

# Circuitos Lógicos



## Módulo # 3

### Circuitos Combinacionais



#### Como implementar os circuitos descritos por uma tabela verdade



| A | B | C | F | $\bar{F}$ |
|---|---|---|---|-----------|
| 0 | 0 | 0 | 0 | 1         |
| 0 | 0 | 1 | 0 | 1         |
| 0 | 1 | 0 | 0 | 1         |
| 0 | 1 | 1 | 1 | 0         |
| 1 | 0 | 0 | 1 | 0         |
| 1 | 0 | 1 | 1 | 0         |
| 1 | 1 | 0 | 1 | 0         |
| 1 | 1 | 1 | 1 | 0         |

... Perguntamos à tabela, por exemplo, onde a função F vale 1.  
Qual seria a resposta em linguagem natural ?

| A | B | C | F | $\bar{F}$ |
|---|---|---|---|-----------|
| 0 | 0 | 0 | 0 | 1         |
| 0 | 0 | 1 | 0 | 1         |
| 0 | 1 | 0 | 0 | 1         |
| 0 | 1 | 1 | 1 | 0         |
| 1 | 0 | 0 | 1 | 0         |
| 1 | 0 | 1 | 1 | 0         |
| 1 | 1 | 0 | 1 | 0         |
| 1 | 1 | 1 | 1 | 0         |

Diagram showing connections from the F=1 rows to the output F:

- Row 4 (0,1,1) connects to F=1
- Row 5 (1,0,0) connects to F=1
- Row 6 (1,0,1) connects to F=1
- Row 7 (1,1,0) connects to F=1
- Row 8 (1,1,1) connects to F=1

F = 1 quando  
A=0 e B=1 e C=1  
ou  
A=1 e B=0 e C=0  
ou  
A=1 e B=0 e C=1  
ou  
A=1 e B=1 e C=0  
ou  
A=1 e B=1 e C=1



## Como implementar os circuitos descritos por uma tabela verdade



| A | B | C | F | $\bar{F}$ |
|---|---|---|---|-----------|
| 0 | 0 | 0 | 0 | 1         |
| 0 | 0 | 1 | 0 | 1         |
| 0 | 1 | 0 | 0 | 1         |
| 0 | 1 | 1 | 1 | 0         |
| 1 | 0 | 0 | 1 | 0         |
| 1 | 0 | 1 | 1 | 0         |
| 1 | 1 | 0 | 1 | 0         |
| 1 | 1 | 1 | 1 | 0         |

F = 1 quando  
 A=0 e B=1 e C=1  
 ou  
 A=1 e B=0 e C=0  
 ou  
 A=1 e B=0 e C=1  
 ou  
 A=1 e B=1 e C=0  
 ou  
 A=1 e B=1 e C=1

Funções lógicas

F = 1 quando  
 $A=0 \cdot B=1 \cdot C=1$   
 +  
 $A=1 \cdot B=0 \cdot C=0$   
 +  
 $A=1 \cdot B=0 \cdot C=1$   
 +  
 $A=1 \cdot B=1 \cdot C=0$   
 +  
 $A=1 \cdot B=1 \cdot C=1$

... e, finalmente,  $F = A'.B.C + A.B'.C' + A.B'.C + A.B.C' + A.B.C$



## Como implementar os circuitos descritos por uma tabela verdade



| A | B | C | F | $\bar{F}$ |
|---|---|---|---|-----------|
| 0 | 0 | 0 | 0 | 1         |
| 0 | 0 | 1 | 0 | 1         |
| 0 | 1 | 0 | 0 | 1         |
| 0 | 1 | 1 | 1 | 0         |
| 1 | 0 | 0 | 1 | 0         |
| 1 | 0 | 1 | 1 | 0         |
| 1 | 1 | 0 | 1 | 0         |
| 1 | 1 | 1 | 1 | 0         |

$F = A'.B.C + A.B'.C' + A.B'.C + A.B.C' + A.B.C$

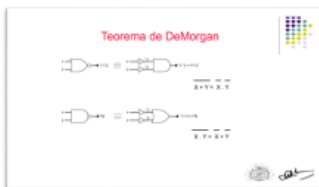
... poderíamos fazer também ...

$F' = A'.B'.C' + A'.B'.C + A'.B.C'$

... complementando ambos os lados da expressão...

$F = (A'.B'.C' + A'.B'.C + A'.B.C')'$

... lembram do teorema de DeMorgan ?





... por DeMorgan temos que  $(X + Y)' = X' \cdot Y'$ . Então ...

$$F = (A' \cdot B' \cdot C' + A' \cdot B' \cdot C + A' \cdot B \cdot C')' = (A' \cdot B' \cdot C')' \cdot (A' \cdot B' \cdot C)' \cdot (A' \cdot B \cdot C)'$$

... também por DeMorgan temos que  $(X \cdot Y \cdot Z)' = X' + Y' + Z'$ . Então, finalmente, ...

$$F = (A + B + C) \cdot (A + B + C') \cdot (A + B' + C) \quad \text{-----> EXPRESSÕES EQUIVALENTES}$$

| A | B | C | F | $\bar{F}$ |
|---|---|---|---|-----------|
| 0 | 0 | 0 | 0 | 1         |
| 0 | 0 | 1 | 0 | 1         |
| 0 | 1 | 0 | 0 | 1         |
| 0 | 1 | 1 | 1 | 0         |
| 1 | 0 | 0 | 1 | 0         |
| 1 | 0 | 1 | 1 | 0         |
| 1 | 1 | 0 | 1 | 0         |
| 1 | 1 | 1 | 1 | 0         |

$$F = A' \cdot B \cdot C + A \cdot B' \cdot C' + A \cdot B' \cdot C + A \cdot B \cdot C' + A \cdot B \cdot C$$



### Nomenclatura



$$F = A' \cdot B \cdot C + A \cdot B' \cdot C' + A \cdot B' \cdot C + A \cdot B \cdot C' + A \cdot B \cdot C$$

$$F = (A + B + C) \cdot (A + B + C') \cdot (A + B' + C)$$

**Soma de produtos - Forma canônica de mintermos**

**Produto de somas - Forma canônica de Maxtermos**

| A | B | C | mintermos                       |
|---|---|---|---------------------------------|
| 0 | 0 | 0 | $\bar{A} \bar{B} \bar{C} = m_0$ |
| 0 | 0 | 1 | $\bar{A} \bar{B} C = m_1$       |
| 0 | 1 | 0 | $\bar{A} B \bar{C} = m_2$       |
| 0 | 1 | 1 | $\bar{A} B C = m_3$             |
| 1 | 0 | 0 | $A \bar{B} \bar{C} = m_4$       |
| 1 | 0 | 1 | $A \bar{B} C = m_5$             |
| 1 | 1 | 0 | $A B \bar{C} = m_6$             |
| 1 | 1 | 1 | $A B C = m_7$                   |

| A | B | C | F | $\bar{F}$ |
|---|---|---|---|-----------|
| 0 | 0 | 0 | 0 | 1         |
| 0 | 0 | 1 | 0 | 1         |
| 0 | 1 | 0 | 0 | 1         |
| 0 | 1 | 1 | 1 | 0         |
| 1 | 0 | 0 | 1 | 0         |
| 1 | 0 | 1 | 1 | 0         |
| 1 | 1 | 0 | 1 | 0         |
| 1 | 1 | 1 | 1 | 0         |

