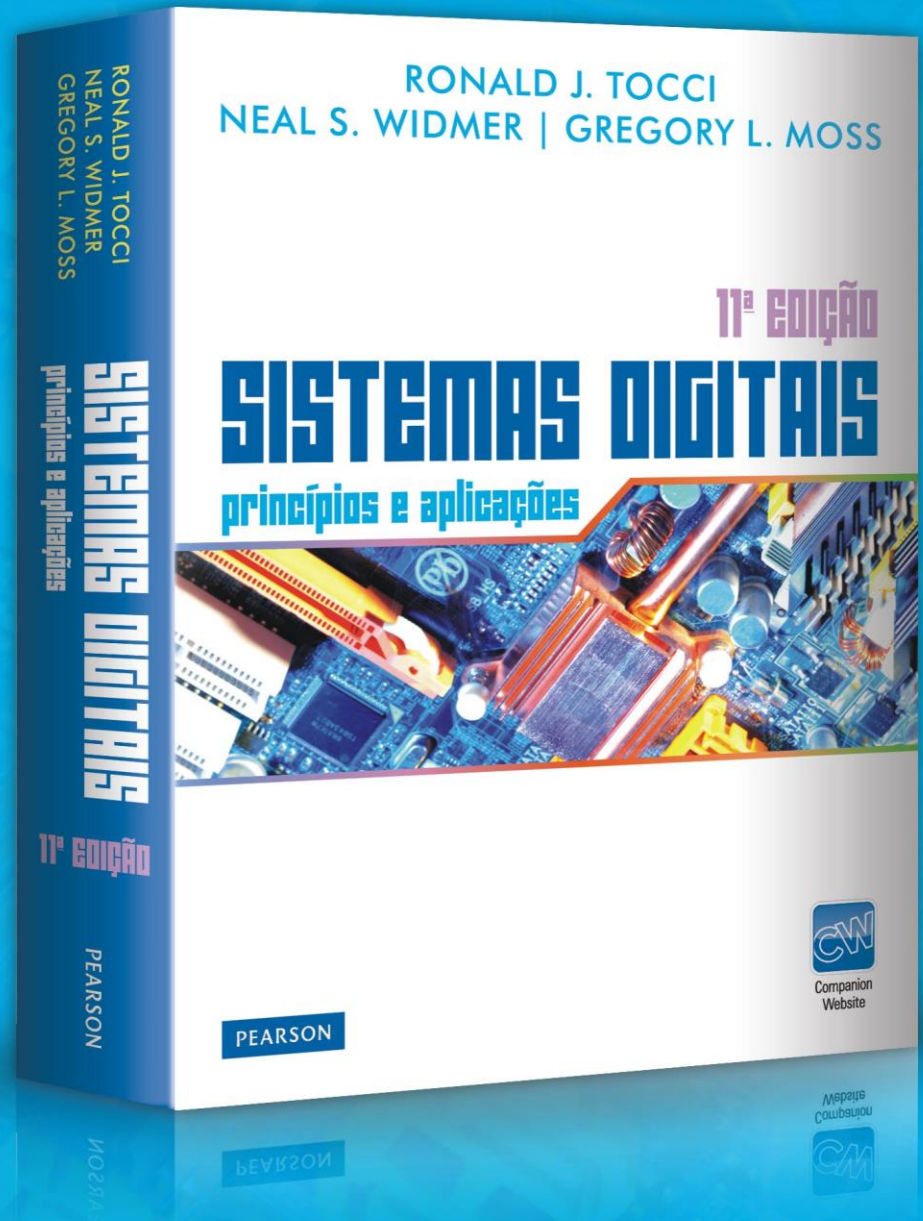


# Capítulo 1

# Conceitos Introdutórios

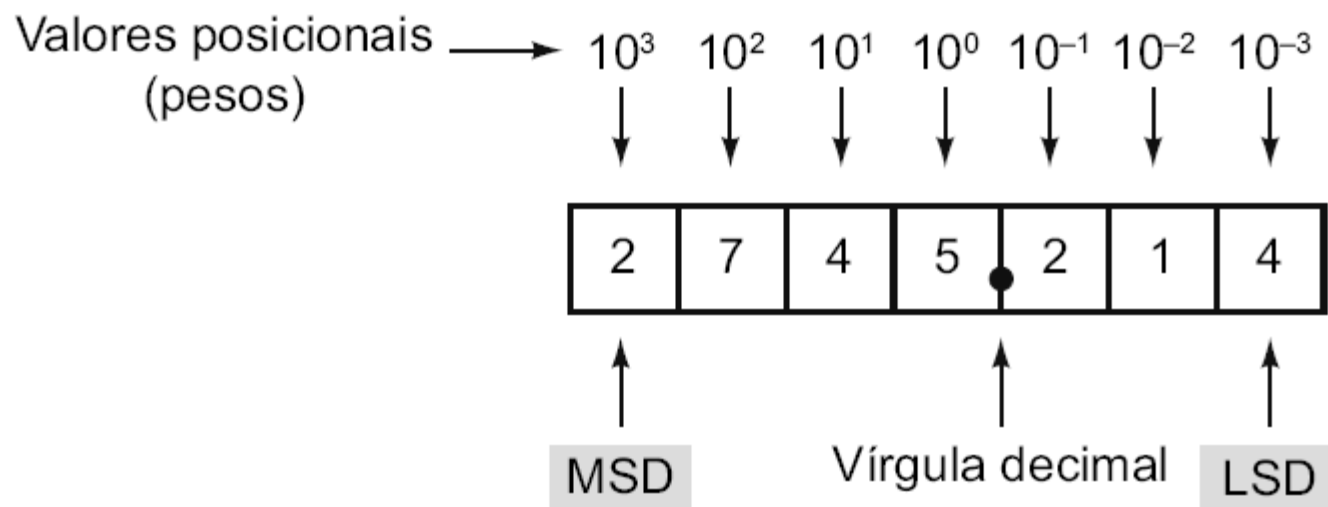


## 1.4 Sistemas de Números Digitais

- Compreender os sistemas digitais requer um entendimento dos sistemas decimal, binário, octal e hexadecimal.
  - Decimal – dez símbolos (base 10)
  - Binário – dois símbolos (base 2)
  - Octal – oito símbolos (base 8)
  - Hexadecimal – dezesseis símbolos (base 16)

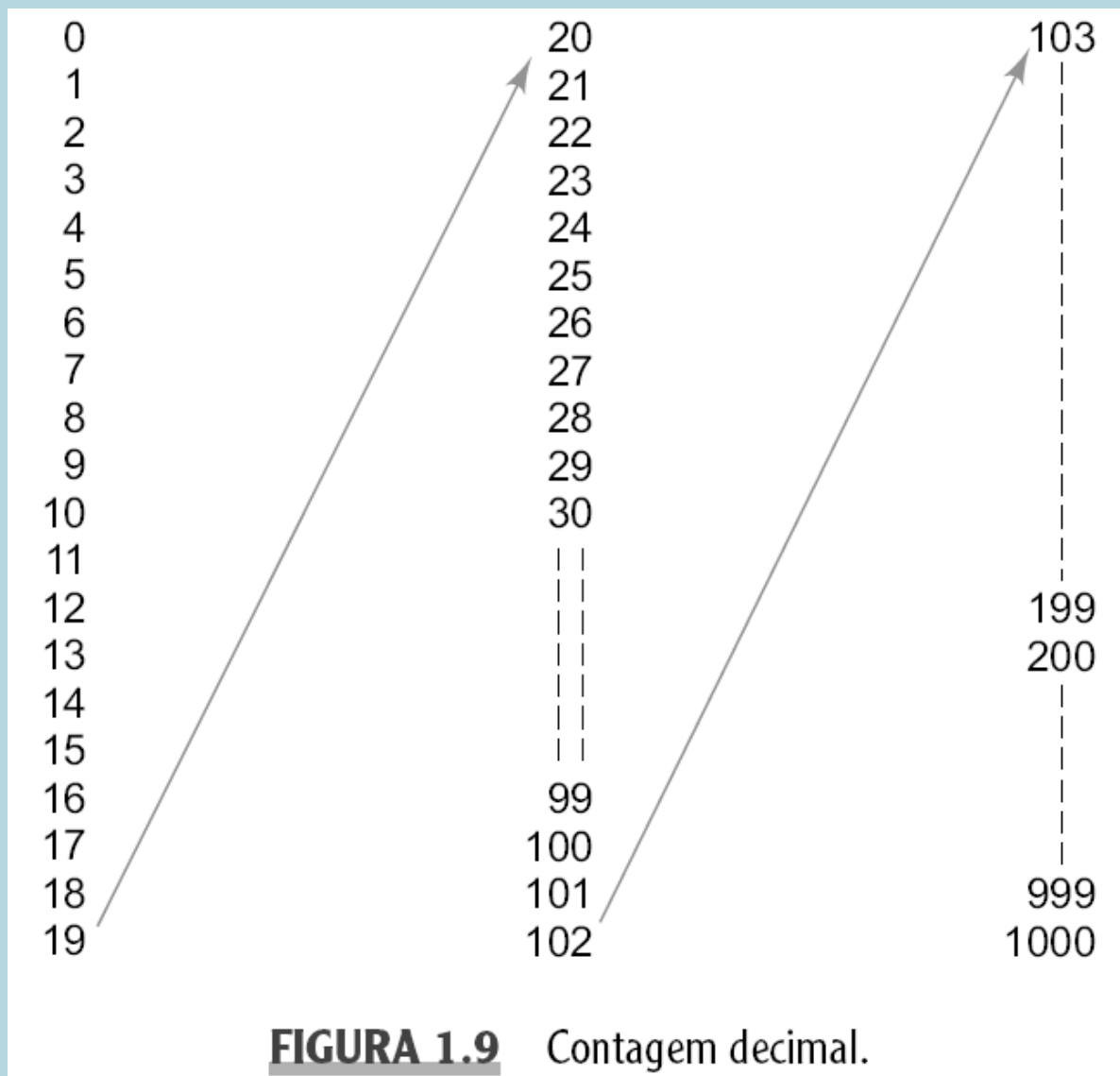
## 1.4 Sistemas de Números Digitais

- O Sistema Decimal:
  - Dez símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
    - Cada número é um *dígito* (do latim, dedo).
- Dígitos mais significantes (MSD) e dígitos menos significantes (LSD).
- Valor posicional pode ser declarado como um dígito multiplicado por uma potência de 10.



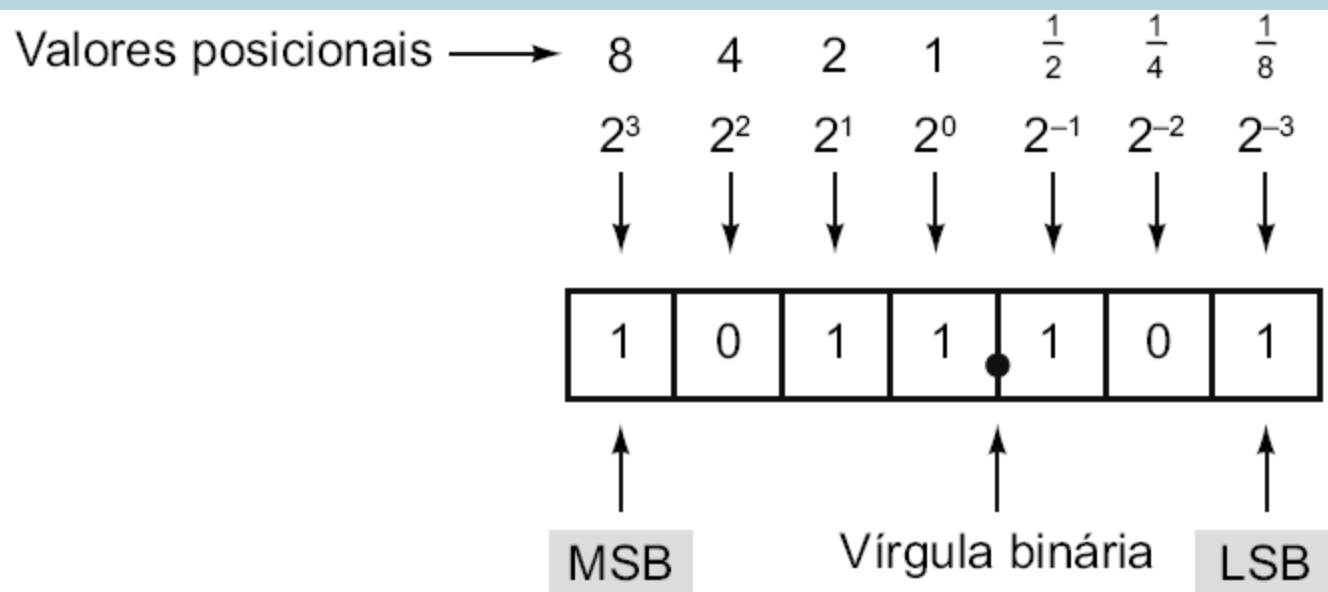
**FIGURA 1.8** Valores posicionais de um número decimal expresso como potências de 10.

# SISTEMAS DIGITAIS



## 1.4 Sistemas de Números Digitais

- O Sistema Binário (base 02) :
  - Dois símbolos: 0 e 1.
    - Empresta-se ao projeto de circuitos eletrônicos com apenas dois diferentes níveis de tensão obrigatórios.
- Valor posicional pode ser indicado como um dígito multiplicado por uma potência de 2.



**FIGURA 1.10** Valores posicionais de um número binário expresso como potências de 2.

Pesos →	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$		Número decimal equivalente
	0	0	0	0	→	0
	0	0	0	1	→	1
	0	0	1	0		2
	0	0	1	1		3
	0	1	0	0		4
	0	1	0	1		5
	0	1	1	0		6
	0	1	1	1		7
	1	0	0	0		8
	1	0	0	1		9
	1	0	1	0		10
	1	0	1	1		11
	1	1	0	0		12
	1	1	0	1		13
	1	1	1	0	→	14
	1	1	1	1	→	15

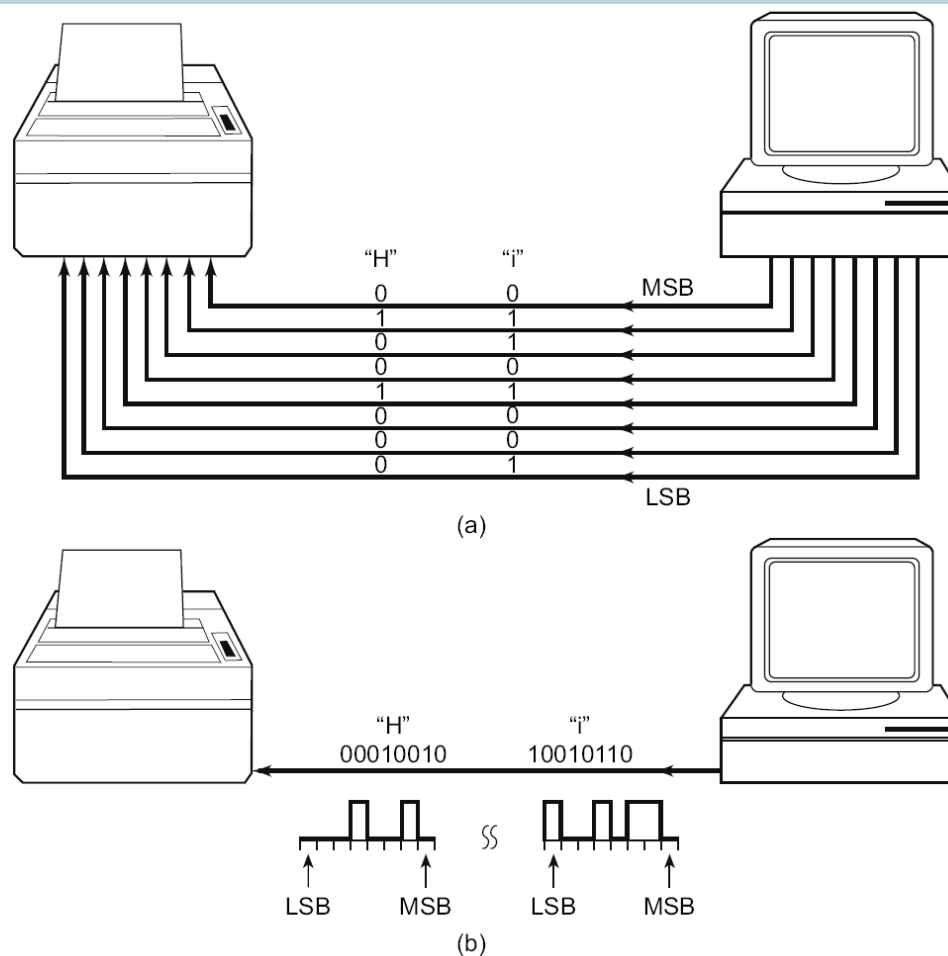
↑  
LSB

**FIGURA 1.11** Sequência de contagem binária.



## 1.7 Transmissão Paralela e Serial

- Na transmissão paralela, todos os bits em um número binário são transmitidos simultaneamente.
  - Uma linha separada é exigida para cada bit.
- Na transmissão serial, cada bit em um número binário é transmitido em algum intervalo de tempo.

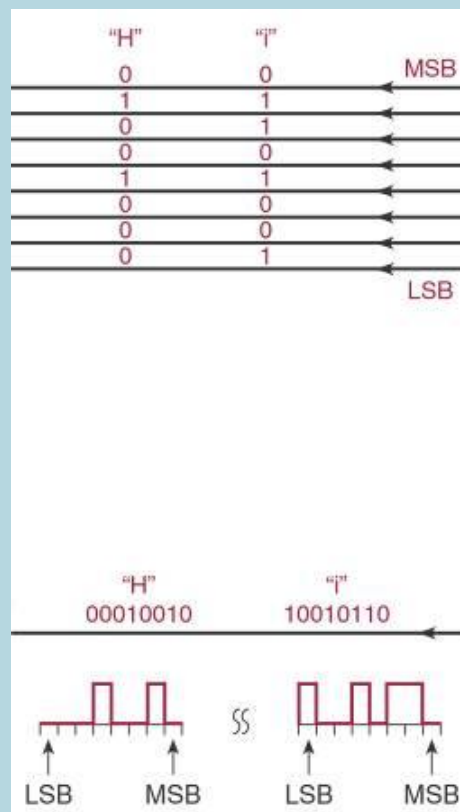


**FIGURA 1.15** (a) A transmissão paralela usa uma linha de conexão por bit, e todos os bits são transmitidos simultaneamente; (b) a transmissão serial usa apenas uma linha de sinal, na qual os bits são transmitidos serialmente (um de cada vez).

## 1.7 Transmissão Paralela e Serial

A transmissão paralela é mais rápida, porém requer mais caminhos.

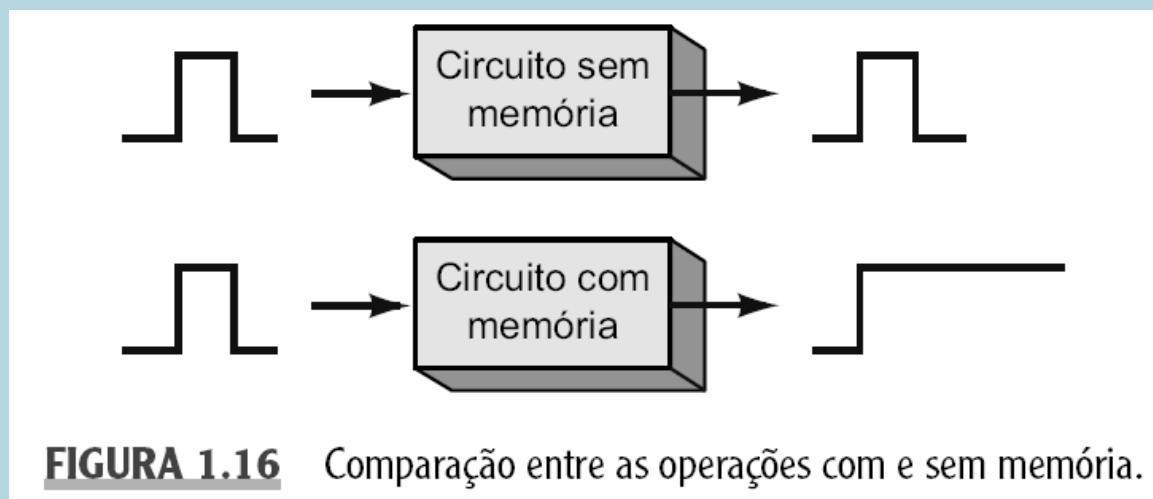
A transmissão serial é mais lenta, mas requer um único caminho.



## 1.8 Memória

A memória é exibida através de um circuito que mantém uma resposta a uma entrada momentânea.

Ela é importante porque proporciona uma maneira de armazenar números binários, temporária ou permanentemente.



