

# Unidade III

## 7 CIRCUITOS CODIFICADORES E DECODIFICADORES

### 7.1 Codificadores e decodificadores

Sistemas digitais utilizam-se de valores lógicos para armazenarem informações. A informação, para ser armazenada, deve ser codificada, isto é, convertida em um conjunto de valores lógicos 0 e 1. De forma análoga, tem de ser possível também realizar a operação inversa, ou seja, decodificar os valores lógicos, devolvendo seus valores numéricos correspondentes no sistema decimal. Codificadores e decodificadores são um importante grupo entre os circuitos combinacionais.

Um código é uma forma de transmitir uma ideia utilizando-se de outra linguagem, simbologia ou notação. Assim, a conversão de uma base numérica para outra, como visto no tópico 1, pode ser considerada uma codificação. Do mesmo modo, o *display* de sete segmentos (SSD) visto no tópico 6 também pode ser considerado um codificador, pois ele transforma a informação contida em um conjunto de *bits* para uma forma que possa ser exibida pelo acionamento de elementos luminosos.

De maneira semelhante, teclados de diversos dispositivos precisam codificar para *bits* o correspondente a cada tecla digitada, para que tal informação seja armazenada e processada, e ela precisará ser decodificada após esse processamento para ser devolvida ao usuário de modo que seja compreensível.

Neste tópico serão apresentados os quatro principais códigos utilizados computacionalmente: **BCD 8421, BCH, Excesso 3 e código de Gray**. Também será vista outra abordagem para o *display* de sete segmentos (SSD), na qual será tratado como um decodificador, sendo suas saídas geradas por um valor codificado pelo código BCD 8421.

Cada código terá as regras por trás da codificação apresentada, e serão desenvolvidos os respectivos circuitos codificadores e decodificadores. Todos os códigos que forem apresentados convertem informações para um conjunto de 4 *bits*, sendo que cada *bit* corresponde à saída de um circuito lógico. Para os circuitos decodificadores, quando mapas de Karnaugh se fizerem necessários, no caso dos códigos BCD 8421, Excesso 3 e o SSD, apenas os mapas e a respectiva expressão final já simplificada serão apresentados, não sendo feita a simplificação passo a passo como nos tópicos anteriores.



#### Lembrete

Para um conjunto de 4 *bits*, até  $2^4 = 16$  valores numéricos diferentes podem ser representados.

### 7.2 Código BCD 8421

O código BCD 8421 (do inglês *Binary Coded Decimal*, Decimal Codificado em Binário) é um dos códigos mais utilizados e codifica os valores numéricos de 0 a 9 diretamente para um valor binário composto de 4 *bits*; cada um representa uma potência de 2 (8, 4, 2 e 1, o que dá o nome ao código). As saídas serão denominadas canais e receberão a numeração do seu valor correspondente. A tabela a seguir apresenta as saídas dos quatro circuitos desse codificador.

**Tabela 59 – Código BCD 8421**

Decimal	BCD			
	Canal 8	Canal 4	Canal 2	Canal 1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

#### Codificador BCD 8421

Considerando que cada um dos dez valores é uma entrada isolada, quando uma entrada for igual a 1, automaticamente todas as outras serão iguais a 0, e a construção das saídas é facilmente obtida a partir da tabela, sem a necessidade de executar o mapa de Karnaugh:

$$\text{Canal 8} = E8 + E9$$

$$\text{Canal 4} = E4 + E5 + E6 + E7$$

$$\text{Canal 2} = E2 + E3 + E6 + E7$$

$$\text{Canal 1} = E1 + E3 + E5 + E7 + E9$$

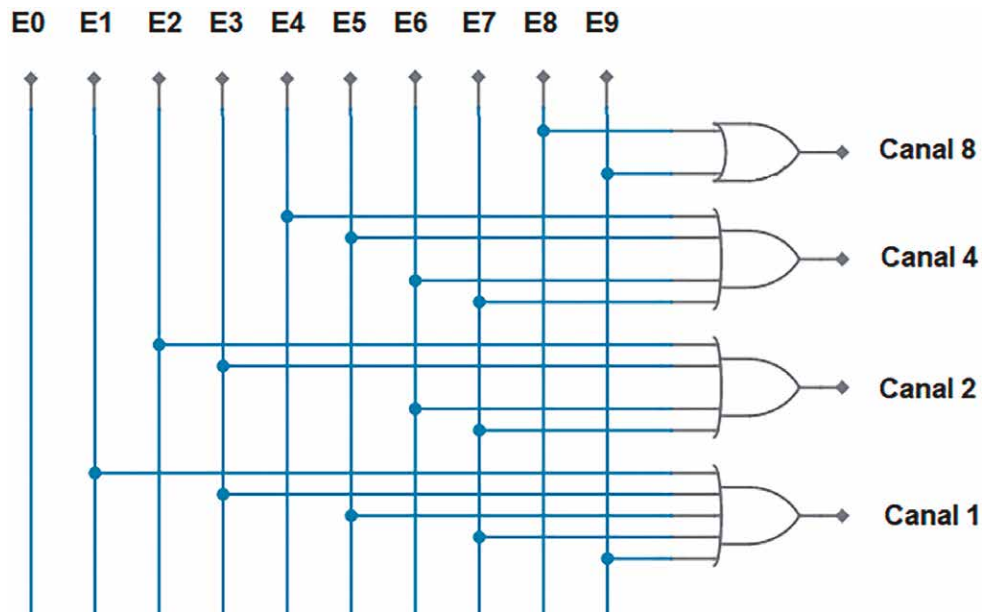


Figura 91 – Circuito codificador BCD 8421

## Decodificador BCD 8421

Para construir um decodificador, as entradas (E0 a E9) se tornarão as saídas, S0 a S9, respectivamente, uma vez que receberemos quatro entradas correspondentes a cada um dos *bits* de um número binário. Assim, as saídas do codificador se tornarão a entrada do decodificador.



### Observação

Para todos os códigos apresentados, as saídas do codificador (C1, C2, C4 e C8) serão as entradas do decodificador, e vice-versa.

Em primeiro lugar, é necessário construir a tabela verdade, tornando os canais 8421 em entradas e os valores de 0 a 9 em saídas:

Tabela 60 – Decodificador para o código BCD 8421

Entradas				Saídas									
Canal 8	Canal 4	Canal 2	Canal 1	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0

Entradas				Saídas									
Canal 8	Canal 4	Canal 2	Canal 1	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Ao construir os mapas de Karnaugh para cada saída, podemos considerar cada uma das seis combinações de entradas ausentes como irrelevantes: elas assumirão os valores lógicos 1 ou 0 conforme seja conveniente no mapa. A seguir, são apresentados os mapas de Karnaugh e as expressões resultantes para cada uma das dez saídas:

		C8C4			
		00	01	11	10
C2C1	00	0 <sup>0</sup>	4 <sup>0</sup>	12 <sup>X</sup>	8 <sup>0</sup>
	01	1 <sup>0</sup>	5 <sup>0</sup>	13 <sup>X</sup>	9 <sup>0</sup>
	11	3 <sup>0</sup>	7 <sup>0</sup>	15 <sup>X</sup>	11 <sup>X</sup>
	10	2 <sup>0</sup>	6 <sup>0</sup>	14 <sup>X</sup>	10 <sup>X</sup>

Figura 92 –  $S0 = \overline{C8 + C4 + C2 + C1}$

		C8C4			
		00	01	11	10
C2C1	00	0 <sup>0</sup>	4 <sup>0</sup>	12 <sup>X</sup>	8 <sup>0</sup>
	01	1 <sup>1</sup>	5 <sup>0</sup>	13 <sup>X</sup>	9 <sup>0</sup>
	11	3 <sup>0</sup>	7 <sup>0</sup>	15 <sup>X</sup>	11 <sup>X</sup>
	10	2 <sup>0</sup>	6 <sup>0</sup>	14 <sup>X</sup>	10 <sup>X</sup>

Figura 93 –  $S1 = \overline{(C8 + C4 + C2)} \cdot C1$

		C8C4			
		00	01	11	10
C2C1	00	0 <sup>0</sup>	4 <sup>0</sup>	12 <sup>X</sup>	8 <sup>0</sup>
	01	1 <sup>0</sup>	5 <sup>0</sup>	13 <sup>X</sup>	9 <sup>0</sup>
	11	3 <sup>0</sup>	7 <sup>0</sup>	15 <sup>X</sup>	11 <sup>X</sup>
	10	2 <sup>1</sup>	6 <sup>0</sup>	14 <sup>X</sup>	10 <sup>X</sup>

Figura 94 –  $S2 = \overline{(C4 + C1)} \cdot C2$

		C8C4			
		00	01	11	10
C2C1	00	0 <sup>0</sup>	4 <sup>0</sup>	12 <sup>X</sup>	8 <sup>0</sup>
	01	1 <sup>0</sup>	5 <sup>0</sup>	13 <sup>X</sup>	9 <sup>0</sup>
	11	3 <sup>1</sup>	7 <sup>0</sup>	15 <sup>X</sup>	11 <sup>X</sup>
	10	2 <sup>0</sup>	6 <sup>0</sup>	14 <sup>X</sup>	10 <sup>X</sup>

Figura 95 –  $S3 = \overline{C4} \cdot C2 \cdot C1$

		C8C4			
		00	01	11	10
C2C1	00	0	1	X	0
	01	0	0	X	0
	11	0	0	X	X
	10	0	0	X	X

Figura 96 –  $S4 = \overline{(C2 + C1)} \cdot C4$

		C8C4			
		00	01	11	10
C2C1	00	0	0	X	0
	01	0	1	X	0
	11	0	0	X	X
	10	0	0	X	X

Figura 97 –  $S5 = C4 \cdot \overline{C2} \cdot C1$

		C8C4			
		00	01	11	10
C2C1	00	0	0	X	0
	01	0	0	X	0
	11	0	0	X	X
	10	0	1	X	X

Figura 98 –  $S6 = C4 \cdot C2 \cdot \overline{C1}$

		C8C4			
		00	01	11	10
C2C1	00	0	0	X	0
	01	0	0	X	0
	11	0	1	X	X
	10	0	0	X	X

Figura 99 –  $S7 = C4 \cdot C2 \cdot C1$

		C8C4			
		00	01	11	10
C2C1	00	0	0	X	1
	01	0	0	X	0
	11	0	0	X	X
	10	0	0	X	X

Figura 100 –  $S8 = C8 \cdot \overline{C1}$

		C8C4			
		00	01	11	10
C2C1	00	0	0	X	0
	01	0	0	X	1
	11	0	0	X	X
	10	0	0	X	X

Figura 101 –  $S9 = C8 \cdot C1$

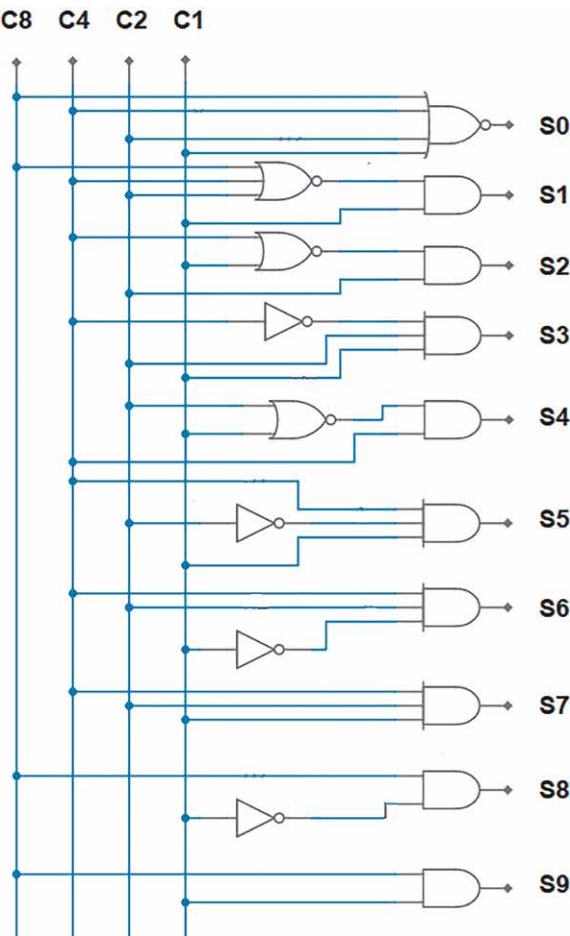


Figura 102 – Circuito decodificador BCD 8421

7.3 Código BCH

O Código BCH (do inglês *Binary Coded Hexadecimal*, Hexadecimal Codificado em Binário) é muito semelhante ao código BCD, mas é utilizado para codificar os 16 algarismos do sistema hexadecimal para o sistema binário. A tabela a seguir apresenta as quatro saídas dos circuitos desse codificador.

