

## CIRCUITOS LÓGICOS COMBINACIONAIS

### I. Objetivos:

- Realizar os passos necessários para obter uma expressão do tipo soma-de-produtos com o objetivo de projetar um circuito lógico na sua forma mais simples.
- Utilizar a álgebra booleana e o mapa de Karnaugh como ferramentas para simplificação e projeto de circuitos lógicos.
- Citar as características básicas de CI's digitais.
- Compreender as diferenças de operação existentes entre circuitos TTL e CMOS.

### II. Definição:

- Circuito lógico combinacional: Circuitos formados por portas lógicas, nos quais o nível lógico do sinal de saída depende, em qualquer instante de tempo, da combinação dos níveis lógicos presentes nas entradas.
- Um circuito combinacional não possui memória, e portanto sua saída depende apenas dos valores atuais das entradas.

### III. Representações das portas lógicas (recordação):

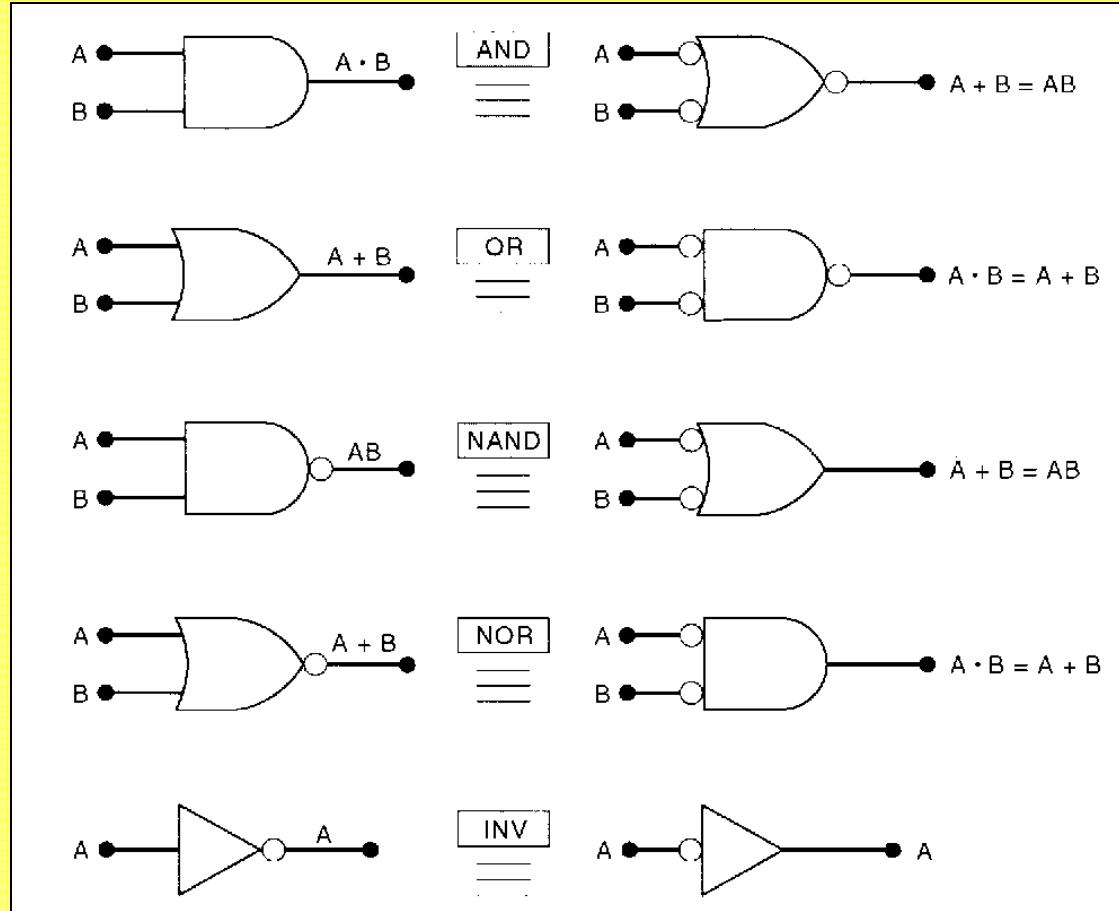


Figura 1: Símbolos padronizados e alternativos para várias portas lógicas e para o inversor

#### IV. Teoremas (recordação):

1	$X + Y = Y + X$	Comutatividade
2	$X \cdot Y = Y \cdot X$	Comutatividade
3	$X + (Y + Z) = (X + Y) + Z$	Associatividade
4	$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$	Associatividade
5	$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$	Distributividade
6	$(W + X) \cdot (Y + Z) = WY + XY + WZ + XZ$	Distributividade
7	$X + XY = X$	-
8	$X + 1 = 1$	-
9	$X \cdot 0 = 0$	-
10	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	DeMorgan
11	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	DeMorgan

#### V. Forma de soma-de-produtos:

- Os métodos de simplificação e projeto de circuitos lógicos que estudaremos exigem que a expressão esteja na forma de soma-de-produtos.
  - $ABC + \overline{ABC}$
  - $AB + \overline{ABC} + \overline{CD} + D$

<b>A</b>	<b>B</b>	<b>C</b>	<b>x</b>
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 $\rightarrow \overline{A}BC$
1	0	0	0
1	0	1	1 $\rightarrow A\overline{B}C$
1	1	0	1 $\rightarrow AB\overline{C}$
1	1	1	1 $\rightarrow ABC$

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

## VI. Simplificação de circuitos lógicos:

- Uma vez obtida a expressão de um circuito lógico, podemos ser capazes de reduzí-la a uma forma mais simples, que contenha um menor número de termos ou variáveis em um ou mais termos da expressão.
- Formas de simplificação:
  - Algébrica
  - Mapas de Karnaugh

## EXEMPLO 1

Simplifique o circuito lógico mostrado na figura 2, utilizando propriedades da álgebra de boole.