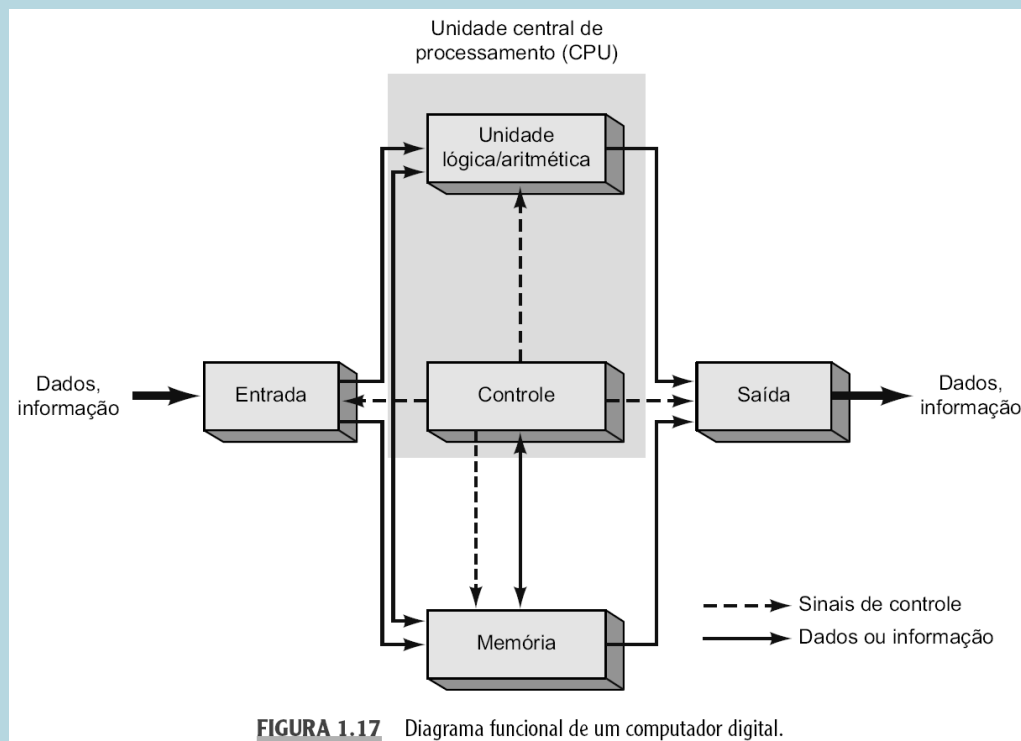
**FIGURA 1.16** Comparação entre as operações com e sem memória.

Elementos de memória: magnéticos, ópticos e circuitos eletrônicos de travamento.

## 1.9 Computadores Digitais

- O computador é um sistema de *hardware* que executa operações aritméticas, manipula dados e toma decisões.
- Realiza operações com base nas instruções sob a forma de um *programa* em alta velocidade e com alto grau de precisão.

## 1.9 Computadores Digitais



Muitas vezes, as unidades de controle e de aritmética/ lógica são tratadas como uma coisa só. Chama-se, então, unidade central de processamento (CPU).

## 2.3 Sistema Numérico Hexadecimal

Relações entre os números hexadecimais, decimais e binários.

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

## 2.4 Código BCD

- Converta o número  $874_{10}$  para BCD.  
Cada dígito decimal é representado por 4 bits.
- Cada grupo de 4 bits não pode ser superior a 9.

8	7	4	(decimal)
↓	↓	↓	
1000	0111	0100	(BCD)

- Inverta o processo para converter o BCD para decimal.

9	4	3	(decimal)
↓	↓	↓	
1001	0100	0011	(BCD)

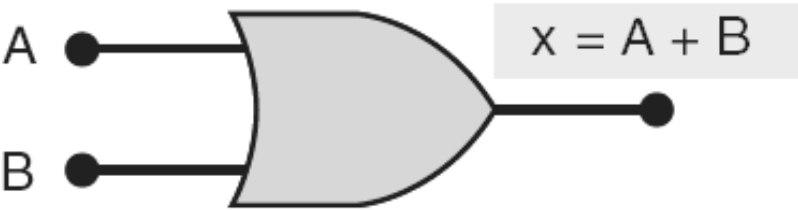


## 3.3 Operação OR com porta OR

Uma porta OR é um circuito com uma ou mais entradas, cuja saída é igual à combinação OR das entradas.

**Tabela-verdade símbolo de circuito para duas entradas da porta OR.**

A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1



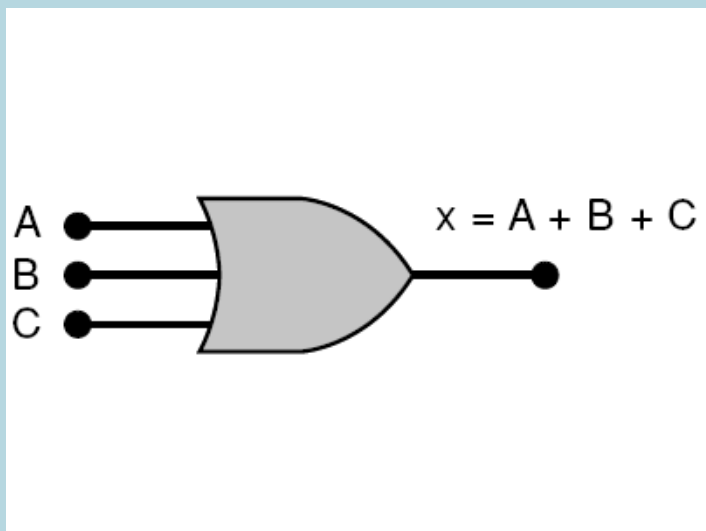
Porta OR

$x = A + B$

## 3.3 Operação OR com porta OR

A porta **OR** é um circuito com duas ou mais entradas, cuja saída é igual a combinação OR das entradas.

**Tabela-verdade símbolo de circuito para três entradas da porta OR.**



A	B	C	$x = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

## 3.4 Operação AND (“E”) com portas AND

- A operação AND é similar a multiplicação convencional.

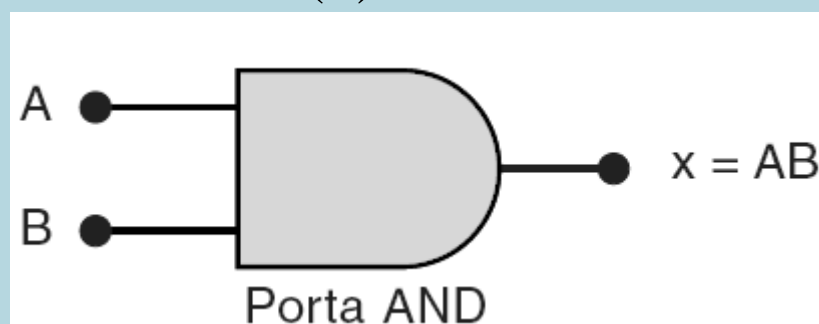
$$X = A \cdot B \cdot C \text{ — Leia “}X \text{ é igual a } A \text{ e } B \text{ e } C\text{”}.$$

O sinal  $\bullet +$  não se aplica para soma, mas sim para operações AND.

*X é verdadeiro (1) quando A e B e C são verdadeiros (1).*

AND		
A	B	$x = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

**Tabela-Verdade**

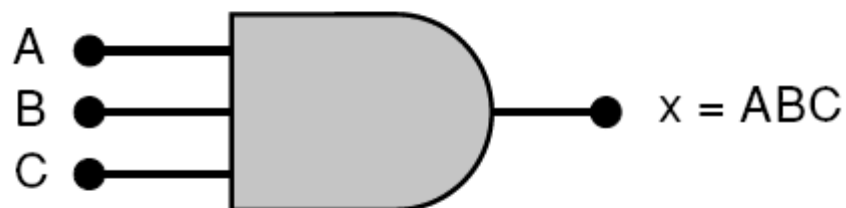


**Simbolo da Porta**

## 3.4 Operação AND com porta AND

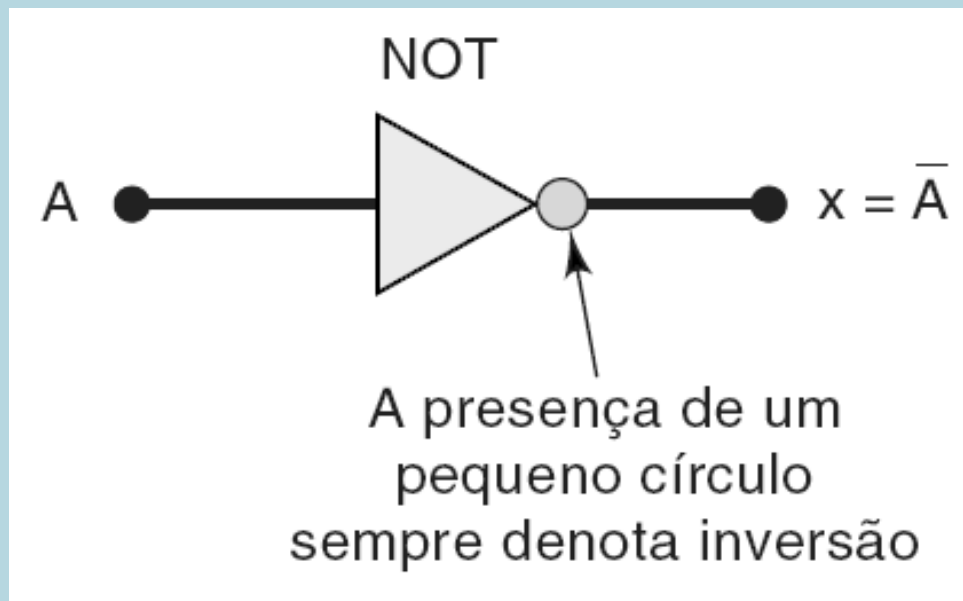
Tabela-verdade símbolo de circuito para três entradas e porta AND.

A	B	C	$x = ABC$
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	1	0	0
1	1	1	1



## 3.5 Operação NOT

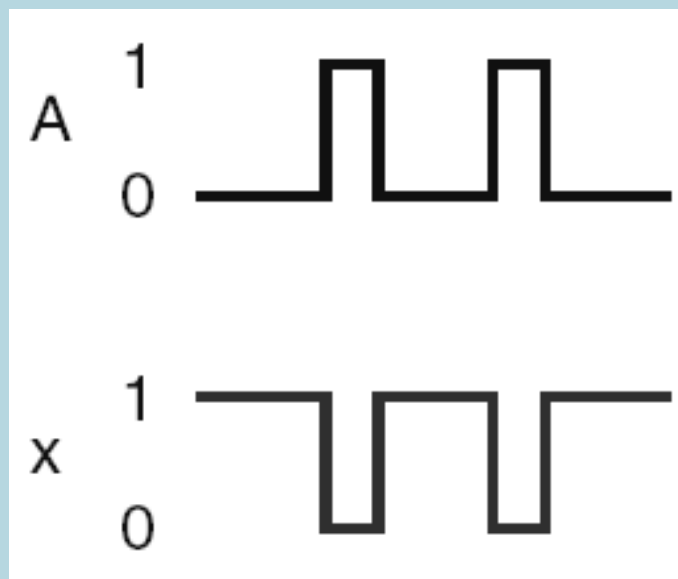
Um circuito NOT é comumente chamado de inversor.



Esses circuitos sempre têm uma única entrada, e a lógica da saída é sempre oposta ao nível da lógica da entrada.

## 3.5 Operação NOT

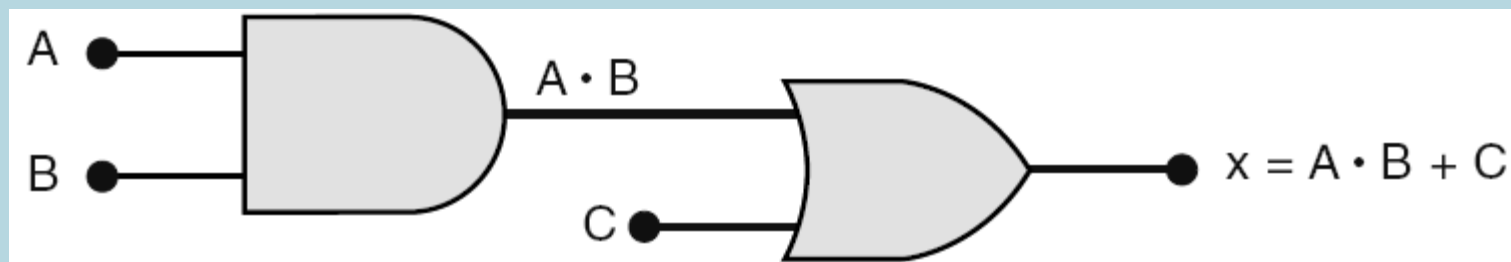
O INVERSOR inverte (complementa) o sinal da entrada, em todos os pontos, na forma de onda.



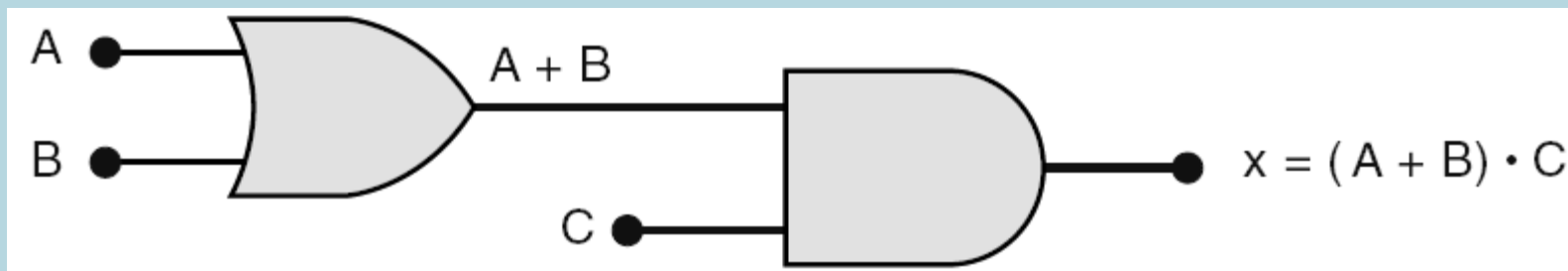
Sempre que a entrada = 0 a saída = 1 e vice-versa.

## 3.6 Descrevendo Circuitos Lógicos Algebricamente

- Se uma expressão contém ambas as portas – **AND** e **OR** – a operação **AND** irá acontecer anteriormente.



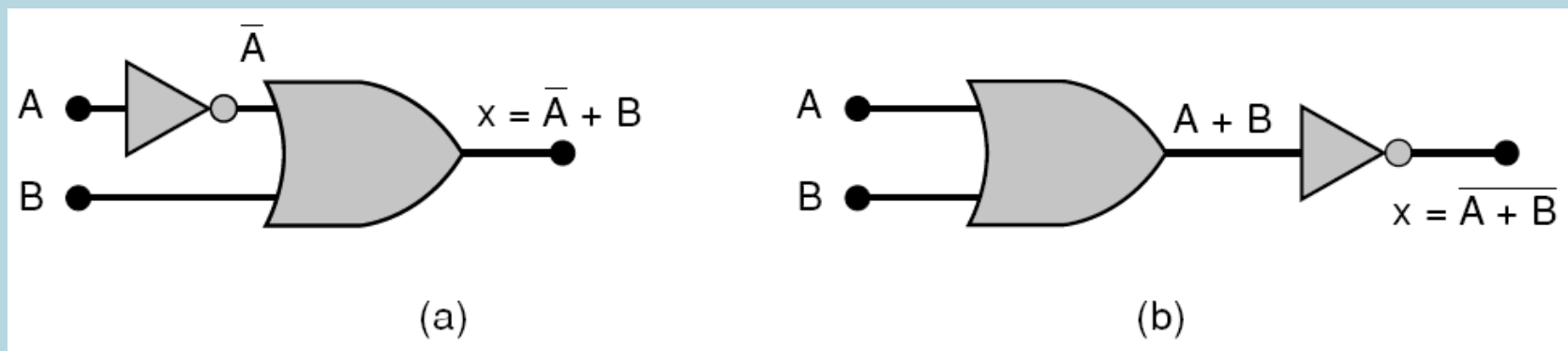
- A menos que existam parêntesis na expressão.



## 3.6 Descrevendo Circuitos Lógicos Algebricamente

- Sempre que um INVERSOR estiver presente, a saída é equivalente a entrada, com uma barra sobre ele.

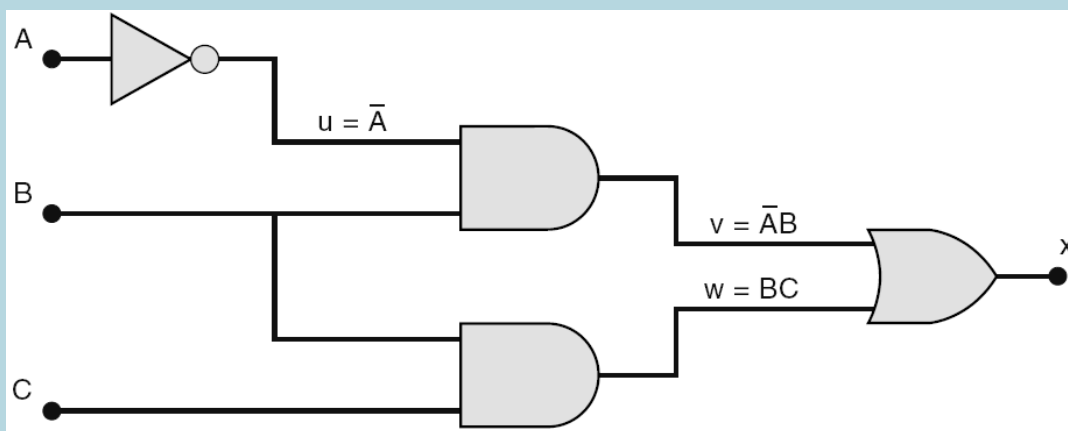
Entrada  $A$  através de um inversor é igual a  $\bar{A}$ .





## 3.7 Avaliando as Saídas dos Circuitos Lógicos

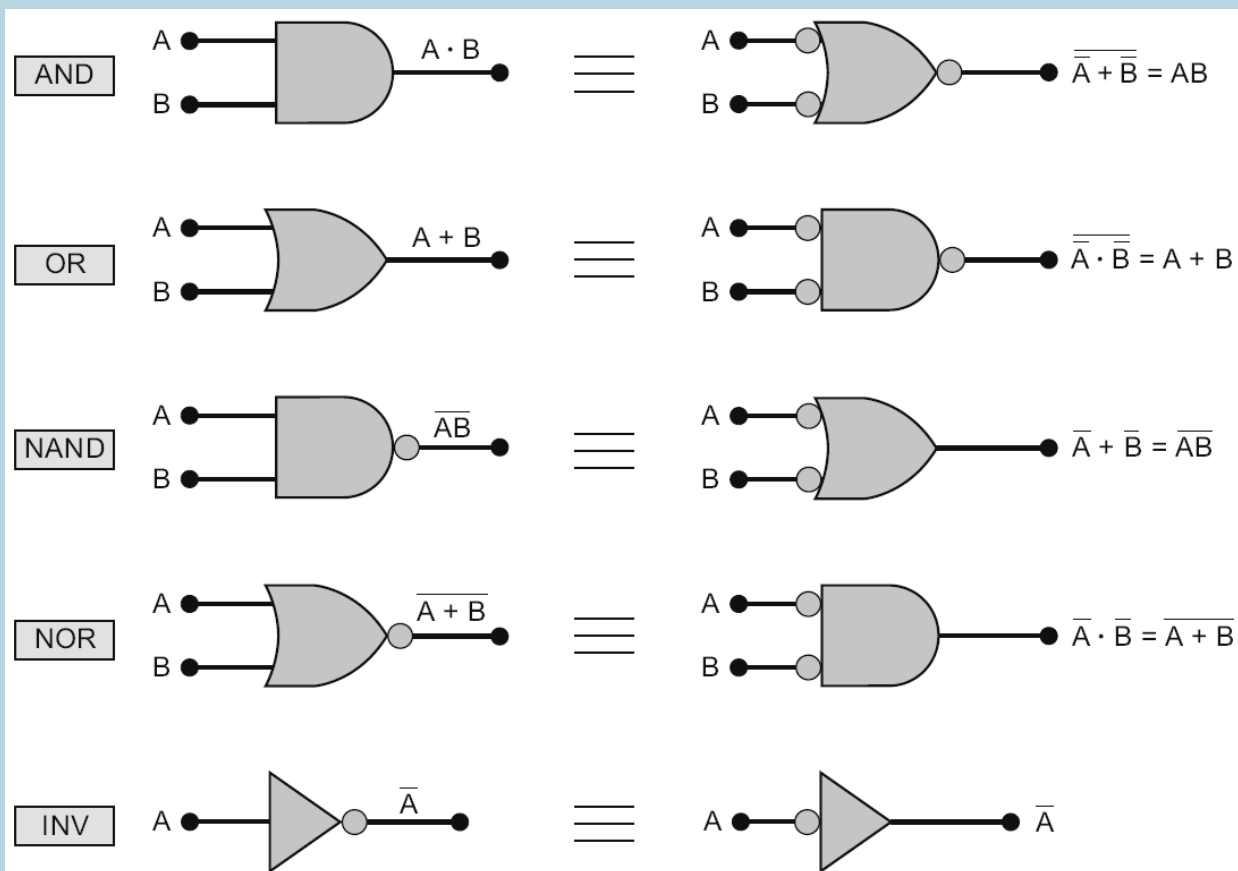
- Logicamente, a etapa final é a combinação das colunas V e W para prever a *saída* x.