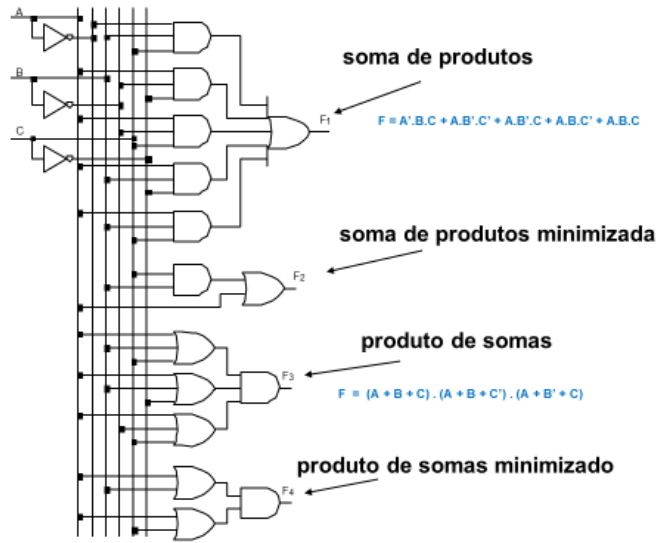
| A | B | C | Maxtermos         |
|---|---|---|-------------------|
| 0 | 0 | 0 | $A + B + C = M_0$ |
| 0 | 0 | 1 | $A + B + C = M_1$ |
| 0 | 1 | 0 | $A + B + C = M_2$ |
| 0 | 1 | 1 | $A + B + C = M_3$ |
| 1 | 0 | 0 | $A + B + C = M_4$ |
| 1 | 0 | 1 | $A + B + C = M_5$ |
| 1 | 1 | 0 | $A + B + C = M_6$ |
| 1 | 1 | 1 | $A + B + C = M_7$ |

$$F = m_3 + m_4 + m_5 + m_6 + m_7$$

$$F = M_0 \cdot M_1 \cdot M_2 = (A+B+C) \cdot (A+B+C') \cdot (A+B'+C)$$



Como, afinal, devemos implementar a função F ?



*Handwritten signature*

... como podemos, então, minimizar a função F ?

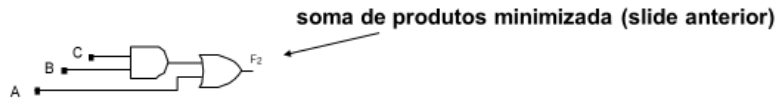
$$F = A'.B.C + A.B'.C' + A.B'.C + A.B.C' + A.B.C$$



... procurando dentro da função termos adjacentes, isto é, entre eles uma única variável troca de valor e é eliminada.

$$\Rightarrow A.B.C' + A.B.C \rightarrow A.B.(C' + C) = A.B$$

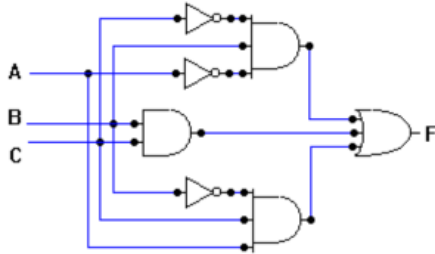
... teremos então  $F = A'.B.C + A.B' + A.B = A'.B.C + A = A + B.C$



*Handwritten signature*

**Exercício 1** : Simplificar as seguintes funções lógicas

$L(A,B,C,D) = m(0,2,5,7,8,10,13,15)$



**Exercício 2** : Implementar os circuitos acima utilizando apenas portas NAND



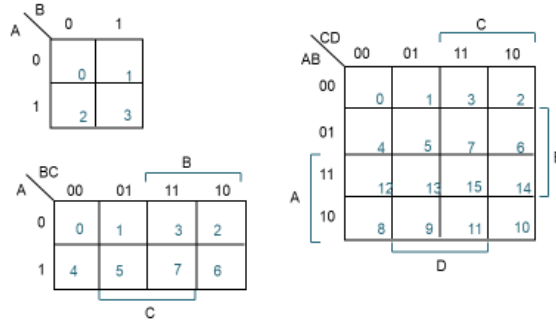
## Mapas de Karnaugh : algoritmo

- 1 - Construir o mapa ;
- 2 - Fazer o menor número possível de agrupamentos ;
- 3 - Dentro de cada agrupamento realizado, deve haver o maior número possível de termos;
- 4 - Começar os agrupamentos pelos termos que têm apenas uma possibilidade de fazê-lo.



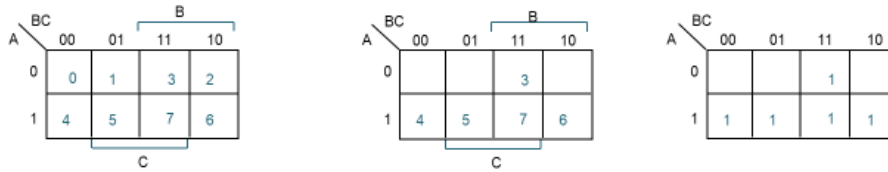


## Mapas de Karnaugh



## Mapas de Karnaugh : exemplo

$$F = A'.B.C + A.B'.C' + A.B'.C + A.B.C' + A.B.C = m3 + m4 + m5 + m6 + m7$$



Concluindo, lembrando o algoritmo, ...

**Mapas de Karnaugh : algoritmo**

- 1- Construir o mapa;
- 2- Fazer o menor número possível de agrupamentos;
- 3- Dentro de cada agrupamento realizado, deve haver o maior número possível de termos;
- 4- Começar os agrupamentos pelos termos que têm apenas uma possibilidade de fazê-lo.



Simplificar a expressão abaixo:


$$F(A,B,C,D) = m(0,2,5,8,9,10,11,12,13,14,15)$$

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

... lembrando o algoritmo, ...

Mapas de Karnaugh : algoritmo

- 1- Construir o mapa ;
- 2- Fazer o menor número possível de agrupamentos ;
- 3- Dentro de cada agrupamento realizado, deve fazer o maior número possível de termos ;
- 4- Começar os agrupamentos pelos termos que têm apenas uma possibilidade de factis.



Simplificar a expressão abaixo:

$$F(A,B,C,D) = m(0,2,5,8,9,10,11,12,13,14,15)$$

... poderia ser escrita como

$$F(A,B,C,D) = M(1,3,6,7)$$


... e aí ?

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

... lembrando o algoritmo, ...

