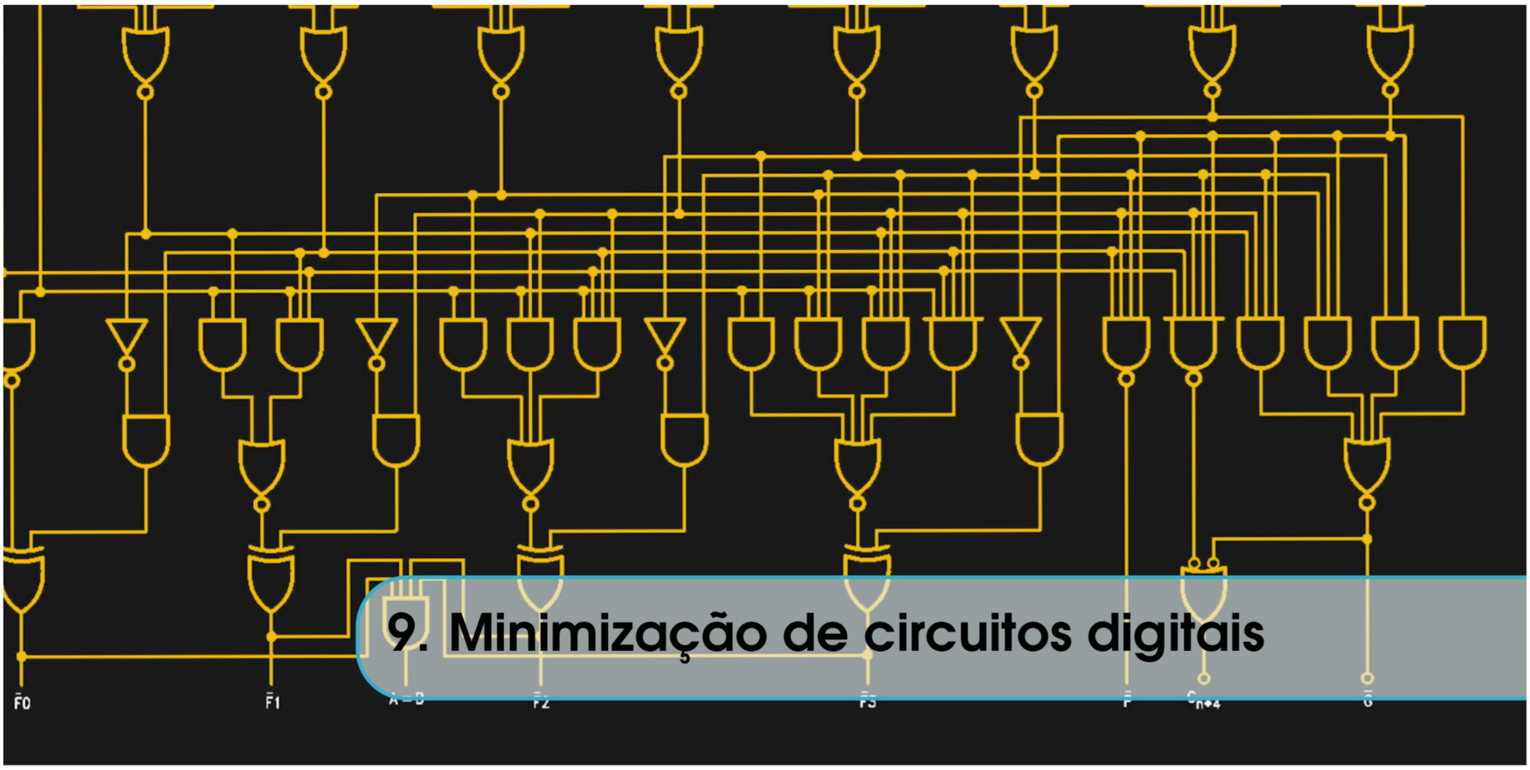


9. Minimização de circuitos digitais

Neste capítulo são mostrados os fundamentos da implementação de circuitos digitais mínimos de dois níveis. Para isso, é discutida a teoria envolvida na obtenção de expressões mínimas de dois níveis bem como sua aplicação pelos métodos de inspeção usando mapas-K, e tabular Quine-McCluskey.

9.1 Obtenção de expressões mínimas na forma SDP

Na Seção 8.2 são discutidas as premissas consideradas para a obtenção de funções de chaveamento minímas que representam uma certa tabela verdade/sistema. Usando as premissas Seção 8.2 são apresentados na presente seção a teoria para obtenção de expressões mínimas na forma SPD bem como os métodos de mapa-K e tabular para obtenção de tais expressões na forma SDP.

9.1.1 Teoria para a obtenção de SDP mínimas

Para estabelecer um critério de minimização é necessário se definir alguns conceitos. O primeiro deles é o conceito de implicante de uma função mostrado na Definição 9.1.1.

Definição 9.1.1 (Implicante de uma função) Um termo produto p é definido como um implicante da função f, se para toda atribuição de variáveis que resulta em p igual a 1 também resulta em f igual a 1.

O Exemplo 9.1 mostra um exemplo de um implicante de uma função.

■ **Exemplo 9.1** (Implicante de uma função)

Como exemplo para implicantes, considere o termo produto p, $p(x_2,x_1,x_0) = x_1{\cdot}x_0$ que é um implicante da função $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,3,6,7)$. Isso porque todas as atribuições para as variáveis $x_2x_1x_0$ que tornam $p = 1$ também tornam $f(x_2,x_1,x_0) = 1$. As atribuições para as variáveis $x_2x_1x_0$ que tornam $p = 1$ são $x_2x_1x_0 = 011$ e $x_2x_1x_0 = 111$. Essa duas atribuições correspondem, em decimal, aos índices 3 e 7 e ambos estão incluídos no Conjunto-UM() de $f(x_2,x_1,x_0)$.
■

Observe que todas as funções de chaveamento podem ser escrita como soma de implicantes. Um exemplo disso é a soma de mintermos vista no Capítulo **??**. Perceba que cada mintermo incluído em uma função específica constitui em um implicante da função.

O Exemplo 9.2 mostra a obtenção de todos os implicantes de um função.

■ **Exemplo 9.2** (Lista de implicantes de uma função)

Considere a função $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(5,7,13,15)$. Essa função possui 9 implicantes p_1 a p_9 , todos destacados na Fig. 9.1. São os quatro mintermos $p_1 = \overline{x_3}{\cdot}x_2{\cdot}\overline{x_1}{\cdot}x_0$, $p_2 = \overline{x_3}{\cdot}x_2{\cdot}x_1{\cdot}x_0$, $p_3 = x_3{\cdot}x_2{\cdot}\overline{x_1}{\cdot}x_0$ e $p_4 = x_3{\cdot}x_2{\cdot}x_1{\cdot}x_0$ (Fig. 9.1a), os termos $p_5 = \overline{x_3}{\cdot}x_2{\cdot}x_0$ e $p_6 = x_3{\cdot}x_2{\cdot}x_0$ (duplas horizontais na Fig. 9.1b), os termos $p_7 = x_2{\cdot}\overline{x_1}{\cdot}x_0$ e $p_8 = x_2{\cdot}x_1{\cdot}x_0$ (duplas verticais na Fig. 9.1c) e o termo $p_9 = x_2{\cdot}x_0$ (quarteto na Fig. 9.1d)

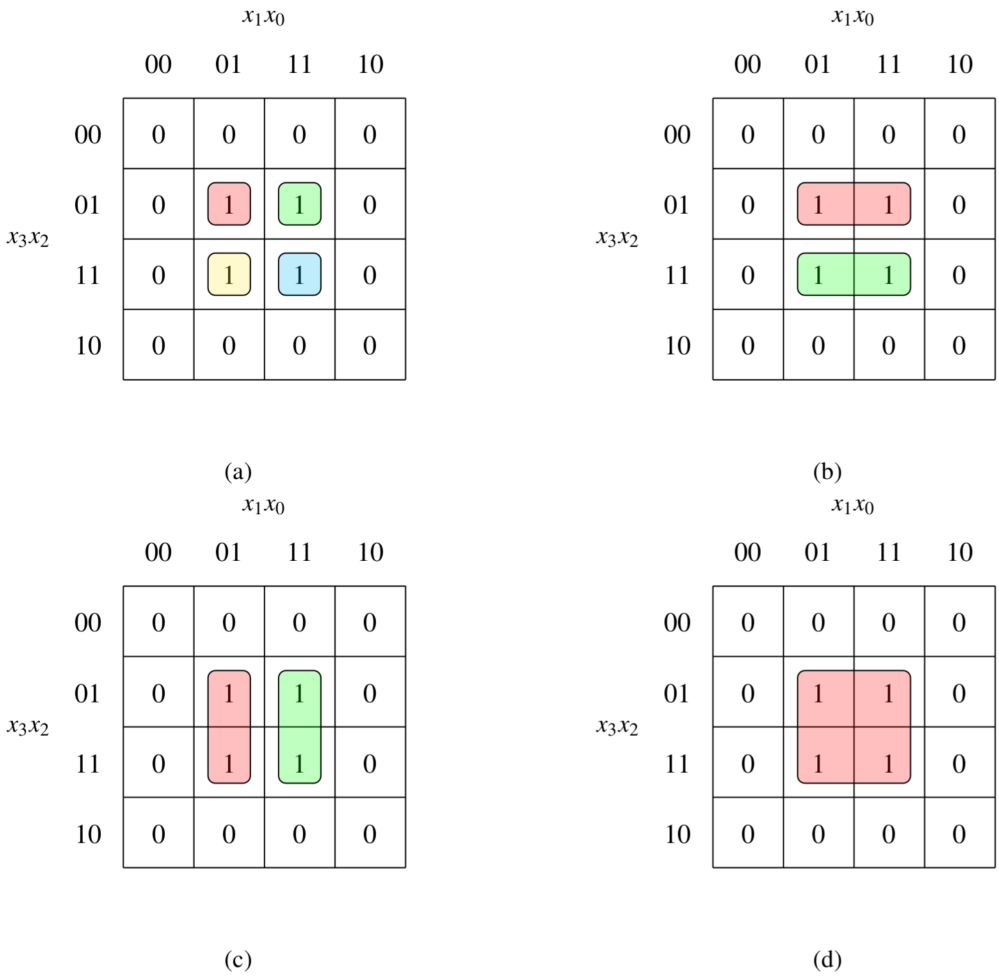


Figura 9.1: Todos os implicantes da $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(5,7,13,15)$ mostrados por agrupamentos de 1s no mapa-K.

■

É possível notar no Exemplo 9.2 e na Fig. 9.1 que existem implicantes "maiores" (*i.e.* com mais variáveis) e implicantes "menores" (*i.e.* com menos variáveis). Uma relação objetiva entre implicantes é estabelecida na Definição 9.1.2.

Definição 9.1.2 (Relação de cobertura entre implicantes) Considere dois implicantes p_1 e p_2 de f tais que $p_1 \neq p_2$. Se TODAS as atribuições de variáveis que tornam $p_1 = 1$ também resultarem em $p_2 = 1$ diz-se que p_2 cobre p_1. Aqui a notação usada é: $p_2 \succ p_1$ (p_2 cobre p_1) ou $p_1 \prec p_2$ (p_1 é coberto por p_2) .