A	B	C	$\underline{u} = \bar{A}$	$\underline{v} = \bar{A}B$	$w = BC$	$x = v + w$
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	1	1

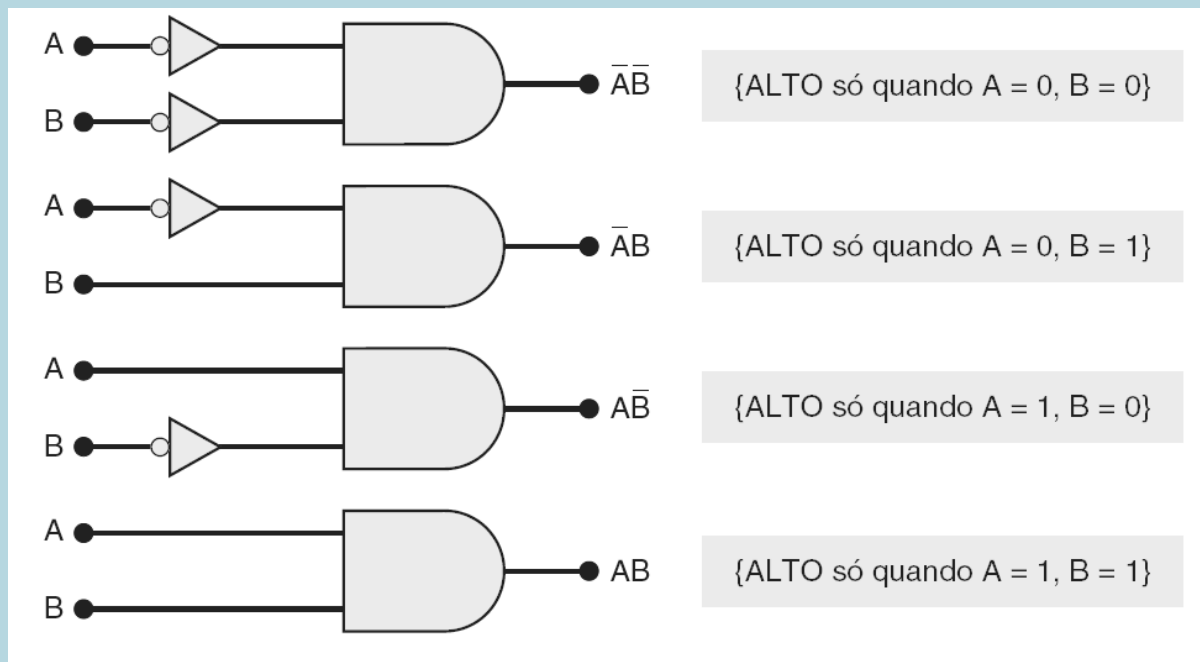
Desde que  $x = v + w$ , a saída  $x$  será ALTO quando  $v$   
**OU**  $w$  for ALTO.

## 3.13 Alternar Representações para Portas Lógicas



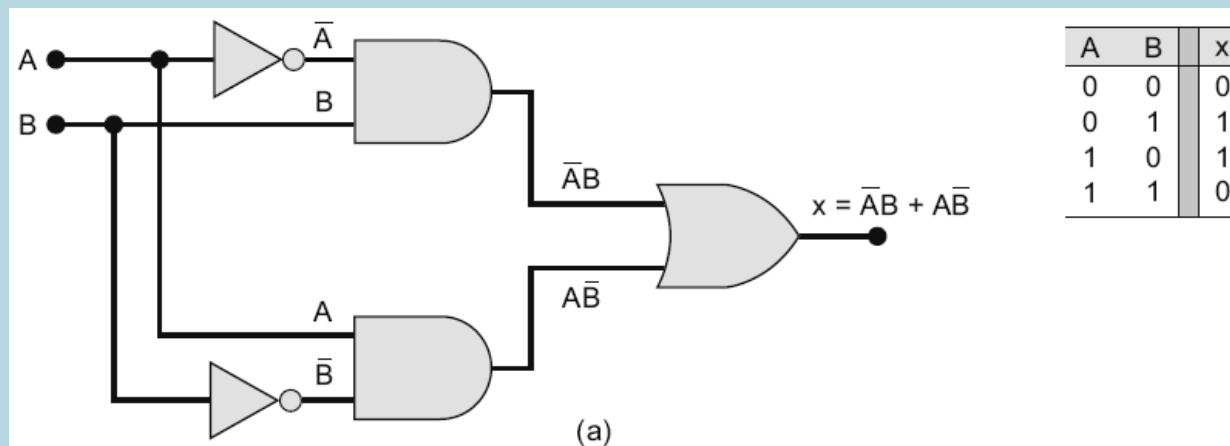
## 4.4 Projetando Circuitos Lógicos Combinacionais

Uma porta AND, com entradas apropriadas, pode ser usada para produzir uma saída em nível 1 para um conjunto específico de níveis de entrada.



## 4.6 Circuitos Exclusive-OR e Exclusive-NOR

Circuito exclusive-OR e tabela-verdade.

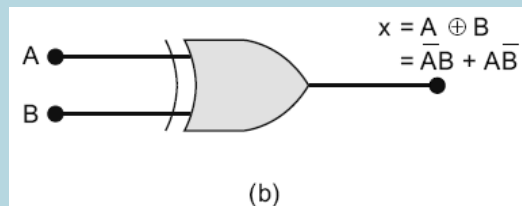


Expressão da saída :  $x = \bar{A}B + A\bar{B}$

Esse circuito produz uma saída ALTA sempre que as duas entradas estiverem em níveis diferentes.

## 4.6 Circuitos Exclusive-OR e Exclusive-NOR

Símbolo tradicional para a porta XOR:



Uma porta XOR tem apenas *duas* entradas combinadas, de forma que  $x = \overline{A}B + A\overline{B}$ .

A forma abreviada para indicar a expressão de saída XOR é:  $x = A \oplus B$ .

...em que o símbolo  $\oplus$  representa a operação da porta XOR.

A saída é ALTA somente quando as duas entradas estão em níveis diferentes.

CI's com chips quádruplos de portas XOR:

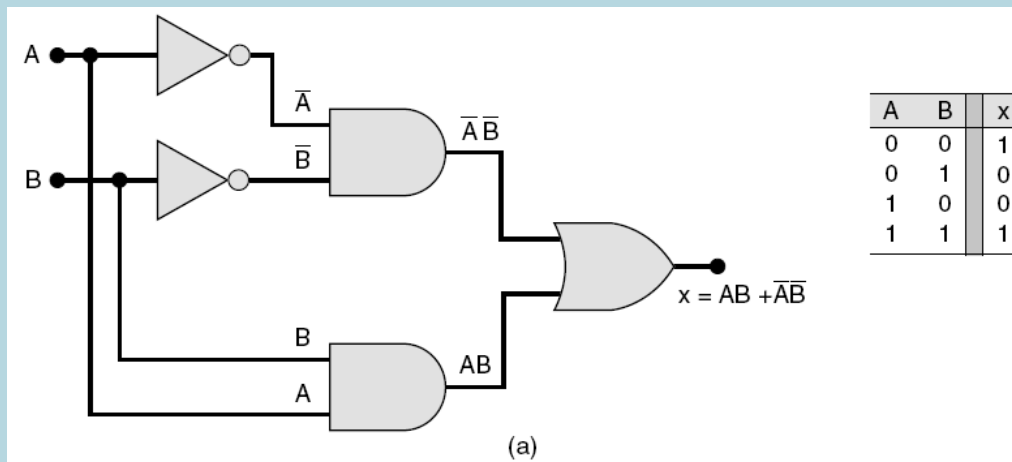
74LS86 CI quádruplo XOR (família TTL)

74C86 CI quádruplo XOR (família CMOS)

74HC86 CI quádruplo XOR (CMOS de alta velocidade)

## 4.6 Circuitos Exclusive-OR e Exclusive-NOR

Circuito exclusive-**NOR** e tabela-verdade.

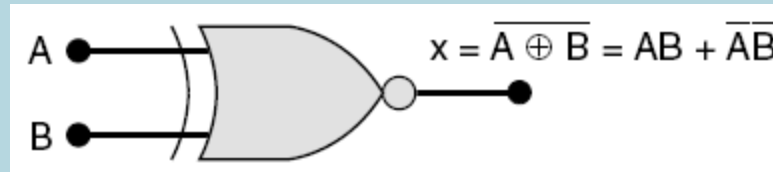


Expressão de saída :  $x = AB + \bar{A}\bar{B}$

XNOR produz uma saída ALTA sempre que duas entradas estão no mesmo nível.

## 4.6 Circuitos Exclusive-OR e Exclusive-NOR

Símbolo tradicional de porta XNOR



Uma porta XNOR tem apenas *duas* entradas combinadas. Assim,  $x = AB + \overline{A}\overline{B}$ .

A forma abreviada para indicar a expressão de saída XOR é:  $x = \overline{A \oplus B}$ .

**XNOR** representa o inverso da operação **XOR**.

A saída é ALTA apenas quando as entradas estão no mesmo nível.

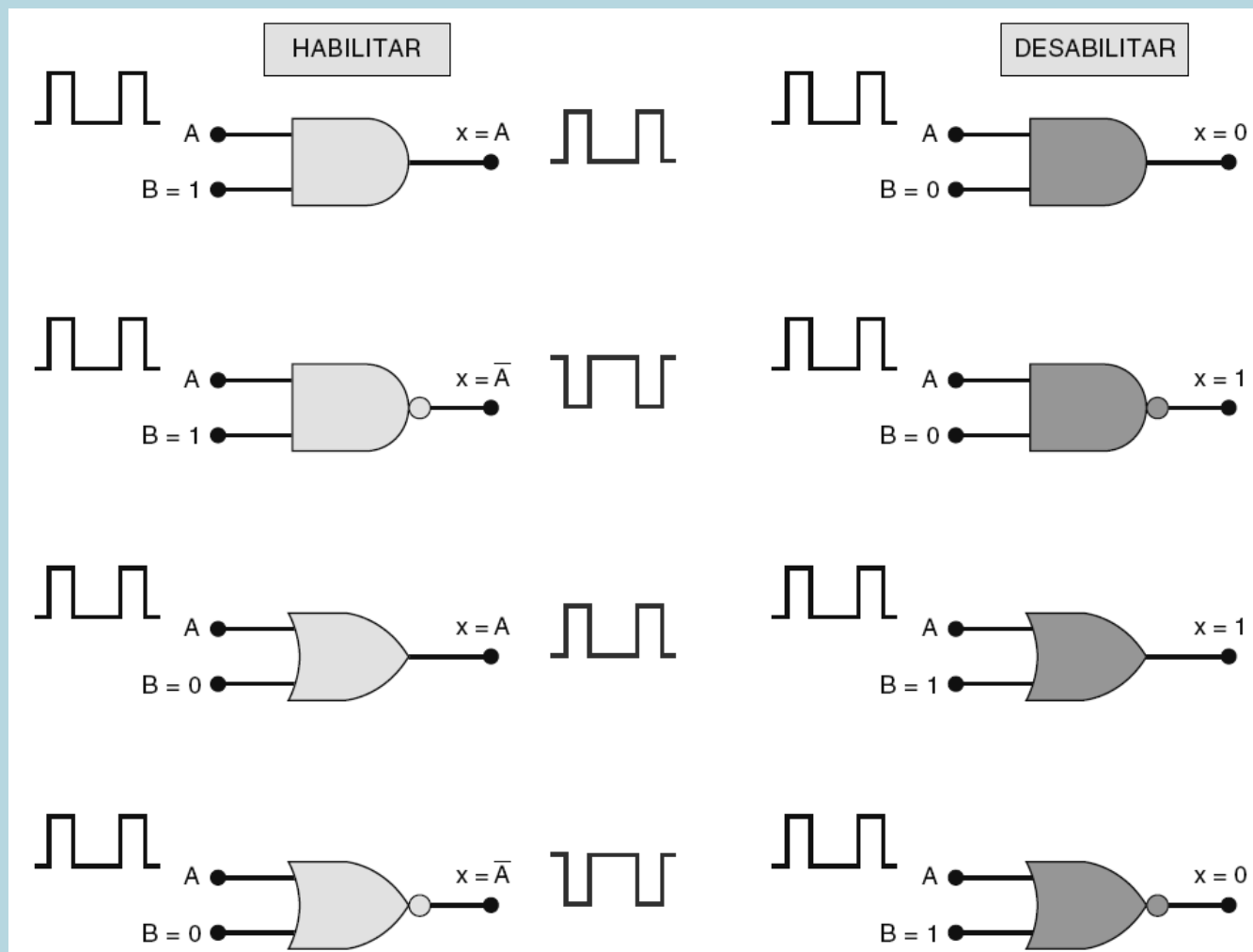
CI com chips quádruplos de portas XNOR

74LS266 CI quádruplo **XNOR** (família TTL)

74C266 CI quádruplo **XOR** (família CMOS)

74HC266 CI quádruplo **XOR** (CMOS de alta velocidade)

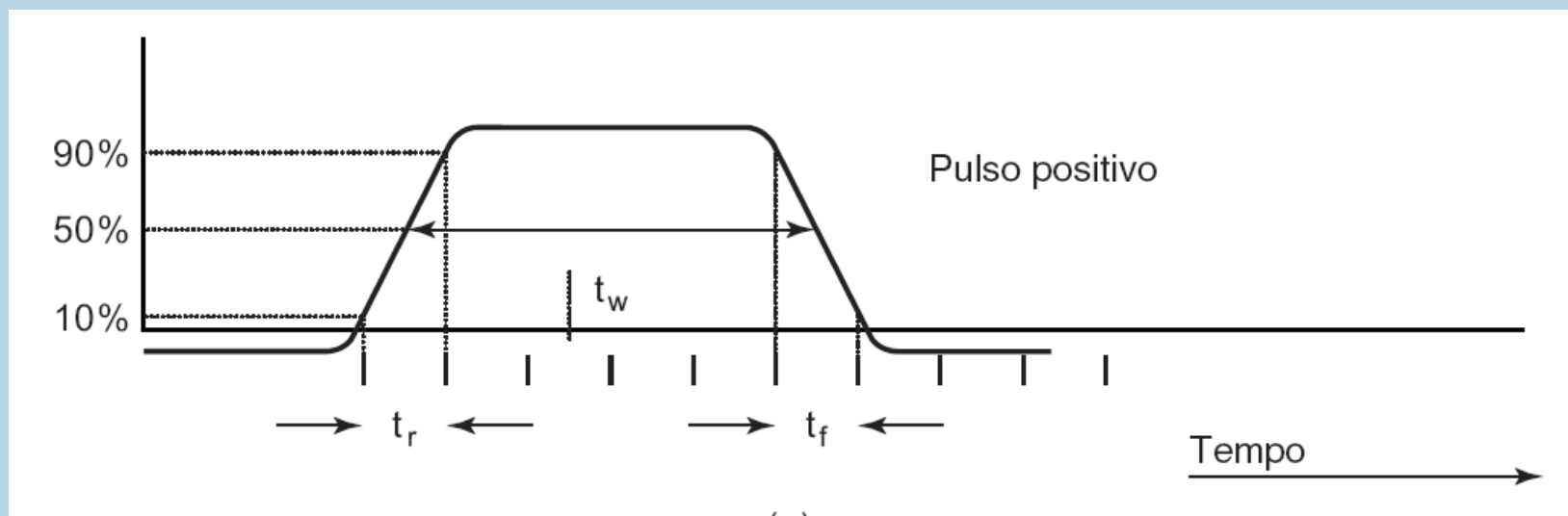
## 4.8 Circuitos para Habilitar/ Desabilitar





## 5.4 Pulsos Digitais

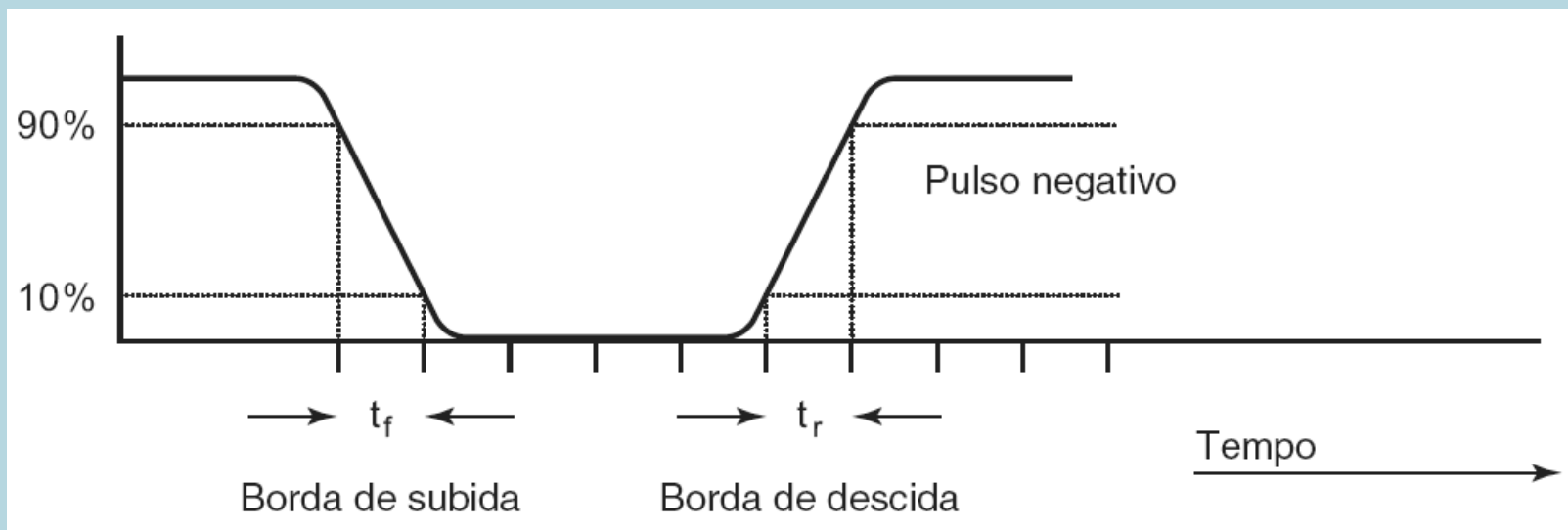
Sinais que se alternam entre os estados ativo e inativo são chamados de pulsos de onda.



Um pulso positivo tem um nível lógico ALTO.

## 5.4 Pulsos Digitais

Sinais que se alternam entre os estados ativo e inativo são chamados de pulso de onda.

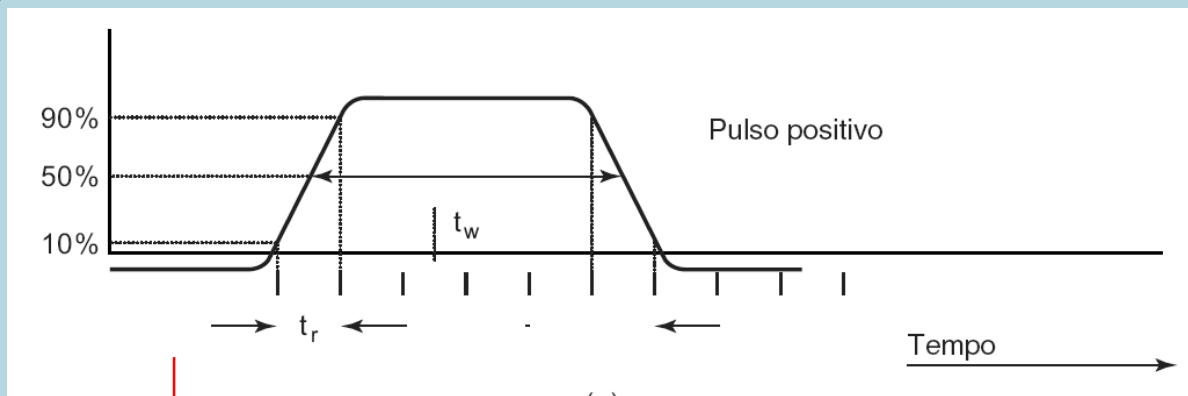


Um pulso negativo tem um nível ativo BAIXO.

## 5.4 Pulsos Digitais

- Em circuitos reais, leva tempo para uma onda de pulso mudar de um nível para outro.

A transição de BAIXO para ALTO em pulso positivo é chamada *tempo de subida* ( $t_r$ ).