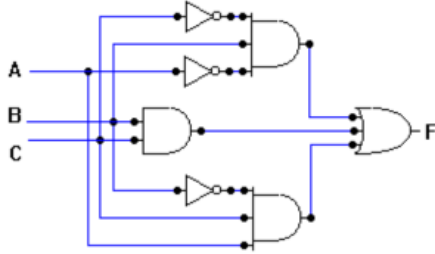
Mapas de Karnaugh : algoritmo

- 1- Construir o mapa ;
- 2- Fazer o menor número possível de agrupamentos ;
- 3- Dentro de cada agrupamento realizado, deve fazer o maior número possível de termos ;
- 4- Começar os agrupamentos pelos termos que têm apenas uma possibilidade de factis.



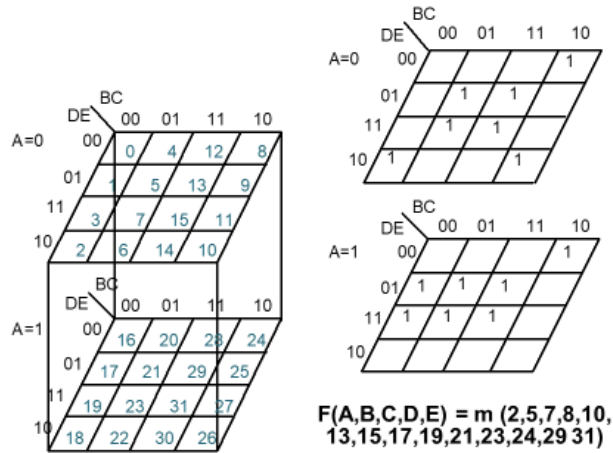
**Exercício :** Simplificar as seguintes funções lógicas, utilizando MK

$$L(A,B,C,D) = m(0,2,5,7,8,10,13,15)$$



*Handwritten signature*

**Outros mapas, com maior número de variáveis ...**



*Handwritten signature*





## Projeto de Circuitos Combinacionais

- Entendimento do problema
- Análise das informações
- Minimização da expressão
- Minimização do circuito \*
- Testes

\* pode variar, dependendo da implementação



## **Circuitos combinacionais utilizando códigos binários**

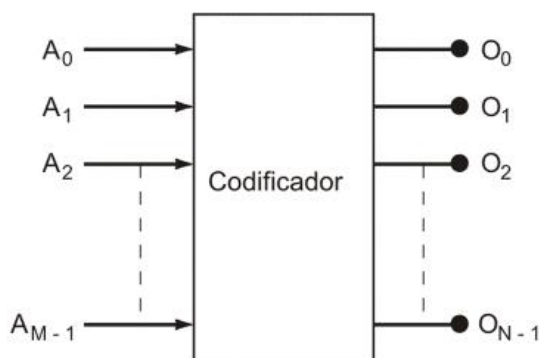
**Codificadores**

**Decodificadores**

**Conversores de códigos**

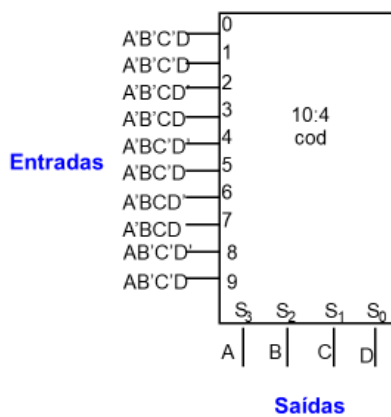


**Codificadores**



*Handwritten signature*

**Codificador decimal => binário**

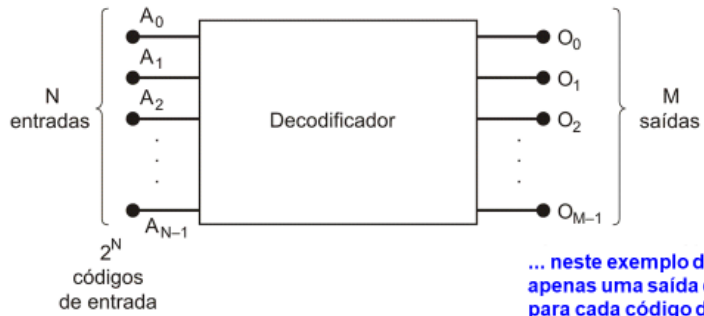


**Que circuito implementa a função ?**



*Handwritten signature*

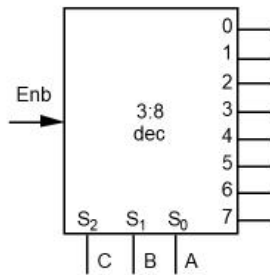
## Decodificadores



... neste exemplo de projeto, apenas uma saída é ativada para cada código de entrada.



Projeto 2 : Decodificador binário (3 bits) => oito saídas

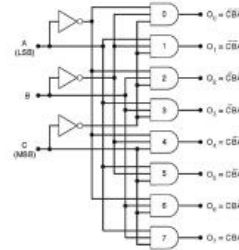
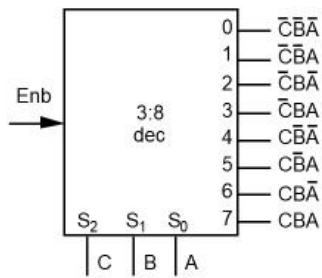


Que circuito implementa a função ?

O que é o Enb ?



Projeto 2 : Decodificador binário (3 bits) => oito saídas

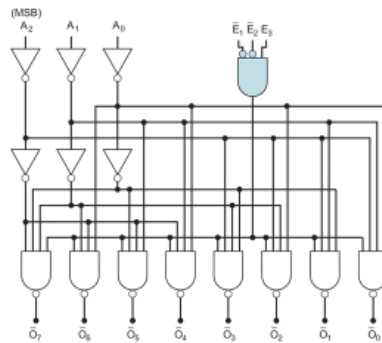


| C | B | A | O <sub>7</sub> | O <sub>6</sub> | O <sub>5</sub> | O <sub>4</sub> | O <sub>3</sub> | O <sub>2</sub> | O <sub>1</sub> | O <sub>0</sub> |
|---|---|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              |
| 0 | 0 | 1 | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              |
| 0 | 1 | 0 | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              |
| 0 | 1 | 1 | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              |
| 1 | 0 | 0 | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              |
| 1 | 0 | 1 | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              |
| 1 | 1 | 0 | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              |
| 1 | 1 | 1 | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |

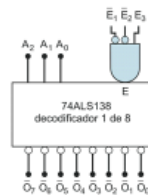
... e o Enb ?