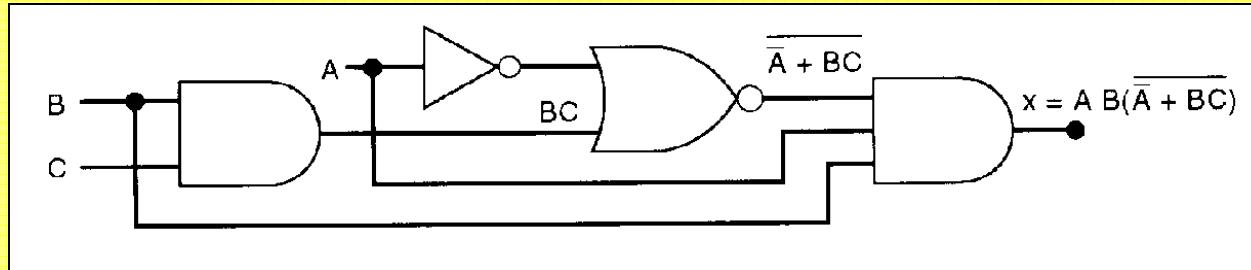


Figura 2: Circuito lógico

Solução:

**1º passo:** obter a expressão da função lógica

$$x = AB(\overline{A} + BC)$$

**2º passo:** aplicar os teoremas

$$x = AB(\overline{A} \cdot \overline{BC})$$

$$x = ABA(\overline{BC})$$

$$x = AB(\overline{B} + \overline{C})$$

$$x = ABB\bar{B} + ABC$$

$$x = ABC$$

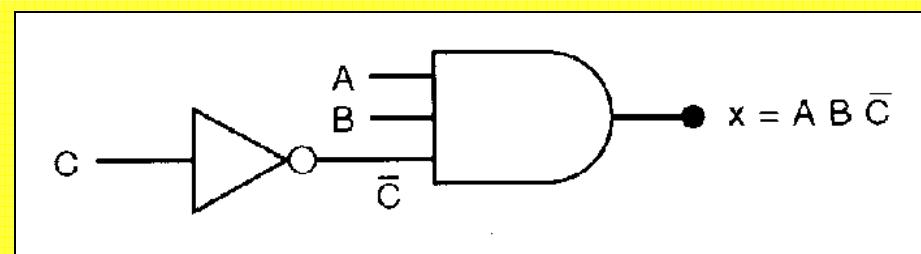
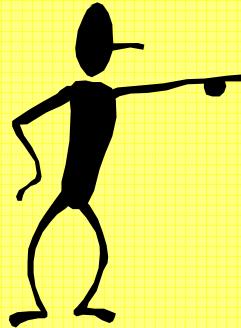


Figura 3: Circuito simplificado



- Infelizmente, nem sempre é óbvio qual teorema deve ser aplicado de modo a produzir o resultado mais simples.
- Além disso, não existe um modo fácil de constatar se a expressão obtida está em sua forma mais simples.
- Portanto, a simplificação algébrica freqüentemente se torna um processo de tentativa e erro.

### **Mapa de Karnaugh**

- Método gráfico usado para simplificar uma equação lógica ou converter uma tabela-verdade no circuito lógico correspondente, de modo simples e ordenado.
- Embora um mapa de Karnaugh possa ser usado em problemas que envolvam qualquer número de variáveis de entrada, sua utilidade prática está limitada a seis variáveis.

#### **Técnica:**

- a) Representar a função no mapa de Karnaugh.
- b) Escrever a função com o menor número de termos possíveis, englobando todos os “1” .
- c) Cada termo deve incluir o maior número possível de “1”, desde que sejam adjacentes entre si e que sejam grupos de  $2^n$  (1, 2, 4, 8, ...) de “1”.
- d) Para se obter um termo devemos escrever todas as variáveis comuns entre estes “1”, complementadas ou não, dependendo se a variável comum vale “0” ou “1”, respectivamente.

Exemplos:

A	B	X
0	0	1 → $\bar{A}\bar{B}$
0	1	0
1	0	0
1	1	1 → $A\bar{B}$

$$\left\{ X = \bar{A}\bar{B} + A\bar{B} \right\}$$

	B	B
$\bar{A}$	1	0
A	0	1

(a)

A	B	C	X
0	0	0	1 → $\bar{A}\bar{B}\bar{C}$
0	0	1	1 → $\bar{A}\bar{B}C$
0	1	0	1 → $\bar{A}BC$
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1 → $ABC$
1	1	1	0

$$\left\{ X = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + ABC \right\}$$

	$\bar{C}$	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$	1	0
AB	1	0
A $\bar{B}$	0	0

(b)

A	B	C	D	X
0	0	0	0	0
0	0	0	1	1 → $\bar{A}\bar{B}\bar{C}\bar{D}$
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1 → $\bar{A}\bar{B}\bar{C}\bar{D}$
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1 → $A\bar{B}\bar{C}\bar{D}$
1	1	1	0	0
1	1	1	1	1 → $A\bar{B}CD$

$$\left\{ \begin{array}{l} X = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD \\ + A\bar{B}\bar{C}\bar{D} + ABCD \end{array} \right.$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
A $\bar{B}$	0	0	0	0

(c)

Figura 4: Mapas de Karnaugh e tabelas-verdades para (a) duas, (b) três e (c) quatro variáveis.

