = 1$ e $g(1,1,0) = 1$
   $g_4(x_1, x_2, x_3) = \bar{x}_1.\bar{x}_2.\bar{x}_3 + \bar{x}_1.\bar{x}_2.x_3 + \bar{x}_1.x_2.\bar{x}_3 + \bar{x}_1.x_2.x_3 + x_1.x_2.\bar{x}_3$
   $= \bar{x}_1 + x_2.\bar{x}_3$

Analisando as quatro possíveis soluções mínimas concluímo que a melhor a ser implementada é a função $g_2 = \bar{x}_1$ para $g(0,1,0)=1$ e $g(1,1,0) = 0$

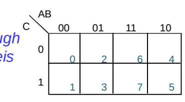
Mapa de Karnaugh

- Mapa de Karnaugh é um método alternativo para se representar uma tabela verdade. Ele propõe uma solução gráfica da informação contida na tabela verdade.

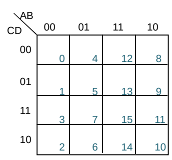
☑ Mapa de Karnaugh para duas variáveis



☑ Mapa de Karnaugh para três variáveis



☑ Mapa de Karnaugh para quatro variáveis



Mapa de Karnaugh

- *Um mapa de Karnaugh para uma função f(a, b, c, d, ...) de r variáveis contem 2^r quadrados (células) existindo sempre uma célula para cada valor possível da r-tupla [x_1, x_2,....., x_r].*
- *O valor "1" em uma das possíveis posições significa a combinação para a qual a saída de "1" é desejada (função igual a "1"). O valor "0" representa uma combinação na qual uma saía de "0" é desejável (função igual a "0") e "d" ou "x" naquelas células que correspondem a condição de entrada "don't care".*

☑ *Mapa de Karnaugh para duas variáveis*

B \ A	0	1
0	f(0,0) 0	f(1,0) 2
1	f(0,1) 1	f(1,1) 3

☑ *Mapa de Karnaugh para três variáveis*

C \ AB	00	01	11	10
0	f(0,0,0) 0	f(0,1,0) 2	f(1,1,0) 6	f(1,0,0) 4
1	f(0,0,1) 1	f(0,1,1) 3	f(1,1,1) 7	f(1,0,1) 5

Mapa de Karnaugh

- *As duas leis algébricas básicas que nós fazemos uso no mapa de Karnaugh para redução de expressões lógicas são:*

$XY+\overline{X}Y=Y$

y \ x	0	1
0	0	0
1	1	1

$X+\overline{X}Y=X+Y$

y \ x	0	1
0	0	1
1	1	1

- *Devemos construir o mapa tal que fique fácil analisar a expressão por inspeção.*
- *Note que qualquer dois quadrados adjacentes no mapa correspondem a r-tuplas as quais diferem em apenas um literal.*
- *A r-tupla correspondente ao quadrado mais a esquerda de qualquer fila difere apenas de um literal da r-tupla mais a direita da mesma fila.(**contornos na linha**)*
- *Da mesma forma, o topo de qualquer coluna difere apenas de um literal do quadrado da base da mesma coluna. (**contornos na coluna**)*

Mapa de Karnaugh

- *Adjacências no Mapa de Karnaugh*

C \ AB	00	01	11	10
0	000	010	110	100
1	001	011	111	101

Mapa de Karnaugh

- *Exemplos de simplificação de funções usando Mapa de Karnaugh*

B \ A	0	1
0	0	1
1	0	1

F = A

- **A constante**
- **B varia**

→ Elimina-se B

B \ A	0	1
0	1	1
1	0	0

G = B'

- **B complementado constante**
- **A varia**

→ Elimina-se A

Mapa de Karnaugh

- *Exemplos de simplificação de funções usando Mapa de Karnaugh*

Cin \ A B	00	01	11	10
0	0	0	1	0
1	0	1	1	1

f(Cin,A,B) = A B + B Cin + A Cin

C \ AB	00	01	11	10
0	0	0	1	1
1	0	0	1	1

F(A,B,C) = A

Mapa de Karnaugh

- *Procedimentos*
  - *Toda célula deve ser contada pelo menos uma vez.*
  - *Qualquer combinação deveria ser a maior possível. Assim, uma célula não deveria ser considerada isolada se ela pode fazer parte de um grupo de duas ou mais células adjacentes.*
  - *Todas as células deveriam ser referenciadas em um menor número de grupos possíveis.*
- *Características de mapeamento*
  - *Cada vez que nós combinamos dois minitermos eliminamos uma das variáveis no termo produto. A variável que é eliminada é uma que aparece na forma negada em um minitermo e na forma não negada no outro minitermo.*
  - *Quando 2^r minitermos são combinados, nós eliminamos r variáveis*
  - *As filas e colunas de um mapa K são marcados de forma que apenas uma variável muda quando caminhamos de fila-em-fila ou de coluna-em-coluna.*

Mapa de Karnaugh

- *Processo de redução*
  - *Identifique e marque todas as células individuais que não podem ser combinadas com quaisquer outras células.*
  - *Identifique todas as células que podem ser combinadas com apenas uma outra célula. Use estes pares para formar grupos duplos.*
  - *Identifique todas as células que podem ser combinadas em grupo de 4 células contanto que todas as células não estejam já cobertas por outros grupos (preferencialmente).*
  - *Repita o processo de combinação para grupos de 8 células contanto que todas as células no grupo não estejam cobertas (preferencialmente).*
  - *Investigue qualquer célula ainda não contida em um grupo. Arbitrariamente forme o maior grupo possível que pode ser formado e que inclui a maioria das células não cobertas.*

Mapa de Karnaugh

- *Mapa de Karnaugh para 3 variáveis*

C \ AB	00	01	11	10
0	1	0	0	1
1	0	0	1	1

F(A,B,C) = Σm(0,4,5,7)

F = B' C' + A C

- **Células adjacentes nas extremidades da direita e esquerda da fila.**
  → Elimina-se uma variável, a variável A
- **Células adjacentes na mesma fila**
  → Elimina-se a variável B.