Tabela 61 – Código BCH

Decimal	BCH			
	Canal 8	Canal 4	Canal 2	Canal 1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0

Decimal	BCH			
	Canal 8	Canal 4	Canal 2	Canal 1
7	0	1	1	1
8	1	0	0	0
9	1	0	0	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

## Codificador BCH

Como para o BCD 8421, cada uma das entradas é isolada, e assim também podemos obter as saídas diretamente a partir da tabela:

$$\text{Canal 8} = E8 + E9 + Ea + Eb + Ec + Ed + Ee + Ef$$

$$\text{Canal 4} = E4 + E5 + E6 + E7 + Ec + Ed + Ee + Ef$$

$$\text{Canal 2} = E2 + E3 + E6 + E7 + Ea + Eb + Ee + Ef$$

$$\text{Canal 1} = E1 + E3 + E5 + E7 + E9 + Eb + Ed + Ef$$

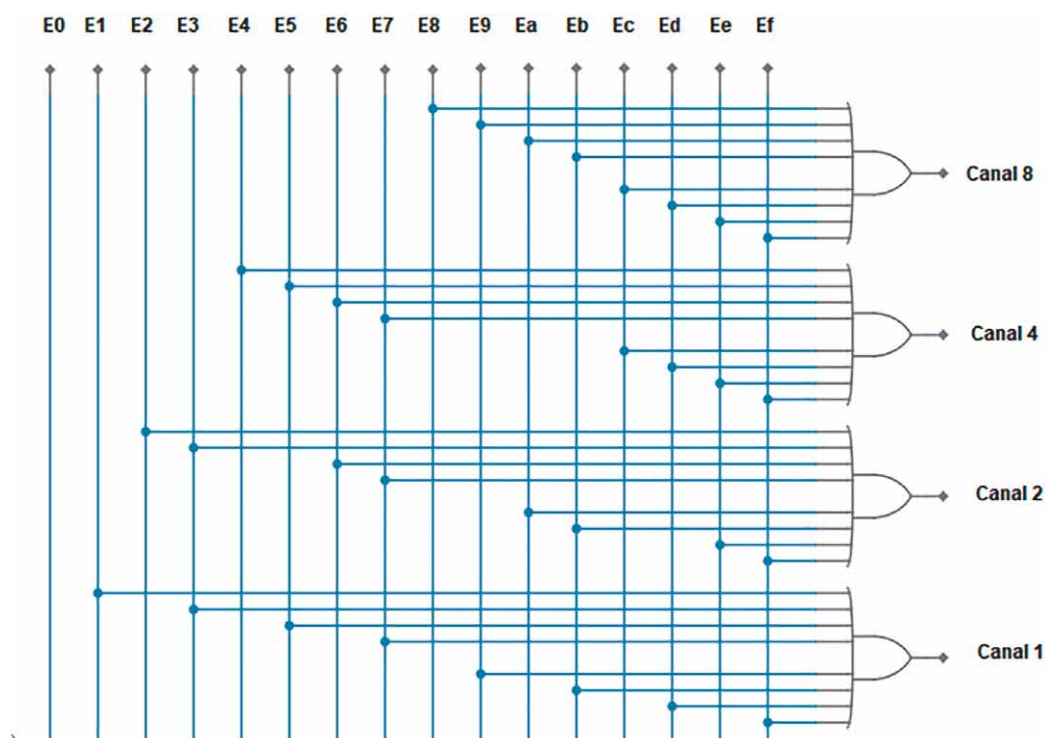


Figura 103 – Circuito codificador BCH

## Decodificador BCH

Construiremos a tabela verdade, tornando os canais 8421 em entradas e os valores de 0 a F em saídas. Para maior clareza, a tabela foi dividida em duas partes:

**Tabela 62 – Decodificador para o código BCH: (a) saídas S0 a S7; (b) saídas S8 a Sf**

(a)

Entradas				Saídas							
Canal 8	Canal 4	Canal 2	Canal 1	S0	S1	S2	S3	S4	S5	S6	S7
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	0	0	0	1

(b)

**Tabela 63**

Entradas				Saídas							
Canal 8	Canal 4	Canal 2	Canal 1	S8	S9	Sa	Sb	Sc	Sd	Se	Sf
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

Como para cada uma das combinações das entradas existe apenas uma saída verdadeira, não há necessidade de se construir os mapas de Karnaugh. As expressões lógicas de cada saída, com suas respectivas simplificações, utilizando-se a lei de De Morgan para agrupar conjuntos de negações sob uma porta NOR, são indicadas a seguir:



$$S0 = \overline{(C8 + C4 + C2 + C1)}$$

$$S1 = \overline{(C8 + C4 + C2)} \cdot C1$$

$$S2 = \overline{(C8 + C4 + C1)} \cdot C2$$

$$S3 = \overline{(C8 + C4)} \cdot C2 \cdot C1$$

$$S4 = \overline{(C8 + C2 + C1)} \cdot C4$$

$$S5 = \overline{(C8 + C2)} \cdot C4 \cdot C1$$

$$S6 = \overline{(C8 + C1)} \cdot C4 \cdot C2$$

$$S7 = \overline{C8} \cdot C4 \cdot C2 \cdot C1$$

$$S8 = \overline{(C4 + C2 + C1)} \cdot C8$$

$$S9 = \overline{(C4 + C2)} \cdot C8 \cdot C1$$

$$Sa = \overline{(C4 + C1)} \cdot C8 \cdot C2$$

$$Sb = \overline{C4} \cdot C8 \cdot C2 \cdot C1$$

$$Sc = \overline{(C2 + C1)} \cdot C8 \cdot C4$$

$$Sd = \overline{C2} \cdot C8 \cdot C4 \cdot C1$$

$$Se = \overline{C1} \cdot C8 \cdot C4 \cdot C2$$

$$Sf = C8 \cdot C4 \cdot C2 \cdot C1$$

Para facilitar a visualização, o circuito decodificador BCH também será dividido em duas partes, uma com as saídas de S0 a S7 e outra com as saídas de S8 a Sf:

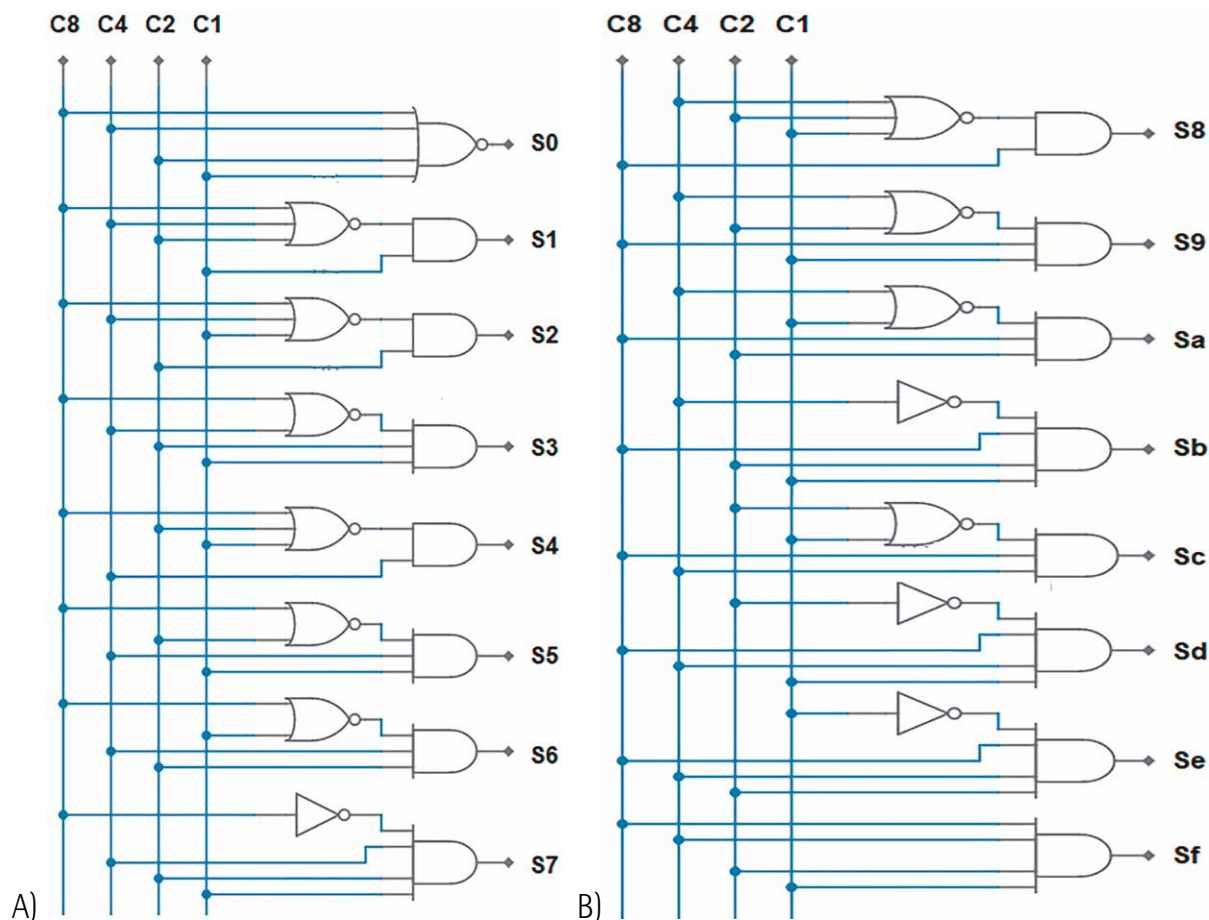


Figura 104 – Circuito decodificador BCH: (a) saídas S0 a S7; (b) saídas S8 a Sf

## 7.4 Código Excesso 3

Após a transcrição binária direta, os primeiros códigos de importância foram os chamados **códigos de Excesso de N**. Este tipo de codificação era muito utilizado nos primeiros computadores e calculadoras, sendo a regra de formação destes códigos o acréscimo constante de um valor N ao valor binário. O código Excesso 3 ( $N = 3$ ) também é conhecido como código de Stibitz, em homenagem a seu criador George Stibitz (1904-1995), um dos pais da computação eletrônica.

O código Excesso 3 é muito semelhante ao código BCD, com a diferença que cada número é acrescido de 3 (0011 no sistema binário). Ele foi criado para facilitar as operações de subtração.