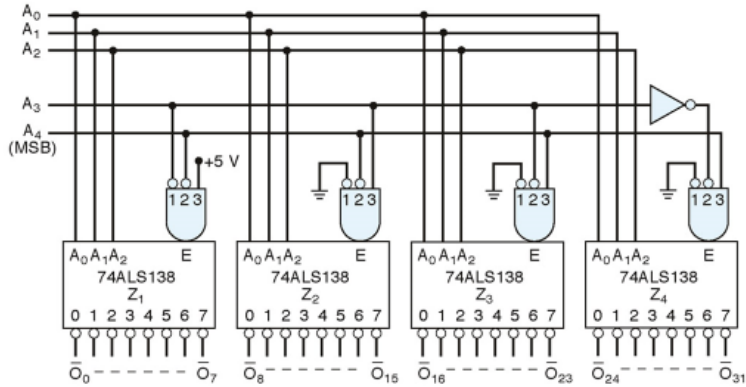
Dec 3:8 74ALS138



| E <sub>1</sub> | E <sub>2</sub> | E <sub>3</sub> | Saídas   |
|----------------|----------------|----------------|--|
| 0              | 0              | 1              | Responde ao código de entrada A <sub>2</sub> A <sub>1</sub> A <sub>0</sub> |
| 1              | X              | X              | Desabilitadas - todas em nível ALTO  |
| X              | 1              | X              | Desabilitadas - todas em nível ALTO  |
| X              | X              | 0              | Desabilitadas - todas em nível ALTO  |



Quatro CIs 74ALS138 formando um decodificador 5:32

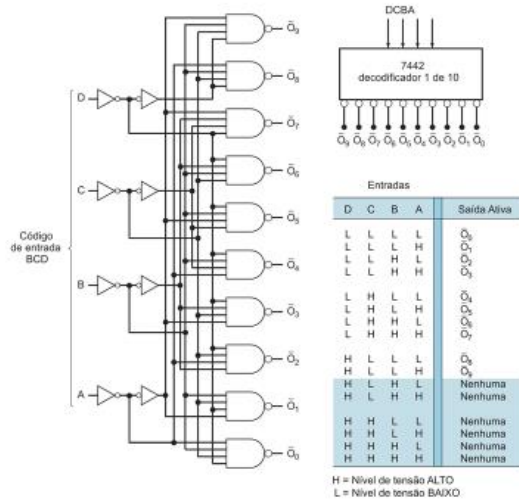


... explicar.



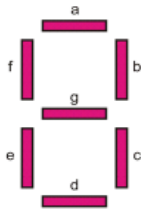
*Handwritten signature*

Decodificador BCD : decimal



*Handwritten signature*

## Display de 7 segmentos



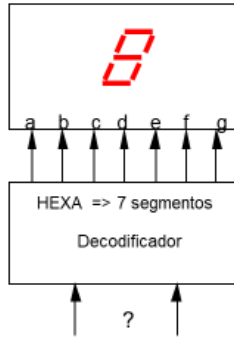
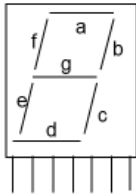
Números na base 16



Handwritten signature.

## Projeto de Circuitos Combinacionais

Projetar um Decodificador Hexadecimal => 7 segmentos



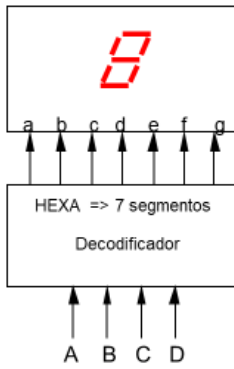
entradas



Handwritten signature.

**Projeto de Circuitos Combinacionais**

**Projetar um Decodificador Hexadecimal => 7 segmentos**



entradas

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

a

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

b

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

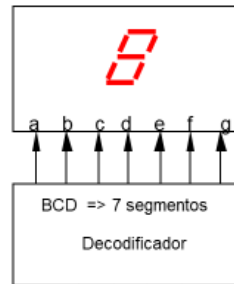
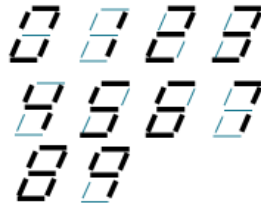
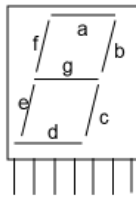
c



*Handwritten signature*

**Projeto de Circuitos Combinacionais**

**Decodificador BCD => 7 segmentos**



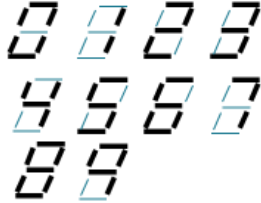
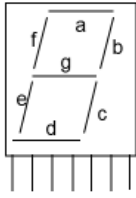
entradas

**O que muda no projeto ?**



*Handwritten signature*

**Projeto de Circuitos Combinacionais**  
**Decodificador BCD => 7 segmentos**



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

a

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

b

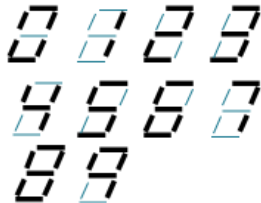
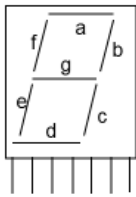
| AB \ CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 1  | 1  | X  | 1  |
| 01      | 1  | 1  | X  | 1  |
| 11      | 1  | 1  | X  | X  |
| 10      | 0  | 1  | X  | X  |

c



*Handwritten signature*

**Projeto de Circuitos Combinacionais**  
**Decodificador BCD => 7 segmentos**



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

a

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

b

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

c

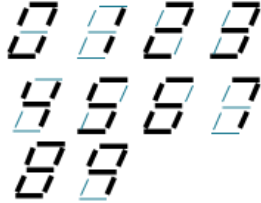
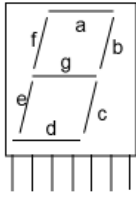


*Handwritten signature*



## Projeto de Circuitos Combinacionais

### Decodificador BCD => 7 segmentos



#### Projeto de Circuitos Combinacionais

- Entendimento do problema
- Análise das informações
- Minimização da expressão
- Minimização do circuito \*
- Testes

\* pode variar, dependendo da implementação

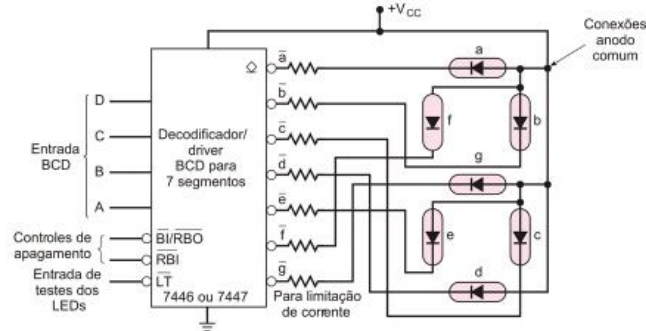
$$\begin{aligned} a &= A + B D + C + B' D' \\ b &= A + C' D' + C D + B' \\ c &= A + B + C' + D \end{aligned}$$

- . é esse o circuito que vamos implementar ?
- . quantos CIs de portas serão necessários ?