C \ AB	00	01	11	10
0	0	1	1	0
1	1	1	0	0

F(A,B,C) = Σm(1,2,3,6)

F' = B C' + A' C

Mapa de Karnaugh

- *Mapa de Karnaugh para 4-variáveis*

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	1	1
10	1	1	1	1

F(A,B,C,D) = Σm(0,2,3,5,6,7,8,10,11,14,15)

F = C + A' B D + B' D'

Encontrar o menor número de grandes subgrupos de células adjacentes, que cobrem "1"s.

Mapa de Karnaugh

- *Don't Care*

CD \ AB	00	01	11	10
00	0	0	X	0
01	1	1	X	1
11	1	1	0	0
10	0	X	0	0

CD \ AB	00	01	11	10
00	0	0	X	0
01	1	1	X	1
11	1	1	0	0
10	0	X	0	0

F(A,B,C,D) = Σm(1,3,5,7,9) + Σd(6,12,13)

- **O don't care pode ser "1" ou "0".**
- **O projetista deve escolher aquele valor que permita máxima redução de sua lógica.**

Assim:
a) Usando Minitermos (Se x=1)
F = A'D + C' D' = D(A' + C')

b) Usando maxitermos (Se x=0)
F= (C+D)(A' + C')(C'+D) =
(C C'+CD+ C'D+D)(A' + C')=
D(A' + C')
ou direto, agrupando oito vizinhos:
F= D(A' + C')

Mapa de Karnaugh - Exemplos

W

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

W=B'D'+BD

X

CD \ AB	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	X

X=BD'+B'D

Y

CD \ AB	00	01	11	10
00	0	1	0	0
01	1	1	1	1
11	0	1	0	0
10	0	1	0	0

Y=A'B+C'D

Z

CD \ AB	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

Z=D'

Mapa de Karnaugh - Exercícios

☑ *Encontrar as funções de cada Mapa-K*

W

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

X

			A	
CD \ AB	00	01	11	10
00	0	1	1	0
01	1	1	0	1
C 11	1	0	0	1
C 10	0	1	1	X

Y

CD \ AB	00	01	11	10
00	0	1	0	0
01	1	1	1	1
11	0	1	0	0
10	0	1	0	0

Z

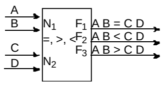
CD \ AB	00	01	11	10
00	1	1	X	1
01	0	0	1	0
11	0	0	1	1
10	1	1	1	1

Projeto de um comparador de dois bits

simplificação em dois níveis

- *Implementar um comparador de dois vetores de dois bits*

Diagrama de bloco



Um mapa K de 4- variáveis e 3 funcões saídas

Tabela verdade

A	B	C	D	F_1	F_2	F_3
0	0	0	0	1	0	0
		0	1	0	1	0
		1	0	0	1	0
		1	1	0	1	0
0	1	0	0	0	0	1
		0	1	1	0	0
		1	0	0	1	0
		1	1	0	1	0
1	0	0	0	0	0	1
		0	1	0	0	1
		1	0	1	0	0
		1	1	0	1	0
1	1	0	0	0	0	1
		0	1	0	0	1
		1	0	0	0	1
		1	1	1	0	0

Projeto de um comparador de dois bits

simplificação em dois níveis

- *Implementação do comparador*

CD \ AB	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

K-map for F_1

CD \ AB	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

K-map for F_2

CD \ AB	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

K-map for F_3

F1 = A' B' C' D' + A' B C' D + A B C D + A B' C D' = (A xnor C) (B xnor D)

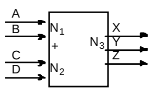
F2 = A' B' D + A' C + B'CD

F3 = B C' D' + A C' + A B D'

Projeto de um somador de dois bits

- *Implementação do somador*

Diagrama de bloco



Problema com 4 variáveis de entrada e 3 saídas

Tabela verdade

A	B	C	D	X	Y	Z
0	0	0	0	0	0	0
		0	1	0	0	1
		1	0	0	1	0
		1	1	0	1	1
0	1	0	0	0	0	1
		0	1	0	1	0
		1	0	0	1	1
		1	1	1	0	0
1	0	0	0	0	1	0
		0	1	0	1	1
		1	0	1	0	0
		1	1	1	0	1
1	1	0	0	0	1	1
		0	1	1	0	0
		1	0	1	0	1
		1	1	1	1	0

Projeto de um somador de dois bits

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

K-map for X

CD \ AB	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	1	0	1	0
10	1	1	0	0

K-map for Y

CD \ AB	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

K-map for Z

X = A C + B C D + A B D

Z = B D' + B' D = B xor D

Y = A' B' C + A B' C' + A' B C' D + A' B C D' + A B C' D' + A B C D

= B' (A xor C) + A' B (C xor D) + A B (C xnor D)

= B' (A xor C) + B (A xor C xor D)