Tabela 64 – Código Excesso 3

Decimal	Excesso 3			
	Canal 8	Canal 4	Canal 2	Canal 1
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1

Decimal	Excesso 3			
	Canal 8	Canal 4	Canal 2	Canal 1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

## Codificador Excesso 3

O processo de obtenção das expressões referentes ao codificador é análogo àquele que foi utilizado para os códigos BCD 8421 e BCH: simplesmente conectam-se as entradas por meio de portas OR:

$$\text{Canal 8} = E5 + E6 + E7 + E8 + E9$$

$$\text{Canal 4} = E1 + E2 + E3 + E4 + E9$$

$$\text{Canal 2} = E0 + E3 + E4 + E7 + E8$$

$$\text{Canal 1} = E0 + E2 + E4 + E6 + E8$$

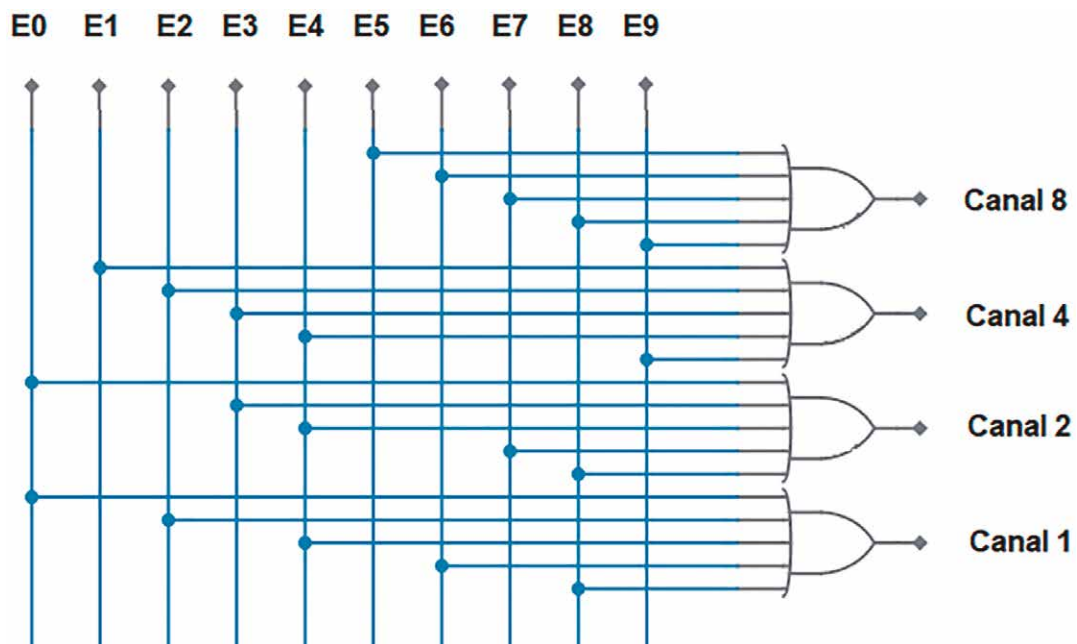


Figura 105 – Circuito codificador Excesso 3

## Decodificador Excesso 3

Para construir um decodificador, as entradas (E0 a E9) novamente se tornarão as saídas (S0 a S9, respectivamente), como feito para os decodificadores anteriores.

Assim, será construída a tabela verdade, tornando os canais 8420 em entradas e os valores de 0 a 9 em saídas, como mostrado na sequência.

**Tabela 65 – Decodificador para o código Excesso 3**

Entradas				Saídas									
Canal 8	Canal 4	Canal 2	Canal 1	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	1	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	0	0	0	0	0	1

A seguir, faremos o mapa de Karnaugh, considerando que as entradas de 0 a 2 e de 13 a 15, em binário, não são possíveis; sendo assim, tais valores são indicados como irrelevantes nos mapas:

		C8C4			
		00	01	11	10
C2C1	00	0 X	4 0	12 0	8 0
	01	1 X	5 0	13 X	9 0
	11	3 1	7 0	15 X	11 0
	10	2 X	6 0	14 X	10 0

Figura 106 –  $S0 = \overline{(C8 + C4)}$

		C8C4			
		00	01	11	10
C2C1	00	0 X	4 1	12 0	8 0
	01	1 X	5 0	13 X	9 0
	11	3 0	7 0	15 X	11 0
	10	2 X	6 0	14 X	10 0

Figura 107 –  $S1 = \overline{(C8 + C2 + C1)}$

		C8C4			
		00	01	11	10
C2C1	00	0 X	4 0	12 0	8 0
	01	1 X	5 1	13 X	9 0
	11	3 0	7 0	15 X	11 0
	10	2 X	6 0	14 X	10 0

Figura 108 –  $S2 = \overline{(C8 + C2)} \cdot C1$

		C8C4			
		00	01	11	10
C2C1	00	0 X	4 0	12 0	8 0
	01	1 X	5 0	13 X	9 0
	11	3 0	7 0	15 X	11 0
	10	2 X	6 1	14 X	10 0

Figura 109 –  $S3 = \overline{(C8 + C1)} \cdot C2$

		C8C4			
		00	01	11	10
C2C1	00	0 X	4 0	12 0	8 0
	01	1 X	5 0	13 X	9 0
	11	3 0	7 1	15 X	11 0
	10	2 X	6 0	14 X	10 0

Figura 110 –  $S4 = C4 \cdot C2 \cdot C1$

		C8C4			
		00	01	11	10
C2C1	00	0 X	4 0	12 0	8 1
	01	1 X	5 0	13 X	9 0
	11	3 0	7 0	15 X	11 0
	10	2 X	6 0	14 X	10 0

Figura 111 –  $S5 = \overline{(C4 + C2 + C1)}$

		C8C4			
		00	01	11	10
C2C1	00	0 X	4 0	12 0	8 0
	01	1 X	5 0	13 X	9 1
	11	3 0	7 0	15 X	11 0
	10	2 X	6 0	14 X	10 0

Figura 112 –  $S6 = C8 \cdot \overline{C2} \cdot C1$

		C8C4			
		00	01	11	10
C2C1	00	0 X	4 0	12 0	8 0
	01	1 X	5 0	13 X	9 0
	11	3 0	7 0	15 X	11 0
	10	2 X	6 0	14 X	10 1

Figura 113 –  $S7 = C8 \cdot C2 \cdot \overline{C1}$

		C8C4			
		00	01	11	10
C2C1	00	<sup>0</sup> X	<sup>4</sup> 0	<sup>12</sup> 0	<sup>8</sup> 0
	01	<sup>1</sup> X	<sup>5</sup> 0	<sup>13</sup> X	<sup>9</sup> 0
	11	<sup>3</sup> 0	<sup>7</sup> 0	<sup>15</sup> X	<sup>11</sup> 1
	10	<sup>2</sup> X	<sup>6</sup> 0	<sup>14</sup> X	<sup>10</sup> 0

Figura 114 -  $S8 = C8 \cdot C2 \cdot C1$

		C8C4			
		00	01	11	10
C2C1	00	<sup>0</sup> X	<sup>4</sup> 0	<sup>12</sup> 1	<sup>8</sup> 0
	01	<sup>1</sup> X	<sup>5</sup> 0	<sup>13</sup> X	<sup>9</sup> 0
	11	<sup>3</sup> 0	<sup>7</sup> 0	<sup>15</sup> X	<sup>11</sup> 0
	10	<sup>2</sup> X	<sup>6</sup> 0	<sup>14</sup> X	<sup>10</sup> 0

Figura 115 -  $S9 = C8 \cdot C4$

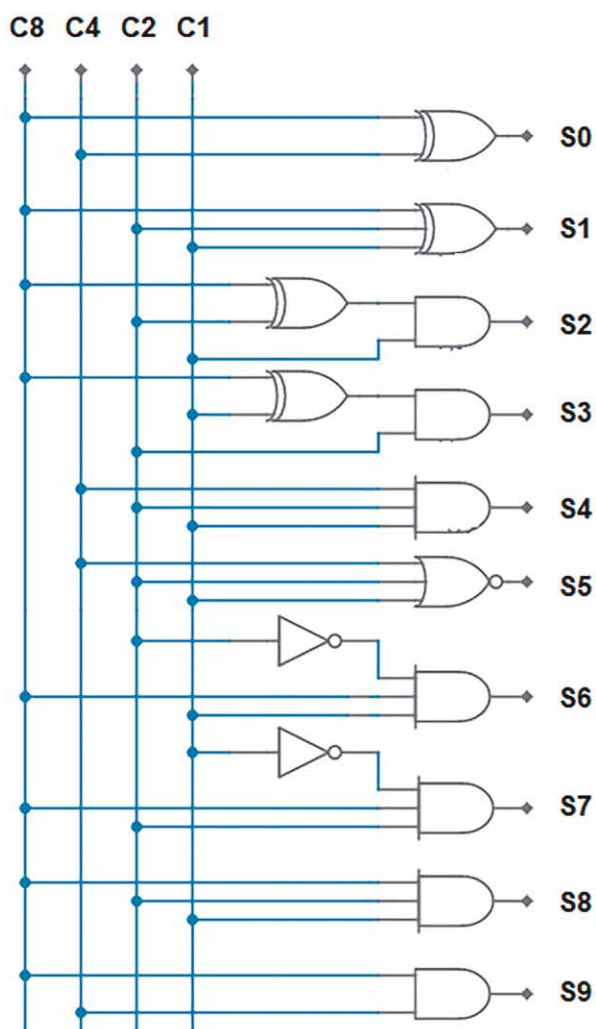


Figura 116 - Circuito decodificador Excesso 3

## 7.5 Código de Gray

O código de Gray apresenta como característica principal o fato de que apenas um *bit* varia na mudança de um número para o subsequente. Esse código surgiu nos primórdios da computação, quando ainda se utilizavam válvulas e relés: o consumo de energia e o ruído gerado na transição de estado lógico eram elevados, então objetivava-se minimizar o número de tais transições. Ele continua em uso, uma vez que uma das ideias por trás dele, que é diminuir o tempo de transição e de resposta do circuito, ainda se aplica. O nome do código vem de seu criador, Frank Gray (1887-1969).

A tabela a seguir apresenta o código de Gray. Para facilitar a visualização, cada *bit* que foi alterado em relação à linha anterior encontra-se em destaque:

**Tabela 66 – Código de Gray**

Decimal	Gray			
	Canal 8	Canal 4	Canal 2	Canal 1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
A	1	1	1	1
B	1	1	1	0
C	1	0	1	0
D	1	0	1	1
E	1	0	0	1
F	1	0	0	0

### Codificador código de Gray

A obtenção das expressões para o circuito codificador segue a mesma metodologia empregada para todos os codificadores anteriores:

$$\text{Canal 8} = E8 + E9 + Ea + Eb + Ec + Ed + Ee + Ef$$

$$\text{Canal 4} = E4 + E5 + E6 + E7 + E8 + E9 + Ea + Eb$$

$$\text{Canal 2} = E2 + E3 + E4 + E5 + Ea + Eb + Ec + Ed$$

$$\text{Canal 1} = E1 + E2 + E5 + E6 + E9 + Ea + Ed + Ee$$

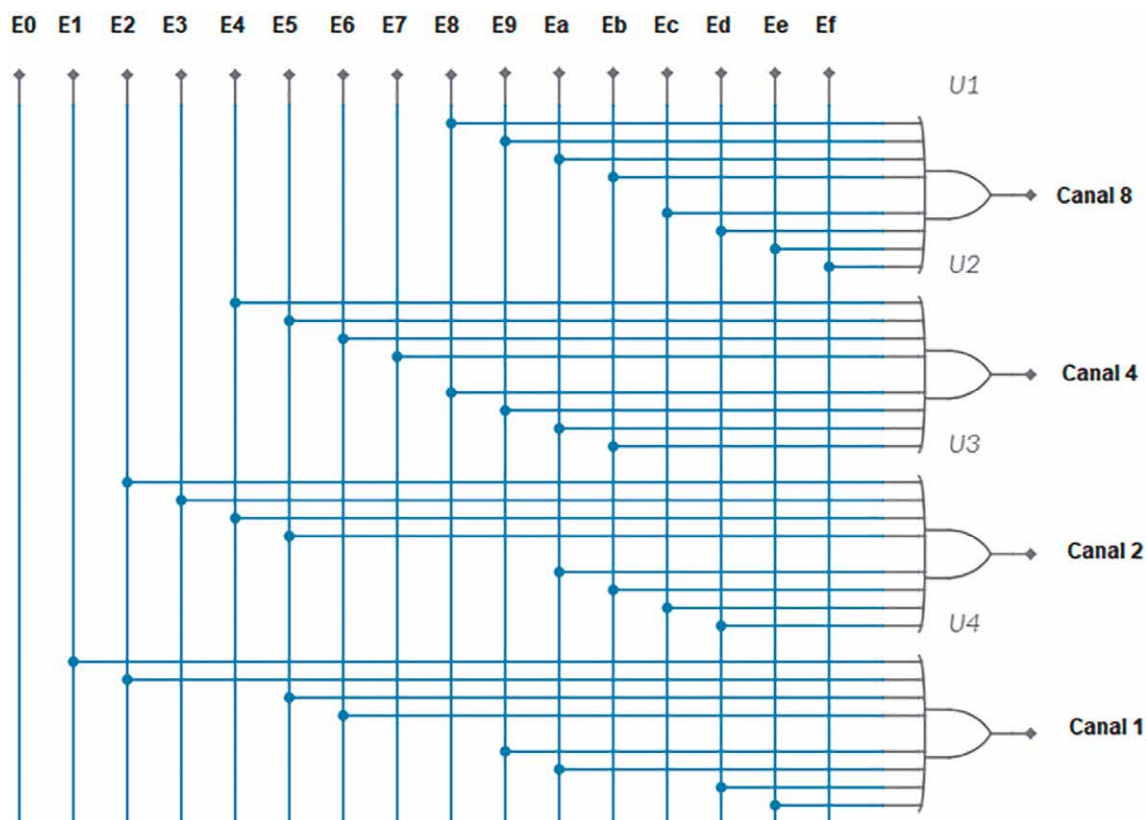


Figura 117 – Circuito codificador para o código de Gray

Decodificador código de Gray

Para construir um decodificador, as entradas (E0 a E9) novamente se tornarão as saídas (S0 a S9, respectivamente), como feito para os decodificadores anteriores. Constam a seguir as tabelas verdade para esse decodificador:

Tabela 67 – Decodificador para o código de Gray: (a) saídas S0 a S7; (b) saídas S8 a Sf

(a)

Entradas				Saídas							
Canal 8	Canal 4	Canal 2	Canal 1	S0	S1	S2	S3	S4	S5	S6	S7
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	1
0	1	0	1	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	1	0	0



(b)

Entradas				Saídas							
Canal 8	Canal 4	Canal 2	Canal 1	S8	S9	Sa	Sb	Sc	Sd	Se	Sf
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	1	0	0	0
1	0	1	1	0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	1	0	0	0	0
1	1	1	1	0	0	1	0	0	0	0	0

Para cada uma das combinações das entradas existe apenas uma saída verdadeira, assim como foi feito para o código BCH. As expressões lógicas de cada saída, com suas respectivas simplificações, utilizando-se a lei de De Morgan para agrupar conjuntos de negações sob uma porta NOR, são indicadas a seguir:

$$S0 = \overline{(C8 + C4 + C2 + C1)}$$

$$S1 = \overline{(C8 + C4 + C2)} \cdot C1$$

$$S2 = \overline{(C8 + C4)} \cdot C2 \cdot C1$$

$$S3 = \overline{(C8 + C4 + C1)} \cdot C2$$

$$S4 = \overline{(C8 + C1)} \cdot C4 \cdot C2$$

$$S5 = \overline{C8} \cdot C4 \cdot C2 \cdot C1$$

$$S6 = \overline{(C8 + C2)} \cdot C4 \cdot C1$$

$$S7 = \overline{(C8 + C2 + C1)} \cdot C4$$

$$S8 = \overline{(C2 + C1)} \cdot C8 \cdot C4$$

$$S9 = \overline{C2} \cdot C8 \cdot C4 \cdot C1$$

$$Sa = C8 \cdot C4 \cdot C2 \cdot C1$$

$$Sb = \overline{C1} \cdot C8 \cdot C4 \cdot C2$$

$$S_c = \overline{(C_4 + C_1)} \cdot C_8 \cdot C_2$$

$$S_d = \overline{C_4} \cdot C_8 \cdot C_2 \cdot C_1$$

$$S_e = \overline{(C_4 + C_2)} \cdot C_8 \cdot C_1$$

$$S_f = \overline{(C_4 + C_2 + C_1)} \cdot C_8$$



## Observação

Os circuitos do decodificador do código de Gray são os mesmos do BCH, em outra ordem. Por exemplo, a saída S8 no decodificador Gray é igual à S<sub>c</sub> do BCH.

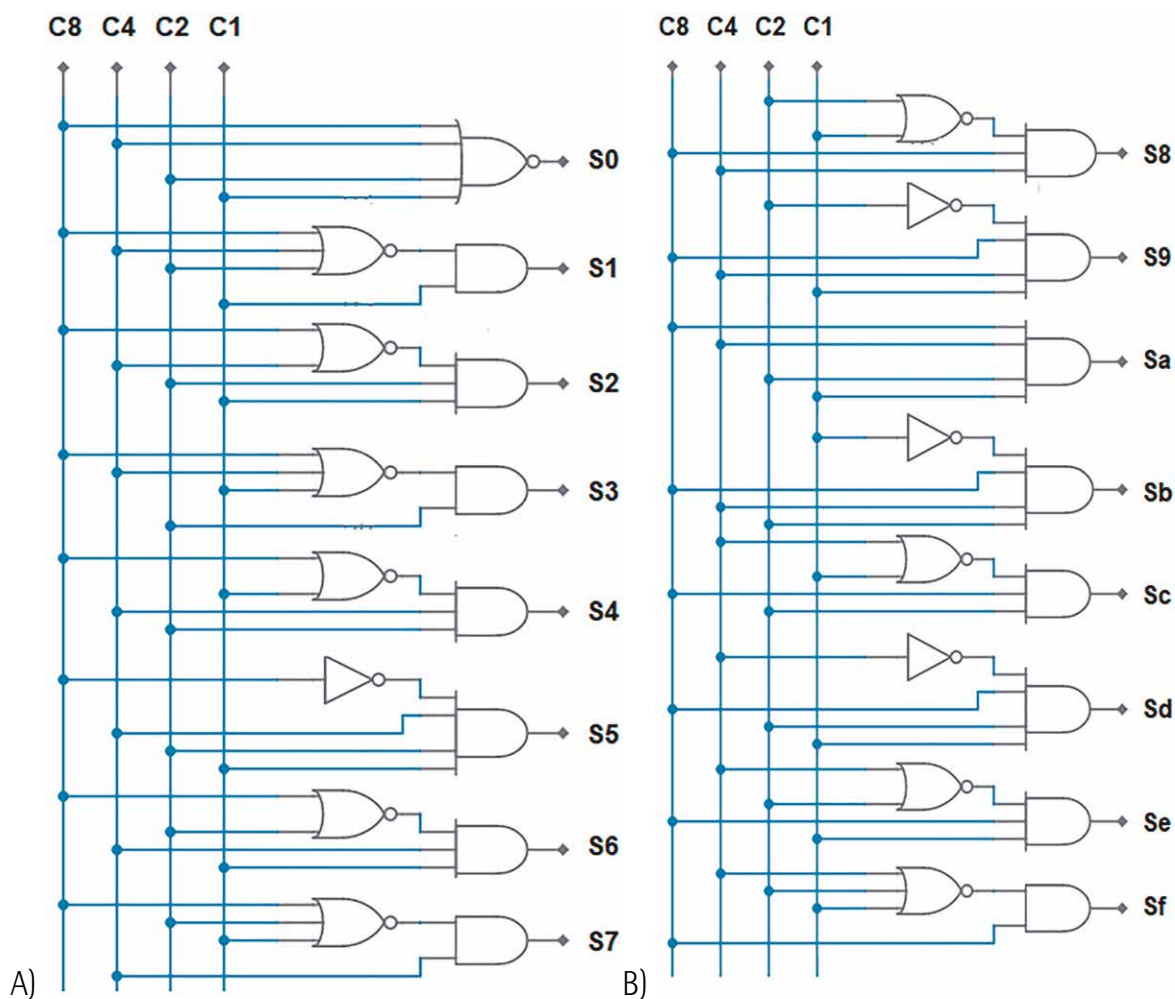


Figura 118 – Circuito decodificador para código de Gray: (a) saídas S0 a S7; (b) saídas S8 a Sf

## 7.6 Decodificador BCD 8420 para SSD

Retornamos ao SSD, mas agora ele será tratado como um decodificador BCD 8421. Portanto, as saídas não podem mais ser obtidas diretamente da tabela, uma vez que as entradas deixam de ser mutuamente exclusivas, apenas uma entrada acionada por vez. A tabela a seguir acrescenta as entradas BCD 8421 à tabela "Entradas e saídas dos circuitos de um SSD" que foi apresentada no tópico anterior:

**Tabela 68 – Saídas para um SSD com as entradas BCD 8421**

Decimal (Entrada)	Exibição	A (8)	B (4)	C (2)	D (1)	Saídas						
						A	B	C	D	E	F	G
0	0	0	0	0	0	1	1	1	1	1	0	1
1	1	0	0	0	1	0	1	1	0	0	0	0
2	2	0	0	1	0	1	1	0	1	1	0	1
3	3	0	0	1	1	1	1	1	1	0	0	1
4	4	0	1	0	0	0	1	1	0	0	1	1
5	5	0	1	0	1	1	0	1	1	0	1	1
6	6	0	1	1	0	1	0	1	1	1	1	1
7	7	0	1	1	1	1	1	1	0	0	0	0
8	8	1	0	0	0	1	1	1	1	1	1	1
9	9	1	0	0	1	1	1	1	1	0	1	1

Adaptada de: Wikimedia (s.d.)b.

Assim, para obter as saídas, é necessária a construção dos mapas de Karnaugh para cada uma delas. A seguir são apresentados os mapas, com a expressão lógica já simplificada de cada saída:

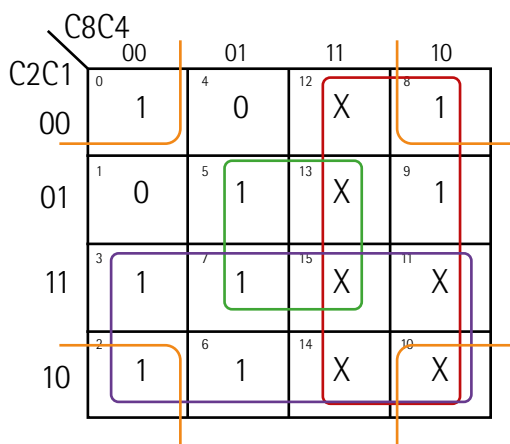


Figura 119 –  $A = C8 + C2 + \overline{(C4 \oplus C1)}$

		C8C4			
		00	01	11	10
C2C1	00	0 1	4 1	12 X	8 1
	01	1 1	5 0	13 X	9 1
	11	3 1	7 1	15 X	11 X
	10	2 1	6 0	14 X	10 X

Figura 120 -  $B = \overline{C4} + (\overline{C2} \oplus C1)$

		C8C4			
		00	01	11	10
C2C1	00	0 1	4 1	12 X	8 1
	01	1 1	5 1	13 X	9 1
	11	3 1	7 1	15 X	11 X
	10	2 0	6 1	14 X	10 X

Figura 121 -  $C = \overline{C2} + C4 + C1$

		C8C4			
		00	01	11	10
C2C1	00	0 1	4 0	12 X	8 1
	01	1 0	5 1	13 X	9 1
	11	3 1	7 0	15 X	11 X
	10	2 1	6 1	14 X	10 X

Figura 122 -  $D = C8 + \overline{C4} \cdot C2 + C2 \cdot \overline{C1} + C4 \cdot \overline{C2} \cdot C1 + (\overline{C4} + \overline{C1})$

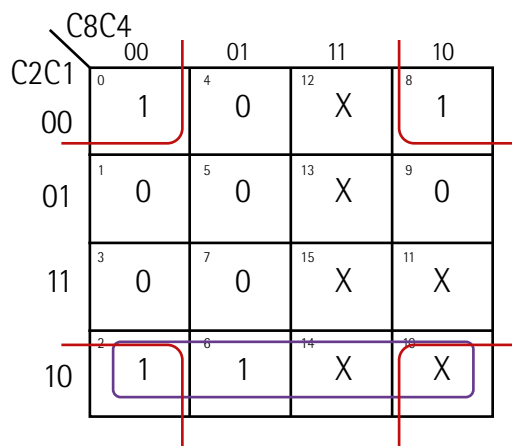


Figura 123 -  $E = \overline{C4 + C2} + C2 . \overline{C1}$

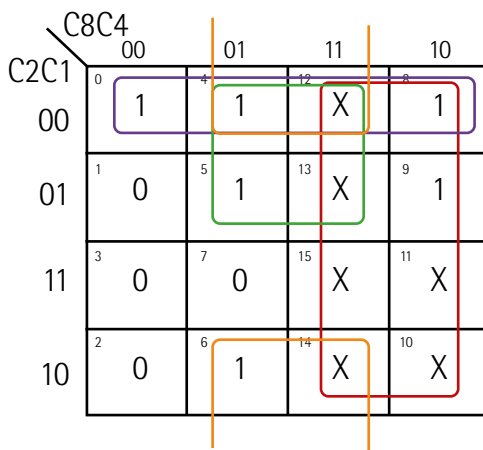


Figura 124 -  $F = C8 + \overline{C2 + C1} + C4 . \overline{C2 . C1}$

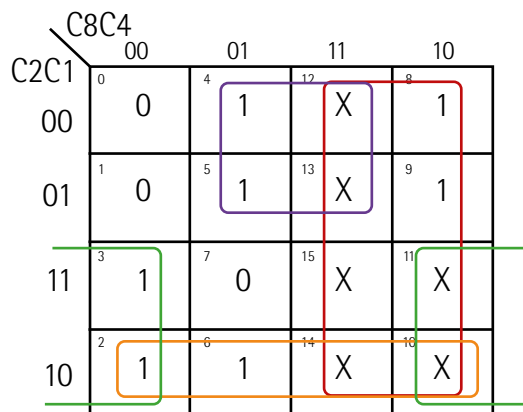


Figura 125 -  $G = C8 + (C4 \oplus C2) + C2 . \overline{C1}$

Assim, o circuito completo, com as sete saídas devidamente ligadas ao *display*, ficará:

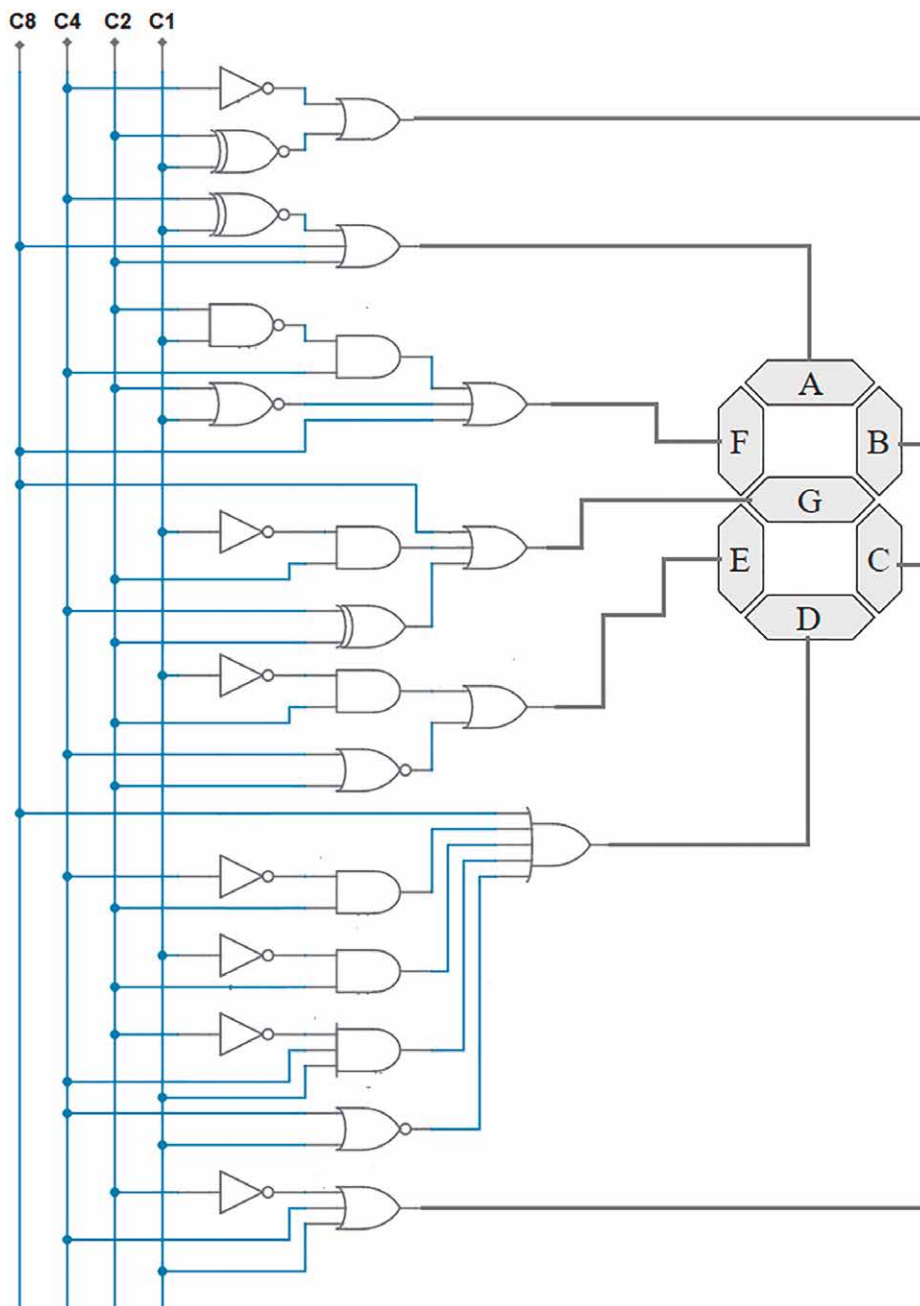


Figura 126 – SSD controlado por um circuito decodificador BCD 8421

### 8 CIRCUITOS ARITMÉTICOS

Circuitos aritméticos são circuitos que emulam, por meio do uso de portas lógicas, operações aritméticas. Neste tópico serão apresentados circuitos meio somador e meio subtrator, que fazem as respectivas operações para entradas de um *bit*, e os circuitos completos para ambas as operações, que permitem operações para quaisquer quantidades de *bits* ao serem ligados um na saída do outro.

Circuitos multiplicadores são circuitos sequenciais, e, por fugirem ao escopo deste livro-texto, não serão apresentados.

Circuitos aritméticos são a base da unidade lógica e aritmética (*arithmetic logic unit*), que é um circuito lógico digital constituinte de uma unidade central de processamento (CPU, em inglês). Seu conceito foi proposto por John von Neumann (1903-1957) em 1945, ou seja, ainda nos primórdios da computação eletrônica.

## 8.1 Circuitos meio somadores

Um circuito meio somador (*half adder*, em inglês) é um circuito que emula, por meio de operações lógicas, o resultado de uma soma entre dois algarismos binários de um *bit* cada.



### Lembrete

Vejam na sequência a tabuada da soma de números binários:

$$0 + 0 = 00$$

$$0 + 1 = 01$$

$$1 + 1 = 10$$

$$1 + 1 + 1 = 11$$

A soma de dois números de um *bit* resulta em dois *bits*, cada uma sendo uma saída do circuito. A tabela a seguir apresenta esta saída, na qual A e B são os números a serem somados e sendo S2 o algarismo mais significativo (maior) e S1 o algarismo menos significativo:

**Tabela 69 – Saídas S1 e S2 para um circuito meio somador**

A	B	S2	S1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Uma análise rápida da tabela, sem a necessidade de mapa de Karnaugh, já indica que  $S1 = A \oplus B$  e  $S2 = A \cdot B$

Assim, um circuito meio somador seria bastante simples, como mostrado a seguir:

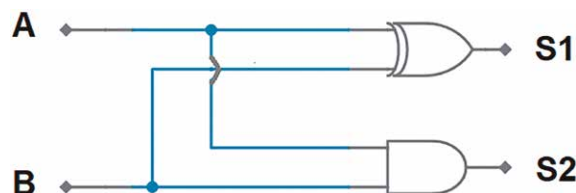


Figura 127 – Circuito meio somador

Um circuito meio somador pode ser reproduzido em algumas linguagens de programação que permitam a realização de operações lógicas com números inteiros, por exemplo, o Python. A figura a seguir ilustra um código em Python que executa exatamente as mesmas operações lógicas sobre dois números:

```
a= int(input("Digite A:"))
b=int(input("Digite B:"))
print("Soma: ", (a and b), (a ^ b))
```

### Execução:

```
Digite A: 1
Digite B: 1
Soma:  1 0
>>> |
```

Figura 128 – Código em Python em um circuito meio somador



### Saiba mais

A tipagem fraca do Python permite o uso de operadores lógicos sobre valores inteiros. Detalhes adicionais podem ser encontrados em:

W3SCHOOLS. *Python Data Types*. [s.d.]a.

Disponível em: <https://bit.ly/2Tr7jGA>. Acesso em: 10 jun. 2021.

W3SCHOOLS. *Python Operators*. [s.d.]b.

Disponível em: <https://bit.ly/2Tr7jGA>. Acesso em: 10 jun. 2021.

Assim, para somarmos dois números de dois *bits*, por exemplo, podemos fazer uma combinação de ambos os circuitos: usaremos dois meio somadores, um para somar os dois primeiros dígitos e outro para somar os dois segundos dígitos. Com o objetivo de representar o "vai-um", podemos



conectar a saída S2 do primeiro à saída S2 do segundo por meio de um terceiro meio somador e a saída deste último à saída S2 do segundo somador por meio de uma porta OR. Para deixar mais claro, usaremos como forma resumida do circuito meio somador o esquema apresentado na figura a seguir:



Figura 129 – Versão esquematizada de um circuito meio somador

Assim, para realizar uma soma de dois números de dois *bits*, A2A1 e B2B1 (sendo A2 e B2 os dígitos mais significativos de cada um dos dois números), como apresentada na figura anterior, o circuito ficará conforme indicado na figura a seguir.

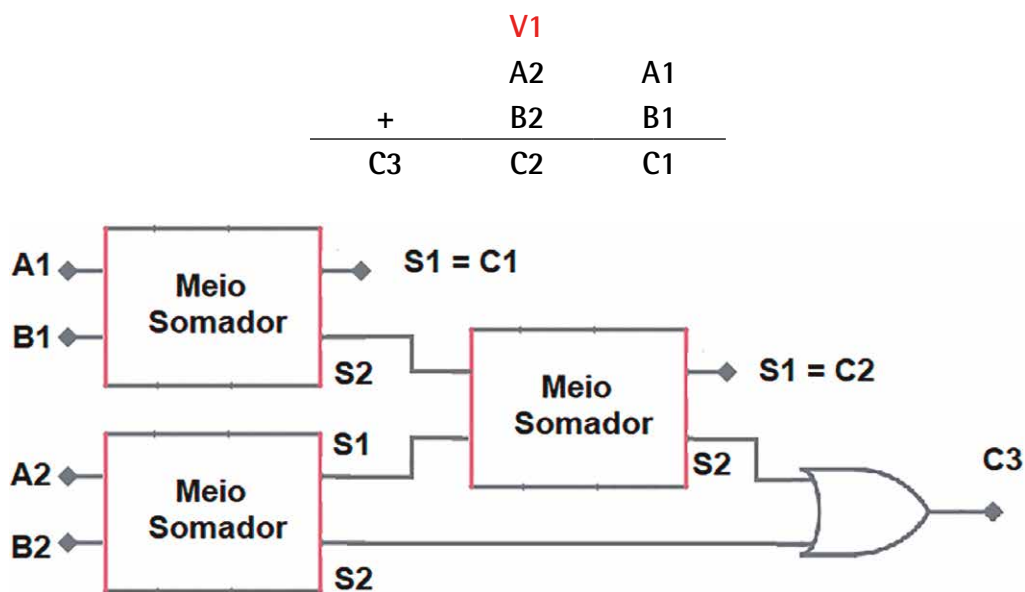


Figura 130 – Arranjo de meio somadores para a soma de um número de dois *bits*

O circuito lógico, com os meio somadores destacados, é apresentado a seguir. As expressões lógicas de cada saída do circuito são:

$$C1 = A1 \oplus B1$$

$$C2 = (A1 \cdot B1) \oplus (A2 \oplus B2)$$

$$C3 = (A1 \cdot B1) \cdot (A2 \oplus B2) + (A2 \cdot B2)$$

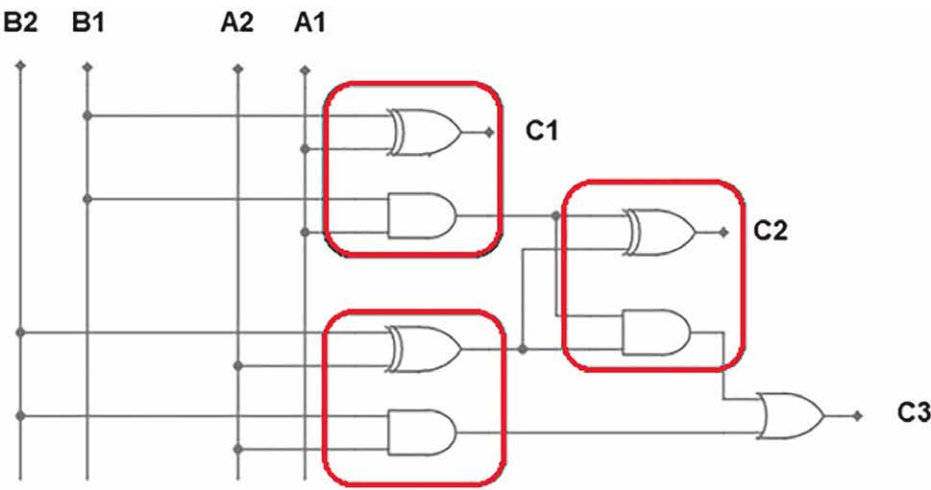


Figura 131 – Arranjo de três meio somadores para realizar a soma de dois números de dois bits

