2. Lógica Digital

Prof. Luiz Otavio Murta Jr

Local: Depto. de Computação e Matemática (FFCLRP/USP)

Principais Tópicos

2. Revisão de Lógica Digital

- 2.1. Álgebra Booleana
- 2.2. Portas Lógicas
- 2.3. Circuitos Combinatórios
- 2.4. Circuitos Sequenciais

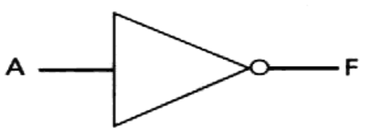
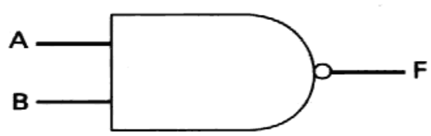
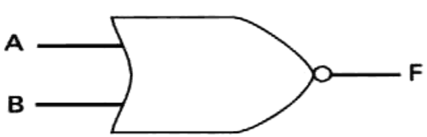
2.2. Portas Lógicas

- Bloco fundamental na construção de circuitos digitais

- Funções lógicas são implementadas através da conexão de portas lógicas

- Circuito eletrônico que produz um sinal de saída resultante de uma operação lógica sobre os sinais de entrada
  - AND, OR, NOT, NAND, NOR, XOR

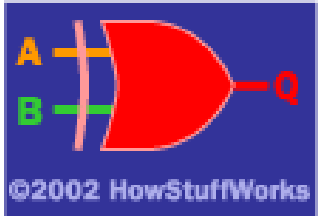
2.2. Portas Lógicas

Nome	Símbolo gráfico	Função algébrica	Tabela verdade
AND		F = A • B ou F = AB	A B | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		F = A + B	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NOT		F = $\overline{A}$ ou F = A’	A | F 0 | 1 1 | 0
NAND		F = $(\overline{AB})$	A B | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		F = $(\overline{A + B})$	A B | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0

XOR

Q=A⊕B

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0



Principais Tópicos

2. Introdução à Lógica Digital

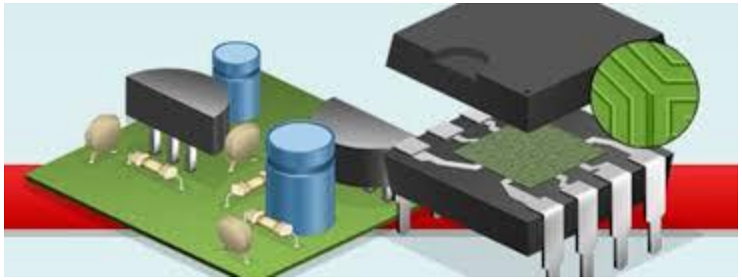
- 2.1. Álgebra Booleana
- 2.2. Portas Lógicas
- 2.3. Circuitos Combinatórios
  - 2.3.1. Multiplexadores
  - 2.3.2. Decodificadores
  - 2.3.3. Memória apenas de leitura
  - 2.3.4. Somadores
- 2.4. Circuitos Sequenciais

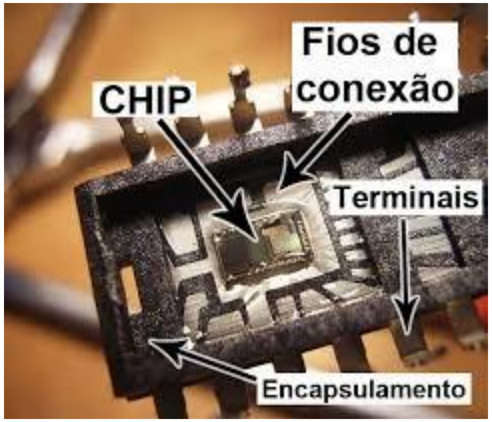
2.2. Portas Lógicas

- Um circuito digital é aquele em que estão presentes somente dois valores lógicos.
- O normal é que um sinal entre 0 e 0,5 volt represente um valor (por exemplo, 0 binário) e um sinal entre 1 e 1,5 volt represente o outro valor (por exemplo, 1 binário).
- Não são permitidas tensões fora dessas duas faixas.
- Minúsculos dispositivos eletrônicos, denominados portas (gates), podem calcular várias funções desses sinais de dois valores.

2.2. Portas Lógicas

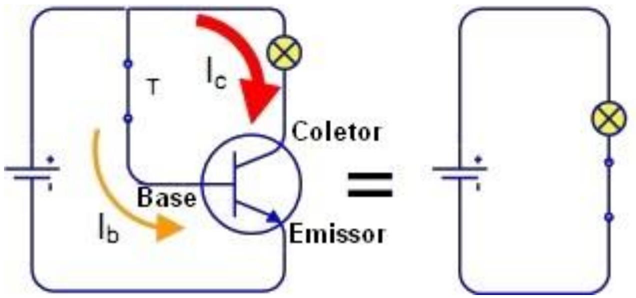
- Essas portas formam a base do hardware sobre a qual todos os computadores digitais são construídos.
- Os detalhes do funcionamento interno das portas estão fora do escopo deste curso, pois pertencem ao nível de dispositivo, que está abaixo do nível 0.
- Não obstante, agora vamos divagar um pouco e examinar rapidamente a ideia básica, que não é difícil.

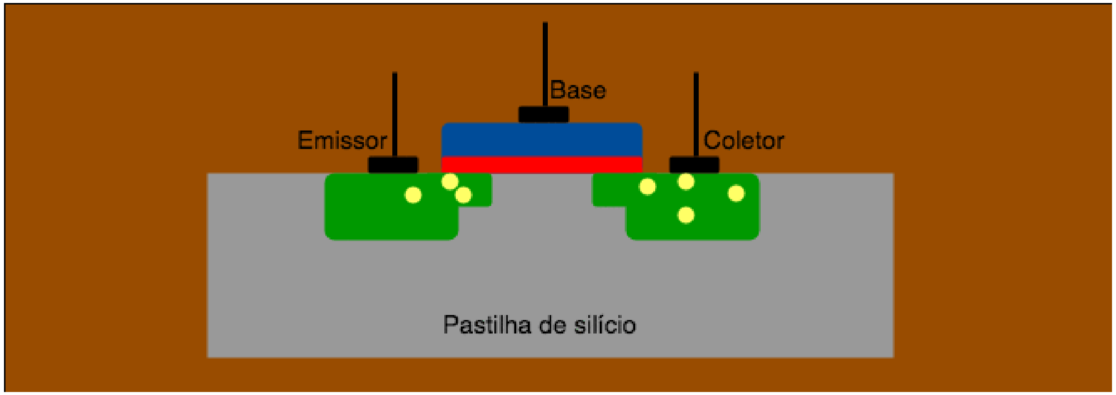




2.2. Portas Lógicas

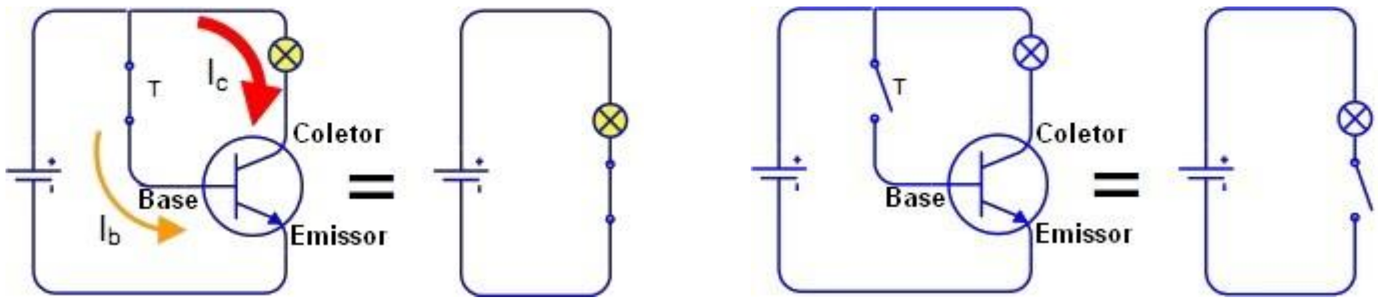
- No fundo, toda a lógica digital moderna se apoia no fato de que um transistor pode funcionar como um comutador binário muito rápido.
- Na figura (a), mostra um transistor bipolar (representado pelo círculo) inserido em um circuito simples.
- Esse transistor tem três conexões com o mundo exterior:
- o coletor, a base e o emissor.