Começa a ficar claro que, na escrita de expressões mínimas, é preferível se optar por implicantes que cobrem à implicantes que são cobertos. A título de exemplo da relação de cobertura, no Exemplo 9.2 são elencados todos os 9 implicantes da função $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(5,7,13,15)$ e relacionados como p_1 a p_9. Ao usar a Definição 9.1.2 à $f(x_3,x_2,x_1,x_0)$ conclui-se facilmente que:

- p_9 cobre p_1 a p_8,
- p_8 cobre p_2 e p_4,
- p_7 cobre p_1 e p_3,
- p_6 cobre p_3 e p_4,
- p_5 cobre p_1 e p_2.

Observe pelo exemplo que, necessariamente, se $p_2 \succ p_1$ isso significa que o termo produto p_2 possui menos variáveis que o termo produto p_1. Ainda observando o Exemplo 9.2 fica claro que escrever a função $f(x_3,x_2,x_1,x_0)$ usando o implicante p_9 gera uma expressão algébrica menor do que escrever $f(x_3,x_2,x_1,x_0)$ usando a forma canônica SDP, *i.e.* usando a soma dos implicantes p_1 a p_4. É portanto de interesse de um processo de minimização de funções identificar implicantes do tipo p_9 do exemplo. O conceito de implicante primo ajuda a identificar implicantes desse tipo. Ele é definido na Definição 9.1.3

Definição 9.1.3 (Implicante primo de uma função) Implicante p da função f é dito implicante primo (IP) de f, se não existir nenhum outro implicante p' ($p \neq p'$) da função f tal que TODAS as atribuições de variáveis que tornam $p = 1$ também tornam $p' = 1$. Em outras palavras p é implicante primo da função f se não existir nenhum outro implicante p' da da função f tal que p' cubra p.

O IP de uma função f é um implicante que não é coberto por nenhum outro implicante da função f. O Exemplo 9.3 mostra a aplicação do conceito de implicantes primo.

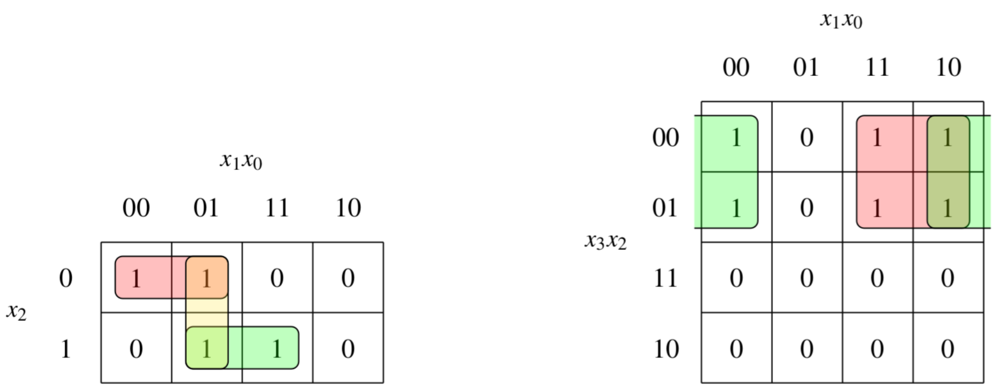
■ **Exemplo 9.3** (Implicantes primos)

No Exemplo 9.2 foram elencados os nove implicantes da função dada. O único implicante primo é o p_9 (quarteto de 1s). Pelos agrupamentos mostrados na Fig. 9.1 fica claro que os implicantes de um único 1 são cobertos por implicantes formados por duplas de 1s, enquanto que o implicantes formados por duplas de 1s são cobertos pelo implicante formado pelo quarteto de uns.

Mais exemplos de implicantes primos estão mostrados na Fig. 9.2. A Fig. 9.2 destaca os agrupamentos de 1s referentes aos IP das funções $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,5,7)$ (Fig. 9.2a) e $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(0,2,3,4,6,7)$ (Fig. 9.2b). Perceba que pode haver uma superposição parcial entre agrupamentos que representam IP. Um implicante somente deixa de ser primo se ele for totalmente superposto por outro implicante.

■

Como já discutido, os IP de uma função são termos candidatos para compor uma soma mínima



(a) (b)

Figura 9.2: Agrupamentos de 1s que representam os implicantes primos das funções: (a) $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,5,7)$ e (b) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(0,2,3,4,6,7)$.

na forma SDP. Nessa linha, o Teorema 9.1.1 mostra uma condição necessária para a obtenção de expressões mínimas do tipo SDP.

Teorema 9.1.1 (Soma mínima na forma SDP)
Toda soma mínima na forma SDP consiste numa soma de implicantes primos.

Demonstração. Uma prova por absurdo será desenvolvida. Assuma que a expressão

$$E = p_1 + p_2 + \ldots + p_k + \ldots + p_n \tag{9.1}$$

é a mínima para uma função f e que p_k são todos IP de f, exceto por p_1, que é apenas implicante (não primo) de f. Ora, se p_1 é implicante da função e não é primo, por definição é obrigatório que exista um implicante primo p' que cubra p_1 (senão p_1 seria IP). Como p' necessariamente tem menos variáveis que p_1 e o cobre, é possível substituir na expressão p_1 por p', obtendo uma expressão mais simples. Logo, não é possível que E seja mínima contendo implicantes não primos o que prova que E só pode ser miníma se for composta apenas por IPs. ■

A condição mostrada no Teorema 9.1.1 é necessária mas não suficiente para obtenção da soma mínima. Muitas vezes, a função possui mais IPs do que necessário para escrevê-la de forma mínima. Por exemplo, a função $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,5,7)$ cujo mapa-K está mostrado na Fig. 9.2a possui três IPs: $\overline{x_2}\cdot\overline{x_1}$ (em vermelho na Fig. 9.2a), $x_2\cdot x_0$ (em verde na Fig. 9.2a) e $\overline{x_1}\cdot x_0$ (em amarelo Fig. 9.2a). A soma mínima composta por IPs seria:

$$f(x_2,x_1,x_0) = \overline{x_2}\cdot\overline{x_1} + x_2\cdot x_0 + \overline{x_1}\cdot x_0. \tag{9.2}$$

No entanto, usando os teoremas e postulados da álgebra de boole aplicados a (9.2) vem:

$$\begin{aligned}
f(x_2,x_1,x_0) &= \overline{x_2}\cdot\overline{x_1} + x_2\cdot x_0 + \overline{x_1}\cdot x_0 \\
&= \overline{x_2}\cdot\overline{x_1} + x_2\cdot x_0 + \overline{x_1}\cdot x_0\cdot 1 \\
&= \overline{x_2}\cdot\overline{x_1} + x_2\cdot x_0 + \overline{x_1}\cdot x_0\cdot (x_2+\overline{x_2}) \\
&= \overline{x_2}\cdot\overline{x_1} + x_2\cdot x_0 + x_2\cdot\overline{x_1}\cdot x_0 + \overline{x_2}\cdot\overline{x_1}\cdot x_0 \\
&= \overline{x_2}\cdot\overline{x_1} + \overline{x_2}\cdot\overline{x_1}\cdot x_0 + x_2\cdot x_0 + x_2\cdot\overline{x_1}\cdot x_0 \\
&= \overline{x_2}\cdot\overline{x_1} + x_2\cdot x_0.
\end{aligned} \tag{9.3}$$

Ou seja, apenas os IPs $\overline{x_2}\cdot\overline{x_1}$ e $x_2\cdot x_0$ devem ser incluídos na soma mínima enquanto o IP $\overline{x_1}\cdot x_0$ deve ser descartado. Isso acontece porque o IP $\overline{x_1}\cdot x_0$ "cobre" dois 1s que já estão cobertos pelos IPs $\overline{x_2}\cdot\overline{x_1}$ e $x_2\cdot x_0$, o que pode ser visto claramente nos agrupamentos mostrados na Fig. 9.2a. A definição formal para cobertura de 1s por parte de implicantes é dada na Definição 9.1.4.

Definição 9.1.4 (Cobertura de um 1 de uma função f) Considere a função f definida por seu conjunto um, $f = \text{Conjunto-UM}(u_0, u_1, \ldots u_{m-1})$ (refira-se à notação apresentadas na Seção **??**). Se a função f for escrita na forma de soma de implicantes p_k, *i.e.* $f = \sum p_k$, diz-se que o 1 u_m está coberto na soma $\sum p_k$ se existir pelo menos um implicante p_k em $\sum p_k$ tal que $p_k(u_m) = 1$.

A pergunta que surge agora é como escolher os implicantes de uma função para escrevê-la de forma miníma? Primeiro passo é adicionar à soma miníma os implicantes fundamentais para compô-la. Continuando com o exemplo da $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,5,7)$ (mostrada na Fig. 9.2a) observe que os IPs $\overline{x_2}\cdot\overline{x_1}$ e $x_2\cdot x_0$ são especiais: O único IP que cobre o 1 em $x_2x_1x_0 = 000$ é o $\overline{x_2}\cdot\overline{x_1}$, enquanto o único IP que cobre o 1 em $x_2x_1x_0 = 111$ é o $x_2\cdot x_0$. Esses IPs têm um nome especial, são chamados de implicantes primos essenciais. A Definição 9.1.5 os define.

Definição 9.1.5 (Implicante primo essencial de uma função) Um implicante primo p da função f é dito implicante primo essencial (IPE) se existir pelo menos uma atribuição v de variáveis que torna $p = 1$ e essa mesma atribuição v nao torna nenhum outro implicante primo p' ($p \neq p'$) de f igual a 1. Matematicamente, p é um implicante primo essencial de f se existe v tal que $p(v) = 1$ e $p'(v) = 0$ para todos os outros implicantes primos p' de f.

O Exemplo 9.4 mostra mais exemplos de IPEs. Como todos os 1s de uma função tem que ser cobertos por pelo menos 1 dos implicantes primos da soma mínima, os IPEs devem ser sempre incluídos na soma mínima conforme estabelecido no Teorema 9.1.2.

■ **Exemplo 9.4** A Fig. 9.3a mostra os IP da função $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,3,4,6,7)$, Alguns são IPEs e outros não. O agrupamento marcado em vermelho representa um IPE de f pois ele é o único IP que cobre o 1 em $x_2x_1x_0 = 000$. O mesmo ocorre com o agrupamento marcado em verde, pois ele é o único IP que cobre o 1 em $x_2x_1x_0 = 011$ sendo assim um IPE de f. Já os IPs em amarelo e azul possuem todos os uns cobertos por outros IPs não sendo portanto IPEs. A Fig. 9.3b mostra os agrupamentos relativos aos IPE da função $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,2,3,5,7,12,8)$. ■

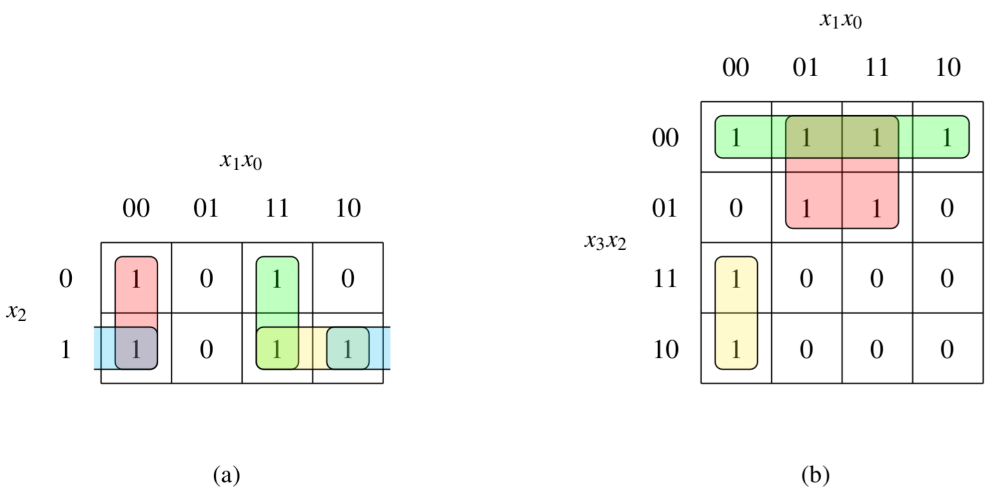


Figura 9.3: Exemplos de implicante primos essenciais.