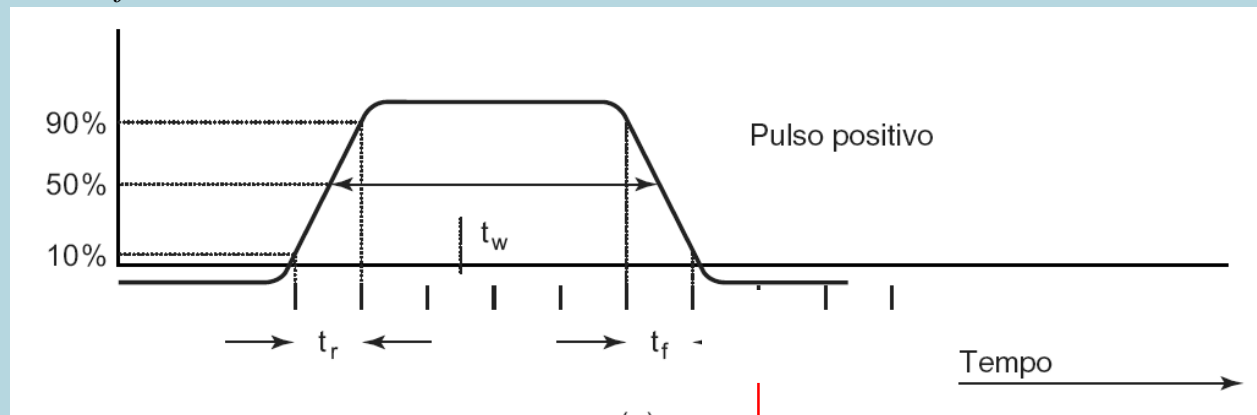


Medida entre 10% e 90% dos pontos no início da onda de tensão.

## 5.4 Pulsos Digitais

- Em circuitos reais, leva tempo para uma onda de pulso mudar de um nível para outro.

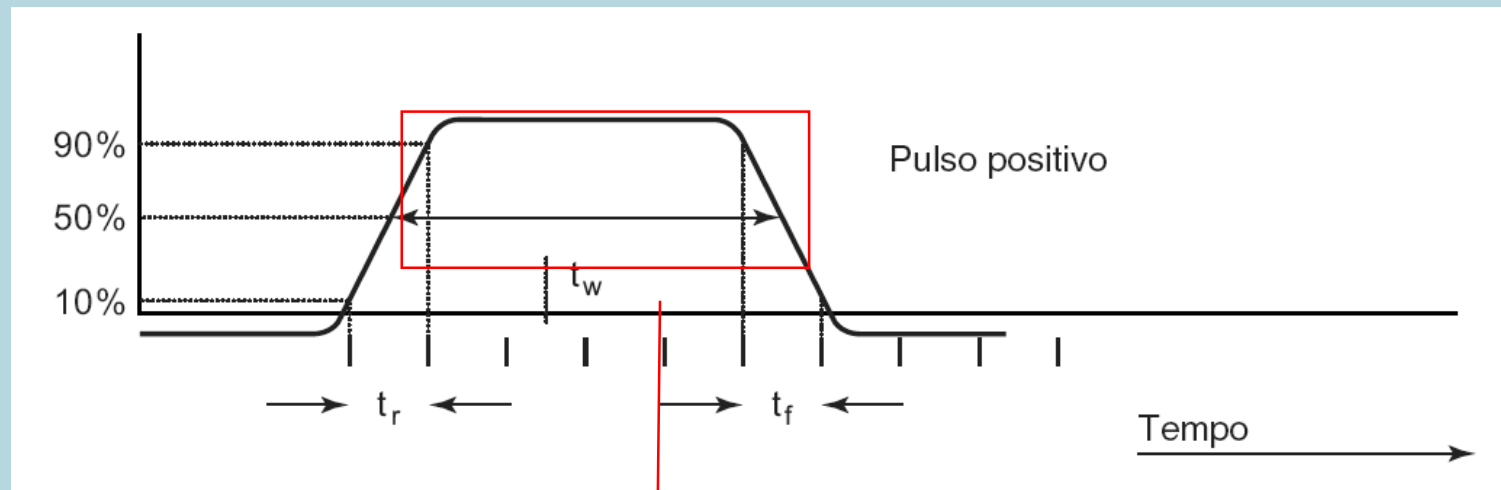
A transição de ALTO para BAIXO em um pulso positivo é chamada *tempo de descida* ( $t_f$ ).



Medida entre 10% e 90% dos pontos no final da onda de tensão.

## 5.4 Pulsos Digitais

- Em circuitos reais, leva tempo para uma onda de pulso mudar de um nível para outro. Um pulso também tem *duração* (largura) ( $t_w$ ).

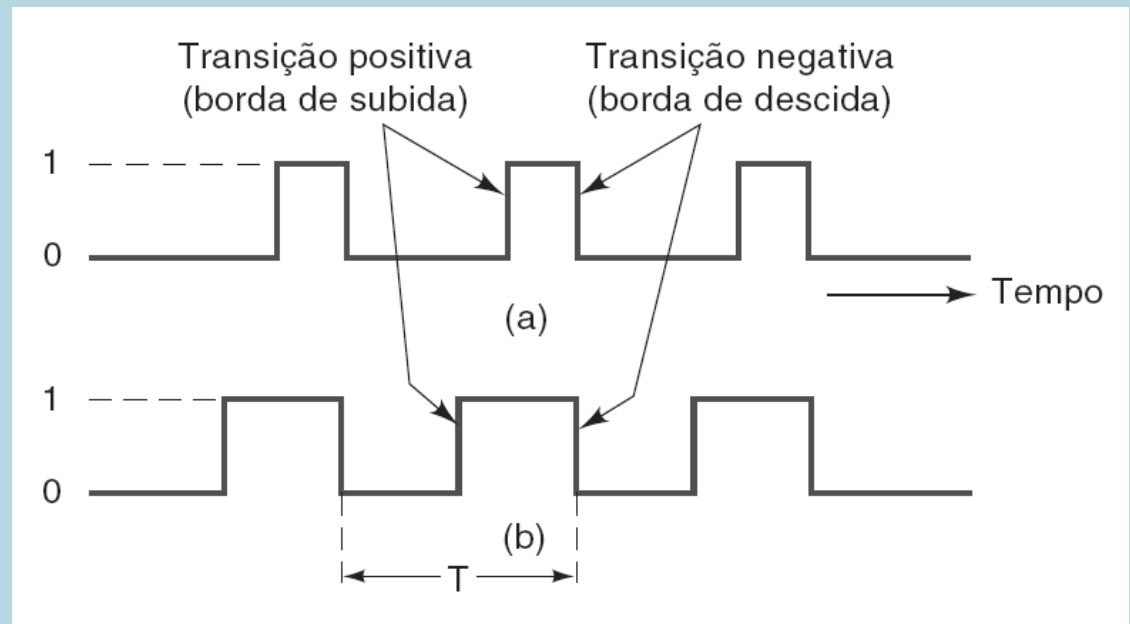


Tempo entre os pontos quando as bordas esquerda e direita estão em 50% da tensão de nível ALTO.

## 5.5 Sinais de Clock e Flip-Flops com Clock

- O sinal de clock é um trem de pulsos retangulares ou uma onda quadrada.  
Transição positiva (borda de subida): pulso do clock vai de 0 a 1.  
Transição negativa (borda de descida): pulso do clock vai de 1 a 0.

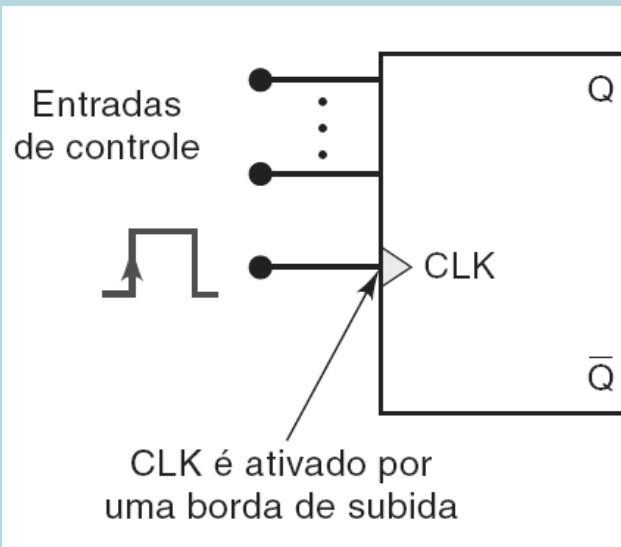
Transições também são chamadas de *bordas*.



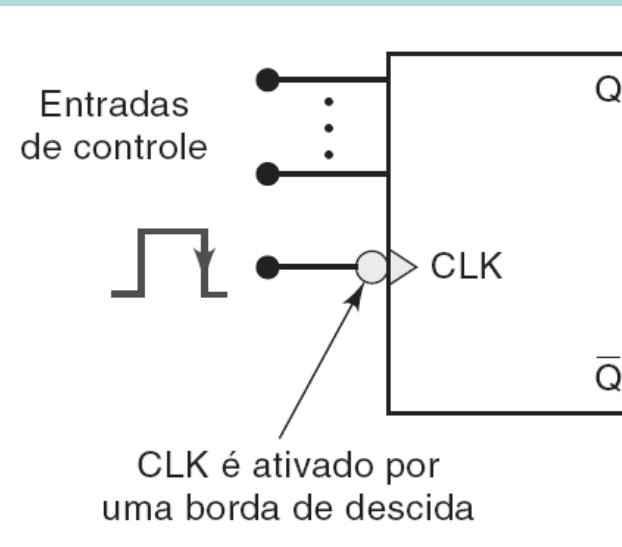
## 5.5 Sinais de Clock e Flip-Flops com Clock

FFs com clock mudam de estado em uma das transições do sinal de clock e têm entradas de clock denominadas CLK, CK, ou CP.

Um pequeno triângulo na entrada CLK indica que a entrada é ativada com um borda de subida.



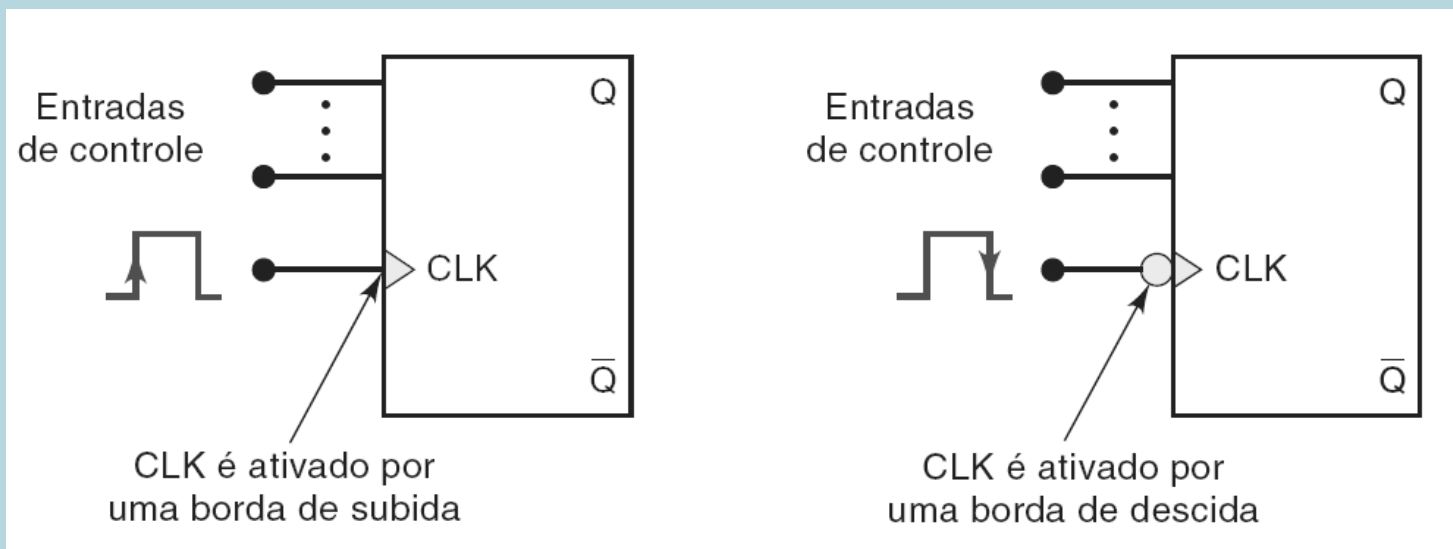
Uma bolha e um triângulo indicam que a entrada CLK é ativada com um borda de descida.



## 5.5 Sinais de Clock e Flip-Flops com Clock

Entradas de controle têm um efeito sobre a saída apenas na transição ativa do clock (borda de descida ou borda de subida), por isso são chamadas entradas de controle síncronos.

As entradas de controle preparam as saídas para mudar, mas a transição ativa da entrada CLK é que dispara a mudança de estado.



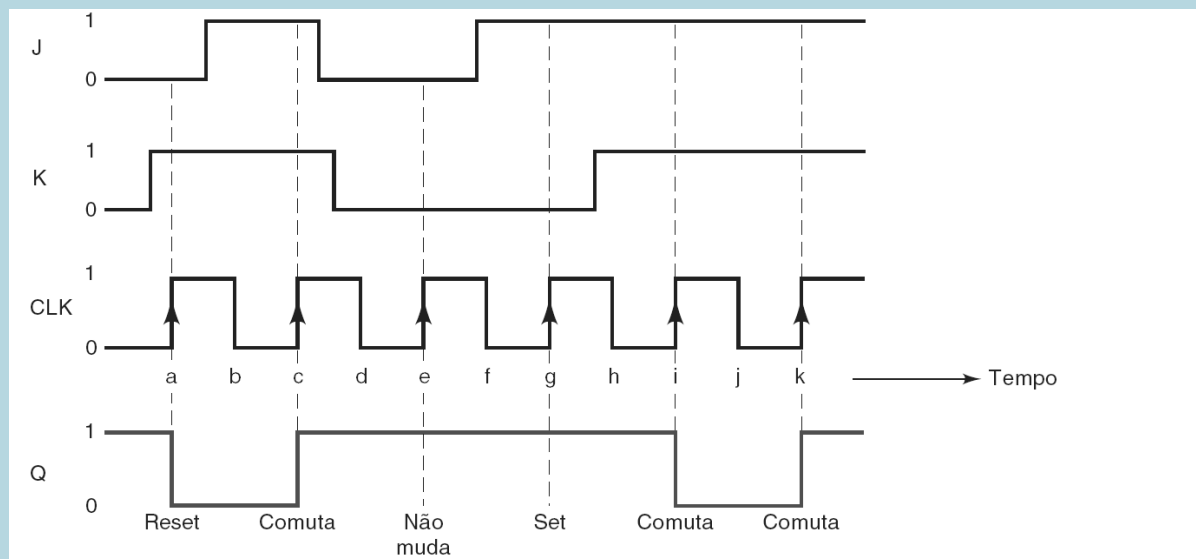
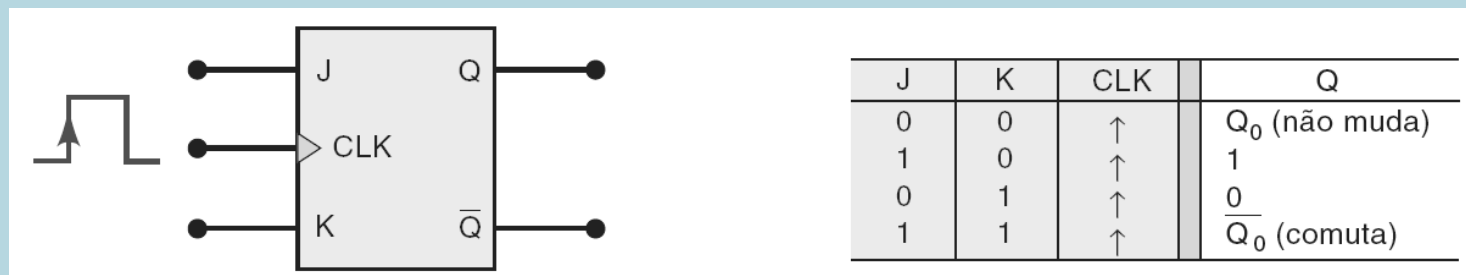


## 5.7 Flip-Flop J-K com Clock - Circuito Interno

- Opera como o FF S-R.
- J é SET, K é CLEAR.
- Quando J e K são ambos ALTO, a saída é alternada para o estado oposto.
- O gatilho do clock pode ser positivo ou negativo.
- Muito mais versátil do que o flip-flop SR, já que não tem estados ambíguos.
- Tem a capacidade de fazer tudo o que o FF SR faz, além de operar em modo de alternância.

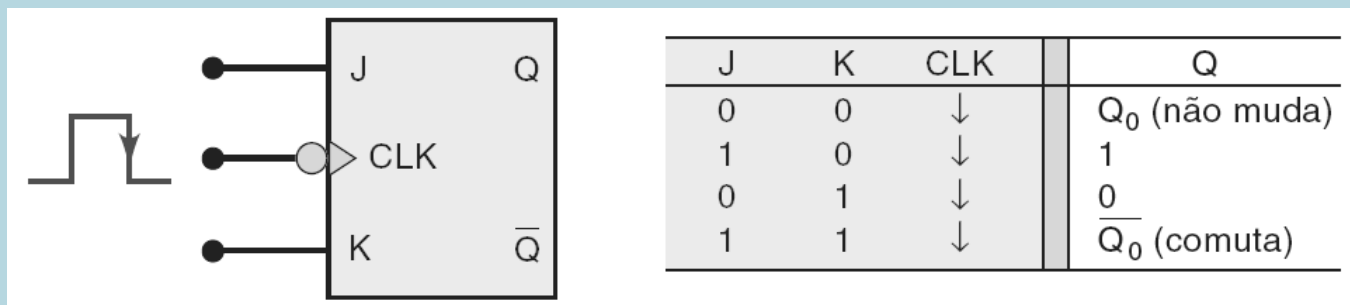
## 5.7 Flip-Flop J-K com Clock

Flip-flop JK com clock que responde apenas à borda de subida do clock.



## 5.7 Flip-Flop J-K com Clock

Flip-flop JK com clock que responde apenas à borda de descida do clock.

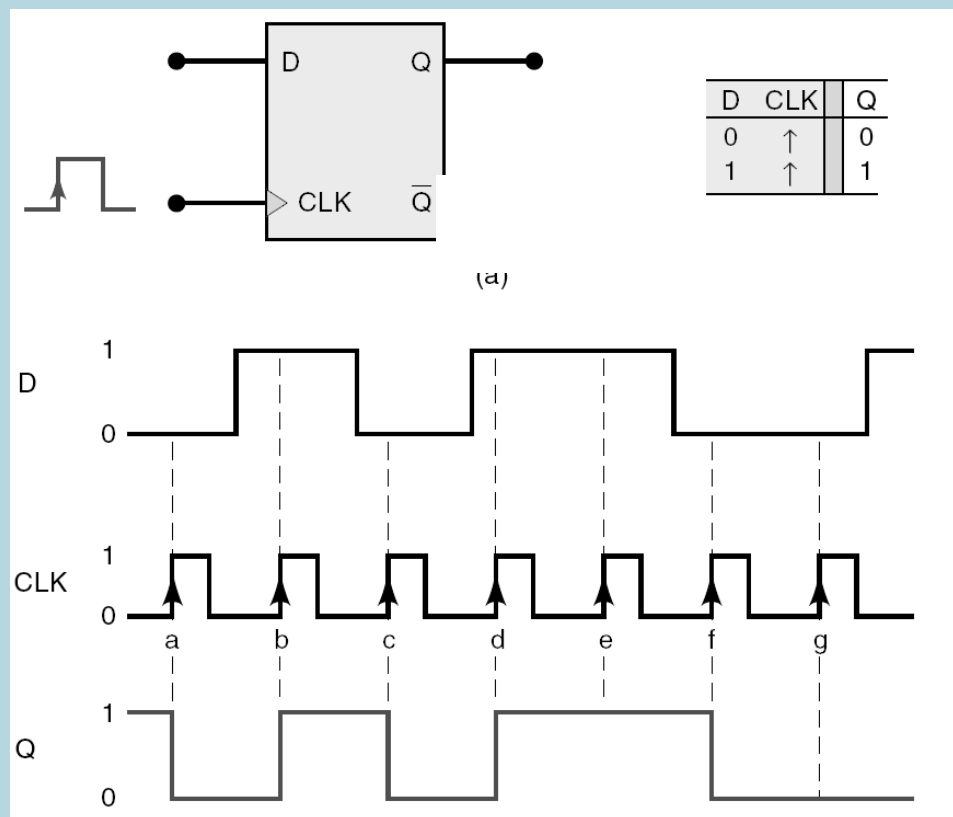


## 5.8 Flip-Flop D com Clock

- A saída muda para o valor da entrada tanto no gatilho positivo quanto no negativo do relógio.
- Pode ser implementado com um FF J-K ligando a entrada J à K, através de um INVERSOR.
- É útil para transferência de dados em paralelo.