Tabela 70 – Saídas do circuito emulando uma soma binária

A2	A1		B2	B1		C3	C2	C1
0	0	+	0	0	=	0	0	0
0	0	+	0	1	=	0	0	1
0	0	+	1	0	=	0	1	0
0	0	+	1	1	=	0	1	1
0	1	+	0	0	=	0	0	1
0	1	+	0	1	=	0	1	0
0	1	+	1	0	=	0	1	1
0	1	+	1	1	=	1	0	0
1	0	+	0	0	=	0	1	0
1	0	+	0	1	=	0	1	1
1	0	+	1	0	=	1	0	0
1	0	+	1	1	=	1	0	1
1	1	+	0	0	=	0	1	1
1	1	+	0	1	=	1	0	0
1	1	+	1	0	=	1	0	1
1	1	+	1	1	=	1	1	0

A maioria dos números possui mais de um *bit*, e a combinação apenas de diversos circuitos meio somadores não é o suficiente para a realização de somas com grandes quantidades de dígitos. Seria necessário o acréscimo de dois meio somadores para cada novo par de dígitos a ser somado para tratar com o “vai um” que surge para cada dígito além do primeiro.

Logo, há necessidade de um somador completo que realize a operação de adição de três valores.

## 8.2 Circuitos somadores completos

Circuitos somadores completos são, de certa forma, uma extensão dos meio somadores, possibilitando a soma de três dígitos de um *bit* cada. Assim, a soma desses três valores de entrada irá resultar em duas saídas de um *bit* cada, para comportar todos os possíveis valores das operações. A tabela a seguir é a tabela verdade para a soma de três valores A, B e C, sendo novamente S2 o dígito mais significativo e S1 o dígito menos significativo.

**Tabela 71 – Saídas S1 e S2 para um circuito somador completo**

A	B	C	S2	S1
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Montando os mapas de Karnaugh para as duas saídas:

**Saída S1:**

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

Figura 132

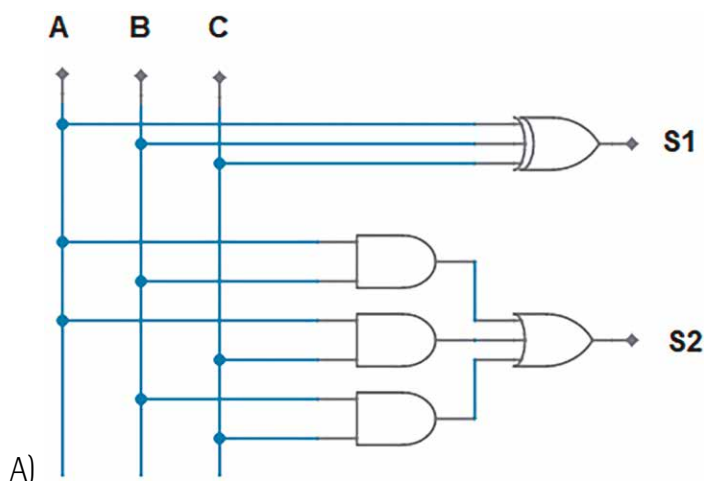
A expressão resultante será  $A \oplus B \oplus C$ .

Saída S2:

AB \ C		00	01	11	10
C	0	0 <sup>0</sup> 0	2 <sup>2</sup> 0	6 <sup>6</sup> 1	4 <sup>4</sup> 0
	1	1 <sup>1</sup> 0	3 <sup>3</sup> 1	7 <sup>7</sup> 1	5 <sup>5</sup> 1

Figura 133

A expressão resultante será  $A \cdot B + A \cdot C + B \cdot C$ , e o circuito será o mostrado na figura a seguir, bem como o código equivalente em Python, como foi feito para o meio somador:



```
a= int(input("Digite A:"))
b= int(input("Digite B:"))
c= int(input("Digite C:"))
print("Soma: ", ((a and b) or (a and c) or (b and c)), (a ^ b ^ c))
```

B)

Figura 134 – (a) Circuito somador completo; (b) código em Python que emula um circuito somador completo

Desta forma, a questão da soma de uma quantidade de *bits* pode ser resolvida iniciando-se o circuito com um meio somador, e sua saída S2 seria uma das entradas de um circuito somador completo, representando o "vai-um". A figura a seguir ilustra de forma esquemática tal arranjo, enquanto a figura adiante mostra o circuito com as portas lógicas (utilizando a mesma atribuição dos valores somados indicada na figura "Versão esquematizada de um circuito meio somador"):

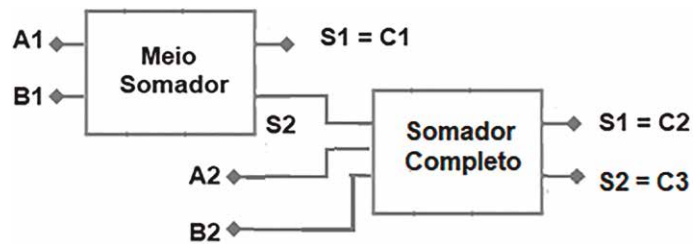


Figura 135 – Arranjo de somadores para a soma de dois números de dois bits

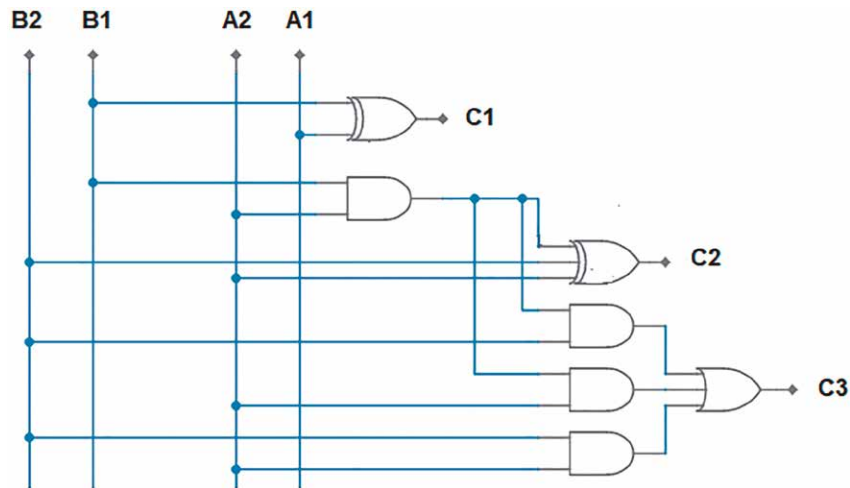


Figura 136 – Arranjo de um meio somador e um somador completo para a soma de dois números de dois bits

Tal arranjo é mais eficiente do que o de meio somadores apenas, uma vez que, para anexar um *bit* a cada um dos números que está sendo somado, basta conectar outro somador completo à saída S2 do último somador completo do circuito, ligando-o também com os dois novos *bits*. A figura a seguir ilustra, a título de exemplo, o arranjo esquemático para a soma de dois números de três bits A3A2A1 e B3B2B1:

		V2	V1	
		A3	A2	A1
+		B3	B2	B1
<hr/>				
C4	C3	C2	C1	

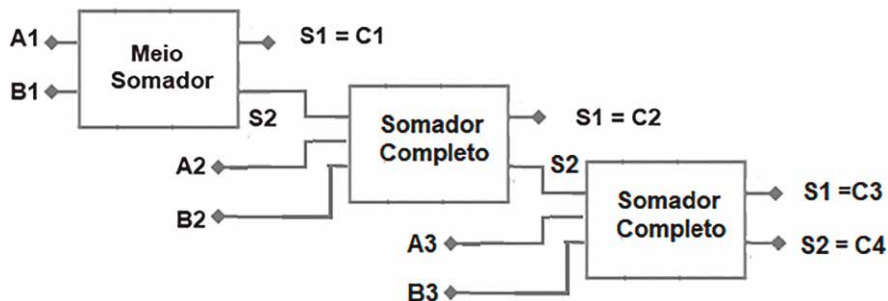


Figura 137 – Arranjo de um meio somador e dois somadores completos para a soma de dois números de três bits

## 8.3 Circuitos meio subtratores

O circuito meio subtrator (*half subtractor*, em inglês) é semelhante ao meio somador, porém ele emula por meio de operações lógicas o resultado de uma subtração entre dois algarismos binários de um *bit*.



### Lembrete

Vejam na sequência a tabuada da subtração de números binários:

$$0 - 0 = 00$$

$$1 - 1 = 00$$

$$1 - 0 = 01$$

$$0 - 1 = 11 \text{ (resulta em 1 e "empresta 1")}$$

A tabela verdade para este circuito, considerando uma subtração  $A - B$ , fica sendo a indicada na tabela a seguir:

**Tabela 72 – Saídas S e E para um circuito meio subtrator**

A	B	S	E1
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

A saída **E1** significa que A é menor que B, por isso A "empresta 1" do valor seguinte da subtração. A saída **S** representa o resultado da subtração, levando em consideração caso tenha sido "emprestado 1" do valor seguinte.

Como para o meio somador, podemos extrair as expressões lógicas diretamente da tabela verdade, sendo  $E1 = A \oplus B$  e  $S = \overline{A} \cdot B$ . O circuito lógico do circuito meio subtrator é mostrado na figura a seguir.

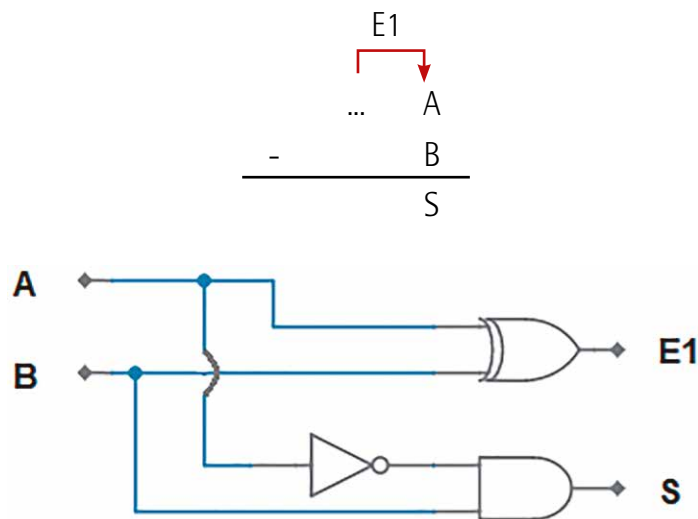


Figura 138 – Circuito meio subtrator e operação de subtração correspondente

Diferentemente do meio somador, não podemos fazer uma combinação de diversos meio subtratores para realizar uma subtração com mais *bits*, uma vez que a entrada do circuito não comporta o eventual "empréstimo 1" que possa ter vindo da operação de subtração do dígito anterior. Portanto, precisamos de um subtrator completo, que possua três entradas, sendo uma delas para o empréstimo que possa ter ocorrido de operações anteriores.

## 8.4 Circuitos subtratores completos

Este circuito considera que possa ter sido emprestado 1 da operação anterior, como é mostrado a seguir, onde T1 indica o valor que foi emprestado. Assim, a tabela verdade ficará a indicada na tabela a seguir.