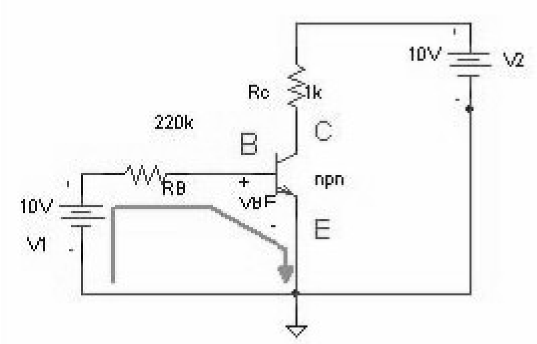
2.2. Portas Lógicas

- Quando a voltagem de entrada, V_{in}, está abaixo de certo valor crítico, o transistor desliga e age como uma resistência infinita.
- Isso faz com que a saída do circuito, V_{out}, assuma um valor próximo a V_{cc}, uma voltagem regulada externamente, em geral +1,5 volt para esse tipo de transistor.
- Quando V_{in} excede o valor crítico, o transistor liga e age como um fio, fazendo V_{out} ficar conectado com a terra (por convenção, 0 volt).



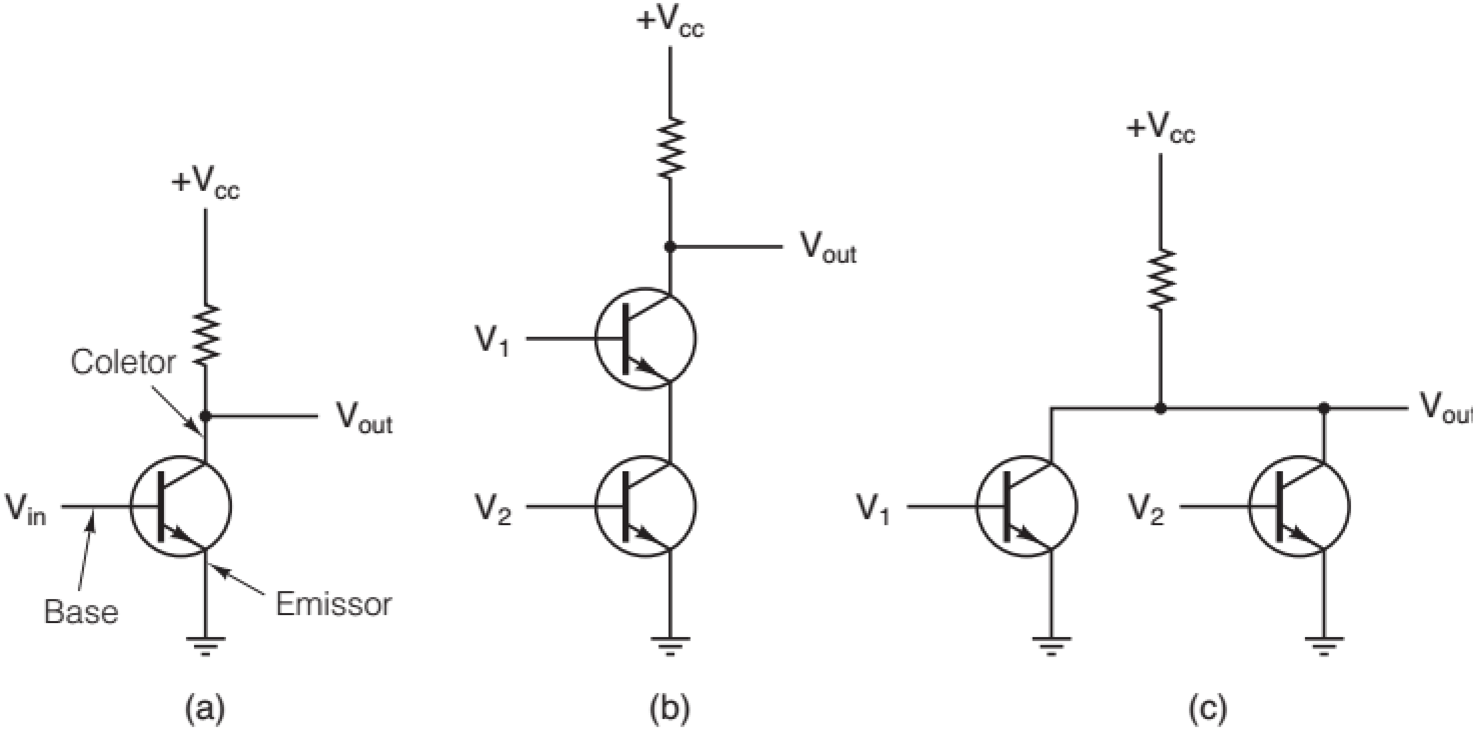
2.2. Portas Lógicas

- O importante é notar que, quando V_{in} é baixa, V_{out} é alta, e vice-versa.
- Assim, esse circuito é um inversor, que converte um 0 lógico em um 1 lógico e um 1 lógico em um 0 lógico.
- O resistor (linha serrilhada) é necessário para limitar a quantidade de corrente drenada pelo transistor, de modo que ele não queime.
- O tempo típico exigido para passar de um estado para outro é tipicamente de um nanossegundo ou menos.



2.2. Portas Lógicas

- (a) Inversor de transistor. (b) Porta nand. (c) Porta nor.



2.2. Portas Lógicas

- Na figura (b), dois transistores estão ligados em série.
- Se ambas, V_1 e V_2, forem altas, ambos os transistores conduzirão e V_{out} cairá.
- Se qualquer das entradas for baixa, o transistor correspondente se desligará e a saída será alta.
- Em outras palavras, V_{out} será baixa se, e somente se, ambas, V_1 e V_2, forem altas.
- Na figura (c), os dois transistores estão ligados em paralelo em vez de em série.