Teorema 9.1.2 (Inclusão de IPEs na soma mínima) Toda soma mínima de uma função na forma SDP deve incluir todos os IPEs da função.

Ou seja, a soma mínima será composta primeiramente pelos IPEs. Caso os IPEs não cubram todos os 1s do Conjunto-UM da função, um conjunto mínimo de IPs devem ser incluídos na soma mínima. Somente devem ser incluídos IPs na soma mínima se eles cobrirem 1s ainda não cobertos pelos outros IPs/IPEs já incluídos na soma mínima.

O procedimento mostrado no Algoritmo 1 resume como se obter a expressão mínima na forma soma de produtos.

Algorithm 1 Minimização na forma soma de produtos.

Dado: Conjunto-UM() de uma função de chaveamento f de n variáveis

```
Determine todos os implicantes primos da função f
Dentre os implicantes primos determine os implicantes primos essenciais de f
Inclua todos os implicantes primos essenciais na soma de produtos.
se (Ainda há 1s não cobertos pelo IPEs) então
    Escolha dentre os implicantes primos restantes o conjunto mínimo que cubra todos os 1s faltantes.
fim se
retorne A soma mínima para f na forma SPD.
```

9.1.2 Procedimento de minimização usando mapas-K.

Para facilitar a aplicação do procedimento de minimização mostrado no Algoritmo 1 aos mapas-K é possível se estabelecer algumas regras que devem ser seguidas e observadas simultaneamente no processo de agrupamento de 1s nos mapas-K:

1. Um novo agrupamento de 1s só deve ser feito se ele contiver 1s ainda não agrupados.
2. Sobreposição parcial de agrupamentos são permitidas.

3. O mesmo 1 pode ser agrupado várias vezes no mapa-K.
4. Os maiores agrupamentos possíveis de 2^n 1s vizinhos devem ser feitos sempre que possível.
5. A menor quantidade possível de grupos deve ser formada.
6. Todos os 1s devem estar agrupados pelo menos uma vez.
7. Os 0s não devem ser agrupados.

Perceba que todas as regras tem que ser obedecidas simultaneamente. Não é possível aplicar uma das regras se ela viola outra regra. No caso de um mapa-K de 4 variáveis pode ser aplicado o Algoritmo 2 para a obtenção dos agrupamentos de 1s.

Algorithm 2 Agrupamentos de 1s em mapas-K de 4 variáveis.

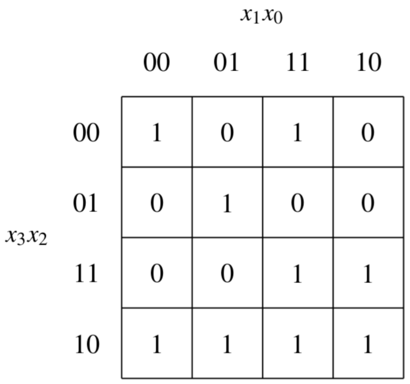
Dado: Conjunto-UM() de uma função de chaveamento f de n variáveis
1: Preencha o mapa-K com os 1s e 0s correspondentes Conjunto-UM() da função f.
2: Agrupe os 1s isolados no mapa-K;
3: Agrupe os 1s que só possuem um único vizinho no mapa-K;
4: **para** $i = 0$ até $n-2$ **faça**
5:     **enquanto** (Houver 1s ainda não agrupados que formem grupos de 2^{n-i} 1s) **faça**
6:         Faça um agrupamento de 2^{n-i} 1s contendo pelo menos um 1 ainda não agrupado.
7:     **fim enquanto**
8: **fim para**
9: Escreva o termo produto p_k de cada agrupamento feito no mapa-K.
10: **retorne** A soma mínima para f na forma SPD em que $f = \sum p_k$ ($\sum$ representa operações OR)

A Fig. 9.4a mostra o passo a passo da aplicação do Algoritmo 2 a uma função de chaveamento de 4 variáveis: $f(x_3,x_2,x_1,x_0) =$ Conjunto-UM$(0,3,5,8,9,10,11,14,15)$. Após a realização do passo 1 do Algoritmo 2 se chega ao resultado mostrado em na Fig. 9.4a. Após procurar por uns isolados no mapa-K (linha 2) encontra-se como resultado o 1 destacado na Fig. 9.4b. Na linha 3, procura-se por 1s que possuem um único vizinho no mapa-K. Após a aplicação desse passo chega-se aos agrupamentos mostrados em verde e amarelo (os 1s em 0000 e em 0011 possuem um único vizinho) na Fig. 9.4b.

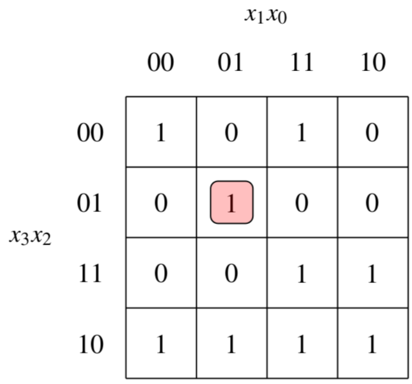
Entra-se no laço para da linha 4. Ainda há 1s não agrupados no mapa-K mas não há agrupamentos de 2^{n-i} ($2^{4-0} = 16$) 1s. Incrementa-se o valor de i do laço. Novamente ainda há 1s não agrupados no mapa-K mas não há agrupamentos de 2^{n-i} ($2^{4-1} = 8$) 1s no mapa-K. Incrementa-se o valor de i do laço. Ainda há 1s não agrupados no mapa-K mas agora há dois agrupamentos de 1s contendo 2^{n-i} ($2^{4-2} = 4$) possíveis no mapa-K. Aplica-se o passo da linha 6 e se faz o agrupamento mostrado em azul na Fig. 9.4d. Novamente se verifica a condição do enquanto da linha 5 pois ainda há 1s não agrupados no mapa-K e um agrupamento de 1s contendo 2^{n-i} ($2^{4-2} = 4$) 1s. Aplica-se o passo da linha 6 e se faz o agrupamento mostrado em roxo na Fig. 9.4e.

Como todos os 1s foram agrupados o Algoritmo 2 não tenta fazer mais nenhum agrupamento indo para o passo 9. Aplicando-se o passo 9 chega-se aos termos produtos, $p_1 = \overline{x_3}{\cdot}x_2{\cdot}\overline{x_1}{\cdot}x_0$ (vermelho), $p_2 = \overline{x_2}{\cdot}\overline{x_1}{\cdot}\overline{x_0}$ (verde), $p_3 = \overline{x_2}{\cdot}x_1{\cdot}x_0$ (amarelo), $p_4 = x_3{\cdot}\overline{x_2}$ (azul), $p_5 = x_3{\cdot}x_1$ (roxo). E portanto a soma mínima para f na forma SPD é:

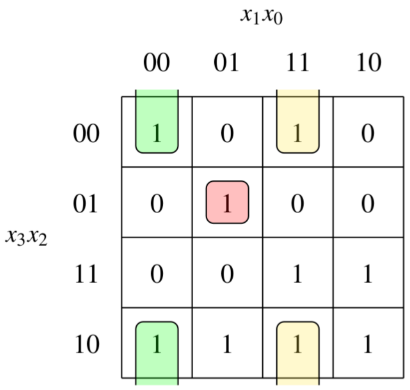
$$f(x_3,x_2,x_1,x_0) = \sum p_k = \overline{x_3}{\cdot}x_2{\cdot}\overline{x_1}{\cdot}x_0 + \overline{x_2}{\cdot}\overline{x_1}{\cdot}\overline{x_0} + \overline{x_2}{\cdot}x_1{\cdot}x_0 + x_3{\cdot}\overline{x_2} + x_3{\cdot}x_1.$$



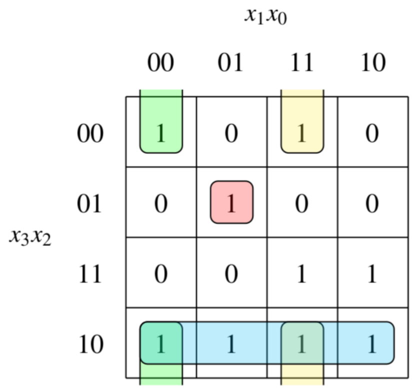
(a)



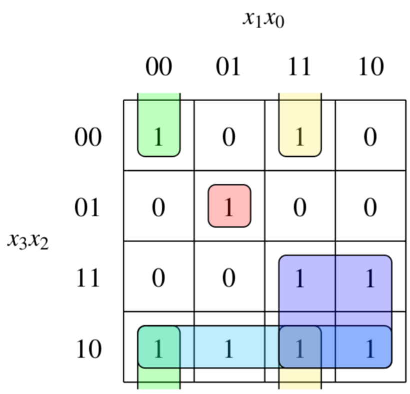
(b)



(c)



(d)



(e)

Figura 9.4: Aplicação do Algoritmo 2 na obtenção dos agrupamentos de 1s que formam a expressão mínima na forma SDP da função $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(0,3,5,8,9,10,11,14,15)$.

9.1.3 Exemplos de implementação de sistemas mínimos usando mapas-K

Nesta seção são mostrados a implementação mínima para alguns sistemas exemplo usando a metodologia de mapas-K.

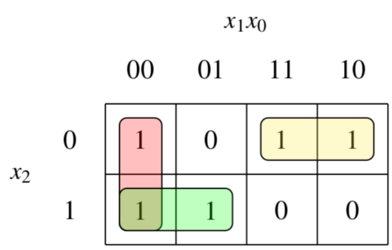
■ **Exemplo 9.5** (Implementação mínima de um sistema de três variáveis)

Um sistema combinacional S com três entradas e uma saída é descrito por meio da função de chaveamento $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,2,3,4,5)$. Encontre a implementação mínima para esse sistema.

A tabela verdade desse sistema está mostrada na Fig. 9.12a. Preenche-se o mapa-K usando a tabela verdade. O mapa-K preenchido para a função está na Fig. 9.12b. Aplicando a teoria vista na Seção 9.1.1 e as regras da Seção 9.1.2 chega-se aos agrupamentos mostrados na Fig. 9.12b.

i	x_2	x_1	x_0	$f(x_2,x_1,x_0)$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

(a)



(b)

Figura 9.5: Tabelas usadas na obtenção da expressão mínima da função do Exemplo 9.12

Os agrupamentos mostrados têm as seguintes expressões: $\overline{x_1}{\cdot}\overline{x_0}$ (vermelho), $x_2{\cdot}\overline{x_1}$ (verde), $\overline{x_2}{\cdot}x_1$ (amarelo). A implementação mínima para esse sistema é portanto dada por:

$$f(x_2,x_1,x_0) = \overline{x_1}{\cdot}\overline{x_0} + x_2{\cdot}\overline{x_1} + \overline{x_2}{\cdot}x_1$$

■

■ **Exemplo 9.6** (Codificador de prioridades) Encontre a implementação mínima de um codificador de prioridades de 4 *bits*.