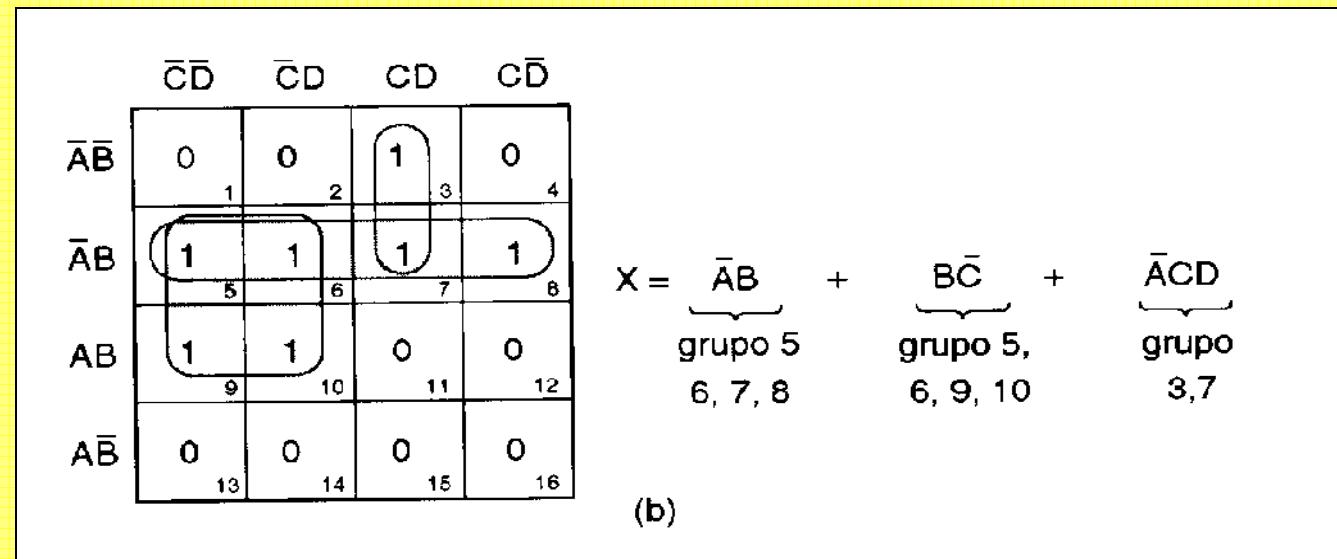
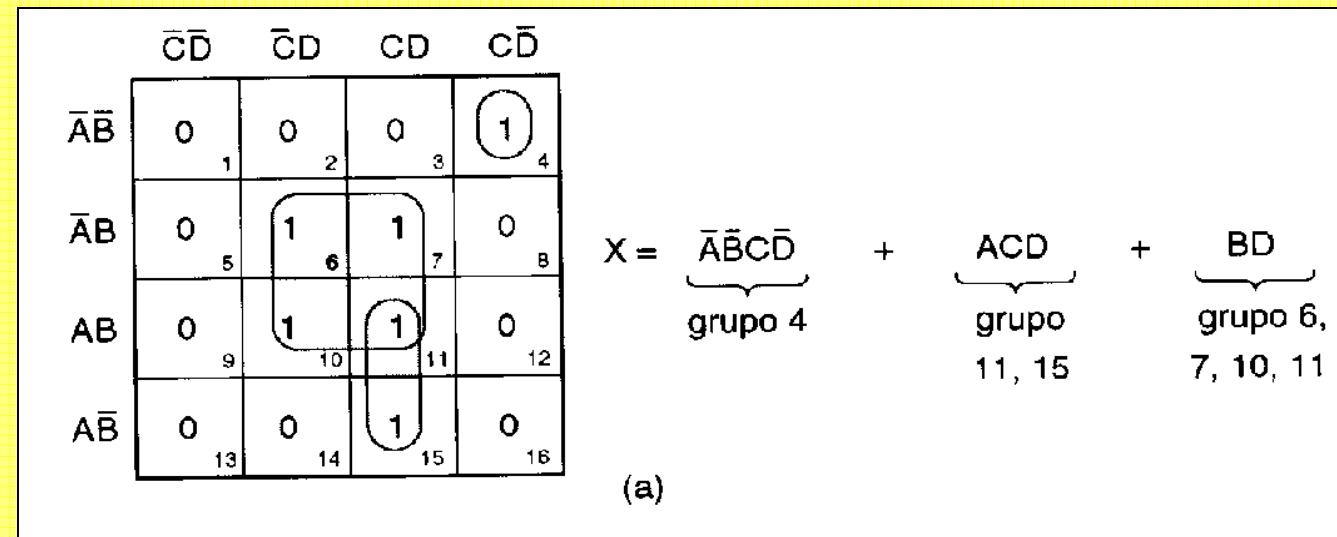
Exemplos:

Figura 5: Exemplos de aplicação da técnica de mapas de Karnaugh.