## 5.8 Flip-Flop D com Clock

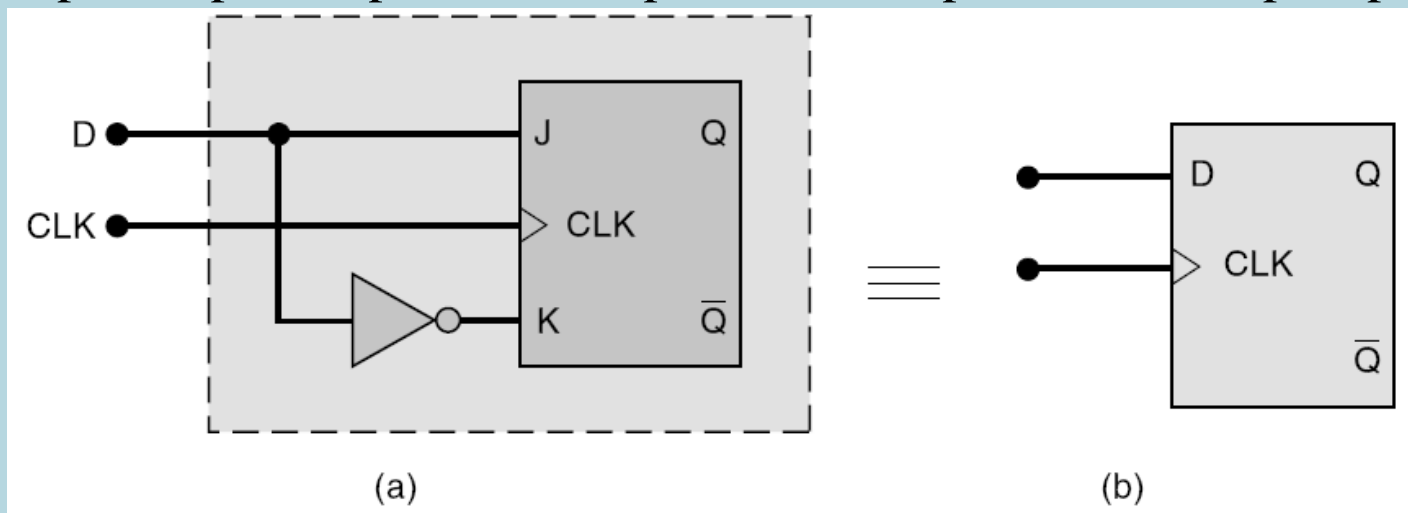
Flip-flop D acionado apenas em transições positivas.



## 5.8 Flip-Flop D com Clock

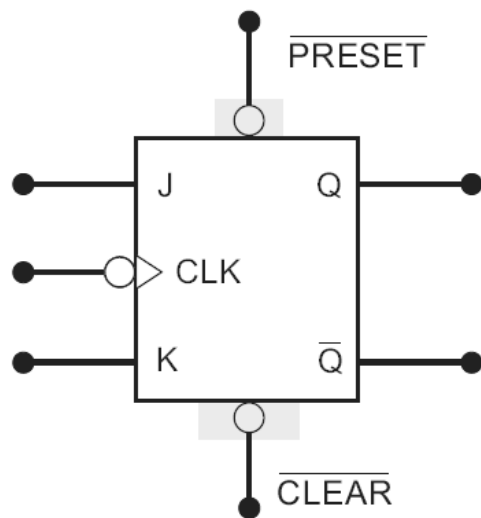
### Implementação

- Um flip-flop D disparado por borda é implementado pela adição de um único INVERSOR flip-flop J-K disparado por borda.
- O mesmo pode ser feito para converter um flip-flop S-R para um flip-flop D.
- Flip-flop D disparado por borda implementado a partir de um flip-flop J-K.



## 5.10 Entradas Assíncronas

Flip-flop J-K com clock com entradas assíncronas.



J	K	Clk	$\overline{\text{PRE}}$	$\overline{\text{CLR}}$	Q
0	0	↓	1	1	Q (não muda)
0	1	↓	1	1	0 (reset síncrono)
1	0	↓	1	1	1 (set síncrono)
1	1	↓	1	1	$\overline{\text{Q}}$ (toggle síncrono ou comutação síncrona)
x	x	x	1	1	Q (não muda)
x	x	x	1	0	0 (clear assíncrono)
x	x	x	0	1	1 (preset assíncrono)
x	x	x	0	0	(Inválido)

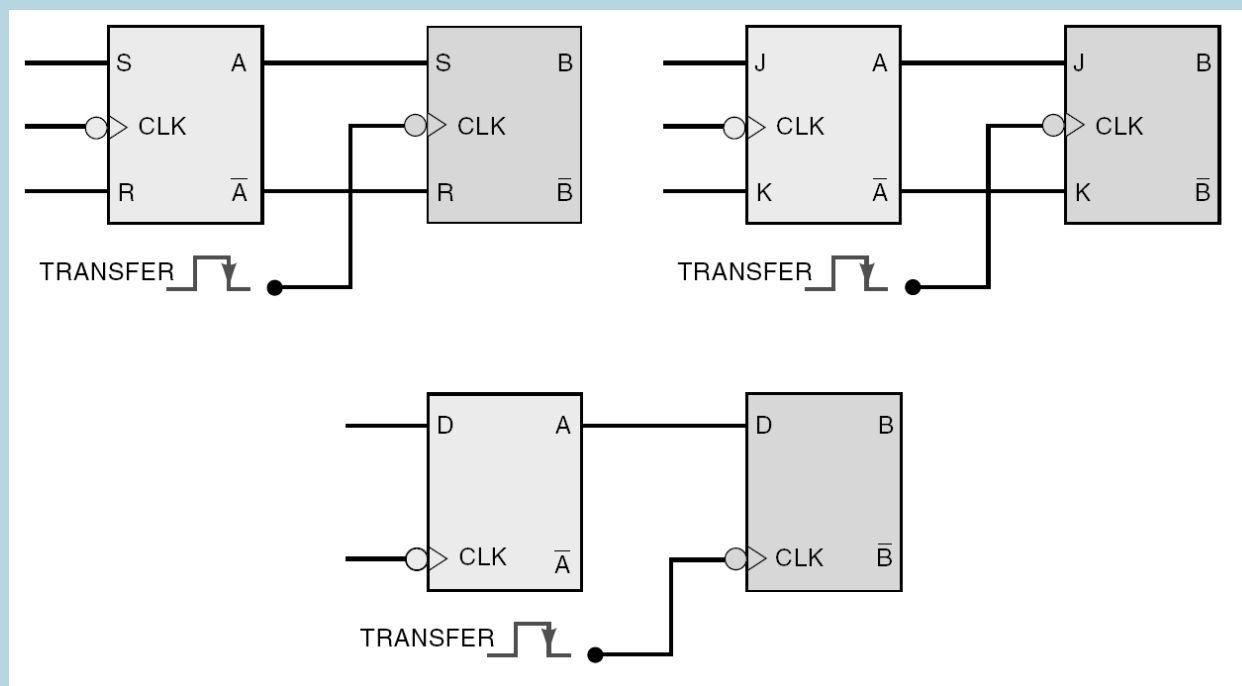
## 5.13 Aplicações com Flip-Flops

- Exemplos de aplicações:
  - Contagem.
  - Armazenamento de dados binários.
  - Transferência de dados entre locais.
- Muitas aplicações FF são continuamente categorizadas.
  - As saídas seguem uma sequência predeterminada de estados.



## 5.16 Armazenamento e Transferência de Dados

Operação de transferência síncrona de dados por vários FFs com clock.

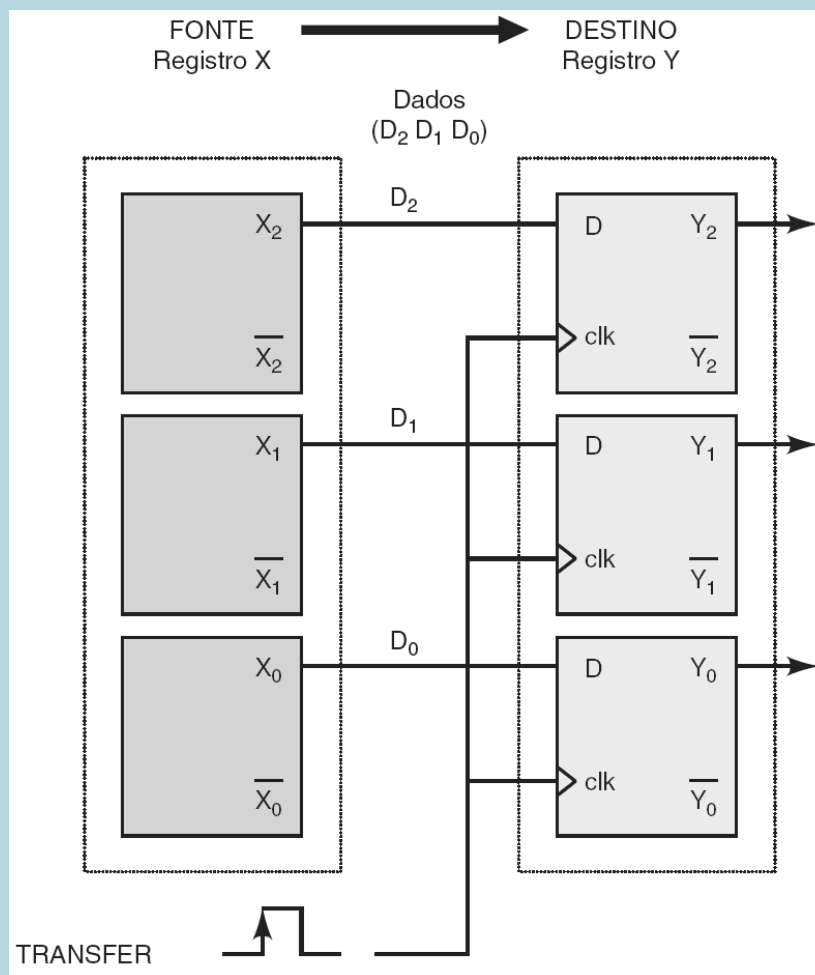


Entradas CLK são usados para executar a transferência.

## 5.16 Armazenamento e Transferência de Dados

A transferência simultânea de dados de um registrador para outro é denominada *paralela*.

Se a transferência for realizada um bit de cada vez será denominada *serial*.



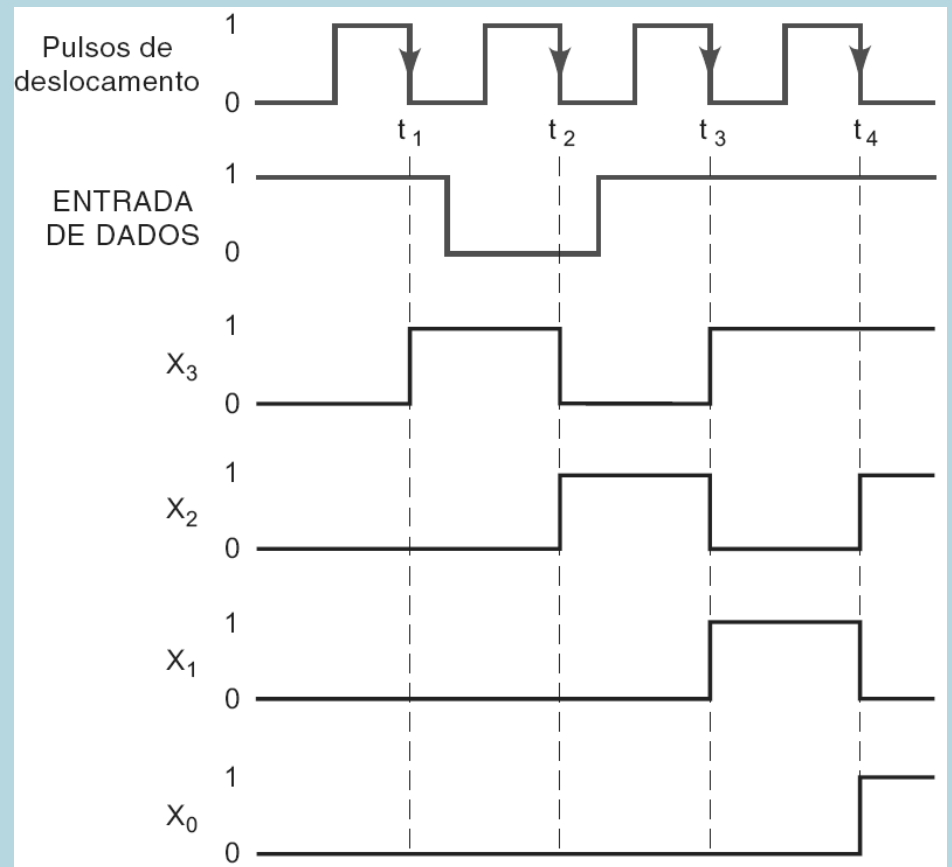


## 5.17 Transferência Serial de Dados: Registradores de Deslocamento

Dados de entrada são deslocados da esquerda para a direita, de um FF para outro, à medida em que pulsos de deslocamento são aplicadas.

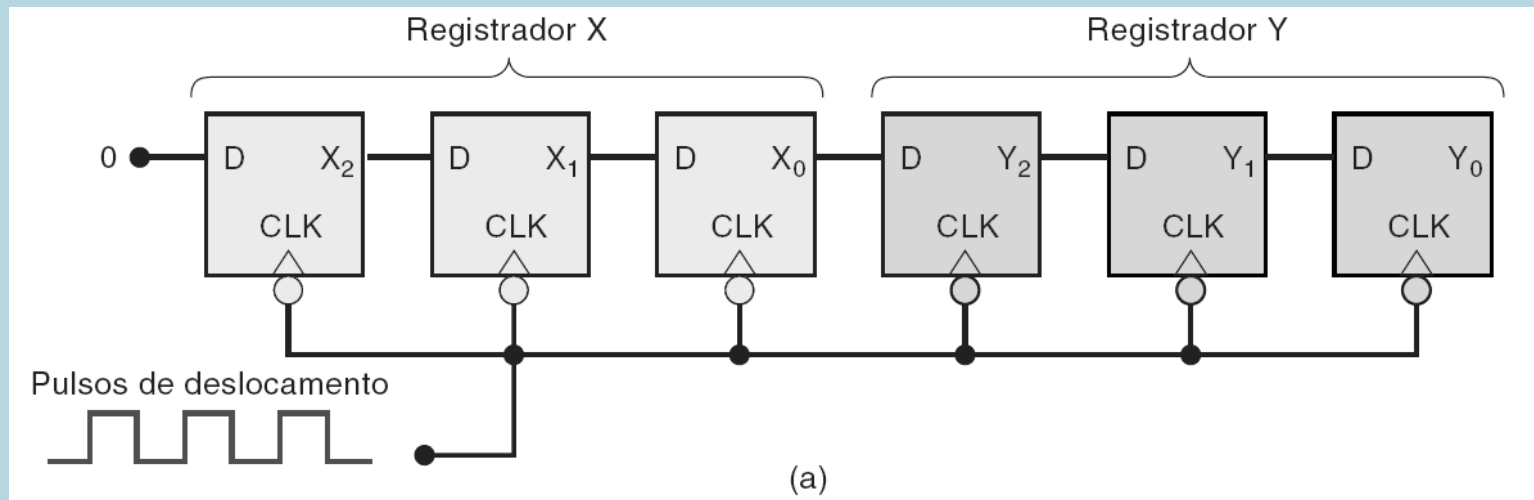
- Neste arranjo de registrador de deslocamento, é necessário ter FFs com requisitos de espera (tempo de hold) muito pequenos.

Há momentos em que as entradas  $J$ ,  $K$  mudam de estado simultaneamente à transição  $CLK$ .



## 5.17 Transferência Serial de Dados: Registradores de Deslocamento

Dois registradores de mudança de três bits conectados.



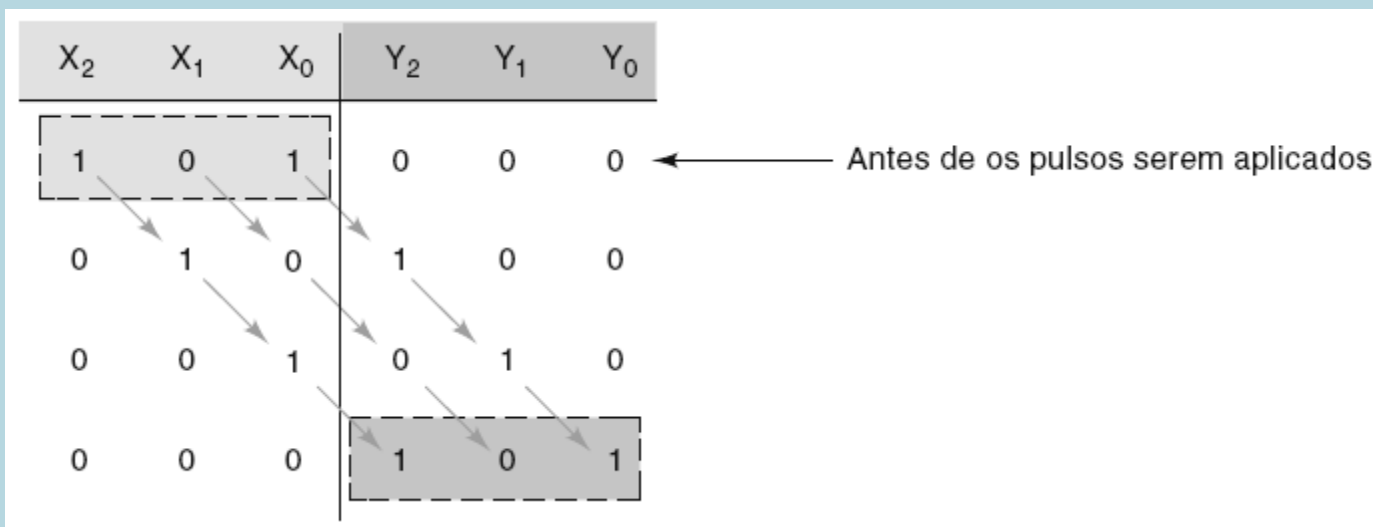
O conteúdo do registrador X será transferido em série (deslocado) para o registrador Y.

Os flip-flops D em cada registrador requerem menos conexões que os flip-flops J-K.

## 5.17 Transferência Serial de Dados: Registradores de Deslocamento

Dois registradores de mudança de três bits conectados.

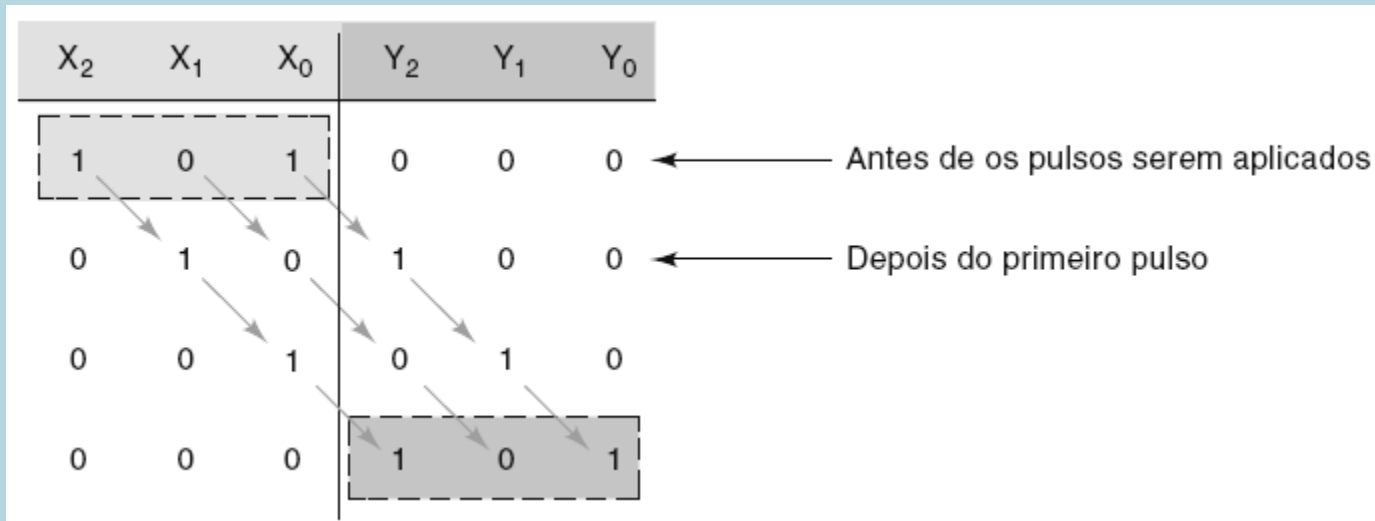
A transferência completa dos três bits de dados exige três pulsos de deslocamento.



## 5.17 Transferência Serial de Dados: Registradores de Deslocamento

Dois registradores de mudança de três bits conectados.

Na borda de descida de cada pulso, cada FF assume o valor armazenado no FF à sua esquerda, antes do pulso.