Esse é um sistema apresenta quatro entrada x_3, x_2, x_1 e x_0 e três saídas y_1, y_0 e c. O funcionamento do sistema pode ser resumido pela Tabela 9.1. Na tabela os traços verticais indicam que os valores referentes não importam.

Tabela 9.1: Tabela de funcionamento do codificador de prioridades.

x_3	x_2	x_1	x_0	y_1	y_0	c
1	-	-	-	1	1	1
0	1	-	-	1	0	1
0	0	1	-	0	1	1
0	0	0	1	0	0	1
0	0	0	0	0	0	0

Existem 4 entradas nesse sistema, cada entrada está ativa com um 1 e inativa com um 0. O sistema deve indicar na saída qual a entrada de maior prioridade está ativa no momento. A saída c indica com 1 se alguma das 4 entradas está ativa e fica em 0 caso contrário. A ordem de prioridade das entradas é, nessa ordem, x_3, x_2, x_1 e x_0. As saídas y_1, y_0 indicam o índice da entrada de maior prioridade ativa. Por exemplo, $y_1 y_0 = 11$ indica que x_3 é a entrada de maior prioridade ativa. Perceba que $(11)_2 = (3)_{10}$. A tabela verdade completa para esse sistema está mostrada na Tabela 9.2.

Tabela 9.2: Tabela verdade do codificador de prioridades.

i	x_3	x_2	x_1	x_0	y_1	y_0	c
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1
2	0	0	1	0	0	1	1
3	0	0	1	1	0	1	1
4	0	1	0	0	1	0	1
5	0	1	0	1	1	0	1
6	0	1	1	0	1	0	1
7	0	1	1	1	1	0	1
8	1	0	0	0	1	1	1
9	1	0	0	1	1	1	1
10	1	0	1	0	1	1	1
11	1	0	1	1	1	1	1
12	1	1	0	0	1	1	1
13	1	1	0	1	1	1	1
14	1	1	1	0	1	1	1
15	1	1	1	1	1	1	1

Perceba que a saída c tem resolução trivial, sendo claramente implementada por uma porta OR. Para as outras saídas, são necessários dois mapas-K (mostrados na Fig. 9.6), um para y_1 (Fig. 9.6a) e outro para y_0 (Fig. 9.6b). Os agrupamentos para implementação da função miníma desse sistema também está mostrado na Fig. 9.6.

Escrevendo as expressões mínimas a partir dos agrupamentos mostrados vem:

$$y_1 = x_2 + x_3 \tag{9.4}$$

e

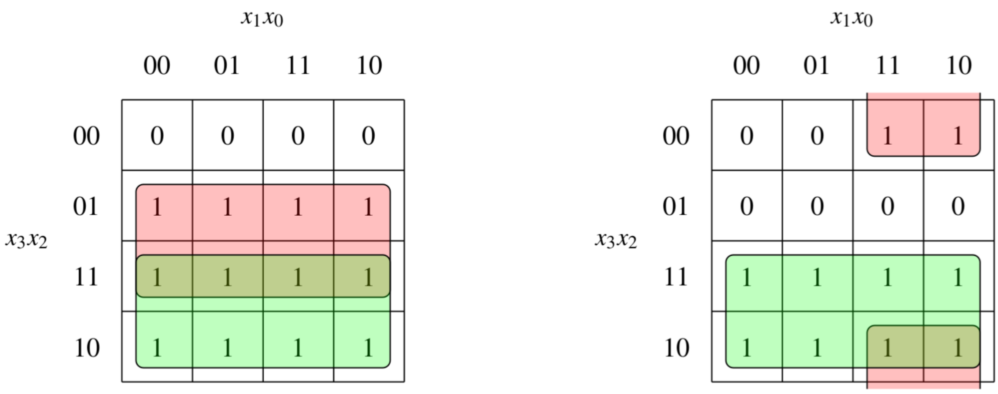
$$y_0 = \overline{x_2} \cdot x_1 + x_3 \tag{9.5}$$

■

9.1.4 Implementação de sistemas mínimos usando mapas-K com funções não completamente especificadas.

No caso de uma função ser não completamente especificada. Os valores não especificados devem ser identificados e preenchidos no mapa-K com um “x” (ou “-”). Todo o procedimento de simplificação aplicado aos mapas-K pode ser aplicado nesse contexto com duas diferenças:

- Os “x” (ou “-”) podem ou não serem agrupados como se fossem 1s.



(a) Mapa-K para a saída y_1 (b) Mapa-K para a saídada y_0

Figura 9.6: Mapas-K para implementação mínima do codificador de prioridades discutido no Exemplo 9.6.

- Os "x" (ou "-") devem ser agrupados quando, ao serem agrupados, gerem agrupamentos maiores (resultando em mais variáveis eliminadas).
- Não é obrigatório o agrupamento de todos os "x" (ou "-") como é obrigatório o agrupamento de todos os 1s.

O Exemplo 9.13 mostra uma função não completamente especificada sendo implementada por meio de mapas-K.

■ **Exemplo 9.7** (Detector de primalidade)

Esse sistema é o um sistema que recebe um dígito BCD em sua entradas (4 *bits* de entrada) e fornece na sua saída (1 *bit* de saída) 1 se o dígito da entrada for um número primo e 0 caso contrário.

As entradas do sistema são os *bits* x_3, x_2, x_1 e x_0 e a saída o *bit* y. Como na entrada se tem 4 bits e um dígito BCD, apenas os número de 0 a 9 são permitidos. Portanto, os números de 10 a 15 apesar de serem padrões de entrada possíveis (pois há quatro *bits* na entrada) são padrões de entrada que nunca serão apresentados ao sistemas constituindo assim padrões aos quais pode se atribuir saídas não especificadas. A Tabela 9.9 mostra a tabela verdade completa desse sistema.

Preenchendo o mapa-K a partir da tabela verdade chega-se ao mapa-K mostrado na Fig. 9.13.

Note que a expressão mínima pode ser obtida conforme os agrupamentos mostrados na Fig. 9.13. Os dois quartetos agrupados incluiram duas saídas "-"(indeterminadas). Escrevendo a expressão mínima de implementação desse sistema vem:

$$y = x_2 \cdot x_0 + \overline{x_2} \cdot x_1 \tag{9.6}$$

■

Tabela 9.3: Tabela verdade do detector de primalidade de para dígitos BCD.

i	x_3	x_2	x_1	x_0	y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	-
11	1	0	1	1	-
12	1	1	0	0	-
13	1	1	0	1	-
14	1	1	1	0	-
15	1	1	1	1	-

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

Figura 9.7: Mapa-K do detector de primalidade de para dígitos BCD.

9.1.5 Método Quinee-Mcluskey para minimização

Esse é um outro método que pode ser usado para se encontrar funções de chaveamento mínimas de dois níveis.

O método é baseado no preenchimento sistemático de duas tabelas. Na primeira tabela, são determinados todos os implicantes primos da função. Na segunda tabela, são determinados os implicantes primos essenciais e quais implicantes devem ser incluídos na soma de produtos mínima que expressa a função.

Obtenção da primeira tabela

O objetivo da primeira tabela é encontrar os implicantes primos da função. Para forma-la, os mintermos componentes da função são escritos em sua forma binária. São testados todas as combinações 2 a 2 desses mintermos. Caso a distância entre os mintermos sob teste seja igual a 1, a identidade de minimização é aplicada. Os mintermos combinados deixam de ser candidatos a implicantes primos da função uma vez que o termo resultante da combinação é uma implicante da função e cobre os mintermos combinados. Por exemplo, os mintermos cujas formas binárias são 1010 e 1011 possuem distância 1. A identidade de minimização pode ser a eles aplicada e sua combinação gera o termo $101-$. Note que a variável eliminada, fruto da combinação dos mintermos, foi substituída por um traço.

Após combinar todos mintermos 2 a dois 2, surge um conjunto de termos que possuem uma variável eliminada. O processo é repetido nesses termos, ou seja, todas as combinações 2 a 2 são tentadas entre os termos que possuem uma variável eliminada. Para aplicar a identidade minimização a esses termos, novamente a distância binária deve ser igual a 1, mas os traços devem estar nas mesmas posições. Por exemplo, os termos $101-$ e $100-$ podem ser combinados pois têm distancia igual a 1 e o traço está na mesma posição. Essa combinação gera o termo com duas variáveis eliminadas: $10--$. O processo é repetido até que Não seja mais possível aplicar combinações 2 a 2. Os mintermos são organizados na primeira coluna da tabela, os termos com uma variável eliminada na segunda coluna da tabela, e assim por diante. Toda vez que um termo é combinado ele recebe um "N" para indicar que ele não é mais um candidato a implicante primo da função. Ao final do processo, todos os termos não marcados com "N" são implicantes primos da função.

Obtenção da segunda tabela

A segunda tabela possui um número de linhas igual ao número de implicantes primos que foram identificados pela primeira tabela. O número de colunas é igual ao número de mintermos da função. Se eu implicante primo da i-ésima linha cobrir o mintermo da j-ésima coluna um "x" é colocado na linha i coluna j. Todas as linhas devem ser preenchidas dessa forma. Colunas que tenham apenas um "x" Significam que apenas um implicante primo cobre o mintermo posicionado naquela coluna. Ou seja, o implicante primo posicionado Na linha que contém um "x" isolado em uma coluna é um implicante primo essencial. Todos os IPEs são assim identificados e incluidos na soma mínima. Numa linha extra adicionada ao final da tabela, as colunas referentes aos mintermos cobertos pelos IPEs devem receber um "x". Feito isso, caso alguma coluna da última linha não possua um "x" outros IPs, dispostos nas linhas, devem ser adicionados para cobrir os mintermos ainda não cobertos. O menor conjunto possível de IPs deve ser adicionado.

Aplicação do método

Nesta seção, o método de de minimização tabular Quinee-Mcluskey será aplicado para a obtenção de expressão mínima da função $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(1,4,6,7,9,10,11,13,15)$.

Para facilitar as combinações dois a dois que tem que ser feitas para a obtenção da primeira tabela, os mintermos devem ser organizados pelo número de 1s. Pois só pode haver mudança de um único bit entre mintermos que contem i 1s e $i+1$ 1s. Os mintermos da funçao f assim organizados estão mostrados na Tabela 9.4.