2.2. Portas Lógicas

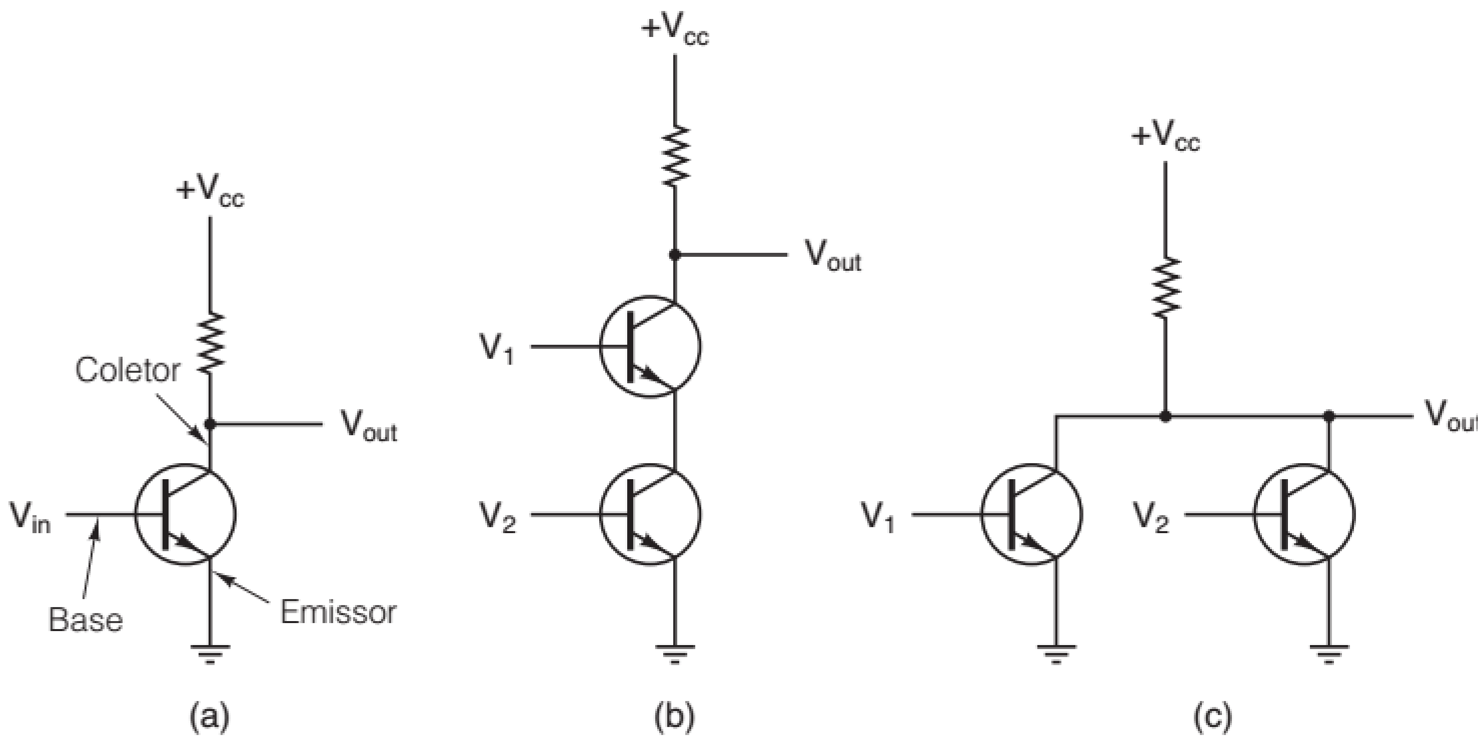
- Nessa configuração, se qualquer das entradas for alta, o transistor correspondente ligará e conectará a saída com a terra.
- Se ambas as entradas forem baixas, a saída permanecerá alta.
- Esses três circuitos, ou seus equivalentes, formam as três portas mais simples e são denominadas portas not, nand e nor, respectivamente.
- Portas not costumam ser denominadas inversoras.

2.2. Portas Lógicas

- Se agora adotarmos a convenção de que “alta” (V_{cc} volts) é um 1 lógico e “baixa” (terra) é um 0 lógico, podemos expressar o valor de saída como uma função dos valores de entrada.
- Os símbolos usados para representar essas portas são mostrados nas figuras (a)-(c) junto com o comportamento funcional de cada circuito.
- Nessas figuras, A e B são entradas e X é a saída. Cada linha especifica a saída para uma combinação diferente das entradas.

2.2. Portas Lógicas

- (a) Inversor de transistor. (b) Porta nand. (c) Porta nor.



2.2. Portas Lógicas

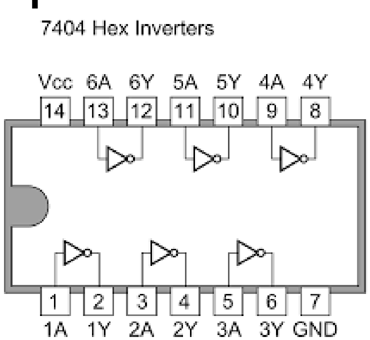
- Se o sinal de saída da figura (b) for alimentado em um circuito inversor,
  - obtemos outro circuito com o inverso exato da porta nand, a saber,
  - um cuja saída é 1 se,
  - e somente se, ambas as entradas forem 1.
- Esse circuito é denominado uma porta and;
  - seu símbolo e descrição funcional são dados na próxima figura (d).

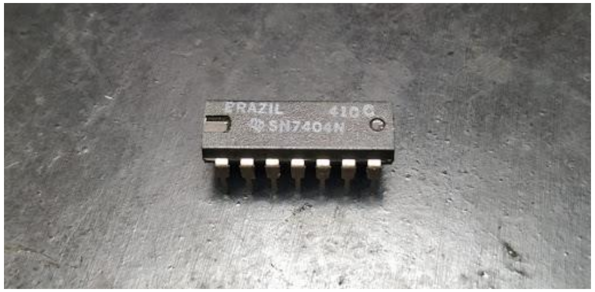
2.2. Portas Lógicas

- De modo semelhante,
  - a porta nor pode ser conectada a um inversor para produzir um circuito cuja saída é 1 se quaisquer das saídas,
  - ou ambas, for um 1, mas 0 se ambas as entradas forem 0.
- O símbolo e a descrição funcional desse circuito, denominado uma porta or, são dados na figura (e).

2.2. Portas Lógicas

- Os pequenos círculos usados como parte dos símbolos para o inversor, porta nand e porta nor, são denominados bolhas de inversão.
- Também são usadas em outros contextos para indicar um sinal invertido.
- As cinco portas da figura são os principais elementos de construção do nível lógico digital.
- A discussão precedente deve ter deixado claro que as portas nand e nor requerem dois transistores cada, ao passo que as portas and e or requerem três cada.





2.2. Portas Lógicas

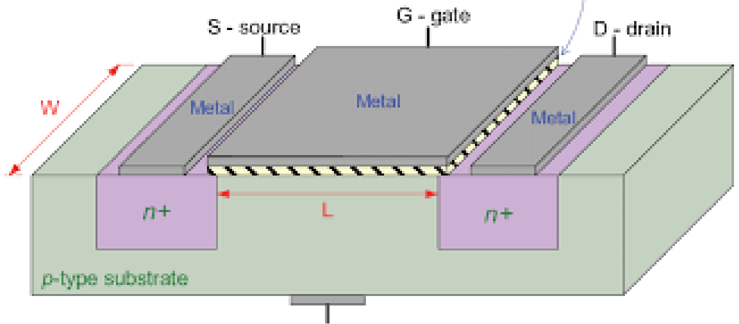
- Por essa razão, muitos computadores são baseados em portas nand e nor em vez das portas mais conhecidas, and e or.
  - (Na prática, todas as portas são executadas de modo um pouco diferente, mas as nand e nor ainda são mais simples do que as and e or.)
- A propósito, vale a pena observar que as portas podem perfeitamente ter mais de duas entradas.
- Em princípio, uma porta nand, por exemplo, pode ter, arbitrariamente, muitas entradas, mas na prática não é comum encontrar mais de oito.