## EXEMPLO 2

Condição *don't care* (irrelevante)

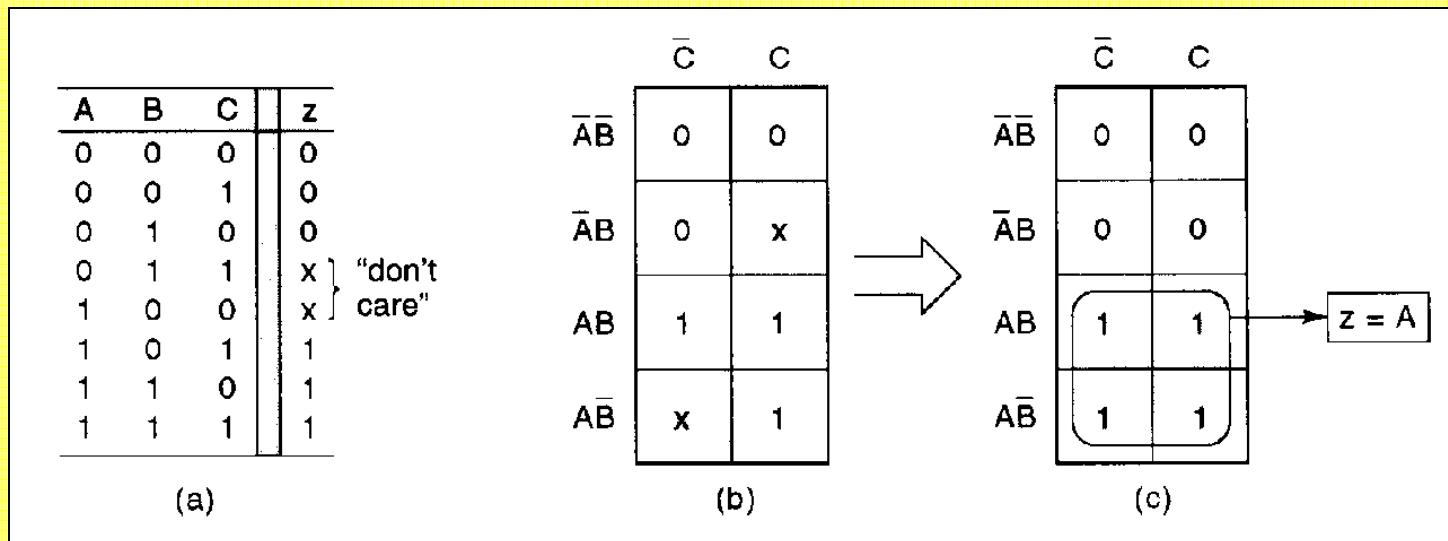
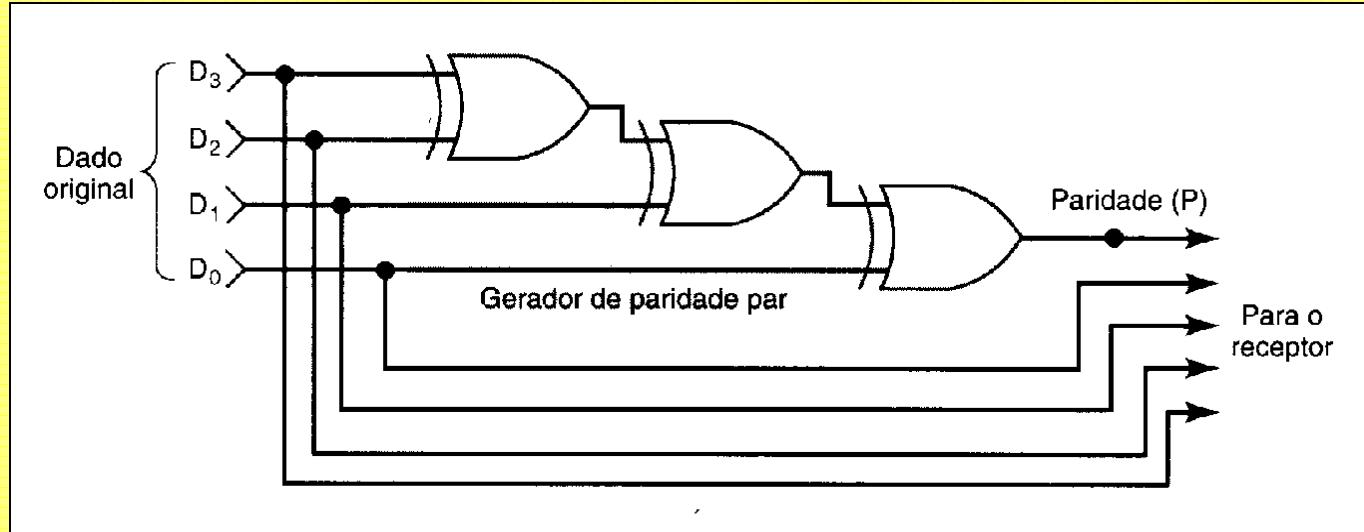


Figura 6: Condições *don't care* podem ser substituídas por “0” ou “1” para produzir o grupo que resulta na expressão mais simples

## VII. Aplicação de circuitos lógicos: Geração de Paridade e Verificação de Paridade

- Na transmissão de dados, o transmissor pode anexar um bit de paridade a um conjunto de bits antes de transmiti-lo para o receptor.
- Na recepção dos dados, o receptor detecta qualquer erro simples em apenas um bit que possa ter ocorrido durante a transmissão.