Tabela 9.4: Mintermos da função f organizados pelo número de 1s do mintermo.

zero 1s	um 1	dois 1s	três 1s	quatro 1s
	0001	0110	0111	1111
	0100	1001	1011	
		1010	1101	

A primeira primeira coluna da primeira tabela é preenchida com os mintermos dispostos em ordem que considera número de 1s de cada mintermo conforme mostrado na Tabela 9.5a. As condenações 2 a 2 começam a ser aplicadas na primeira coluna da Tabela 9.5a. Por exemplo, é testada a combinação dois a dois do mintermo 0001 (tem um único 1) com os mintermos 0110, 1001 e 1010 (que têm dois 1s). Dentre esses três, o mintermo 0001 só possui distancia um com o mintermo 1001. Aplicando a identidades de simplificação chega-se ao termo -001 que é o primeiro termo colocado na segunda coluna da Tabela 9.5b. Os mintermos 0001 e 1001 são marcados com "N" como mostrado na Tabela 9.5b. Esse processo é repetido com todos os mintermos da primeira coluna da Tabela 9.5a e cujo resultado final pode ser visto na Tabela 9.5c. Todo processo é repetido novamente todas as combinações 2 a 2 dos termos presente na segunda coluna da Tabela 9.5c resultado na terceira coluna mostrada na Tabela 9.5d. Como não podem ser feitas mais combinações da terceira coluna da Tabela 9.5d, O processo é encerrado e os termos não marcados com "N" são os IPs da função f, a saber, -001, $01-0$, $011-$, $101-$, -111 e $1--1$ (que em termos das variáveis são $\overline{x_2}{\cdot}\overline{x_1}{\cdot}x_0$, $\overline{x_3}{\cdot}x_2{\cdot}\overline{x_0}$, $\overline{x_3}{\cdot}x_2{\cdot}x_1$, $x_3{\cdot}\overline{x_2}{\cdot}x_1$, $x_2{\cdot}x_1{\cdot}x_0$ e $x_3{\cdot}x_0$).

Inicia-se então, a elaboração da segunda tabela. Na primeira coluna são colocados, linha a linha, os implicantes primos encontrados. Na primeira linha, coluna a coluna, são colocados os mintermos da função. Esse preenchimento está mostrado na Tabela 9.6. Feito isso, identifica-se quais implicantes primos cobrem quais mintermos. Isso é feito marcando "x" da respectiva linha/coluna. Por exemplo, o IP -001 cobre os mintermos 001 e 1001 por isso na linha de -001 são marcados "x" nas colunas respectivas aos mintermos 001 e 1001 conforme mostrado na primeira linhas da Tabela 9.7. Fazendo o mesmo para os outros IPs Preenche-se a tabela conforme na Tabela 9.7. A coluna respectiva o mintermo 0001 (primeira coluna da Tabela 9.7) possui um único "x". Esse "x" está localizado na linha do IP -001, logo o IP -001 é um IPE, pois ele é o único IP que cobre o mintermo 0001. Fazendo o mesmo processso marca-se com um (E) todos os IPEs na tabela resultando nas marcações mosradas na Tabela 9.8. Todos os IPEs devem ser incluídos na soma de produtos mínima. Os mintermos que são cobertos pelos IPEs são marcados com um "x" na última linha da tabela (linha "coberto?"). Perceba que todos os implicantes primos essenciais da função cobrem todos os mintermos da função, exceto um, o mintermo 0111. Há Dois implicantes primos que cobrem esse mintermo não coberto: $011-$ e -111. Como ambos apresentam o mesmo número de variáveis, qualquer um dos dois pode ser escolhido para compor a soma mínima. Portanto, essa função apresenta duas formas mínimas possíveis: a soma de todos os implicantes primos essenciais mais o implicante primo $011-$ ou a soma de todos os implicantes primos essenciais mais o implicante primo -111. No caso da primeira escolha a soma de produtos mínima é dada por:

$$f(x_3,x_2,x_1,x_0) = \overline{x_2}{\cdot}\overline{x_1}{\cdot}x_0 + \overline{x_3}{\cdot}x_2{\cdot}\overline{x_0} + x_3{\cdot}\overline{x_2}{\cdot}x_1 + x_3{\cdot}x_0 + \overline{x_3}{\cdot}x_2{\cdot}x_1 \tag{9.7}$$

e no caso da segunda por:

$$f(x_3,x_2,x_1,x_0) = \overline{x_2}\cdot\overline{x_1}\cdot x_0 + \overline{x_3}\cdot x_2\cdot\overline{x_0} + x_3\cdot\overline{x_2}\cdot x_1 + x_3\cdot x_0 + x_2\cdot x_1\cdot x_0. \quad (9.8)$$

Tabela 9.5: Preenchimento passo a passo da tabela usada no método Quinee-Mcluskey para a identificação dos implicantes primos de uma função.

4 Var	3 Var	2 Var	1 Var
0001			
0100			
0110			
1001			
1010			
0111			
1011			
1101			
1111			

(a)

4 Var		3 Var	2 Var	1 Var
0001	N	-001		
0100				
0110				
1001	N			
1010				
0111				
1011				
1101				
1111				

(b)

4 Var		3 Var	2 Var	1 Var
0001	N	-001		
0100	N	01-0		
0110	N	011-		
1001	N	10-1		
1010	N	1-01		
		101-		
0111	N			
1011	N	-111		
1101	N	1-11		
		11-1		
1111	N			

(c)

4 Var		3 Var		2 Var	1 Var
0001	N	-001		1- -1	
0100	N	01-0			
0110	N	011-			
1001	N	10-1	N		
1010	N	1-01	N		
		101-			
0111	N				
1011	N	-111			
1101	N	1-11	N		
		11-1	N		
1111	N				

(d)

Tabela 9.6: Montagem da tabela de cobertura mínima usada no método Quinee-Mcluskey para a identificação da soma de produtos mínima para a implementação de uma função.

Mintermos IPs	0001	0100	0110	0111	1001	1010	1011	1101	1111
-001									
01-0									
011-									
101-									
-111									
1- -1									
Coberto?									

Tabela 9.7: Preenchimento das coberturas do mintermos por IP na segunda tabela usada no método Quinee-Mcluskey.

Mintermos IPs	0001	0100	0110	0111	1001	1010	1011	1101	1111
-001	x				x				
01-0		x	x						
011-			x	x					
101-						x	x		
-111				x					x
1- -1					x		x	x	x
Coberto?									

Tabela 9.8: Preenchimento das coberturas do mintermos por IP na segunda tabela usada no método Quinee-Mcluskey.

Mintermos IPs	0001	0100	0110	0111	1001	1010	1011	1101	1111
-001 (E)	x				x				
01-0 (E)		x	x						
011-			x	x					
101- (E)						x	x		
-111				x					x
1- -1 (E)					x		x	x	x
Coberto?	x	x	x		x	x	x	x	x

9.2 Obtenção de expressões mínimas na forma PDS

Na Seção 8.2 são discutidas as premissas consideradas para a obtenção de funções de chaveamento minímas que representam uma certa tabela verdade/sistema. Usando as premissas Seção 8.2 são apresentados na presente seção a teoria para obtenção de expressões mínimas na forma PDS usando o método de mapa-K.

9.2.1 Teoria para a obtenção de PDS mínimas

Para estabelecer um critério de minimização é necessário se definir alguns conceitos. O primeiro deles é o conceito de implicado de uma função mostrado na Definição 9.2.1.

> **Definição 9.2.1** (Implicado de uma função) Um termo produto s é definido como um implicado da função f, se para toda atribuição de variáveis que resulta em s igual a 0 também resulta em f igual a 0.

O Exemplo 9.8 mostra um exemplo de um implicado de uma função.

■ **Exemplo 9.8** (Implicado de uma função)

Como exemplo para implicados, considere o termo soma s, $s(x_2,x_1,x_0) = x_1+x_0$ que é um implicado da função $f(x_2,x_1,x_0) =$ Conjunto-ZERO$(0,1,3,4,6,7)$. Isso porque todas as atribuições para as variáveis $x_2x_1x_0$ que tornam $s = 0$ também tornam $f(x_2,x_1,x_0) = 0$. As atribuições para as variáveis $x_2x_1x_0$ que tornam $s = 0$ são $x_2x_1x_0 = 000$ e $x_2x_1x_0 = 100$. Essa duas atribuições correspondem, em decimal, aos índices 0 e 4 e ambos estão incluídos no Conjunto-ZERO() de $f(x_2,x_1,x_0)$. ■

Observe que todas as funções de chaveamento podem ser escrita como produto de implicados. Um exemplo disso é a produto de maxtermos visto no Capítulo **??**. Perceba que cada maxtermo incluído em uma função específica constitui em um implicado da função.

O Exemplo 9.9 mostra a obtenção de todos os implicados de um função.

■ **Exemplo 9.9** (Lista de implicados de uma função)

Considere a função $f(x_3,x_2,x_1,x_0) =$ Conjunto-ZERO$(5,7,13,15)$. Essa função possui 9 implicados s_1 a s_9 , todos destacados na Fig. 9.8. São os quatro maxtermos $s_1 = x_3+\overline{x_2}+x_1+\overline{x_0}$, $s_2 = x_3+\overline{x_2}+\overline{x_1}+\overline{x_0}$, $s_3 = \overline{x_3}+\overline{x_2}+x_1+\overline{x_0}$ e $s_4 = \overline{x_3}+\overline{x_2}+\overline{x_1}+\overline{x_0}$ (Fig. 9.8a), os termos $s_5 = x_3+\overline{x_2}+\overline{x_0}$ e $s_6 = \overline{x_3}+\overline{x_2}+\overline{x_0}$ (duplas horizontais na Fig. 9.8b), os termos $s_7 = \overline{x_2}+x_1+\overline{x_0}$ e $s_8 = \overline{x_2}+\overline{x_1}+\overline{x_0}$ (duplas verticais na Fig. 9.8c) e o termo $s_9 = \overline{x_2}+\overline{x_0}$ (quarteto na Fig. 9.8d)

■

É possível notar no Exemplo 9.9 e na Fig. **??** que existem implicados "maiores" (*i.e.* com mais variáveis) e implicados "menores" (*i.e.* com menos variáveis). Uma relação objetiva entre implicados é estabelecida na Definição 9.2.2.

> **Definição 9.2.2** (Relação de cobertura entre implicados) Considere dois implicados s_1 e s_2 de f tais que $s_1 \neq s_2$. Se TODAS as atribuições de variáveis que tornam $s_1 = 0$ também resultarem em $s_2 = 0$ diz-se que s_2 cobre s_1. Aqui a notação usada é: $s_2 \succ s_1$ (s_2 cobre s_1) ou $s_1 \prec s_2$ (s_1 é coberto por s_2) .