2.2. Portas Lógicas

- Embora a questão do modo como são construídas as portas pertença ao nível do dispositivo,
  - gostaría de mencionar as principais famílias de tecnologia de fabricação porque elas são citadas com muita frequência.
- As duas tecnologias principais são bipolar e MOS (Metal Oxide Semiconductor – semicondutor de óxido metálico).

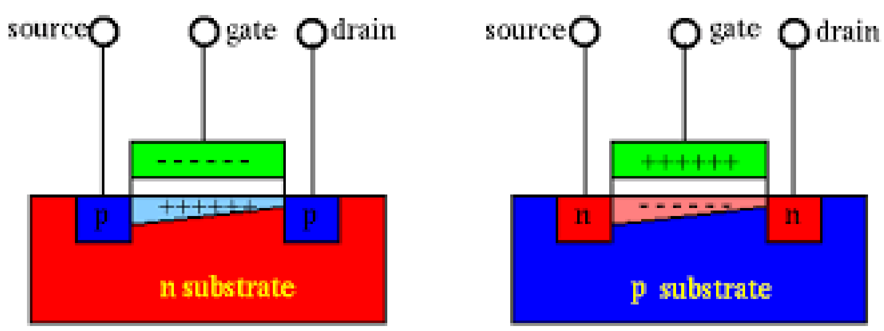
2.2. Portas Lógicas

- Os dois principais tipos bipolares são a TTL
  - (Transistor-Transistor Logic – lógica transistor-transistor),
    - que há muitos anos é o “burro de carga” da eletrônica digital,
  - e a ECL (Emitter-Coupled Logic – lógica de emissor acoplado),
    - que era usada quando se requeria uma operação de velocidade muito alta.
- Para circuitos de computador, o que predomina agora é a tecnologia MOS.



2.2. Portas Lógicas

- Portas MOS são mais lentas do que as TTL e ECL,
  - mas exigem bem menos energia elétrica e ocupam um espaço muito menor,
  - portanto, um grande número delas pode ser compactado e empacotado.
- Há muitas variedades de MOS, entre as quais PMOS, NMOS e CMOS.



pMOS nMOS

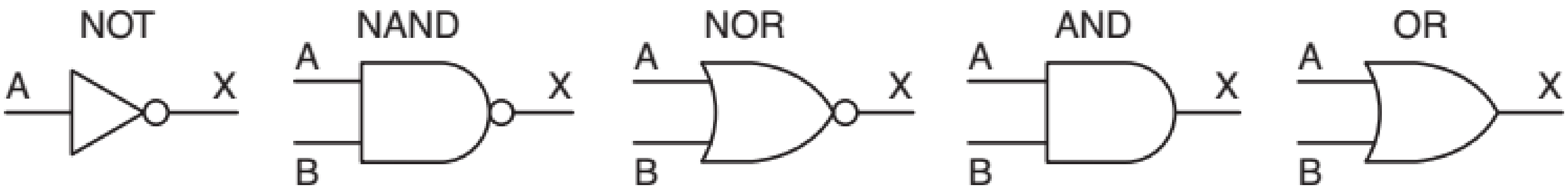
- **Substrato N:** dopado com doadores de elétrons (Fósforo, Arsênio)
- **Substrato P:** dopado com receptores de elétrons ou lacunas (Boro, Galio)

2.2. Portas Lógicas

- Embora os modos de construção dos transistores MOS e dos transistores bipolares sejam diferentes, sua capacidade de funcionar como comutadores eletrônicos é a mesma.
- A maioria das CPUs e memórias modernas usa tecnologia CMOS, que funciona a +1,5 volt.
- E isso é tudo o que diremos sobre o nível de dispositivo.

2.2. Portas Lógicas

- Símbolos e comportamento funcional das cinco portas básicas.



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

2.2. Portas Lógicas

- Para descrever os circuitos que podem ser construídos combinando portas,
    - é necessário um novo tipo de álgebra, no qual variáveis e funções podem assumir somente os valores 0 e 1.
- Essa álgebra é denominada álgebra booleana, nome que se deve a seu descobridor ou inventor, o matemático inglês George Boole (1815–1864).

2.2. Portas Lógicas

- Em termos estritos, estamos nos referindo a um tipo específico de álgebra booleana, uma álgebra de comutação,
  - o termo “álgebra booleana” é tão utilizado no lugar de “álgebra de comutação” que não fazemos distinção.
- Assim como há funções na álgebra “ordinária” (isto é, a álgebra do colegial), também há funções na álgebra booleana.
- Uma função booleana tem uma ou mais variáveis de entrada e produz um resultado que depende somente dos valores dessas variáveis.

2.2. Portas Lógicas

- Uma função simples, f, pode ser definida ao se dizer que f(A) é 1 se A for 0 e f(A) é 0 se A for 1.
- Essa função é a função not da figura (a).
- Como uma função booleana de n variáveis só tem 2^n combinações possíveis de valores de entrada,
  - ela pode ser completamente descrita por uma tabela com 2^n linhas, na qual cada linha informa o valor da função para uma combinação diferente de valores de entrada.
- Ela é denominada tabela verdade.

2.2. Portas Lógicas

- As tabelas da figura são todas exemplos de tabelas verdade.
- Se concordarmos em sempre listar as linhas de uma tabela verdade em ordem numérica (base 2),
  - isto é, para duas variáveis na ordem 00, 01, 10 e 11,
  - a função pode ser completamente descrita pelo número binário de 2n bits obtido pela leitura vertical da coluna de resultado da tabela verdade.
- Assim, nand é 1110, nor é 1000, and é 0001 e or é 0111.

2.2. Portas Lógicas

- É óbvio que só existem 16 funções booleanas de duas variáveis, correspondentes às 16 possíveis sequências de 4 bits resultantes.
- Por outro lado,
  - a álgebra ordinária tem um número infinito de funções de duas variáveis,
  - nenhuma das quais pode ser descrita por meio de uma tabela de saídas para todas as entradas possíveis,
  - porque cada variável pode assumir qualquer valor de um número infinito de valores possíveis.