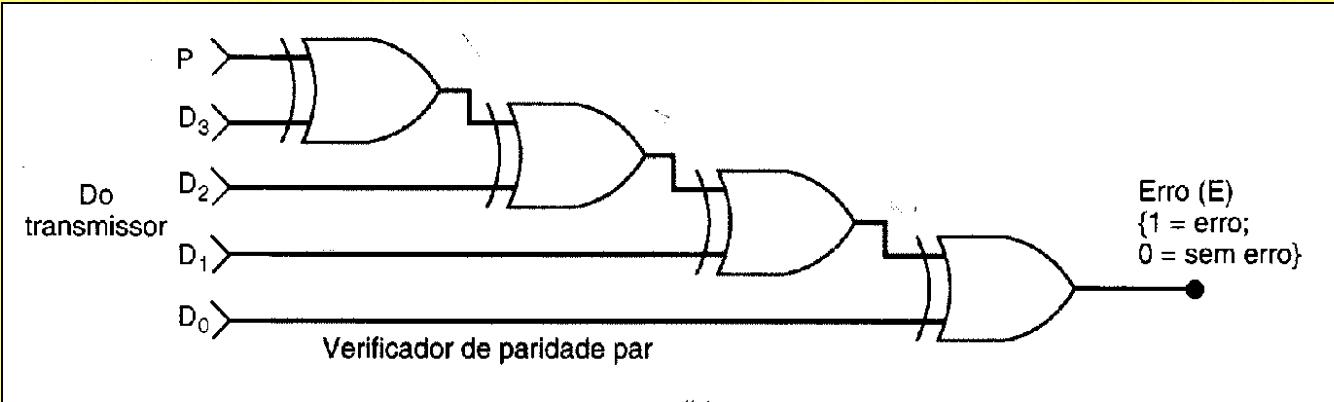
### Gerador de Paridade



	$D_3$	$D_2$	$D_1$	$D_0$	$P$
a)	0	1	1	1	1
b)	1	0	0	1	0
c)	0	0	0	0	0
d)	0	1	0	0	1

Figura 7: Gerador de Paridade.

## Verificação de Paridade



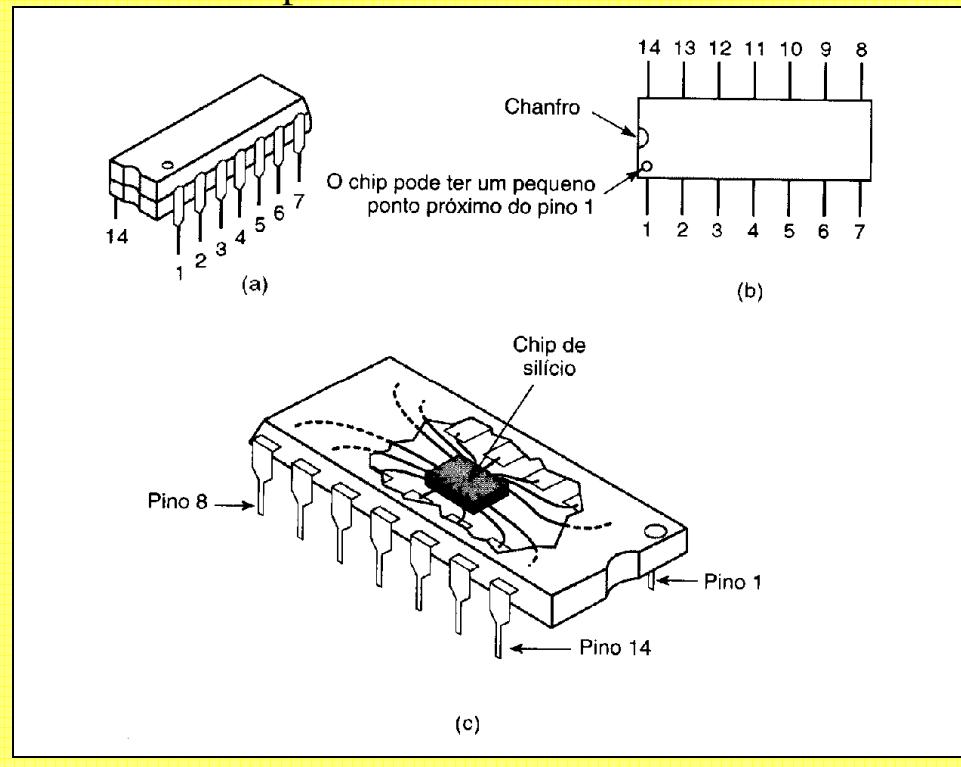
	<b>P</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
a)	0	1	0	1	0
b)	1	1	1	1	0
c)	1	1	1	1	1
d)	1	0	0	0	0

- Os casos (c) e (d) apresentam erro de paridade.

Figura 8: Verificador de Paridade.

### VIII. Características básicas de CIs digitais

- CIs digitais são uma coleção de resistores, diodos e transistores fabricados em uma única peça de material semicondutor (geralmente de silício), e comumente chamado de *chip*.
- Este *chip* é então encapsulado em uma embalagem protetora de plástico ou de cerâmica, a partir da qual saem pinos para tornar possível a ligação do CI com outros dispositivos.
- Um dos encapsulamentos mais comuns é o *dual-in-line package (DIP)*, mostrado na Figura 9.



- Existem DIPs de 14, 16, 20, 24, 28, 40 e 64 pinos.

Figura 9: (a) Encapsulamento *dual-in-line* (DIP); (b) vista superior; (c) o chip de silício é muito menor que o encapsulamento

- O DIP é provavelmente o encapsulamento para CI's mais comum de ser encontrado em equipamentos digitais, embora outros tipos estejam se tornando cada vez mais populares.