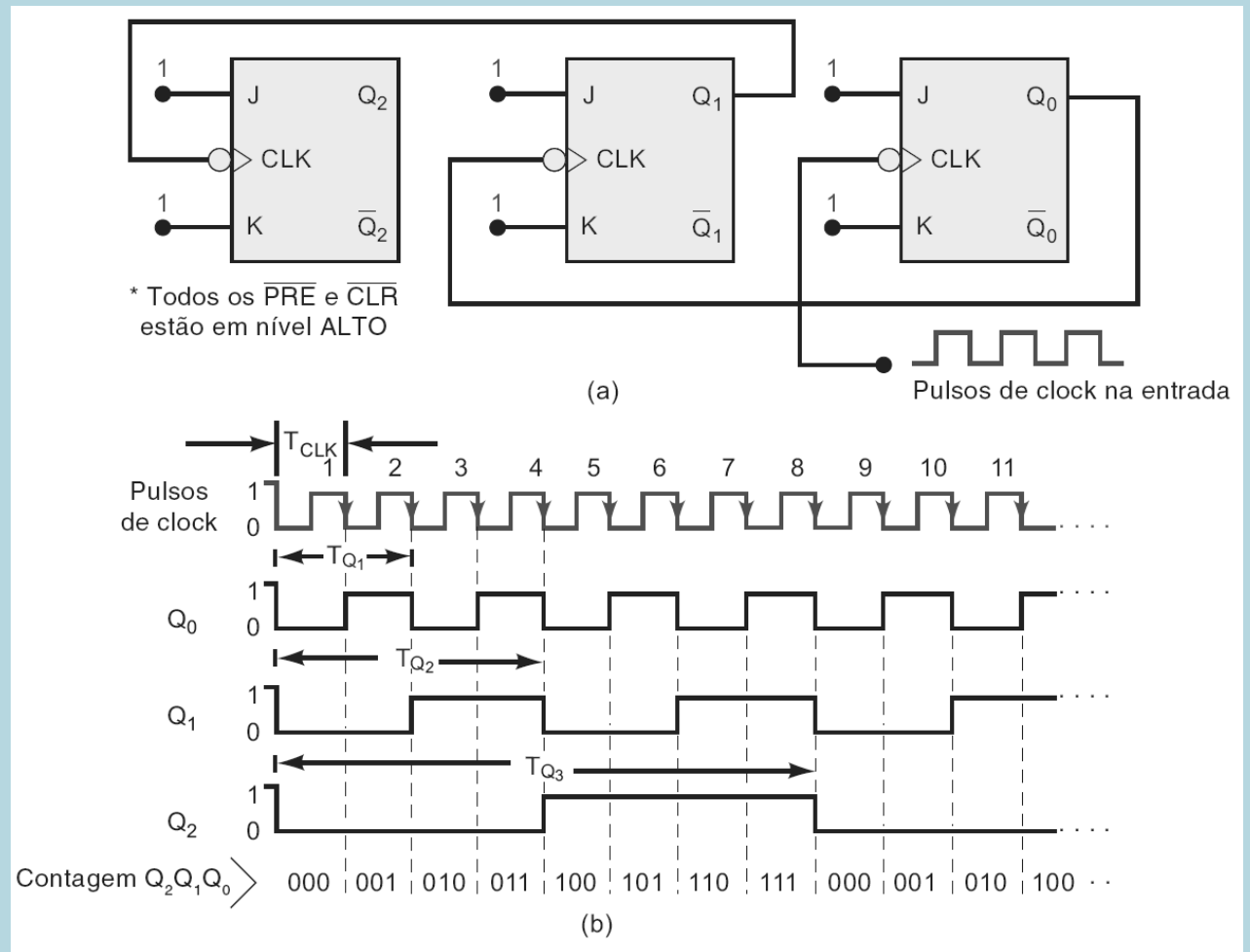


## 5.18 Divisão de Frequência e Contagem

Flip-flops J-K configurados como um contador binário de três bits (módulo 8).

Saídas irão contar de  $000_2$  a  $111_2$  ou de  $0_{10}$  a  $7_{10}$ .

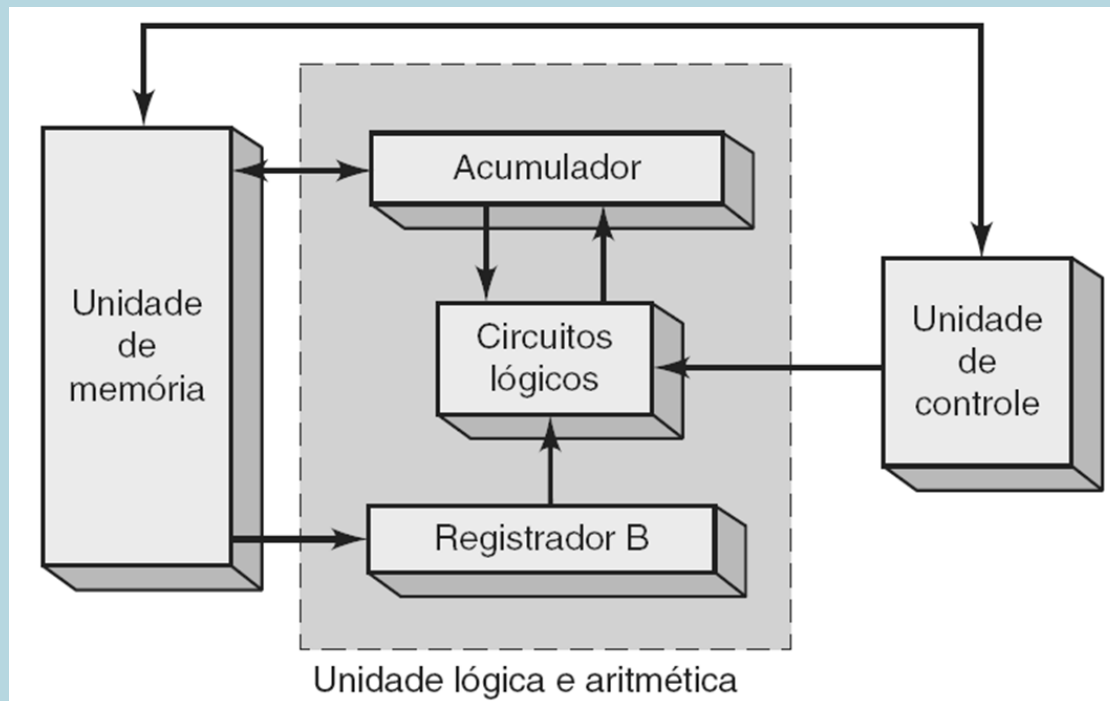
O número de estados possíveis em um contador é o número de módulo ou MOD.





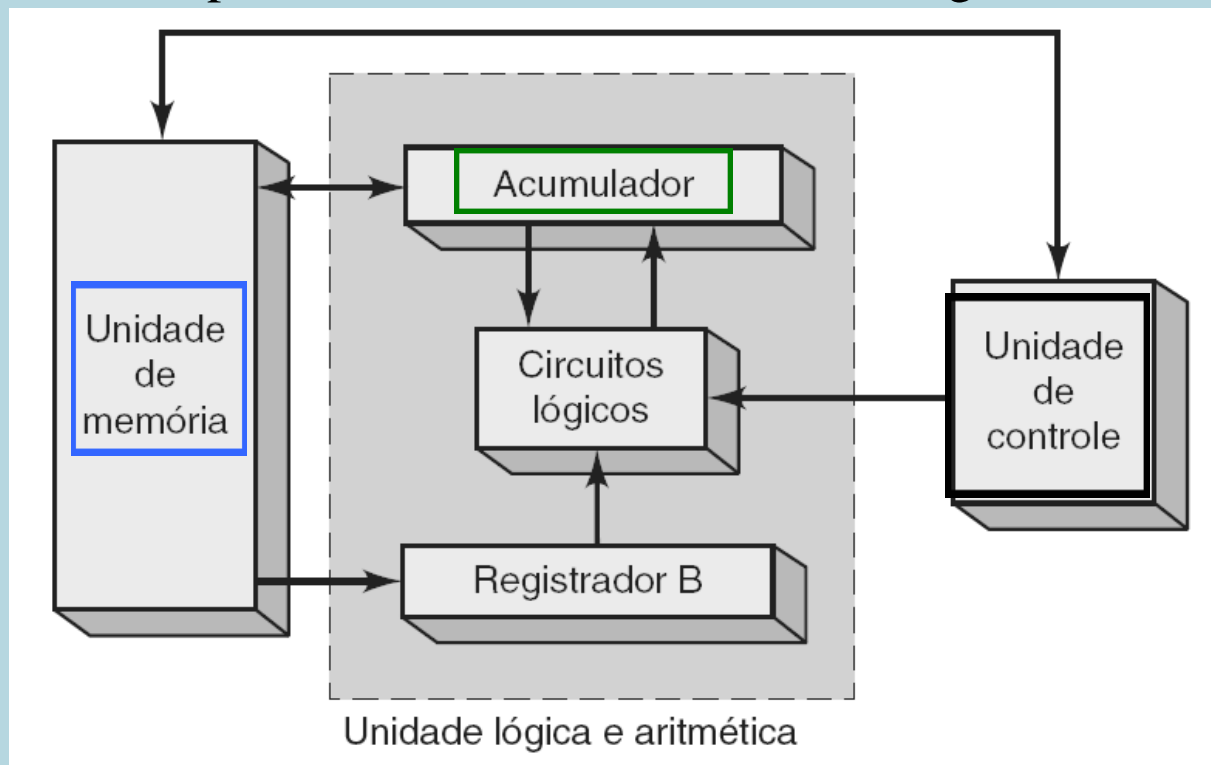
## 6.9 Circuitos Aritméticos

- Uma unidade lógica e aritmética (ULA) recebe os dados armazenados na memória e executa operações aritméticas e lógicas com instruções provenientes da unidade de controle.



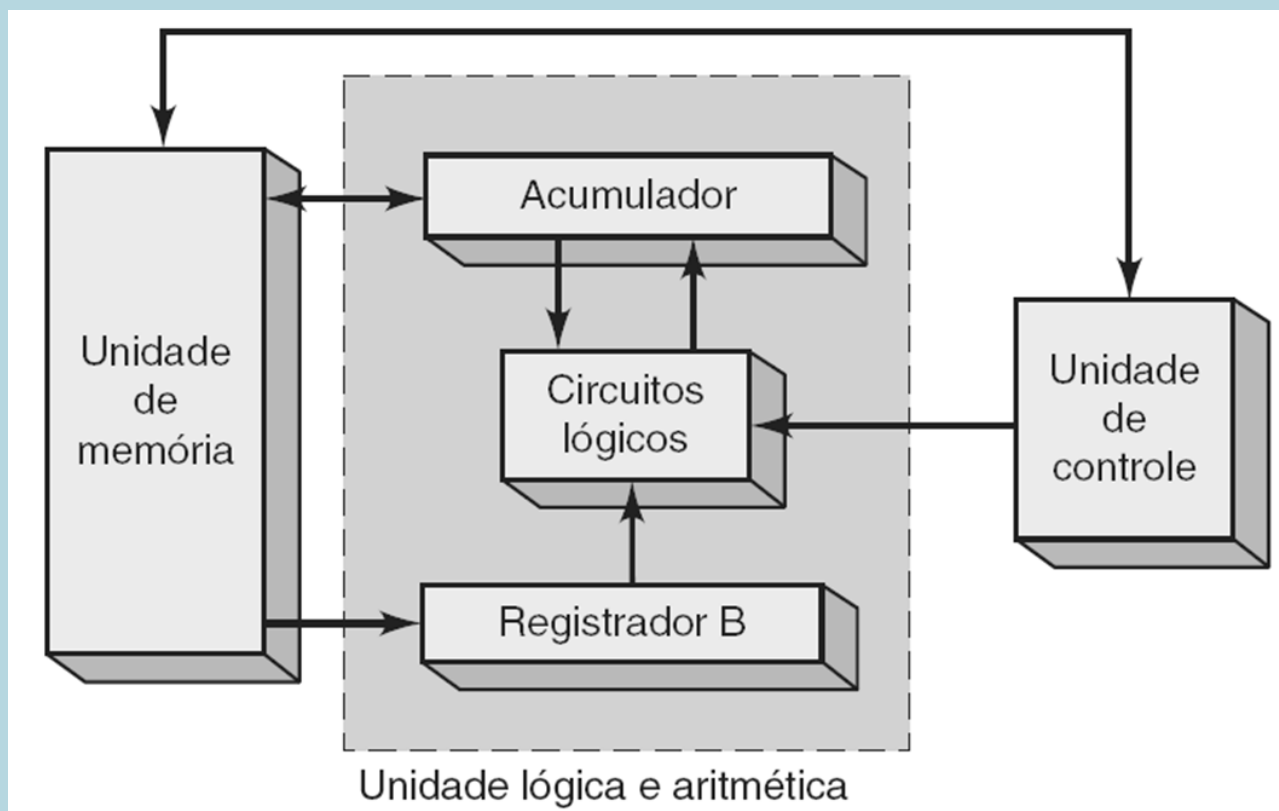
## 6.9 Circuitos Aritméticos

- A unidade de controle é instruída a adicionar um número específico de um local da memória para um número armazenado no registrador acumulador.



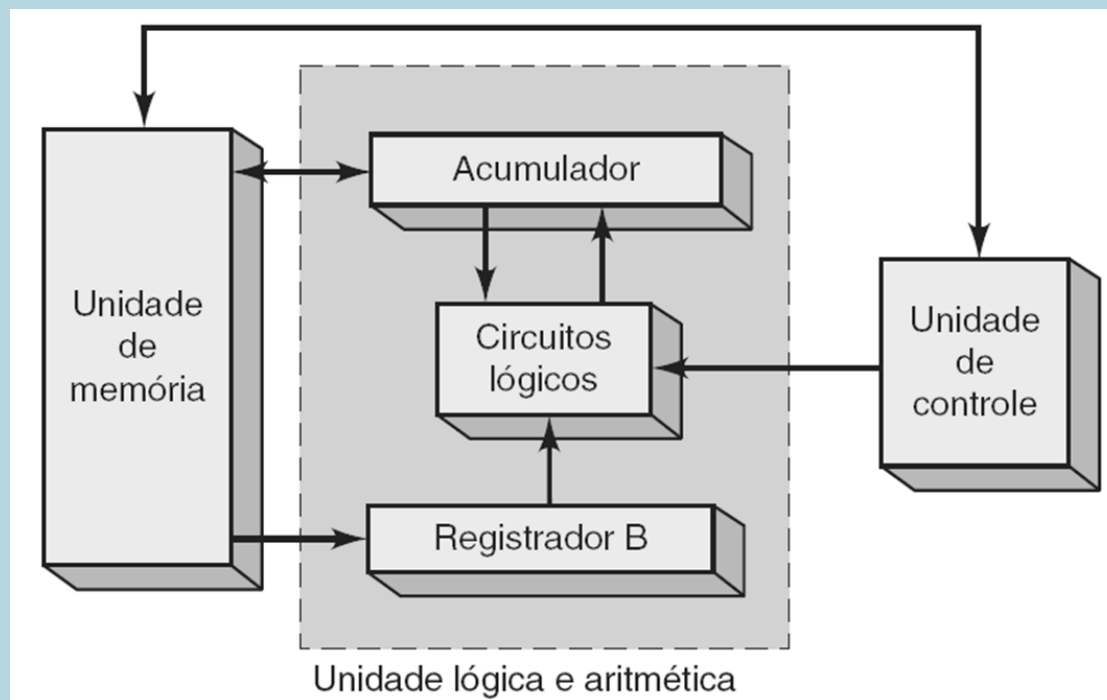
## 6.9 Circuitos Aritméticos

- O número é transferido da memória para o registrador B.



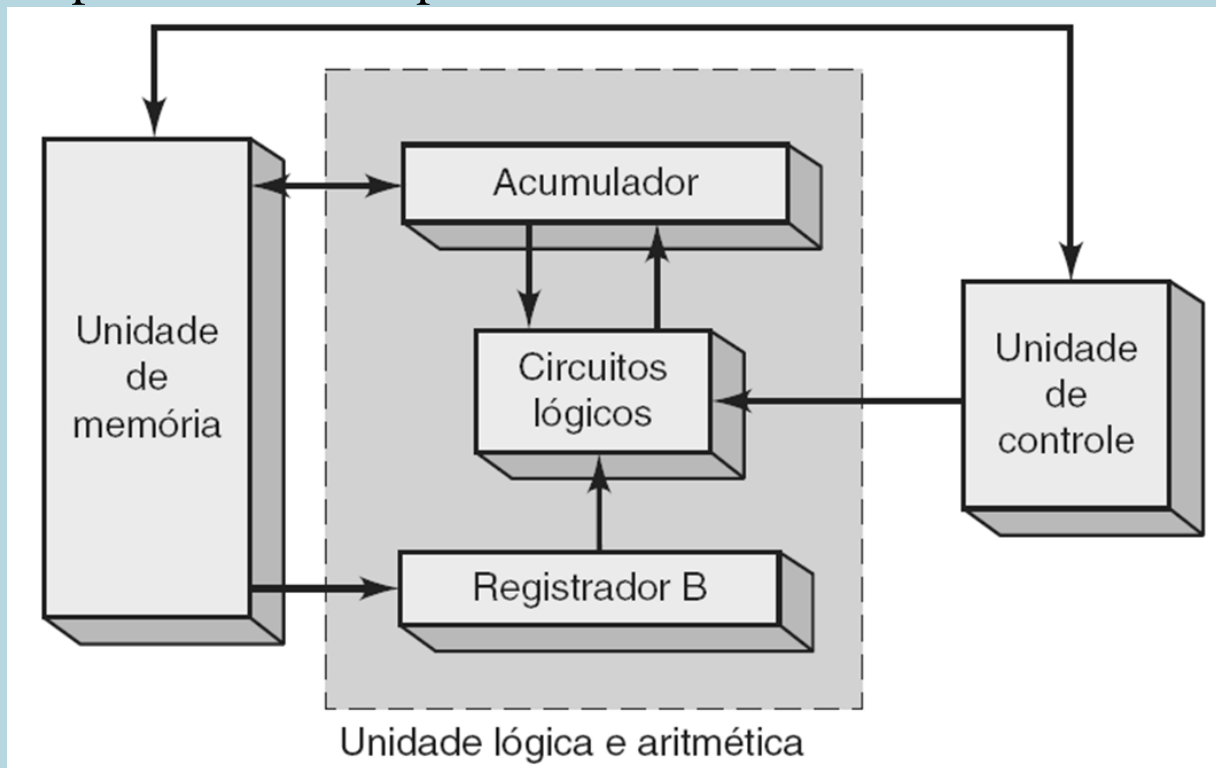
## 6.9 Circuitos Aritméticos

- O número no registrador B e o número do registrador acumulador são somados no circuito lógico e o resultado é enviado para o acumulador para ser armazenado.



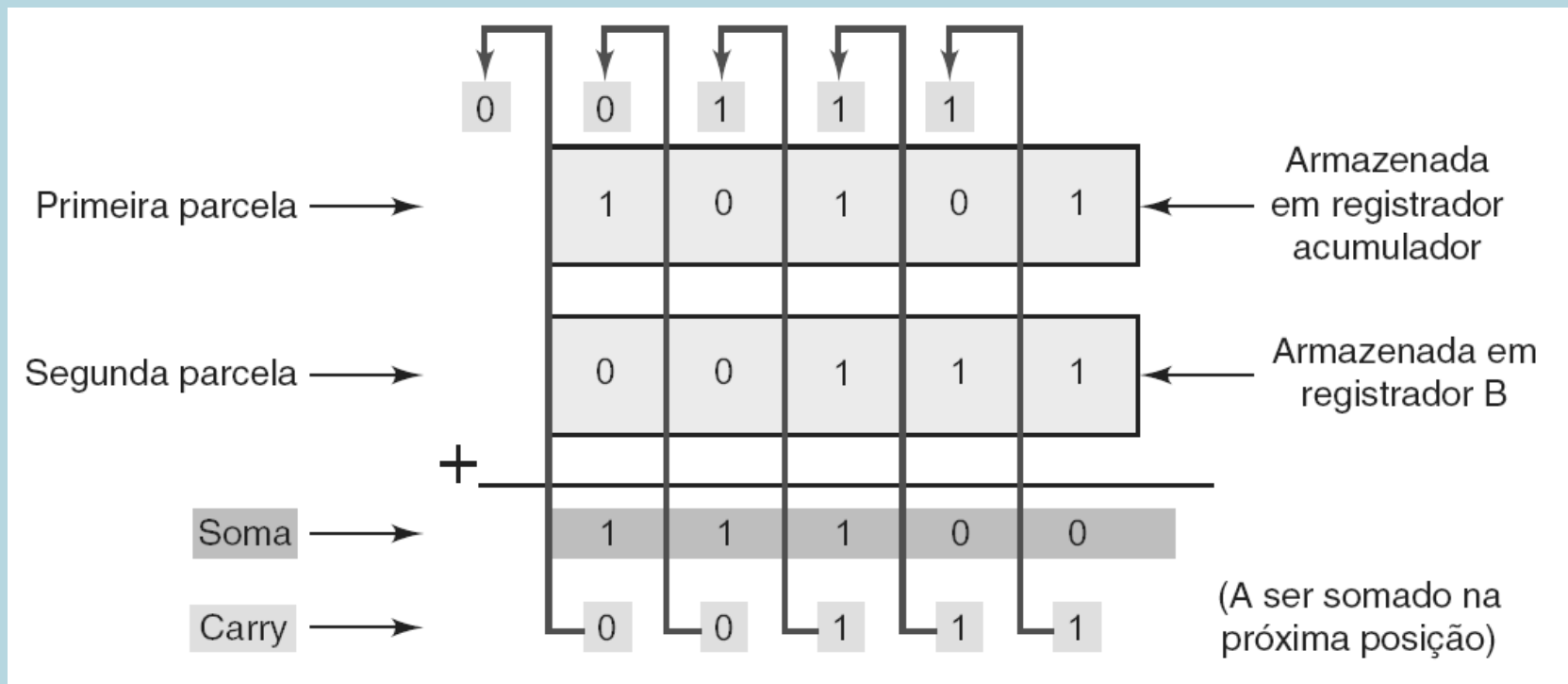
## 6.9 Circuitos Aritméticos

- O novo número permanece no acumulador para outras operações ou pode ser transferido para a memória para ser armazenado.