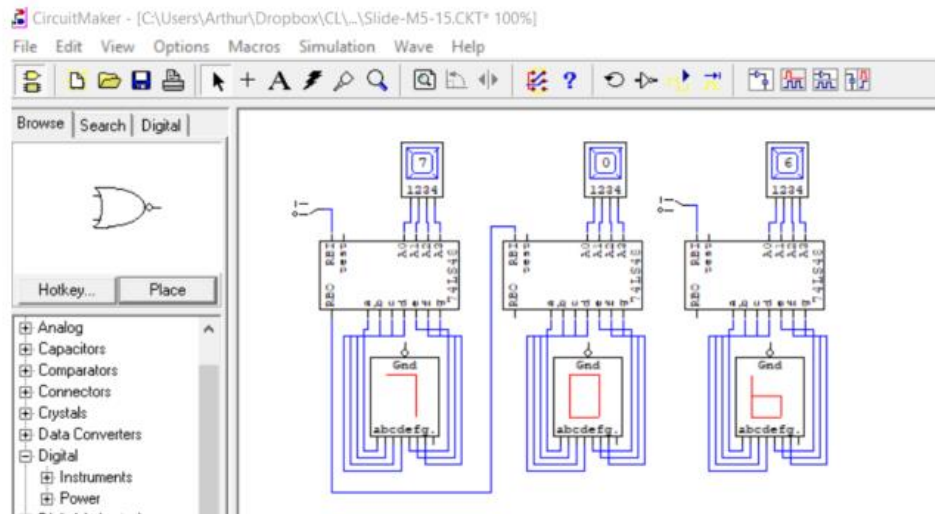


*Handwritten signature*

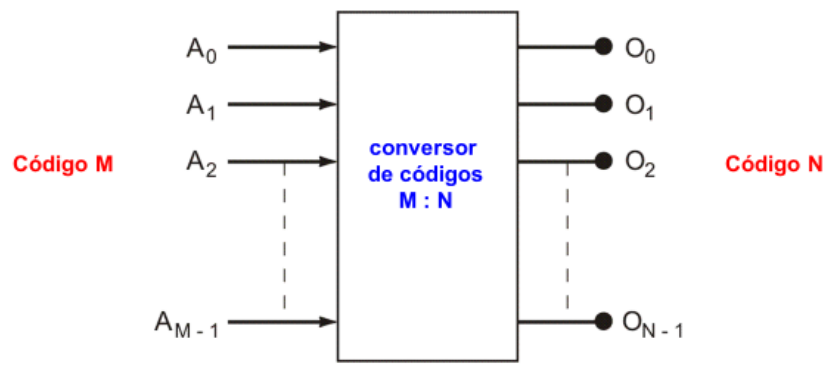
### Decodificador/driver BCD para 7 segmentos para display de LEDs de 7 segmentos tipo anodo comum



*Handwritten signature*



**Conversores de códigos**



## Código ASCII

TABELA 2.4 Códigos ASCII padrão.

| Caractere        | HEX | Decimal | Caractere | HEX | Decimal | Caractere | HEX | Decimal | Caractere | HEX | Decimal |
|------------------|-----|---------|-----------|-----|---------|-----------|-----|---------|-----------|-----|---------|
| NUL (null)       | 0   | 0       | Space     | 20  | 32      | @         | 40  | 64      |           | 60  | 96      |
| Start Heading    | 1   | 1       | !         | 21  | 33      | A         | 41  | 65      | a         | 61  | 97      |
| Start Text       | 2   | 2       | "         | 22  | 34      | B         | 42  | 66      | b         | 62  | 98      |
| End Text         | 3   | 3       | #         | 23  | 35      | C         | 43  | 67      | c         | 63  | 99      |
| End Transmit     | 4   | 4       | \$        | 24  | 36      | D         | 44  | 68      | d         | 64  | 100     |
| Enquiry          | 5   | 5       | %         | 25  | 37      | E         | 45  | 69      | e         | 65  | 101     |
| Acknowledge      | 6   | 6       | &         | 26  | 38      | F         | 46  | 70      | f         | 66  | 102     |
| Bell             | 7   | 7       | '         | 27  | 39      | G         | 47  | 71      | g         | 67  | 103     |
| Backspace        | 8   | 8       | (         | 28  | 40      | H         | 48  | 72      | h         | 68  | 104     |
| Horiz Tab        | 9   | 9       | )         | 29  | 41      | I         | 49  | 73      | i         | 69  | 105     |
| Line Feed        | A   | 10      | *         | 2A  | 42      | J         | 4A  | 74      | j         | 6A  | 106     |
| Vert. Tab        | B   | 11      | +         | 2B  | 43      | K         | 4B  | 75      | k         | 6B  | 107     |
| Form Feed        | C   | 12      | ,         | 2C  | 44      | L         | 4C  | 76      | l         | 6C  | 108     |
| Carriage Return  | D   | 13      | -         | 2D  | 45      | M         | 4D  | 77      | m         | 6D  | 109     |
| Shift Out        | E   | 14      | .         | 2E  | 46      | N         | 4E  | 78      | n         | 6E  | 110     |
| Shift In         | F   | 15      | /         | 2F  | 47      | O         | 4F  | 79      | o         | 6F  | 111     |
| Data Link Esc    | 10  | 16      | 0         | 30  | 48      | P         | 50  | 80      | p         | 70  | 112     |
| Direct Control 1 | 11  | 17      | 1         | 31  | 49      | Q         | 51  | 81      | q         | 71  | 113     |
| Direct Control 2 | 12  | 18      | 2         | 32  | 50      | R         | 52  | 82      | r         | 72  | 114     |
| Direct Control 3 | 13  | 19      | 3         | 33  | 51      | S         | 53  | 83      | s         | 73  | 115     |
| Direct Control 4 | 14  | 20      | 4         | 34  | 52      | T         | 54  | 84      | t         | 74  | 116     |
| Negative ACK     | 15  | 21      | 5         | 35  | 53      | U         | 55  | 85      | u         | 75  | 117     |
| Synch Idle       | 16  | 22      | 6         | 36  | 54      | V         | 56  | 86      | v         | 76  | 118     |
| End Trans Block  | 17  | 23      | 7         | 37  | 55      | W         | 57  | 87      | w         | 77  | 119     |
| Cancel           | 18  | 24      | 8         | 38  | 56      | X         | 58  | 88      | x         | 78  | 120     |
| End of Medium    | 19  | 25      | 9         | 39  | 57      | Y         | 59  | 89      | y         | 79  | 121     |
| Substitute       | 1A  | 26      | :         | 3A  | 58      | Z         | 5A  | 90      | z         | 7A  | 122     |
| Escape           | 1B  | 27      | ;         | 3B  | 59      | [         | 5B  | 91      | {         | 7B  | 123     |
| Form separator   | 1C  | 28      | <         | 3C  | 60      | \         | 5C  | 92      |           | 7C  | 124     |
| Group separator  | 1D  | 29      | =         | 3D  | 61      | ]         | 5D  | 93      | }         | 7D  | 125     |
| Record Separator | 1E  | 30      | >         | 3E  | 62      | ^         | 5E  | 94      | ~         | 7E  | 126     |
| Unit Separator   | 1F  | 31      | ?         | 3F  | 63      | _         | 5F  | 95      | Delete    | 7F  | 127     |



*Handwritten signature*

## Código de Gray

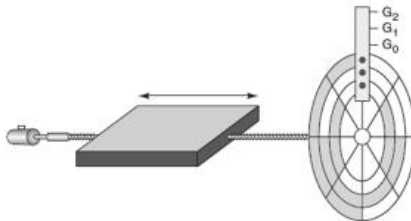


FIGURA 2.3 Codificador de posição (encoder) de eixo de três bits e oito posições.