2.2. Portas Lógicas

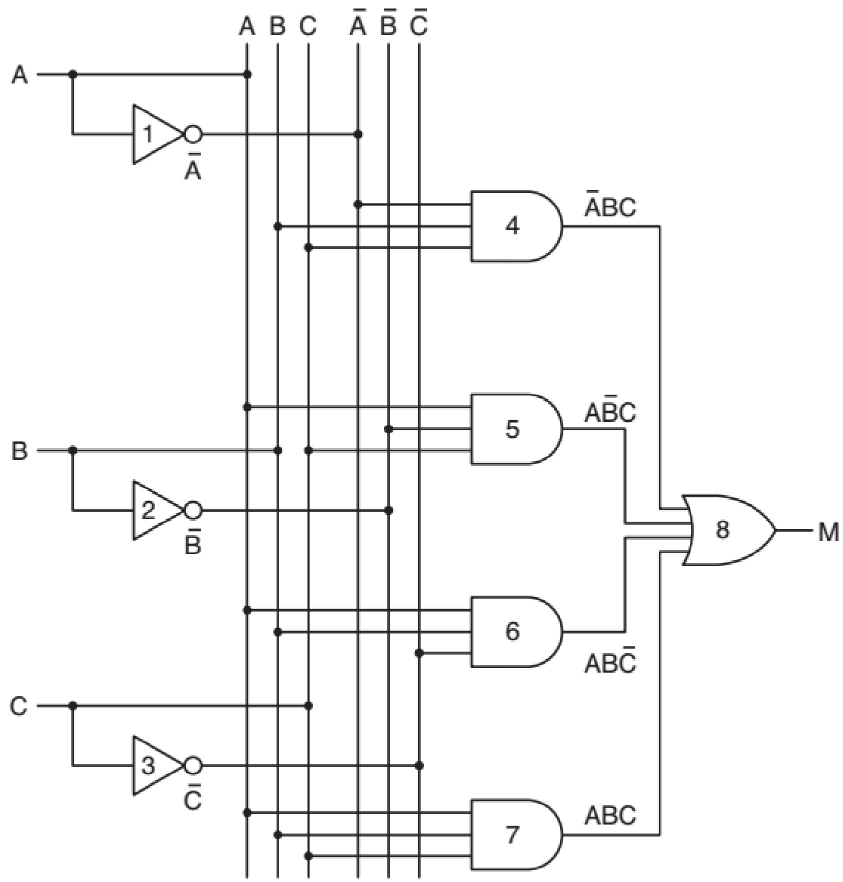
- A próxima figura (a) mostra a tabela verdade para uma função booleana de três variáveis: M = f(A, B, C).
- Essa função é a de lógica majoritária, isto é, ela é 0 se a maioria de suas entradas for 0, e 1 se a maioria de suas entradas for 1.
- Embora qualquer função booleana possa ser completamente especificada dada sua tabela verdade,
  - à medida que aumenta o número de variáveis,
  - essa notação fica cada vez mais trabalhosa.

2.2. Portas Lógicas

- (a) Tabela verdade para a função majoritária de três variáveis. (b) Circuito para (a).

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(a)



(b)

2.2. Portas Lógicas

- Para ver como ocorre essa outra notação,
  - observe que qualquer função booleana pode ser especificada ao se dizer quais combinações de variáveis de entrada dão um valor de saída igual a 1.
- Para a função da figura (a),
  - há quatro combinações de variáveis de entrada que fazem com que M seja 1.
- Por convenção, marcaremos a variável de entrada com uma barra para indicar que seu valor é invertido.
- A ausência de uma barra significa que o valor não é invertido.

2.2. Portas Lógicas

- Além disso, usaremos a multiplicação implícita ou um ponto para representar a função booleana and e + para representar a função booleana or.
  - Assim, por exemplo, $A\overline{B}C$ assume o valor 1 somente quando A = 1 e B = 0 e C = 1.
- Além disso,
  - $A\overline{B} + B\overline{C}$ é 1 somente quando (A = 1 e B = 0) ou (B = 1 e C = 0).
- As quatro linhas da figura (a) que produzem bits 1 na saída são:
  - $\overline{A}BC$, $A\overline{B}C$, $AB\overline{C}$ e ABC.

2.2. Portas Lógicas

- Ela é descrita por uma tabela verdade ou por uma função booleana como
  - $F = A\overline{B}C + AB\overline{C}$
  - Uma função booleana pode ser executada por um circuito eletrônico
  - usando sinais que representam as variáveis de entrada e saída e portas como and, or e not.
- Em geral, empregaremos a notação and, or e not quando nos referirmos aos operadores booleanos,
  - e and, or e not quando nos referirmos a portas,
  - embora essa notação quase sempre seja ambígua em se tratando de indicar funções ou portas.

2.2. Portas Lógicas

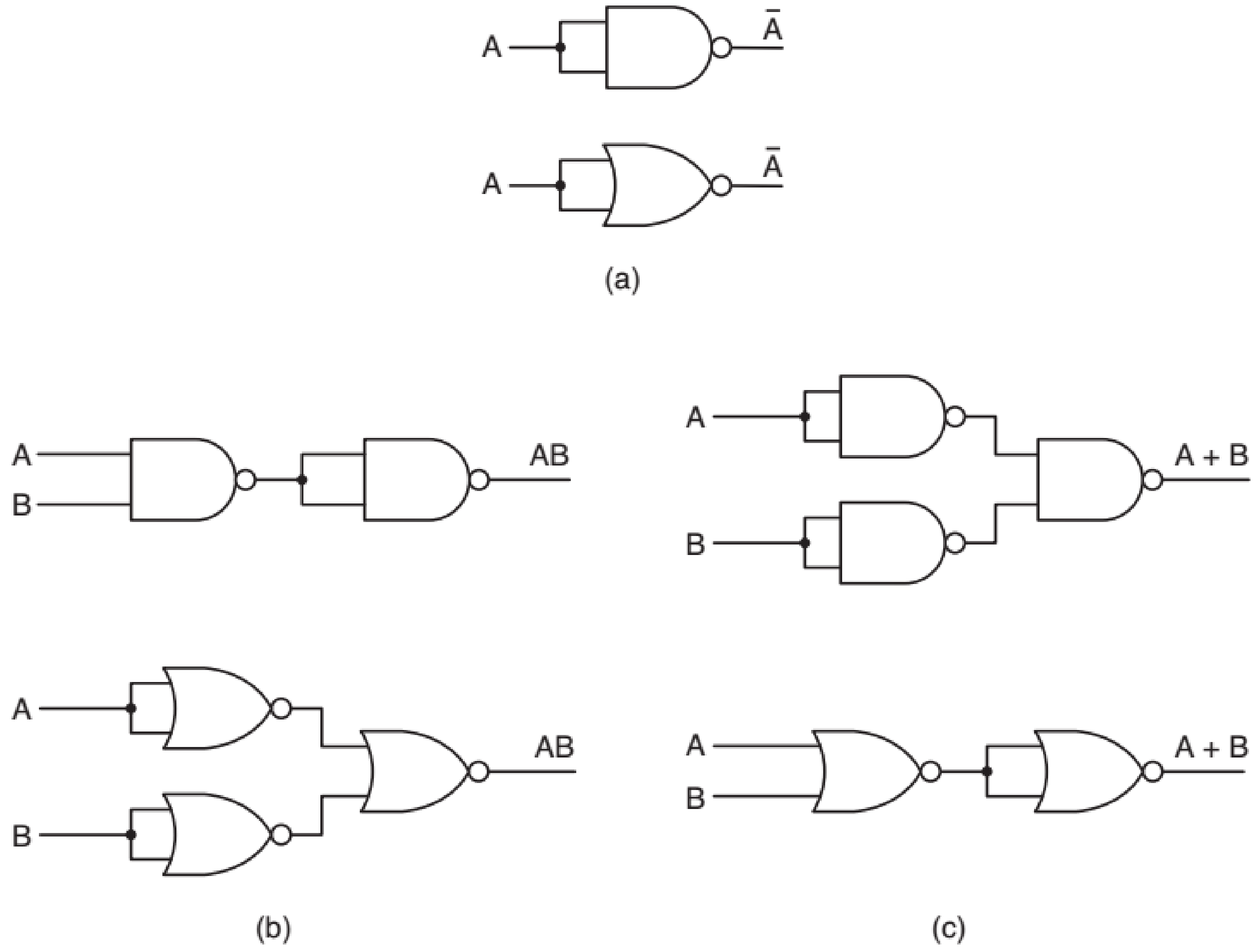
- É importante ter em mente a distinção entre uma função booleana abstrata e sua execução por um circuito eletrônico.
- Uma função booleana consiste em variáveis,
  - como A, B e C, e operadores booleanos,
  - como and, or e not.