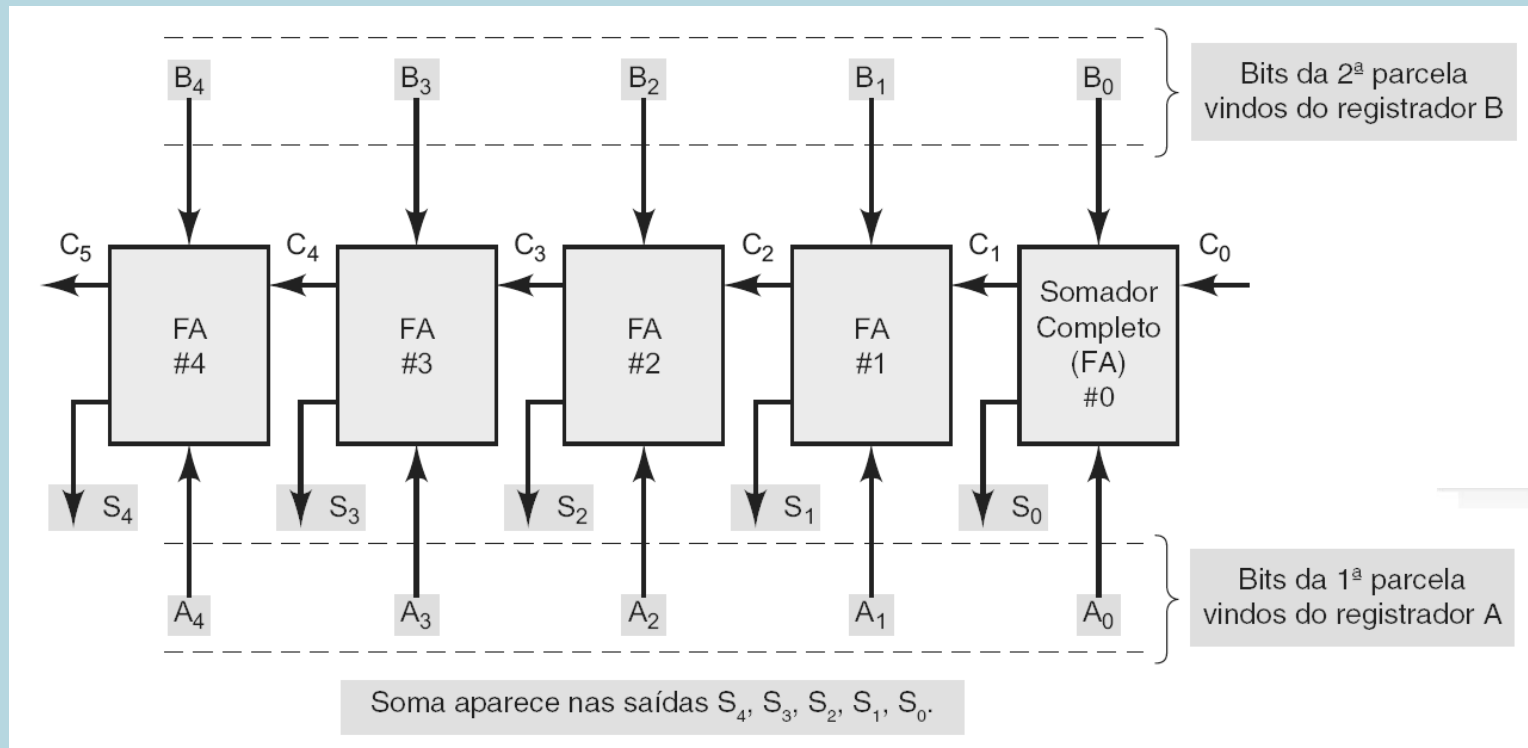
## 6.10 Somador Binário Paralelo

- Computadores e calculadoras realizam operações de adição sobre dois números de cada vez em que cada número binário pode ter vários dígitos binários



## 6.10 Somador Binário Paralelo

Diagrama em blocos de um circuito somador paralelo de cinco bits utilizando somadores completos.



## 6.11 Projeto de um Somador Completo

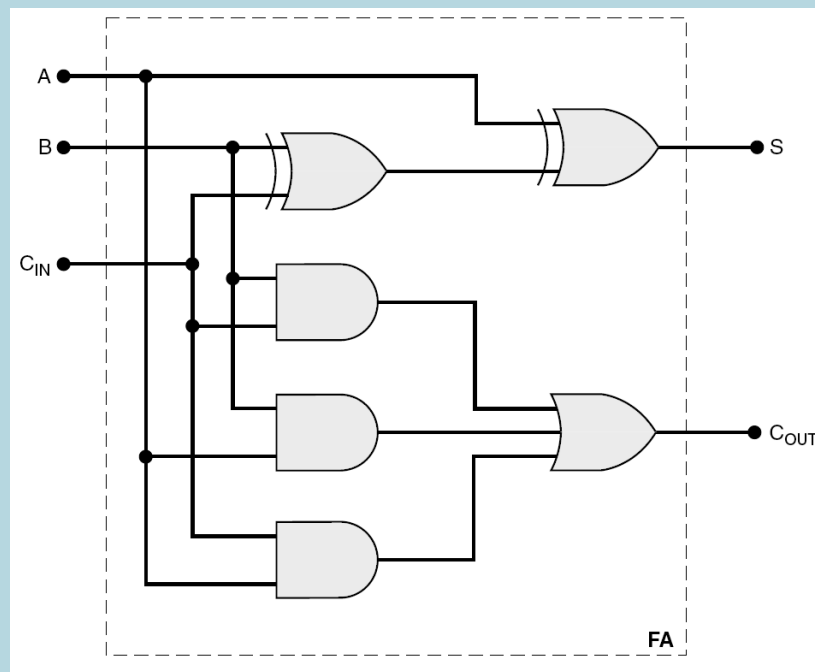
- Construindo uma tabela-verdade com:
  - 3 entradas (2 números para serem adicionados e carry in).
  - 2 saídas (soma e carry out).

Entradas de bits da primeira parcela	Entradas de bits da segunda parcela	Entradas de bits do carry	Saída de bits da soma	Saída de bits do carry
A	B	$C_{IN}$	S	$C_{OUT}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



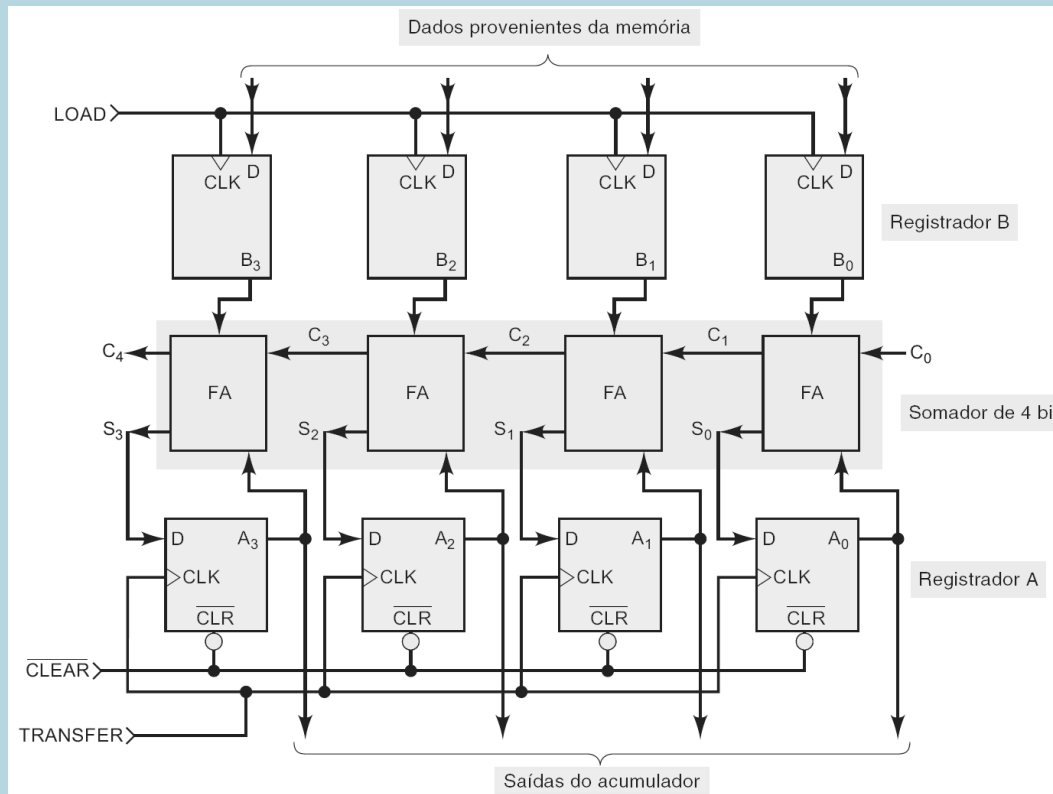
## 6.11 Projeto de um Somador Completo

- Usando métodos algébricos ou mapas K para simplificar a forma SOP resultante, o circuito completo com as entradas A, B e  $C_{IN}$  e as saídas S e  $C_{OUT}$  representará o somador completo (FA).



## 6.12 Somador Paralelo Completo com Registradores

Circuito somador paralelo de quatro bits, incluindo os registradores de armazenamento.



## 6.12 Somador Paralelo Completo com Registradores

- Processo pelo qual o circuito anterior soma os binários 1001 e 0101:
  - Um pulso CLR será aplicado nas entradas assíncronas, no instante  $t_1$ .
  - O número binário 1001 será transferido da memória para o registrador B, em  $t_2$ .
  - .
  - A soma do 1001 e 0000 será transferida para o registrador A, em  $t_3$ .
  - O número binário 0101 será transferido da memória para o registrador B, em  $t_4$ .
  - .
  - Os resultados da soma serão transferidos para o registrador A, em  $t_5$ .
  - A soma dos dois números estará presente no acumulador.

## 6.12 Somador Paralelo Completo com Registradores

- Colchetes indicam o conteúdo de um registrador.  
Por exemplo:  $[A] = 1011$  equivale a  $A_3 = 1, A_2 = 0, A_1 = 1, A_0 = 1$   
Pode-se pensar em  $[A]$  como o conteúdo do registrador A.
- A transferência de dados para um registrador ou a partir dele pode ser indicada com uma seta:

$$[B] \rightarrow [A]$$

*(O conteúdo do registrador B foi transferido para o A)*

## 6.13 Propagação do Carry

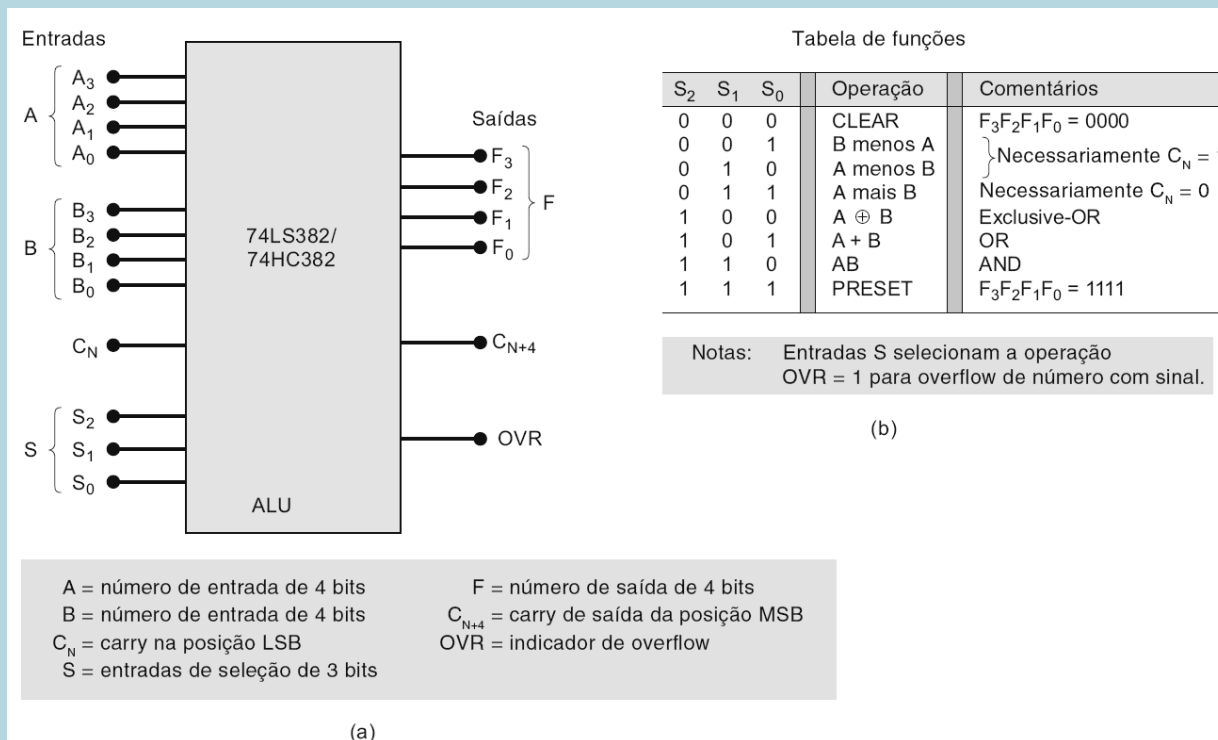
- A velocidade de um somador paralelo é limitada por um efeito denominado propagação do carry (ou *carry ripple*):
  - A adição dos bits da 1ª posição gera um carry para a posição seguinte.
  - O último carry gera outro para a última posição (MSB).
  - A soma do bit gerado na última posição depende do carry que foi gerado na adição da primeira posição (LSB).
- O esquema do carry antecipado (ou *look-ahead carry*) é usado em dispositivos de alta velocidade para reduzir o atraso de propagação.

## 6.16 Circuito Integrado ALU

- Existem vários CIs disponíveis denominados unidades lógicas e aritméticas (*arithmetic/logic units* - ALUs), ainda que não tenham toda a capacidade de uma ALU de um computador.
- Esses chips ALU são capazes de realizar diversas operações lógicas e aritméticas sobre dados binários de entrada.
- A operação específica realizada pela ALU é determinada por um código binário específico colocado nas entradas de seleção de funções.
- ALUs podem realizar operações lógicas e aritméticas diferentes como determinado por um código binário nas entradas selecionadoras de função.

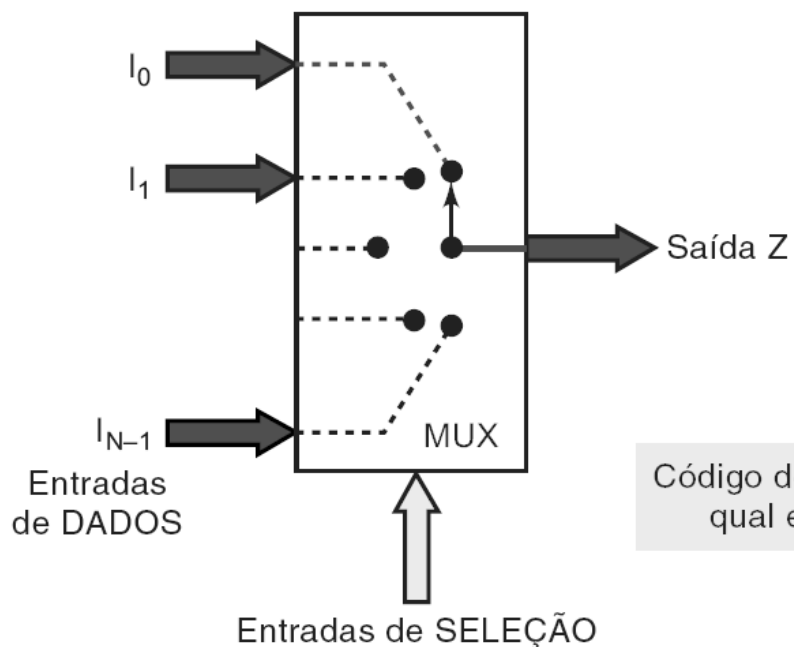
## 6.16 Circuito Integrado ALU

A ALU 74LS382 (TTL) e 74HC382 (CMOS) é um dispositivo que pode realizar oito funções diferentes.



## 9-6 Multiplexadores (Seletores de dados)

- Um **multiplexador (MUX)** seleciona uma de N fontes de entrada de dados e transmite os dados selecionados para uma única saída.
- Um *multiplexador digital* ou seletor de dados é um circuito lógico que executa a mesma tarefa.



O controle de roteamento da entrada de dados desejada para a saída feito pelas

entradas de SELEÇÃO

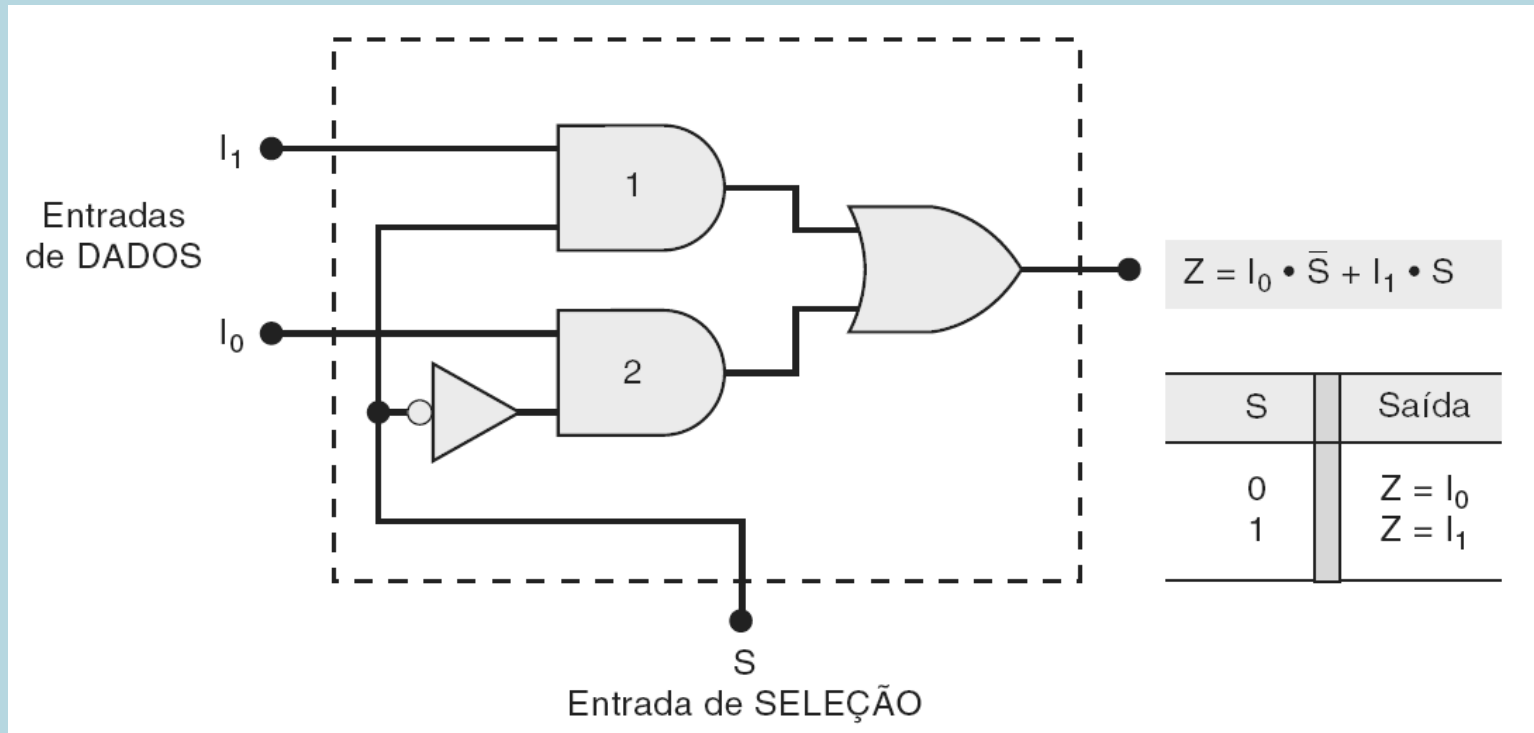
Código de entrada de SELEÇÃO determina qual entrada é transmitida à saída Z