TABELA 2.3 Representação dos números decimais.

| Decimal | Binário | Hexadecimal | BCD       | GRAY |
|---------|---------|-------------|-----------|------|
| 0       | 0       | 0           | 0000      | 0000 |
| 1       | 1       | 1           | 0001      | 0001 |
| 2       | 10      | 2           | 0010      | 0011 |
| 3       | 11      | 3           | 0011      | 0010 |
| 4       | 100     | 4           | 0100      | 0110 |
| 5       | 101     | 5           | 0101      | 0111 |
| 6       | 110     | 6           | 0110      | 0101 |
| 7       | 111     | 7           | 0111      | 0100 |
| 8       | 1000    | 8           | 1000      | 1100 |
| 9       | 1001    | 9           | 1001      | 1101 |
| 10      | 1010    | A           | 0001 0000 | 1111 |
| 11      | 1011    | B           | 0001 0001 | 1110 |
| 12      | 1100    | C           | 0001 0010 | 1010 |
| 13      | 1101    | D           | 0001 0011 | 1011 |
| 14      | 1110    | E           | 0001 0100 | 1001 |
| 15      | 1111    | F           | 0001 0101 | 1000 |



*Handwritten signature*

Projetar um conversor BCD => Código de Gray



tabela verdade

| A | B | C | D | W | X | Y | Z |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X |

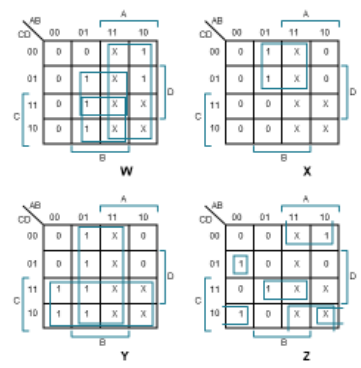


Projetar um conversor BCD => Código de Gray



tabela verdade

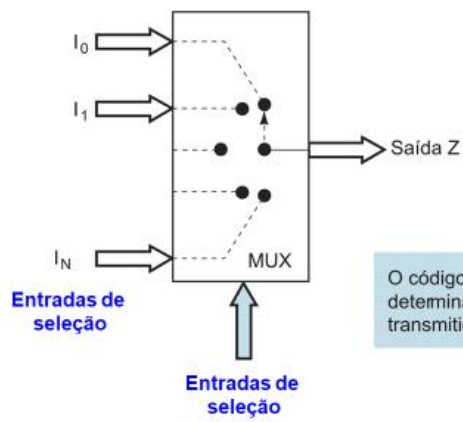
| A | B | C | D | W | X | Y | Z |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X |



$$\begin{aligned}
 W &= A + BD + BC \\
 X &= B'C \\
 Y &= B + C \\
 Z &= A'B'C'D + BCD + AD' + B'C'D'
 \end{aligned}$$



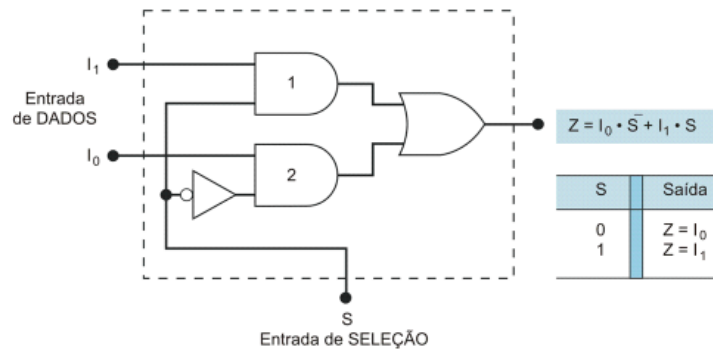
## Multiplexadores



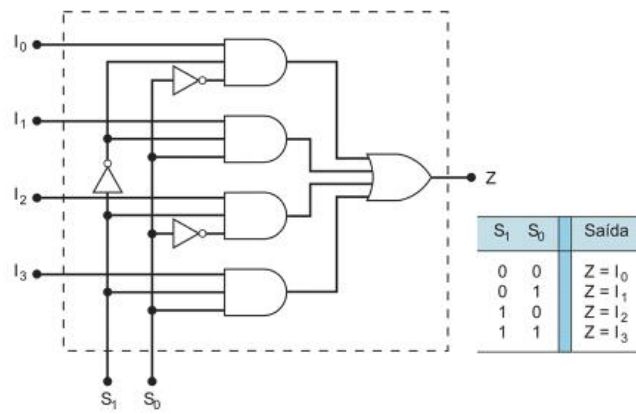
O código na entrada de SELEÇÃO determina a entrada que é transmitida para a saída Z



*Handwritten signature*

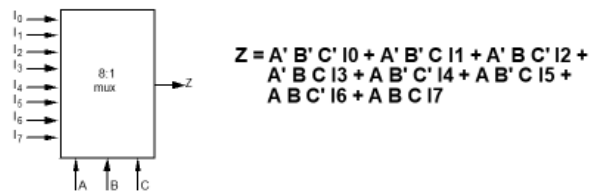
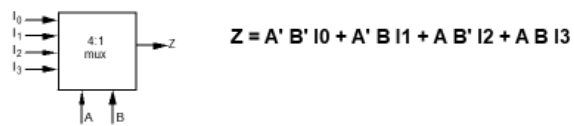
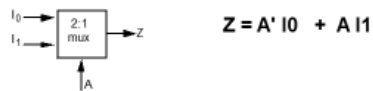


*Handwritten signature*



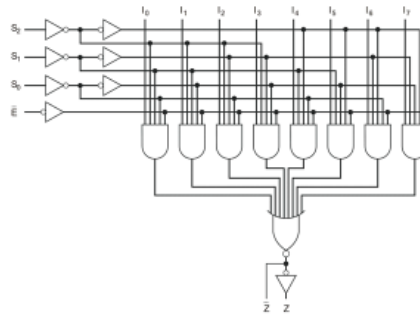
Handwritten signature.

### Multiplexadores (seletores)

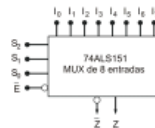


Handwritten signature.