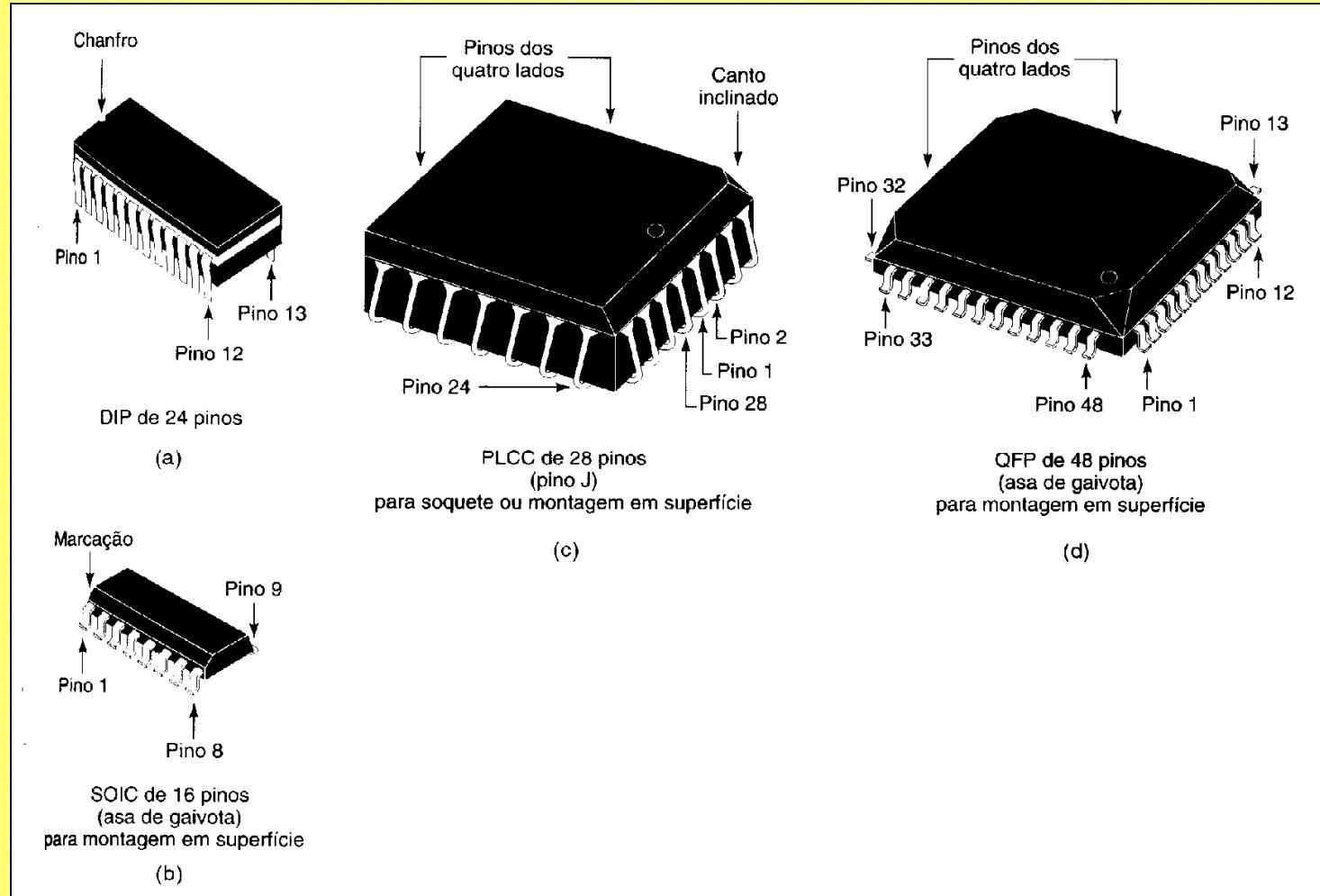


Figura 10: Encapsulamentos comuns de CI's.

- Os CIs digitais são muitas vezes classificados de acordo com a complexidade de seus circuitos, que é medida pelo número de portas lógicas equivalentes no seu substrato.