Começa a ficar claro que, na escrita de expressões mínimas, é preferível se optar por implicados que cobrem à implicados que são cobertos. A título de exemplo da relação de cobertura, no Exem-

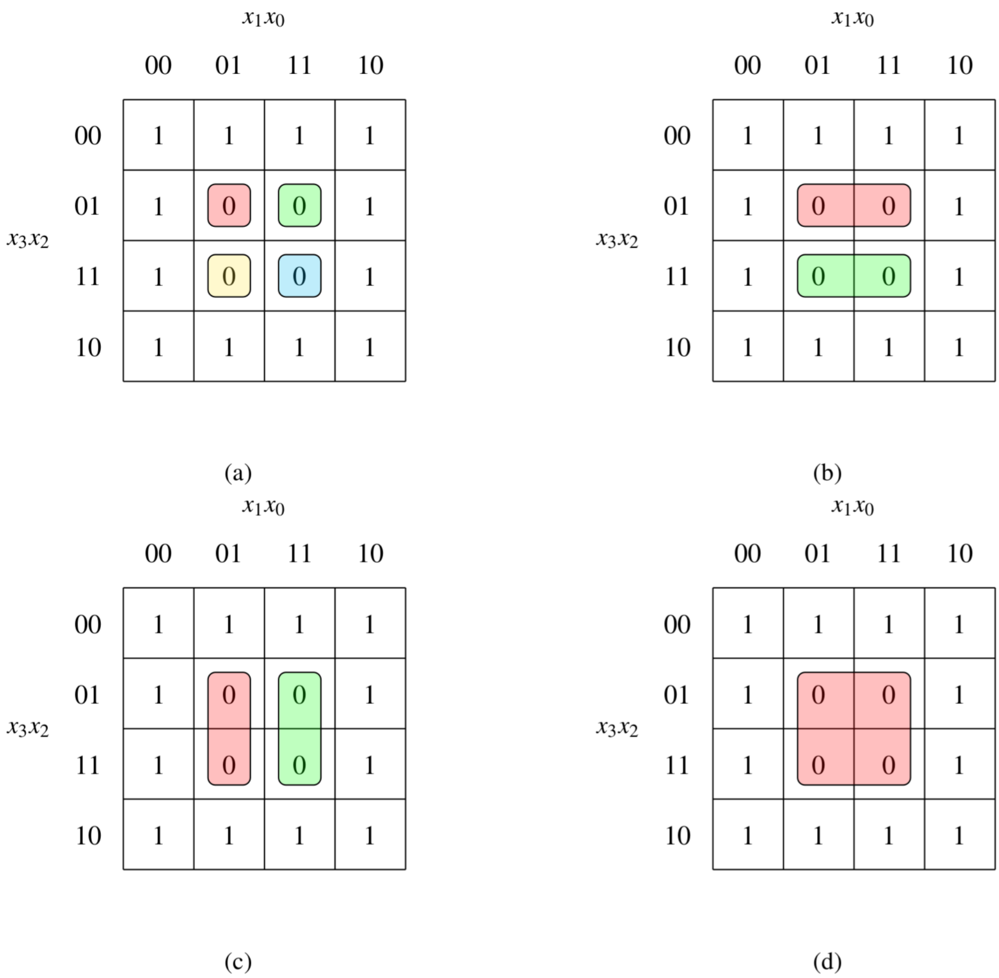


Figura 9.8: Todos os implicados da $f(x_3,x_2,x_1,x_0) =$ Conjunto-ZERO$(5,7,13,15)$ mostrados por agrupamentos de 0s no mapa-K.

plo 9.9 são elencados todos os 9 implicados da função $f(x_3,x_2,x_1,x_0) =$ Conjunto-ZERO$(5,7,13,15)$ e relacionados como s_1 a s_9. Ao usar a Definição 9.2.2 à $f(x_3,x_2,x_1,x_0)$ conclui-se facilmente que:

- s_9 cobre s_1 a s_8,
- s_8 cobre s_2 e s_4,
- s_7 cobre s_1 e s_3,
- s_6 cobre s_3 e s_4,
- s_5 cobre s_1 e s_2.

Observe pelo exemplo que, necessariamente, se $s_2 \succ s_1$ isso significa que o termo soma s_2 possui menos variáveis que o termo soma s_1. Ainda observando o Exemplo 9.9 fica claro que escrever a função $f(x_3,x_2,x_1,x_0)$ usando o implicado s_9 gera uma expressão algébrica menor do que escrever $f(x_3,x_2,x_1,x_0)$ usando a forma canônica PDS, *i.e.* usando a produto dos implicados s_1 a s_4. É portanto de interesse de um processo de minimização de funções identificar implicados do tipo s_9 do exemplo. O conceito de implicado primo ajuda a identificar implicados desse tipo. Ele é definido na Definição 9.2.3

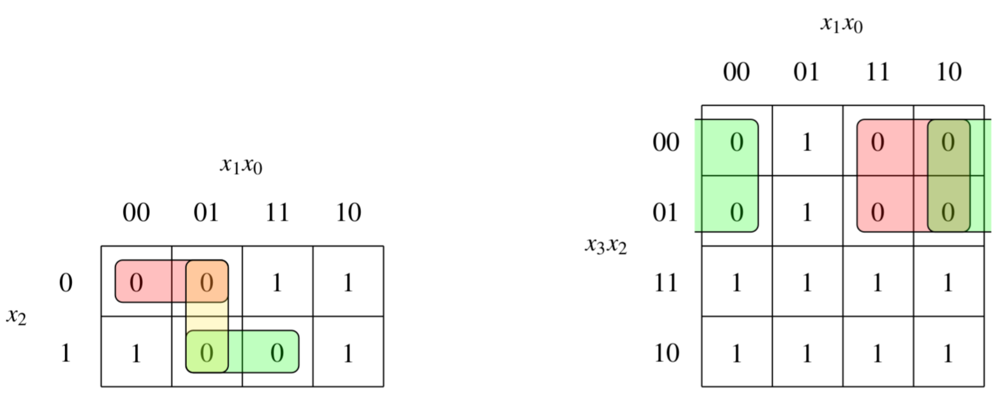
Definição 9.2.3 (Implicado primo de uma função) O Implicado s da função f é dito implicado primo (IIP) de f, se não existir nenhum outro implicado s' ($s \neq s'$) da função f tal que TODAS as atribuições de variáveis que tornam $s = 0$ também tornam $s' = 0$. Em outras palavras s é implicado primo da função f se não existir nenhum outro implicado s' da da função f tal que s' cubra s.

O IIP de uma função f é um implicado que não é coberto por nenhum outro implicado da função f. O Exemplo 9.10 mostra a aplicação do conceito de implicados primo.

■ **Exemplo 9.10** (Implicados primos)

No Exemplo 9.9 foram elencados os nove implicados da função dada. O único implicado primo é o s_9 (quarteto de 0s). Pelos agrupamentos mostrados na Fig. 9.8 fica claro que os implicados de um único 0 são cobertos por implicados formados por duplas de 0s, enquanto que o implicados formados por duplas de 0s são cobertos pelo implicado formado pelo quarteto de 0s.

Mais exemplos de implicados primos estão mostrados na Fig. 9.9. A Fig. 9.9 destaca os agrupamentos de 0s referentes aos IIP das funções $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,1,5,7)$ (Fig. 9.9a) e $f(x_3,x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,2,3,4,6,7)$ (Fig. 9.9b). Perceba que pode haver uma superposição parcial entre agrupamentos que representam IIPs. Um implicado somente deixa de ser primo se ele for totalmente superposto por outro implicado.



(a) (b)

Figura 9.9: Agrupamentos de 1s que representam os implicados primos das funções: (a) $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,1,5,7)$ e (b) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,2,3,4,6,7)$.

■

Como já discutido, os IIP de uma função são termos candidatos para compor uma produto mínima na forma PDS. Nessa linha, o Teorema 9.2.1 mostra uma condição necessária para a obtenção de expressões mínimas do tipo PDS.

Teorema 9.2.1 (produto mínimo na forma PDS)
Toda produto mínima na forma PDS consiste num produto de implicados primos.

Demonstração. Uma prova por absurdo será desenvolvida. Assuma que a expressão

$$E = s_1 \cdot s_2 \cdot \ldots \cdot s_k \cdot \ldots \cdot s_n \tag{9.9}$$

é a mínima para uma função f e que s_k são todos IIP de f, exceto por s_1, que é apenas implicado (não primo) de f. Ora, se s_1 é implicado da função e não é primo, por definição é obrigatório que exista um implicado primo s' que cubra s_1 (senão s_1 seria IIP). Como s' necessariamente tem menos variáveis que s_1 e o cobre, é possível substituir na expressão s_1 por s', obtendo uma expressão mais simples. Logo, não é possível que E seja mínima contendo implicados não primos o que prova que E só pode ser miníma se for composta apenas por IIPs. ■

A condição mostrada no Teorema 9.2.1 é necessária mas não suficiente para obtenção d0 produto mínimo. Muitas vezes, a função possui mais IIPs do que necessário para escrevê-la de forma mínima. Por exemplo, a função $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,1,5,7)$ cujo mapa-K está mostrado na Fig. 9.9a possui três IIPs: $x_2{+}x_1$ (em vermelho na Fig. 9.9a), $\overline{x_2}{+}\overline{x_0}$ (em verde na Fig. 9.9a) e $x_1{+}\overline{x_0}$ (em amarelo Fig. 9.9a). O produto mínimo composto por todos os IIPs seria:

$$f(x_2,x_1,x_0) = (x_2{+}x_1) \cdot (\overline{x_2}{+}\overline{x_0}) \cdot (x_1{+}\overline{x_0}). \tag{9.10}$$

No entanto, usando os teoremas e postulados da álgebra de boole aplicados a (9.10) vem:

$$\begin{aligned}
f(x_2,x_1,x_0) &= (x_2{+}x_1) \cdot (\overline{x_2}{+}\overline{x_0}) \cdot (x_1{+}\overline{x_0}).\\
&= (x_2{+}x_1) \cdot (\overline{x_2}{+}\overline{x_0}) \cdot (x_1{+}\overline{x_0}+0).\\
&= (x_2{+}x_1) \cdot (\overline{x_2}{+}\overline{x_0}) \cdot (x_1{+}\overline{x_0}+x_2\cdot\overline{x_2}).\\
&= (x_2{+}x_1) \cdot (\overline{x_2}{+}\overline{x_0}) \cdot (x_2{+}x_1{+}\overline{x_0}) \cdot (\overline{x_2}{+}x_1{+}\overline{x_0})\\
&= (x_2{+}x_1) \cdot (x_2{+}x_1{+}\overline{x_0}) \cdot (\overline{x_2}{+}\overline{x_0}) \cdot (\overline{x_2}{+}x_1{+}\overline{x_0})\\
&= (x_2{+}x_1) \cdot (\overline{x_2}{+}\overline{x_0}).
\end{aligned} \tag{9.11}$$

Ou seja, apenas os IIPs $x_2{+}x_1$ e $\overline{x_2}{+}\overline{x_0}$ devem ser incluídos no produto mínimo enquanto o IIP $x_1{+}\overline{x_0}$ deve ser descartado. Isso acontece porque o IIP $x_1{+}\overline{x_0}$ "cobre" dois 0s que já estão cobertos pelos IIPs $x_2{+}x_1$ e $\overline{x_2}{+}\overline{x_0}$, o que pode ser visto claramente nos agrupamentos mostrados na Fig. 9.9a. A definição formal para cobertura de 0s por parte de implicados é dada na Definição 9.2.4.

Definição 9.2.4 (Cobertura de um 0 de uma função f) Considere a função f definida por seu conjunto um, $f = \text{Conjunto-ZERO}(u_0,u_1,\ldots u_{m-1})$ (refira-se à notação apresentadas na Seção **??**). Se a função f for escrita na forma de produto de implicados s_k, *i.e.* $f = \prod s_k$, diz-se que o 0 u_m está coberto no produto $\prod s_k$ se existir pelo menos um implicado s_k em $\prod s_k$ tal que $s_k(u_m) = 0$.

A pergunta que surge agora é como escolher os implicados de uma função para escrevê-la de forma miníma? Primeiro passo é adicionar ao produto mínimo os implicados fundamentais para compô-lo. Continuando com o exemplo da $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,1,5,7)$ (mostrada na Fig. 9.9a) observe que os IIPs $x_2{+}x_1$ e $\overline{x_2}{+}\overline{x_0}$ são especiais: O único IIP que cobre o 0 em

$x_2x_1x_0 = 000$ é o x_2+x_1, enquanto o único IIP que cobre o 0 em $x_2x_1x_0 = 111$ é o $\overline{x_2}+\overline{x_0}$. Esses IIPs têm um nome especial, são chamados de implicados primos essenciais. A Definição 9.2.5 os define.