2.2. Portas Lógicas

- Em geral, empregaremos a notação and, or e not quando nos referirmos aos operadores booleanos,
  - e and, or e not quando nos referirmos a portas,
  - embora essa notação quase sempre seja ambígua em se tratando de indicar funções ou portas.

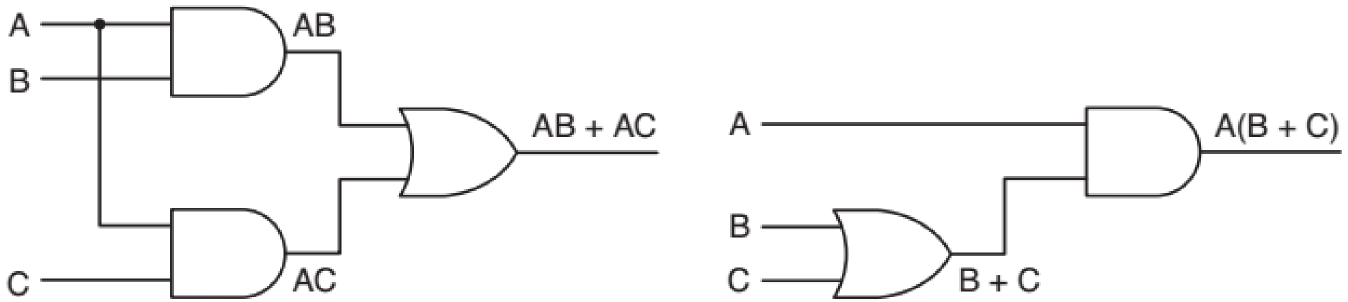
2.2. Portas Lógicas

- Construção de portas (a) not, (b) and e (c) or usando somente portas nand ou somente portas nor.



2.2. Portas Lógicas

- Duas funções equivalentes. (a) AB + AC. (b) A(B + C).



A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

(a)

A	B	C	A	B + C	A(B + C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

(b)

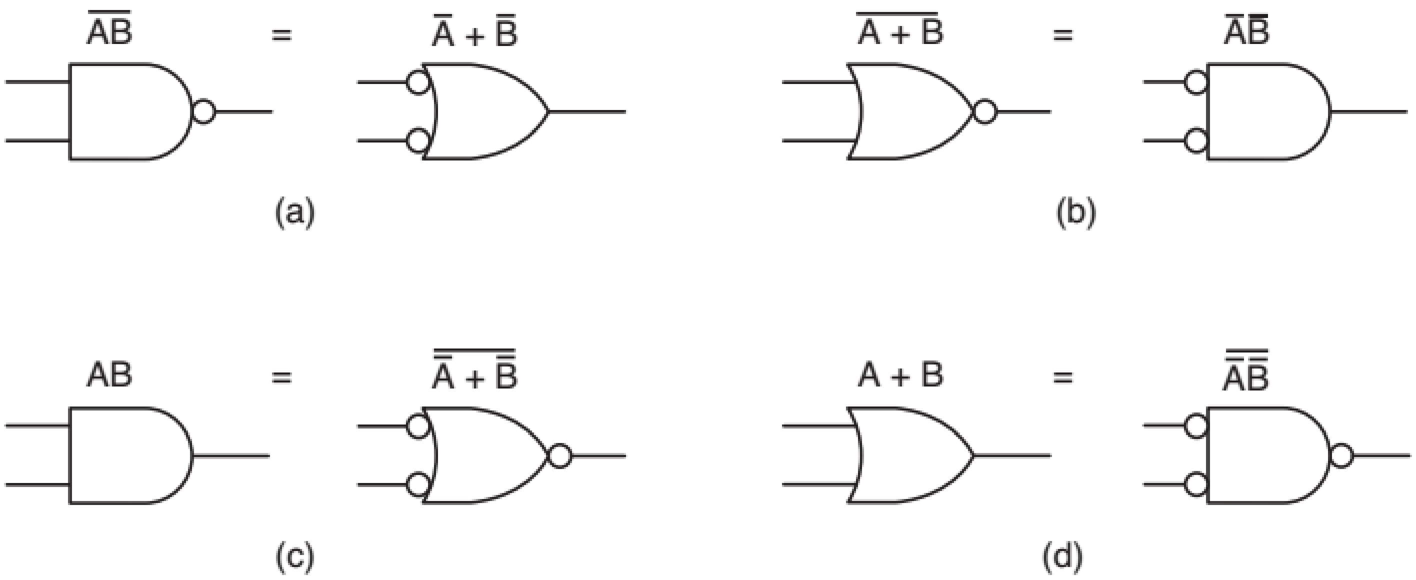
2.2. Portas Lógicas

- Algumas identidades da álgebra Booleana.

Nome	Forma AND	Forma OR
Lei da identidade	$1A = A$	$0 + A = A$
Lei do elemento nulo	$0A = 0$	$1 + A = 1$
Lei idempotente	$AA = A$	$A + A = A$
Lei do inverso	$A\overline{A} = 0$	$A + \overline{A} = 1$
Lei comutativa	$AB = BA$	$A + B = B + A$
Lei associativa	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Lei distributiva	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Lei da absorção	$A(A + B) = A$	$A + AB = A$
Lei de De Morgan	$\overline{A}\,\overline{B} = \overline{A} + \overline{B}$	$\overline{A} + \overline{B} = \overline{A}\,\overline{B}$

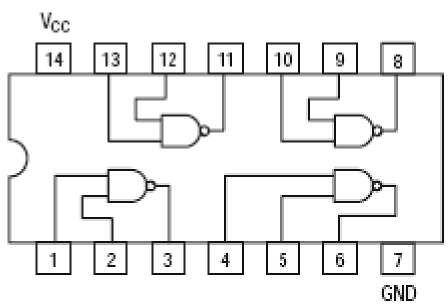
2.2. Portas Lógicas

- Símbolos alternativos para algumas portas: (a) nand. (b) nor. (c) and. (d) or.

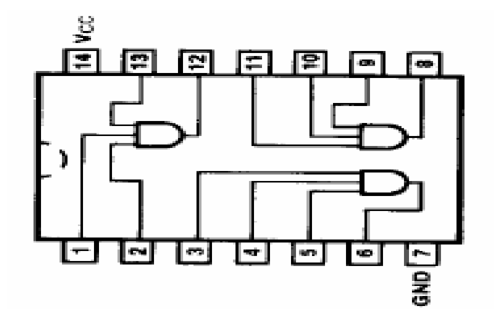


2.2. Portas Lógicas

- **Exemplos de Circuitos Integrados com portas lógicas**
  - 7400
    - 4 portas NAND com duas entradas cada



  - 7411
    - 3 portas AND com três entradas cada



2.2. Portas Lógicas

- **Geralmente, nem todos os tipos de portas lógicas são utilizados em implementações**
  - O projeto se torna mais simples se poucos tipos de portas são utilizados
  - Algumas portas utilizam menos componentes eletrônicos que outras

- **Conjuntos de portas lógicas funcionalmente completos**
  - Qualquer porta lógica pode ser implementada utilizando-se apenas as portas deste conjunto

2.2. Portas Lógicas

- **Conjuntos de portas funcionalmente completos**
  1. AND, OR, NOT
  2. AND, NOT
  3. OR, NOT
  4. NAND
  5. NOR