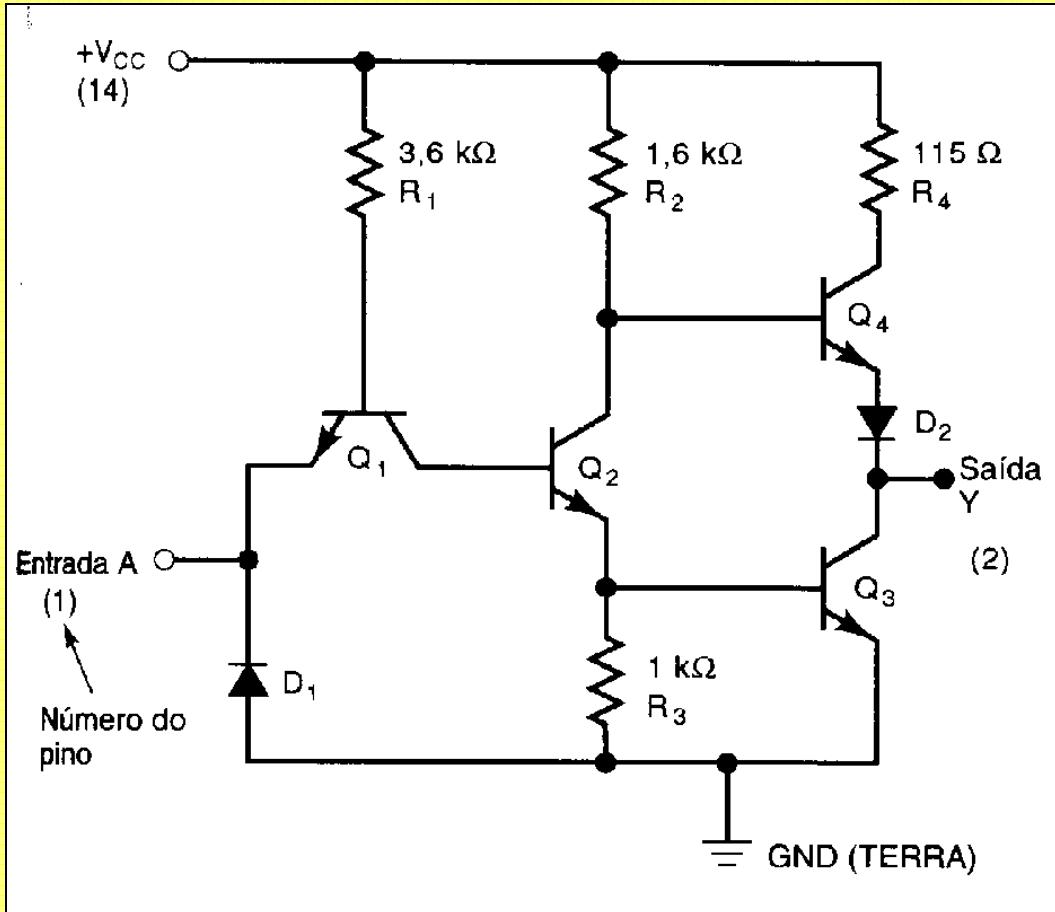
Complexidade	Número de Portas
Small-scale integration (SSI)	Menor do que 12
Medium-scale integration (MSI)	12 a 99
Large-scale integration (LSI)	100 a 9.999
Very large-scale integration (VLSI)	10.000 a 99.999
Ultra large-scale integration (ULSI)	100.000 a 999.999
Giga-scale integration (GSI)	1.000.000 ou mais

Figura 11: Classificação dos CIs quanto a complexidade de seus circuitos.

## IX. CIs digitais Bipolares e Unipolares

- Classificados conforme o tipo de componente eletrônico usado.
- **Bipolares:** utilizam o transistor bipolar (NPN e PNP)
- **Unijunção:** transistores de efeito-de-campo (NMOS e PMOS)

## Família TTL (*Transistor-Transistor-Logic*)

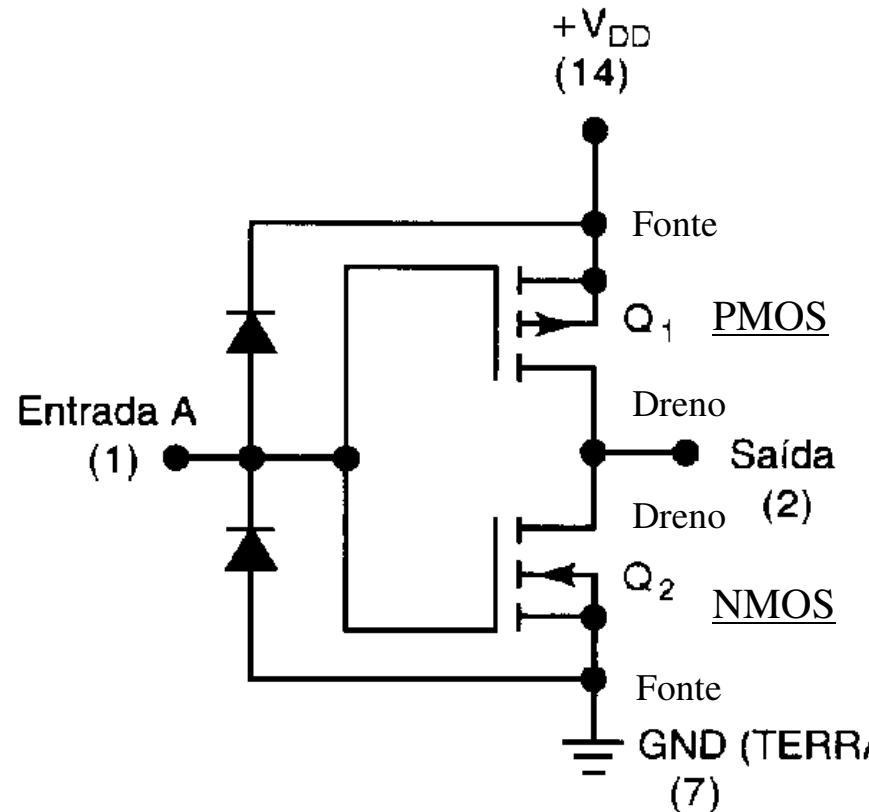


A	Q1	Q2	Q3	Q4	Y
5 V	C	S	S	C	GND
0 V	S	C	C	S	Vcc

C = Cortado

S = Saturado

Figura 12: Circuito do INVERSOR TTL.

Família CMOS (*Complementary Metal-Oxide-Semiconductor*)

A	Q1	Q2	Sáida
0 V	Conduz	Corta	V <sub>DD</sub>
5 V	Corta	Conduz	GND

Figura 13: Circuito do INVERSOR CMOS.