referidas como entradas



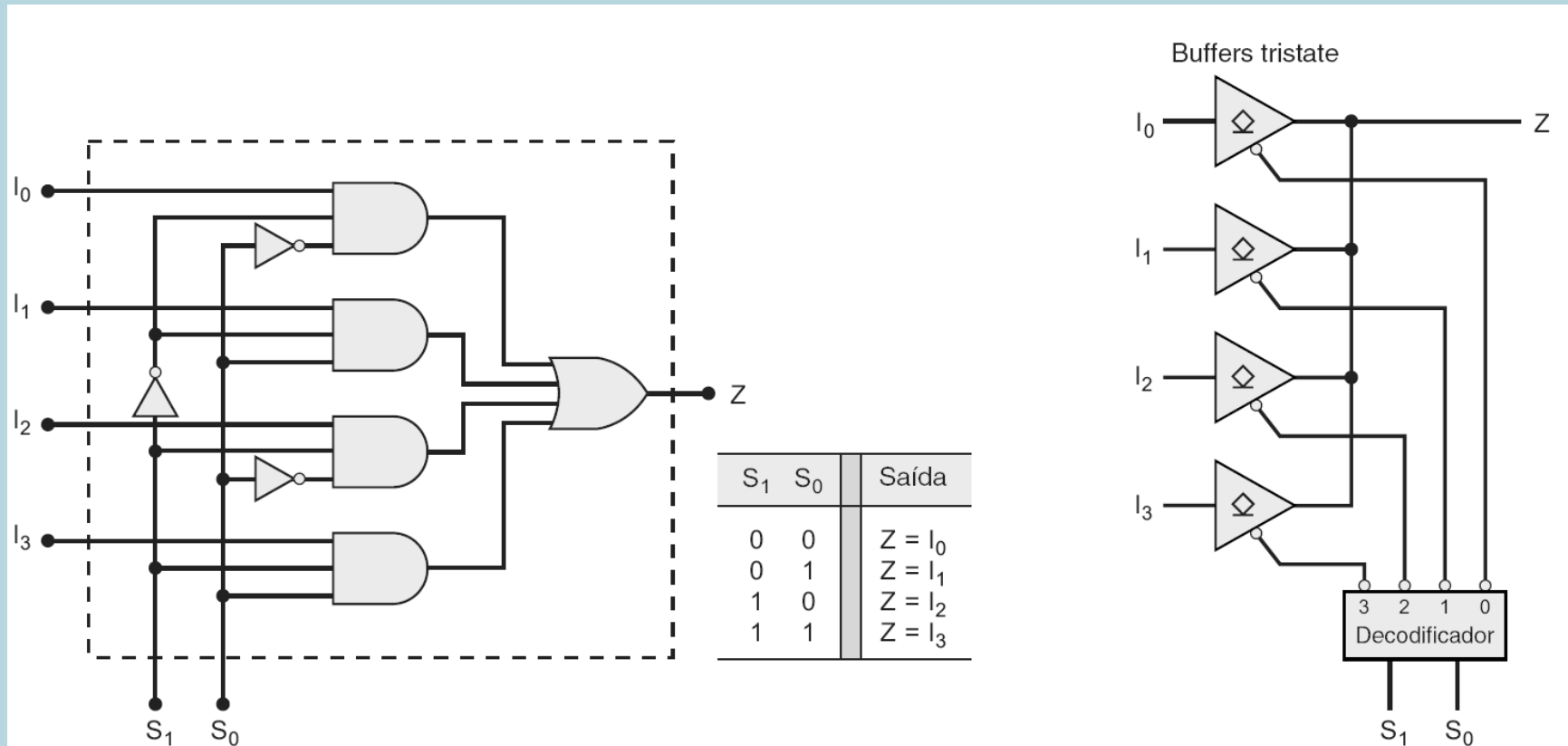
## 9-6 Multiplexadores (Seletores de dados)

- Um MUX de duas entradas pode ser usado em um sistema digital que usa dois sinais MASTER CLOCK diferentes: um clock de alta velocidade em um dos modos e um clock de baixa velocidade no outro.



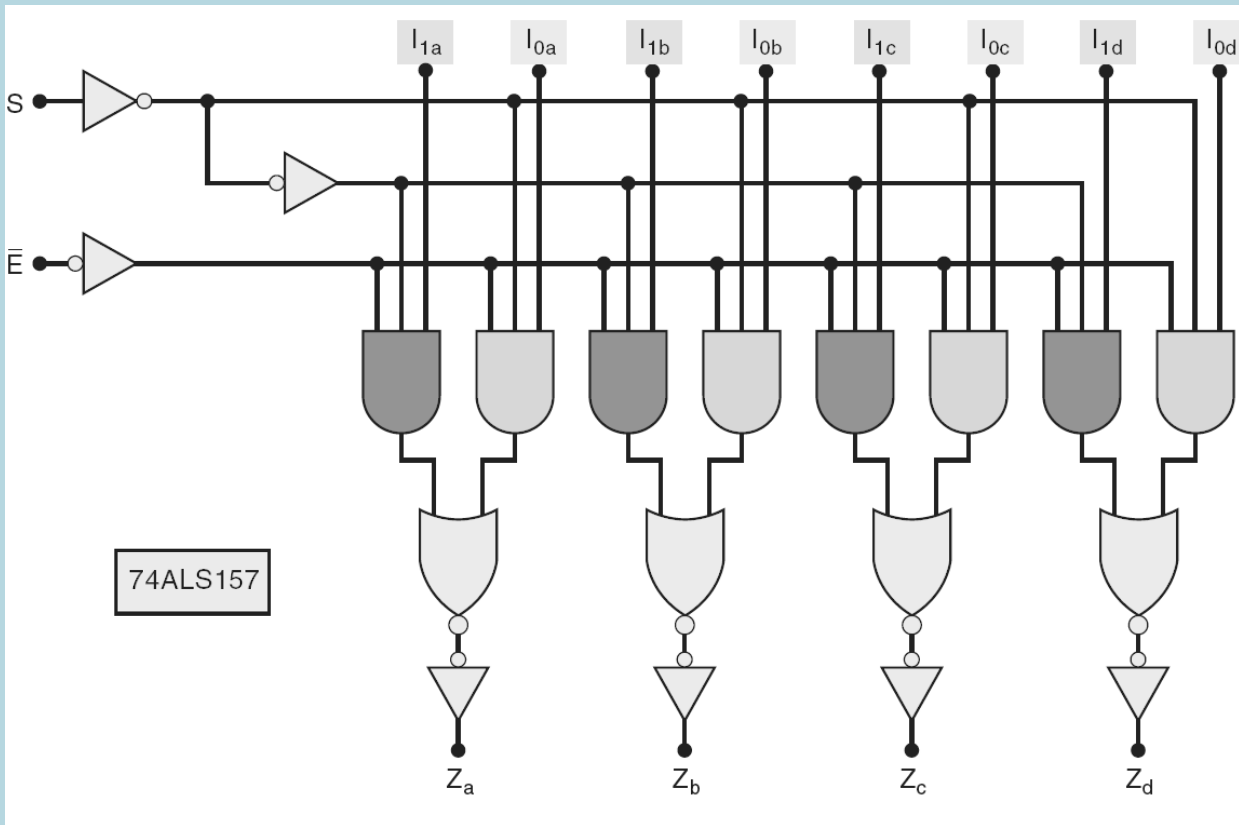
## 9-6 Multiplexadores (Seletores de dados)

- Multiplexadores de dois, quatro, oito e 16 entradas estão disponíveis nas famílias lógicas TTL e CMOS.
  - Esses CIs básicos podem ser combinados para formar multiplexadores com um número maior de entradas.

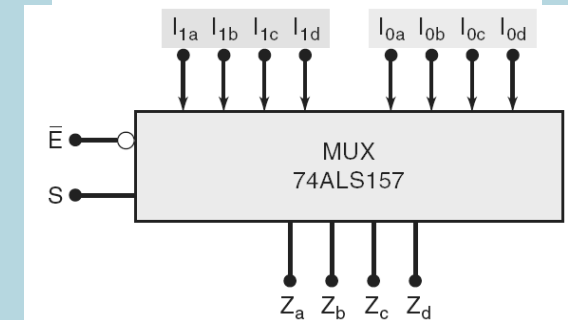
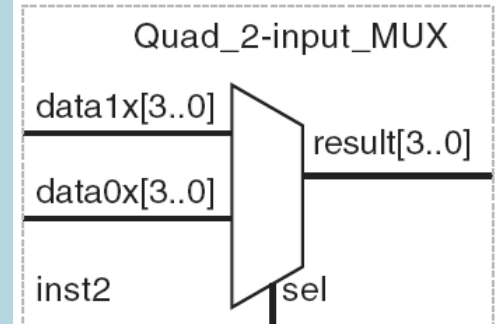


## 9-6 Multiplexadores (Seletores de dados)

O 74ALS157 contém quatro multiplexadores de duas entradas.

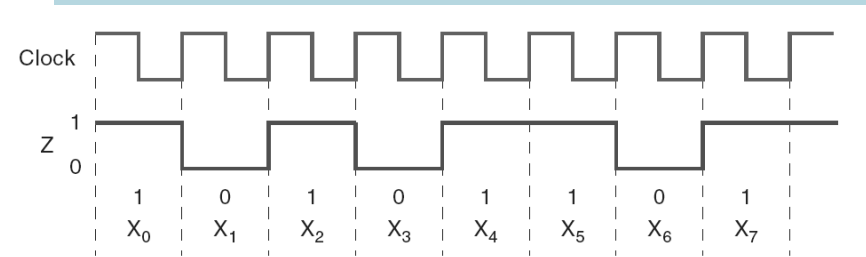
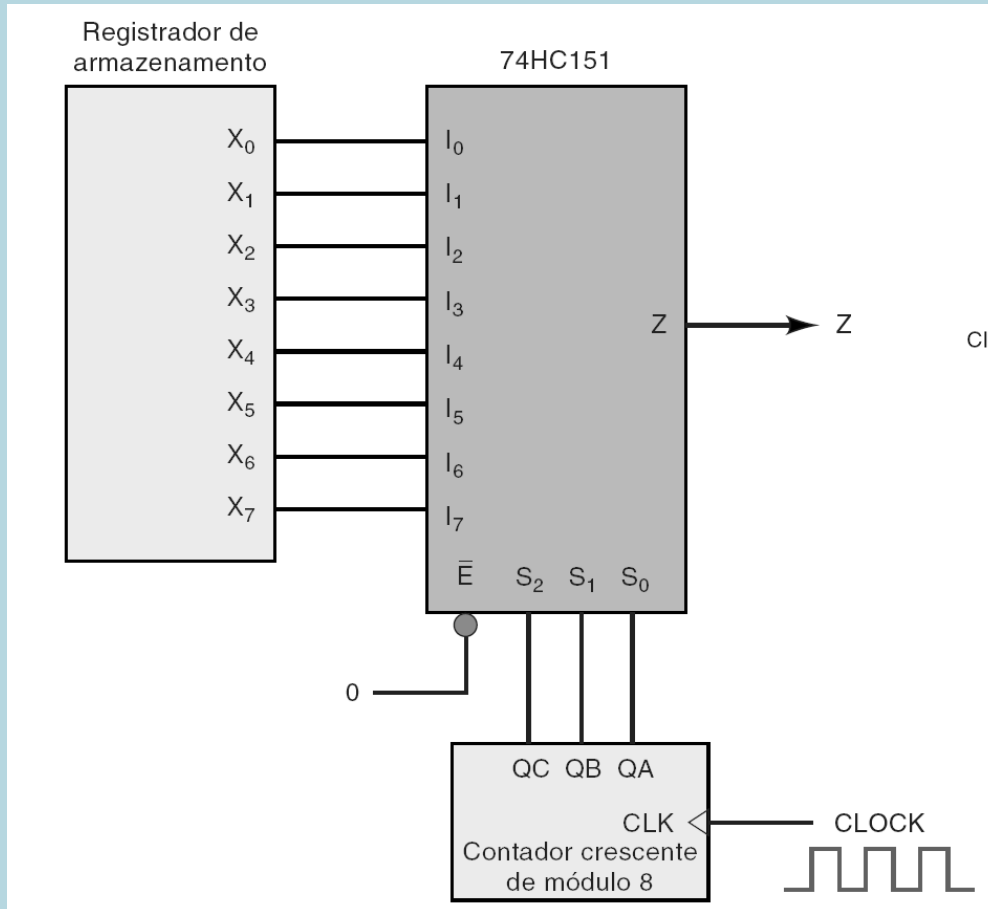


$\bar{E}$	S	$Z_a$	$Z_b$	$Z_c$	$Z_d$
H	X	L	L	L	L
L	L	$I_{0a}$	$I_{0b}$	$I_{0c}$	$I_{0d}$
L	H	$I_{1a}$	$I_{1b}$	$I_{1c}$	$I_{1d}$



## 9-7 Aplicações de multiplexadores

### Conversão paralelo-série.



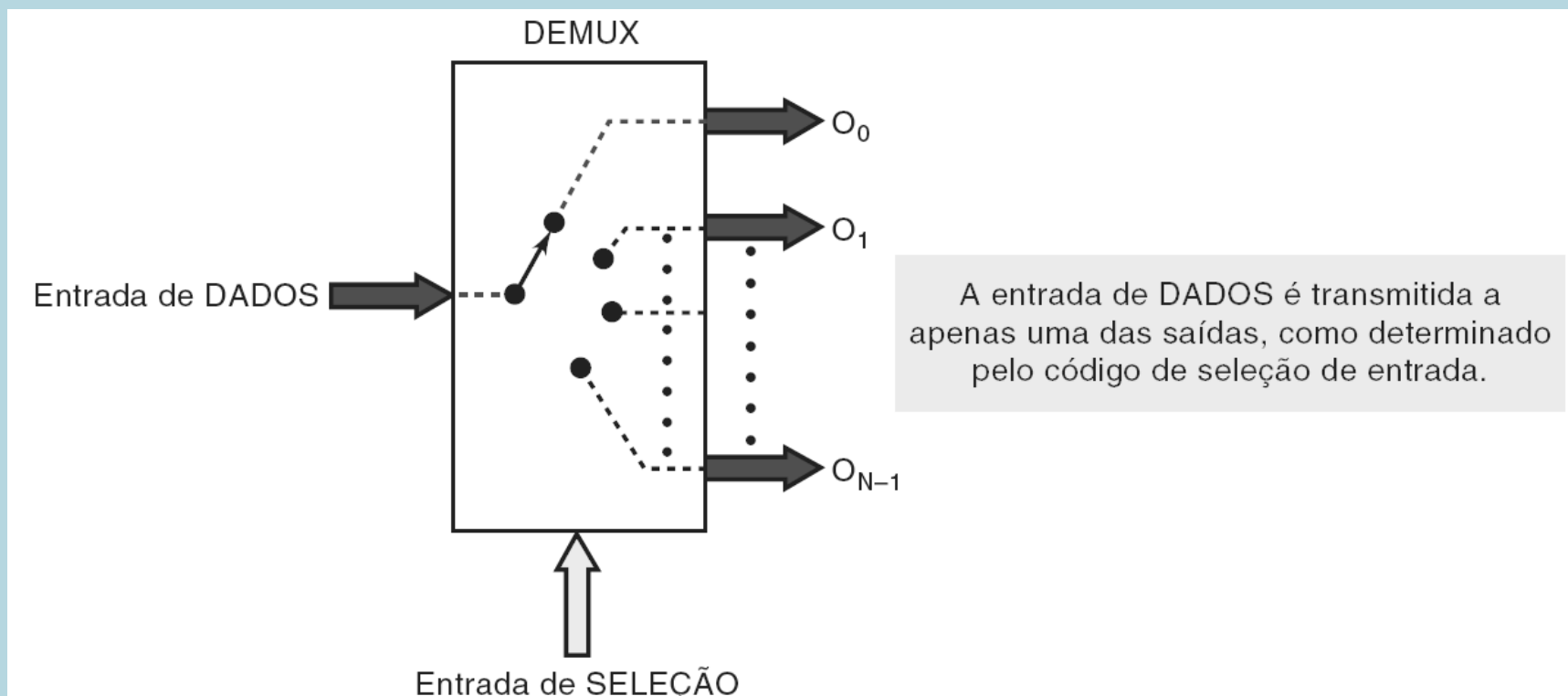
Formas de onda  
para

$$X_7 X_6 X_5 X_4 X_3 X_2 X_1$$

$$X_0 =$$

## 9-8 Demultiplexadores (Distribuidores de dados)

- Um **demultiplexador (DEMUX)** recebe uma única entrada e a distribui para várias saídas.
- O código de entrada de seleção determina para qual saída os dados de entrada serão transmitidos.

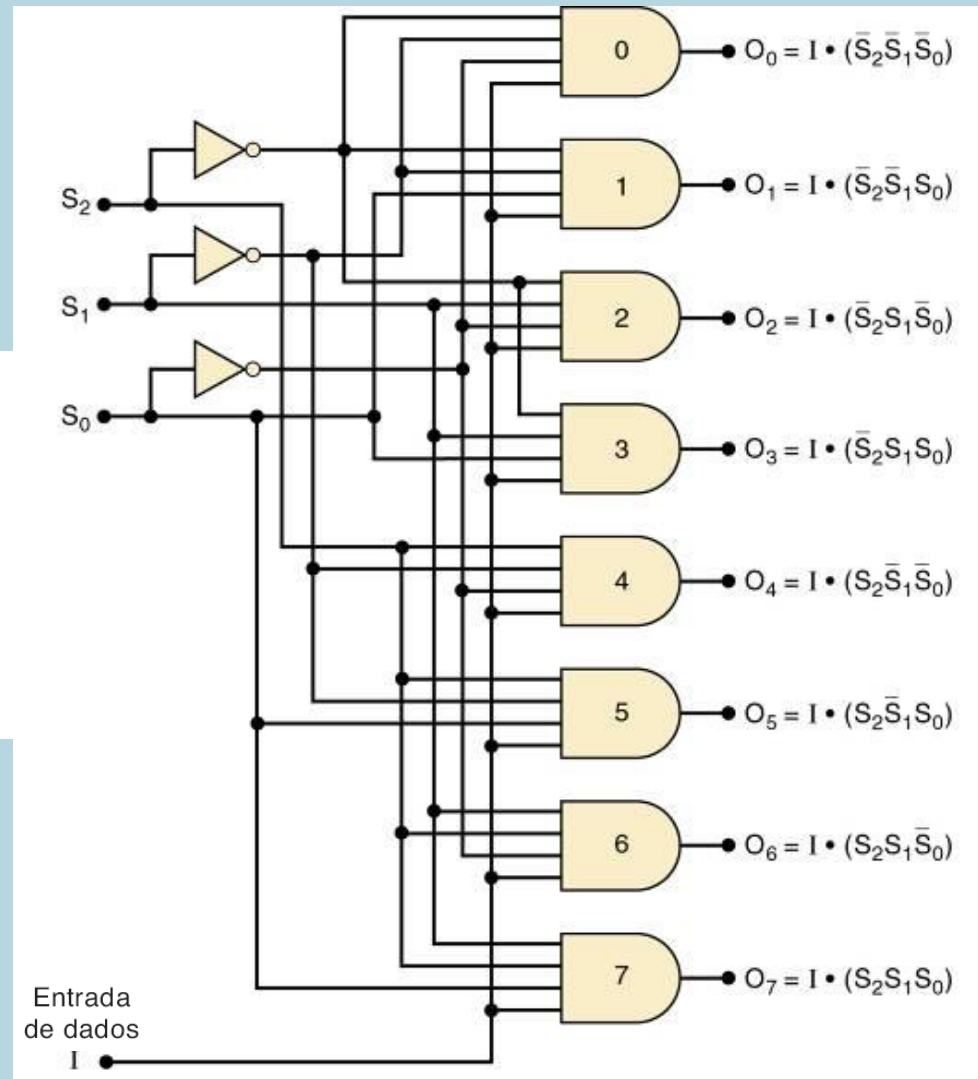


## 9-8 Demultiplexadores (Distribuidores de dados)

Um demultiplexador de 1 para 8 linhas.

Código de Seleção			Saídas							
$S_2$	$S_1$	$S_0$	$O_7$	$O_6$	$O_5$	$O_4$	$O_3$	$O_2$	$O_1$	$O_0$
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	0	1	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	1	0	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

Observação: I é a entrada de dados



## 9-12 Barramento de dados

- Nos computadores, a transferência de dados ocorre em um grupo comum de linhas de conexão chamado **barramento de dados**.
- Dispositivos conectados ao barramento de dados terão geralmente saídas tristate ou serão conectados ao barramento de dados por buffers tristate.
- Dispositivos frequentemente conectados a um barramento de dados:
  - Microprocessadores.
  - CIs de memória semicondutora.
  - Conversores digital-analógicos (DACs) e analógico-digitais (ADCs).