2.2. Portas Lógicas

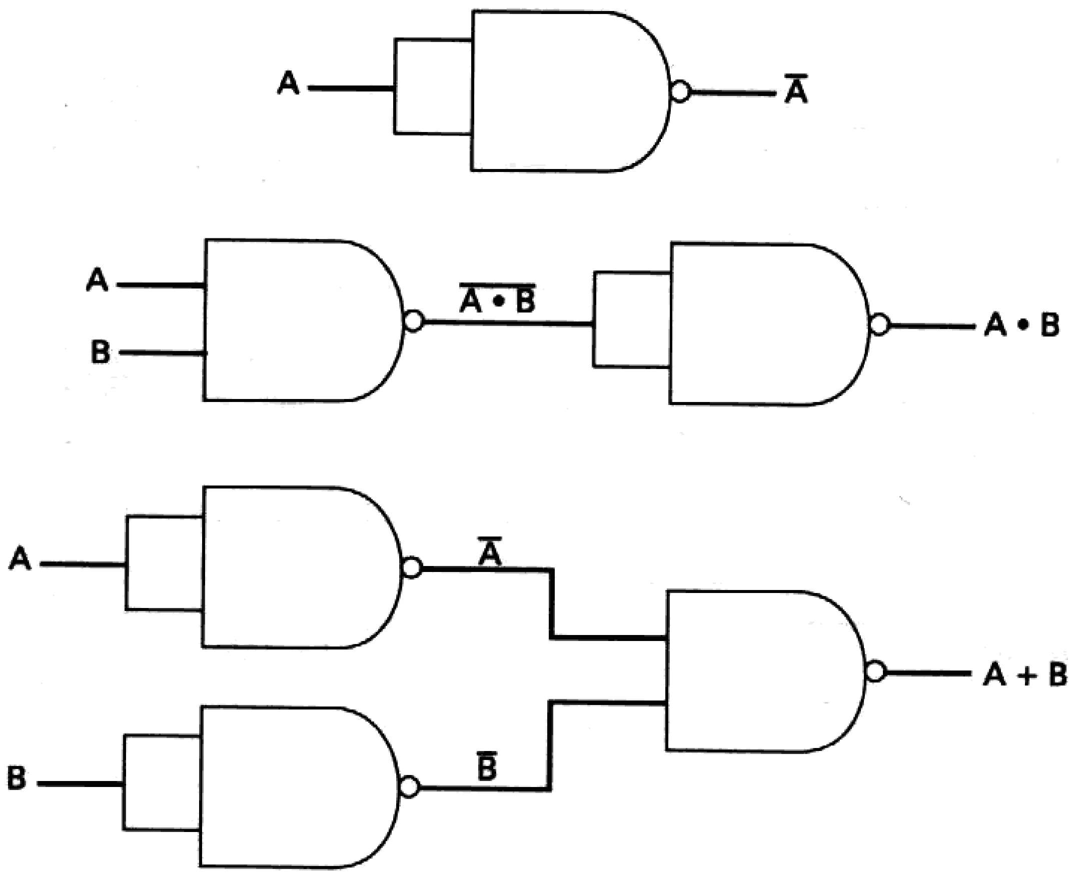
1. **AND, OR, NOT**
   - Representam as três operações básicas
   - As operações NAND, NOR e XOR podem ser facilmente deduzidas a partir destas três operações
2. **AND, NOT**
   - Basta expressar a operação OR
     - Usando a seguinte lei de DeMorgan

$$A + B = \overline{\overline{A} \bullet \overline{B}}$$

3. **OR, NOT**
   - Basta expressar a operação AND
     - Usando a seguinte lei de DeMorgan

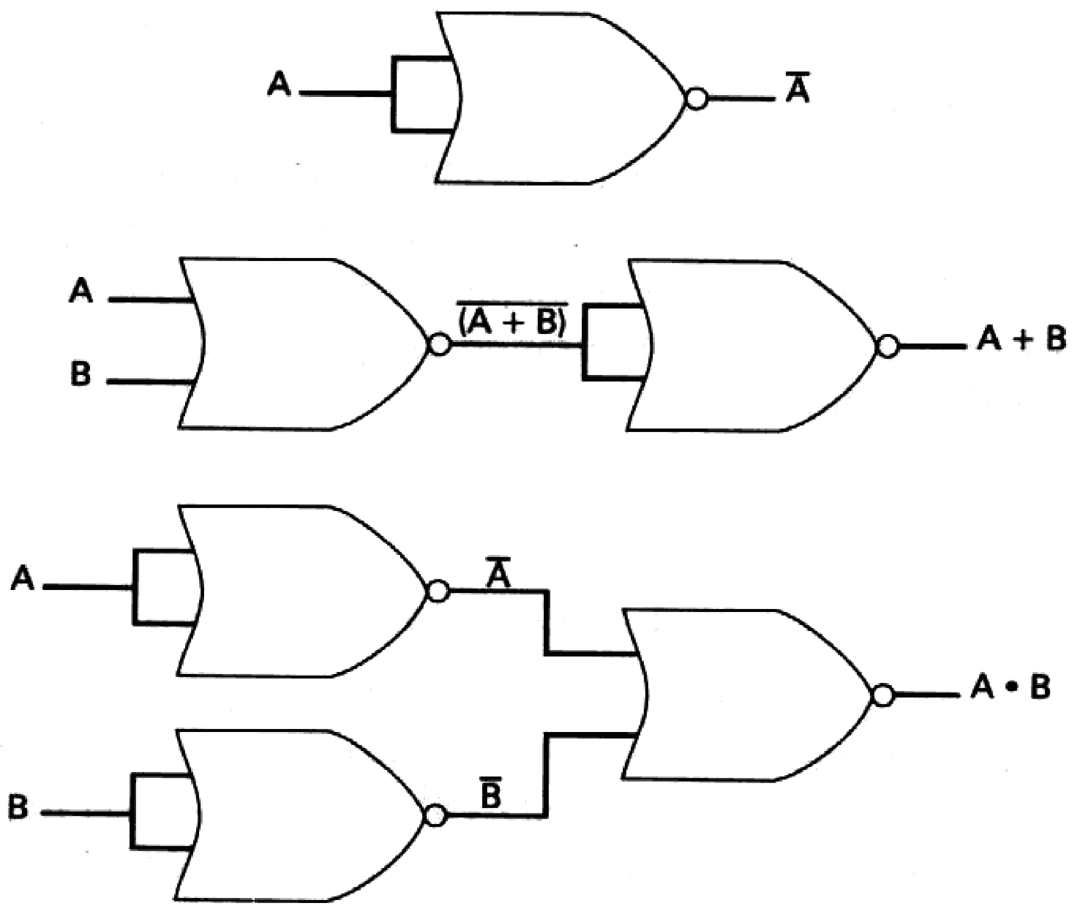
$$A \bullet B = \overline{\overline{A} + \overline{B}}$$

4. NAND


A
$\overline{A}$
A
B
$\overline{A \cdot B}$
$A \cdot B$
A
$\overline{A}$
B
$\overline{B}$
$A + B$

2.2. Portas Lógicas

5. NOR



2.2. Portas Lógicas

- Portas não são fabricadas nem vendidas individualmente,
  - mas em unidades denominadas circuitos integrados,
  - muitas vezes denominados ICs ou chips.
- Um IC é um pedaço quadrado de silício de tamanho variado,
  - dependendo de quantas portas são necessárias para executar os componentes do chip.
- Substratos pequenos medirão cerca de 2 × 2 mm, enquanto os maiores podem ter até 18 × 18 mm.