Séries dentro da família TTL

<b>Subfamílias TTL</b>	<b>Prefixo</b>	<b>Exemplo de CI</b>
TTL padrão	74	7404 INVERSOR sêxtuplo
TTL Schottky	74S	74S04 INVERSOR sêxtuplo
TTL Schottky de baixa potência	74LS	74LS04 INVERSOR sêxtuplo
TTL Schottky avançada	74AS	74AS04 INVERSOR sêxtuplo
TTL Schottky avançada de baixa potência	74ALS	74ALS04 INVERSOR sêxtuplo

## Séries dentro da família CMOS

<b>Subfamílias CMOS</b>	<b>Prefixo</b>	<b>Exemplo de CI</b>
CMOS de porta metálica	40	4001 portas NOR quádruplas
CMOS de porta metálica; pinagem compatível com TTL	74C	74C02 portas NOR quádruplas
CMOS de porta de silício; alta velocidade; pinagem compatível com TTL	74HC	74HC02 portas NOR quádruplas
CMOS de porta de silício; alta velocidade; pinagem compatível com TTL; eletricamente compatível com TTL	74HCT	74HCT02 portas NOR quádruplas
CMOS avançada; pinagem incompatível com TTL; eletricamente incompatível com TTL	74AC	74AC02 portas NOR quádruplas
CMOS avançada; pinagem incompatível com TTL; eletricamente compatível com TTL	74ACT	74ACT02 portas NOR quádruplas

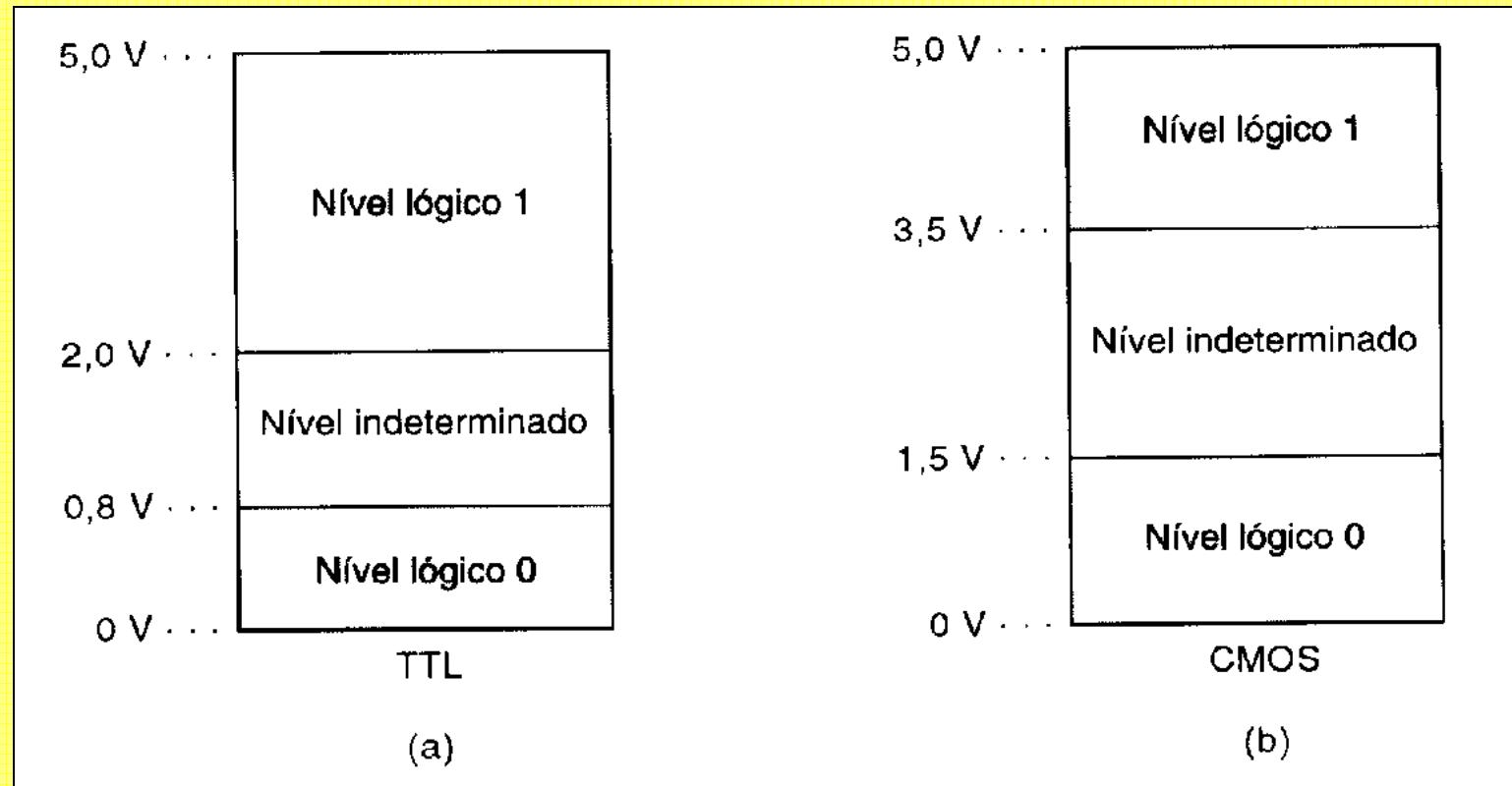
Faixas de tensão para os níveis lógicos

Figura 14: Níveis lógicos de entrada para CI's digitais TTL e CMOS.