### Diagrama lógico do multiplexador 74ALS151



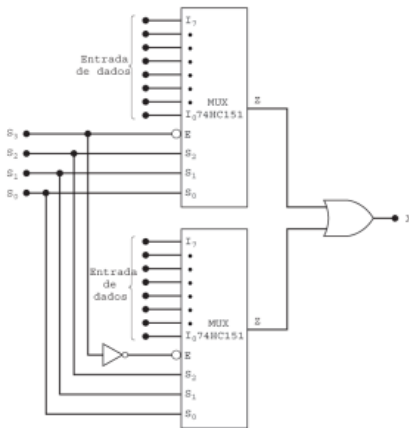
| Inputs |                |                |                | Saída |                |
|--------|----------------|----------------|----------------|-------|----------------|
| E      | S <sub>2</sub> | S <sub>1</sub> | S <sub>0</sub> | Z     | Z              |
| H      | X              | X              | X              | H     | I <sub>0</sub> |
| L      | L              | L              | L              | H     | I <sub>1</sub> |
| L      | L              | L              | H              | L     | I <sub>2</sub> |
| L      | L              | H              | L              | L     | I <sub>3</sub> |
| L      | L              | H              | H              | L     | I <sub>4</sub> |
| L      | H              | L              | L              | L     | I <sub>5</sub> |
| L      | H              | L              | H              | L     | I <sub>6</sub> |
| L      | H              | H              | H              | L     | I <sub>7</sub> |



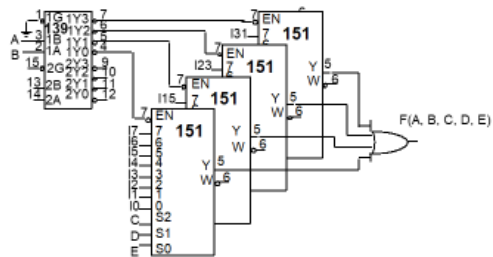
*Handwritten signature*



### 74HC151 combinados para formar um multiplexador de 16 entradas



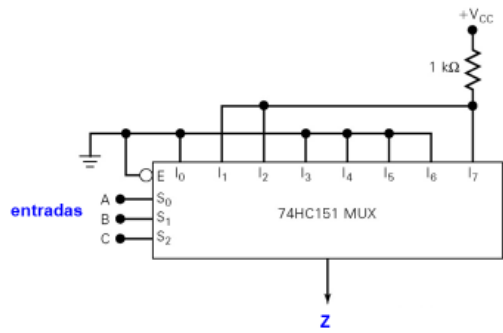
### 74HC151 combinados para formar um multiplexador de 32 entradas (observe o decodificador 74139)



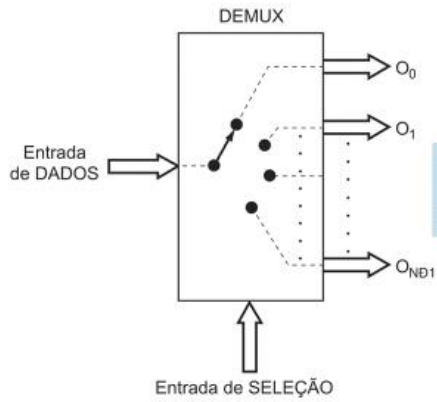
*Handwritten signature*



Analisar o circuito abaixo



### Demultiplexador

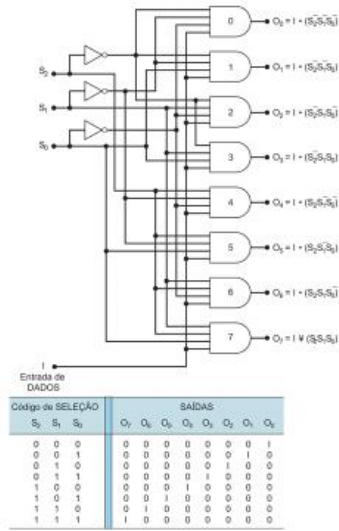


A entrada de DADOS é transmitida apenas para uma das saídas, conforme determinado pelo código de seleção de entrada.