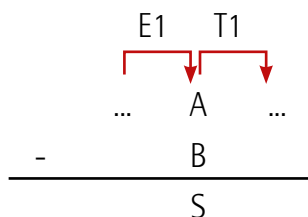


Tabela 73 – Saídas S1 e S2 para um circuito somador completo

A	B	T1	S	E1
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0

A	B	T1	S	E1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Montando os mapas de Karnaugh para as saídas:

**Saída S:**

AB \ T1		AB			
		00	01	11	10
T1	0	<sup>0</sup> 0	<sup>2</sup> 1	<sup>6</sup> 0	<sup>4</sup> 1
	1	<sup>1</sup> 1	<sup>3</sup> 0	<sup>7</sup> 1	<sup>5</sup> 0

Figura 139

A expressão resultante será  $A \oplus B \oplus T1$ .

**Saída E1:**

AB \ T1		AB			
		00	01	11	10
T1	0	<sup>0</sup> 0	<sup>2</sup> 1	<sup>6</sup> 0	<sup>4</sup> 0
	1	<sup>1</sup> 1	<sup>3</sup> 1	<sup>7</sup> 1	<sup>5</sup> 0

Figura 140

A expressão resultante será  $\bar{A} \cdot (B + T1) + B \cdot T1$ .



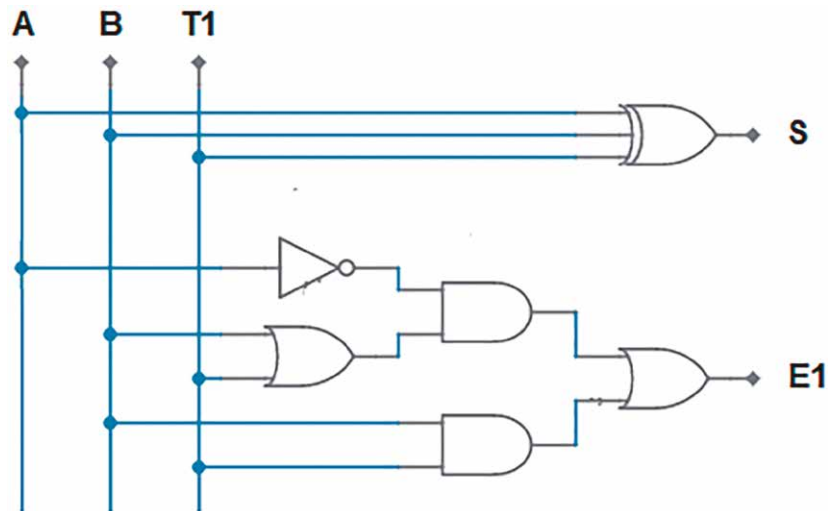


Figura 141 – Circuito subtrator completo

Assim como foi feito para os circuitos somador completo e meio somador, podemos combinar os circuitos subtrator completo e meio subtrator para realizar uma operação binária com mais números que possuam mais de um *bit*. Utilizando a mesma esquematização utilizada para os circuitos somadores, podemos construir um circuito que realiza a operação **A2A1 - B2B1**, sendo dois números de dois *bits*:

$$\begin{array}{r}
 \text{E2} \quad \text{E1} \\
 \text{A2} \quad \text{A1} \\
 - \quad \text{B2} \quad \text{B1} \\
 \hline
 \text{C2} \quad \text{C1}
 \end{array}$$

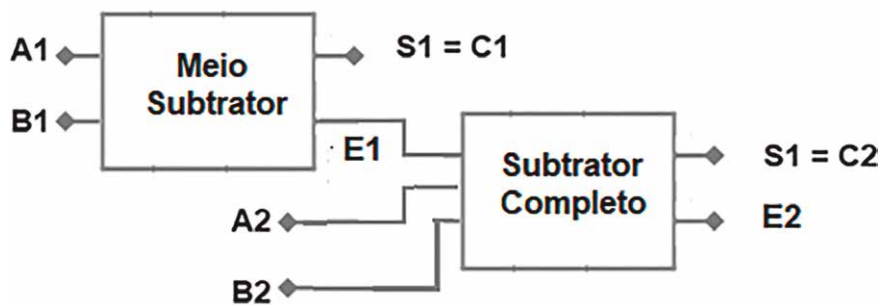


Figura 142 – Arranjo de subtratores para a subtração de dois números de dois *bits*

A figura a seguir apresenta o circuito com suas portas lógicas. É importante perceber que a saída E1 sempre será "morta", ou seja, não terá significado na saída final do circuito lógico.

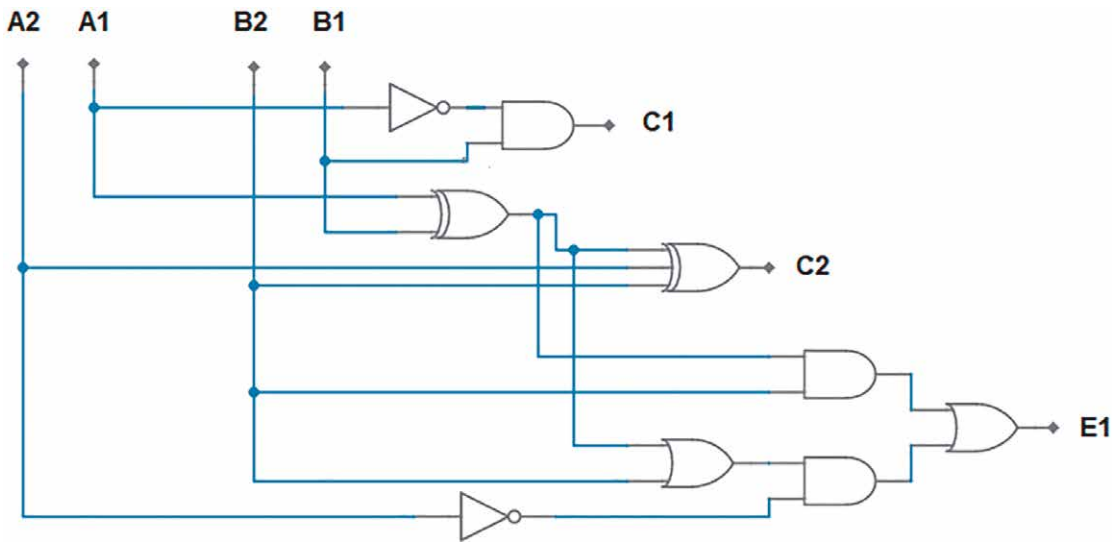


Figura 143 – Arranjo de um meio subtrator e um subtrator completo para a subtração de dois números de dois bits

A tabela a seguir mostra o resultado das saídas desse circuito para a soma de dois números de dois bits. Nas linhas destacadas não há um resultado, porque nesses casos está sendo feita a subtração de um número maior a partir de um número menor. Embora ao construir esse circuito em simuladores haja uma saída lógica para tais casos, a saída não possui significado aritmético, uma vez que esse circuito (e nenhum circuito feito a partir de arranjos dos circuitos subtratores apresentados) não permite operações com números negativos.

Tabela 74 – Saídas do circuito emulando uma subtração binária

A2	A1		B2	B1		C2	C1
0	0	-	0	0	=	0	0
0	0	-	0	1	=		
0	0	-	1	0	=		
0	0	-	1	1	=		
0	1	-	0	0	=	0	1
0	1	-	0	1	=	0	0
0	1	-	1	0	=		
0	1	-	1	1	=		
1	0	-	0	0	=	1	0
1	0	-	0	1	=	0	1
1	0	-	1	0	=	0	0
1	0	-	1	1	=		
1	1	-	0	0	=	1	1
1	1	-	0	1	=	1	0
1	1	-	1	0	=	0	1
1	1	-	1	1	=	0	0

A combinação de circuitos subtratores pode ser feita com mais circuitos, assim como foi feito para os somadores. A figura a seguir apresenta o arranjo para um subtrator que realiza a subtração entre dois números de três *bits*. Como foi dito, a última saída “empresta 1” (E3, na figura), não possuindo significado aritmético.

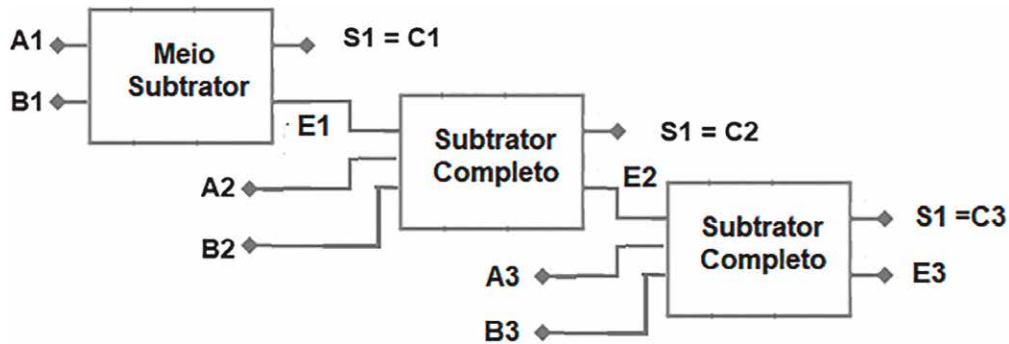


Figura 144 – Arranjo de subtratores para a subtração de dois números de três *bits*



### Saiba mais

Além dos circuitos aritméticos, existem multiplicadores, que são circuitos sequenciais. A fim de conhecer melhor, leia o capítulo 5 do seguinte livro:

LOURENÇO, A. C. *et al. Circuitos digitais: estude e use*. São Paulo: Érica, 2008.

Com a combinação de circuitos aritméticos, encerramos a unidade sobre circuitos combinacionais. Duas importantes categorias, que são os multiplexadores e comparadores, não foram aqui tratadas, visando não estender demais o conteúdo apresentado. Além destes, não foi abordado nenhum tipo de circuito sequencial nem de memórias, tópicos que fogem da ementa da disciplina.

O objetivo, conforme dito na Introdução, é fornecer apenas os fundamentos básicos para a compreensão da lógica envolvida em circuitos, e tentar fazer um paralelo com a lógica de programação.

### Exemplo de aplicação

Todos os exemplos de circuitos combinacionais aqui apresentados podem ser combinados entre si. Uma proposta interessante seria projetar um circuito somador para a soma de dois dígitos de três *bits* (circuito com seis entradas), resultando em uma saída de quatro *bits*; esta saída alimentaria um SSD via um decodificador BCD 8421. Execute-o e reflita a respeito do resultado.



### Resumo

Nesta unidade, vimos que os circuitos aritméticos são um grupo de circuitos combinacionais que emulam operações aritméticas com números binários por meio do funcionamento de portas lógicas. Um circuito aritmético não realiza exatamente a operação aritmética correspondente, mas devolve um valor lógico (0 ou 1) correspondente ao resultado da operação de cada par de dígitos dos números que estão sendo operados.

Observamos que os circuitos meio somadores realizam a soma de dois valores de um *bit*. Embora vários desses circuitos possam ser combinados para formar um circuito capaz de somar números com uma quantidade maior de *bits*, este arranjo acaba ficando muito grande e complexo. Assim, faz-se uso dos somadores completos, que realizam a soma de três valores de um *bit* e podem receber o "vai um" de uma operação de soma anterior.

Entendemos que os circuitos meio subtratores fazem uma subtração de forma análoga aos meio somadores, mas não podem ser combinados para realizar uma subtração com mais dígitos, uma vez que nesta operação os valores podem ir em ambas as direções, emprestando do valor seguinte e para o valor anterior. Portanto, o subtrator completo executa a operação considerando tais casos, e pode ser combinado com o meio subtrator para realizar operações com uma quantidade maior de *bits*.

Os circuitos codificadores e decodificadores são uma classe de circuitos combinacionais que têm como finalidade converter um sinal decimal de entrada em um sinal binário de saída (codificadores) ou vice-versa.

Foram exibidos quatro tipos de códigos:

Código BCD 8421, que faz a conversão direta de um dígito de 0 a 9 para o sistema binário, sendo um dos códigos mais utilizados atualmente.

Código BCH, que pode ser considerado uma extensão do código BCD, fazendo a conversão de valores hexadecimais de 0 a 16 (0 a F) para o sistema binário.

Código Excesso 3 ou código de Stibitz, que converte um número decimal para um número binário acrescido de 3, daí o seu nome.

Código de Gray, que tem como característica o fato de que cada dígito codificado possui alteração de apenas um *bit* em relação aos dígitos imediatamente anterior e posterior. Era um código muito utilizado quando os circuitos ainda eram valvulados, para diminuir o consumo de energia e o tempo de resposta.