Definição 9.2.5 (Implicado primo essencial de uma função) Um implicado primo s da função f é dito implicado primo essencial (IIPE) se existir pelo menos uma atribuição v de variáveis que torna $s = 0$ e essa mesma atribuição v não torna nenhum outro implicado primo s' ($s \neq p'$) de f igual a 0. Matematicamente, s é um implicado primo essencial de f se existe v tal que $s(v) = 0$ e $s'(v) = 1$ para todos os outros implicados primos s' de f.

O Exemplo 9.11 mostra mais exemplos de IIPEs. Como todos os 0s de uma função tem que ser cobertos por pelo menos um dos implicados primos do produto mínimo, os IIPEs devem ser sempre incluídos no produto mínimo conforme estabelecido no Teorema 9.2.2.

■ **Exemplo 9.11** A Fig. 9.10a mostra os IIP da função $f(x_2,x_1,x_0) =$ Conjunto-ZERO$(0,3,4,6,7)$, alguns são IIPEs e outros não. O agrupamento marcado em vermelho representa um IIPE de f pois ele é o único IIP que cobre o 0 em $x_2x_1x_0 = 000$. O mesmo ocorre com o agrupamento marcado em verde, pois ele é o único IIP que cobre o 0 em $x_2x_1x_0 = 011$ sendo assim um IIPE de f. Já os IIPs em amarelo e azul possuem todos os 0s cobertos por outros IIPs não sendo portanto IIPEs. A Fig. 9.10b mostra os agrupamentos relativos aos IIPE da função $f(x_3,x_2,x_1,x_0) =$ Conjunto-ZERO$(0,1,2,3,5,7,12,8)$.

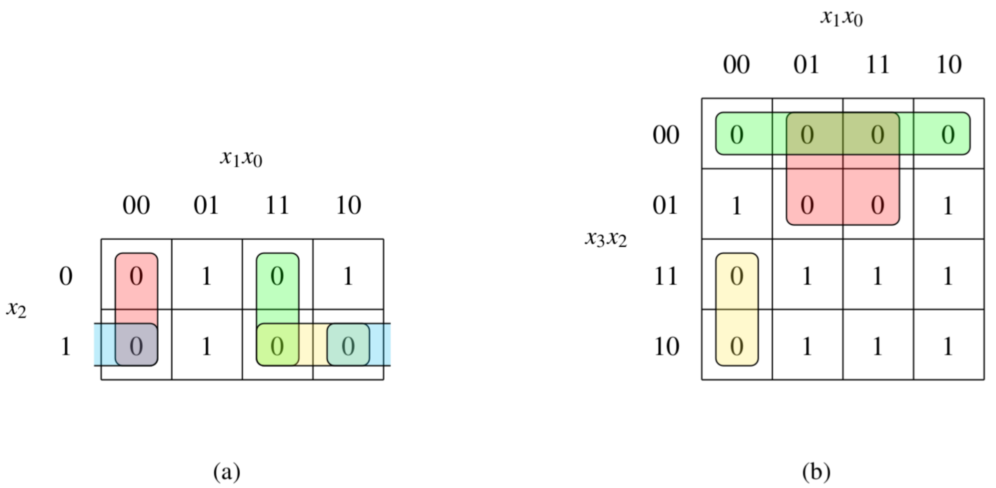


Figura 9.10: Exemplos de implicado primos essenciais.

■

Teorema 9.2.2 (Inclusão de IIPEs no produto mínimo) Todo produto mínimo de uma função na forma PDS deve incluir todos os IIPEs da função.

Ou seja, o produto mínimo será composto primeiramente pelos IIPEs. Caso os IIPEs não cubram todos os 0s do Conjunto-UM da função, um conjunto mínimo de IIPs devem ser incluídos no produto mínimo. Somente devem ser incluídos IIPs no produto mínimo se eles cobrirem 0s ainda não cobertos pelos outros IIPs/IIPEs já incluídos.

O procedimento mostrado no Algoritmo 3 resume como se obter a expressão mínima na forma produto de somas.

Algorithm 3 Minimização na forma produto de somas.

```
Dado: Conjunto-ZERO() de uma função de chaveamento f de n variáveis
Determine todos os implicados primos da função f
Dentre os implicados primos determine os implicados primos essenciais de f
Inclua todos os implicados primos essenciais no produto de somas.
se (Ainda há 0s não cobertos pelo IIPEs) então
    Escolha dentre os implicados primos restantes o conjunto mínimo que cubra todos os 0s
    faltantes.
fim se
retorne A produto mínima para f na forma SPD.
```

9.2.2 Procedimento de minimização usando mapas-K.

Para facilitar a aplicação do procedimento de minimização mostrado no Algoritmo 3 aos mapas-K é possível se estabelecer algumas regras que devem ser seguidas e observadas simultaneamente no processo de agrupamento de 0s nos mapas-K:

1. Um novo agrupamento de 0s só deve ser feito se ele contiver 0s ainda não agrupados.
2. Sobreposição parcial de agrupamentos são permitidas.
3. O mesmo 0 pode ser agrupado várias vezes no mapa-K.
4. Os maiores agrupamentos possíveis de 2^n 0s vizinhos devem ser feitos sempre que possível.
5. A menor quantidade possível de grupos deve ser formada.
6. Todos os 0s devem estar agrupados pelo menos uma vez.
7. Os 1s não devem ser agrupados.

Perceba que todas as regras tem que ser obedecidas simultaneamente. Não é possível aplicar uma das regras se ela viola outra regra. No caso de um mapa-K de 4 variáveis pode ser aplicado o Algoritmo 4 para a obtenção dos agrupamentos de 0s.

Algorithm 4 Agrupamentos de 0s em mapas-K de 4 variáveis.

```
Dado: Conjunto-ZERO() de uma função de chaveamento f de n variáveis
Preencha o mapa-K com os 1s e 0s correspondentes Conjunto-ZERO() da função f.
Agrupe os 0s isolados no mapa-K;
Agrupe os 0s que só possuem um único vizinho no mapa-K;
para i = 0 até n − 2 faça
    enquanto (Houver 0s ainda não agrupados que formem grupos de 2^(n−i) 0s) faça
        Faça um agrupamento de 2^(n−i) 0s contendo pelo menos um 0 ainda não agrupado.
    fim enquanto
fim para
Escreva o termo soma s_k de cada agrupamento feito no mapa-K.
retorne O produto mínimo para f na forma PDS em que f = ∏ s_k (∏ representa operações
AND)
```

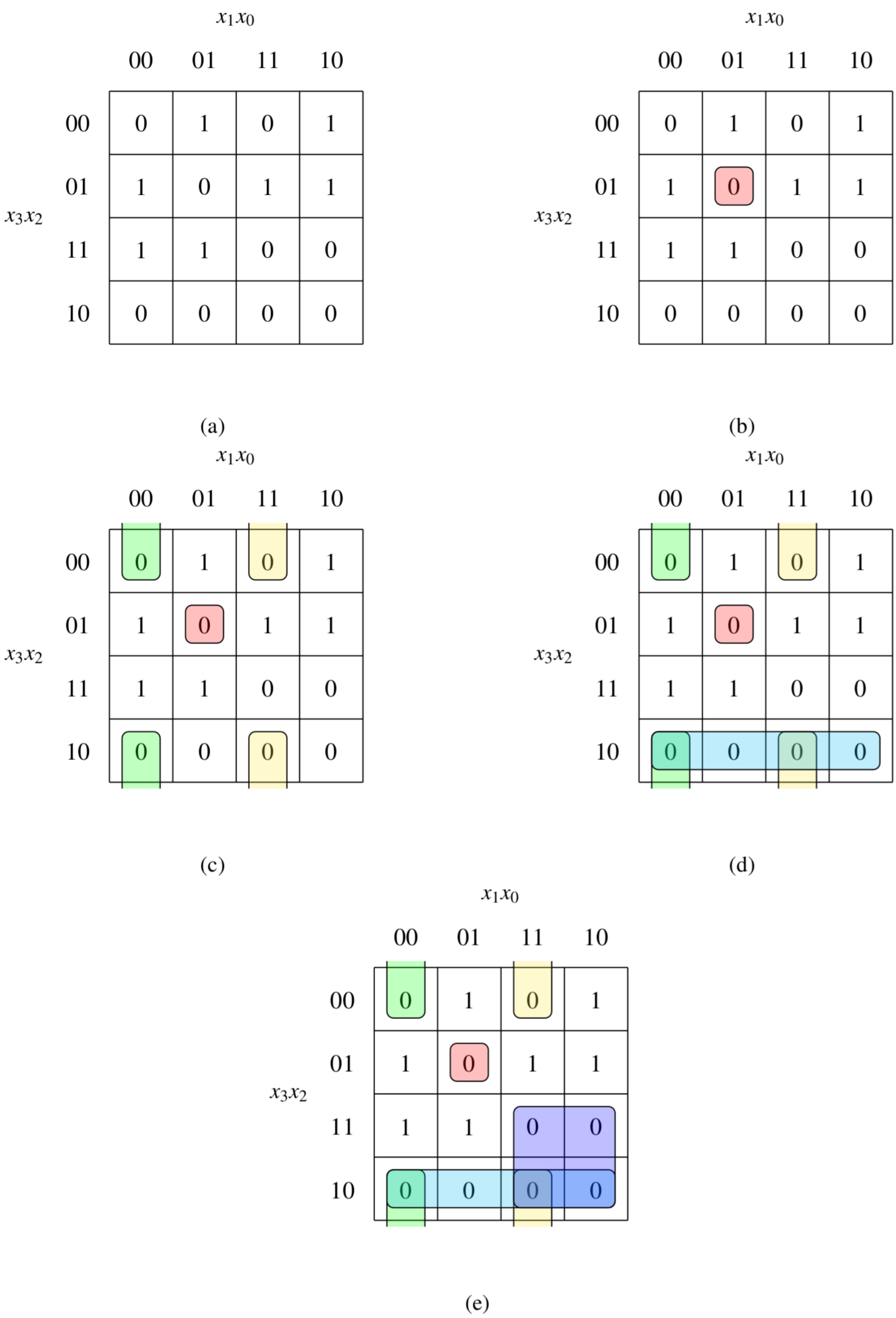


Figura 9.11: Aplicação do Algoritmo 4 na obtenção dos agrupamentos de 0s que formam a expressão mínima na forma PDS da função $f(x_3,x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,3,5,8,9,10,11,14,15)$.

A Fig. 9.11a mostra o passo a passo da aplicação do Algoritmo 4 a uma função de chaveamento de 4 variáveis: $f(x_3,x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,3,5,8,9,10,11,14,15)$. Após a realização do passo 1 do Algoritmo 4 se chega ao resultado mostrado em na Fig. 9.11a. Após procurar por os isolados no mapa-K (linha 2) encontra-se como resultado o 0 destacado na Fig. 9.11b. Na linha 3, procura-se por 0s que possuem um único vizinho no mapa-K. Após a aplicação desse passo chega-se aos agrupamentos mostrados em verde e amarelo (os 0s em 0000 e em 0011 possuem um único vizinho) na Fig. 9.11b.