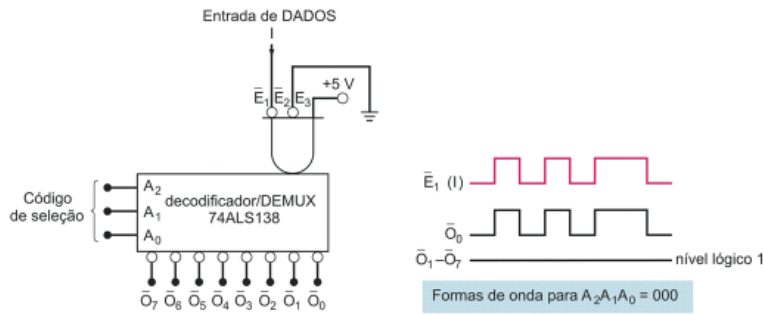


## Demultiplexador de 1 para 8 linhas



*Handwritten signature*

## O decodificador 74ALS138 como um demultiplexador com E<sub>1</sub> usada como entrada de dado



*Handwritten signature*

## Comparador de dois números



*Handwritten signature*

## Comparador de magnitude de quatro bits 74HC85 (7485, 74LS85)

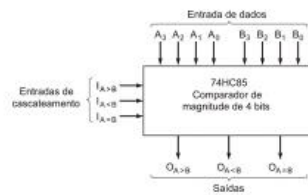


TABELA-VERDADE

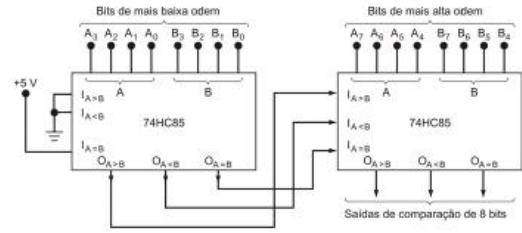
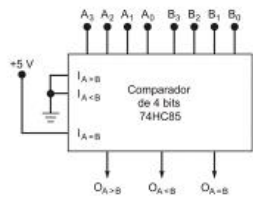
| ENTRADAS DE COMPARAÇÃO         |                                |                                |                                | ENTRADAS DE CASCATEAMENTO |                  |                     | SAÍDAS              |                  |                     |
|--------------------------------|--------------------------------|--------------------------------|--------------------------------|---------------------------|------------------|---------------------|---------------------|------------------|---------------------|
| A <sub>3</sub> B <sub>3</sub>  | A <sub>2</sub> B <sub>2</sub>  | A <sub>1</sub> B <sub>1</sub>  | A <sub>0</sub> B <sub>0</sub>  | I <sub>a&gt;b</sub>       | I <sub>a=b</sub> | I <sub>a&lt;b</sub> | O <sub>a&gt;b</sub> | O <sub>a=b</sub> | O <sub>a&lt;b</sub> |
| A <sub>3</sub> >B <sub>3</sub> | X                              | X                              | X                              | X                         | X                | X                   | H                   | L                | L                   |
| A <sub>3</sub> <B <sub>3</sub> | X                              | X                              | X                              | X                         | X                | X                   | L                   | H                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> >B <sub>2</sub> | X                              | X                              | X                         | X                | X                   | H                   | L                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | X                              | X                              | X                         | X                | X                   | L                   | H                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | A <sub>1</sub> >B <sub>1</sub> | X                              | X                         | X                | X                   | H                   | L                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | A <sub>1</sub> <B <sub>1</sub> | X                              | X                         | X                | X                   | L                   | H                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | A <sub>1</sub> =B <sub>1</sub> | A <sub>0</sub> >B <sub>0</sub> | X                         | X                | X                   | H                   | L                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | A <sub>1</sub> =B <sub>1</sub> | A <sub>0</sub> <B <sub>0</sub> | X                         | X                | X                   | L                   | H                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | A <sub>1</sub> =B <sub>1</sub> | A <sub>0</sub> =B <sub>0</sub> | H                         | L                | L                   | H                   | L                | L                   |
| A <sub>3</sub> >B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | A <sub>1</sub> =B <sub>1</sub> | A <sub>0</sub> =B <sub>0</sub> | L                         | H                | L                   | L                   | H                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> >B <sub>2</sub> | A <sub>1</sub> =B <sub>1</sub> | A <sub>0</sub> =B <sub>0</sub> | X                         | X                | H                   | L                   | L                | H                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | A <sub>1</sub> >B <sub>1</sub> | A <sub>0</sub> =B <sub>0</sub> | L                         | L                | L                   | H                   | H                | L                   |
| A <sub>3</sub> =B <sub>3</sub> | A <sub>2</sub> =B <sub>2</sub> | A <sub>1</sub> =B <sub>1</sub> | A <sub>0</sub> =B <sub>0</sub> | H                         | H                | L                   | L                   | L                | L                   |

H = Nível de tensão ALTO  
L = Nível de tensão BAIXO  
X = Irrelevante



*Handwritten signature*

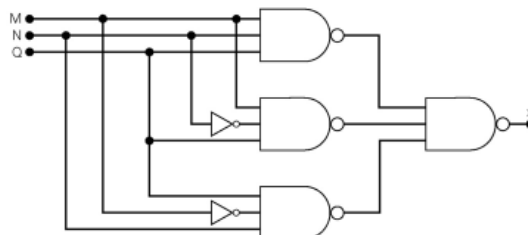
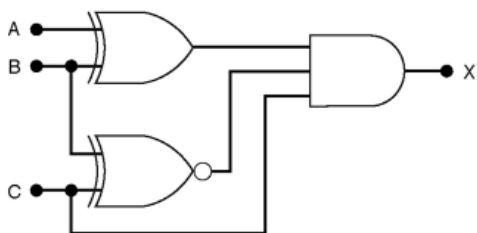
Dois CIs 74HC85 cascadeados para formar um comparador de oito bits.



*Handwritten signature or mark.*

## Exercícios

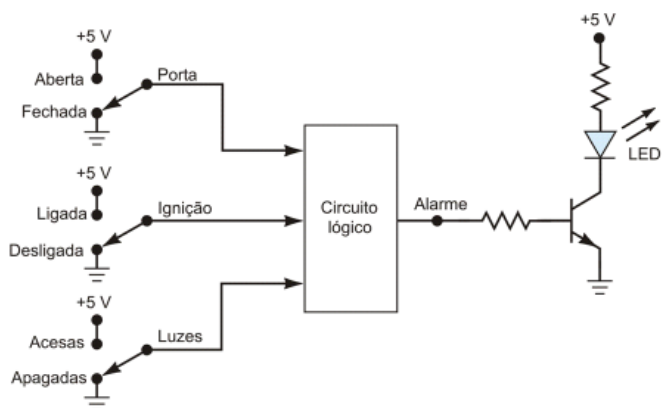
Analisar os circuitos abaixo e implementá-los com um menor número de CIs.



*Handwritten signature*

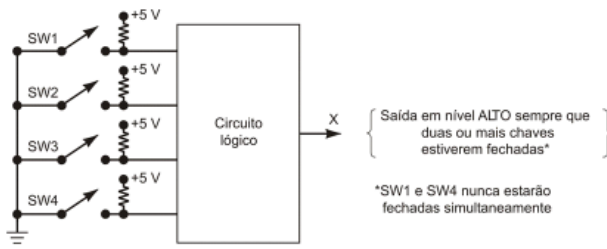
Seja o circuito abaixo um alarme de automóvel. Projetar o circuito lógico para dar alarme quando:

- a) os faróis estão acesos e a ignição está desligada
- b) a porta está aberta e a ignição está ligada



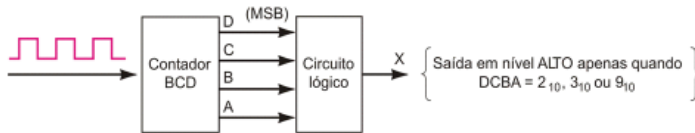
*Handwritten signature*

Projetar o circuito que atenda à função X



Projetar o circuito lógico que atenda à função X

- utilizando circuitos discretos ;
- utilizando um MUX 4:1 e portas.



## # Leitura indicada



Maini, A.K. "Digital Electronics – Principles and Integrated Circuits"

a) Sec. 5.1 – 5.6 , pgs. 161 – 195

b) Sec. 7.1 – 7.5 , pgs. 239 – 262

### 5

#### Boolean Algebra and Simplification Techniques

##### LEARNING OBJECTIVES

After completing this chapter, you will learn the following:

- Variables and literals of a Boolean expression.
- Equivalent, dual, and complement of a Boolean expression.
- Postulates of Boolean algebra.
- Laws and theorems of Boolean algebra including commutative law, associative law, distributive law, idempotent law, involution law, absorption law, transposition theorem, and DeMorgan's theorem.
- Sums-of-products and product-of-sums Boolean expressions.
- $\Sigma$  and  $\Pi$  notations of writing Boolean expressions.
- Expanded form of Boolean expressions.
- Karnaugh map method of simplifying Boolean expressions.
- Quine–McCluskey method of simplifying Boolean expressions.
- Simplification of Boolean expressions with larger number of variables.
- Simplification of multi-output logic functions.

### 7

#### Combinational Logic Circuits

##### LEARNING OBJECTIVES

After completing this chapter, you will learn the following:

- Operational basics of a multiplexer.
- Implementation of Boolean functions with multiplexers.
- Use of multiplexers for parallel-to-serial data conversion.
- Cascading multiplexers.
- Operational basics of demultiplexers and decoders.
- Implementation of Boolean functions with decoders.
- Cascading decoders.
- Encoders and priority encoders.
- Parity generation and checking.
- Application-relevant data on commonly used multiplexers, demultiplexers, and related devices.