Por fim, vimos que um *display* de sete segmentos (SSD) pode ser controlado por um decodificador BCD 8421, e os circuitos correspondentes foram apresentados.



### Exercícios

**Questão 1.** Codificadores e decodificadores são importantes grupos entre os circuitos digitais combinacionais. Os codificadores são circuitos que fazem a conversão de uma informação expressa em um formato qualquer para um conjunto de sinais binários, que podem ser utilizados por circuitos digitais subsequentes. Os decodificadores, por sua vez, convertem informações binárias em outro formato, comumente mais agradável de ser interpretado pelo usuário. Para que essa conversão seja efetivada, existem códigos capazes de fazer a comunicação entre diferentes formatos. A respeito desses códigos, avalie as afirmativas.

I – O código BCD 8421 é principalmente utilizado para decodificar sinais digitais em valores numéricos hexadecimais em um *display* de sete segmentos.

II – O código BCH foi desenvolvido especialmente para trabalhar com codificação ou decodificação entre os sistemas octal e binário.

III – O código de Gray apresenta como principal característica o fato de apenas um *bit* variar na representação binária de um número para o seu subsequente.

É correto o que se afirma em:

A) I, apenas.

B) II, apenas.

C) III, apenas.

D) II e III, apenas.

E) I, II e III.

Resposta correta: alternativa C.

### Análise das afirmativas

I – Afirmativa incorreta.

Justificativa: o código BCD (*Binary Coded Decimal*) 8421, mais conhecido apenas como código BCD, traduz cada um dos valores numéricos decimais de 0 a 9 para sinais binários. Assim, cada algarismo decimal é transformado em um conjunto de 4 *bits*. Entre os códigos numéricos estudados, esse costuma ser o mais fácil de ser compreendido. Utilizando tal código, podemos construir decodificadores, que recebem como entrada conjuntos de *bits*, que serão exibidos como algarismos de 0 a 9 (a cada 4 *bits*, teremos 1 algarismo decimal). Também podemos construir codificadores que fazem o processo inverso: recebem a informação decimal e transformam cada algarismo em quatro dígitos binários. O código BCD não é adequado, portanto, para trabalhar com o sistema hexadecimal, já que ele não abrange os algarismos de A a F. Ele seria conveniente para decodificar sinais digitais em valores numéricos decimais em um *display* de sete segmentos.

II – Afirmativa incorreta.

Justificativa: o código BCH (*Binary Coded Hexadecimal*) trabalha de forma semelhante ao código BCD. No entanto, o BCH representa cada um dos algarismos do sistema hexadecimal por um conjunto de 4 *bits*. Dessa forma, ele foi desenvolvido para trabalhar com codificação ou decodificação entre os sistemas hexadecimal e binário.

III – Afirmativa correta.

Justificativa: segundo o código de Gray, a representação binária do número 3 é (0010) e a do número 4 é (0110). Note que apenas o segundo *bit*, da esquerda para a direita, mudou. Essa característica, chamada de adjacência, facilita o processo de simplificação por mapas de Karnaugh e ajuda a aumentar a confiabilidade do sistema.

**Questão 2.** Observe a figura a seguir, que representa o diagrama de blocos de um circuito somador binário. Sabe-se que  $C_i$  representa *carry-in* (ou "vem-um"),  $C_o$  representa *carry-out* (ou "vai-um"), S representa a entrega do resultado da soma do bloco e A e B representam *bits* dos valores que queremos somar. O *carry-in* de um bloco recebe o *carry-out* de um possível bloco anterior que realizou a soma de um *bit* menos significativo.

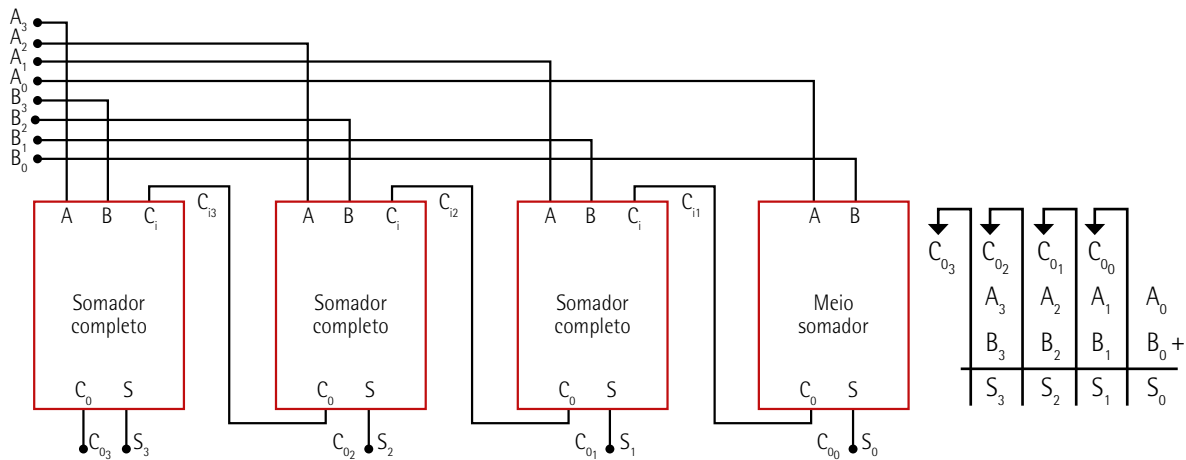


Figura 145

Adaptada de: LOURENÇO, A. C. et al. *Circuitos digitais*. 9. ed. São Paulo: Érica, 2007. p. 189.

A respeito deste circuito, avalie as afirmativas.

I – Trata-se de um exemplo de circuito digital sequencial.

II – O circuito é capaz de realizar a soma de dois números contendo 4 *bits* cada.

III – O circuito meio somador é responsável por somar os *bits* mais significativos de cada um dos dois números.

É correto o que se afirma em:

A) I, apenas.

B) II, apenas.

C) III, apenas.

D) II e III, apenas.

E) I, II e III.

Resposta correta: alternativa B.

## Análise das afirmativas

### I – Afirmativa incorreta.

Justificativa: circuitos aritméticos, como é o caso dos circuitos somadores que vemos na figura, são circuitos combinacionais. Isso significa que o resultado de suas saídas depende exclusivamente dos valores atuais das entradas, não importando quais operações foram feitas no passado. Circuitos combinacionais como este são unidirecionais, ou seja, a informação contida nos estados lógicos flui em apenas uma direção, das entradas para as saídas. Circuitos sequenciais são aqueles em que as saídas dependem tanto dos valores atuais das entradas quanto dos valores anteriores das saídas, sendo muito utilizados em memórias (como nos registradores da unidade lógica e aritmética de um processador). Para possuímos o efeito sequencial, temos que realimentar as entradas do circuito com suas próprias saídas. Note que, no circuito da figura, o fato de o *carry-out* do bloco meio somador alimentar o *carry-in* do bloco somador completo adjacente não caracteriza o circuito como sequencial. Nesse caso, a saída de um bloco do circuito alimenta a entrada de outro bloco, gerando um fluxo unidirecional dos sinais.

### II – Afirmativa correta.

Justificativa: o circuito é capaz de somar um número binário de formato  $A_3A_2A_1A_0$  (sendo  $A_3$  o *bit* mais significativo e  $A_0$  o *bit* menos significativo) a outro número binário  $B_3B_2B_1B_0$ . Cada um dos números tem, portanto, 4 *bits*. Cada bloco do circuito será responsável por realizar uma etapa do cálculo: um bloco soma  $A_0$  a  $B_0$ , outro soma  $A_1$  a  $B_1$ , e assim por diante.

### III – Afirmativa incorreta.

Justificativa: o circuito meio somador é responsável por somar os *bits* menos significativos de cada um dos dois números. Ele precisa, portanto, somar  $A_0$  a  $B_0$ . Os *bits* menos significativos são os únicos que trabalham bem com circuitos meio somadores, pois não há etapa anterior de cálculo a ser considerada, ou seja, eles não têm de se preocupar em receber o "vai-um" de um bloco anterior. O meio somador realiza a primeira etapa que faríamos em um cálculo à mão, como mostrado na figura a seguir.

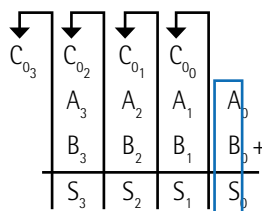


Figura 146



## REFERÊNCIAS

### Textuais

- ABAR, C. *Mapa de Karnaugh*. São Paulo: PUC, 2004. Disponível em: <https://bit.ly/3gh3m08>. Acesso em: 10 jun. 2021.
- ALENCAR FILHO, E. *Iniciação à lógica matemática*. São Paulo: Nobel, 2008.
- ANALOG DEVICES. *LTspice@*. [s.d.]. Disponível em: <https://bit.ly/3gdVRqp>. Acesso em: 10 jun. 2021.
- CAPUANO, F. G. *Sistemas digitais: circuitos combinacionais e sequenciais*. São Paulo: Saraiva, 2014.
- CAPUANO, F. G.; IDOETA, I. V. *Elementos de eletrônica digital*. São Paulo: Érica, 1998.
- FEITOSA, H. A.; PAULOVICH, L. *Um prelúdio à lógica*. São Paulo: Unesp, 2005.
- LAMBERT, S. *Digital circuit simulation*. [s.d.]. Disponível em: <https://bit.ly/3wtQtFH>. Acesso em: 10 jun. 2021.
- LOURENÇO, A. C. et al. *Circuitos digitais: estude e use*. São Paulo: Érica, 2008.
- MATHEMATIK.UNI. *Karnaugh-Veitch Map*. [s.d.]. Disponível em: <https://bit.ly/35jepiJ>. Acesso em: 10 jun. 2021.
- MOKARZEL, F.; SOMA, N. *Introdução à ciência da computação*. Rio de Janeiro: Campus, 2008.
- PROFELECTRO.INFO. *Mapa de Karnaugh online: para simplificação de funções booleanas a partir da tabela da verdade*. [s.d.]. Disponível em: <https://bit.ly/3znyJNK>. Acesso em: 10 jun. 2021.
- SEBESTA, R. W. *Conceitos de linguagens de programação*. Porto Alegre: Bookman, 2011.
- SHANNON, C. E. A symbolic analysis of relay and switching circuits. *Transactions American Institute of Electrical Engineers*, v. 57, 1938. Disponível em: <https://bit.ly/3whYPQk>. Acesso em: 9 jun. 2021.
- VEITCH, E. A chart method for simplifying truth functions. *ACM*, v. 52, 1952. Disponível em: <https://bit.ly/3gpGlkZ>. Acesso em: 9 jun. 2021.
- W3SCHOOLS. *Python Data Types*. [s.d.]a. Disponível em: <https://bit.ly/2Tr7jGA>. Acesso em: 10 jun. 2021.
- W3SCHOOLS. *Python Operators*. [s.d.]b. Disponível em: <https://bit.ly/2Tr7jGA>. Acesso em: 10 jun. 2021.
- WAGNER, F. R.; REIS, A. I.; RIBAS, R. P. *Fundamentos de circuitos digitais*. Porto Alegre: Bookman, 2008.
- WATANABE, O. K. *Iniciação à lógica matemática*. Embu das Artes: Alexa Cultural, 2010.

WIKIMEDIA COMMONS. *ANSI logic gates*. [s.d.]a. Disponível em: <https://bit.ly/3cA1Xj0>. Acesso em: 9 jun. 2021.

WIKIMEDIA COMMONS. *Seven segment display*. [s.d.]b. Disponível em: <https://bit.ly/3wgf8NJ>. Acesso em: 9 jun. 2021.



Handwriting practice lines consisting of 30 horizontal blue lines. Each line is preceded by a small blue dot, serving as a guide for letter height and placement.



A series of horizontal lines for writing, consisting of 30 evenly spaced lines across the page.



Handwriting practice lines consisting of 30 horizontal blue lines. Each line is preceded by a small blue vertical margin line on the left side.



Handwriting practice lines consisting of 30 horizontal blue lines. The first line is a solid blue line, and the subsequent 29 lines are pairs of dashed blue lines, providing a guide for letter height and placement.





# Interativa

Informações:  
[www.sepi.unip.br](http://www.sepi.unip.br) ou 0800 010 9000