Entra-se no laço para da linha 4. Ainda há 0s não agrupados no mapa-K mas não há agrupamentos de 2^{n-i} ($2^{4-0} = 16$) 0s. Incrementa-se o valor de i do laço. Novamente ainda há 0s não agrupados no mapa-K mas não há agrupamentos de 2^{n-i} ($2^{4-1} = 8$) 0s no mapa-K. Incrementa-se o valor de i do laço. Ainda há 0s não agrupados no mapa-K mas agora há dois agrupamentos de 0s contendo 2^{n-i} ($2^{4-2} = 4$) possíveis no mapa-K. Aplica-se o passo da linha 6 e se faz o agrupamento mostrado em azul na Fig. 9.11d. Novamente se verifica a condição do enquanto da linha 5 pois ainda há 0s não agrupados no mapa-K e um agrupamento de 0s contendo 2^{n-i} ($2^{4-2} = 4$) 0s. Aplica-se o passo da linha 6 e se faz o agrupamento mostrado em roxo na Fig. 9.11e.

Como todos os 0s foram agrupados o Algoritmo 4 não tenta fazer mais nenhum agrupamento indo para o passo 9. Aplicando-se o passo 9 chega-se aos termos produtos, $s_1 = x_3+\overline{x_2}+x_1+\overline{x_0}$ (vermelho), $s_2 = x_2+x_1+x_0$ (verde), $s_3 = x_2+\overline{x_1}+\overline{x_0}$ (amarelo), $s_4 = \overline{x_3}+x_2$ (azul), $s_5 = \overline{x_3}+\overline{x_1}$ (roxo) e portanto o produto mínimo para f na forma PDS é:

$$f(x_3,x_2,x_1,x_0) = \prod s_k = (x_3+\overline{x_2}+x_1+\overline{x_0}) \cdot (x_2+x_1+x_0) \cdot (x_2+\overline{x_1}+\overline{x_0}) \cdot (\overline{x_3}+x_2) \cdot (\overline{x_3}+\overline{x_1}).$$

9.2.3 Exemplos de implementação de sistemas mínimos usando mapas-K

Nesta seção são mostrados a implementação mínima para alguns sistemas exemplo usando a metodologia de mapas-K.

■ **Exemplo 9.12** (Implementação mínima de um sistema de três variáveis)

Um sistema combinacional S com três entradas e uma saída é descrito por meio da função de chaveamento $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,2,3,4,5)$. Encontre a implementação mínima para esse sistema na forma de produto de somas.

A tabela verdade desse sistema está mostrada na Fig. 9.12a. Preenche-se o mapa-K usando a tabela verdade. O mapa-K preenchido para a função está na Fig. 9.12b. Aplicando a teoria vista na Seção **??** e as regras da Seção 9.2.2 chega-se aos agrupamentos mostrados na Fig. 9.12b.

Os agrupamentos mostrados têm as seguintes expressões: x_1+x_0 (vermelho), $\overline{x_2}+x_1$ (verde), $x_2+\overline{x_1}$ (amarelo). A implementação mínima para esse sistema é portanto dada por:

$$f(x_2,x_1,x_0) = (x_1+x_0) \cdot (\overline{x_2}+x_1) \cdot (x_2+\overline{x_1})$$

■

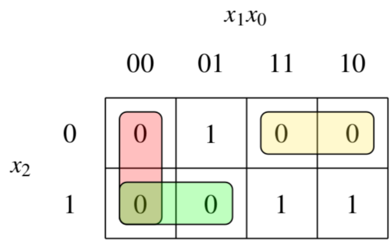
9.2.4 Implementação de sistemas mínimos usando mapas-K com funções não completamente especificadas.

No caso de uma função ser não completamente especificada. Os valores não especificados devem ser identificados e preenchidos no mapa-K com um "x" (ou "-"). Todo o procedimento de simplificação aplicado aos mapas-K pode ser aplicado nesse contexto com duas diferenças:

- Os "x" (ou "-") podem ou não serem agrupados como se fossem 0s.

i	x_2	x_1	x_0	$f(x_2,x_1,x_0)$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

(a)



(b)

Figura 9.12: Tabelas usadas na obtenção da expressão mínima da função do Exemplo 9.12

- Os "x" (ou "-") devem ser agrupados quando, ao serem agrupados, gerem agrupamentos maiores (resultando em mais variáveis eliminadas).
- Não é obrigatório o agrupamento de todos os "x" (ou "-") como é obrigatório o agrupamento de todos os 0s.

O Exemplo 9.13 mostra uma função não completamente especificada sendo implementada por meio de mapas-K.

Exemplo 9.13 (Detector de primalidade)

Esse sistema é o um sistema que recebe um dígito BCD em sua entradas (4 *bits* de entrada) e fornece na sua saída (1 *bit* de saída) 0 se o dígito da entrada for um número primo e 1 caso contrário.

As entradas do sistema são os *bits* x_3, x_2, x_1 e x_0 e a saída o *bit* y. Como na entrada se tem 4 bits e um dígito BCD, apenas os número de 0 a 9 são permitidos. Portanto, os números de 10 a 15 apesar de serem padrões de entrada possíveis (pois há quatro *bits* na entrada) são padrões de entrada que nunca serão apresentados ao sistemas constituindo assim padrões aos quais pode se atribuir saídas não especificadas. A Tabela 9.9 mostra a tabela verdade completa desse sistema.

Preenchendo o mapa-K a partir da tabela verdade chega-se ao mapa-K mostrado na Fig. 9.13.

Note que a expressão mínima pode ser obtida conforme os agrupamentos mostrados na Fig. 9.13. Os dois quartetos agrupados incluiram duas saídas "-"(indeterminadas). Escrevendo a expressão mínima de implementação desse sistema vem:

$$y = (\overline{x_2}+\overline{x_0})\cdot(x_2+\overline{x_1}) \qquad (9.12)$$

■

Tabela 9.9: Tabela verdade do detector de primalidade de para dígitos BCD.

i	x_3	x_2	x_1	x_0	y
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	-
11	1	0	1	1	-
12	1	1	0	0	-
13	1	1	0	1	-
14	1	1	1	0	-
15	1	1	1	1	-

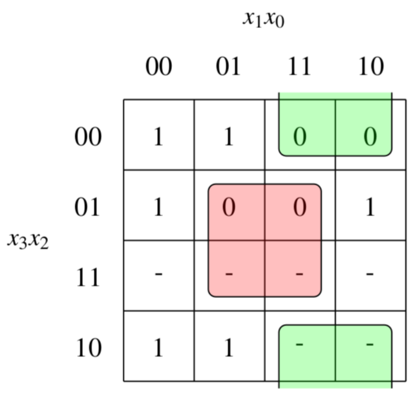


Figura 9.13: Mapa-K do detector de primalidade de para dígitos BCD.

9.3 Problemas propostos

Problema 9.1 Preencha os mapas-K das funções dadas e determine seus implicantes primos e implicantes primos essenciais.

a) $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,2,6)$
b) $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,3,5,6)$
c) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(1,2,5,9,13,15)$
d) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(4,5,7,9,10,12)$

Problema 9.2 Preencha os mapas-K das funções dadas e determine seus implicados primos e implicados primos essenciais.

a) $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,2,3,5,6)$
b) $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(1,2,7)$
c) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-ZERO}(9,10,11,14,15)$
d) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,3,8,10,13,15)$

Problema 9.3 Aplique a técnica de minimização usando mapas-K às função nos itens abaixo e obtenha a expressão mínima na forma SDP e a sua implementação usando portas lógicas (esquematize o circuito).

a) $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,3,4,5,7)$
b) $f(x_2,x_1,x_0) = \text{Conjunto-UM}(0,2,3,4,7)$
c) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,2,7,8,11,14)$
d) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(0,2,6,7,8,10,11,12,15)$
e) $f(x_2,x_1,x_0) = \sum m(1,3,5)$ e $\sum dc(0,4)$
f) $f(x_2,x_1,x_0) = \sum m(2,3,6)$ e $\sum dc(0,4)$
g) $f(x_3,x_2,x_1,x_0) = \sum m(5,7,10)$ e $\sum dc(0,2,13,14,15)$
h) $f(x_3,x_2,x_1,x_0) = \sum m(9,11,12,13,15)$ e $\sum dc(0,1,2,3)$

Problema 9.4 Aplique a técnica de minimização usando mapas-K às função nos itens abaixo e obtenha a expressão mínima na forma PDS e a sua implementação usando portas lógicas (esquematize o circuito).

a) $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,3,5,6,7)$
b) $f(x_2,x_1,x_0) = \text{Conjunto-ZERO}(0,2,3,5,6)$
c) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-ZERO}(3,4,7,9,10,12,14,15)$
d) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-ZERO}(1,3,6,7,8,12,13,15)$
e) $f(x_2,x_1,x_0) = \prod M(1,3,3)$ e $\prod dc(0,7)$
f) $f(x_2,x_1,x_0) = \prod M(0,1,3,4)$ e $\prod dc(7)$
g) $f(x_3,x_2,x_1,x_0) = \prod M(1,4,10,11,12,13)$ e $\prod dc(8,9,14,15)$

Problema 9.5 Aplique a técnica de minimização Quine McCluskey às função nos itens abaixo e obtenha a expressão mínima na forma SDP e a sua implementação usando portas lógicas (esquematize o circuito). Liste os IPs e os IPEs encontrados

a) $f(x_2,x_1,x_0) = \text{Conjunto-UM}(1,2,4,6,7)$
b) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(0,1,3,6,10,11,12,13,15)$
c) $f(x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(0,2,3,4,5,6,7,8,9,10,11,14)$
d) $f(x_4,x_3,x_2,x_1,x_0) = \text{Conjunto-UM}(6,7,8,11,12,15,18,21,22,28,29)$

e) $f(x_5, x_4, x_3, x_2, x_1, x_0) = \text{Conjunto-UM}(8, 19, 28, 34, 35, 37, 42, 47, 52, 53, 60, 61, 63)$

Problema 9.6 Em uma determinada empresa as decisões são tomadas por um conselho composto por quatro conselheiros (C_1, C_2, C_3 e C_4). Para cada questão levantada nas reuniões do conselho, cada conselheiro deve se manifestar exclusivamente de duas formas: a favor ou contra. Um engenheiro foi contratado para implementar um sistema de votação eletrônica na sala onde as reuniões acontecem. O sistema deve ter um botão de votação para cada conselheiro e deve mostrar o resultado final da votação por meio de dois LEDs, um para indicar se a maioria dos conselheiros votaram a favor e outro para mostrar se a maioria dos conselheiros votaram contra. Em caso de empates o desempate será dado pelo voto do conselheiro C_1. Utilize mapas-K para projetar e implementar um sistema digital mínimo capaz de implementar o sistema de votação especificado.

Problema 9.7 Projete e implemente uma rede de dois níveis utilizando mapas K um circuito digital que atenda às seguintes especificações:

* Entrada: um digito BCD codificado em binário.
* Saídas: transformar o inteiro presente na entrada para um número de três dígitos na base 3. Dois bits devem ser usados para representar os dois algarismos menos significativos e deve ser usado apenas um bit para o algarismos mais significativo.
* Lembre que a implementação mínima deve ser fornecida.