2.2. Portas Lógicas

- ICs costumam ser montados em pacotes retangulares de plástico ou cerâmica,
  - que podem ser muito maiores que os substratos que eles abrigam,
  - se forem necessários muitos pinos para conectar o chip ao mundo exterior.
- Cada pino se conecta com a entrada ou saída de alguma porta no chip ou à fonte de energia, ou ao terra.
- A figura mostra uma série de pacotes de IC comuns, usados para os chips de hoje.

2.2. Portas Lógicas

- Chips menores, como os usados para microcontroladores domésticos ou chips de RAM,
  - usarão pacotes duplos em linha (DIPs – Dual Inline Packages).
- Um DIP é um pacote com duas fileiras de pinos que se encaixam em um soquete correspondente na placa-mãe.
- Os pacotes mais comuns têm 14, 16, 18, 20, 22, 24, 28, 40, 64 ou 68 pinos.
- Para chips grandes costumam ser usados pacotes quadrados com pinos nos quatro lados ou na parte de baixo.

2.2. Portas Lógicas

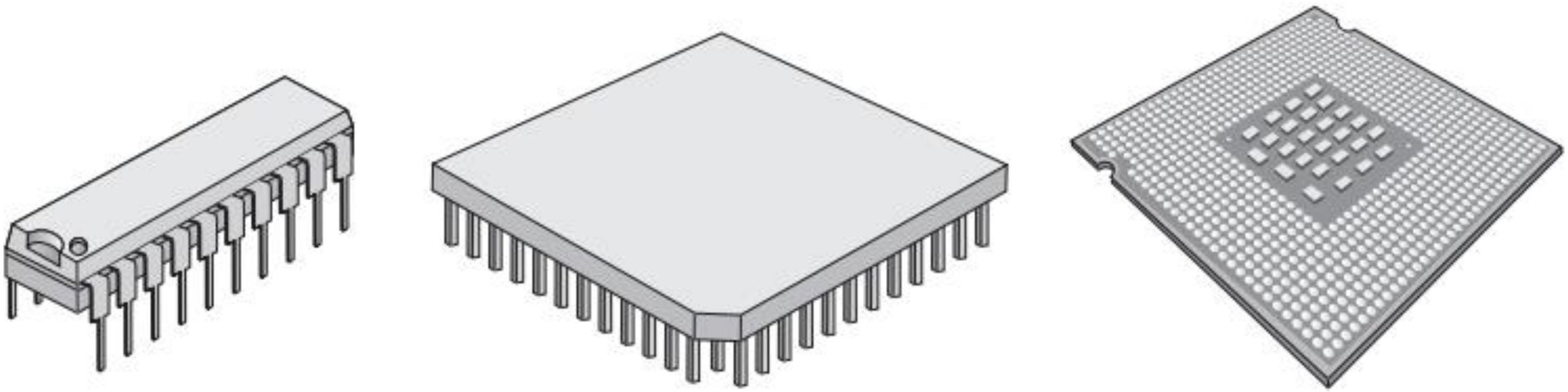
- Dois pacotes comuns para chips maiores são
  - Pin Grid Arrays, ou PGAs,
  - e Land Grid Arrays, ou LGAs.
- PGAs possuem pinos na parte inferior do pacote,
  - que se encaixam em um soquete correspondente na placa-mãe.
- Soquetes PGA normalmente utilizam um mecanismo com força de inserção nula,
  - onde uma alavanca aplica pressão lateral sobre todos os pinos do PGA, mantendo-o firmemente no soquete PGA.

2.2. Portas Lógicas

- LGAs, por outro lado, possuem pequenas plataformas planas na parte inferior do chip,
  - e um soquete LGA terá uma capa que se encaixa sobre o LGA e aplica uma força para baixo no chip,
  - garantindo que todas as plataformas do LGA façam contato com as plataformas do soquete LGA.

2.2. Portas Lógicas

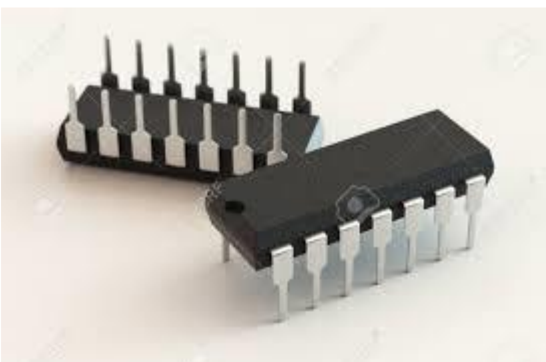
- Tipos comuns de pacotes de circuito integrado, incluindo um pacote dual-in-line, ou DIP (a), PGA (b) e LGA (c).



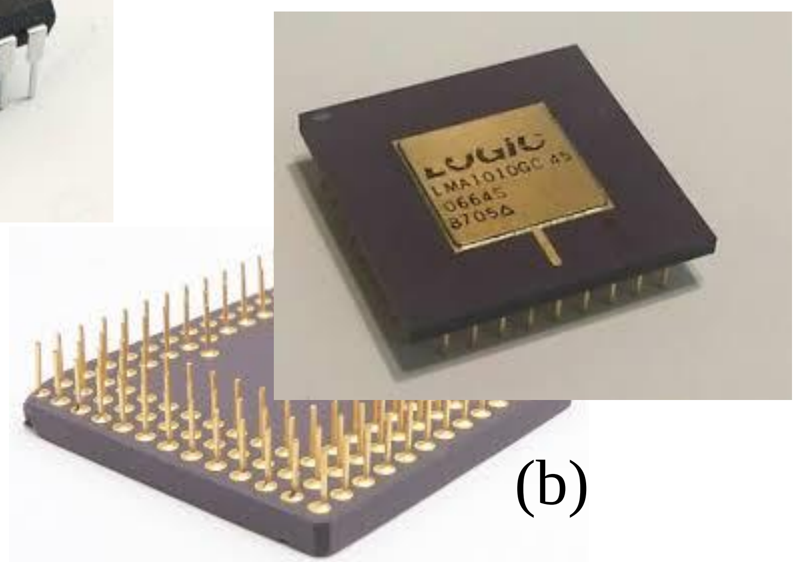
(a) (b) (c)

2.2. Portas Lógicas

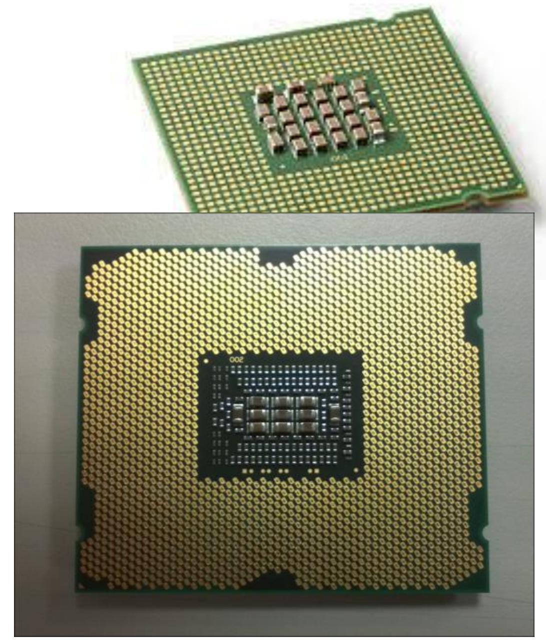
- Tipos comuns de pacotes de circuito integrado, incluindo um pacote dual-in-line, ou DIP (a), PGA (b) e LGA (c).



(a)



(b)



(c)